Analysis of Knapsack Cryptanalysis via Lattice- A Survey

Jinsu Kim †

†*Faculty of mathematics, Naval Academy, Gyungnam, 51704 Republic of Korea*

Summary

Historically, lattices and lattice reduction, the LLL algorithm, have played important roles in mathematics and cryptography as a problem-solving tool. In this paper, among applications of LLL algorithm, we analyze several algorithms in order to attack knapsack cryptosystems using LLL. A key part of these attacks is to convert knapsack problems into lattice ones. In this paper, we show how to do these works. We expect that this work gives us insights into converting hard problems to lattice problems in a variety of cryptographic situations.

Key words:

LLL, knapsack, lattice, lattice reduction, cryptanalysis

1. Introduction

Informally, a lattice is a regular arrangement of points in an n-dimensional space. We can define it as an additive discrete subgroup of $\mathbb{R}^n$. Graphically, a lattice is the set of vertices of an n-dimensional grid. All lattice can be represented by a basis which is a set of $n\ (\leq m)$ linearly independent vectors and have infinitely many bases. However, some of them are more useful than others.

The goal of lattice (basis) reduction is given an integer lattice basis as input, in order to find a basis with short, nearly orthogonal vectors. Lattice reduction is not only an important task of lattice research but also a powerful tool for cryptology. Many lattice problems can be solved by lattice reduction directly, and even some NP-hard lattice problems (e.g. SVP, CVP) can be approximately solved by reduction. On the other hand, some cryptanalysis can be reduced to lattice reduction problems. The mathematical point of view, the history of lattice reduction goes back to Minkowski's geometry of numbers and the theory of quadratic forms developed by Lagrange, Gauss, Hermite, Korkine, Zolotare and many others. But the most important modern advance in lattice reduction is made by Lenstra, Lenstra, Lovasz (LLL). In 1982, they found a lattice basis reduction algorithm called LLL that finds a moderately small and orthogonal basis in polynomial time [12]. And this algorithm approximates the shortest vector in a lattice to within some factor. After that time, further refinements of the LLL algorithm were proposed by Schnorr [18], [19]. The relevance of LLL algorithms to cryptography was immediately understood. Typically, in 1982, Shamir [20] found a polynomial-time algorithm breaking the Merkle-Hellman public key cryptosystem [14] based on the subset sum problem that had been basically the unique alternative to RSA at that time. He used Lenstra's integer programming algorithm but, in the same year, Adleman extended Shamir's work by treating the cryptographic problem as a lattice problem. Further improvements of these attacks were obtained by Lagarias and Odlyzko [13], Coster et. Al. [3], and more recently Nguyen and Stern proposed in [16] a natural generalization of the Lagarias-Odlyzko [13] lattices by using orthogonal lattices.

In this paper, we focus on the transformation of some cryptographic problems into lattice reduction problems. The most important part is to construct a suitable lattice which gives a desired solution when a polynomial time lattice reduction, LLL, is applied. Especially, we examine the three attacks arising from knapsack cryptosystem: Shamir's attack, low-density attack, secret key recovery attack using orthogonal lattice.

1.1 Organization

In Section 2, we introduce some definitions and concepts about lattice, knapsack cryptosystem and simultaneous Diophantine approximation. In Section 3, 4, 5, we look into basic idea of the attack methods and how the attacks are converted to lattice problems. In Section 6, we present some concluding remark about a common property of lattices appeared in 3, 4, 5.

2. Preliminaries

2.1 Lattice and Lattice Basis Reduction

We need some description about lattice which is subset of the vector space $\mathbb{R}^n$ and the LLL algorithm. We write all vectors as rows and denote vector by bold face lowercase letter, $\|\boldsymbol{v}\|$ the Euclidean norm induced by inner product $\langle\, ,\, \rangle$.

Definition 2.1. Let $\{\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n\}$ be a linearly independent set of vectors in $\mathbb{R}(n \leq m)$. The lattice L generated by $\{\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n\}$ is the set L: $= \{\sum_{i=1}^{n} l_i \boldsymbol{b}_i : l_i \in \mathbb{Z}\}$ of integer linear combinations of the $\boldsymbol{b}_i$.

The set of vectors $\{\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n\}$ is called a lattice basis. The dimension of lattice L is n. If n = m then L is said to be a

Manuscript received September 5, 2020
Manuscript revised September 20, 2020

DOI: 10.22937/IJCSNS.2020.20.09.22

full rank lattice. We say that L is a sublattice of a lattice in R^m if contains L and if both have the same dimension.

Definition 2.2. A basis matrix B of a lattice L is an n $\times$ m matrix formed by taking the rows to be basis vectors $b_i = (b_{i,1}, \dots, b_{i,m})$.

Thus $B_{i,j}$ be the j-th entry of the row vector $\boldsymbol{b}_i$ and L $= \{xB \,:\, x \in \mathbb{Z}^n\}$. By assumption, the rows of a basis matrix are always linearly independent. The determinant of a lattice L is the volume of the fundamental parallelepiped of any basis. Therefore, it depends on the basis. However, it is well-defined regardless of basis, we can define the determinant of L as following.

Definition 2.3. Let L be a lattice in$\mathbb{R}^m$ of rank n with basis matrix B. The n $\times$ n Gram matrix of B is BB^t. This is a matrix whose (i, j) entry is $\langle b_i,\ b_j\rangle$. Then

$$\det(\mathrm{L}) := \sqrt{BB^t}$$

In particular, if L is a full rank lattice then $\det(L) = |\det(B)|$.

