Breaking Knapsack Cryptosystems by l_∞–norm Enumeration

H. Ritter*
Johann Wolfgang Goethe–Universität Frankfurt
Fachbereich Mathematik
Postfach 111932, D–60054 Frankfurt a. M., Germany

Abstract

At EUROCRYPT '94 G. Orton proposed a public key cryptosystem based on dense compact knapsacks. We present an efficient depth first search enumeration of l_∞–norm short lattice vectors based on Hoelder's inequality and apply this algorithm to break Orton's cryptosystem.

Keywords: NP–hardness, Knapsack problem, Subset sum problem, Breaking knapsack cryptosystems, Shortest lattice vector problem, Lattice basis reduction, Hoelder's inequality.

1 Introduction and Summary

A number of cryptosystems have been based on knapsack problems and it was hoped that the NP–hardness of the knapsack problem makes it hard to break the corresponding cryptosystem. A *knapsack* consists of positive integers $a_1, \ldots, a_n, y$. A solution are integers $x_1, \ldots, x_n$ in some interval $[0, 2^s)$ that satisfy $\sum_{i=1}^n a_i x_i = y$. If $s > 1$ the knapsack is called *compact*, knapsack problems with $s = 1$ are *subset sum problems.* The *density* of a knapsack is the quotient $(n * s)/(\text{bitlength of the maximal } a_i)$. Merkle–Hellman [MH78] use knapsacks with density < 1 for a public key cryptosystem. Lagarias, Odlyzko et al. [LO85, CJLOSS92] represent subset sum problems by lattices. They show that, for density $< 0.9408\ldots$, a shortest nonzero lattice vector in l_2–norm almost always transforms into a solution of the subset sum problem. It is an open problem wether it is possible to find l_2–norm shortest lattice vectors in polynomial time. In practice the L^3–algorithm of Lenstra, Lenstra, Lovász [LLL82] and block reduction [SE94, S87, S94] are used to find short lattice vectors.

To prevent low density attacks Orton [O94] proposes a cryptosystem based on compact knapsacks with density > 1. In this paper we introduce new techniques for solving dense compact knapsacks and in particular the Orton–scheme. The algorithm of this paper for the first time enumerates short lattice vectors in the l_∞–norm. It is surprisingly efficient even though the problem of finding an l_∞–norm shortest lattice vector is NP–hard and thus believed to be more difficult than finding shortest lattice vectors in the l_2–norm. We greatly improve the enumeration of short lattice vectors in the l_∞–norm by pruning the

*e–mail: ritter@mi.informatik.uni-frankfurt.de

enumeration via Hoelder's inequality. This pruning reduces the costs of the enumeration by an exponential factor 0.82^n without missing the shortest lattice vector.
Throughout the paper let $\lfloor x \rfloor$, $\lceil x \rceil$ denote the greatest (resp. smallest) integer smaller (resp. greater) or equal x and $\lceil x \rfloor := \lfloor x + 0.5 \rfloor$.

2 The Cryptosystem

Orton [O94] proposes for a public key cryptosystem a multiple–iterated trapdoor for dense compact knapsacks. We demonstrate how to break the scheme with pruned enumeration. Here is a brief description of the Orton–scheme, for further details see [O94].

Public parameters: positive integers r, n, s. (Messages consist of n blocks with s bits each; r is the number of rounds for key generation.)

Secret key: a series of integers $a_i^{(0)},\ i = 1, \dots, n$ with $a_1^{(0)} = 1,\ a_i^{(0)} > (2^s - 1) \sum_{j=1}^{i-1} a_j^{(0)}$ and positive integers $q_2,\ p^{(k)}, w^{(k)}$ for $k = 1, \dots, r$, where $q_1 := p^{(r)}/q_2 \in \mathbb{Z}$.

The secret key $\{a_i^{(0)}\}$ representing an "easy" knapsack is transformed into a "hard" knapsack which represents the public key by the operations

$$\begin{aligned}
a_i^{(k)} &:= a_i^{(k-1)} w^{(k)} \bmod p^{(k)} \text{ for } i = 1, \dots, n+k-1, \quad a_{n+k}^{(k)} := -p^{(k)}, \\
f_i^{(k)} &:= 2^{-\mathrm{prec}(k)} \lfloor a_i^{(k)} 2^{\mathrm{prec}(k)} / p^{(k)} \rfloor \text{ for } i = 1, \dots, n+k-1, \quad k = 1, \dots, r, \\
a_{i,j} &:= a_i^{(r)} \bmod q_j \text{ for } i = 1, \dots, n+r-1,\ j = 1, 2
\end{aligned}$$

using the secret "trapdoor" $q_2,\ p^{(k)}, w^{(k)}$ for $k = 1, \dots, r$. $\mathrm{prec}(k)$ is the number of precision bits for the fractions $f_i^{(k)}$ in the k–th round. Orton proposes $\mathrm{prec}(k) = s + \log_2 n + k + 2$. This choice guarantees unique encryption and prevents known attacks like Brickell's [B84] and Shamir's [S79].

