Solving LWE problem with bounded errors in polynomial time

Jintai Ding[1,2]

Southern Chinese University of Technology, 1
University of Cincinnati, 2
ding@math.uc.edu

Abstract. In this paper, we present a new algorithm, such that, for the learning with errors (LWE) problems, if the errors are bounded – the errors do not span the whole prime finite field F_q but a fixed known subset of size D ($D < q$), which we call the learning with bounded errors (LWBE) problems, we can solve it with complexity $O(n^D)$.

Keywords: LWE, Lattice, bounded errors, multivariate polynomials, linerization

1 Introduction

Recently, the Learning with Errors (LWE) problem, introduced by Regev in 2005 [4], has attracted a lot of attentions in theory and applications due to its usage in cryptographic constructions with some good provable secure properties. The main claim is that it is hard as worst-case lattice problems and hence the related cryptographic constructions.

LWE problem can be described as follows.

First, we have a parameter n, a prime modulus q , and an "error" probability distribution κ on the finite field F_q with q elements.

Let $\Pi_{S,\kappa}$ on F_q be the probability distribution obtained by selecting an element A in F_q^n randomly and uniformly, choosing $e \in F_q$ according to κ, and outputting $(A, < A, S > +e)$, where + is the addition that is performed in F_q.

An algorithm that solves LWE with modulus q and error distribution κ, if, for any S in F_q^n , with an arbitrary number of independent samples from $\Pi_{S,\kappa}$, it outputs S (with high probability).

In the case q = 2, this problem corresponds to the learning parity with noise (LPN) problem.

There are several ways to solve this family of problems. One naive way to solve LWE is through a maximum likelihood algorithm by directly solving about O(n) equations. This leads to an algorithm that uses only O(n) samples, and runs in time $2^O(nlogn)$. There are other similar naive algorithms with similar complexity. A more sophisticated algorithm is developed by Blum, Kalai, and

Wasserman [2] , and it requires $2^{O(n)}$ samples and time. This algorithm is based on the method to find a special small set of equations among $2^{O(n)}$ equations to solve the problem.

According to the recent survey of Regev[5], the Blum et al. algorithm is the best known algorithm for the LWE problem, which is related to the fact that the best known algorithms for lattice problems require $2^{O(n)}$ time.

On the theory side, there are many arguments that the LWE problem is very hard due to the complexity of current algorithms and due to the connection of LWE problems with various known hard problems such as the LPN problem, and the worst-case lattice problems including GAPSVP (the decision version of the shortest vector problem) and SIVP (the shortest independent vectors problem). It is even considered to be a hard problem for quantum computers.

In this paper, we present a new algorithm to solve a subclass of the LWE problems, which, we call, the learning with bounded errors (LWBE) problems, namely the errors from the queries do not span the whole finite field but a fixed known subset of size D ($D < q$). We show that we can solved this problem with complexity $O(n^D)$ and $O(n^D)$ queries.

This paper is organized as follows. Section 2 presents the algorithm and explains how it works with an example. Section 3 presents the basic analysis of the algorithm and the complexity. The last section is devoted to conclusion and discussions.

2 The new algorithm

Let us first define LWBE problem.

2.1 The LWBE

LWBE problem is given as follows.

There are a parameter n, a prime modulus q, and a bounded "error" probability distribution κ on the finite field F_q with q elements such that there are only D (and $D < q$) elements whose distribution probability from κ is not zero while the rest are all zero.

Let $ES = \{e_1, .., e_D\}$ be the set of elements whose probability in the distribution κ is not zero and we call this set the error set. This set could include the zero element in F_q and not necessarily.

Let $\Pi_{S,\kappa}$ on F_q be the probability distribution obtained by selecting an element A in F_q^n randomly and uniformly, choosing $e \in F_q$ according to κ, and outputting $(A, < A, S > +e)$, where additions are performed in F_q.

An algorithm that solves LWBE with modulus q and error distribution κ, if, for any S in F_q^n , with an arbitrary number of independent samples from $\Pi_{S,\kappa}$, it outputs S (with high probability).

Surely we first conclude this problem excludes the case that $q = 2$, since the error can only be 1.

The main motivation to consider this problem comes from the consideration that often in the lattice related problems, the short vector (or the 'error') are often select mainly from the small set $\{1, -1\}$. Another motivation come from the consideration that some of the distribution κ in LWE can be approximated well by bounded distributions.

2.2 The new algorithm

Let

$$S = \begin{pmatrix} x_1 \\ x_2 \\ . \\ . \\ . \\ x_n \end{pmatrix}.$$

Let

$$Q = \binom{D}{n+D} = \frac{(n+D)!}{D!(n-1)!},$$

which is the number of monomial (including 1) in the polynomial ring $F_q[x_1, ..., x_n]$ when D is less than q. Therefore the number of monomials (excluding 1) is exactly $Q - 1$.

For a fixed D, clearly Q is of the class $O(n^D)$.