Definition 2.4. Let L $\subset \mathbb{R}^m$ be a lattice of rank n. The successive minima of L are $\lambda_1, \dots, \lambda_n \in \mathbb{R}$ such that, for $1 \le$ i $\le$ n, λ_i is minimal such that there exist i number of linearly independent vectors $\boldsymbol{v}_1, \dots, \boldsymbol{v}_i \in L$ with $\|\boldsymbol{v}_j\| \le \lambda_i$ for 1 $\le$ j $\le$ i.

It follows that $0 < \lambda_1 \le \lambda_2 \le \cdots \le \lambda_n$. In general, there is not a basis consisting of vectors whose lengths are equal to the successive minim. All base of L spans the same R-vector subspace of $\mathbb{R}^m$ which we denote by Sp_L . The dimension of Sp_L over $\mathbb{R}$ is equal to the dimension of L. Define the lattice $\bar{L} := Sp_L \cap \mathbb{Z}^m$. L is a sublattice of $\bar{\mathrm{L}}$. We say that L is a complete lattice, if L $= \bar{L}$. In particular, $\bar{L}$ is a complete lattice.

Definition 2.5. Let L $\in \mathbb{Z}^m$ be a lattice. F $= (Sp_L)^\perp$ be the orthogonal vector subspace with respect to the inner product. We define the orthogonal lattice to be $L^\perp := F \cap \mathbb{Z}^m$. Thus, $L^\perp$ is a complete lattice in $\mathbb{Z}^m$ with dimension m $-$ n if n is the dimension of L.

Theorem 2.6. [16] Let L $\in \mathbb{Z}^m$ be a complete lattice. Then $\det(L^\perp) = \det(\mathrm{L})$ and $\det((L^\perp)^\perp) = det(L^\perp) = \det(\bar{L})$.

As we have mentioned in introduction, computational problems of lattices can be easy if one has a basis that is orthogonal, or "sufficiently close to orthogonal". Therefore, we need an efficient lattice basis reduction algorithm for solving lattice problems. Now, we present the important definition of LLL reduced basis from [12] and some parts of its consequences. Recall that if $\boldsymbol{b_1}, \dots, \boldsymbol{b}_n$ is a set of vectors in Rm then one can define the Gram-Schmidt orthogonalization $\boldsymbol{b_1}^*, \dots, \boldsymbol{b}_n{}^*$. We use the notation $\mu_{i,j} = \langle \boldsymbol{b}_i, \boldsymbol{b}_j{}^*\rangle \,/\, \langle \boldsymbol{b}_j{}^*, \boldsymbol{b}_j{}^*\rangle$.

Definition 2.7. Let $\{\boldsymbol{b_1}, \dots, \boldsymbol{b}_n\}$ be an ordered basis for a lattice L and $\{\boldsymbol{b_1}^*, \dots, \boldsymbol{b}_n{}^*\}$ be its Gram-Schmidt orthogonalization. Let $B_i = \|\boldsymbol{b}_i{}^*\|^2 = \langle \boldsymbol{b}_i{}^*, \boldsymbol{b}_i{}^*\rangle$ for 1 $\le$ i $\le$ n. The basis $\{\boldsymbol{b_1}, \dots, \boldsymbol{b}_n\}$ is called LLL reduced with factor $1/4 < i < 1$ if the following two conditions hold (typically, $\delta = 3/4$).

1. $|\mu_{i,j}| \le 1/2$ for 1 $\le$ j < i $\le$ n
2. $B_i \ge (\delta - \mu_{i,i-1}{}^2) B_{i-1}$

Remark 2.8. The condition 2 means an LLL-reduced basis is close to orthogonal, since a lattice basis is close to orthogonal" if the lengths of the Gram-Schmidt vectors do not decrease too rapidly.

The following Theorem shows that an LLL-reduced lattice basis does have good properties.

Theorem 2.9. Let $\{\boldsymbol{b_1}, \dots, \boldsymbol{b}_n\}$ be an LLL-reduced basis of a lattice L. Then

1. $\|\boldsymbol{b_1}\| \le 2^{(n-1)/2}\lambda_1$
2. $\|\boldsymbol{b_j}\| \le 2^{(n-1)/2}\lambda_i$ for 1 $\le$ j $\le$ i $\le$ n
3. $2^{(1-i)/2}\lambda_i \le \|\boldsymbol{b_i}\| \le 2^{(n-1)/2}\lambda_i$
4. $\det(L) < \prod_{i=1}^{n}\|\boldsymbol{b_i}\| \le 2^{n(n-1)/4}\det(L)$
5. $\|\boldsymbol{b_1}\| \le 2^{(n-1)/4}\det(\mathrm{L})^{1/n}$

Lenstra, Lenstra and Lovasz [12] proves polynomial termination for any lattice in $\mathbb{R}^n$ but only gives a precise complexity for lattices in $\mathbb{Z}^n$.