Public key: positive integers $q_1,\ \mathrm{prec}(k)$ for $k = 1, \dots, r-1$,
nonnegative integers $a_{i,j}$ for $i = 1, \dots, n+r-1, \quad j = 1, 2$,
rational numbers $f_i^{(k)} \in 2^{-\mathrm{prec}(k)}[0, 2^{\mathrm{prec}(k)})$ for $k = 1, \dots, r-1,\ i = 1, \dots, n+k-1$.

ENCRYPTION
INPUT: public key, message $x_1, \dots, x_n \in [0, 2^s)$
1. $x_{n+k} := \lfloor \sum_{i=1}^{n+k-1} x_i f_i^{(k)} \rfloor$ for $k = 1, \dots, r-1$
2. $y_1 := \sum_{i=1}^{n+r-1} x_i a_{i,1} \bmod q_1, \quad y_2 := \sum_{i=1}^{n+r-1} x_i a_{i,2}$
OUTPUT: ciphertext y_1, y_2

DECRYPTION
INPUT: public and secret key, ciphertext y_1, y_2
1. recombine $y^{(r)} \equiv y_j \bmod q_j$ $(j = 1, 2)$ with Chinese remainder theorem.
   $y^{(r)} := q_2((y_1 - y_2) q_2^{-1} \bmod q_1) + y_2$
2. $y^{(k-1)} := y^{(k)} (w^{(k)})^{-1} \bmod p^{(k)}$ for $k = r, \dots, 1$
3. solve $\sum_{i=1}^{n} x_i a_i^{(0)} = y^{(0)}$ with $x_i \in [0, 2^s)$ (this is easy since $a_i^{(0)} > (2^s - 1) \sum_{j=1}^{i-1} a_j^{(0)}$).
OUTPUT: cleartext message $x_1, \dots, x_n$

3 The l_∞–norm shortest lattice vector attack

We associate to the decryption problem linearly independent integer vectors $b_1, \dots, b_{m+2} \in \mathbb{Z}^{m+r+2}$ so that any integer linear combination of these vectors with l_∞–norm 1 yields the original message. The l_∞–norm $\|v\|_\infty$ of a vector v is the maximal absolute value of its coefficients v_i. The integer linear combinations of the *basis* vectors $b_1, \dots, b_{m+2}$ form a *lattice*. The L^3–algorithm of Lenstra, Lenstra, Lovász [LLL82, SE94] transforms the given lattice basis into a lattice basis consisting of l_2–norm short vectors. This *reduced* basis allows us to find a lattice vector v with l_∞–norm 1 via pruned enumeration.

The decryption problem is stated as follows: Given the public key, $y_1 \bmod q_1$ and y_2 find integers $x_1, \dots, x_n \in [0, 2^s)$, $x_{n+k} \in [0, 2^{s+k+\log_2 n-1})$ satisfying

$$\sum_{i=1}^{n+r-1} x_i a_{i,1} = y_1 \bmod q_1 \tag{1}$$

$$\sum_{i=1}^{n+r-1} x_i a_{i,2} = y_2 \tag{2}$$

$$x_{n+k} = \lfloor \sum_{i=1}^{n+k-1} x_i f_i^{(k)} \rfloor \quad \text{for } k = 1, \dots, r-1 \tag{3}$$

We transform equations (1)–(3) into a set of $r+1$ integer linear equations with m 0–1–unknowns, where $m := ns + (r-1)(r/2 + s + \lceil \log_2 n \rceil - 1) + \sum_{k=1}^{r-1} \mathrm{prec}(k)$ (see (6) below). Since $f_i^{(k)} 2^{\mathrm{prec}(k)} \in [0, 2^{\mathrm{prec}(k)})$ is integral we can write (3) as

$$x_{n+k} 2^{\mathrm{prec}(k)} = \sum_{i=1}^{n+k-1} x_i f_i^{(k)} 2^{\mathrm{prec}(k)} - x_{n+r+k-1} \quad \text{for } k = 1, \dots, r-1, \tag{4}$$

where the additional variables $x_{n+r+k-1}$ are integers in $[0, 2^{\mathrm{prec}(k)})$.
With $a_{i,k+2} := f_i^{(k)} 2^{\mathrm{prec}(k)}$ for $i = 1, \dots, n+k-1$, $a_{n+k,k+2} := -2^{\mathrm{prec}(k)}$, $a_{n+r+k-1,k+2} := -1$ and $a_{i,k+2} := 0$ else equations (4) simplifiy to

$$\sum_{i=1}^{n+2r-2} x_i a_{i,k+2} = 0 \quad \text{for } k = 1, \dots, r-1 \tag{5}$$
$$\text{with} \qquad x_{n+r+k-1} \in [0, 2^{\mathrm{prec}(k)}) \quad \text{for } k = 1, \dots, r-1.$$

The unique solution of (1),(2),(5) directly transforms into the unique solution of (1)–(3).

