Reusable Fuzzy Extractor from LWE

Yunhua Wen[1,2] and Shengli Liu[1,2,3](✉)

[1] Department of Computer Science and Engineering, Shanghai Jiao Tong University,
Shanghai 200240, China
{happyle8,slliu}@sjtu.edu.cn
[2] State Key Laboratory of Cryptology, P.O. Box 5159, Beijing 100878, China
[3] Westone Cryptologic Research Center, Beijing 100070, China

Abstract. Fuzzy extractor converts the reading of a noisy non-uniform source to a reproducible and almost uniform output R. The output R in turn is used in some cryptographic system as a secret key. To enable multiple extractions of keys $\mathsf{R}_1, \mathsf{R}_2, \ldots, \mathsf{R}_\rho$ from the same noisy non-uniform source and applications of different R_i, the concept of reusable fuzzy extractor is proposed to guarantee the pseudorandomness of R_i even conditioned on other extracted keys R_j (from the same source).

In this work, we construct a reusable fuzzy extractor from the Learning With Errors (LWE) assumption. Our reusable fuzzy extractor provides resilience to linear fraction of errors. Moreover, our construction is simple and efficient and imposes no special requirement on the statistical structure of the multiple readings of the source.

Keywords: Fuzzy extractor · Reusability · The LWE assumption

1 Introduction

In a cryptographic system, it is assumed that the secret key is sampled from a random source and uniformly distributed, since the security of the system heavily relies on the uniformity of the secret key. In reality, such a uniform secret key is hard to create, remember or store by users of the system. On the other hand, there are lots of random sources available like biometric data (fingerprint, iris, etc.), physical unclonable function (PUF) [17,18], or quantum information [4,19]. These sources do not provide uniform distributions though they may possess high entropy. Moreover, the readings of the source may introduce errors and only result in noisy versions. To address the issues, *fuzzy extractor* [10] is proposed to allow for reproducible extraction of an almost uniform key from a noisy non-uniform source.

Fuzzy Extractor. A fuzzy extractor consists of two algorithms $(\mathsf{Gen}, \mathsf{Rep})$. The generation algorithm Gen takes as input w (a reading of the source), and outputs a string R and a public helper string P. The reproduction algorithm Rep will reproduce R from w' with the help of P if the distance between w' and w is smaller enough. Note that the difference between w' and w is caused by errors and the

W. Susilo and G. Yang (Eds.): ACISP 2018, LNCS 10946, pp. 13–27, 2018.
https://doi.org/10.1007/978-3-319-93638-3_2

distance of w' and w evaluates the number of errors. Let n be the bit-length of w. We say that the fuzzy extractor supports linear fraction errors if it can correct up to $O(n)$ bits of errors. The security of fuzzy extractor guarantees that if w has enough min-entropy, then R is almost uniform or at least pseudorandom conditioned on P.

With a fuzzy extractor, it is convenient to implement key management for a cryptosystem. For example, a user can distill a uniform and accurately reproducible key R from his biometric data, via the generation algorithm of a fuzzy extractor, i.e., $(\mathsf{P}, \mathsf{R}) \leftarrow \mathsf{Gen}(\mathsf{w})$. Then he uses key R for cryptographic applications. When R is needed again, the user does another reading w' of his biometric data and reproduces R by the Rep algorithm with the help of P, i.e., $\mathsf{R} \leftarrow \mathsf{Rep}(\mathsf{P}, \mathsf{w}')$. During the application, the user never stores R. The public helper string P suffices for the reproduction of R.

Given a source W, multiple extractions of W by the generation algorithm result in multiple distilled key R_j and public helper strings P_j. When those keys R_j are employed in different cryptosystems, it is not desirable that the corruption of R_j endangers the usage of R_i. However, the distilled keys $\{\mathsf{R}_1, \ldots, \mathsf{R}_\rho\}$ are correlated via W. Information theoretically, given $\{(\mathsf{P}_j, \mathsf{R}_j)\}_{j \neq i}$, there might be no entropy left in R_i. Therefore most of the fuzzy extractors do not support multiple extractions of the same source [5–7,16]. This gives rise to another issue: how to support multiple extractions of the same source data? This issue is addressed by *reusable fuzzy extractor.*

Reusable Fuzzy Extractor. Reusable fuzzy extractor was first formalized by Boyen [7]. For multiple correlated samples $(\mathsf{w}, \mathsf{w}_1, \cdots, \mathsf{w}_\rho)$ of the same source, say biometric iris, applying the generation algorithm of reusable fuzzy extractor to $(\mathsf{w}, \mathsf{w}_1, \cdots, \mathsf{w}_\rho)$ respectively results in multiple pairs $(\mathsf{P}, \mathsf{R}), (\mathsf{P}_1, \mathsf{R}_1), \cdots, (\mathsf{P}_\rho, \mathsf{R}_\rho)$. The security of reusable fuzzy extractor asks for the (pseudo)randomness of R conditioned on $(\mathsf{P}, \mathsf{P}_1, \mathsf{R}_1, \cdots, \mathsf{P}_\rho, \mathsf{R}_\rho)$.

In [7], two constructions of reusable fuzzy extractor were presented. One achieves outsider security in the information theoretical setting, the other achieves insider security based on the random oracle model. Both constructions require that the difference $\delta_i = \mathsf{w}_i - \mathsf{w}$ is independent of w. Outsider security is weak in the sense that it only guarantees the randomness of R conditioned on the public helper string $(\mathsf{P}, \mathsf{P}_1, \cdots, \mathsf{P}_\rho)$.

Canetti et al. [8] constructed a reusable fuzzy extractor from a powerful tool "digital locker", and there is no assumption on how multiple readings are correlated. However, their construction can only tolerate sub-linear fraction of errors. Following the paradigm of constructing reusable fuzzy extractor from digital locker [8], Alamélou et.al. [2] built a reusable fuzzy extractor which can tolerate linear fraction of errors. However, "digital locker" is too powerful to find good instantiations. The available digital locker is either instantiated with a hash function modeled as a random oracle or based on a non-standard assumption.

As a promising post-quantum hard problem, the learning with errors (LWE) problem attracts lots of attentions from cryptographers. Great efforts have been and are devoted to the designs of a variety of cryptographic primitives from the

LWE assumption. The first fuzzy extractor from the LWE assumption is due to Fuller et al. [11]. Later, Apon et al. [3] extended the construction of fuzzy extractor to a reusable one. In their security model of reusable fuzzy extractor, the error δ_i can be adaptively manipulated by a probabilistic polynomial-time (PPT) adversary. As their construction uses the same error correction algorithm as [11], it can only tolerate logarithmic fraction of errors, i.e., for an input w of length n, it tolerates $O(\log n)$ errors. Another restriction of their construction is that components of $\mathsf{w} = (\mathsf{w}[1], \mathsf{w}[2], \ldots, \mathsf{w}[n]) \in \mathbb{Z}_q^n$ must be independently chosen according to some distribution χ, where χ is the error distribution in the LWE problem. It is hard to imagine that our biometric data follow discrete Gaussian distributions. Therefore this restriction is unreasonable.

Up to now, no construction is available for reusable fuzzy extractor, which is based on the LWE assumption and supports linear fraction of errors.

1.1 Our Contribution

In this work, we propose a simple and efficient construction of reusable fuzzy extractor based on the LWE assumption. Our security model is similar to [3], where the difference δ_i between the readings is adaptively chosen by a PPT adversary. Compared with the work of Apon et al. [3] which gave the only reusable fuzzy extractor based on the LWE assumption, our construction enjoys the following nice properties.

- Our construction is resilient to linear fraction of errors, whereas the fuzzy extractor in [3] can only tolerate logarithm fraction of errors.
- Our construction imposes no special structure requirement on the input w except that w should have enough entropy (as fuzzy extractors always required). Recall that for an input $\mathsf{w} \in \mathbb{Z}_q^n$, reusable fuzzy extractor by Apon et al. requires that each coordinate of w is chosen independently according to χ, which is the error distribution in the LWE problem.

We stress that our construction is the first reusable fuzzy extractor resilient to linear fraction of errors based on the LWE assumption. In Table 1, we compare our work with previous fuzzy extractor with reusability or from the LWE assumption.

Our Approach. Our construction makes use of a universal hash function and a secure sketch [9]. A secure sketch consists of a pair of algorithms $(\mathsf{SS.Gen}, \mathsf{SS.Rec})$ and works as follows. The generation algorithm $\mathsf{SS.Gen}$ on input w, outputs a sketch s; the recovery algorithm $\mathsf{SS.Rec}$, on input s, can recover w from w' if w' is close to w. The security of secure sketch guarantees that s does not leak too much information of w.

