Electronic Colloquium on Computational Complexity - Reports Series 1995 - available via:

ECCC
TR95-026

FTP: ftp.eccc.uni-trier.de:/pub/eccc/
WWW: http://www.eccc.uni-trier.de/eccc/
Email: ftpmail@ftp.eccc.uni-trier.de with subject 'help eccc'

Attacking the Chor–Rivest Cryptosystem by Improved Lattice Reduction

C.P. Schnorr and H.H. Hörner
Johann Wolfgang Goethe–Universität Frankfurt
Fachbereich Mathematik/Informatik
Postfach 111932, D–60054 Frankfurt a.M., Germany

Abstract. We introduce algorithms for lattice basis reduction that are improvements of the famous L^3-algorithm. If a random L^3–reduced lattice basis $b_1, \ldots, b_n$ is given such that the vector of reduced Gram–Schmidt coefficients $(\{\mu_{i,j}\} \quad 1 \le j < i \le n)$ is uniformly distributed in $[0,1)^{\binom{n}{2}}$, then the pruned enumeration finds with positive probability a shortest lattice vector. We demonstrate the power of these algorithms by solving random subset sum problems of arbitrary density with 74 and 82 many weights, by breaking the Chor–Rivest cryptoscheme in dimensions 103 and 151 and by breaking Damgård's hash function.

1 Introduction and Summary

We address the challenging problem whether it is possible to find, for a given integer lattice basis $b_1, \ldots, b_n \in \mathbf{Z}^m$, in polynomial time a nonzero lattice vector of length $n^{O(1)}\lambda_1$, where λ_1 is the minimal length of nonzero lattice vectors. The L^3–algorithm of Lenstra,Lenstra, Lovász [LLL82] finds in polynomial time a lattice vector of length $2^{\frac{n}{2}}\lambda_1$. Schnorr [S87, S94] has extended this algorithm from block size $\beta = 2$ to arbitrary block sizes $2 \le \beta \le n$. Roughly speaking, this extension goes as follows. Whereas the L^3-algorithm iteratively swaps two consecutive basis vectors b_i, b_{i+1} if this decreases the length of $\widehat{b}_i$, the orthogonal projection of b_i in $\mathrm{span}(b_1, \ldots, b_{i-1})^{\perp}$, block reduction with block size β iteratively transforms blocks $b_i, b_{i+1}, \ldots, b_{i+\beta-1}$ of β consecutive basis vectors as to minimize $\widehat{b}_i$. The first vector of a block reduced basis satisfies $\|b_1\| \le \gamma_\beta^{\frac{n-1}{\beta-1}}\lambda_1$, where $\gamma_\beta \sim \frac{\beta}{\pi e}$ is the Hermite constant of dimension β. For an implementation of block reduction, see the algorithm BKZ of [SE94]. With block size $\beta = 20$ it is only 10 times slower than L^3-reduction but for large block sizes β the delay factor is about $\beta^{O(\beta)}$. This delay factor is the time to construct a shortest vector $\widehat{b}_i$ for a block of size β using complete enumeration of all short lattice vectors. A shortest vector of the entire lattice can be found by the algorithm of Kannan [KA87] in exponential time $n^{O(n)}$.

In this paper we present and analyse a new rule for pruning the enumeration of short lattice vectors. This pruning very likely finds a shortest lattice vector, and is exponentially faster than complete enumeration. It is based on the Gaussian volume heuristic that estimates the number of points of lattice L in nice subsets $S \subset \mathrm{span}(L)$ as $\mathrm{vol}(S)/\det L$. If a random L^3–reduced lattice basis $b_1, \ldots, b_n$ is given such that the vector of reduced Gram–Schmidt coefficients $(\{\mu_{i,j}\} \quad 1 \le j < i \le n)$ is uniformly distributed in $[0,1)^{\binom{n}{2}}$, then the pruned

enumeration finds with positive probability a shortest lattice vector. We let $\{r\}$ denote the residue modulo 1 of the real number r in the interval $[0,1)$.

Pruning the enumeration by the Gaussian volume heuristic is more powerful and more flexible than the previous pruning rule of [SE94]. We combine the new pruning with the block reduction algorithm BKZ of [SE94]. This pruned block reduction is the most powerful lattice reduction algorithm so far. It solves almost all subset sum problems of dimension 74 and 82 for all densities, it breaks the Chor–Rivest cryptosystem in dimensions 103 and 151, and it easily breaks Damgård's knapsack hash function [DA89]. Our experiments raise new hope that almost shortest lattice vectors can be found in polynomial time.

Lagarias and Odlyzko [LO85] have been the first to solve subset sum problems by lattice reduction. Their attack on subset sum problems of low density was improved by [RK88]. Since then the main progress came from block reduction [SE94], [S87], [S94] and by introducing a superior lattice basis [CJLOSS92]. Kaib and Ritter [KR94] propose an alternative approach based on lattice reduction in the l_∞–norm.