To get 0–1–variables we regard the binary representation of the integer variables:

We set $d_i := \begin{cases} s & \text{for } 1 \le i \le n \\ s + i + \lceil \log_2 n \rceil - n - 1 & \text{for } n+1 \le i \le n+r-1 \\ \mathrm{prec}(i - (n+r-1)) & \text{for } n+r \le i \le n+2r-2 \end{cases}$ and $D_i := \sum_{j=1}^{i-1} d_j$.

Let $t_{D_i+1}, \dots, t_{D_i+d_i} \in \{0, 1\}$ be the binary representation of x_i, i.e. $x_i = \sum_{l=0}^{d_i-1} t_{D_i+l+1} 2^l$, and set $A_{D_i+l+1,j} := a_{i,j} 2^l$ for $i = 1, \dots, n+2r-2$, $j = 1, \dots, r+1$, $l = 0, \dots, d_i - 1$, where

$a_{i,1} := a_{i,2} := 0$ for $i > n + r - 1$.
With $y_3 := \ldots := y_{r+1} := 0$ equations (1),(2),(5) simplify to

$$\begin{aligned} \sum_{i=1}^{m} t_i A_{i,1} &= y_1 + z q_1 \\ \sum_{i=1}^{m} t_i A_{i,j} &= y_j \quad \text{for } j = 2, \ldots, r+1, \\ \text{where} \qquad & t_i \in \{0,1\}, \quad z \in \mathbb{Z} \end{aligned} \tag{6}$$

We regard the row vectors $b_1, \ldots, b_{m+2} \in \mathbb{Z}^{m+r+2}$ of the following matrix (7) as basis of the lattice L.

$$\begin{pmatrix} 0 & 2 & 0 & \cdots & 0 & NA_{1,1} & NA_{1,2} & \cdots & NA_{1,r+1} \\ 0 & 0 & 2 & \ddots & 0 & NA_{2,1} & NA_{2,2} & \cdots & NA_{2,r+1} \\ \vdots & \vdots & \ddots & \ddots & \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & 0 & 2 & NA_{m,1} & NA_{m,2} & \cdots & NA_{m,r+1} \\ 0 & 0 & \cdots & 0 & 0 & Nq_1 & 0 & \cdots & 0 \\ 1 & 1 & \cdots & 1 & 1 & Ny_1 & Ny_2 & \cdots & Ny_{r+1} \end{pmatrix} \tag{7}$$

For every integer $N \geq 2$ the following statement holds:
Every vector $v = (v_0, \ldots, v_{m+r+1}) = \sum_{i=1}^{m+2} c_i b_i \in L$ with l_∞–norm 1 is a l_∞–norm shortest nonzero lattice vector and has the form $\{\pm 1\}^{m+1} \times 0^{r+1}$, where $c_{m+2} \in \{\pm 1\}$, $c_{m+1} \in \mathbb{Z}$ and $c_1, \ldots, c_m \in \{0, -c_{m+2}\}$. The zero in the last $r+1$ coefficients imply

$$\sum_{i=1}^{m} c_i A_{i,1} + c_{m+2} y_1 = 0 \bmod q_1 \tag{8}$$

$$\sum_{i=1}^{m} c_i A_{i,j} + c_{m+2} y_j = 0 \quad \text{for } j = 2, \ldots, r+1. \tag{9}$$

With $t_i := |c_i| = (|v_i - v_0|)/2$ for $i = 1, \ldots, m$ we obtain the unique solution of (6) which directly transforms into the original message.

4 Enumeration of shortest lattice vectors

Let $\mathbb{R}^n$ be the n–dimensional real vector space with ordinary inner product $<.,.>$, l_2–norm $\|x\|_2 = <x,x>^{1/2}$, l_∞–norm $\|x\|_\infty = \max_i(|x_i|)$ and l_1–norm $\|x\|_1 = \sum_{i=1}^{n} |x_i|$.