1. **Step 1. Queries**
   We will make $Q' = Q + O(n)$ queries.
   For the i-th query, we shall derive a linear equation that

   $$\sum a_{i,j} x_j = b_i,$$

   where b_i carries the errors. Therefore it is only probabilistically true.
   For each such linear equation, we will produce the degree D equation:

   $$\prod_{k=1}^{D} (\sum a_{i,j} x_j - b_i + e_k) = 0.$$

   We collect those degree D equations to form a new set we call C.

   Note here that D needs to be less than q, otherwise the equation above will be totally trivial, namely the so-called field equations:

   $$x_i^q = x_i.$$

2. **Step 2. Linearization**
   We linearize the set of equations C such that we assign each monomial of $x_1, .., x_n$ (not including 1) a new variable y_i and the number of monomial is exactly $Q-1$ and we assign x_i to be $y_{Q-n+i-1}$.
   Then this linearization will produce a new set of linear equations in the form of

$$L \times Y = B,$$

   where

$$Y = \begin{pmatrix} y_1 \\ y_2 \\ . \\ . \\ . \\ y_{Q-1} \end{pmatrix}$$

   and L is a $Q' \times (Q-1)$ matrix and B a constant vector
3. **Step 3. Solving** the linear equation

$$LY = B$$

   and the output $(y_{Q-n-1}, ..., y_{Q-1})$ gives the solution S and we end the algorithm.

   Note here that we have more rows than columns in L.

   If we can not find the solution (there are too many depend equations), we make another R (of size $O(n)$) queries to derive a new set of linear equations

$$\sum a'_{i,j} x_j = b'_i,$$

   and produce another R degree D equation:

$$\prod_{k=1}^{D} (\sum a'_{i,j} x_j - b_i - e_k) = 0.$$

   Then we amend these equation to C and then go to Step 2 again.

The reason that Step 2 works is very obvious since we know that one of the linear factors of the polynomial $\prod_{k=1}^{D}(a_{i,j}x_j - b_i + e_k)$ must be zero since it covers all possible errors.

The key point is that the degree D equations in C are precise equations and therefore 100 percent correct unlike the linear equations. This fundamental idea behind this method is the same as that is used in [1], namely to use interpolation formula to even out the noise to derive a set of precise equations

3 A toy example

We will do a example over $GF(5)$.

Let n=2.

Let us assume that our error set is $ES = (0, 1)$ and in this case $D = 2$. This means the error e can only be 1 (or 0 – no error).

We also have

$$Q = C(\binom{D+n}{D}) = C(\binom{2}{2+2}) = 6.$$

In this case, let us assume that we make 6 queries and the query vectors are randomly selected as

$$\begin{pmatrix} (1,1) \\ (3,2) \\ (-1,3) \\ (1,-1) \\ (2-1) \\ (3,-1) \end{pmatrix}$$

Then query results are given as

$$W = \begin{pmatrix} 1 \\ 2 \\ 2 \\ 0 \\ 1 \\ 3 \end{pmatrix}.$$

The corresponding probabilistic linear equations can be written as the set:

$$\begin{pmatrix} (x_1 + x_2 - 1) = 0 \\ (3x_1 + 2x_2 - 2) = 0 \\ (-x_1 + 3x_2 - 2) = 0 \\ (x_1 - x_2) = 0 \\ (2x_1 - x_2 - 1) = 0 \\ (3x_1 - x_2 - 3) = 0, \end{pmatrix}$$

which are the probabilistically true.

From this, because the error set is $\{0, 1\}$, we can derive the corresponding quadratic (d=2) equations as:

$$\begin{pmatrix} (x_1 + x_2 - 1)(x_1 + x_2) = 0 \\ (3x_1 + 2x_2 - 2)(3x_1 + 2x_2 - 1) = 0 \\ (-x_1 + 3x_2 - 2)(-x_1 + 2x_2 - 1) = 0 \\ (x_1 - x_2)(x_1 - x_2 + 1) = 0 \\ (2x_1 - x_2 - 1)(2x_1 - x_2) = 0 \\ (3x_1 - x_2 - 3)(3x_1 - x_2 - 2) = 0, \end{pmatrix}$$

which are 100 percent true.
Now we assign the linearization variables as:

$$Y = \begin{pmatrix} y_1 \\ y_2 \\ \cdot \\ \cdot \\ \cdot \\ y_5 \end{pmatrix} = \begin{pmatrix} x_1^2 \\ x_1 * x_2 \\ x_2^2 \\ x_1 \\ x_2 \end{pmatrix}$$

Then we derive the linear equation:

$$L \times Y = B$$

$$\begin{pmatrix} 1\ 2\ 1\ 4\ 4 \\ 4\ 2\ 4\ 1\ 4 \\ 1\ 0\ 1\ 3\ 3 \\ 1\ 3\ 1\ 1\ 4 \\ 4\ 1\ 1\ 3\ 1 \\ 4\ 4\ 1\ 0\ 0 \end{pmatrix} \times \begin{pmatrix} y_1 \\ y_2 \\ \cdot \\ \cdot \\ \cdot \\ y_5 \end{pmatrix} = \begin{pmatrix} 0 \\ 2 \\ 2 \\ 0 \\ 0 \\ 1 \end{pmatrix}.$$