2 Basic concepts for efficient lattice reduction

Let $\mathbb{R}^n$ be the m-dimensional real vector space with ordinary inner product $\langle\,,\,\rangle$ and Euclidean length $\|y\| = \langle y,y\rangle^{1/2}$. A discrete, additive subgroup $L \subset \mathbb{R}^m$ is called a *lattice*. Every lattice is generated by some set of linearly independent vectors $b_1,\ldots,b_n \in L$, called a *basis* of L, $L = \{t_1b_1 + \cdots + t_nb_n \mid t_1,\ldots,t_n \in \mathbb{Z}\}$. Let $L(b_1,\ldots,b_n)$ denote the lattice with basis $b_1,\ldots,b_n$. Its *rank* or *dimension* is n and its *determinant* is $\det L = \det[\langle b_i,b_j\rangle_{1\le i,j\le n}]^{1/2}$.

With an ordered lattice basis $b_1,\ldots,b_n \in \mathbb{R}^m$ we associate the Gram-Schmidt orthogonalisation $\widehat{b}_1,\ldots,\widehat{b}_n \in \mathbb{R}^m$ which can be computed together with the Gram-Schmidt coefficients $\mu_{i,j} = \langle b_i,\widehat{b}_j\rangle / \langle \widehat{b}_j,\widehat{b}_j\rangle$ by the recursion $\widehat{b}_1 = b_1$, $\widehat{b}_i = b_i - \sum_{j=1}^{i-1}\mu_{i,j}\widehat{b}_j$ for $i = 2,\ldots,n$. We let π_i denote the orthogonal projection $\pi_i : \mathbb{R}^m \to \mathrm{span}(b_1,\ldots,b_{i-1})^\perp$ for $i = 1,\ldots,n$, $\pi_i(b_j) = \sum_{s=i}^{j}\mu_{s,j}\widehat{b}_s$. Then $\pi_i(L)$ is a lattice of rank $n-i+1$.

An ordered basis $b_1,\ldots,b_n \in \mathbb{R}^m$ is L^3-*reduced*, according to A.K. Lenstra, H.W. Lenstra and L. Lovász [LLL82], with $\delta \in [1/4,1)$ if (1) and (2) hold:

$$|\mu_{i,j}| \le 1/2 \quad \text{for} \quad 1 \le j < i \le n \tag{1}$$

$$\delta \cdot \|\widehat{b}_{k-1}\|^2 \le \|\widehat{b}_k + \mu_{k,k-1}\widehat{b}_{k-1}\|^2 \quad \text{for} \quad k = 2,\ldots,n\ . \tag{2}$$

A basis satisfying (1) is called *size-reduced*. The L^3–algorithm of Lovász [LLL82] transforms an integer lattice basis in polynomial time into an L^3-reduced basis of the same lattice. Schnorr, Euchner [SE94] propose a floating point version $\mathrm{L}^3\mathrm{FP}$ of the L^3–algorithm. This algorithm is used whenever we apply L^3–reduction.

A lattice basis $b_1,\ldots,b_n$ is *block reduced* with block size β if it is size reduced and if $\widehat{b}_i$, for $i = 1,\ldots,n$, is the shortest nonzero vector of the lattice $\pi_i L(b_i,\ldots,b_{\min(i+\beta-1,n)})$. Block reduction has been analysed in [S87], [S94].

We consider the following function c_t with integer entries $u_t, \ldots, u_n$

$$c_t(u_t, \ldots, u_n) := \|\pi_t \sum_{i=t}^{n} u_i b_i\|^2 = \sum_{j=t}^{n} \left(\sum_{i=j}^{n} u_i \mu_{i,j} \right)^2 \|\widehat{b}_j\|^2 \text{ for } t = 1, \ldots, n\,.$$

We present the core of the procedure ENUM of [SE94] that generates a shortest lattice vector by complete enumeration in depth first order.

Algorithm ENUM
INPUT $\|\widehat{b}_i\|^2,\ \mu_{i,t}$ for $1 \le t \le i \le n$.
OUTPUT a minimal nonzero place $(u_1, \ldots, u_n)$ and a minimal value $\bar{c}_1$ for the function c_1.

