Low-Density Attack Revisited

Tetsuya Izu† Jun Kogure† Takeshi Koshiba‡ Takeshi Shimoyama†

† *Secure Comuting Laboratory, FUJITSU LABORATORIES Ltd., 4-1-1, Kamikodanaka, Nakahara-ku, Kawasaki 211-8588, Japan.*

‡ *Division of Mathematics, Electronics and Informatics, Graduate School of Science and Engeneering, Saitama University, 255 Shimo-Okubo, Sakura, Saitama 338-8570, Japan.*
Email: koshiba@tcs.ics.saitama-u.ac.jp

Abstract

The low-density attack proposed by Lagarias and Odlyzko is a powerful algorithm against the subset sum problem. The improvement algorithm due to Coster et al. would solve almost all the problems of density $< 0.9408...$ in the asymptotical sense. On the other hand, the subset sum problem itself is known as an NP-hard problem, and a lot of efforts have been paid to establish public-key cryptosystems based on the problem. In these cryptosystems, densities of the subset sum problems should be higher than 0.9408... in order to avoid the low-density attack. For example, the Chor-Rivest cryptosystem adopted subset sum problems with relatively high densities. In this paper, we further improve the low-density attack by incorporating an idea that integral lattice points can be covered with polynomially many spheres of shorter radius and of lower dimension. As a result, the success probability of our attack can be higher than that of Coster et al.'s attack for fixed dimensions. The density bound is also improved for fixed dimensions. Moreover, we numerically show that our improved low-density attack makes the success probability higher in case of low Hamming weight solution, such as the Chor-Rivest cryptosystem, if we assume SVP oracle calls.

Keywords. subset sum problem, knapsack-based cryptosystem, low-density attack, lattice problem, public-key cryptosystem

1 Introduction

For a given set of positive integers $A = \{a_1, \ldots, a_n\}$ ($a_i \neq a_j$) and a given positive integer s, determining whether there exists a subset of A with its sum being s, or finding a vector $e = (e_1, \ldots, e_n) \in \{0,1\}^n$ satisfying $\sum_{i=1}^{n} a_i e_i = s$, is called the *subset sum problem* (or the knapsack problem), and is known as an NP-hard problem in general (see, e.g., [4]). Brickell [1] and Lagarias and Odlyzko (LO algorithm) [6] independently proposed an algorithm to solve subset sum problems, using lattice reductions. Both methods almost always solve the problem in polynomial time if the

density of the subset sum problem is < 0.6463, where the density d is defined by

$$d = n/(\log_2 \max_i a_i).$$

Then Coster, Joux, LaMacchia, Odlyzko, Schnorr, and Stern (CJLOSS algorithm) improved the bound to 0.9408 [2]. Since these algorithms are effective against relatively-low-density subset sum problems, they are sometimes called the "low-density attack". But the problem is still hard in general density case. In these attacks, the subset sum problem is reduced to the *Shortest Vector Problem* (SVP) of a related lattice, and a single SVP oracle call is assumed. While no polynomial-time algorithms are known to solve the SVP precisely, the polynomial-time algorithm by Lenstra, Lenstra and Lovász (LLL algorithm) solves it with good approximation in practice [5]. One can also use the BKZ algorithm [11] (as in [12]), which provides better approximation but may not work in polynomial-time.

In this paper, we improve the success probability and the density bound of the low-density attack by using polynomially many lattice oracle calls. Note that Coster et al. showed that their algorithm is optimal in a sense as $n \to \infty$ (Proposition 5.1 in [2]). Our improvement is a natural extension of CJLOSS algorithm and the asymptotic behavior of our algorithm coincides with that of CJLOSS algorithm. Since we consider an improvement for any fixed n and the optimality of CJLOSS algorithm is obtained in an asymptotic sense, our results do not contradict that of Coster et al.

Because of the NP-hardness of the subset sum problem, many researchers have used it to establish secure public-key cryptosystems. Merkle and Hellman [7] firstly proposed some cryptosystems by using the subset sum problem and they then were attacked by Shamir [13] on the charge of their intrinsic weakness. After that, Brickell [1] and Lagarias and Odlyzko [6] independently proposed the low-density attack and derived that the density of the subset sum problem used in the cryptosystem should be > 0.6463 in order to avoid the attack. Furthermore, Chor and Rivest proposed a cryptosystem that can use subset sum problems with relatively high densities [3]. While the cryptosystem was attacked by an algebraic approach [14], the attack may not be valid in general cases. Moreover, Okamoto, Tanaka and Uchiyama [10] proposed another cryptosystem intended to resist adversaries that can run on quantum computers.

In some cryptosystems such as the Chor-Rivest cryptosystem, the Hamming weight of solutions is bounded by βn for a small constant $\beta \leq 1/2$. We can take $\beta = 1/2$ in general case, whereas we may assume that $\beta \approx 0.1$ in case of the Chor-Rivest cryptosystem with its recommended parameters. In [2], Coster et al. gave a remark that the density bound of the attack can be improved when the solution is known to have small Hamming weight. Through this paper, we refer the algorithm based on their remark as CJLOSS+ algorithm.