Theorem 2.10. Let L be a lattice in $\mathbb{Z}^m$ with basis $\{\boldsymbol{b_1}, \dots, \boldsymbol{b}_n\}$ and $\|b_i\|^2$ for 1 $\le$ i $\le$ n where X $\in \mathbb{N}$. Then the LLL algorithm requires $O(n^3 \mathrm{m}\log(\mathrm{X}))$ arithmetic operations on integers of size $O(\mathrm{n}\log(\mathrm{X}))$. Using naive arithmetic gives running time $O(n^5 \mathrm{m}\log(\mathrm{X})^3)$ bit operations.

2.2 Knapsack Cryptosystem

The Merkle-Hellman knapsack cryptosystem [14] was one of the earliest public key cryptosystems invented by Ralph Merkle and Martin Hellman in 1978. Its ideas are elegant, and far simpler and more efficient than RSA. Although the underlying problem is NP-complete by the modular multiplication transformation, it has surprisingly been broken by Shamir [20] because of the special structure of the private key. After that a lot of knapsack-type cryptosystems had been proposed due to their NPcompleteness nature and high speed in encryption and decryption. Unfortunately, most of them are shown vulnerable to various attacks: the low-density attacks ([1], [13], [3]), the simultaneous Diophantine approximation attack [11] and the orthogonal lattice attack [16] and so on. For more details of the rise and fall of knapsack cryptosystems, we refer readers to the survey papers [10], [17].

Among knapsack or knapsack-type cryptosystems, we look into the Merkle-Hellman knapsack cryptosystem [14]. Firstly, the knapsack problem (or subset sum problem) is described as follows.

Given $\mathbf{a} = (a_1, \dots, a_n) \in N^n$ and a target integer s, determine if there exist $x_1, \dots, x_n \in \{0, 1\}$ satisfying, $x_1 a_1 + \cdots + x_n a_n = s$.

The following decisional version of the problem is equivalent to following search version.

Given $\mathbf{a} = (a_1, \dots, a_n) \in N^n$ and a target integer s, find exist $x_1, \dots, x_n \in \{0, 1\}$ satisfying, $x_1 a_1 + \cdots + x_n a_n = s$.

The general knapsack problem is an NP-complete problem, so it is considered very hard. However, some knapsack problems are very easy to solve. Suppose the weights $a_1, \dots, a_n$ are super-increasing, $a_n > a_1 + \cdots + a_{j-1}$ for each $1 < \mathrm{j} \leq \mathrm{n}$. Then we can easily find x_n, since $x_n \Leftrightarrow s > a_1 + \cdots + a_{n-1}$. Having determined x_n, we are reduced to the lower dimensional knapsack problem $x_1 a_1 + \cdots + x_{n-1} a_{n-1} = s - x_n a_n$, so we can recover $x_{n-1}, \dots, x_1$ recursively. Unfortunately, since $a_1, \dots, a_n$ are public, an attacker can decrypt the message easily. But the solution was proposed by Merkle and Hellman in [14]. They devised a method to convert super-increasing sequences into hard knapsacks that look like random. Their knapsack cryptosystem is of the following form.

Key-Gen(1^n): Choose a super-increasing $b_1, \dots, b_n$ and M, W∈ Z with $M > b_1 + \cdots + b_n$ and gcd(M, W) = 1, and a permutation π on the integers $\{1, \dots, n\}$.

Then secret key is $b_{1,\dots,}b_n$, M, W, π and public key is $a_1, \dots, a_n$ with $\mathrm{a}_j \equiv W b_{\pi(j)} \bmod M$

Encryption: For plaintext $\mathrm{x} = \{x_1, \dots, x\} \in \{0,1\}^n$, ciphertext is $\mathrm{s} = x_1 a_1 + \cdots + x_n a_n$.

Decryption: $\mathrm{c} \equiv W^{-1} s \equiv \sum_{j=1}^{n} b_{\pi^{-1}(j)} x_{\pi^{-1}(j)} \bmod M$. The modulus is large, so c exactly equals the sum. Also $b_1, \dots, b_n$ is super-increasing, so one can easily recover the plaintext x.

Typically, the size of each b_i is $\mathrm{n} + \mathrm{i}$ bits, for $1 \leq \mathrm{i} \leq \mathrm{n}$, the size of M is $2\mathrm{n} + 1$ bits. In the original Merkle-Hellman cryptosystem n = 100. A characteristic of knapsack cryptosystem is density. The density of a knapsack is defined to be $\mathrm{d} = \mathrm{n}/\mathrm{N}$ where $\mathrm{N} = \max_i \log a_i$. The density is approximately the information rate. A cryptosystem's density has a great effect on its vulnerability. When the density is small (namely, less than 0.94…), one can solve the knapsack problem directly by using a lattice reduction with high probability. Such attack is called low-density attack. However, this attack is not still effective against high-density knapsacks. Therefore, some knapsack cryptosystems with high density have been proposed [21].

3. Attacks of Knapsack Cryptosystem

3.1 Shamir's attack

At Crypto'82, Adi Shamir [14] gave the first attack on the original knapsack cryptosystem. We now present his idea to compute both M and $\mathrm{U} = W^{-1} \bmod \mathrm{M}$. Without loss of generality, we assume that no permutation is used in Merkle-Hellman knapsack cryptosystem. The starting point is to note that for $1 \leq \mathrm{i} \leq \mathrm{n}1$, there are integers k_i such that

$$a_i U - k_i M = b_i$$

and $0 \leq k_i < a_i$. Hence,

$$0 \leq \frac{U}{M} - \frac{k_i}{a_i} = \frac{b_i}{a_i M}$$

Since b_i's are super increasing, we have $b_i < \frac{M}{2^{n-i}}$ and so $0 \leq \frac{U}{M} - \frac{k_i}{a_i} = \frac{1}{a_i 2^{n-i}}$. In particular,

$$\frac{U}{M} - \frac{k_1}{a_1} = \frac{1}{a_1 2^{n-1}}$$

is very small.