Hoelder's inequality: $|<x,y>| \leq \|x\|_\infty \|y\|_1$ for all $x, y \in \mathbb{R}^n$.

With an ordered lattice basis $b_1, \ldots, b_m \in \mathbb{R}^n$ we associate the Gram–Schmidt orthogonalisation $\hat{b}_1, \ldots, \hat{b}_m \in \mathbb{R}^n$ which can be computed together with the Gram–Schmidt coefficients $\mu_{i,j} = <b_i, \hat{b}_j>/<\hat{b}_j, \hat{b}_j>$ by the recursion $\hat{b}_1 = b_1$, $\hat{b}_i = b_i - \sum_{j=1}^{i-1} \mu_{i,j} \hat{b}_j$ for $i = 2, \ldots, m$. We define the orthogonal projections $\pi_i : \mathbb{R}^n \to \text{span}(b_1, \ldots, b_{i-1})^\perp$ for $i = 1, \ldots, m$. Clearly, $\pi_i(b_j) = \sum_{t=i}^{j} \mu_{i,t} \hat{b}_t$.

For $t = m, \ldots, 1$ we define the following functions w_t, $\tilde{c}_t$ with integer arguments $\tilde{u}_t, \ldots, \tilde{u}_m$:

$$\begin{aligned} w_t := w_t(\tilde{u}_t, \ldots, \tilde{u}_m) &:= \pi_t(\sum_{i=t}^{m} \tilde{u}_i b_i) = w_{t+1} + \left(\sum_{i=t}^{m} \tilde{u}_i \mu_{i,t} \right) \hat{b}_t \\ \tilde{c}_t := \tilde{c}_t(\tilde{u}_t, \ldots, \tilde{u}_m) &:= \|w_t\|_2^2 = \tilde{c}_{t+1} + \left(\sum_{i=t}^{m} \tilde{u}_i \mu_{i,t} \right)^2 \|\hat{b}_t\|_2^2 \end{aligned}$$

The algorithm ENUM of [SE94] enumerates in depth first search order all nonzero integer vectors $(\tilde{u}_t, \ldots, \tilde{u}_m)$ for $t = m, \ldots, 1$ satisfying $\tilde{c}_t(\tilde{u}_t, \ldots, \tilde{u}_m) < \bar{c}_1$, where $\bar{c}_1$ is the current minimum for the function $\tilde{c}_1(\tilde{u}_1, \ldots, \tilde{u}_m)$.
We modify this algorithm to enumerate all short lattice vectors with respect to the l_∞–norm. We recursively enumerate all nonzero integer vectors $(\tilde{u}_t, \ldots, \tilde{u}_m)$ for $t = m, \ldots, 1$ satisfying $\tilde{c}_t(\tilde{u}_t, \ldots, \tilde{u}_m) < n\bar{B}^2$, where $\bar{B}$ is the current minimal l_∞–norm of all enumerated lattice vectors w_1. The resulting enumeration area is illustrated in figure 1. We enumerate all vectors $w_t(\tilde{u}_t, \ldots, \tilde{u}_m)$ inside the sphere B with radius $\sqrt{n}\,\bar{B}$ centered at the origin. To avoid redundancies all enumerated vectors satisfy $\tilde{u}_s > 0$, where s is the largest i with $\tilde{u}_i \neq 0$. For fixed $\tilde{u}_{t+1}, \ldots, \tilde{u}_m$ the sequence of values for $\tilde{u}_t$ is chosen so that the function $\tilde{c}_t(\tilde{u}_t, \ldots, \tilde{u}_m)$ is non–decreasing. We can prune the enumeration using the following observations.
Since, for fixed $\tilde{u}_t, \ldots, \tilde{u}_m$, we can only reach lattice vectors in the hyperplane H orthogonal to $w_t(\tilde{u}_t, \ldots, \tilde{u}_m)$, we can prune the enumeration as soon as this hyperplane doesn't intersect with the set M of all points with l_∞–norm less or equal $\bar{B}$. Using Hoelder's inequality we get $\tilde{c}_t(\tilde{u}_t, \ldots, \tilde{u}_m) > \bar{B}\,\|w_t(\tilde{u}_t, \ldots, \tilde{u}_m)\|_1$ whenever the intersection is empty. In this case we don't need to enumerate any integers $\tilde{u}_{t-1}, \ldots, \tilde{u}_1$ for the fixed $\tilde{u}_t, \ldots, \tilde{u}_m$. The inequality can be tested in linear time and restricts the enumeration to the shaded area U of figure 1, where U is the union of all balls with radius $\frac{1}{2}\sqrt{n}\,\bar{B}$ centered in $\{\pm\bar{B}/2\}^n$.