As mentioned, we improve CJLOSS+ algorithm and show that our improvement achieves higher success probability and better density bound for any fixed n. To this end, we firstly give a full analysis of CJLOSS+ algorithm, incorporate a further property of high dimensional lattices into the analysis technique, and then analyze our improved algorithm by using the new technique. (Note that Coster et al. [2] did not give detailed analysis of CJLOSS+ algorithm.) Consequently, we obtain that our algorithm can achieve better density bound than CJLOSS+ algorithm for any fixed

n in general subset sum problems. We also obtain that our algorithm can work with high success probability in low Hamming weight case such as the Chor-Rivest cryptosystem.

2 Previous Works: From the Viewpoint of Lattice Covering Problem

In this section, we review the low-density attack by Lagarias-Odlyzko (LO algorithm) and an improvement by Coster et al. (CJLOSS/CJLOSS+ algorithm), from the viewpoint of the lattice covering problem. The success probability of these algorithms is closely related to the radius of n-spheres covering the solution candidates in the n-cube. Specifically, the radius of the spheres, the center points of the spheres and the number of spheres are important parameters for the algorithms.

Let $(e_1, \ldots, e_n) \in \{0,1\}^n$ and β be a rational constant. We denote the set of integer lattice points satisfying $\sum_{i=1}^{n} e_i \leq \beta n$ as M_β. Note that $M_1 = \{0,1\}^n$.

LO algorithm covers lattice points $M_{1/2}$ with a single sphere of radius $r_{\mathrm{LO}} = \sqrt{n/2}$ centered at $(0, \ldots, 0)$, and by the symmetry of the lattice, it covers M_1 with two spheres of radius r_{LO}. CJLOSS algorithm covers M_1 with a single sphere of radius $r_{\mathrm{C}} = \sqrt{n/4}$ centered at $(1/2, \ldots, 1/2)$. Coster et al. remarked the further improvement(CJLOSS+ algorithm) for small β by using a sphere centered at $(\beta, \ldots, \beta)$ with the radius $\sqrt{\beta(1-\beta)n}$. In addition, Coster et al. showed that CJLOSS algorithm is optimal in the following sense:

Proposition 2.1 (Proposition 5.1, [2]) Any sphere of radius $\sqrt{\gamma n}$, $\gamma < 1/4$, in $\mathbb{R}^n$ contains at most $(2-\delta)^n$ points of $\{0,1\}^n$, for $\delta = 2(1 - e^{\gamma - 1/4}) > 0$.

At a glance, Proposition 2.1 seems to claim that it is impossible to cover $M_1 = \{0,1\}^n$ with polynomially many spheres of radius smaller than r_{C}. However, it does not say that covering $M_{1/2}$ with polynomially many spheres of radius $\sqrt{n/4 - o(n)}$ is impossible. In fact, in this paper, we cover $M_{1/2}$ with polynomially many spheres of radius $\sqrt{n/4 - O(1)}$.

Table 1 summarizes attributes of each algorithm. In Table 1, k is a positive integer with $k \leq \beta n$ and $\beta_k = \frac{\beta n}{n-k} > \beta$. Other details on our proposed algorithm will be described later.

Table 1: Attributes of each low-density attack

Algorithm	Center point(s)	Radius	Lattice points	#sphere
LO	$(0, \ldots, 0)$	$\sqrt{n/2}$	$M_{1/2}$	1
	$((0, \ldots, 0), (1, \ldots, 1))$	$(\sqrt{n/2})$	(M_1)	(2)
CJLOSS	$(1/2, \ldots, 1/2)$	$\sqrt{n/4}$	M_1	1
CJLOSS+	$(\beta, \ldots, \beta)$	$\sqrt{\beta(1-\beta)n}$	M_β $(\beta \leq 1/2)$	1
Ours	$(0, \ldots, 0, \beta_k, \ldots, \beta_k), \ldots$	$\sqrt{\beta(1-\beta_k)n}$	M_β $(\beta \leq 1/2)$	$O(n^k)$
	$(\beta_k, \ldots, \beta_k, 1, \ldots, 1)$	for $\beta_k > \beta$		(k: *const.*)

3 Theoretical Results

In this section, we improve the low-density attack by using polynomially many lattice oracle calls. Before describing our algorithm, we analyze the suggested improvement of Coster et al. based on the remark in Section 5 of [2] (CJLOSS+ algorithm). Note that, as mentioned in the previous section, we use spheres of radius $\sqrt{\beta(1-\beta)n - O(1)}$ while CJLOSS+ algorithm uses a sphere of radius $\sqrt{\beta(1-\beta)n}$. This implies that the asymptotical behavior of our algorithm coincides with CJLOSS+ algorithm. However, for any fixed n, our algorithm can achieve better success probability than CJLOSS+ algorithm regarding one lattice oracle call. Numerical comparison will be given in the next section.

3.1 Analysis of CJLOSS+ Algorithm

With regard to CJLOSS+ algorithm, we have the following theorem.

Theorem 3.1 Let $\beta \leq 1/2$ be a positive rational constant, A a positive integer, and $a_1, \ldots, a_n$ random integers with $0 < a_i \leq A$ for $1 \leq i \leq n$. Let $e = (e_1, \ldots, e_n) \in \{0,1\}^n$ satisfy $\sum_{i=1}^n e_i \leq \beta n$ and let $s = \sum_{i=1}^n e_i a_i$. If the density d of $\{a_1, \ldots, a_n\}$ satisfies

$$d < d_0 = ((\log_2 e)\delta_{\beta,0}(u_0))^{-1},$$

then the subset sum problem defined by $a_1, \ldots, a_n$ and s can be almost always solved in polynomial-time with a single call to a lattice oracle.