- To correct errors, we apply secure sketch to w to generate a sketch s.
- To distill a random string, we apply the universal hash function $\mathsf{H_i}$ to w.

Observe that if w has enough min-entropy, then by the security of the secure sketch and the leftover hash lemma, $\mathsf{H_i}(w)$ is statistically indistinguishable from

Table 1. Comparison with some known fuzzy extractor schemes. "Reusability?" asks whether the fuzzy extractor achieves reusability; "Standard Assumption?" asks whether the fuzzy extractor is based on standard assumptions. "Linear Fraction of Errors?" asks whether the scheme can correct linear fraction of errors. "–" represents the scheme is an information theoretical one.

FE Schemes	Reusabiliy?	Standard Assumption?	Linear Fraction of Errors?
FMR13 [11]	✗	✔ (LWE)	✗
DRS04 [10], Boy04 [7]	Weak	–	✔
CFPRS16 [8]	✔	✗	✗
Boy04 [7] ABCG16 [2]	✔	✗	✔
ACEK17 [3]	✔	✔ (LWE)	✗
Ours	✔	✔ (LWE)	✔

uniformly random. However, for multiples readings $(\mathsf{w}, \mathsf{w}_1, \cdots, \mathsf{w}_\rho)$ of the same source, if two reading are identical then the outputs of the hash function will be identical as well. Obviously, this approach is impossible to achieve reusability.

To solve this problem, we do not use the output of the universal hash function $\mathsf{H}_\mathsf{i}(\mathsf{w})$ as the final output of fuzzy extractor. Instead, we use $\mathsf{H}_\mathsf{i}(\mathsf{w})$ as the secret key of a symmetric LWE-based encryption scheme. Then the LWE-based scheme encrypts a randomly distributed string R which serves as the extracted key, and the ciphertext and sketch serve as the public helper string P. At the same time, we require that the universal hash function and secure sketch should be homomorphic. This helps our fuzzy extractor to achieve reusability.

2 Preliminaries

Let λ be the security parameter. Vectors are used in the column form. We use boldface letters to denote vectors or matrices. For a column vector $\mathbf{x}$, let $\mathbf{x}[i]$ denote the i-th element of $\mathbf{x}$. Let $\mathbf{I}_l$ denote the identity matrix of $l \times l$. For a real number x, let $\lfloor x \rceil$ denote the integer closest to x. By $[\rho]$, we denote set $\{1, 2 \cdots, \rho\}$."PPT" is short for probabilistic polynomial-time. For a distribution X, let $x \leftarrow X$ denote the process of sampling x according to X. For a set $\mathcal{X}$, $x \leftarrow_\$ \mathcal{X}$ denotes choosing x from $\mathcal{X}$ uniformly at random and $|\mathcal{X}|$ denotes the cardinality of the set. We use game-based security proof. Let the notation $\mathsf{G} \Rightarrow 1$ denote the event that game G returns 1, and notion $x \overset{\mathsf{G}}{=} y$ denote that x equals y or is computed as y in game G.

2.1 Metric Spaces

A metric space is a set $\mathcal{M}$ with a distance function $\mathsf{dis}\colon \mathcal{M} \times \mathcal{M} \mapsto \mathbb{Z}^+ \cup \{0\}$. In this paper, we consider $\mathcal{M} = \mathcal{F}^n$ for some alphabet $\mathcal{F}$ equipped with the Hamming distance. For any two elements $\mathsf{w}, \mathsf{w}' \in \mathcal{M}$, the Hamming distance $\mathsf{dis}(\mathsf{w}, \mathsf{w}')$ is the number of coordinates in which they differ.

2.2 Min-Entropy and Statistical Distance

Definition 1 (Average Min-Entropy). *For two random variables X and Y, the* average min-entropy *of X given Y is defined by*

$$\widetilde{H}_\infty(X \mid Y) := -\log\left[\mathbb{E}_{y\leftarrow Y}(\max_x \Pr[X = x \mid Y = y])\right].$$

Definition 2 (Statistical Distance). *For two random variables X and Y over a set $\mathcal{M}$, the* statistical distance *of X and Y is given by* $\mathbf{SD}(X,Y) := \frac{1}{2}\sum_{\mathsf{w}\in\mathcal{M}} |\Pr[X = \mathsf{w}] - \Pr[Y = \mathsf{w}]|$. *If $\mathbf{SD}(X,Y) \leq \varepsilon$, X and Y are called* ε-statistically indistinguishable*, denoted by $X \overset{\varepsilon}{\approx} Y$.*

2.3 Universal Hashing

Definition 3 (Universal Hash Functions[9]). *A family of hash functions $\mathcal{H} = \{\mathsf{H}_\mathsf{i} : \mathcal{X} \to \mathcal{Y} \mid \mathsf{i} \in \mathcal{I}\}$ is universal, if for all $x \neq x' \in \mathcal{X}$, it holds that $\Pr_{\mathsf{i}\leftarrow\!\$\,\mathcal{I}}[\mathsf{H}_\mathsf{i}(x) = \mathsf{H}_\mathsf{i}(x')] \leq \frac{1}{|\mathcal{Y}|}$.*

Concrete Construction of Universal Hash Functions. Let q be a prime. For $\mathbf{w} \in \mathbb{Z}_q^{l'}, \mathbf{A} \in \mathbb{Z}_q^{nl\times l'}$, define

$$\mathsf{H}_\mathbf{A}(\mathbf{w}) := \mathbf{A}\mathbf{w}, \tag{1}$$

then $\mathcal{H} = \{\mathsf{H}_\mathbf{A} : \mathbb{Z}_q^{l'} \to \mathbb{Z}_q^{nl} \mid \mathbf{A} \in \mathbb{Z}_q^{nl\times l'}\}$ is a family of universal hash functions.

Note that the above hash function is homomorphic in the sense that

$$\mathsf{H}_\mathbf{A}(\mathbf{w}+\mathbf{w}') = \mathbf{A}(\mathbf{w}+\mathbf{w}') = \mathbf{A}\mathbf{w} + \mathbf{A}\mathbf{w}' = \mathsf{H}_\mathbf{A}(\mathbf{w}) + \mathsf{H}_\mathbf{A}(\mathbf{w}'). \tag{2}$$

One can easily interpret a vector in $\mathbb{Z}_q^{nl}$ as a matrix in $\mathbb{Z}_q^{n\times l}$. Thus we get a family of homomorphic universal hash functions $\mathcal{H} = \{\mathsf{H}_\mathbf{A} : \mathbb{Z}_q^{l'} \to \mathbb{Z}_q^{n\times l} \mid \mathbf{A} \in \mathbb{Z}_q^{nl\times l'}\}$.

Remark 1. The reason why we interpret a vector in $\mathbb{Z}_q^{nl}$ as a matrix in $\mathbb{Z}_q^{n\times l}$ is for the convenience of the later construction of reusable fuzzy extractor in Sect. 3.

Lemma 1 (Generalized Leftover Hash Lemma [9,15]). *If $\mathcal{H} = \{\mathsf{H}_\mathsf{i} : \mathbb{Z}_q^{l'} \to \mathbb{Z}_q^{n\times l}, \mathsf{i} \in \mathcal{I}\}$ is a family of universal hash functions, then for any random variable W taking values in $\mathbb{Z}_q^{l'}$ and any random variable Y,*

$$\mathbf{SD}\Big((\mathsf{H}_I(W), I, Y), (U, I, Y)\Big) \leq \frac{1}{2}\sqrt{2^{-\widetilde{H}_\infty(W|Y)}q^{nl}},$$

where I and U are uniformly distributed over $\mathcal{I}$ and $\mathbb{Z}_q^{n\times l}$, respectively.

