Lattice Based Cryptography

Justin French

March 17, 2016

1 Overview

Lattice based cryptography is, surprisingly, cryptography based on lattices. Just as RSA and ElGamal are based on hard problems in number theory, latticed based cryptosystems such as GGH or NTRU are based on hard problems relating to lattices, which will be discussed in greater detail later. Some lattice problems have been actually proven to be hard problems, an advantage over number theoretic problems like factorization that are believed to be hard without proof. Furthermore, number theoretic problems such as factorization have been shown not to be hard problems when using quantum computers, but lattice based problems are believed to be hard even when using quantum computers. Despite these theoretical advantages, lattice based cryptosystems are to inefficient for actual use, and given that quantum computers do not yet exist there is not currently much need for them.

2 Lattices

It would be difficult to present information on latticed based cryptography without discussing what a lattice is. Formally, given n linearly independent vectors $b_1, b_2, ...b_n \in \mathbb{R}^n$, the lattice generated by them is the set of vectors

$$\mathcal{L}(b_1, b_2, ...b_n) = \{\sum_{i=1}^{n} x_i b_i : x_i \in \mathbb{Z}_i\}$$

Figure 1: A square lattice

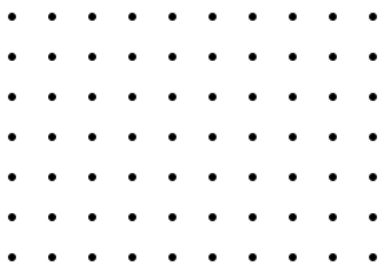
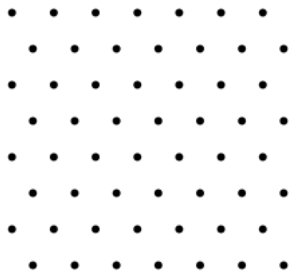


Figure 2: A equilateral triangular lattice



For example, the grid of all two dimensional integer points is generated by the vectors $(0, 1)$ and $(1, 0)$, while the grid of equilateral triangles covering the plane is generated by the vectors $(0, 1)$ and $(\frac{1}{2}, \frac{\sqrt{3}}{2})$. In the case that the vectors $b_1, b_2, ...b_n$ generates a lattice L, they are called a basis for the L. One lattice can have multiple bases, and the fact that some lattice problems are easier to solve with different bases is a key insight in lattice based cryptography. Bases are usually represented with a matrix; for example, if $(0, 0, 1)$, $(0, 3, 0)$, and $(2, 0, 0)$ are a basis for a lattice the basis may be written as

$$\begin{bmatrix} 0 & 0 & 2 \\ 0 & 3 & 0 \\ 1 & 0 & 0 \end{bmatrix}$$

with each column representing a vector in the basis. If B is a matrix representing the basis of an n-dimensional lattice, each vector of the lattice can be represented as an element of $\mathbb{Z}^n$ times B.

3 Lattice Problems

As mentioned earlier, modern cryptosystems are based on hard problems, and lattice-based cryptosystems are based on hard problems regarding lattices. Some examples of hard lattice problems include:

Closest Vector Problem: Given some lattice L and a vector v not necessarily in the lattice, find the vector in L closest to v. This is the problem used in the GGH Cryptosystem, which will be demonstrated in an example later.

Shortest Vector Problem: Given some lattice L, find the shortest non-zero vector in L. Often, the approximation variant of this problem is used, where instead of finding the shortest vector, one finds a short vector within an 'approximation factor' γ from the shortest vector. γ is a function of the dimension of the lattice. There are polynomial time algorithms to solve this problem, but in this case the approximation factor is exponential; known algorithms with polynomial approximation factor take exponential time.

4 GGH Cryptosystem

The GGH Cryptosystem is an example of a cryptosystem based on a hard lattice problem, specifically the closest vector problem. It takes advantage of the phenomenon that the closest vector problem is easier with certain bases. Unfortunately, this cryptosystem is to inefficient to be used in practice.

Private Key: The private key is a basis B for some lattice L with short, almost orthogonal vectors. The known algorithms for the closest vector problem do not work unless the basis used has sufficiently orthogonal vectors.

Public Key: The public key is a 'bad' basis H for L.

Procedure:

1. Alice uses H to pick some vector $m \in L$ that represents the message. Converting a message into a vector is a non-trivial step, but it is somewhat technical so it will not be discussed here.

2. Alice then picks some small (in magnitude) vector $e \in \mathbb{R}^n$. The ciphertext is $c = m + e$. In practice, any value of e is either small enough that one can easily break the cipher without the private key, or it is so large that even with the private key one cannot decipher the message. Because of this impracticality, GGH is not used except in theory.

3. Bob computes m from c using the good basis B in the private key.

   A common algorithm for the actual decryption is the Babai rounding algorithm. The procedure is as follows:

1. Find B^{-1}. Then compute $z = cB^{-1}$.

2. Round each element of z to the nearest integer. Call this rounded vector r.

3. Calculate rB.

   If the vectors of B are sufficiently orthogonal, this decrypts the message; otherwise, it can return a vector very far off from the correct answer. It is guaranteed to work when the vectors are exactly orthogonal, but when the private key is not perfectly orthogonal there may be messages that cannot be decrypted with this method. This is another reason that GGH is not used in practice.

5 GGH Example

Bob picks his private key to be

$$B = \begin{bmatrix} 5 & 0 \\ 0 & 3 \end{bmatrix}$$

He picks his public key to be

$$H = \begin{bmatrix} 35 & 20 \\ 15 & 9 \end{bmatrix}$$

It is not difficult to show that these are bases for the same lattice, but the proof will be omitted. Let $m = (25, 12)$ and $e = (1, -2)$. Then $c = m + e = (26, 11)$. We will now show the decryption of this ciphertext, using the Babai

rounding algorithm, with the private key basis and with the public key basis. Using the good matrix,

$$B^{-1} = \begin{bmatrix} \frac{1}{5} & 0 \\ 0 & \frac{1}{3} \end{bmatrix}$$
$$(26, 11) \times B^{-1} = (5.2, 3.66...)$$

Rounded, this becomes $(5, 4)$ and

$$(5, 4) \times B = (25, 12)$$

$(25, 12)$ is the original plain text, so the decryption works. Using the bad matrix,

$$H^{-1} = \begin{bmatrix} \frac{3}{5} & -\frac{4}{5} \\ -1 & \frac{7}{3} \end{bmatrix}$$
$$(26, 11) \times H^{-1} = (4.6, -9)$$

Rounded, this becomes $(5, -9)$ and

$$(5, -9) \times H = (40, 19)$$

$(40, 19)$ is nowhere near the original plain text, so the decryption does not work.

6 Resources

`http://web.eecs.umich.edu/~cpeikert/pubs/lattice-survey.pdf`

`https://www.cims.nyu.edu/~regev/papers/pqc.pdf`

`http://eprint.iacr.org/2014/070.pdf`

`https://eprint.iacr.org/2015/938.pdf`

`http://www.cs.bris.ac.uk/pgrad/csjhvdp/files/ThesisJvdPol.pdf`