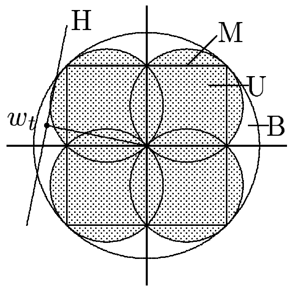

Figure 1

The volume of U is an exponential fraction ($\approx 0.82^{n-1}$) of the volume of B. Lemma 1 formalizes and generalizes this pruning rule.

Lemma 1 *Let* $(\tilde{u}_t, \ldots, \tilde{u}_m) \in \mathbb{Z}^{m-t+1}$ *be fixed.*
Assume we are given a vector $(\lambda_t, \ldots, \lambda_m) \in \mathbb{R}^{m-t+1}$ *satisfying*

$$|\sum_{i=t}^{m} \lambda_i \tilde{c}_i(\tilde{u}_i, \ldots, \tilde{u}_m)| > c\,\|\sum_{i=t}^{m} \lambda_i w_i(\tilde{u}_i, \ldots, \tilde{u}_m)\|_1. \tag{10}$$

Then $\|\sum_{i=1}^{m} \tilde{u}_i b_i\|_\infty > c$ *for all* $\tilde{u}_1, \ldots, \tilde{u}_{t-1} \in \mathbb{Z}$.

We can even do better. For all $\tilde{u}_1, \ldots, \tilde{u}_m$ the vectors $w_1(\tilde{u}_1, \ldots, \tilde{u}_m), \ldots, w_m(\tilde{u}_m)$ all lie on the surface of the ball W with radius $\frac{1}{2}\|w_1\|_2$ centered at $\frac{1}{2}w_1$. Hence W has to be a subset of U if $\|w_1\|_\infty \le \bar{B}$. Therefore, the whole line between w_{t+1} and w_t must be part of U. Thus we can stop the enumeration of all coefficients $\tilde{u}'_t = (1+\lambda)\tilde{u}_t - \lambda \sum_{i=t+1}^m \tilde{u}_i \mu_{i,t}$ for fixed $\tilde{u}_{t+1}, \ldots, \tilde{u}_m$ and $\lambda > 0$ whenever $\tilde{c}_t(\tilde{u}_t, \ldots, \tilde{u}_m) > \bar{B}\,\|w_t(\tilde{u}_t, \ldots, \tilde{u}_m)\|_1$. This coefficients would yield vectors w'_t on the dotted line of figure 2 and thus the line between w_{t+1} and w'_t would not be part of U.

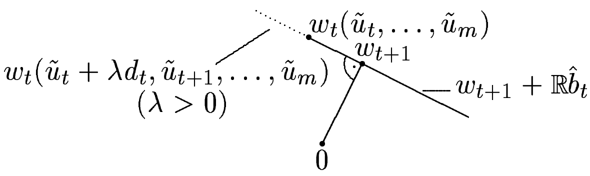

Figure 2

The additional pruning rule is formalized and generalized in lemma 2.

Lemma 2 *Let* $(\tilde{u}_t, \ldots, \tilde{u}_m) \in \mathbb{Z}^{m-t+1}$ *be fixed and* $d_t := \tilde{u}_t - \sum_{i=t+1}^m \tilde{u}_i \mu_{i,t}$.
Assume that (10) holds for a given $(\lambda_t, \ldots, \lambda_m)$ *with* $\lambda_t > 0$ *and* $\sum_{i=t}^m \lambda_i \tilde{c}_i(\tilde{u}_t, \ldots, \tilde{u}_m) \ge 0$.
Then $\|\sum_{i=1}^m \tilde{u}_i b_i + \lambda d_t b_t\|_\infty > c$ *holds for all* $\lambda > 0$ *and all* $\tilde{u}_1, \ldots, \tilde{u}_{t-1} \in \mathbb{Z}$.

Proof of Lemma 1: Since $w_i = \pi_i(\sum_{j=i}^m \tilde{u}_j b_j) \in \operatorname{span}(b_1, \ldots, b_{i-1})^\perp$ for $i = t, \ldots, m$ we have $w_i \perp \sum_{j=1}^m \tilde{u}_j b_j - w_i$ and thus $<w_i, w_i> \; = \; <\sum_{j=1}^m \tilde{u}_j b_j, w_i>$ for all $\tilde{u}_1, \ldots, \tilde{u}_{t-1} \in \mathbb{Z}$. With Hoelder's inequality we get

$$\begin{aligned} c\,\|\sum_{i=t}^m \lambda_i w_i\|_1 \quad &< \quad |\sum_{i=t}^m \lambda_i \tilde{c}_i| = |\sum_{i=t}^m \lambda_i <w_i, w_i>| = |\sum_{i=t}^m \lambda_i <\sum_{j=1}^m \tilde{u}_j b_j, w_i>| \\ &= \quad |<\sum_{j=1}^m \tilde{u}_j b_j, \sum_{i=t}^m \lambda_i w_i>| \le \|\sum_{j=1}^m \tilde{u}_j b_j\|_\infty \, \|\sum_{i=t}^m \lambda_i w_i\|_1. \qquad \square \end{aligned}$$

