Solving LWE via List Decoding

Mingqiang Wang[1], Kunxian Xia[1], and Jincheng Zhuang[3]

[1]School of Mathematics, Shandong University
wangmingqiang@sdu.edu.cn,
[2]Institute of Advanced Study, Tsinghua University
xiaoyunwang@tsinghua.edu.cn
[3]State Key Laboratory of Information Security
Institute of Information Engineering, Chinese Academy of Sciences
zhuangjincheng@iie.ac.cn

Abstract

Learning with errors (LWE) was introduced by Regev in 2005, which enjoys attractive worst-case hardness properties. It has served as the foundation for a variety of cryptographic schemes. There are two main types of attacks against LWE: one for the decision version of LWE [31], the other for the search version of LWE [26] [24].

In this paper, we apply the list decoding method to solve search version of LWE. Our algorithm runs in probabilistic polynomial time and results in specific security estimates for a large range of parameters. To our knowledge, it is the first time to apply the list decoding method to recover the key of LWE. Our algorithm improves Laine and Lauter's result [24].

Key words: Hidden number problem, LWE, list decoding, multiplication code.

1 Introduction

Inspired by the hardness of the learning from parity with noise problem, Regev [35] introduced the generalized learning with errors (LWE) problem in 2005. Informally, the LWE problem is to distinguish (decision version) or solve (search version) random congruent linear equations with some amount of noise. In order to explain the apparent difficulty of the problem, Regev established quantum reductions from certain approximate lattice problems, namely the decision version of the shortest vector problem (GapSVP) and the shortest independent vectors problem (SIVP) within approximation factor $\gamma = poly(n)$ to LWE. Thus if one can solve the average case of LWE problem efficiently, then one can solve the worst case of aforementioned approximate lattice problems efficiently via quantum algorithms. Peikert [33] and Brakerski et al. [7] have managed to establish the hardness of LWE via classical reductions based on more restricted lattice problems.

The LWE problem has been very attractive as a plausible candidate primitive since its proposition. There are mainly two lines of research around this problem and its variant (say, Ring-LWE [28]): one line focuses on building cryptographic schemes based on the LWE problem as an assumed firm foundation; another line focuses on analyzing the security of this primitive. These two lines enjoy asymmetric developments, as outlined in the following.

A variety of cryptographic schemes have been proposed utilizing the LWE as a primitive. These applications range over several aspects of cryptology. To name a few, Regev [35], Peikert and Waters [34], Peikert [33], Lindner and Peikert [26], Micciancio and Peikert [29] have designed public key encryption schemes based on LWE problem respectively. Gentry, Peikert, and Vaikuntanathan [18], Cash, Hofheinz, Kiltz, and Peikert [13], Agrawal, Boneh, and Boyen [1] [2] have proposed identity-based encryption (IBE) schemes based on LWE problem respectively. Brakerski and Vaikuntanathan [10], Brakerski, Gentry, and Vaikuntanathan [9] have presented fully homomorphic encryption (FHE) schemes based on LWE problem respectively. Besides, there are many other applications based on LWE.

Compared with the diverse applications of LWE, not so many efficient cryptanalysis algorithms are known to attack LWE. Micciancio and Regev [31] presented the distinguishing attack for the decision version of LWE. Besides, they recommended certain parameter ranges to avoid such attack. Lindner and Peiker [26] provided the decoding attack against the search version of LWE. Their attack combined the basis reduction technique and a post-reduction decoding algorithm. The decoding attack is preferable to the distinguishing attack in that it recovers the secret key. But due to its utilization of the BKZ algorithm, it does not run in polynomial time and its performance is difficult to analyze. Recently, Laine and Lauter [24] constructed an algorithm to recover the key of LWE by applying LLL algorithm [25], which runs in polynomial time. Their algorithm may some threaten homomorphic encryption schemes. They also remarked that with the pure LLL the attack does not threaten commonly recommended practical parameters.

Boneh and Venkatesan [12] introduced the hidden number problem (HNP) originally to study the bit security of Diffie-Hellman key exchange protocol. The hidden number problem has proven to be useful in different settings. Galbraith and Shani [19] defined the multivariate hidden number problem as a generalization of the hidden number problem, and developed the tool of Fourier learning to solve it.

Goldreich and Levin [21] introduced a general list decoding algorithm for Hadamard code to prove hard-core predicates for general one-way functions. Akavia, Goldwasser, and Safra (AGS) [4] formalized the list decoding methodology and applied it to a broad family of conjectured one-way functions. Morillo and Ràfols [30] extended the AGS result to prove the unpredictability of every individual bit for these functions. Duc and Jetchev [15] showed how to extend to elliptic curve-based one-way functions which do not necessarily enjoy the homomorphic property. FGPS [17] proved for a weak CDH problem (*i.e.* Partial-CDH problem) the unpredictability of every single bit of one of the coordinates of the secret CDH value over finite fields $\mathbb{F}_{p^2}$. Wang *et al* [39] showed that almost all bits of the CDH value are hardcore over finite field $\mathbb{F}_{p^t}$ for $t > 1$ over a random representation of the finite field. Zhang [40] proved that any bit of each coordinate of hyperelliptic curves Diffie-Hellman secret value in genus 2 is hard as the entire Diffie-Hellman value.