In the above statement, $\delta_{\beta,0}(u_0)$ is the minimum value of the following function of $u \in \mathbb{R}^+$:

$$\delta_{\beta,0}(u) = \beta(1-\beta)u + \ln \theta(e^{-u}), \quad \theta(z) = 1 + 2\sum_{j=1}^{\infty} z^{j^2}.$$

We denote $(\log_2 e)\delta_{\beta,0}(u_0)$ by c_0. The proof, we will give in the following, is based on the proof in [2]. Their proof uses results by Mazo and Odlyzko [8] as a main technique. Because the centers of the covering spheres are $(0, ..., 0)$ or $(1/2, ..., 1/2)$ in [2], their proof uses a special case of results in [8], while the following proof uses general cases.

Proof. Let $e \neq (0, \ldots, 0)$ be fixed, $s = \sum_{i=1}^n e_i a_i$, and $t = \sum_{i=1}^n a_i$. LO algorithm uses the following vectors $b_1, b_2, \ldots, b_n, b_{n+1}$:

$$\begin{aligned} b_1 &= (1, 0, \ldots, 0, Na_1), \\ b_2 &= (0, 1, \ldots, 0, Na_2), \\ &\vdots \\ b_n &= (0, 0, \ldots, 1, Na_n), \\ b_{n+1} &= (0, 0, \ldots, 0, Ns), \end{aligned}$$

where N is a positive integer larger than $\sqrt{n}/2$. Let L be an $(n+1)$-dimensional lattice spanned by $b_1, \ldots, b_{n+1}$, namely,

$$L = \left\{ \sum_{i=1}^{n+1} z_i b_i \mid z_i \in \mathbb{Z},\ 1 \leq i \leq n+1 \right\}.$$

Then the vector $\hat{e} = (e_1, \ldots, e_n, 0)$ is contained in L.

CJLOSS+ algorithm uses

$$b'_{n+1} = (\beta, \ldots, \beta, Ns)$$

instead of b_{n+1}. Let L' be an $(n+1)$-dimensional lattice spanned by $b_1, \ldots, b_n, b'_{n+1}$. Then the vector $\hat{e}$ is not contained in L'; but instead

$$\hat{e}' = (e'_1, \ldots, e'_n, 0) = (e_1 - \beta, \ldots, e_n - \beta, 0) \in L'.$$

Since $0 < \beta \leq 1/2$ and $\sum_{i=1}^{n} e_i \leq \beta n$, we have $||\hat{e}'||^2 \leq \beta(1-\beta)n$. We should consider the probability that there exists a vector $\hat{x} = (x_1, \ldots, x_{n+1})$ satisfying the following conditions:

$$||\hat{x}|| \leq ||\hat{e}'||, \quad \hat{x} \in L', \quad \hat{x} \notin \{0, \pm\hat{e}'\}. \tag{1}$$

We choose a positive integer N with $N > \sqrt{\beta(1-\beta)n}$. Then $\hat{x}$ satisfies the condition (1) only when $x_{n+1} = 0$, because, if not, we have $||\hat{x}|| \geq |x_{n+1}| \geq N > \sqrt{\beta(1-\beta)n} \geq ||\hat{e}'||$ which contradicts the condition (1).

Without loss of generality, we may assume $|t - s/\beta| \geq \alpha/2$ for $\alpha = \max a_i$ [1]. If

$$\hat{x} = \sum_{i=1}^{n} y_i b_i + y b'_{n+1}$$

satisfies the condition (1), then $\hat{x} = (x_1, \ldots, x_{n+1})$ is given by

$$x_i = y_i + \beta y \quad (i = 1, \ldots, n), \qquad x_{n+1} = N\left(sy + \sum_{i=1}^{n} a_i y_i\right) = 0.$$

Hence we have $-ys = \sum_{i=1}^{n} a_i(x_i - \beta y)$, namely, $\sum_{i=1}^{n} a_i x_i = \beta y(t - s/\beta)$. Thus we have

$$|\beta y(t - s/\beta)| = \left|\sum_{i=1}^{n} x_i a_i\right| \leq \left|\sum_{i=1}^{n} ||\hat{x}|| a_i\right| \leq ||\hat{x}|| \left|\sum_{i=1}^{n} a_i\right| \leq n\sqrt{n}\alpha\sqrt{\beta(1-\beta)}.$$

Since $|t - s/\beta| \geq \alpha/2$, we have

$$|y| \leq 2\sqrt{\beta^{-1} - 1} \cdot n^{3/2}.$$

Let us estimate the probability P where there exists a vector $\hat{x}$ which satisfies the condition (1). If we denote the denominator of the reduced fraction of β by D, the vector $x = (x_1, \ldots, x_n)$ satisfies the condition

$$x \in \{z + (j/D, \ldots, j/D) \mid z \in \mathbb{Z}^n, 0 \leq j < D\}.$$

[1] Let us consider the case $|t - s/\beta| < \alpha/2$. If α is included in s, by setting $s' = s - \alpha$, $t' = t - \alpha$, we have $|t' - s'/\beta| = |t - \alpha - s/\beta + \alpha/\beta| = |t - s/\beta + \alpha(1/\beta - 1)| \geq \alpha/2$. If α is included in $t - s$, by setting $s' = s$, $t' = t - \alpha$, we have $|t' - s'/\beta| = |t - s/\beta - \alpha| \geq \alpha/2$.