2.4 Secure Sketch

Definition 4 (Secure Sketch [9]**).** *An* $(\mathcal{M}, \mathfrak{m}, \hat{\mathfrak{m}}, t)$-secure sketch (SS) $\mathsf{SS} = (\mathsf{SS.Gen}, \mathsf{SS.Rec})$ *for metric space* $\mathcal{M}$ *with distance function* dis*, consists of a pair of PPT algorithms and satisfies correctness and security.*

- $\mathsf{SS.Gen}$ *on input* $\mathsf{w} \in \mathcal{M}$*, outputs a sketch* s*.*
- $\mathsf{SS.Rec}$ *takes as input a sketch* s *and* $\mathsf{w}' \in \mathcal{M}$*, and outputs* $\widetilde{\mathsf{w}}$*.*

Correctness. *For any* $\mathsf{w} \in \mathcal{M}$*, any* $s \leftarrow \mathsf{SS.Gen}(\mathsf{w})$*, if* $\mathsf{dis}(\mathsf{w}, \mathsf{w}') \leq t$*, then* $\mathsf{SS.Rec}(s, \mathsf{w}') = \mathsf{w}$*.*

Security. *For any random variable* W *over* $\mathcal{M}$ *with min-entropy* $\mathfrak{m}$*, we have* $\widetilde{H}_\infty(W \mid \mathsf{SS.Gen}(W)) \geq \hat{\mathfrak{m}}$*.*

A secure sketch is homomorphic if $\mathsf{SS.Gen}(\mathsf{w} + \mathsf{w}') = \mathsf{SS.Gen}(\mathsf{w}) + \mathsf{SS.Gen}(\mathsf{w}')$.

An efficient $[n, k, 2t+1]_{\mathbb{F}}$-linear error correcting code $\mathcal{E}$ over $\mathbb{F}^n$ is a subspace of $\mathbb{F}^n$ and $\mathcal{E} = \{\mathsf{w} \in \mathbb{F}^n | \mathbf{H}\mathsf{w} = 0\}$, where matrix $\mathbf{H}$ is the $(n-k) \times n$ parity-check matrix of $\mathcal{E}$. For $\mathsf{w} \in \mathbb{F}^n$, define syndrome $\mathsf{syn}(\mathsf{w}) = \mathbf{H}\mathsf{w}$. For any $\mathbf{c} \in \mathcal{E}$, $\mathsf{syn}(\mathbf{c}+\mathbf{e}) = \mathsf{syn}(\mathbf{c}) + \mathsf{syn}(\mathbf{e}) = \mathsf{syn}(\mathbf{e})$. The syndrome captures all the information necessary for decoding.

As suggested in [9], based on an $[n, k, 2t+1]_{\mathbb{F}}$-linear error correcting code, a syndrome-based secure sketch can be constructed as follows.

Syndrome-Based Construction of Secure Sketch. [9] Define

$$\mathsf{SS.Gen}(\mathsf{w}) := \mathsf{syn}(\mathsf{w}) = \mathbf{H}\mathsf{w} = \mathsf{s}, \quad \mathsf{SS.Rec}(\mathsf{s}, \mathsf{w}') := \mathsf{w}' - \mathbf{e}, \tag{3}$$

where $\mathbf{e}$ is the unique vector of Hamming weight less than t such that $\mathsf{syn}(\mathbf{e}) = \mathsf{syn}(\mathsf{w}') - \mathsf{s}$.

Lemma 2. [9] *Given an* $[n, k, 2t+1]_{\mathbb{F}}$ *error-correcting code, one can construct an* $(\mathbb{F}^n, \mathfrak{m}, \mathfrak{m} - (n-k)|\mathbb{F}|, t)$ *secure sketch, which is efficient if encoding and decoding are efficient.*

Since there exist efficient $[n, k, 2t+1]_{\mathbb{F}}$-linear error correcting codes such that $t = O(n)$, the syndrome-based Secure Sketch can correct up to linear fraction of errors. Meanwhile, the fact that $\mathsf{SS.Gen}(\mathsf{w} + \mathsf{w}') := \mathsf{syn}(\mathsf{w} + \mathsf{w}') = \mathbf{H}(\mathsf{w} + \mathsf{w}') = \mathbf{H}\mathsf{w} + \mathbf{H}\mathsf{w}'$ suggests that the syndrome-based Secure Sketch is also homomorphic.

2.5 Learning with Error (LWE) Problem

The learning with errors (LWE) problem was introduced by Regev [13,14].

Definition 5 (Learning with errors (LWE**) problem).** *Let integers* $n = n(\lambda)$*,* $m = m(\lambda)$ *and* $q = q(\lambda) \geq 2$*. Let* $\chi(\lambda)$ *be a distribution over* $\mathbb{Z}_q$*. The decisional* $\mathsf{LWE}_{n,m,q,\chi}$ *problem is to distinguish* $(\mathbf{A}, \mathbf{A}\mathbf{s} + \mathbf{e})$ *from* $(\mathbf{A}, \mathbf{u})$*, where* $\mathbf{A} \leftarrow_{\$} \mathbb{Z}_q^{m \times n}$*,* $\mathbf{s} \leftarrow_{\$} \mathbb{Z}_q^n$*,* $\mathbf{e} \leftarrow \chi^m$ *and* $\mathbf{u} \leftarrow_{\$} \mathbb{Z}_q^m$*.*

The decisional $\mathsf{LWE}_{n,m,q,\chi}$ *problem is* ϵ*-hard if for any PPT adversary* $\mathcal{A}$*, its advantage* $\mathsf{Adv}^{n,m,q,\chi}_{\mathsf{LWE},\mathcal{A}}(\lambda)$ *is upper bounded by* ϵ*, i.e.,*

$$\mathsf{Adv}^{n,m,q,\chi}_{\mathsf{LWE},\mathcal{A}}(\lambda) := |\Pr[\mathcal{A}^{\mathcal{O}_{\mathsf{LWE}}(\mathbf{s})} = 1] - \Pr[\mathcal{A}^{\mathcal{O}_U} = 1]| \leq \epsilon.$$

Here the oracle $\mathcal{O}_{\mathsf{LWE}}$ *returns* $(\mathbf{A}, \mathbf{As} + \mathbf{e})$ *where* $\mathbf{A} \leftarrow_\$ \mathbb{Z}_q^{m \times n}$*,* $\mathbf{s} \leftarrow_\$ \mathbb{Z}_q^n$*,* $\mathbf{e} \leftarrow \chi^m$ *and the oracle* $\mathcal{O}_U$ *returns* $(\mathbf{A}, \mathbf{u})$ *where* $\mathbf{A} \leftarrow_\$ \mathbb{Z}_q^{m \times n}$ *and* $\mathbf{u} \leftarrow_\$ \mathbb{Z}_q^m$*, and* $\mathcal{A}$ *is limited to make at most one call to the oracle. The decisional* $\mathsf{LWE}_{n,m,q,\chi}$ *problem is hard if for any PPT adversary* $\mathcal{A}$*, its advantage* $\mathsf{Adv}^{n,m,q,\chi}_{\mathsf{LWE},\mathcal{A}}(\lambda)$ *is negligible.*

The decisional $\mathsf{LWE}_{n,m,l,q,\chi}$ problem is to distinguish $(\mathbf{A}, \mathbf{AS} + \mathbf{E})$ from $(\mathbf{A}, \mathbf{U})$, where $\mathbf{A} \leftarrow_\$ \mathbb{Z}_q^{m \times n}$, $\mathbf{S} \leftarrow_\$ \mathbb{Z}_q^{n \times l}$, $\mathbf{E} \leftarrow \chi^{m \times l}$ and $\mathbf{U} \leftarrow_\$ \mathbb{Z}_q^{m \times l}$. By a simple hybrid argument, one can show that the decisional $\mathsf{LWE}_{n,m,l,q,\chi}$ problem is hard if the decisional $\mathsf{LWE}_{n,m,q,\chi}$ problem is hard.

Lemma 3. [12] *If the decisional* $\mathsf{LWE}_{n,m,q,\chi}$ *problem is* ϵ*-hard, then the decisional* $\mathsf{LWE}_{n,m,l,q,\chi}$ *problem is* $\epsilon \cdot l$*-hard. More precisely,*

$$\mathsf{Adv}^{n,m,l,q,\chi}_{\mathsf{LWE},\mathcal{A}}(\lambda) := |\Pr[\mathcal{A}^{\mathcal{O}_{\mathsf{LWE}}(\mathbf{S})} = 1] - \Pr[\mathcal{A}^{\mathcal{O}_U} = 1]| \leq \epsilon \cdot l.$$

Here the oracle $\mathcal{O}_{\mathsf{LWE}}$ *returns* $(\mathbf{A}, \mathbf{AS} + \mathbf{E})$ *where* $\mathbf{A} \leftarrow_\$ \mathbb{Z}_q^{m \times n}$*,* $\mathbf{S} \leftarrow_\$ \mathbb{Z}_q^{n \times l}$*,* $\mathbf{E} \leftarrow \chi^{m \times l}$ *and the oracle* $\mathcal{O}_U$ *returns* $(\mathbf{A}, \mathbf{U})$ *where* $\mathbf{A} \leftarrow_\$ \mathbb{Z}_q^{m \times n}$ *and* $\mathbf{U} \leftarrow_\$ \mathbb{Z}_q^{m \times l}$*, and* $\mathcal{A}$ *is limited to make at most one call to the oracle.*

If $m = \rho m'$ with $m, m', \rho \in \mathbb{Z}^+$, the above lemma has an equivalent form.