We now observe that to break the Merkle-Hellman knapsack cryptosystem, it is sufficient to find any pair (u, m) of positive integers such that $u a_i \bmod m$ is another super increasing sequence which gives the original plaintext. To see this, we write $u = \lambda U, m = \lambda M + \varepsilon$ where ε is a small error and λ is a scaling factor. Substituting into (1), we obtain

$$u a_i - m k_i = \lambda(U a_i - M k_i) + \varepsilon k_i = \lambda b_i + \varepsilon k_i$$

The right side of (3) is just the super increasing sequence of λb_i that is perturbed by the quantities εk_i, so it will be super increasing whenever ε is small enough.

We don't know U, M, k_i and b_i, but a_i. However, we can see the size of ai's and U are the same as M's and $b_i \leq 2^{n+i}$. Subtracting the case $i = 1$ of equation (2) from the i-th gives

$$\frac{k_1}{a_1} - \frac{k_i}{a_i} = \frac{b_i}{a_i M} - \frac{b_1}{a_1 M} = \frac{a_1 b_i - a_i b_1}{a_1 a_i M}$$

and so, for $2 \leq i \leq n$

$$|a_i k_1 - a_1 k_i| = \frac{|a_1 b_i - a_i b_1|}{M} < \frac{2 M b_i}{M} = 2 b_i$$

In particular, for $2 \leq i \leq 5$,

$$|a_i k_1 - a_1 k_i| < 2b_i \leq 2^{n+6}$$

the size of each of them is 2n bits, so the size of $a_1 k_i$ and $a_i k_1$is 4n bits. But the size of the difference of two such terms to be n + 6 bits, which requires some very special structure. From above observations, Shamir showed how to find the k_i's ($1 \leq i \leq n$) in polynomial time by invoking H.W. Lenstra's theorem that the integer programming problem in a fixed number of variables can be solved in polynomial time [15].

We now present a method using lattices. That is another way to find the integer k_1 when only the integers $a_1, \ldots, a_n$ are given. If we write the equation (4),

$$\left|\frac{a_i}{a_1} - \frac{k_i}{k_1}\right| < \frac{M}{a_1 k_1 2^{n-i-1}}$$

and we see that this problem is precisely simultaneous Diophantine approximation. The simultaneous Diophantine approximation problem states that, given rational numbers $r_1, \ldots, r_n, \varepsilon > 0$ and an integer $Q \geq \varepsilon^{-n}$ find integers $p_1, \ldots, p_n$ and q such that $0 \leq q \leq Q$, and

$$\left|r_i - \frac{p_i}{q}\right| < \frac{\varepsilon}{q}$$

for all $1 \leq i \leq n$. There exists a solution to the simultaneous Diophantine approximation problem if $Q \geq \varepsilon^{-n}$. And a solution can be obtained from lattice basis reduction algorithm (LLL). Let's consider following matrix, A, form a lattice L:

$$A = \begin{pmatrix} \varepsilon/Q & r_1 & r_2 & \cdots & r_n \\ 0 & -1 & 0 & \cdots & 0 \\ 0 & 0 & -1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & -1 \end{pmatrix} = \begin{pmatrix} \boldsymbol{a}_1 \\ \boldsymbol{a}_2 \\ \boldsymbol{a}_3 \\ \vdots \\ \boldsymbol{a}_{n+1} \end{pmatrix}$$

After computing LLL-reduced basis of L, let $\boldsymbol{b}_1$ be the first LLL-reduced vector. Then the $\boldsymbol{b_1}$ can be used to solve the simultaneous Diophantine approximation problem. Since $\boldsymbol{b_1} \in$ L, there exist integers $p_1, \ldots, p_n$ and q such that

$$\begin{aligned} \boldsymbol{b_1} &= \mathrm{q}\boldsymbol{a}_1 + p_1\boldsymbol{a}_2 + \cdots + p_{n-1}\boldsymbol{a}_n + p_n\boldsymbol{a}_{n+1} \\ &= \left(\frac{q\varepsilon}{Q}, qr_1 - p_1, \ldots, qr_n - p_n\right) \end{aligned}$$

All $|qr_i - p_i|$'s is small, since $\boldsymbol{b}_1$ is short. Hence, all $|r_i - p_i/q|$'s ais small. This observation illustrates the relation between the lattice reduction algorithms and the simultaneous Diophantine approximation problem. More precisely, we can find an approximation satisfying some condition as following.

Theorem 3.1. [6] Let $r_1, \ldots, r_n \in \mathbb{Q}$ be given as rational numbers with numerator and denominator bounded in absolute value by X. Let $0 < \varepsilon < 1$, $Q = 2^{n(n+1)/4}\varepsilon^{-n}$. One can compute in polynomial-time integers $q, p_1, \ldots, p_n$ such that $0 < q < 2^{n(n+1)/4}\varepsilon^{-n}$ and $|r_i - p_i/q| < \varepsilon/q$ for all $1 \leq i \leq n$.