Informally, list decoding algorithm works as following. Given a one-way function $f\colon \mathcal{D} \to \mathcal{R}$ and a predicate π, one would have to identify an error-correcting code $\mathcal{C}^{\pi} = \{C_{\alpha}\colon \mathcal{D} \to \{\pm 1\}\}_{\alpha \in \mathcal{D}}$ such that every input α of the one-way function is associated with a codeword C_{α}. The code needs to satisfy the following properties: *Accessibility, Concentration, Recoverability.* Then one can invert the one-way function f.

1.1 Our results

This paper aims to adding another powerful tool in the toolkit of attacking LWE. Our methodology is to relate the LWE problem to the hidden number problem, and then apply the list decoding technique to solve the resulting problem.

In this paper, we apply the list decoding method to recover the key of LWE. The list decoding approach allows to consider hardness of single bits, even for noisy oracles that only have a non-negligible advantage over the bias of the function in question. Here, we consider the most significant bit(MSB) function over $\mathbb{Z}_q$. Let $\mathbf{s} = (s_1, \ldots, s_n) \in \mathbb{Z}_q^n$ be the LWE problem the security key. The key of our method is that $MSB([\langle \mathbf{a}, \mathbf{s} \rangle]_q)$ can be learned easily from $MSB([\langle \mathbf{a}, \mathbf{s} \rangle + e]_q)$ with high probability, for the error e centered around 0 with high probability. This means we can get a noise code of $C_{\mathbf{s}}(\mathbf{x}) = MSB([\langle \mathbf{x}, \mathbf{s} \rangle]_q)$ under the uniform queries.

Our algorithm runs in probabilistic polynomial time. The success probability is high for a large range of parameters. Our result improves the work in Laine and Lauter [24]. Besides, we estimate the parameter security of some specific ranges utilizing our result.

1.2 Roadmap of the paper

In Section 2 we review some relevant background, particularly of Fourier transform and error correcting codes. In Section 3 we establish the relation between two classes of most significant bits. In Section 4 we describe our main algorithm to recover the key of LWE. In Section 5 we present some specific implication for specific parameters. In Section 6 we conclude the paper.

2 Preliminaries

2.1 Notation

We use the standard symbols $\mathbb{N}$, $\mathbb{Z}$, $\mathbb{R}$ and $\mathbb{C}$ to denote the natural numbers, the integers, the real numbers and the complex numbers, respectively. Let $\mathbb{Z}_+$ and $\mathbb{R}_+$ stand for the positive integers and reals, respectively. A function $\nu(l) \colon \mathbb{N} \to \mathbb{R}$ is *negligible* if for every constant $c \in \mathbb{R}_+$ there exists $l_c \in \mathbb{N}$ such that $\nu(l) < l^{-c}$ for all $l > l_c$. A function $\rho(l) \colon \mathbb{N} \to \mathbb{R}$ is *non-negligible* if there exists a constant $c \in \mathbb{R}_+$ and $l_c \in \mathbb{N}$ such that $\rho(l) > l^{-c}$ for all $l > l_c$. For a Boolean function $f \colon \mathcal{D} \to \{\pm 1\}$ over an arbitrary domain $\mathcal{D}$, denote by $\mathsf{maj}_f = \mathsf{max}_{\{b=\pm 1\}} \Pr_{\alpha \in \mathcal{D}}[f(\alpha) = b]$ the *bias* of f toward its majority value. By $\mathbb{Z}_m$ we denote integers modulo m, but as a set of representatives for the congruence classes we use integers in the interval $[0, m-1]$. By a subscript m we denote the unique representative of an integer modulo m within this interval.