Then P is estimated by

$$\begin{aligned} P &\le \Pr\Bigg[\exists \hat{x}, y \mid ||\hat{x}|| \le ||\hat{e}'||,\ |y| \le 2\sqrt{\beta^{-1}-1}\cdot n^{3/2}, \\ &\qquad\qquad \hat{x} \notin \{0, \pm\hat{e}'\},\ \sum_{i=1}^{n} x_i a_i = \beta y(t - s/\beta)\Bigg] \\ &\le \Pr\left[\sum_{i=1}^{n} x_i a_i = \beta y(t-s/\beta) \,\middle|\, ||\hat{x}|| \le ||\hat{e}'||,\ |y| \le 2\sqrt{\beta^{-1}-1}\cdot n^{3/2},\ \hat{x} \notin \{0, \pm\hat{e}'\}\right] \\ &\qquad \cdot \left|\{\hat{x} : ||\hat{x}|| \le ||\hat{e}'||\}\right| \cdot \left|\left\{y \mid |y| \le 2\sqrt{\beta^{-1}-1}\cdot n^{3/2}\right\}\right|. \end{aligned} \tag{2}$$

For the first factor of the equation (2), we rewrite $\sum_{i=1}^{n} x_i a_i = \beta y(t - s/\beta)$ as

$$\sum_{i=1}^{n} z_i a_i = 0 \qquad (z_i = x_i - \beta y + y e_i).$$

Since $\hat{x} \neq 0$, we have $z = (z_1, \ldots, z_n) \neq 0$. By multiplying the probability bound by n, we may assume without loss of generality that $z_1 \neq 0$. If we set $z' = -(\sum_{i=2}^{n} a_i z_i / z_1)$,

$$\begin{aligned} \Pr\left[\sum_{i=1}^{n} a_i z_i = 0\right] &= \Pr[a_1 = z'] = \sum_{j=1}^{A} \Pr[a_1 = z' | z' = j] \cdot \Pr[z' = j] \\ &= \sum_{j=1}^{A} \Pr[a_1 = j] \cdot \Pr[z' = j] = \frac{1}{A} \sum_{j=1}^{A} \Pr[z' = j] \quad \le \frac{1}{A}. \end{aligned}$$

The second factor of the equation (2) is estimated by

$$\begin{aligned} &|\{\hat{x} \mid ||\hat{x}|| \le ||\hat{e}'||\}| \\ &\quad \le \left|\{x : ||x|| \le \sqrt{\beta(1-\beta)n}\}\right| \\ &\quad \le \left|\{w \in \mathbb{Z}^n : ||w|| \le \sqrt{\beta(1-\beta)n}\}\right| \\ &\qquad + \sum_{j=1}^{D-1} \left|\{w \in \mathbb{Z}^n \mid ||w - (j/D, \ldots, j/D)|| \le \sqrt{\beta(1-\beta)n}\}\right|. \end{aligned}$$

The first term is bounded by

$$2^{(\log_2 e)\delta_{\beta,0}(u)n}$$

for arbitrary $u \in \mathbb{R}^+$ by using the technique of Mazo and Odlyzko [8]. The absolute value of each term in the summation is bounded by

$$2^{(\log_2 e)\delta_{\beta,0}(u)n + \gamma_\beta \sqrt{n}}$$

for some constant γ_β, by using Theorem 2 in [8]. Thus, we have

$$\begin{aligned} &|\{x : ||x|| \le \sqrt{\beta(1-\beta)n}\}| \\ &\quad \le \min_u 2^{(\log_2 e)\delta_{\beta,0}(u)n} + (D-1) \min_{u'} 2^{(\log_2 e)\delta_{\beta,0}(u')n + \gamma_\beta\sqrt{n}} \\ &\quad \le 2^{(\log_2 e)\delta_{\beta,0}(u_0)n}(1 + (D-1)2^{\gamma_\beta\sqrt{n}}) \\ &\quad = 2^{c_0 n}(1 + (D-1)2^{\gamma_\beta\sqrt{n}}). \end{aligned}$$

By putting them all together, we have

$$P \leq n\left(4\sqrt{\beta^{-1}-1}\cdot n^{3/2}+1\right)\frac{2^{c_0 n}(1+(D-1)2^{\gamma_\beta\sqrt{n}})}{A}.$$

If the density of the subset sum problem is smaller than $1/c_0$, we have $P = 0$ (as $n \to \infty$). □

3.2 Covering with Polynomially Many Spheres

As we mentioned in Section 2, Proposition 2.1 does not imply the impossibility to cover $M_{1/2}$ with polynomially many spheres of radius $\sqrt{n/4 - o(n)}$. In this section, we discuss the case when the radius is $\sqrt{n/4 - O(1)}$.