Hence, now consider the following basis matrix where $1 \leq h \leq n$, $0 < \lambda < 1$ is a parameter analogous to ε/Q.

$$A = \begin{pmatrix} \lambda & a_2 & a_3 & \cdots & a_h \\ 0 & -a_1 & 0 & \cdots & 0 \\ 0 & 0 & -a_1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & -a_1 \end{pmatrix}$$

This lattice contains the vector $(\lambda k_1, k_1 a_2 - k_2 a_1, k_1 a_3 - k_3 a_1, \ldots, k_1 a_h - k_h a_1)$ which is short by (). After performing lattice basis reduction on A, we obtain a guess for k_1 from the first vector of LLL reduced basis. Informally, if $\left(k_{2/}k_1, k_{3/}k_1, \ldots, k_h/k_1\right)$ is a very good approximation of $\left(a_{2/}a_1, a_{3/}a_1, \ldots, a_h/a_1\right)$ and is uniquely determined by (), then we can get the desired solution vector with suitable parameters λ and h. A formal analysis of this method is given by Lagarias [11]. As with other attacks on knapsack cryptosystems, the results are heuristic in the sense that they are proved by considering a random knapsack instance. Lagarias [11] suggest that one can take $h > \frac{1}{d} + 1$ for almost all random vector $\left(a_{2/}a_1, a_{3/}a_1, \ldots, a_h/a_1\right)$ has a very good simultaneous Diophantine approximation, where d is the density of the instance. In practice, this method works for rather small values of h.

3.2 Low-density attack

Recall that, given a set of positive integers $\{a_1, \ldots, a_n\}$ or a vector $(a_1, \ldots, a_n)$ and positive integer s, we try to determine whether there exists a subset of A with its sum being s, or finding a vector $\boldsymbol{e} = (e_1, \ldots, e_n) \in \{0,1\}^n$ satisfying

$$\sum_{i=1}^{n} a_i e_i = s$$

Brickell [2] and Lagarias and Odlyzko [13] independently proposed algorithms to solve subset sum problems using lattice reductions. But their underlying ideas are similar. Both methods almost always solve the problem in polynomial time if the density of the subset sum problem is less than 0.6463. A observation is that the solution vector $\boldsymbol{e} \in \{0,1\}^n$ is a short vector. So we need to construct a lattice that contains the solution vector $\boldsymbol{e}$. Then we must be able to get the the solution vector after performing lattice reduction algorithm, LLL. How can we make a such lattice from above observation? Lagarias and Odlyzko considered following $(n+1) \times (n+1)$ matrix B:

$$B = \begin{pmatrix} 1 & 0 & \cdots & 0 & -a_1 \\ 0 & 1 & \cdots & 0 & -a_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & -a_n \\ 0 & 0 & \cdots & 0 & s \end{pmatrix} = \begin{pmatrix} \boldsymbol{b}_1 \\ \boldsymbol{b}_2 \\ \vdots \\ \boldsymbol{b}_n \\ \boldsymbol{b}_{n+1} \end{pmatrix}$$

Let L is a lattice generated by the matrix B. Then

$$e_1\boldsymbol{b}_1 + e_2\boldsymbol{b}_2 + \cdots + e_n\boldsymbol{b}_n + \boldsymbol{b}_{n+1} = (e_1, e_2, \ldots, e_n,\ s - \textstyle\sum_{i=1}^{n} a_i e_i\ )$$

$$= (e_1, e_2, \ldots, e_n, 0)\ \in L.$$

It suffices to find $\tilde{\boldsymbol{e}} = (e_1, e_2, \ldots, e_n, 0)\ \in L$. More precisely, let's consider the algorithm they proposed.

Table 1. SV algorithm

Finding a solution of subset sum problem

Input: $(a_1, a_2, \ldots, a_n)\ \in\ \mathbb{N}^n, s\ \in \mathbb{N}$

Output: $(e_1, e_2, \ldots, e_n)\ \in\ \mathbb{N}^n$ such that

$$\sum_{i=1}^{n} a_i e_i = s$$

1. Take the following vectors $\boldsymbol{b}_1, \boldsymbol{b}_2, \ldots, \boldsymbol{b}_{n+1}$ as a basis for an $(n+1)$ dimensional integer lattice L :

2. Find an LLL reduced basis $\boldsymbol{b}*_1, \boldsymbol{b}*_2, \ldots, \boldsymbol{b}*_{n+1}$ of L.

3. Check if any $\boldsymbol{b}^*{}_i = (b^*_{i,1}\ b^*_{i,2}, \ldots, b^*_{i,n+1})$ has all $b^*_{i,j} = 0$ or λ for some fixed λ for $1\ \le j\ \le n$. For any such $\boldsymbol{b}*_i$, check whether $x_j = \lambda^{-1} b^*_{i,j}$ for $1\ \le j\ \le n$ gives a solution.

4. Repeat steps 1- 3 with s replaced by $s' = \sum_{i=1}^{n} a_i - s$. Then stop.

We immediately obtain polynomially bounded running time since SV algorithm is essentially two applications of the LLL algorithm. And if this algorithm produces a solution to (7) we say it succeeds; otherwise, it fails. Therefore, we want to be appeared the vectors $\tilde{\boldsymbol{e}}$ or $\lambda\tilde{\boldsymbol{e}}$ in LLL reduced basis of L. An ideal case is that the vector $\tilde{\boldsymbol{e}}$ is a shortest vector of L generated by B and the first LLL-reduced vector is also a shortest vector of L. In other words, we try to reduce the subset sum problem to the shortest vector problem.