2.2 Fourier Transform

Let $\mathbb{G}$ be a finite abelian group. For any two functions $f, g \colon \mathbb{G} \to \mathbb{C}$, their *inner product* is defined as $\langle f, g \rangle = 1/|\mathbb{G}| \sum_{x \in \mathbb{G}} \overline{f(x)} g(x)$. The l_2-norm of f on the vector space $\mathbb{C}(\mathbb{G})$ is defined as $\|f\|_2 = \sqrt{\langle f, f \rangle}$. A *character* of $\mathbb{G}$ is a homomorphism $\chi \colon \mathbb{G} \to \mathbb{C}^*$, *i.e.,* $\chi(x+y) = \chi(x)\chi(y)$ for all $x, y \in \mathbb{G}$. The set of all characters of $\mathbb{G}$ forms a *character group* $\widehat{\mathbb{G}}$, whose elements form an orthogonal basis (the *Fourier basis*) for the vector space $\mathbb{C}(\mathbb{G})$. One can then describe any function $f \in \mathbb{C}(\mathbb{G})$ via its *Fourier expansion* $\sum_{\chi \in \widehat{\mathbb{G}}} \widehat{f}(\chi)\chi$, where $\widehat{f} \colon \widehat{\mathbb{G}} \to \mathbb{C}$ is the *Fourier transform* of f and we have $\widehat{f}(\chi) = \langle f, \chi \rangle$. The coefficients $\widehat{f}(\chi)$ in the Fourier basis $\{\chi\}_{\chi \in \widehat{\mathbb{G}}}$ are the *Fourier coefficients* of f. The *weight* of a Fourier coefficient is denoted by $|\widehat{f}(\chi)|^2$. When $\mathbb{G} = \mathbb{Z}_n$ (*i.e.,* the additive group of integers modulo n) and $\widehat{\mathbb{G}} = \widehat{\mathbb{Z}}_n$, for each $\alpha \in \mathbb{Z}_n$, the α-character is defined as a function $\chi_\alpha \colon \mathbb{Z}_n \to \mathbb{C}$ such that $\chi_\alpha(x) = \omega_n^{\alpha x}$, where $\omega_n = e^{2\pi i/n}$. If Γ is a subset of $\mathbb{Z}_n$ then it is natural to consider the projection of f in set Γ, *i.e.,* $f_{|\Gamma} = \sum_{\alpha \in \Gamma} \widehat{f}(\alpha)\chi_\alpha$, where $\widehat{f}(\alpha) = \langle f, \chi_\alpha \rangle$. Since the characters are orthogonal, we have $\|f\|_2^2 = \sum_{\alpha \in \mathbb{Z}_n} |\widehat{f}(\alpha)|^2$ and $\|f_{|\Gamma}\|_2^2 = \sum_{\alpha \in \Gamma} |\widehat{f}(\alpha)|^2$.

Definition 1. (Fourier concentrated function [4]). *A function $f\colon \mathbb{Z}_n \to \mathbb{C}$ is Fourier ϵ-concentrated if there exists a set $\Gamma \subseteq \mathbb{Z}_n$ consisting of poly$(\log n, 1/\epsilon)$ characters, so that*

$$\|f - f_{|\Gamma}\|_2^2 = \sum_{\alpha \notin \Gamma} |\widehat{f}(\alpha)|^2 \leq \epsilon.$$

A function f is called Fourier concentrated if it is Fourier ϵ-concentrated for every $\epsilon > 0$.

Definition 2. (τ-heavy characters [4]). *Given a threshold $\tau > 0$ and an arbitrary function $f\colon \mathbb{Z}_n \to \mathbb{C}$, we say that a character $\chi_\alpha \in \mathsf{Heavy}_\tau(f)$ is τ-heavy if the weight of its corresponding Fourier coefficient is at least τ. The set of all heavy characters is denoted by*

$$\mathsf{Heavy}_\tau(f) = \{\chi_\alpha\colon |\widehat{f}(\alpha)|^2 \geq \tau\}.$$

2.3 Error Correcting Codes: Definitions and Properties

Error correcting codes can encode messages into codewords by adding redundant data such that it can be recovered even in the presence of noise. The code to be discussed here encodes each element $\alpha \in \mathbb{Z}_n$ into a codeword C_α of length n. Each codeword C_α can be represented by a function $C_\alpha\colon \mathbb{Z}_n \to \{\pm 1\}$. We now recall a number of definitions and lemmata [4, 15] about codes over $\mathbb{Z}_n$.

Definition 3. (Fourier concentrated code). *A code $\mathcal{C} = \{C_\alpha\colon \mathbb{Z}_n \to \{\pm 1\}\}$ is concentrated if each of its codewords C_α is Fourier concentrated.*

Definition 4. (Recoverable code). *A code $\mathcal{C} = \{C_\alpha\colon \mathbb{Z}_n \to \{\pm 1\}\}$ is recoverable, if there exists a recovery algorithm that, given a character $\chi \in \widehat{\mathbb{Z}}_n$ and a threshold τ, returns in time poly$(\log n, 1/\tau)$ a list of all elements α associated with codewords C_α for which χ is a τ-heavy coefficient (i.e., $\{\alpha \in \mathbb{Z}_n\colon \chi \in \mathsf{Heavy}_\tau(C_\alpha)\}$).*

Lemma 1 below shows that in a concentrated code $\mathcal{C}$, any corrupted ("noisy") version $\widetilde{C}_\alpha$ of codeword C_α share at least one heavy coefficient with C_α. Lemma 2 shows that when given query access to any function f one can efficiently learn all its heavy characters.

Lemma 1. ([4, Lemma 1]). *Let $f, g\colon \mathbb{Z}_n \to \{\pm 1\}$ such that f is concentrated and for some $\epsilon > 0$,*

$$\Pr_{\alpha \in \mathbb{Z}_n}[f(\alpha) = g(\alpha)] \geq \mathsf{maj}_f + \epsilon.$$

There exists a threshold τ such that $1/\tau \in$ poly$(1/\epsilon, \log n)$, and there exists a non-trivial character $\chi \neq 0$ heavy for f and g: $\chi \in \mathsf{Heavy}_\tau(f) \cap \mathsf{Heavy}_\tau(g)$.