Let k be a fixed positive integer with $k \leq \beta n$. Our basic strategy is as follows. In order to find a solution $e \in \{0,1\}^n$, we firstly fix k coordinates (i.e., k bits) and check remaining $(n-k)$ bits by calling lattice oracles. We have $\binom{n}{k}$ ways to select k bits and 2^k ways of the truth assignment on these coordinates. Then the dimension of the lattice call is reduced from $n+1$ to $n-k+1$ (for k is fixed). Moreover, the radius can be reduced to

$$r = \sqrt{\beta(1-\beta_k)n} < \sqrt{\beta(1-\beta)n}, \quad \textit{where} \quad \beta_k = \frac{\beta n}{n-k}.$$

As to the reason we can use $\beta(1-\beta_k)$ in dimension n instead of $\beta_k(1-\beta_k)$ in dimension $n-k$, refer to the proof of Theorem 3.2. Since β_k is not a constant, this case is not ruled by Proposition 2.1. At the return of each oracle call, we will check whether the returned value provides the true solution.

Thus our proposed algorithm is summarized as follows:

INPUT: $a_1, ..., a_n$ and s
OUTPUT: $(e_1, ..., e_n) \in \{0,1\}^n$ s.t. $\sum_{i=1}^n a_i e_i = s$
PROCEDURE:

foreach $J \subset \{1, ..., n\}$ with $|J| = k$
  /* $J = \{j_1, ..., j_k\}$ and $I = \{1, ..., n\} \setminus J = \{i_1, ..., i_{n-k}\}$ */
  foreach $(u_1, ..., u_k) \in \{0,1\}^k$
    invoke a lattice oracle with the following basis
    for $(n-k+1)$-dimensional lattice:

$$\begin{aligned} b_{i_1} &= (1, 0, ..., 0, Na_{i_1}), \\ b_{i_2} &= (0, 1, ..., 0, Na_{i_2}), \\ &\vdots \\ b_{i_{n-k}} &= (0, 0, ..., 1, Na_{i_{n-k}}), \\ b'_{n-k+1} &= (\beta_k, \beta_k, ..., \beta_k, N(s - \textstyle\sum_{l=1}^k a_{j_l} u_l)); \end{aligned}$$

    let $(e'_{i_1}, ..., e'_{i_{n-k}}, 0)$ be the return value;
    let $(e_{i_1}, ..., e_{i_{n-k}}) = (e'_{i_1} + \beta_k, ..., e'_{i_{n-k}} + \beta_k)$;
    let $(e_{j_1}, ..., e_{j_k}) = (u_1, ..., u_k)$;
    if $\sum_{i=1}^n a_i e_i = s$ and $(e_1, ..., e_n) \in \{0,1\}^n$

then output $(e_1, ..., e_n)$ and halt;
end;
end.

We remark some special case of our algorithm. In the above algorithm, there is a step in which we choose $(u_1, ..., u_k)$ from $\{0,1\}^k$. If the Hamming weight of the solution is guaranteed to be larger than k, like in Chor-Rivest cryptosystem case, we may set $(u_1, ..., u_k) = (1, ..., 1)$. In this case we can speed up the algorithm 2^k times faster and the radius $\sqrt{\beta(1-\beta_k)n}$ is reduced to $\sqrt{\beta'_k(1-\beta)n}$, where $\beta'_k = \frac{\beta n - k}{n-k}$.[2]

With regard to the above algorithm, we have the following theorem.

Theorem 3.2 Let $\beta \leq 1/2$ be a positive rational constant and k a positive integer with $k \leq \beta n$. Let A be a positive integer, and $a_1, \ldots, a_n$ random integers with $0 < a_i \leq A$ for $1 \leq i \leq n$. Let $e = (e_1, \ldots, e_n) \in \{0,1\}^n$ satisfy $\sum_{i=1}^n e_i \leq \beta n$ and let $s = \sum_{i=1}^n e_i a_i$. Let d be a constant.[3] If the density d of $\{a_1, \ldots, a_n\}$ satisfies

$$d < d_k = ((\log_2 e)\delta_{\beta,k}(u_k))^{-1},$$

then the subset sum problem defined by $a_1, \ldots, a_n$ and s can be almost always solved in polynomial time with $O(n^k)$ calls to a lattice oracle.

In the above statement, $\delta_{\beta,k}(u_k)$ is the minimum value of the following function of $u \in \mathbb{R}^+$:

$$\delta_{\beta,k}(u) = \beta(1-\beta_k)u + \ln \theta(e^{-u}).$$

We denote $(\log_2 e)\delta_{\beta,k}(u_k)$ by c_k. Note that the function $\delta_{\beta,0}$ used in Theorem 1 is a special case of the function $\delta_{\beta,k}(u)$. In other words, our result is a natural generalization of the improvement by Coster et al.

Proof. Our proof is similar to the proof of Theorem 1. Let $V_n(\alpha n)$ denote the number of lattice points in the n-dimensional sphere of radius $\sqrt{\alpha n}$ centered at some fixed point. In the estimation of the failure probability, we should consider $V_{n-k}(\beta_k(1-\beta_k)(n-k))$ in stead of $V_n(\beta(1-\beta)n)$, as we use lattice oracle calls of dimension $n-k$. Nevertheless, we may consider $V_n(\beta(1-\beta_k)n)$ in our case, because the following inequality holds:

$$V_{n-k}(\beta_k(1-\beta_k)(n-k)) \leq V_n(\beta_k(1-\beta_k)(n-k)) = V_n(\beta(1-\beta_k)n).$$

[2]The reduced radius is obtained from the fact that $V_{n-k}(\beta'_k(1-\beta'_k)(n-k)) \leq V_n(\beta'_k(1-\beta'_k)(n-k)) = V_n(\beta'_k(1-\beta)n)$, where $V_n(\alpha n)$ denote the number of lattice points in the n-dimensional sphere of radius $\sqrt{\alpha n}$ centered at some fixed point.