Lemma 4. [12] *Let* $m = \rho m'$ *with* $m, m', \rho \in \mathbb{Z}^+$*. If the decisional* $\mathsf{LWE}_{n,m,q,\chi}$ *problem is* ε*-hard, then the decisional* $\mathsf{LWE}_{n,m,l,q,\chi}$ *problem is* $\epsilon \cdot l$*-hard. More precisely,*

$$\mathsf{Adv}^{n,m,l,q,\chi}_{\mathsf{LWE},\mathcal{A}}(\lambda) := |\Pr[\mathcal{A}^{\mathcal{O}_{\mathsf{LWE}}(\mathbf{S})} = 1] - \Pr[\mathcal{A}^{\mathcal{O}_U} = 1]| \leq \epsilon \cdot l.$$

Here the oracle $\mathcal{O}_{\mathsf{LWE}}$ *returns* $(\mathbf{A}, \mathbf{AS} + \mathbf{E})$ *where* $\mathbf{A} \leftarrow_\$ \mathbb{Z}_q^{m' \times n}$*,* $\mathbf{S} \leftarrow_\$ \mathbb{Z}_q^{n \times l}$*,* $\mathbf{E} \leftarrow \chi^{m' \times l}$ *and the oracle* $\mathcal{O}_U$ *returns* $(\mathbf{A}, \mathbf{U})$ *where* $\mathbf{A} \leftarrow_\$ \mathbb{Z}_q^{m' \times n}$ *and* $\mathbf{U} \leftarrow_\$ \mathbb{Z}_q^{m' \times l}$*, and* $\mathcal{A}$ *is limited to make at most* ρ *calls to the oracle.*

Consider a real parameter $\alpha = \alpha(n) \in (0, 1)$ and a prime q. Denote by $\mathbb{T} = \mathbb{R}/\mathbb{Z}$, i.e., the group of reals [0, 1) with modulo 1 addition. Define Ψ_α to be the distribution on $\mathbb{T}$ of a normal variable with mean 0 and standard deviation $\alpha/\sqrt{2\pi}$ reduced modulo 1. We denote by $\bar{\Psi}_\alpha$ the discrete distribution over $\mathbb{Z}_q$ of the random variable $\lfloor qX \rceil \mod q$ where the random variable X has distribution Ψ_α.

Lemma 5. [13] *If there exists an efficient, possibly quantum, algorithm for the decisional* $\mathsf{LWE}_{n,m,q,\bar{\Psi}_\alpha}$ *problem for* $q > 2\sqrt{n/\alpha}$*, then there exists an efficient quantum algorithm for approximating the SIVP and GapSVP problems, to within* $O((n/\alpha) \cdot \log^c n)$ *factors in the* l_2 *norm, in the worst case.*

Lemma 6. [1] *Let* $\mathbf{x}$ *be some vector in* $\{0, 1\}^m$ *and let* $\mathbf{e} \leftarrow \bar{\Psi}_\alpha^m$*. Then the quantity* $|\mathbf{x}^\top \mathbf{e}|$ *treated as an integer in* $[0, q-1]$ *satisfies*

$$|\mathbf{x}^\top \mathbf{e}| \leq \sqrt{m} q \alpha \omega(\sqrt{\log m}) + m/2$$

with all but negligible probability in m*.*

3 Reusable Fuzzy Extractor

Definition 6 (Reusable Fuzzy Extractor). *An* $(\mathcal{M}, \mathfrak{m}, \mathcal{R}, t, \varepsilon, \rho)$-resuable fuzzy extractor (rFE) *for metric space* $\mathcal{M}$ *consists of three PPT algorithms* $(\mathsf{Init}, \mathsf{Gen}, \mathsf{Rep})$*,*

- $\mathsf{Init}(1^\lambda)$*: the initialization algorithm takes as input the security parameters and outputs the public parameters* pp*.*
- $\mathsf{Gen}(\mathsf{pp}, \mathsf{w})$*: the generation algorithm takes as input the public parameters* pp *and* $\mathsf{w} \in \mathcal{M}$*. It outputs a public helper string* P *and an extracted string* $\mathsf{R} \in \mathcal{R}$*.*
- $\mathsf{Rep}(\mathsf{pp}, \mathsf{P}, \mathsf{w}')$*: the reproduction algorithm takes as input the public parameters* pp*, public helper string* P *and* $\mathsf{w}' \in \mathcal{M}$*, and outputs an extracted string* R *or* $\perp$*.*

It satisfies the following properties.

Correctness. *For all* $\mathsf{w}, \mathsf{w}' \in \mathcal{M}$ *with* $\mathsf{dis}(\mathsf{w}, \mathsf{w}') \leq t$*, for all* $\mathsf{pp} \leftarrow \mathsf{Init}(1^\lambda)$*,* $(\mathsf{P}, \mathsf{R}) \leftarrow \mathsf{Gen}(\mathsf{pp}, \mathsf{w})$ *and* $\widetilde{\mathsf{R}} \leftarrow \mathsf{Rep}(\mathsf{pp}, \mathsf{P}, \mathsf{w}')$*, it holds that* $\widetilde{\mathsf{R}} = \mathsf{R}$ *with overwhelming probability.*

Reusability. *For any distribution* W *over metric space* $\mathcal{M}$ *with* $H_\infty(W) \geq \mathfrak{m}$*, any PPT adversary* $\mathcal{A}$*, its advantage defined below satisfies*

$$\mathsf{Adv}^{\mathsf{reu}}_{\mathsf{rFE},\mathcal{A}}(1^\lambda) := |\Pr[\mathsf{Exp}^{\mathsf{reu}}_{\mathsf{rFE},\mathcal{A}}(1) \Rightarrow 1] - \Pr[\mathsf{Exp}^{\mathsf{reu}}_{\mathsf{rFE},\mathcal{A}}(0) \Rightarrow 1]| \leq \varepsilon,$$

where $\mathsf{Exp}^{\mathsf{reu}}_{\mathsf{rFE},\mathcal{A}}(\beta)$*,* $\beta \in \{0,1\}$*, describes the reusability experiment played between a challenger* $\mathcal{C}$ *and an adversary* $\mathcal{A}$*.*

$\underline{\mathsf{Exp}^{\mathsf{reu}}_{\mathsf{rFE},\mathcal{A}}(\beta):}$ // $\beta \in \{0,1\}$

1. *Challenger* $\mathcal{C}$ *invokes* $\mathsf{pp} \leftarrow \mathsf{Init}(1^\lambda)$ *and returns* pp *to* $\mathcal{A}$*.*
2. *Challenger* $\mathcal{C}$ *samples* $\mathsf{w} \leftarrow W$ *and invokes* $(\mathsf{P}, \mathsf{R}) \leftarrow \mathsf{Gen}(\mathsf{pp}, w)$*. If* $\beta = 1$*,* $\mathcal{C}$ *returns* (P, R) *to* $\mathcal{A}$*; if* $\beta = 0$*, it chooses* $U \leftarrow_\$ \mathcal{R}$ *and returns* (P, U) *to* $\mathcal{A}$*.*
3. $\mathcal{A}$ *may adaptively make at most* ρ *queries of the following form:*
   - $\mathcal{A}$ *submits a shift* $\delta_i \in \mathcal{M}$ *to* $\mathcal{C}$*.*
   - $\mathcal{C}$ *invokes* $(\mathsf{P}_i, \mathsf{R}_i) \leftarrow \mathsf{Gen}(\mathsf{pp}, \mathsf{w} + \delta_i)$*, and returns* $(\mathsf{P}_i, \mathsf{R}_i)$ *to* $\mathcal{A}$*.*
4. *As long as* $\mathcal{A}$ *outputs a guessing bit* β'*, the experiment outputs* β'*.*

3.1 Construction of Reusable Fuzzy Extractor from LWE

Our construction of reusable fuzzy extractor $\mathsf{rFE} = (\mathsf{Init}, \mathsf{Gen}, \mathsf{Rep})$ is shown in Fig. 1, which uses the following building blocks.