Lemma 2. ([4, Theorem 6]). *There is a probabilistic algorithm that, given query access to $w\colon \mathbb{Z}_n \to \{\pm 1\}$, $\tau > 0$ and $0 < \delta < 1$, outputs a list L of $O(1/\tau)$ characters containing $\mathsf{Heavy}_\tau(w)$ with probability at least $1 - \delta$, whose running time is $\widetilde{O}\left(\log(n) \cdot \ln^2 \frac{(1/\delta)}{\tau^{5.5}}\right)$.*

3 Learning the significant bit of the LWE sample

3.1 LWE problem

Definition 5. *(**search-LWE**). Let n be a security parameter, $q(n)$ a prime integer modulus, and χ an error distribution over $\mathbb{Z}_q$. Let $\mathbf{s} \in \mathbb{Z}_q^n$ be a fixed secret vector chosen uniformly at random. Given access to d samples of the form*

$$(\mathbf{a}, [\langle \mathbf{a}, \mathbf{s} \rangle + e]_q) \in \mathbb{Z}_q^n \times \mathbb{Z}_q,$$

where $\mathbf{a} \in \mathbb{Z}_q^n$ are chosen uniformly at random and e are sampled from the error distribution χ, the problem search-$LWE_{n,\chi}$ is to recover $\mathbf{s}$.

In practice, the distribution χ is always taken to be a *discrete Guassian distribution* $D_{\mathbb{Z},\sigma}$. This is the probability distribution over $\mathbb{Z}$ that assigns to an integer x a probability

$$\Pr(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp(-\frac{x^2}{2\sigma^2}),$$

where σ is the standard deviation.

3.2 Extract the most significant bit

Let $\mathbf{s} = (s_1, \ldots, s_n) \in \mathbb{Z}_q^n$, and $\mathbf{a} = (a_1, \ldots, a_n) \in \mathbb{Z}_q^n$. Define the most significant bit(MSB) function over $\mathbb{Z}_q$ as the following

$$MSB(x) = \begin{cases} 0, & 0 \leq x < \frac{q}{2}, \\ 1, & \frac{q}{2} < x < q. \end{cases}$$

In the LWE problem the security key $\mathbf{s}$ should be not zero, i.e. not all $s_i = 0$. It is easy to see that $[\langle \mathbf{a}, \mathbf{s} \rangle + e]_q$ is uniform when $\mathbf{a}$ is uniform in $\mathbb{Z}_q^n$. Hence the bias of $MSB(\langle \mathbf{x}, \mathbf{s} \rangle)$ is $\frac{1}{2}$.

Given an LWE sample, it's trivial to calculate its most significant bit $MSB([\langle \mathbf{a}, \mathbf{s} \rangle + e]_q)$. Now we observe that $MSB([\langle \mathbf{a}, \mathbf{s} \rangle]_q)$ can be learned easily from $MSB([\langle \mathbf{a}, \mathbf{s} \rangle + e]_q)$ with high probability, for the error e centered around 0 with high probability.

The following lemma is well known.

Lemma 3. *Let $B \geq \sigma$. Then*

$$\Pr[|D_{\mathbb{Z},\sigma}| \geq B] \leq \frac{B}{\sigma} \exp(\frac{1}{2} - \frac{B^2}{2\sigma^2}).$$

Lemma 3 implies that for $\delta \geq \sigma$

$$\Pr[|e| \leq \delta] > 1 - \frac{\delta}{\sigma} \exp(\frac{1}{2} - \frac{\delta^2}{2\sigma^2}).$$

Since $\mathbf{a}$ is chosen uniformly from $\mathbb{Z}_q^n$ and $\mathbf{s}$ is a fixed vector, their inner product $\langle \mathbf{a}, \mathbf{s} \rangle$ is a random variable distributed uniformly on $\mathbb{Z}_q$.

Lemma 4. *For $\delta \geq \sigma$, the probability that the most significant bit(MSB) of X is equal to the most significant bit of $X + e$ is at least $(1 - \frac{4\delta}{q})(1 - \frac{\delta}{\sigma} \exp(\frac{1}{2} - \frac{\delta^2}{2\sigma^2}))$, where X is uniformly distributed on $\mathbb{Z}_q$ and e is chosen according to $D_{\mathbb{Z},\sigma}$, moreover, X and e are independent variables.*

Proof. Using conditional probability formulas, we have

$$\begin{aligned}
& \Pr[MSB(X) = \mathrm{MSB}(X+e)] \\
\geq\ & \Pr[MSB(X) = MSB(X+e) \mid |e| \leq \delta] \\
=\ & \Pr[0 \leq X+e < \frac{q}{2}, 0 \leq X < \frac{q}{2} \mid |e| \leq \delta] \\
& + \Pr[\frac{q}{2} \leq X+e < q, \frac{q}{2} \leq X < q \mid |e| \leq \delta] \\
\geq\ & (\Pr[\delta \leq X < \frac{q}{2} - \delta] + \Pr[\frac{q}{2} + \delta \leq X < q - \delta]) \times \Pr[|e| \leq \delta] \\
\geq\ & (\frac{q-4\delta}{q}) \times \Pr[|e| \leq \delta].
\end{aligned}$$

Therefore, we get the desired result

$$\Pr[MSB(X) = MSB(X+e)] \geq (1 - \frac{4\delta}{q})(1 - \frac{\delta}{\sigma}\exp(\frac{1}{2} - \frac{\delta^2}{2\sigma^2})).$$