There are two problems on this idea. First, the NP-hardness of SVP(under randomized reductions) means that there is no polynomial-time algorithm that solves SVP. However, this is acceptable up to reasonably high dimensions n. Because it turns out that in practice, one can hope that standard lattice reduction algorithm, LLL behave like SVP-oracles, up to reasonably high dimensions (by experimental result). Second, there may exist many short vectors that are shorter than $\tilde{\boldsymbol{e}}$ according to a_i's. For example, if $a_1 =\ a_2 + 1$, then $(1, -1, 0, \ldots, 0, -1)\ \in L$. Hence, we see that this approach to solve the subset sum problem is very heuristic. Informally, such short vectors (like example) do not arise for the small n, the large M. (Recall the ai 's are randomly chosen from $[1,\ M-1]$). As mentioned earlier, Lagarias and Odlyzko showed that the subset sum problem can be reduced to SVP if the density of the subset sum problem is less than 0.6463.

Theorem 3.2. [13] if the density is bounded by 0.6463... then the lattice oracle (LLL) is guaranteed to find the solution vector with high probability.

The matrix used in this theorem 3.2. is as follows:

$$\begin{pmatrix} 1 & 0 & \cdots & 0 & Na_1 \\ 0 & 1 & \cdots & 0 & Na_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & Na_n \\ 0 & 0 & \cdots & 0 & Ns \end{pmatrix}$$

Here the constant N makes proof easier than Lagarias and Odlyzko's original proof. In fact, above proof is a simple version of the original proof due to [4].

The core part in the proof is counting lattice points in a sphere. Since the number of lattice points in spheres in n-dimensional space gives the result $1.54724\ldots^{-1} \approx 0.6463$. For detail see ([13], Theorem 3.2). Hence if we can reduce the constant $1.54724\ldots$, the condition 0.6463 will increase. We can see this can be done by reducing $\|\boldsymbol{e}\|$. Then how can we do this by modifying the lattice (7)?

Coster, Joux, LaMacchia, Odlyzko, Schnorr, and Stern improved the bound to 0.9408 [3] by using following matrix:

$$\begin{pmatrix} 1 & 0 & \cdots & 0 & Na_1 \\ 0 & 1 & \cdots & 0 & Na_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & Na_n \\ 1/2 & 1/2 & \cdots & 1/2 & Ns \end{pmatrix}$$

The only different point is the last row vector $(1/2, 1/2, \ldots, 1/2, Ns)$. This leads to following consequence.

Theorem 3.4 [3] Let A be a positive integer, and let $a_1, \ldots, a_n$ be random integers with $0 < a_i \leq A$ for $1 \leq i \leq n$. Let $e = (e_1, \ldots, e_n) \in \{0,1\}^n$ be arbitrary, and let $s = \sum_{i=1}^{n} e_i a_i$. If the density $d < 0.9408 \ldots$, then the subset sum problem defined by $a_1, \ldots, a_n$ and s may "almost always" be solved in polynomial time with a single call to a lattice oracle.
This theorem shows a improvement of the density bound from 0.6463 ... to 0.9408 ... by modifying last row vector in the lattice basis. In fact, this problem is closely connected to lattice covering problems. From analysis of proof, we see that it needs to cover the vertices of the n-cube (representing the possible e solution vectors) within a polynomial number of n-dim spheres of radius $\sqrt{\alpha n}$. In fact, Lagarias and Odlyzko used two n-dim spheres of radius $\sqrt{n/2}$ that are centered at (0, 0,..., 0) and (1, 1, ... , 1) respectively, and Coster et al. used one n-dim sphere of radius $\sqrt{n/4}$ centered at $(1/2, 1/2, \ldots, 1/2)$ to cover all the points. And they showed that the asymptotic bound 0.9408 ... can not be improved in their way. (see for detail [3] proposition 5.1)

3.3 Secret key recovery attack using orthogonal lattice

For any integer lattice L in $\mathbb{Z}^n$, the orthogonal lattice $L^{\perp}$ as the set of integer vectors orthogonal to L, that is, the set of $\boldsymbol{x} \in \mathbb{Z}^n$ such that the dot product $\langle \boldsymbol{x}, \boldsymbol{y} \rangle = 0$ for all $\boldsymbol{y} \in L$. As mentioned earlier, the lattice $L^{\perp}$ has dimension $n - dim(L)$. Thus, if a lattice in $\mathbb{Z}^n$ is low-dimensional, its orthogonal lattice is high-dimensional with a volume at most equal. Hence the successive minima of the orthogonal lattice are likely to be much shorter than the ones of the original lattice. That property of orthogonal lattices has led to effective (though heuristic) lattice-based attacks on various cryptographic schemes. In particular, it was used to find secret key in [22], [23].

The main idea of those attack is to find a LLL reduced basis of $(n-2)$ dimensional orthogonal lattice from secret key $\boldsymbol{b} = (b_1, b_2, \ldots, b_n)$ and public key $\boldsymbol{a} = (a_1, a_2, \ldots, a_n)$. However, we do not know the secret key, so we need following heuristic.