- A homomorphic $(\mathbb{Z}_q^{l'}, \mathfrak{m}, \hat{\mathfrak{m}}, t)$-secure sketch $\mathsf{SS} = (\mathsf{SS.Gen}, \mathsf{SS.Rec})$.
- A family of universal hash functions $\mathcal{H} = \{\mathsf{H_i} : \mathbb{Z}_q^{l'} \rightarrow \mathbb{Z}_q^{n \times l}, \mathsf{i} \in \mathcal{I}\}$ with homomorphic property as defined by (2).

```
$\mathsf{pp} \leftarrow \mathsf{Init}(1^\lambda)$:
  $\mathsf{H_i} \leftarrow_\$ \mathcal{H}$.
  $\mathsf{pp} := \mathsf{H_i}$.
  Return $\mathsf{pp}$.
```

```
$(\mathsf{P}, \mathsf{R}) \leftarrow \mathsf{Gen}(\mathsf{pp}, \mathsf{w})$:    // $\mathsf{w} \in \mathbb{Z}_q^{l'}$
  $s \leftarrow \mathsf{SS.Gen}(\mathsf{w})$.
  $\mathbf{S} := \mathsf{H_i}(\mathsf{w}) \in \mathbb{Z}_q^{n \times l}$.
  $\mathbf{A} \leftarrow_\$ \mathbb{Z}_q^{m \times n}$.
  $\mathbf{E} \leftarrow \chi^{m \times l}$.
  $\mathbf{B} := (\mathbf{A}, \mathbf{A} \cdot \mathbf{S} + \mathbf{E}) \in \mathbb{Z}_q^{m \times (n+l)}$
  $\mathbf{x} \leftarrow_\$ \{0,1\}^m$.
  $\mathbf{m} \leftarrow_\$ \{0,1\}^l$.
  $\mathbf{c}^\top = \mathbf{x}^\top \mathbf{B} + (\mathbf{0}^\top, \mathbf{m}^\top \cdot \lfloor \frac{q}{2} \rceil)$.
  $\mathsf{P} := (s, \mathbf{c})$, $\mathsf{R} := \mathbf{m}$.
  Return $(\mathsf{P}, \mathsf{R})$.
```

```
$\mathsf{R} \leftarrow \mathsf{Rep}(\mathsf{pp}, \mathsf{P}, \mathsf{w}')$:
  Parse $\mathsf{P} = (s, \mathbf{c})$.
  $\widetilde{\mathsf{w}} \leftarrow \mathsf{SS.Rec}(s, \mathsf{w}')$.
  $\mathbf{S} := \mathsf{H_i}(\widetilde{\mathsf{w}}) \in \mathbb{Z}_q^{n \times l}$.
  $\mathbf{d} = \mathbf{c}^\top \cdot \begin{pmatrix} -\mathbf{S} \\ \mathbf{I}_l \end{pmatrix} \in \mathbb{Z}_q^l$.
  For $i = 1$ to $l$
    $\mathbf{m}[i] = \begin{cases} 1 & \text{if } \mathbf{d}[i] \in [\frac{1}{4}q, \frac{3}{4}q] \\ 0 & \text{else} \end{cases}$
  $\mathsf{R} := \mathbf{m}$.
  Return $\mathsf{R}$.
```

Fig. 1. Construction of rFE from LWE.

Remark 2. The content in the dashed frame is an LWE-based symmetric encryption scheme which is adapted from [12], the secret key is $\mathbf{S}$ and the message is $\mathbf{m}$.

Theorem 1. *If* SS *is a homomorphic* $(\mathbb{Z}_q^{l'}, \mathfrak{m}, \hat{\mathfrak{m}}, t)$*-secure sketch,* $\mathcal{H}$ *is a universal family of hash functions* $\mathcal{H} = \{\mathsf{H_i} \colon \mathbb{Z}_q^{l'} \to \mathbb{Z}_q^{n \times l}, \mathsf{i} \in \mathcal{I}\}$ *with homomorphic property as defined by (2), it satisfies* $\hat{\mathfrak{m}} - nl \log q \geq \omega(\log \lambda)$*, and the* $\mathsf{LWE}_{n,(\rho+1)m,l,q,\chi}$ *problem is* ϵ*-hard, where* χ *is the discrete Gaussian distribution* Ψ_α*,* $q \geq 4m$*,* $\alpha \leq 1/(8 \cdot \sqrt{m} \cdot g(n))$ *for any* $g(n) = \omega(\sqrt{\log n})$ *and* $m \geq (n+l)\log q + \omega(\log \lambda)$*, then* rFE *in Fig. 1 is an* $(\mathbb{Z}_p^{n \times l'}, \mathfrak{m}, \{0,1\}^l, t, \varepsilon, \rho)$*-reusable fuzzy extractor, where* $\varepsilon \leq 2^{-\omega(\log \lambda)} + 2\epsilon$*.*

Proof. Let us analyze the correctness first. If $\mathsf{dis}(\mathsf{w}, \mathsf{w}') \leq t$, then by the correctness of SS, we have $\mathsf{w} = \widetilde{\mathsf{w}}$, where $\widetilde{\mathsf{w}} \leftarrow \mathsf{SS.Rec}(s, \mathsf{w}')$ and $s = \mathsf{SS.Gen}(\mathsf{w})$. As a consequence, $\mathbf{S}$ can be correctly recovered. Next, we have

$$\begin{aligned} \mathbf{d} = \mathbf{c}^\top \cdot \begin{pmatrix} -\mathbf{S} \\ \mathbf{I}_l \end{pmatrix} &= \left(\mathbf{x}^\top \mathbf{B} + (\mathbf{0}^\top, \mathbf{m}^\top \cdot \left\lfloor \frac{q}{2} \right\rceil)\right) \cdot \begin{pmatrix} -\mathbf{S} \\ \mathbf{I}_l \end{pmatrix} \\ &= \left(\mathbf{x}^\top (\mathbf{A}, \mathbf{A} \cdot \mathbf{S} + \mathbf{E}) + (\mathbf{0}^\top, \mathbf{m}^\top \cdot \left\lfloor \frac{q}{2} \right\rceil)\right) \cdot \begin{pmatrix} -\mathbf{S} \\ \mathbf{I}_l \end{pmatrix} \\ &= \mathbf{x}^\top \mathbf{E} + \mathbf{m}^\top \cdot \left\lfloor \frac{q}{2} \right\rceil . \end{aligned}$$

Denote $\mathbf{E} = (\mathbf{e}_1, \cdots, \mathbf{e}_l)$, where $\mathbf{e}_i \leftarrow \chi^m$. Since $q \geq 4m$, $\alpha \leq 1/(8 \cdot \sqrt{m} \cdot g(n))$ for any $g(n) = \omega(\sqrt{\log n})$ and $\chi = \Psi_\alpha$, by Lemma 6, we have $|\mathbf{x}^\top \mathbf{e}_i| \leq q/4$ with overwhelming probability. Consequently, $\mathbf{m}$ can be correctly reproduced with overwhelming probability. The correctness of rFE follows.

Now we show its reusability by defining a sequence of games, and proving the adjacent games indistinguishable. The differences between adjacent games will be highlighted by underline.

Game G_0 : It is the game $\mathsf{Exp}^{\mathsf{reu}}_{\mathsf{rFE},\mathcal{A}}(1)$. More precisely,

1. Challenger $\mathcal{C}$ samples $\mathsf{H_i} \leftarrow_{\$} \mathcal{H}$, sets $\mathsf{pp} := \mathsf{H_i}$, and returns pp to $\mathcal{A}$.
2. Challenger $\mathcal{C}$ samples $\mathsf{w} \leftarrow W$, invokes $s \leftarrow \mathsf{SS.Gen}(\mathsf{w})$, $\mathbf{S} := \mathsf{H_i}(\mathsf{w})$, samples $\mathbf{A} \leftarrow_{\$} \mathbb{Z}_q^{m\times n}$, $\mathbf{E} \leftarrow \chi^{m\times l}$, sets $\mathbf{B} := (\mathbf{A}, \mathbf{A}\cdot\mathbf{S}+\mathbf{E})$, samples $\mathbf{x} \leftarrow_{\$} \{0,1\}^m$, $\mathbf{m} \leftarrow_{\$} \{0,1\}^l$, sets $\mathbf{c}^\top := \mathbf{x}^\top\mathbf{B} + (\mathbf{0}^\top, \mathbf{m}^\top \cdot \lfloor\frac{q}{2}\rceil)$, $\mathsf{P} := (s, \mathbf{c})$ and $\mathsf{R} := \mathbf{m}$. Finally, it returns (P, R) to $\mathcal{A}$.
3. Upon receiving a shift $\delta_i \in \mathcal{M}$ from $\mathcal{A}$, challenger $\mathcal{C}$ invokes $s_i \leftarrow \mathsf{SS.Gen}(\mathsf{w}+\delta_i)$, $\mathbf{S}_i := \mathsf{H_i}(\mathsf{w}+\delta_i)$, samples $\mathbf{A}_i \leftarrow_{\$} \mathbb{Z}_q^{m\times n}$, $\mathbf{E}_i \leftarrow \chi^{m\times l}$, sets $\mathbf{B}_i := (\mathbf{A}_i, \mathbf{A}_i\cdot\mathbf{S}_i+\mathbf{E}_i)$, samples $\mathbf{x}_i \leftarrow_{\$} \{0,1\}^m$, $\mathbf{m}_i \leftarrow_{\$} \{0,1\}^l$, sets $\mathbf{c}_i^\top := \mathbf{x}_i^\top\mathbf{B}_i + (\mathbf{0}^\top, \mathbf{m}_i^\top \cdot \lfloor\frac{q}{2}\rceil)$, $\mathsf{P}_i := (s_i, \mathbf{c}_i)$ and $\mathsf{R}_i := \mathbf{m}_i$. Finally, it returns $(\mathsf{P}_i, \mathsf{R}_i)$ to $\mathcal{A}$.
4. As long as $\mathcal{A}$ outputs a guessing bit β', the experiment outputs β'.