1. FOR $i = 1, \ldots, n$ DO $\widetilde{c}_i := u_i := \widetilde{u}_i := y_i := 0$
   $\widetilde{u}_1 := u_1 := 1\ ,\quad t := 1\ ,\quad \bar{c}_1 := \widetilde{c}_1 := \|\widehat{b}_1\|^2\ .$
   (we always have $\widetilde{c}_t = c_t\ (\widetilde{u}_t, \ldots, \widetilde{u}_n)$, $\bar{c}_1$ is the current minimum of c_1)
2. WHILE $t \le n$
   $\widetilde{c}_t := \widetilde{c}_{t+1} + (y_t + \widetilde{u}_t)^2\ \|\widehat{b}_t\|^2$
   IF $\widetilde{c}_t < \bar{c}_1$
   THEN IF $t > 1$
   THEN $t := t - 1\ ,\quad y_t := \sum_{i=t+1}^{t_{max}} \widetilde{u}_i \mu_{i,t}\ ,\quad \widetilde{u}_t := \lceil -y_t \rfloor$
   ELSE $\bar{c}_1 := \widetilde{c}_1\ ,\quad u_i := \widetilde{u}_i$ for $i = 1, \ldots, n$
   ELSE $t := t + 1$
   $$\widetilde{u}_t := \begin{cases} \widetilde{u}_t + 1 & \text{if } t = t_{max} \\ \text{next}(\widetilde{u}_t, -y_t) & \text{otherwise} \end{cases}$$
   END while

Here $\lceil r \rfloor \stackrel{def}{=} \lceil r - 0.5 \rceil$, t_{max} is the maximal previous value of t. We define $a' = \text{next}(a, r)$ to be the integer which is, next to $a \in \mathbb{Z}$, nearest to $r \in \mathbb{R}$. We have $|a - r| \le |a' - r| \le |a - r| + 1$, $sign(a' - r) \ne sign(a - r)$, $|a - r| = |a' - r| \Rightarrow a < r < a'$.

Correctness. The algorithm ENUM enumerates in depth first order all nonzero integer vectors $(\widetilde{u}_t, \ldots, \widetilde{u}_n)$ for $t = 1, \ldots, n$ that satisfy $c_t(\widetilde{u}_t, \ldots, \widetilde{u}_n) < \bar{c}_1$ where $\bar{c}_1$ is the actual minimal value for the function c_1. All enumerated vectors satisfy $\widetilde{u}_i > 0$ for the largest i with $\widetilde{u}_i \ne 0$. For fixed $\widetilde{u}_{t+1}, \ldots, \widetilde{u}_n$, the sequence of values $\widetilde{u}_t$, generated by iterating the function $\text{next}(*, -y_t)$, makes the sequence $c_t(\widetilde{u}_t, \ldots, \widetilde{u}_n)$ non decreasing. Therefore, if the test $\widetilde{c}_t < \bar{c}_1$ fails for the current vector $(\widetilde{u}_t, \ldots, \widetilde{u}_n)$, the subsequent increment of stage t has the effect to *discard* all vectors $(u, \widetilde{u}_{t+1}, \ldots, \widetilde{u}_n)$ where $\widetilde{u}_t$ preceeds u in the iteration of $\text{next}(*, -y_t)$. The discarded vectors can not lead to the minimum of the function c_1.

3 Pruning the enumeration

We prune the enumeration of vectors $(\widetilde{u}_t, \ldots, \widetilde{u}_n)$ in ENUM by tightening up the test "IF $\widetilde{c}_t < \bar{c}_1$". We cut off the depth first search at $(\widetilde{u}_t, \ldots, \widetilde{u}_n)$ if the

probability that $(\widetilde{u}_t, \ldots, \widetilde{u}_n)$ can be completed as to satisfy $c_1(\widetilde{u}_1, \ldots, \widetilde{u}_n) < \bar{c}_1$ is less than a chosen threshold 2^{-p}.

The Gaussian volume heuristic. A general principle, dating back to Gauss, estimates the number of points of lattice L in nice subsets $S \subset \mathrm{span}(L)$ as $\mathrm{vol}(S)/\det L$.

How to apply it. Suppose we have chosen integers $\widetilde{u}_t, \ldots, \widetilde{u}_n$ and we search for $\widetilde{u}_1, \ldots, \widetilde{u}_{t-1}$ as to satisfy $c_1(\widetilde{u}_1, \ldots, \widetilde{u}_n) < \bar{c}_1$. We let $\bar{L}$ denote the lattice $\bar{L} = L(b_1, \ldots, b_{t-1})$. So we want to add to the given lattice vector $b = \sum_{i=t}^{n} \widetilde{u}_i b_i$ a vector $\bar{b} = \sum_{i=1}^{t-1} \widetilde{u}_i b_i$ in $\bar{L}$ as to satisfy $\|b + \bar{b}\|^2 < \bar{c}_1$. We decompose b into orthogonal parts $b = y - z$ with $z = -\sum_{j=1}^{t-1} \sum_{i=t}^{n} \widetilde{u}_i \mu_{i,j} \widehat{b}_j \in \mathrm{span}(\bar{L}), y \in \mathrm{span}(\bar{L})^{\perp}$, $\widetilde{c}_t = \|y\|^2$. This means, we search for a point in