This completes the proof of this lemma. □

Corollary 1. *The probability that $MSB(\langle \mathbf{a}, \mathbf{s} \rangle)$ can be read correctly from $MSB(\langle \mathbf{a}, \mathbf{s} \rangle + e)$ is at least $(1 - \frac{4\delta}{q})(1 - \frac{\delta}{\sigma}\exp(\frac{1}{2} - \frac{\delta^2}{2\sigma^2}))$.*

Take $\delta = k\sigma$, for $k > 1$, then

$$\Pr[MSB([\langle \mathbf{a}, \mathbf{s} \rangle + e]_q) = MSB([\langle \mathbf{a}, \mathbf{s} \rangle]_q)] > (1 - \frac{4k\sigma}{q})(1 - k\exp(\frac{1}{2})\exp(\frac{-k^2}{2})).$$

Assume ϵ is a non-negligible quantity. Fixed q and σ, define

$$S_{q,\sigma,\epsilon} = \{k \mid \Pr[MSB([\langle \mathbf{a}, \mathbf{s} \rangle + e]_q) = MSB([\langle \mathbf{a}, \mathbf{s} \rangle]_q) > \frac{1}{2} + \epsilon\}. \quad (1)$$

It is easy to see that S_ϵ is nonempty when σ is sufficient small compared with q.

4 Algorithm to recovery the key of LWE

4.1 Hidden number problems

In order to study the bit security of Diffie-Hellman key exchange, Boneh and Venkatesan [12] introduced the hidden number problem over $\mathbb{F}_p$. Fix p and k. Let $\mathcal{O}_\alpha(x)$ be an oracle that on input x computes the k most significant bits of $\alpha x \bmod p$:

$$\mathcal{O}_\alpha(x) = MSB_k(\alpha x).$$

The task is to compute the hidden number $\alpha \bmod p$, in expected polynomial (in $\log p$) time, given access to the oracle $\mathcal{O}_\alpha(x)$.

In this paper, we consider the one bit hidden number problem.

Definition 6. *let $\alpha \neq 0$ be a secret element in $\mathbb{F}_p^*$ and let $f : \mathbb{F}_p^* \to \{-1, 1\}$. The goal is to compute the secret element α in polynomial time, using oracle access to the function $C_\alpha(x) = f(x\alpha)$.*

Theorem 1. *[3] Let $\mathcal{A}$ be an algorithm that learns the τ-heavy Fourier coefficients C_α of functions defined over $\mathbb{F}_p$. For any concentrated function $f : \mathbb{F}_p \to \{-1, 1\}$, there exists an algorithm that solves the hidden number problem in $\mathbb{F}_p^*$.*

4.2 Main result

Let ϵ be the probability advantage of $MSB(\langle \mathbf{a}, \mathbf{s}\rangle + e) = MSB(\langle \mathbf{a}, \mathbf{s}\rangle)$, and β be the bias of $MSB(\langle \mathbf{x}, \mathbf{s}\rangle)$.

Theorem 2. *Let $\mathbf{s} \in \mathbb{Z}_q^n$, given an oracle $\mathcal{O}$ that, given $\mathbf{x} \in \mathbb{Z}_q^n$ as input, outputs the LWE sample $\langle \mathbf{x}, \mathbf{s}\rangle + e$, where $e \in D_{\mathbb{Z},\sigma}$, and $S_{q,\sigma,\epsilon}$ is nonempty. Then, there exists a probabilistic polynomial algorithm that solves the LWE over $\mathbb{Z}_q^n$.*

Define codes. For any element $\mathbf{x} \in \mathbb{Z}_q^n$, we construct the following encoding of $\langle \mathbf{x}, \mathbf{s}\rangle$:

$$C_{\mathbf{s}} \colon \mathbb{Z}_q^n \to \{\pm 1\} \text{ such that } C_{\mathbf{s}}(\mathbf{x}) = MSB(\langle \mathbf{x}, \mathbf{s}\rangle).$$

Since, given access to the LWE oracle, it is easy to verify wether $s_i = 0$ or not, for $i = 1, \ldots, n$. Without loss generality, we assume that $s_1 \neq 0$. For any element $x \in \mathbb{Z}_q$,

$$\widetilde{C}_{\mathbf{s}} \colon \mathbb{Z}_q \to \{\pm 1\} \text{ such that } \widetilde{C}_{\mathbf{s}}(x) = MSB(\langle (x, 0, \ldots, 0), \mathbf{s}\rangle).$$

The following lemma shows that $C_{\mathbf{s}}$ is concentrated.