Clearly, we have

$$\Pr[\mathsf{G}_0 \Rightarrow 1] = \Pr[\mathsf{Exp}^{\mathsf{reu}}_{\mathsf{rFE},\mathcal{A}}(1) \Rightarrow 1]. \tag{4}$$

$\underline{\text{Game}\ \mathsf{G}_1 :}$ It is the same as G_0, except that $s_i \leftarrow \mathsf{SS.Gen}(\mathsf{w}+\delta_i)$ now is changed to $s_i = s + \mathsf{SS.Gen}(\delta_i)$ and $\mathbf{S}_i = \mathsf{H_i}(\mathsf{w}+\delta_i)$ now is changed to $\mathbf{S}_i = \mathbf{S} + \mathsf{H_i}(\delta_i)$ in step 3. More precisely,

3. Upon receiving a shift $\delta_i \in \mathcal{M}$ from $\mathcal{A}$, challenger $\mathcal{C}$ computes $\underline{s_i = s + \mathsf{SS.Gen}(\delta_i), \mathbf{S}_i := \mathbf{S} + \mathsf{H_i}(\delta_i)}$, samples $\mathbf{A}_i \leftarrow_{\$} \mathbb{Z}_q^{m\times n}$, $\mathbf{E}_i \leftarrow \chi^{m\times l}$, sets $\mathbf{B}_i := (\mathbf{A}_i, \mathbf{A}_i\cdot\mathbf{S}_i+\mathbf{E}_i)$, samples $\mathbf{x}_i \leftarrow_{\$} \{0,1\}^m$, $\mathbf{m}_i \leftarrow_{\$} \{0,1\}^l$, sets $\mathbf{c}_i^\top := \mathbf{x}_i^\top\mathbf{B}_i + (\mathbf{0}^\top, \mathbf{m}_i^\top \cdot \lfloor\frac{q}{2}\rceil)$, $\mathsf{P}_i := (s_i, \mathbf{c}_i)$ and $\mathsf{R}_i := \mathbf{m}_i$. Finally, it returns $(\mathsf{P}_i, \mathsf{R}_i)$ to $\mathcal{A}$.

Lemma 7. $\Pr[\mathsf{G}_0 \Rightarrow 1] = \Pr[\mathsf{G}_1 \Rightarrow 1]$.

Proof. By the homomorphic property of SS, we have

$$s_i \overset{\mathsf{G}_0}{=} \mathsf{SS.Gen}(\mathsf{w}+\delta_i) = \mathsf{SS.Gen}(\mathsf{w}) + \mathsf{SS.Gen}(\delta_i) = s + \mathsf{SS.Gen}(\delta_i) \overset{\mathsf{G}_1}{=} s_i.$$

By the homomorphic property of $\mathsf{H_i}$, we have

$$\mathbf{S}_i \overset{\mathsf{G}_0}{=} \mathsf{H_i}(\mathsf{w}+\delta_i) = \mathsf{H_i}(\mathsf{w}) + \mathsf{H_i}(\delta_i) = \mathbf{S} + \mathsf{H_i}(\delta_i) \overset{\mathsf{G}_1}{=} \mathbf{S}_i.$$

As a result, the changes from G_0 to G_1 are just conceptual, thus

$$\Pr[\mathsf{G}_0 \Rightarrow 1] = \Pr[\mathsf{G}_1 \Rightarrow 1].$$

□

$\underline{\text{Game}\ \mathsf{G}_2 :}$ It is the same as G_1, except that in G_2, $\mathbf{S}$ is uniformly chosen from $\mathbb{Z}_q^{n\times l}$ instead of $\mathbf{S} = \mathsf{H_i}(\mathsf{w})$ in step 2. More precisely,

2. Challenger $\mathcal{C}$ samples $\mathsf{w} \leftarrow W$, invokes $s \leftarrow \mathsf{SS.Gen}(\mathsf{w})$, $\underline{\mathbf{S} \leftarrow_{\$} \mathbb{Z}_q^{n\times l}}$, samples $\mathbf{A} \leftarrow_{\$} \mathbb{Z}_q^{m\times n}$, $\mathbf{E} \leftarrow \chi^{m\times l}$, sets $\mathbf{B} := (\mathbf{A}, \mathbf{A}\cdot\mathbf{S}+\mathbf{E})$, samples $\mathbf{x} \leftarrow_{\$} \{0,1\}^m$, $\mathbf{m} \leftarrow_{\$} \{0,1\}^l$, sets $\mathbf{c}^\top := \mathbf{x}^\top\mathbf{B} + (\mathbf{0}^\top, \mathbf{m}^\top \cdot \lfloor\frac{q}{2}\rceil)$, $\mathsf{P} := (s, \mathbf{c})$ and $\mathsf{R} := \mathbf{m}$. Finally, it returns (P, R) to $\mathcal{A}$.

Lemma 8.

$$|\Pr[\mathsf{G}_1 \Rightarrow 1] - \Pr[\mathsf{G}_2 \Rightarrow 1]| \leq 2^{-\omega(\log\lambda)}.$$

Proof. We consider the information about the source w that is used in G_1.

- In step 1, challenger $\mathcal{C}$ does not need w.
- In step 2, challenger $\mathcal{C}$ uses w to generate the sketch s and extract $\mathbf{S}$, where $s \leftarrow \mathsf{SS.Gen}(\mathsf{w})$, $\mathbf{S} = \mathsf{H_i}(\mathsf{w})$.
- In step 3, upon receiving a shift δ_i from $\mathcal{A}$, challenger $\mathcal{C}$ computes $s_i = s + \mathsf{SS.Gen}(\delta_i)$, $\mathbf{S}_i = \mathbf{S} + \mathsf{H_i}(\delta_i)$. In this step, challenger $\mathcal{C}$ can perfectly answer adversary $\mathcal{A}$'s query with s and $\mathbf{S}$, and does not need w anymore.
- In step 4, challenger $\mathcal{C}$ does not need w.

From above analysis, we observe that all the information about w leaked to the adversary $\mathcal{A}$, except $\mathbf{S}$, is by the sketch $s \leftarrow \mathsf{SS.Gen}(\mathsf{w})$. Since our SS is $(\mathbb{Z}_q^l, \mathfrak{m}, \hat{\mathfrak{m}}, t)$-secure sketch and $\widetilde{H}(W) \geq \mathfrak{m}$, we have

$$\widetilde{H}(W|\mathsf{SS.Gen}(W)) \geq \hat{\mathfrak{m}}. \tag{5}$$

By the leftover hash lemma (Lemma 1), we have the statistical distance between $\mathbf{S}$ and $\mathbf{U}$ is less than $2^{-\omega(\log \lambda)}$, where $\mathbf{S} \leftarrow \mathsf{H_i}(\mathsf{w})$ and $\mathbf{U} \leftarrow_\$ \mathbb{Z}_q^{n\times l}$. The lemma follows. □

Game G_3 : It is the same as G_2, except that in G_3, $\mathbf{B}, \mathbf{B}_i$ are uniformly sampled from $\mathbb{Z}_q^{m\times(n+l)}$. More precisely,