$$(b + \bar{L}) \cap S(\sqrt{\bar{c}_1 - \widetilde{c}_t}, y) \;=\; \bar{L} \cap S(\sqrt{\bar{c}_1 - \widetilde{c}_t}, z)$$

where $S(r, y)$ is the $(t-1)$-dimensional sphere with radius r and center y in $y + \mathrm{span}(\bar{L})$. Here the equality holds since $z = y - b$. Now we apply the volume heuristic to the lattice $\bar{L}$ and the sphere $S(\sqrt{\bar{c}_1 - \widetilde{c}_t}, z) \subset \mathrm{span}(\bar{L})$. Hence the expected number of vectors $(\widetilde{u}_1, \ldots, \widetilde{u}_{t-1}) \in \mathbb{Z}^{t-1}$ satisfying $c_1(\widetilde{u}_1, \ldots, \widetilde{u}_n) \leq \bar{c}_1$ is $\mathrm{vol}\, S(\sqrt{\bar{c}_1 - \widetilde{c}_t}, z)/\det \bar{L}$. We propose to cut off the enumeration of $(\widetilde{u}_1, \ldots, \widetilde{u}_{t-1})$ if this ratio is less than 2^{-p} for a fixed chosen p.

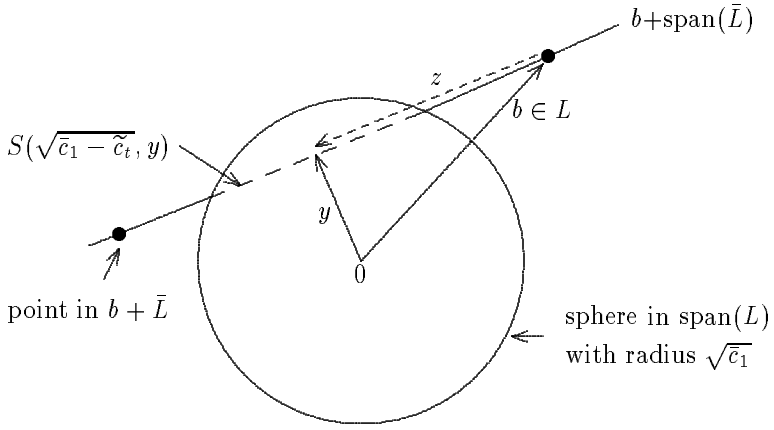

Figure: the volume heuristic

GAUSS–ENUM. We replace in ENUM the condition "IF $\tilde{c}_t < \bar{c}_1$" by "IF vol $S(\sqrt{\bar{c}_1 - \tilde{c}_t}, z)/\det \bar{L} < 2^{-p}$". We call the new procedure GAUSS–ENUM. The parameter p controls the pruning. Large values p correspond to weak pruning, $p = \infty$ corresponds to complete enumeration (no pruning). The inequality vol $S(\sqrt{\bar{c}_1 - \tilde{c}_t}, z)/\det \bar{L} < 2^{-p}$ is equivalent to $\tilde{c}_t < \bar{c}_1 - \eta$ where

$$\eta \;=\; \frac{1}{\pi}\left(\frac{t-1}{2}\right)!^{\frac{2}{t-1}} \left(2^{-p}\prod_{i=1}^{t-1}\|\hat{b}_i\|\right)^{\frac{2}{t-1}} \quad .$$

If GAUSS–ENUM cuts off the depth first search at $(\tilde{u}_t, \ldots, \tilde{u}_n)$ the probability, that $(\tilde{u}_t, \ldots, \tilde{u}_n)$ can be completed as to satisfy $c_1(\tilde{u}_1, \ldots, \tilde{u}_n) < \bar{c}_1$, is at most 2^{-p}. In the analysis of GAUSS–ENUM we disregard that GAUSS–ENUM discards, in addition to the vectors $(\tilde{u}_1, \ldots, \tilde{u}_n)$, also the vectors $(\tilde{u}_1, \ldots, \tilde{u}_{t-1}, u, \ldots, \tilde{u}_n)$ where $\tilde{u}_t$ precedes u in the iteration of next$(*, -y_t)$. This can be repaired by a slight change in GAUSS–ENUM. However this yields a reduction algorithm that is less efficient in practice.

Justification of the volume heuristic. The Gaussian principle does not hold in general. Mazo and Odlyzko [MO90] show that it fails even in the case of spheres and the lattice $L = \mathbb{Z}^n$ for particular choices of the center z. However the principle holds if the center of the sphere is "uniformly distributed *(u.d.)* modulo the lattice".