[3]We can relax the condition on the value of d. The value of d could depend on n. Some slight care about the closeness between d and d_k must be taken in order to guarantee that the failure probability P converges to 0.

When we represent β_k by a reduced fraction, the denominator is bounded by nD. Taking these differences into account, the probability estimation will be modified as follows:

$$\begin{aligned}
&|\{x : ||x|| \leq \sqrt{\beta(1-\beta_k)n}\}| \\
&\quad \leq \quad \min_u 2^{(\log_2 e)\delta_{\beta,k}(u)n} + (nD-1)\min_{u'} 2^{(\log_2 e)\delta_{\beta,k}(u')n+\gamma_{\beta,k}\sqrt{n}} \\
&\quad \leq \quad 2^{(\log_2 e)\delta_{\beta,k}(u_k)n}(1+(nD-1)2^{\gamma_{\beta,k}\sqrt{n}}) \\
&\quad = \quad 2^{c_k n}(1+(nD-1)2^{\gamma_{\beta,k}\sqrt{n}})
\end{aligned}$$

for some constant $\gamma_{\beta,k}$. Thus the failure probability P of our algorithm will be

$$P \leq (n-k)\left(4\sqrt{(\beta_k)^{-1}-1}\cdot(n-k)^{3/2}+1\right)\frac{2^{c_k n}(1+(nD-1)2^{\gamma_{\beta,k}\sqrt{n}})}{A}.$$

If the density of the subset sum problem is smaller than $1/c_k$, P is exponentially vanishing since d_k converges to d_0 which does not depend on n. □

4 The Effect of Our Algorithm

In this section we consider the effect of our algorithm numerically. Specifically speaking, we show that our algorithm is more effective than the previous algorithms for instances of any fixed n and density d such that $d_0 < d < d_k$. First, we calculate the value of $\delta_{\beta,k}(u_k)$ for some specific cases and see how much our algorithm improves the success probability of the attack. Then, we also see the Chor-Rivest cryptosystem case with its recommended parameters.

4.1 Improvement of the Success Probability

In the proof of Theorem 3.2, we evaluated the failure probability P of our proposed attack. The dominant term of P is $2^{(1/d_k-1/d)n}$, where $d_k = 1/(\log_2 e)\delta_{\beta,k}(u_k)$ and d is the density of the given problem. Hence, if $d < d_k$, the failure probability P is exponentially vanishing. When n is fixed in the cryptographic-use range, our algorithm makes d_k larger by increasing k, which means that the failure probability of one lattice oracle call becomes smaller.

Here we consider the best case when the Hamming weight of the solution is guaranteed to be larger than k. In this case, we can always take $(u_1, ..., u_k) = (1, ..., 1)$, and the radius $\sqrt{\beta(1-\beta_k)n}$ is reduced to $\sqrt{\beta'_k(1-\beta)n}$, where $\beta'_k = (\beta n - k)/(n-k)$. We can replace $\delta_{\beta,k}(u)$ with $\delta'_{\beta,k}(u)$:

$$\delta'_{\beta,k}(u) = \beta'_k(1-\beta)u + \ln\theta(e^{-u}).$$

In this case we denote the density bound by d'_k. We computed d'_k for $\beta = 0.5,\ 0.3,\ 0.1,\ k = 0, ..., 5,\ n = 100, 200, ..., 500$. We show the values of d'_k's in Tables 2,3,4, and plot them in the appendix. The figures in the appendix show that the effect of our attack is prominent when β is small. Note that the case $k = 0$ coincides with CJLOSS+ algorithm.

For example, let us consider the case $\beta = 0.1$ and $n = 200$. As $1/d'_0 = 0.5264...$ and $1/d'_2 = 0.4903...$, which mean that the density bound is improved, we have $(1/d'_0 - 1/d'_2) \times 200 = 7.22....$

Table 2: $\beta = 0.5$

n	100	200	300	400	500
$k = 0$	0.9408	0.9408	0.9408	0.9408	0.9408
1	0.9467	0.9438	0.9428	0.9423	0.9420
2	0.9529	0.9467	0.9448	0.9438	0.9432
3	0.9593	0.9498	0.9467	0.9452	0.9444
4	0.9660	0.9529	0.9488	0.9467	0.9455
5	0.9729	0.9561	0.9508	0.9483	0.9467

Table 3: $\beta = 0.3$

n	100	200	300	400	500
$k = 0$	1.0502	1.0502	1.0502	1.0502	1.0502
1	1.0666	1.0583	1.0556	1.0542	1.0534
2	1.0840	1.0666	1.0610	1.0583	1.0566
3	1.1027	1.0752	1.0666	1.0624	1.0599
4	1.1228	1.0840	1.0723	1.0666	1.0632
5	1.1444	1.0932	1.0781	1.0708	1.0666

Table 4: $\beta = 0.1$

n	100	200	300	400	500
$k = 0$	1.8994	1.8994	1.8994	1.8994	1.8994
1	2.0392	1.9659	1.9430	1.9318	1.9252
2	2.2113	2.0392	1.9895	1.9659	1.9521
3	2.4287	2.1206	2.0392	2.0016	1.9800
4	2.7128	2.2113	2.0925	2.0392	2.0090
5	3.1010	2.3133	2.1497	2.0788	2.0392

This implies that if we take $k = 2$, our algorithm makes the failure probability of one lattice oracle call $2^{7.2}$ times smaller than CJLOSS+ algorithm case.