2. Challenger $\mathcal{C}$ samples $\mathsf{w} \leftarrow W$, invokes $s \leftarrow \mathsf{SS.Gen}(\mathsf{w})$, samples $\mathbf{S} \leftarrow_\$ \mathbb{Z}_q^{n\times l}$, $\underline{\mathbf{B} \leftarrow_\$ \mathbb{Z}_q^{m\times(n+l)}}$, $\mathbf{x} \leftarrow_\$ \{0,1\}^m$, and $\mathbf{m} \leftarrow_\$ \{0,1\}^l$, sets $\mathbf{c}^\top := \mathbf{x}^\top\mathbf{B} + (\mathbf{0}^\top, \mathbf{m}^\top \cdot \lfloor \frac{q}{2} \rceil)$, $\mathsf{P} := (s, \mathbf{c})$ and $\mathsf{R} := \mathbf{m}$. Finally, it returns (P, R) to $\mathcal{A}$.
3. Upon receiving a shift $\delta_i \in \mathcal{M}$ satisfying $\mathsf{dis}(\delta_i) \leq t$ from $\mathcal{A}$, challenger $\mathcal{C}$ invokes $s_i = s + \mathsf{SS.Gen}(\delta_i)$, $\mathbf{S}_i = \mathbf{S} + \mathsf{H_i}(\delta_i)$, $\underline{\text{samples}\mathbf{B}_i \leftarrow_\$ \mathbb{Z}_q^{m\times(n+l)}}$, $\mathbf{x}_i \leftarrow_\$ \{0,1\}^m$ and $\mathbf{m}_i \leftarrow_\$ \{0,1\}^l$, sets $\mathbf{c}_i^\top := \mathbf{x}_i^\top\mathbf{B}_i + (\mathbf{0}^\top, \mathbf{m}_i^\top \cdot \lfloor \frac{q}{2} \rceil)$, $\mathsf{P}_i := (s_i, \mathbf{c}_i)$ and $\mathsf{R}_i := \mathbf{m}_i$. Finally, it returns $(\mathsf{P}_i, \mathsf{R}_i)$ to $\mathcal{A}$.

Lemma 9.

$$|\Pr[\mathsf{G}_2 \Rightarrow 1] - \Pr[\mathsf{G}_3 \Rightarrow 1]| \leq \mathsf{Adv}_{\mathsf{LWE},\mathcal{B}}^{n,(\rho+1)m,l,q,\chi}(\lambda).$$

Proof. We prove this lemma by showing that if there exists a PPT adversary $\mathcal{A}$ such that $|\Pr[\mathsf{G}_2 \Rightarrow 1] - \Pr[\mathsf{G}_3 \Rightarrow 1]| = \epsilon$, then we can construct a PPT algorithm $\mathcal{B}$, which can solve the decisional $\mathsf{LWE}_{n,(\rho+1)m,l,q,\chi}$ problem with the same probability ϵ. Algorithm $\mathcal{B}$ proceeds as follows.

1. Algorithm $\mathcal{B}$ samples $\mathsf{H_i} \leftarrow_\$ \mathcal{H}$, sets $\mathsf{pp} := \mathsf{H_i}$, and returns pp to $\mathcal{A}$.
2. Algorithm $\mathcal{B}$ queries its own oracle to obtain $\mathbf{B}$. Then it samples $\mathsf{w} \leftarrow W$, invokes $s \leftarrow \mathsf{SS.Gen}(\mathsf{w})$, samples $\mathbf{x} \leftarrow_\$ \{0,1\}^m$ and $\mathbf{m} \leftarrow_\$ \{0,1\}^l$, sets $\mathbf{c}^\top := \mathbf{x}^\top\mathbf{B} + (\mathbf{0}^\top,\ \mathbf{m}^\top \cdot \lfloor \frac{q}{2} \rceil)$, $\mathsf{P} := (s, \mathbf{c})$ and $\mathsf{R} := \mathbf{m}$. Finally, it returns (P, R) to $\mathcal{A}$.

3. Upon receiving a shift $\delta_i \in \mathcal{M}$ from $\mathcal{A}$, algorithm $\mathcal{B}$ computes $\mathbf{S}'_i = \mathsf{H_i}(\delta_i)$ and sets $s_i = s + \mathsf{SS.Gen}(\delta_i)$, then queries its own oracle to obtain $\mathbf{B}'_i = (\mathbf{A}_i,\ \mathbf{C}_i)$, sets $\mathbf{B}_i = (\mathbf{A}_i,\ \mathbf{C}_i + \mathbf{A}_i\mathbf{S}'_i)$, samples $\mathbf{x}_i \leftarrow_\$ \{0,1\}^m$ and $\mathbf{m}_i \leftarrow_\$ \{0,1\}^l$, sets $\mathbf{c}_i^\top := \mathbf{x}_i^\top \mathbf{B}_i + (\mathbf{0}^\top, \mathbf{m}_i^\top \cdot \lfloor \frac{q}{2} \rceil)$, $\mathsf{P}_i := (s_i, \mathbf{c}_i)$ and $\mathsf{R}_i := \mathbf{m}_i$. Finally, it returns $(\mathsf{P}_i, \mathsf{R}_i)$ to $\mathcal{A}$.
4. As long as $\mathcal{A}$ outputs a guessing bit β', $\mathcal{B}$ outputs β' as its own guess.

Now we analyse the advantage of $\mathcal{B}$.

- If $\mathcal{B}$'s oracle is $\mathcal{O}_{\mathsf{LWE}}(\mathbf{S})$, the oracle will return LWE samples $\mathbf{B} = (\mathbf{A},\ \mathbf{AS} + \mathbf{E})$ and $\mathbf{B}'_i = (\mathbf{A}_i,\ \mathbf{A}_i\mathbf{S} + \mathbf{E}_i)$, where $\mathbf{A} \leftarrow_\$ \mathbb{Z}_q^{m\times n}$, $\mathbf{S} \leftarrow_\$ \mathbb{Z}_q^{n\times l}$, $\mathbf{E} \leftarrow \chi^{m\times l}$, $\mathbf{A}_i \leftarrow_\$ \mathbb{Z}_q^{m\times n}$ and $\mathbf{E}_i \leftarrow \chi^{m\times l}$, then $\mathbf{B}_i = (\mathbf{A}_i,\ \mathbf{C}_i + \mathbf{A}_i\mathbf{S}'_i) = (\mathbf{A}_i,\ \mathbf{A}_i\mathbf{S} + \mathbf{E}_i + \mathbf{A}_i\mathsf{H_i}(\delta_i)) = (\mathbf{A}_i,\ \mathbf{A}_i(\mathbf{S} + \mathsf{H_i}(\delta_i)) + \mathbf{E}_i) = (\mathbf{A}_i,\ \mathbf{A}_i\mathbf{S}_i + \mathbf{E}_i)$. In this case, algorithm $\mathcal{B}$ perfectly simulates G_2 for $\mathcal{A}$.
- If $\mathcal{B}$'s oracle is $\mathcal{O}_{\mathsf{U}}$, the oracle will return uniform samples $\mathbf{B}$, $\mathbf{B}'_i$, where $\mathbf{B} \leftarrow_\$ \mathbb{Z}_q^{m\times(n+l)}$, $\mathbf{B}'_i \leftarrow_\$ \mathbb{Z}_q^{m\times(n+l)}$, then $\mathbf{B}_i = (\mathbf{A}_i,\ \mathbf{C}_i + \mathbf{A}_i\mathbf{S}'_i) = (\mathbf{A}_i,\ \mathbf{C}_i) + (0,\ \mathbf{A}_i\mathbf{S}'_i) = \mathbf{B}'_i + (0,\ \mathbf{A}_i\mathbf{S}'_i)$ is uniformly distributed in $\mathbb{Z}_q^{m\times(n+l)}$. In this case, algorithm $\mathcal{B}$ perfectly simulates G_3 for $\mathcal{A}$.

Consequently, $|\Pr[\mathsf{G}_2 \Rightarrow 1] - \Pr[\mathsf{G}_3 \Rightarrow 1]| \le \mathsf{Adv}_{\mathsf{LWE},\mathcal{B}}^{n,(\rho+1)m,q,\chi}(\lambda)$. □

Game G_4 : It is the same as G_3, except that in G_4, the challenger uniformly chooses U from $\{0,1\}^l$, and returns (P, U) to $\mathcal{A}$ instead of returning (P, R) to $\mathcal{A}$.

Lemma 10. $|\Pr[\mathsf{G}_3 \Rightarrow 1] - \Pr[\mathsf{G}_4 \Rightarrow 1]| \le 2^{-\omega(\log\lambda)}$.