Definition. *For a lattice L with basis $b_1, \ldots, b_n$ a probability distribution of points $\sum_{i=1}^n t_i b_i$ in* span(L) *is called u.d. modulo L if the reduced vector $(\{t_i\}\ i = 1, \ldots, n)$ is u.d. in $[0,1)^n$.*

This notion does not depend on the choice of the basis. If $b_1, \ldots, b_n$ and $\bar{b}_1, \ldots, \bar{b}_n$ are two bases of lattice L there is a matrix $U \in \mathrm{GL}_n(\mathbb{Z})$ satisfying $[\bar{b}_1, \ldots, \bar{b}_n] = [b_1, \ldots, b_n]\, U$. Since $|\det U| = 1$ the linear transformation by U transforms the uniform distribution on $\sum_{i=1}^n b_i\ [0,1)$ into the uniform distribution on $\sum_{i=1}^n \bar{b}_i\ [0,1)$. Alternatively we can express the uniformity modulo L in terms of the Gram-Schmidt orthogonalization $\hat{b}_1, \ldots, \hat{b}_n$ associated with the basis $b_1, \ldots, b_n$. The vector $\sum_{i=1}^n t'_i\ \hat{b}_i$ in span(L) is u.d. modulo L if and only if the vector $(\{t'_i\}\ i = 1, \ldots, n)$ is u.d. in $[0,1)^n$.

Lemma 1. *Let L be a lattice and $S(r,z) \subset$ span(L) the sphere with fixed radius r and random center z that is u.d. modulo L. Then $E_z \#(S(r,z) \cap L) =$ vol $S(r,z)/\det L$ holds for the expectation E_z.*

Proof. For two points $z, \bar{z} \in$ span$\,(L)$ that coincide modulo L, i.e. $z = \bar{z} \bmod L$, we have $\#(S(r,z) \cap L) = \#(S(r,\bar{z}) \cap L)$. The average number of lattice points in $S(r,z)$ is the average number of lattice points per volume part vol $S(r,z)$. Hence the expected value of $\#(S(r,z) \cap L)$ is vol $S(r,z)/\det L$. □

We apply Lemma 1 to the situation in GAUSS-ENUM with $\widetilde{u}_t, \ldots, \widetilde{u}_n$ being fixed, $\widetilde{c}_t = c_t(\widetilde{u}_t, \ldots, \widetilde{u}_n)$, $\bar{c}_1 > \widetilde{c}_t$ and a lattice point of $\bar{L}$ is searched in the sphere $S(\sqrt{\bar{c}_1 - \widetilde{c}_t}, z)$ with center $z = -\sum_{j=1}^{t-1} \sum_{i=t}^{n} \widetilde{u}_i \mu_{i,j} \widehat{b}_j$.

Theorem 2. *If the vector* $(\{\mu_{i,j}\}\ 1 \le j < i \le n)$ *is u.d. in* $[0,1)^{\binom{n}{2}}$ *then for every fixed nonzero* $(\widetilde{u}_t, \ldots, \widetilde{u}_n) \in \mathbb{Z}^{n-t+1}$ *the center* z *is u.d. modulo the lattice* $\bar{L} = L(b_1, \ldots, b_{t-1})$. *Moreover*

$$E_z \ \#[\ (\widetilde{u}_1, \ldots, \widetilde{u}_{t-1}) \in \mathbb{Z}^{t-1} : c_1(\widetilde{u}_1, \ldots, \widetilde{u}_n) \le \bar{c}_1] = \text{vol}\ S(\sqrt{\bar{c}_1 - \widetilde{c}_t}, z)/\det \bar{L}.$$

Proof. We can assume that $\widetilde{u}_n \neq 0$ since otherwise we can decrease n. We see that the vectors $(\{\widetilde{u}_n \mu_{n,j}\}\ j = 1, \ldots, t-1)$ and $(\{\sum_{i=t}^{n} \widetilde{u}_i \mu_{i,j}\}\ j = 1, \ldots, t-1)$ are u.d., in $[0,1)^{t-1}$. This shows that z is u.d. modulo $\bar{L}$. Since

$$\#[(\widetilde{u}_1, \ldots, \widetilde{u}_{t-1}) \in \mathbb{Z}^{t-1} : c_1(\widetilde{u}_1, \ldots, \widetilde{u}_n) \le \bar{c}_1] = \#(S(\sqrt{\bar{c}_1 - \widetilde{c}_t}, z) \cap \bar{L})$$

the expression for the expectation E_z follows from Lemma 1. □

Success rate of GAUSS–ENUM. Suppose a distribution of L^3–reduced lattice bases so that the vector $(\{\mu_{i,j}\}\ \ 1 \le j < i \le n)$ is u.d. in $[0,1)^{\binom{n}{2}}$ and let $p > \log_2 n$. Whenever the depth first search is cut off at a fixed vector $(\widetilde{u}_t, \ldots, \widetilde{u}_n) \in \mathbb{Z}^{n-t+1}$ then, by theorem 2, the event that a lattice vector shorter than $\sqrt{\bar{c}_1}$ gets lost, has probability at most 2^{-p}. Therefore the probability of missing the shortest lattice vector is at most 2^{-p} times the average number of cutoffs. While the number of cutoffs can be arbitrarily large for badly reduced bases statistical experiments show that, for random L^3–reduced basis, the average number of cutoffs is proportional to $c_{p,n} 2^p$ where the factor $c_{p,n}$ decreases to 0 as p increases. E.g. for $n < 30$ and $p = 7$ the probability of success is at least 0.1 .