Proof of Lemma 2: Let $\lambda > 0$ be fixed. For abbreviation we set $\tilde{u}'_t := \tilde{u}_t + \lambda d_t$ and $\tilde{u}'_i := \tilde{u}_i$ for $i = t+1, \ldots, m$. With $\lambda'_t := \lambda_t/\lambda$, $\lambda'_{t+1} := \lambda_{t+1} + \lambda_t - \lambda_t/\lambda$ and $\lambda'_i := \lambda_i$ for $i = t+2, \ldots, m$ we have

$$\lambda'_t w_t(\tilde{u}'_t, \ldots, \tilde{u}'_m) + \lambda'_{t+1} w_{t+1}(\tilde{u}'_{t+1}, \ldots, \tilde{u}'_m) = \lambda_t w_t(\tilde{u}_t, \ldots, \tilde{u}_m) + \lambda_{t+1} w_{t+1}(\tilde{u}_{t+1}, \ldots, \tilde{u}_m).$$

We get

$$\begin{aligned} \sum_{i=t}^m \lambda'_i \tilde{c}_i(\tilde{u}'_i, \ldots, \tilde{u}'_m) \quad &\ge \quad \sum_{i=t}^m \lambda_i \tilde{c}_i(\tilde{u}_i, \ldots, \tilde{u}_m) \\ &\overset{(10)}{>} \quad c\,\|\sum_{i=t}^m \lambda_i w_i(\tilde{u}_i, \ldots, \tilde{u}_m)\|_1 = c\,\|\sum_{i=t}^m \lambda'_i w_i(\tilde{u}'_i, \ldots, \tilde{u}'_m)\|_1. \end{aligned}$$

Lemma 1, applied with $(\tilde{u}_t + \lambda d_t, \tilde{u}_{t+1}, \ldots, \tilde{u}_m)$ and $(\lambda'_t, \ldots, \lambda'_m)$, completes the proof. $\square$

Using Hoelder's inequality and the techniques of the ellipsoid method [K79] we can test (10) in polynomial time. In practice we only use the simpler linear–time test (i.e. we test (10) for $(\lambda_t, \ldots, \lambda_m) = (1, 0, \ldots, 0))$ which seems to yield better performance.

The following algorithm ENUM$_\infty$ generates a lattice vector with minimal l_∞–norm by pruned enumeration in depth first search order. For fixed $\tilde{u}_{t+1}, \ldots, \tilde{u}_m$ the enumeration order of the $\tilde{u}_t$–values is controlled by the variables Δ_t, δ_t and η_t. The variables Δ_t, δ_t are the same as in [SE94], η_t is the number of directions at stage t for which the enumeration is already cut according to lemma 2.

Algorithm ENUM$_\infty$
INPUT: $\hat{b}_i,\ c_i := \|\hat{b}_i\|_2^2, \mu_{i,t}$ for $1 \le t \le i \le m$
1. FOR $i = 1, \ldots, m+1$
        $\tilde{c}_i := u_i := \tilde{u}_i := v_i := y_i := \Delta_i := 0,\ \eta_i := \delta_i := 1,\ w_i := (0, \ldots, 0)$
   $u_1 := \tilde{u}_1 := 1,\ s := t := 1,\ \bar{b} := b_1,\ \bar{c} := n\|b_1\|_\infty^2,\ \bar{B} := \|b_1\|_\infty$
2. WHILE $t \le m$
        $\tilde{c}_t := \tilde{c}_{t+1} + (y_t + \tilde{u}_t)^2 c_t$
        IF $\tilde{c}_t < \bar{c}$
        THEN $w_t := w_{t+1} + (y_t + \tilde{u}_t)\hat{b}_t$
                IF $t > 1$
                THEN IF $\tilde{c}_t \ge \bar{B}\,\|w_t\|_1$
                        THEN IF $\eta_t = 1$ THEN INCREASE_t()
                                ELSE $\eta_t := 1,\ \Delta_t := -\Delta_t$
                                        IF $\Delta_t \delta_t \ge 0$ THEN $\Delta_t := \Delta_t + \delta_t$
                                        $\tilde{u}_t := v_t + \Delta_t$
                        ELSE $t := t-1,\ \eta_t := \Delta_t := 0,\ y_t := \sum_{i=t+1}^{s} \tilde{u}_i \mu_{i,t}$
                                $\tilde{u}_t := v_t := \lceil -y_t \rfloor$
                                IF $\tilde{u}_t > -y_t$ THEN $\delta_t := -1$
                                        ELSE $\delta_t := 1$
                ELSE IF $\|w_1\|_\infty < \bar{B}$
                        THEN $(u_1, \ldots, u_m) := (\tilde{u}_1, \ldots, \tilde{u}_m)$
                                $\bar{b} := w_1,\ \bar{c} := n\,\|\bar{b}\|_\infty^2,\ \bar{B} := \|\bar{b}\|_\infty$
        ELSE INCREASE_t()
   END while
OUTPUT: $\bar{b}$