Lemma 5. *[19] Let $f : \mathbb{F}_q \to \{-1, 1\}$, let $\mathbf{s} = (s_1, \ldots, s_n) \in \mathbb{Z}_q^n$ be such that not all $s_i \neq 0$, and let $f_{\mathbf{s}} \colon \mathbb{Z}_q^n \to \{\pm 1\}$ be the function $f_{\mathbf{s}}(\mathbf{x}) = f(\mathbf{s} \cdot \mathbf{x})$. Then, $Heavy_\tau(f) = \{c_1, \ldots, c_t\}$ if and only if $Heavy_\tau(f_{\mathbf{s}}) = \{(c_i s_1, \ldots, c_i s_n) | 1 \leq i \leq t\}$.*

Proof of Theorem 2. The proof of this theorem can be divided two steps. In the Access step, we use the uniform distribution model, at each time the learning algorithm queries, its choice is uniform.
Access to the codes. Let $\mathcal{O}$ be the oracle that access to the LWE samples, i.e. $\mathcal{O}$ takes $\mathbf{x} \in \mathbb{Z}_q^n$ as input, returns $\mathcal{O}(\mathbf{x}) = \langle \mathbf{x}, \mathbf{s}\rangle + e$.

According to the output of $\mathcal{O}$, for $\mathbf{x} \in \mathbb{Z}_q^n$, we can define a corrupted codeword $C'_{\mathbf{s}}$ of $C_{\mathbf{s}}$ as $C'_{\mathbf{s}}(\mathbf{x}) = MSB(\mathcal{O}(\mathbf{x}))$. Therefore, we have $|\Pr_{\mathbf{x}}[C_{\mathbf{s}}(\mathbf{x}) = \widetilde{C}_{\mathbf{s}}(\mathbf{x})]| \geq \beta + \epsilon$. Similarly, for $x \in \mathbb{Z}_q$, we can define a corrupted codeword $\widetilde{C}'_{\mathbf{s}}$ of $\widetilde{C}_{\mathbf{s}}$ as $\widetilde{C}'_{\mathbf{s}}(x) = MSB(\mathcal{O}(x, 0, \ldots, 0))$ and $|\Pr_x[\widetilde{C}'_{\mathbf{s}}(x) = \widetilde{C}_{\mathbf{s}}(x)]| \geq \beta + \epsilon$.

Recover the key. Akavia [3] proved that the most-significant-bit function is concentrated, i.e. $\widetilde{C}_{\mathbf{s}}$ is concentrated. By Theorem 1, there is a probability algorithm to recover the key s_1.

By Lemma 2, there is a probabilistic algorithm that, given query access to $C'_{\mathbf{s}} : \mathbb{Z}_q \to \{\pm 1\}$, $\tau > 0$, outputs two lists $L_{\mathbf{s}}$ of $O(1/\tau)$ characters containing $\mathsf{Heavy}_\tau(C'_{\mathbf{s}})$, where τ satisfies $1/\tau \in poly(1/\epsilon, \log n)$. By Lemma 1, there exists a non-trivial character $\chi_{(cs_1, cs_2, \cdots, cs_n)} \neq 0$ such that $\chi_{(cs_1, cs_2, \cdots, cs_n)} \in \mathsf{Heavy}_\tau(C_{\mathbf{s}}) \cap \mathsf{Heavy}_\tau(C'_{\mathbf{s}})$. This implies that $\chi_{(cs_1, cs_2, \cdots, cs_n)} \in L_{\mathbf{s}}$. Therefore, $\mathbf{s}$ belongs to the following set

$$L = \{(s_1, s_1\beta_1^{-1}\beta_2, s_1\beta_1^{-1}\beta_3, \ldots, s_1\beta_1^{-1}\beta_n) | \chi_{(\beta_1, \beta_2, \cdots, \beta_n) \in L_{\mathbf{s}}}\}.$$

This shows that there exists an algorithm $\mathcal{A}$ to recover $\mathbf{s}$.

5 Implication of security

In this section we determine the range of parameters that can be attacked by our method.

Theorem 2 implies that given a noise oracle that can predict the value of $MSB(\langle \mathbf{a}, \mathbf{s}\rangle)$ with non-negligible probability, we can recover the secret key with high probability. Next we need to determine under what conditions can we achieve that goal.

Table 1: Maximum of σ vulnerable to our attack ($\epsilon = \frac{1}{q}$)

$\log_2 q$	10	11	12	13	14	15	16	17	18	19
$\sigma \leq$	10.9	21.9	43.8	87.7	175.6	351.3	702.7	1405.4	2811	5622.0

In order to get $MSB(\langle \mathbf{a}, \mathbf{s} \rangle)$, we use the oracle of $MSB(\langle \mathbf{a}, \mathbf{s} \rangle + e)$. By the discussion in Section 2, we have

$$\Pr_{\mathbf{a} \in \mathbb{Z}_q^n}[MSB(\langle \mathbf{a}, \mathbf{s} \rangle) = MSB(\langle \mathbf{a}, \mathbf{s} \rangle + e)] \geq \max_{k>1}\{(1 - \frac{4k\sigma}{q})(1 - k \exp^{\frac{1}{2}(1-k^2)})\}.$$

To make our attack work, we need to make sure the above probability is greater than $\frac{1}{2} + \varepsilon$, where ε is some non-negligible function of the modulus q. That is,

$$\max_{k>1}\{(1 - \frac{4k\sigma}{q})(1 - k \exp^{\frac{1}{2}(1-k^2)})\} \geq \frac{1}{2} + \varepsilon.$$