Heuristic 1. Let $\boldsymbol{b}_1, \boldsymbol{b}_2, \ldots, \boldsymbol{b}_{n-1}$ be an LLL reduced basis of $\boldsymbol{a}^{\perp}$. Then the first $n-2$ vectors $\boldsymbol{b}_1, \boldsymbol{b}_2, \ldots, \boldsymbol{b}_{n-2}$ is orthogonal to $\boldsymbol{b}$.

By Heuristic 1, with a computing orthogonal lattice algorithm, we get a 2 -dimensional lattice $(\boldsymbol{b}_1, \boldsymbol{b}_2, \ldots, \boldsymbol{b}_{n-2})^{\perp}$ which contain $\boldsymbol{b}$. Let $\boldsymbol{c}_1, \boldsymbol{c}_2$ be a LLL reduced basis $(\boldsymbol{b}_1, \boldsymbol{b}_2, \ldots, \boldsymbol{b}_{n-2})^{\perp}$. Then $\boldsymbol{b}$ is written,

$$\boldsymbol{b} = x_1 \boldsymbol{c}_1 + x_2 \boldsymbol{c}_2 \text{ where } x_1, x_2 \in \mathbb{Z}$$

Then $\boldsymbol{b}$ is founded by exhaustive search in [22], by Lagrange-Gauss algorithm as shortest vector of the lattice generated by $\boldsymbol{c}_1, \boldsymbol{c}_2$ in [23]. These leads to compute a LLL reduced basis of $L^{\perp}$ from a lattice L. We can compute a LLL-reduced basis of $L^{\perp}$ simply and efficiently due to Nguyen and Stern in [16]. Now we describe the method to compute an LLL-reduced basis of an orthogonal lattice.

Let $\boldsymbol{b}_1, \boldsymbol{b}_2, \ldots, \boldsymbol{b}_{n-1}$ be a basis of L, and $B = (b_{i,j})$ be its corresponding $d \times n$ matrix. Let c be a positive integer constant will be chosen later. Define$\Omega$to be the lattice in $\mathbb{Z}^{n+d}$ generated by the following $n \times (d+n)$ matrix $B^{\perp}$:

$$B^{\perp} = \begin{pmatrix} Nb_{1,1} & Nb_{2,1} & \cdots & Nb_{d,1} & 0 & 0 & \cdots & 0 \\ Nb_{1,2} & Nb_{2,2} & \cdots & Nb_{d,2} & 0 & 0 & \cdots & 0 \\ Nb_{1,3} & Nb_{2,3} & \cdots & Nb_{d,3} & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ Nb_{1,n} & Nb_{2,n} & \cdots & Nb_{d,n} & 1 & 1 & \cdots & 1 \end{pmatrix}$$

Let $P_{\leftarrow}$ and $P_{\rightarrow}$ be the two projections that map any vector of $\mathbb{Z}^{d+n}$ to respectively the vector of $\mathbb{Z}^d$ made of its first d coordinates, and the vector of $\mathbb{Z}^n$ of its last n coordinates, all with respect to the canonical basis. The following algorithm gives an LLL-reduced basis of $L^{\perp}$.

Table 2. Table Label Computing LLL-reduced basis of $L^{\perp}$

Computing LLL-reduced basis of $L^{\perp}$
Input: a basis $(\boldsymbol{b}_1, \boldsymbol{b}_2, \ldots, \boldsymbol{b}_d)$ of a lattice $L \in \mathbb{Z}^n$
Output: $P_{\rightarrow}(\boldsymbol{x}_1), P_{\rightarrow}(\boldsymbol{x}_2), \ldots, P_{\rightarrow}(\boldsymbol{x}_{n-d})$
1. Select a sufficiently large N.
2. Construct the $n \times (n+d)$ integral matrix $B^{\perp}$.
3. Compute an LLL-reduced basis \$$(\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n$ of $B^{\perp}$.

4. Set $P_{\rightarrow}(\boldsymbol{x}_i)$ as the last n coefficients vector for each LLL-reduced vector $\boldsymbol{x}_i$.

Theorem 3.5. For any integer lattice L in $\mathbb{Z}^n$, the algorithm gives a LLL-reduced basis of the orthogonal lattice $L^{\perp}$ correctly in polynomial time.

Proof. Let $\boldsymbol{x}$ be a vector of L and denote.$\boldsymbol{y} = P_{\rightarrow}(\boldsymbol{x})$. Then $P_{\leftarrow}(\boldsymbol{x}) = N(\langle \boldsymbol{y}, \boldsymbol{b}_1 \rangle, \langle \boldsymbol{y}, \boldsymbol{b}_2 \rangle, \ldots, \langle \boldsymbol{y}, \boldsymbol{b}_d \rangle)$. Hence, $\boldsymbol{y} \in L^{\perp}$ if and only if $P_{\leftarrow}(\boldsymbol{x}) = 0$. Furthemore, If $|\boldsymbol{x}| \leq N$, then $P_{\leftarrow}(\boldsymbol{x}) = 0$. Now let $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n$ be an LLL-reduced basis of $B^{\perp}$. Then for sufficiently large N, $\|\boldsymbol{x}_i\| \leq N$ for all i, so $\boldsymbol{x}_i \in L^{\perp}$. Clearly, $P_{\rightarrow}(\boldsymbol{x}_1), P_{\rightarrow}(\boldsymbol{x}_2), \ldots, P_{\rightarrow}(\boldsymbol{x}_n)$ are linearly independent. And these vectors generate $L^{\perp}$ since for all $\boldsymbol{h} \in L^{\perp}$, $(0, \ldots, 0, h_1, \ldots, h_n) \in B^{\perp}$.