This gives us the solution that:

$$\begin{pmatrix} y_1 \\ y_2 \\ \cdot \\ \cdot \\ \cdot \\ y_5 \end{pmatrix} = \begin{pmatrix} 4 \\ 1 \\ 4 \\ 2 \\ 3 \end{pmatrix}.$$

From this we derive that

$$\begin{pmatrix} y_4 \\ y_5 \end{pmatrix} = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 2 \\ 3 \end{pmatrix}.$$

Therefore

$$S = \begin{pmatrix} 2 \\ 3 \end{pmatrix}$$

Therefore the correct query result should be

$$\bar{W} = \begin{pmatrix} 0 \\ 2 \\ 2 \\ -1 \\ 1 \\ 3 \end{pmatrix}.$$

The error vector then is given as

$$\bar{E} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{pmatrix},$$

namely only the first and the third queries carry errors and the rest are correct.

4 Analysis of the Algorithm and Complexity

One can see easily that the success of the algorithm depends on if we can solve the linear equation:

$$L \times Y = B.$$

Since the query vectors A_i are randomly and uniformly chosen, it is not unreasonable to assume that coefficients of the matrix L are somewhat randomly and uniformly chosen, in this case, it is not at all difficult to deduce that the random matrix L has a very good probability (roughly $1 - 1/q$)) to be of rank $Q - 1$ and therefore we can derive a solution. In all the extensive experiments (thousands and $q > 3$), we have never failed in the first round of our algorithm. Therefore the conclusion is that algorithm works nearly 100 percent.

In the case of toy example in section above, if we select from the 6 queries we have 5 queries, which is minimum we need, we can see that among all 6 choices, all but one also are sufficient to solve the problem. The only one that does not work is the case we choose the queries 1,2,4,5,6. This confirms our argument above.

Now let us look at the complexity. It is clear that the matrix size of L if roughly

$$(N^D/D!)^2$$

and therefore the complexity of solving $LY = B$ is roughly

$$N^{3D}/(6 \times D!).$$

Therefore, we conclude that for any fixed D, we have polynomial time solver in terms of n.

On the other hand, surely the biggest memory requirement is to store the matrix L, which is of the size

$$(N^D/D!)^2,$$

and could be a serious problem if D and n is large. However in this case, we can use some of the polynomial solving algorithms such as MXL algorithms [3] to make fewer queries but using more time to solve it, which we are now working on.

Another remark we have is that our algorithm does not really depend on the distribution of the errors on the error set. Similarly we can see easily if we have a distribution κ that is not bounded but some of the subset's errors has **extremely small** probability, then we surely can assume those guys to have zero probability and apply our algorithm. This, in some way, is to approximate a distribution with a bounded distribution. This will definitely open ways for applying our algorithm for new attacks and LWE related cryptosystems.

5 Conclusion

We present a new algorithm to solve the learning with bounded errors (LWBE) problems, whose errors are bounded – the errors do not span the whose finite field but a fixed know subset of size D, with complexity $O(n^D)$.

This algorithm, we believe, present a new direction to look at the security of the cryptographic algorithms that are related to the LWE problem. If in terms of their design, we can reduce them to a LWBE type of problem, it is clear that we can break them in polynomial time. We are now looking at various LWE related problems and cryptosystems and we believe that our algorithm can be used to really enhance exiting attacks.

6 Acknowledgment

The idea of this paper was first discovered by the author in attacking certain family of lattice-based cryptosystems in 2006-2007 when the author was a Humboldt fellow in TU Darmstdat. I would like to thank R. Linderner R. Weimann, A. May for stimulating discussions. I would also like to thank D. Cabarcas for performing experiments.

This work is also very much stimulated by the talk of O. Regev at PQC 2010 in Darmstadt.

This work is partially supported by NSF China and the Taft research center at the University of Cincinnati.

References

1. Sanjeev Arora, Rong Ge, Learning Parities with Structured Noise, TR10-066, April 2010
2. A. Blum, A. Kalai, and H. Wasserman. Noise-tolerant learning, the parity problem, and the statistical query model. Journal of the ACM, 50(4):506519, 2003.
3. Johannes Buchmann, Daniel Cabarcas, Jintai Ding, Mohamed Saied Emam Mohamed: Flexible Partial Enlargement to Accelerate Grbner Basis Computation over F2. AFRICACRYPT 2010: 69-81
4. O. Regev. On lattices, learning with errors, random linear codes, and cryptography. Journal of the ACM, 56(6):34, 2009. Preliminary version in STOC05.
5. Oded Regev, The Learning with Errors Problem (Invited Survey), CCC, pp.191-204, 2010 25th Annual IEEE Conference on Computational Complexity, 2010