Clearly, this is equivalent to

$$\sigma \leq \max_{k>1} \frac{q}{4k} \left(1 - \frac{\frac{1}{2} + \varepsilon}{1 - k \exp^{\frac{1}{2}(1-k^2)}}\right).$$

Therefore, we need only to find out the maximum of the right side of above inequality. By the inequality

$$1 - k \exp^{\frac{1}{2}(1-k^2)} > 1 - k(1 + \frac{k^2 - 1}{2})^{-1} = 1 - \frac{2k}{k^2 + 1},$$

we can just calculate the maximum of

$$\frac{q}{4k} \left(1 - \frac{\frac{1}{2} + \varepsilon}{1 - \frac{2k}{k^2+1}}\right).$$

Take $\epsilon = \frac{1}{8}$ and $k = 8$, we have

$$\frac{1}{4k} \left(1 - \frac{\frac{1}{2} + \varepsilon}{1 - \frac{2k}{k^2+1}}\right) \geq \frac{1}{200}.$$

This means that for $\sigma \leq \frac{1}{200} q$ our method works.

In fact, by the above discussion, we have the following result.

Theorem 3. *Let $\mathbf{s} \in \mathbb{Z}_q^n$, given an oracle $\mathcal{O}$ that, given $\mathbf{x} \in \mathbb{Z}_q^n$ as input, outputs the LWE sample $\langle \mathbf{x}, \mathbf{s} \rangle + e$, where $e \in D_{\mathbb{Z},\sigma}$. Then, there is an absolute constant c, if $\sigma \leq cq$, one can solve the LWE over $\mathbb{Z}_q^n$ in polynomial time with high probability.*

Next, we illustrate the capability of our attack method by estimating parameters vulnerable to our attack shown in Tables 1 and Table 2.

In the seminal work of Regev[35] which introduced LWE problem, a public key cryptosystem was also presented. The error distribution χ is taken to be $\bar{\Psi}_{\alpha(n)}$ for $\alpha(n) = 1/(\sqrt{n} \log^2 n)$). The modulus q is chosen to be some prime number between n^2 and $2n^2$. It's not difficult to show that this distribution

Table 2: Maximum of σ vulnerable to our attack ($\epsilon = \frac{1}{\log_2 q}$)

$\log_2 q$	10	11	12	13	14	15	16	17	18	19
$\sigma \leq$	6.5	13.6	28.5	59.2	122	250.3	512	1044.2	2124.7	4315.3

has the shape of a discrete Gaussian with standard deviation αp. Thus, we can replace σ with αp, since σ is the standard deviation of discrete Gaussian D_σ. By taking ε to be $1/n$, we get

$$(1 - \frac{4k}{\sqrt{n}\log^2 n})\left(1 - k\exp^{\frac{1}{2}(1-k^2)}\right) \geq \frac{1}{2} + \frac{1}{n}.$$

Take $k = 10$, it's trivial to find that if $n \geq 40$, the above inequality is satisfied.

6 Conclusion

We construct a probabilistic polynomial time algorithm to recover the key of LWE, which utilizes the list decoding method at the core. Our algorithm is efficient for a large range of parameters as analyzed, which has security implications for some proposed schemes based on LWE.

Acknowledgments

This paper is supported by National 973 Grant 2013CB834205, NSFC Grant 61272035, and the Strategic Priority Research Program of the Chinese Academy of Sciences Grant XDA06010701.

References

[1] S. Agrawal and D. Boneh and X. Boyen. Efficient lattice (H)IBE in the standard model. In *EUROCRYPT*, pp. 553-572,2010.

[2] S. Agrawal and D. Boneh and X. Boyen. Lattice basis delegation in fixed dimension and shorter-ciphertext hierarchical IBE. In *CRYPTO*, pp. 98-115, 2010.

[3] Akavia, A. Solving Hidden Number Problem with One Bit Oracle and Advice. In: Halevi, S. (ed.) CRYPTO 2009. LNCS, vol. 5677, pp. 337-354. Springer, Heidelberg 2009.

[4] A. Akavia, S. Goldwasser, and S. Safra. Proving hard-core predicates using list decoding. In *FOCS 2003*, pp. 146–157, IEEE Computer Society, 2003.

[5] W. Alexi, B. Chor, O. Goldreich, and C. Schnorr. RSA and rabin functions: Certain parts are as hard as the whole. *SIAM Journal on Computing*, 17(2): 194–209, 1988.

[6] M. Ben-Or. Probabilistic algorithms in finite fields. In *FOCS 1981*, 11: 394–398, 1981.

[7] Z. Brakerski and A. Langlois and C. Peikert and O. Regev and D. Stehlé. Classical hardness of learning with errors. In *STOC 2013*, pp. 575-584, 2013.

[8] M. Blum and S. Micali. How to generate cryptographically strong sequences of pseudorandom bits. *SIAM Journal on Computing*, 13(4): 850–864, 1984.

[9] Z. Brakerski and C. Gentry and V. Vaikuntanathan. (Leveled) fully homomorphic encryption without bootstrapping. In *ICTS*, pp. 309-325, 2012.