Subroutine INCREASE_t()
$t := t+1$
$s := \max(t, s)$
IF $\eta_t = 0$
THEN $\Delta_t := -\Delta_t$
        IF $\Delta_t \delta_t \ge 0$ THEN $\Delta_t := \Delta_t + \delta_t$
ELSE $\Delta_t := \Delta_t + \delta_t$
$\tilde{u}_t := v_t + \Delta_t$

5 Practical algorithm for breaking Orton's Cryptosystem

We use a slightly modified version of ENUM$_\infty$ to find the vector v which transforms into the original message. Since we know that $\|v\|_2^2 = m+1$ and $\|v\|_\infty = 1$, we initialize $\bar{c} := m + 1.0001$, $\bar{B} := 1.0001$ and stop the algorithm as soon as we have found v. In addition to the pruning of lemma 1 and 2 with $(\lambda_t, \ldots, \lambda_m) = (1, 0, \ldots, 0)$ we cut the enumeration for $\tilde{u}_t$ as soon as there is an index $j \in [0, m]$ with $b_{i,j} = 0$ for $i = 1, \ldots, t-1$ and $b_{t,j} \neq 0$, $|w_{t,j}| \neq 1$. We don't miss the solution since $w_{1,j} = w_{t,j} \neq \pm 1$ for all choices of $u_1, \ldots, \tilde{u}_{t-1}$.

Algorithm ATTACK
INPUT: the public key and the encrypted message y_1, y_2
1. build the basis $b_1, \ldots, b_{m+2}$ with $N := n^2$ according to (7)
2. L^3–reduce $b_1, \ldots, b_{m+2}$ with $\delta = 0.99$
3. call ENUM$_\infty$; we get a vector v with $\|v\|_\infty = 1$
4. $x_i := \sum_{l=0}^{s-1} |v_{s(i-1)+l+1} - v_0| 2^{l-1}$ for $i = 1, \ldots, n$

OUTPUT: the original message $x_1, \ldots, x_n$

An ordered lattice basis $b_1, \ldots, b_{m+2}$ is called *L^3–reduced with δ* iff
1. $|\mu_{i,j}| \leq 1/2$ for $1 \leq j < i \leq m+2$
2. $\delta \, \|\hat{b}_{k-1}\|_2^2 \leq \|\hat{b}_k + \mu_{k,k-1}\hat{b}_{k-1}\|_2^2$ for $k = 2, \ldots, m+2$.

The L^3–algorithm of Lenstra, Lenstra, Lovász [LLL82] needs polynomial time to transform a given integer lattice basis into a L^3–reduced basis. We use the floating point version of the L^3–algorithm [SE94]. The resulting basis consists of short and nearly orthogonal lattice vectors. The special structure of the reduced basis makes ENUM$_\infty$ efficient.

The original basis vectors $b_1, \ldots, b_{m+1}$ only depend on the public key. Hence we can precompute the L^3–reduced basis $b'_1, \ldots, b'_{m+1}$ of $b_1, \ldots, b_{m+1}$ once for every public key we want to attack. For all messages which are encrypted with the same public key we use the precomputed vectors $b'_1, \ldots, b'_{m+1}$ together with b_{m+2} instead of the original basis.