4 Solving subset sum problems

Given positive integers $a_1, \ldots, a_n, s$ we wish to solve the equation $\sum_{i=1}^{n} a_i x_i = s$ with $x_1, \ldots, x_n \in \{0,1\}$. We assume that we are also given $q = \sum_{i=1}^{n} x_i$, the number of 1–entries of the solution. So we search for a $\{0,1\}$–solution $(x_1, \ldots, x_n)$ of the two equations $\sum_{i=1}^{n} a_i x_i = s$, $\sum_{i=1}^{n} x_i = q$. Following [CJLOSS92] we associate to this problem the following lattice basis $b_0, \ldots, b_n \in \mathbb{Z}^{n+3}$

$$\begin{array}{rcl} b_0 &=& (\ 1,\ q,\ q,\ \ldots\ q,\ \ n^2 s,\ \ n^2 q\ ) \\ b_1 &=& (\ 0,\ n,\ 0,\ \ldots\ 0,\ n^2 a_1,\ \ n^2\ ) \\ b_2 &=& (\ 0,\ 0,\ n,\ \ldots\ 0,\ n^2 a_2,\ \ n^2\ ) \\ \vdots &=& \ \vdots\ \ \vdots\ \ \vdots\ \ \ddots\ \vdots\ \ \ \vdots\ \ \ \ \vdots \\ b_n &=& (\ 0,\ 0,\ 0,\ \ldots\ n,\ n^2 a_n,\ \ n^2\ ) \end{array} \tag{3}$$

According to [CJLOSS92] the shortest vector z of the lattice $L(b_0, \ldots, b_n)$ solves via (4) almost all subset sum problems of density less then 0.9408, where the density is $n/\max_i \log_2 a_i$. Even beyond this density threshold, solutions of the problems in this paper are associated with very short lattice vectors.

With a $\{0,1\}$–solution $x = (x_1, \ldots, x_n)$ of the subset sum problem we associate lattice vectors $z = (z_0, \ldots, z_{n+2}) = \pm(-b_0 + \sum_{i=1}^{n} x_i b_i)$ that satisfy $|z_0| = 1, z_{n+1} = z_{n+2} = 0,\ z_i/z_0 \in \{q, q-n\}$ for $i = 1, \ldots, n$. Conversely every such lattice vector $z = (z_0, \ldots, z_{n+2})$ induces a subset sum solution

$$(4) \qquad x_i := [\text{ IF } z_i/z_0 = q - n \text{ THEN } 1 \text{ ELSE } 0\,] \quad \text{for} \quad i = 1, \ldots, n$$

We have tested the following algorithm for general subset sum problems with $n = 74$ and $n = 82$ many weights and for the Chor–Rivest subset sum problem with $n = 103$.

Algorithm PRUNED SUBSET SUM
INPUT lattice basis $b_0, \ldots, b_n \in \mathbb{Z}^{n+3}$ as in (3).
Perform four successive stages of reduction :

1. L^3–reduction.

2. block reduction with block size 20.

3. pruned block reduction with block size 50 and $p = 10$.

4. pruned block reduction with block size 70 and $p = 12$.

Algorithmic details. 1. For L^3–reduction we use the algorithm L^3FP of [SE94]. We set $\delta = 0.99$, we apply the deep insertion rule of [SE94] for the first basis vector.

2. Block reduction is done by the algorithm BKZ of [SE94] with $\delta = 0.99$ resulting in a basis $b_0, \ldots, b_n$ satisfying for $i = 0, \ldots, n$

$$(5) \qquad 0.99\, \|\widehat{b}_i\| \le \|\pi_i(b)\| \text{ for all nonzero } b \in L(b_i, \ldots, b_{\min(i+\beta-1,n)}) .$$

3. Pruned block reduction is done the same way as block reduction except that we use instead of algorithm ENUM the algorithm GAUSS–ENUM with an appropriate pruning parameter p. The resulting basis may occasionally fail the inequalities (5).

4. Test for solution and early termination. Subsequent to every size–reduction of a basis vector b_j it is always tested whether b_j solves the subset sum problem, i.e. whether (4) induces a solution x for $z = b_j$. Also for each stage of the reduction, the vectors of the reduced basis are tested for solution. The algorithm terminates as soon as a solution has been found.