4. Concluding Remarks

In this paper, we examined lattices-based cryptanalysis used for attacking knapsack cryptosystems. Designing of proper lattices as examined previous section gives us unexpected attack methods. Additionally, the lattices also can be used to find not only small dependence of numbers, but also vectors and solutions of other cryptographic problems. Indeed, all of those attack techniques are also developed as orthogonal lattice attacks for AGCD problem and dual attacks for LWE one recently. We expect this survey gives an insight for cryptanalysis of lattice crypto world.

Acknowledgments

This work is supported by research fund of 2020 academic research subject from Maritime Institute through Naval Academy of Republic of Korea.

References

[1] E. F. Brickell, “Solving low density knapsacks,” Advances in Cryptology, pp.25-37, 1984.

[2] E. F. Brickell, “Breaking Iterated Knapsacks,” Advances in Cryptology, pp.342-358, 1985.

[3] M. J. Coster, A. Joux, B. A. LaMacchia, A. M. Odlyzko, C. P. Schnorr, and J. Stern, “Improved low-density subset sum algorithms,” Computational Complexity, Vol. 2, pp. 111 – 128, 1992.

[4] A. M. Frieze, “On the Lagarias-Odlyzko algorithm for the subset sum problem,” SIAM J. Computing, Vol. 15, pp.536 – 539, 1986.

[5] M. L. Furst, R. Kannan, “Succinct certificates for almost all subset sum problems,” SIAM J. Computing, Vol. 18, pp.550 – 558, 1989.

[6] S. D. Galbraith, “Mathematics of public key cryptography”, available: http://www.math.auckland.ac.nz/ sgal018/crypto-book/crypto-book.html.

[7] T. Izu, J. Kogure, T. Koshiba, T. Shimoyama, “Low-density attack revisited,” Design Codes and Cryptography, Vol. 43, pp.47 – 59, 2007.

[8] A. Joux, J. Stern, “Lattice reduction: A toolbox for the cryptanalyst,” Journal of Cryptology, pp.161 – 185, 1998.

[9] A. Joux, J. Stern, “The Hardness of the Hidden Subset Sum Problem and Its Cryptographic Implications,” Crypto’99, 1999.

[10] M. K. Lai, “Knapsack cryptosystems: the past and the future,” available: http://www.ics.uci.edu/mingl/ knapsack.html.

[11] J. C. Lagarias, “Knapsack public key systems and diophantine approximation,” crypto ’83, pp.3 – 23, 1984.

[12] A. K. Lenstra, H.W. Lenstra Jr, L. Lovasz, “Factoring polynomials with rational coefficients,” Math Ann., Vol. 261, pp.515 – 534, 1982.

[13] J. C. Lagarias, A. M. Odlyzko, “Solving low-density subset sum problems,” Journal of the Association for Computing Machinery, Vol.32, pp.229 – 246, 1985.

[14] R. C. Merkle, M. E. Hellman, “Hiding information and signatures in trapdoor knapsacks,” IEEE Transactions on Information Theory, Vol.24, pp.525 – 534, 1978.

[15] J.E. Mazo, A.M. Odlyzko, “Lattice points in high-dimensional spheres,” Monatsh Math, Vol. 110, pp.47 – 61, 1990.

[16] P. Nguyen, J. Stern, “Merkle-Hellman revisited: a cryptanalysis of the Qu-Vanstone cryptosystem based on group factorizations,” In Proc. of Crypto’97, Vol. 1294, pp.198 – 212, 1997.

[17] A. M. Odlyzko, “The rise and fall of knapsack cryptosystems,” In Cryptology and Computational Number Theory, Vol. 42, pp.75 – 88, 1990.

[18] C. P. Schnorr, “a hierarchy of polynomial time algorithm for lattice basis reduction algorithm,” Theoretical Computer Science, Vol. 53, pp.201 – 214, 1987.

[19] C. P. Schnorr, “a more efficient algorithm for lattice basis reduction,” Journal of Algorithms, Vol. 9, pp.47 – 62, 1988.

[20] A. Shamir, “A polynomial time algorithm for breaking the basic Merkle-Hellman cryptosystem,” Proceedings of the 23rd Annual Symposium on Foundations of Computer Science, IEEE Computer Society, pp.145 – 152, 1982.

[21] B. Wang, Q. Wu, Y. Hu, “A knapsack-based probabilistic encryption scheme,” Information Sciences, Vol.177, no.19, pp.3981 – 3994, 2007.

[22] P. Nguyen, J. Stern, “Merkle-Hellman Revisited: A Cryptanalysis of the Qu-vanstone Cryptosystem Based on Group Factorizations,” Proc. CRYPTO ’97, pp.198-212, 1997.

[23] P. Nguyen, J. Stern, “Cryptanalysis of a Fast Public-Key Cryptosystem,” Selected Areas in Cryptography 1998, Vol. 1556, pp. 213-218, 1998.

Jinsu Kim received the B.S. M.S. and D.S. degrees in Mathematical Science from Seoul National University in 2008, 2012 and 2018, respectively. He now with naval academy in Republic of Korea.