Practical Results Table 1 shows the parameters (r, n, s) proposed in [O94] together with the size of the corresponding lattice basis B. The column T indicates the number of operations for the strongest known attacks [B84, S79] as calculated in [O94].

r	n	s	T	size of B
3	200	1	2^{100}	246 × 249
4	3	150	2^{91}	1379 × 1383
4	4	170	2^{104}	1729 × 1733
5	2	150	2^{91}	1534 × 1539
5	2	170	2^{101}	1734 × 1739
5	3	170	2^{104}	1912 × 1917

Table 1: residue knapsack parameters

We randomly generate 10 public keys according to the parameters $(r, n, s) = (3, 200, 1)$. For each of these keys we independently encrypt 10 random messages $(x_1, \ldots, x_{200}) \in \{0,1\}^{200}$. We then reconstruct the messages out of the public key and the ciphertext. Table 2 shows the average as well as the minimal and maximal running time of the algorithms ATTACK, L^3–reduction of $b_1, \ldots, b_{m+1}$ and ATTACK after precomputation. All times are in minutes on a HP 735/99 workstation under HP–UX 9.05 ($< 2^{32}$ operations per minute).

Algorithm	average time	min. time	max. time
ATTACK	10.15	8.69	13.79
L^3–reduction of $b_1, \ldots, b_{m+1}$	9.00	8.52	9.44
ATTACK after precomputation	1.48	0.29	5.23

Table 2: experimental results

First experiments show that we are able to reconstruct the original messages for the other parameters listed in table 1 in less than 30 minutes after a precomputation step which needs less than 12 hours.
We successfully attack three challenges of Orton [O96] with $(r, n, s) = (4, 2, 130)$, $(5, 2, 150)$ and $(5, 2, 170)$.
For all experiments done so far with $s \geq 130$ the L^3–algorithm is sufficient to find the original message. For $s = 1$ the L^3–algorithm doesn't find the original message.

6 Conclusion and Acknowledgement

We break a knapsack Cryptosystem using pruned enumeration of short lattice vectors with respect to the l_∞–norm. These techniques also apply to numerous other problems which can be transformed into a shortest or nearest lattice vector problem in some l_p–norm since Lemma 1 and 2 as well as the enumeration algorithm can easyly be extended to arbitrary l_p–norms. Examples for such problems are hash functions based on knapsack problems, construction of t–designs (shortest lattice vector in l_∞–norm), factoring integers via diophantine approximation (near lattice vectors in l_1–norm), etc.
Schnorr and Hörner successfully attack the Chor–Rivest cryptosystem [CR88] which is also based on knapsacks with density > 1. By our techniques we are able to improve the Schnorr–Hörner attack.

The author wishes to thank Claus P. Schnorr for stimulating this work and for a lot of helpful discussions.

References

[B84] E.F. Brickell: Breaking iterated knapsacks; CRYPTO '84, Springer LNCS, pp. 342–358.

[CJLOSS92] M.J. Coster, A. Joux, B.A. LaMacchia, A.M. Odlyzko, C.P. Schnorr and J. Stern: Improved Low–Density Subset Sum Algorithms; comput. complexity 2, Birkhäuser–Verlag Basel (1992), 111–128.

[CR88] B. Chor and R.L. Rivest: A knapsack–type public key cryptosystem based on arithmetic in finite fields; IEEE Trans. Inform. Theory, vol IT–34 (1988), 901–909.

[K79] L.G. Khachian: A Polynomial Algorithm for Linear Programming; Soviet Math. Doklady 20 (1979), 191–194.

[LLL82] A.K. Lenstra, H.W. Lenstra Jr. and L. Lovász: Factoring polynomials with rational coefficients; Math. Annalen 261, (1982), 515–534.

[LO85] J.C. Lagarias and A.M. Odlyzko: Solving low–density subset sum problems; J. Assoc. Comp. Mach. 32(1) (1985), 229–246.

[MH78] R.C. Merkle and M.E. Hellman: Hiding information and signatures in trapdoor knapsacks; IEEE Trans. Inf. Theory IT–24 (1978), 525–530.

[O94] G. Orton: A Multiple–Iterated Trapdoor for Dense Compact Knapsacks; Advances in Cryptology — EUROCRYPT '94, Springer LNCS (1994), 112–130.

[O96] G. Orton: private communication.

[S79] A. Shamir: On the cryptocomplexity of knapsack systems; Proceedings STOC '79, 118–129.

[S87] C.P. Schnorr: A hierarchy of polynomial time lattice basis reduction algorithms; Theoretical Computer Science 53 (1987), 201–224.

[S94] C.P. Schnorr: Block reduced lattice bases and successive minima; Combinatorics, Probability and Computing 3 (1994), 507–522.

[SE94] C.P. Schnorr and M. Euchner: Lattice Basis Reduction: Improved Practical Algorithms and Solving Subset Sum Problems; Mathematical Programming 66 (1994), 181–199.

[SH95] C.P. Schnorr and H.H. Hörner: Attacking the Chor–Rivest Cryptosystem by Improved Lattice Reduction; Advances in Cryptology — EUROCRYPT '95, Springer LNCS (1995), 1–12.