5. Reduction to the sublattice $\widetilde{L} = \{(z_0, \ldots, z_{n+2}) \in L(b_0, \ldots, b_n) \ : \ z_{n+1} = z_{n+2} = 0\}$. After the L^3–reduction in stage 1 we construct a basis of the lattice $\widetilde{L}$ and we continue the reduction process with this basis. Working with the lattice

$\widetilde{L}$ simplifies subsequent reductions since $rank(\widetilde{L}) = rank(L) - 2$. To construct a basis of $\widetilde{L}$ we linearly transform the L^3–reduced basis $b_0, \ldots, b_n$ of L so that $b_{i,j} = 0$ holds for $i = 1, \ldots, n-2$ and $j = n+1, n+2$. Then we eliminate the vectors b_{n-1}, b_n from the basis and we remove from the vectors b_i $i = 0, \ldots, n-2$ the last two coordinates $b_{i,n+1}, b_{i,n+2}$. Upon entry of stage 2 we randomly permute the basis so that it starts with the vectors b_i that have a nonzero coordinate $b_{i,0}$. This enhances the generation of short lattice vectors z which induce via (4) a subset sum solution.

5 Attacks on the Chor–Rivest cryptosystem

Chor, Rivest present a public key encryption method for which deciphering has the form of a subset sum problem of high density, for details see [CR88]. Chor, Rivest propose examples of their scheme with $n = 197$ and $n = 211$ many weights. For testing possible attacks they also designed a small example with $n = 103$ many weights and subset sum problems of density 1.271. The Lagarias–Odlyzko method which is based on L^3–reduction completely failed for the $n = 103$ subset sum problems.

Interestingly, block reduction with pruned enumeration solves the Chor–Rivest subset sum problems with $n = 103$ many weights in only 1.5 hours average time with 42% success rate. Thus the widespread believe that subset sum problems with density greater than 1 cannot be solved via lattice reduction is outright wrong. The Chor–Rivest scheme with $n = 103$ and density 1.271 is even less difficult than random subset sum problems with $n = 82$ and density 1.

Generation of the Chor–Rivest subset sum problems. We take the particular weights $a_1, \ldots, a_{103}$ of the example constructed by Chor, Rivest. We generate 50 random vectors $(x_1, \ldots, x_{103}) \in \{0,1\}^{103}$ so that $\sum_{i=1}^{103} x_i = 12$, and we set $s := \sum_{i=1}^{103} x_i a_i$. In the corresponding subset sum problem we are given $a_1, \ldots, a_{103}, s$ and have to solve the equations $\sum_{i=1}^{103} x_i a_i = s$, $\sum_{i=1}^{103} x_i = 12$ with $x_1, \ldots, x_{103} \in \{0,1\}$. (The number 12 arises from the particular construction of the weights a_i starting from the field $\mathbb{F} = GF(103^{12})$, a generator g for the group of units $\mathbb{F}^\star$, an element $t \in \mathbb{F}$ that is algebraic of degree 12 over $GF(103)$, a random permutation π in $Sym(n)$ and a random number d with $0 \le d < 103^{12} - 2$, and setting $a_i := \log_g(t + \pi(i)) + d$ for $i = 1, \ldots, 103$.) We solve these 50 subset sum problems by applying the algorithm PRUNED SUBSET SUM to the lattice basis (3) with $n = 103$, $q = 12$.

The first table shows, for each of the stages $i = 1, 2, 3, 4$, in column 4 the number of successes on stage i, in column 5 the number of successes up to stage i, in column 6 the average time (with respect to all 50 problems) of stage i, in column 7 the total time up to stage i and in column 8 the maximal time of stage i. The last column contains the total time for all 50 problems divided by the number of successes. All times are in minutes for a HP 715/50 workstation under HP–UX 9.05 .

stage	block size	p	# successes on stage	# successes up to stage	time in minutes average	time in minutes av. total	time in minutes maximal	total time per success
1	2	∞	0	0	0.6	0.6	0.7	∞
2	20	∞	3	3	7.6	8.2	16.9	163.5
3	50	10	18	21	86.8	95.0	247.3	226.1
4	70	12	14	35	173.4	268.4	938.3	383.5

Stage 1 which performs L^3–reduction does not find any solution. This confirms the previous results of Odlyzko showing that L^3–reduction is too weak even if the CJLOSS basis (3) is used which is much stronger than the Lagarias–Odlyzko basis used in the experiments of Odlyzko.

Stage 4 by itself is quite inefficient. It takes a total of 619 minutes per success. This suggests to replace stage 4 by a repetition of stages 1,2,3 with a randomly permuted input basis. The next table shows the results for two repetitions of stages 1,2,3.

