Learning with errors

Ilmari Vahteristo (1107891)

June 11, 2023

Contents

1 Introduction

The goal of this project is to get familiar with the Learning With Errors (LWE) cryptosystem. LWE is the basis for many of the most prominent post-quantum encryption algorithms.
Conceptually, the logic behind the secrecy of LWE is based on representing a secret as a set of linear equations containing noise. This problem can also be viewed as a lattice problem (namely shortest vector problem), which is known to be NP-hard and there is no known polynomial time quantum algorithm either.
In this report, I will introduce the basic concepts and math behind LWE, and describe a simple public key system utilizing the LWE problem.

1.1 History

Cryptographic applications are usually based on hard-to-solve problems from mathematics, which have already been studied for years. The progression of LWE is no different. In 1996 Miklós Ajtai proposed cryptographic systems, that could be based on the hardness of the Unique Shortest Vector Problem (u-SVP). Together with Cynthia Dwork, they later showed that the problem is as hard on average than in the worst case. Later, in 2005 Oded Regev improved the secureness, and efficiency of a lattice-based cryptosystem, by reducing SVP to a generalized version of "parity learning with errors", which allowed for more memory-efficient encryption. He called it Learning With Errors (LWE). Later, this formulation has been made even more efficient and secure by multiple authors. In a research competition for post-quantum cryptographic algorithm by National Institute of Standards and Technology (NIST), three out of four selected final proposals were based on LWE.

2 Learning With Errors problem

Learning with errors relies on the hardness of solving this equation:

$$A\boldsymbol{x} = \boldsymbol{y} \, mod \, q \, + \boldsymbol{e}, \tag{1}$$

where $A \in Z_q^{n \times m}$, $\boldsymbol{x} \in Z_q^m$, $\boldsymbol{y} \in Z^n$, and $\boldsymbol{e} \in \chi^n$ where χ is an error distribution. If $\chi = \{0\}$, then the measures have no error and can be solved in polynomial time with Gaussian elimination.
By setting q, n and χ appropriately guarantees security and correctness, assuming the hardness of LWE, i.e. there is no algorithm for solving SVP in probabilistic polynomial time.

2.1 Public key cryptography using LWE

Public Key Cryptography (PKC) is asymmetric cryptography. In PKC, keys are generated as pairs (p, s), where p is the public part of the key, and s is the secret

(private) part of the key. The keys are related, and if p_i is used to encrypt a message, then ONLY the related key s_i can decrypt the message.
Here we will consider an implementation of a PKC system, that is based on the LWE problem. This scheme is one of the simplest, but definitely not among the best since the *"rate"* (size of encryption/size of message) is $O(n)$, and if this was actually used, the size of the encryption would be > 800 times the size of the original message. Furthermore, the key size is quadratic to the security parameter, which is not good. There are many ways to reduce the rate, and even achieve a rate close to one (ideal), but this report does not consider them.

Let's start by describing the LWE PKC scheme for sending messages between $T(p_t, s_t)$ (transmitter) and $R(p_r, s_r)$ (receiver). The message space $M = \{0,1\}^k$, can represent any sequence of 1's and 0's. So for T to securely send a message $m \in M$, and for R to be able to decrypt it, we need $c = enc(p, m)$ and $m' = dec(s, c)$. We also need the security parameter n, error distribution χ, and the modulus q.
A key (p, s) is generated with the following steps

$$\vec{s} \leftarrow \chi^n,\ A \leftarrow Z_q^{n\times n},\ \vec{e} \leftarrow \chi^n,\ \vec{y} = \vec{s}A + \vec{e},\ (p,\vec{s}) = ((A,\vec{y}),\vec{s}) \tag{2}$$

Define $c = enc(p = (A,\vec{y}), \vec{m})$, where the number of bits in $\vec{m}$ is k

$$R, X \leftarrow \chi^{k\times n},\ \vec{x'} \leftarrow \chi^k,\ W = RA^T + X,\ \vec{u} = Ry + X' + \vec{m}\lfloor q/2 \rfloor mod\ q,\ c = (W,\vec{u}) \tag{3}$$

Define $\vec{m'} = dec(\vec{s}, c = (W,\vec{u}))$ as follows

$$\vec{v} = \vec{u} - \vec{s}W^T\ mod\ q,\ \ \vec{m'} = 0 \text{ if } |v_i| < q/4 \text{ for } v_i \text{ in } \vec{v} \text{ else } 1 \tag{4}$$

We now have a cryptosystem. The selection of q, χ and n is still an active research topic, but in my implementation as part of this project, I use a discrete, bounded Gaussian distribution for χ. Typically q is an exponential or a polynomial of n, and I use $q = n^2$. The specifics depend heavily on the implementation details.
Some bounds on the amount of error are required to guarantee correctness and this implementation is correct, as long as $\chi_{min} > -\sqrt{\frac{q}{4n}}$ and $\chi_{max} < \sqrt{\frac{q}{4n}}$.

3 Implementation

As part of this project, the described cryptosystem is implemented in Python. Also, some tests were created to empirically verify the correctness of the system. There is also a naive method for trying to break the encryption, by calculating the least squares solution to the LWE equation 1, and if there is any noise, it is as bad as random.

4 References

1. Lyubashevsky, Vadim, Chris Peikert, and Oded Regev. "On ideal lattices and learning with errors over rings." Journal of the ACM (JACM) 60.6 (2013): 1-35.

2. Regev, Oded. "The learning with errors problem." Invited survey in CCC 7.30 (2010): 11.

3. https://people.csail.mit.edu/vinodv/CS294/lecture1.pdf