Proof. We will show that G_4 is statistically indistinguishable from the G_3. Note that in G_4, $\mathbf{B}$ is uniformly chosen from $\mathbb{Z}_q^{m\times(n+l)}$ and $\mathbf{x} \leftarrow_\$ \{0,1\}^m$, since $m \ge (n+l)\log q + \omega(\log\lambda)$, by the leftover hash lemma (Lemma 1), we have $\mathbf{x}^\top\mathbf{B}$ is $2^{-\omega(\log\lambda)}$ statistically close to the uniform distribution over $\mathbb{Z}_q^{n+l}$. Consequently, $\mathsf{R} := \mathbf{m}$ is concealed, and $|\Pr[\mathsf{G}_3 \Rightarrow 1] - \Pr[\mathsf{G}_4 \Rightarrow 1]| \le 2^{-\omega(\log\lambda)}$ follows. □

Game G_5 : It is the same as G_4, except that in G_5, $\mathbf{B}, \mathbf{B}'_i$ are changed back to LWE samples.

Lemma 11.

$$|\Pr[\mathsf{G}_4 \Rightarrow 1] - \Pr[\mathsf{G}_5 \Rightarrow 1]| \le \mathsf{Adv}_{\mathsf{LWE},\mathcal{B}}^{n,(\rho+1)m,l,q,\chi}(\lambda).$$

Proof. The proof is similar to the proof of Lemma 9. We omit it here. □

Game G_6 : It is the same as G_5, except that $\mathbf{S} \leftarrow_\$ \mathbb{Z}_q^{n\times l}$ in G_5 is changed back to $\mathbf{S} := \mathsf{H_i}(\mathsf{w})$ in G_6.

Lemma 12.

$$|\Pr[\mathsf{G}_5 \Rightarrow 1] - \Pr[\mathsf{G}_6 \Rightarrow 1]| \le 2^{-\omega(\log\lambda)}.$$

Proof. The proof is similar to the proof of Lemma 8. We omit it here.

Game G_7 : It is the same as G_6, except that

- $s_i := s + \mathsf{SS.Gen}(\delta_i)$ now is changed back to $s_i \leftarrow \mathsf{SS.Gen}(\mathsf{w} + \delta_i)$.
- $\mathbf{S}_i := \mathbf{S} + \mathsf{H_i}(\delta_i)$ now is changed back to $\mathbf{S}_i := \mathsf{H_i}(\mathsf{w} + \delta_i)$.

Lemma 13. $\Pr[\mathsf{G}_6 \Rightarrow 1] = \Pr[\mathsf{G}_7 \Rightarrow 1]$.

Proof. The proof is identical to the proof of Lemma 7. We omit it here. □

Observe that G_7 is identical to $\mathsf{Exp}^{\mathsf{reu}}_{\mathsf{rFE},\mathcal{A}}(0)$, as a result

$$\Pr[\mathsf{G}_7 \Rightarrow 1] = \Pr[\mathsf{Exp}^{\mathsf{reu}}_{\mathsf{rFE},\mathcal{A}}(0) \Rightarrow 1]. \tag{6}$$

Combining Eq. (4), Lemmas 7–13 and Eq. (6) together, we have

$$\mathsf{Adv}^{\mathsf{reu}}_{\mathsf{rFE},\mathcal{A}}(1^\lambda) \leq 2^{-\omega(\log \lambda)} + 2\mathsf{Adv}^{n,(\rho+1)m,l,q,\chi}_{\mathsf{LWE},\mathcal{B}}(\lambda).$$

This completes the proof of Theorem 1. □

If we instantiate SS and $\mathsf{H_i}$ with the syndrome-based secure sketch as defined in (3) and homomorphic universal hashing as defined in (1), the construction of rFE in Fig. 1 results in a reusable fuzzy extractor from the LWE assumption, which is resilient to linear fraction of errors.

4 Conclusion

Traditional fuzzy extractor distills an almost uniform output from a non-uniform noisy source, but the distillation is implemented only once. In this paper, we study on reusable fuzzy extractor which enables multiple distillations from the same non-uniform noisy source and provide the first reusable fuzzy extractor which is resilient to linear fraction of errors from the LWE assumption. In the construction, a secure sketch is used to correct errors, an LWE-type encryption is used to break the correlations between multiple distilled strings, and universal hashing is used to extract uniform strings. The reusability of our construction benefits from the LWE assumption and the homomorphic properties of secure sketch and universal hashing.

Acknowledgements. This work was supported by the National Natural Science Foundation of China (NSFC No. 61672346).

References

1. Agrawal, S., Boneh, D., Boyen, X.: Efficient lattice (H)IBE in the standard model. In: Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 553–572. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-13190-5_28
2. Alamélou, Q., Berthier, P.E., Cachet, C., Cauchie, S., Fuller, B., Gaborit, P., Simhadri, S.: Pseudoentropic isometries: a new framework for fuzzy extractor reusability (2016). http://eprint.iacr.org/2016/1100
3. Apon, D., Cho, C., Eldefrawy, K., Katz, J.: Efficient, reusable fuzzy extractors from LWE. In: Dolev, S., Lodha, S. (eds.) CSCML 2017. LNCS, vol. 10332, pp. 1–18. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-60080-2_1
4. Bennett, C.H., Brassard, G., Robert, J.: Privacy amplification by public discussion. SIAM J. Comput. **17**(2), 210–229 (1988). https://doi.org/10.1137/0217014
5. Blanton, M., Aliasgari, M.: On the (non-)reusability of fuzzy sketches and extractors and security in the computational setting. In: Lopez, J., Samarati, P. (eds.) SECRYPT 2011, pp. 68–77. SciTePress (2011)
6. Blanton, M., Aliasgari, M.: Analysis of reusability of secure sketches and fuzzy extractors. IEEE Trans. Inf. Forensics Secur. **8**(9), 1433–1445 (2013). https://doi.org/10.1109/TIFS.2013.2272786
7. Boyen, X.: Reusable cryptographic fuzzy extractors. In: Atluri, V., Pfitzmann, B., McDaniel, P.D. (eds.) CCS 2004, pp. 82–91. ACM, New York (2004). https://doi.org/10.1145/1030083.1030096
8. Canetti, R., Fuller, B., Paneth, O., Reyzin, L., Smith, A.: Reusable fuzzy extractors for low-entropy distributions. In: Fischlin, M., Coron, J.-S. (eds.) EUROCRYPT 2016. LNCS, vol. 9665, pp. 117–146. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-49890-3_5
9. Dodis, Y., Ostrovsky, R., Reyzin, L., Smith, A.D.: Fuzzy extractors: how to generate strong keys from biometrics and other noisy data. SIAM J. Comput. **38**(1), 97–139 (2008)
10. Dodis, Y., Reyzin, L., Smith, A.: Fuzzy extractors: how to generate strong keys from biometrics and other noisy data. In: Cachin, C., Camenisch, J. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 523–540. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-24676-3_31
11. Fuller, B., Meng, X., Reyzin, L.: Computational fuzzy extractors. In: Sako, K., Sarkar, P. (eds.) ASIACRYPT 2013. LNCS, vol. 8269, pp. 174–193. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-42033-7_10
12. Peikert, C., Vaikuntanathan, V., Waters, B.: A framework for efficient and composable oblivious transfer. In: Wagner, D. (ed.) CRYPTO 2008. LNCS, vol. 5157, pp. 554–571. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-85174-5_31
13. Regev, O.: On lattices, learning with errors, random linear codes, and cryptography. In: Gabow, H.N., Fagin, R. (eds.) STOC 2005, pp. 84–93. ACM, New York (2005). https://doi.org/10.1145/1060590.1060603
14. Regev, O.: The learning with errors problem (invited survey). In: CCC 2010, pp. 191–204. IEEE Computer Society (2010). https://doi.org/10.1109/CCC.2010.26
15. Shoup, V.: A Computational Introduction to Number Theory and Algebra. Cambridge University Press, Cambridge (2006)
16. Simoens, K., Tuyls, P., Preneel, B.: Privacy weaknesses in biometric sketches. In: 30th IEEE Symposium on Security and Privacy, pp. 188–203. IEEE Computer Society (2009). https://doi.org/10.1109/SP.2009.24

17. Tanamoto, T., Yasuda, S., Takaya, S., Fujita, S.: Physically unclonable function using an initial waveform of ring oscillators. IEEE Trans. Circuits Syst. **64**(7), 827–831 (2017). https://doi.org/10.1109/TCSII.2016.2602828
18. Valsesia, D., Coluccia, G., Bianchi, T., Magli, E.: User authentication via PRNU-based physical unclonable functions. IEEE Trans. Inf. Forensics Secur. **12**(8), 1941–1956 (2017). https://doi.org/10.1109/TIFS.2017.2697402
19. Wilde, M.M.: Quantum Information Theory. Cambridge University Press, Cambridge (2017). https://doi.org/10.1017/9781316809976