Regarding the running time, our algorithm needs to make $O(n^k)$ lattice oracle calls, while CJLOSS+ algorithm needs just one lattice oracle call. The dimension of the lattice in our case is $n - k + 1$ and slightly smaller than $n + 1$ in the CJLOSS+ case.

4.2 Chor-Rivest Cryptosystem Case

	CR1	CR2	CR3	CR4
n	197	211	243	256
h	24	24	24	25
β	0.121827	0.113744	0.0987654	0.0976563
d	1.0769191	1.1386548	1.2776327	1.2799999
$k = 0$	1.6728719	1.7470959	1.9150294	1.9294014
1	1.7191829	1.7962555	1.9705846	1.9831705
2	1.7694137	1.8495713	2.0308281	2.0412799
3	1.8240947	1.9076058	2.0963944	2.1042906
4	1.8838584	1.9710296	2.1680401	2.1728660
5	1.9494646	2.0406485	2.2466747	2.2477973

Table 5: Recommended parameters of the Chor-Rivest cryptosystem and d'_k of our improved attack

Next, let us consider the Chor-Rivest cryptosystem. Security parameters of the cryptosystem are a prime power n and a positive integer $h < n$ such that the discrete logarithm problem in $GF(n^h)$ is tractable. A plaintext is an n-bit sequence $e = (e_1, \ldots, e_n) \in \{0,1\}^n$ with Hamming weight h, namely $\sum_{i=1}^n e_i = h$. A ciphertext s corresponding to e is given by $s = \sum_{i=1}^n a_i e_i$ where $a_i \in \mathbb{Z}/(n^h - 1)\mathbb{Z}$. Here the density of the Chor-Rivest cryptosystem is given by

$$d = \frac{n}{\log_2 (n^h - 2)} \approx \frac{n}{h \log_2 n} = \frac{1}{\beta \log_2 n},$$

where $\beta = h/n$. Table 5 shows the recommended parameter sets CR1 $\sim$ CR4 for the Chor-Rivest cryptosystem. Note that these parameters are considered to be secure against the low-density attack by Coster et al. (CJLOSS algorithm) because their densities are beyond 0.9408.

For each parameter set, we compare the density d and the d'_k's of our attack in Table 5. As in the table, the d'_k's of our improved attack (including CJLOSS+ algorithm) are far larger than the density d. Thus the failure probability of our improved attack will be very small.

5 Concluding Remarks

In this paper, we gave yet another improved low-density attack algorithm to solve the subset sum problem. Our improvement was based on the idea that lattice points can be covered with polynomially many spheres of shorter radius and of lower dimension than the previous algorithms. Though

the asymptotical behavior of our algorithm coincides with one of the suggested algorithm by Coster et al., our algorithm makes the success probability higher for fixed parameters. For example, if we consider a typical setting where $n = 200, k = 2, \beta = 0.1$, our algorithm can make the failure probability of one lattice oracle call $2^{7.2}$ times smaller. We also showed that the failure probability of our improved attack will become very small against the Chor-Rivest cryptosystem with its recommended parameters, if we assume an SVP oracle call.

Acknowledgments

We are grateful to Hisashi Usui for helpful discussions about the radius covering problem.

References

[1] E. F. Brickell, Breaking iterated knapsacks, Advances in Cryptology: Proceedings of CRYPTO'84 (G. R. Blakley and D. Chaum, eds.), Lecture Notes in Computer Science, Springer-Verlag, New York, 196 (1985) pp.342–358.

[2] 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 (1992) pp.111–128.

[3] B. Chor, and R. L. Rivest, A knapsack-type public key cryptosystem based on arithmetic in finite fields, IEEE Transactions on Information Theory, Vol.34, No.5 (1988) pp.901–909.

[4] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W. H. Freeman and Co., San Fransisco (1979).

[5] A. K. Lenstra, H. W. Lenstra Jr., and L. Lovász, Factoring polynomials with rational coefficients, Mathematische Annalen, Vol.261 (1982) pp.515–534.

[6] J. C. Lagarias, and A. M. Odlyzko, Solving low-density subset sum problems, Journal of the Association for Computing Machinery, Vol.32, No.1 (1985) pp.229–246.

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

[8] J. E. Mazo, and A. M. Odlyzko, Lattice points in high-dimensional spheres, Monatshefte für Mathematik, Vol.110 (1990) pp.47–61.

[9] P. Q. Nguyen, and J. Stern, The two faces of lattices in cryptology, Cryptography and Lattices: Proceedings of CaLC 2001 (J. H. Silverman, ed.), Lecture Notes in Computer Science, Springer-Verlag, New York, 2146 (2001) pp.146–180.

[10] T. Okamoto, K. Tanaka, and S. Uchiyama, Quantum public-key cryptosystems, Advances in Cryptology: Proceedings of CRYPTO 2000 (M. Bellare, ed.), Lecture Notes in Computer Science, Springer-Verlag, New York, 1880 (2000) pp.147–165.