[10] Z. Brakerski and V. Vaikuntanathan. Efficient fully homomorphic encryption from (standard) LWE. In *FOCS*, pp. 97-106, 2011.

[11] D. Boneh and I. E. Shparlinski. On the unpredictability of bits of the elliptic curve Diffie-Hellman scheme. In *CRYPTO 2001*, LNCS vol. 2139, pp. 201–212, Springer, 2011.

[12] D. Boneh, R. Venkatesan. Hardness of computing the most significant bits of secret keys in Diffie–Hellman and related schemes. In *Crypto '96*, LNCS vol. 1109, pp. 129–142, 1996.

[13] D. Cash and D. Hofheinz and E. Kiltz and C. Peikert. Bonsai trees, or how to delegate a lattice basis. In *EUROCRYPT*, pp. 523-552, 2010.

[14] W. Diffie and M. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, IT–22(6): 644–654, 1976.

[15] A. Duc and D. Jetchev. Hardness of computing individual bits for one-way functions on elliptic curves. In *CRYPTO 2012*, LNCS vol. 7417, pp. 832–849, 2012.

[16] T. ElGamal. A public-key cryptosystem and a signature scheme based on discrete logarithms. *IEEE Transactions on Information Theory*, IT–31(4): 469–472, 1985.

[17] N. Fazio, R. Gennaro, I. M. Perera, and W. E. Skeith III. Hard-core predicates for a Diffie-Hellman problem over finite fields. In *CRYPTO 2013*, LNCS vol. 8043, pp. 148–165, 2013.

[18] C. Gentry and C. Peikert and V. Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In *STOC*, pp. 197-206, 2008.

[19] S. D. Galbraith and B. Shani. The multivariate hidden number problem. In *ICITS 2015*, LNCS vol. 9063, pp. 250–268, 2015.

[20] J. von zur Gathen and J. Gerhard. Modern Computer Algebra. *Cambridge University Press*, 1999.

[21] O. Goldreich and L. A. Levin. A hard-core predicate for all one-way functions. *STOC 1989*, pp. 25–32, ACM press, 1989.

[22] S. Goldwasser and S. Micali. Probabilistic encryption. *JCSS*, 28(2):270–299, 1984.

[23] J. Håstad and M. Näslund. The security of individual RSA bits. *FOCS*, pp. 510–521, 1998.

[24] K. Laine and K. Lauter. Key recovery for LWE in polynomial time. Cryptology ePrint Archive, Report 2015/176 (2015), http://eprint.iacr.org/2015/176.

[25] A. Lenstra, H. Lenstra, L. Lovász. Factoring polynomials with rational coefficients. *Math. Ann.*, 261(4):515-534,1982.

[26] R. Lindner and C. Peikert. Better key sizes (and attacks) for LWE-based encryption. In *CT-RSA*, pp. 319-339,2011.

[27] R. Lidl and H. Niederreiter. Finite Fields. *Addison-Wesley*, 1983.

[28] V. Lyubashevsky and C. Peikert and O. Regev. On Ideal Lattices and Learning with Errors over Rings. *J. ACM*, 60(6):1-35, 2013. Preliminay version in EUROCRYPT 2010.

[29] D. Micciancio and C. Peikert. Trapdoors for lattices: Simpler, tighter, faster, smaller. In *EUROCRYPT*, pp. 700-718, 2012.

[30] P. Morillo and C. Ràfols. The security of all bits using list decoding. In *PKC 2009*, LNCS vol. 5443, pp. 15–33, Springer, 2009.

[31] D. Micciancio and O. Regev. Lattice-based cryptography. In *Post-quantum cryptography*, pp. 147-191, 2009.

[32] M. Näslund. All bits in $ax + b \mod p$ are hard. In *CRYPTO '96*, pp. 114–128, 1996.

[33] C. Peikert. Public-key cryptosystems from the worst-case shortest vector problem. In *STOC 2009*, pp. 333-342, 2009.

[34] C. Peikert and B. Waters. Lossy trapdoor functions and their applications. In *STOC 2008*, pp. 187-196, 2009.

[35] O. Regev. On lattices, learning with errors, random linear codes, and cryptography. *J. ACM*, 56(6):1-40,2009. Preliminary version in STOC 2005.

[36] I. E. Shparlinski. Security of polynomial transformations of the Diffie–Hellman key. *Finite Fields and Their Applications*, vol. 10, Issue 1, pp. 123–131, Jan 2014.

[37] A. Slinko. A generalization of Komlós's theorem on random matrices. *New Zealand J. Math.* 30 (1): 81–86, 2001.

[38] E. R. Verheul. Certificates of recoverability with scalable recovery agent security. In *PKC 2000*, LNCS vol. 1751, pp. 258–275, 2000.

[39] M. Wang, T. Zhan, and H. Zhang. Bits Security of the CDH Problems over Finite Fields. Cryptology ePrint Archive, Report 2014/685 (2014), http://eprint.iacr.org/2014/685.

[40] F. Zhang. Bit Security of the Hyperelliptic Curves Diffie-Hellman Problem. Cryptology ePrint Archive, Report 2015/614 (2015), http://eprint.iacr.org/2015/614.