[11] C. P. Schnorr, and M. Euchner, Lattice basis reduction: Improved practical algorithms and solving subset sum problems, Mathematical Programming, Vol.66 (1994) pp.181–199.

[12] C. P. Schnorr and H. H. Hörner, Attacking the Chor-Rivest cryptosystem by improved lattice reduction, Advances in Cryptology: Proceedings of EUROCRYPT'95 (L. C. Guillou and J.-J. Quisquater, eds.), Lecture Notes in Computer Science, Springer-Verlag, New York, 921 (1995) pp.1–12.

[13] 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, (1982) pp.145–152.

[14] S. Vaudenay, Cryptanalysis of the Chor–Rivest cryptosystem, Journal of Cryptology, Vol.14, No.2 (2001) pp.87–100.

A Figures of d'_k

In this appendix, we give some figures in order to show the effect of our proposed attack. As in Section 4, we computed d'_k of our improved attack for $\beta = 0.5,\ 0.3,\ 0.1$, $k = 0, ..., 5$. We plot the values for each β in Fig.1,2,3, respectively. In each figure, the horizontal axis denotes the value of n and the vertical axis denotes the value of d'_k.

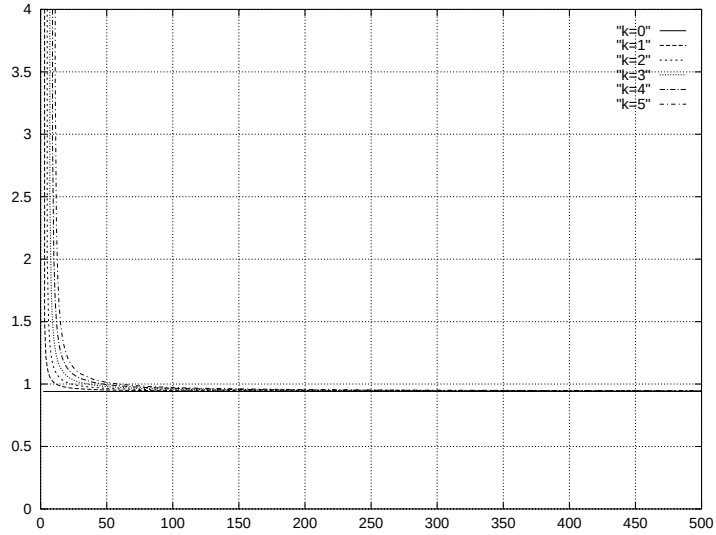

Figure 1: d'_k $(\beta = 0.5; k = 0, 1, 2, 3, 4, 5)$

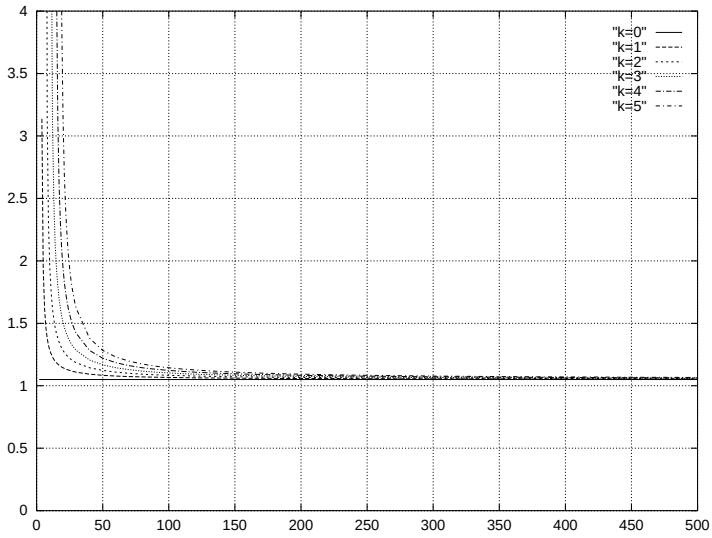

Figure 2: d'_k $(\beta = 0.3; k = 0, 1, 2, 3, 4, 5)$

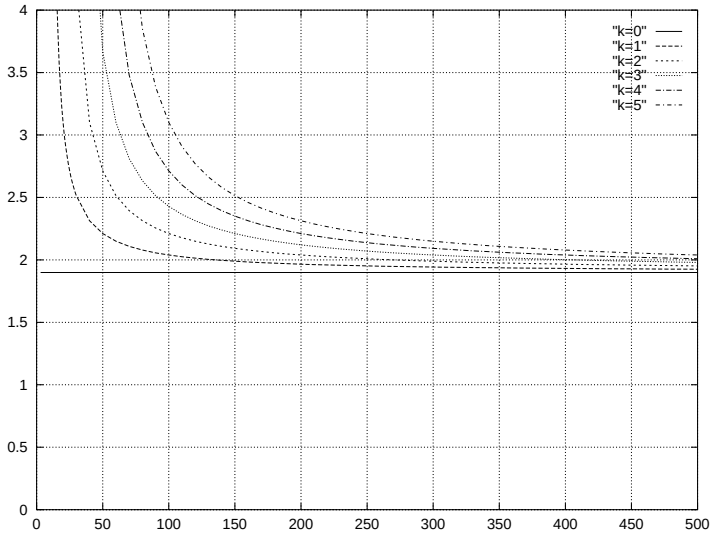

Figure 3: d'_k $(\beta = 0.1; k = 0, 1, 2, 3, 4, 5)$