	# successes in round	# successes total	time in minutes average	time in minutes av. total	total time per success
stages 1,2,3	21	21	95.0	95.0	226.1
1. repetition	11	32	65.3	160.3	250.5
2. repetition	6	38	33.5	193.8	255.0

With two repetitions the success rate is 76% with an average time of 3.2 hours. It may be of interest that an alternative algorithm of Ritter, see [KR94], solves all $n = 103$ Chor–Rivest problems in about 7 hours maximal time.

Chor–Rivest subset sum problems with more weights. A limited number of first experiments have been carried out by H.H. Hörner in attacking a Chor–Rivest cryptosystem with $n = 151$ many weights and $q = 16$ [H94]. So far he could solve 5 out of 50 random problems with an average time of 195 hours for the solved problems.

6 Attacks on Damgård's knapsack hash function

In [DA89] a hash function h his proposed based on the subset sum problem. Choose random numbers $a_1, \ldots, a_{256}$ in the interval $[1, 2^{120} - 1]$ and hash a message m consisting of the bits $m_1, \ldots, m_{256}$ into the integer $h(m_1, \ldots, m_{256}) = \sum_{i=1}^{256} a_i m_i$.

To construct a collision for h it is sufficient to find a nonzero $\{\pm 1, 0\}$–solution $(x_1, \ldots, x_{256})$ of the equation $\sum_{i=1}^{256} a_i x_i = 0$. This yields messages m, m' with bits $m_i = \max\{0, x_i\}$, $m'_i = -\min\{x_i, 0\}$ for $i = 1, \ldots, 256$ satisfying $h(m) = h(m')$.

Following an analysis of Joux, Stern [JS94] collisions exist almost surely even for the restricted problem with 80 out of the 256 weights a_i. We construct nonzero $\{\pm 1, 0\}$–solutions of the equation $\sum_{i=1}^{100} a_i x_i = 0$. We associate to this problem

the following lattice basis $b_1, \ldots, b_n \in \mathbb{Z}^{n+1}$ with $b_i = (0, \ldots, 1^{(i)}, \ldots, 0, na_i)$ for $i = 1, \ldots, n$ and $n = 100$.

A nonzero lattice vector $z = (z_1, \ldots, z_{n+1})$ yields a collision if $z_{n+1} = 0$ and $(z_1, \ldots, z_n) \in \{\pm 1, 0\}^n$. We apply to this basis a two–stage reduction consisting of an L^3–reduction and a single pruned block reduction with block size 50 and alternative p–values $8, 9, \ldots, 12$. We test after each size–reduction whether the reduced vector z yields a collision. (The more powerful reduction algorithm PRUNED SUBSET SUM is less efficient since the shortest lattice vector is most likely not in $\{\pm 1, 0\}^n$. This follows from the analysis in [JS94].)

Each row in the following table corresponds to 20 random vectors $(a_1, \ldots, a_{100}) \in \left[1, 2^{120} - 1\right)^{100}$. We report the number of successes, the average running time in minutes, the minimal and maximal size of the detected collision (the size of a collision $(x_1, \ldots, x_n) \in \{\pm 1, 0\}^n$ is $\#\{i \ : \ x_i \neq 0\}$), and the pruning parameter p.

block size	p	# successes	av. time in minutes	min size	max size	total time per success
50	8	7	235.04	48	58	671.54
50	9	16	261.98	44	62	327.48
50	10	16	365.84	45	59	457.30
50	11	19	388.05	37	61	408.47
50	12	20	386.65	44	60	386.65

A first collision for Damgård's hash function has been constructed in [JG94] using pruned block reduction via the pruning of [SE94]. They report one success for ten problems. The new results demonstrate the superiority of pruning via the volume heuristic.

We can further improve the performance our attack on Damgård's hash function. If the pruning with parameter p decides to cut off the enumeration we repeat the cut off test with $p + 1$. If $p + 1$ does not cut off we perform a reduction in size on the current vector $\sum_{i=t}^{n} \tilde{u}_i b_i$. This yields a relatively short lattice vector $\sum_{i=1}^{n} \tilde{u}'_i \widehat{b}_i$ with $|\tilde{u}'_i| \leq 1/2$. We test if the reduced vector yields a collision. To save some time we cut off the reduction in size if the vector $\sum_{i=t}^{n} \tilde{u}'_i \widehat{b}_i$ becomes to long, e.g. longer than 60 in the following examples. The improved attack yields the following performances:

block size	p	# successes	av. time in minutes	min size	max size	total time per success
50	6	6	77.10	43	59	257.00
50	7	10	128.71	39	58	257.42
50	8	19	121.50	45	59	127.90
50	9	20	127.70	43	60	127.70
50	10	20	115.24	44	59	115.24

7 General subset sum problems

We report on solving random subset sum problems of arbitrary density in dimensions $n = 74$ and 82. The previously most powerful algorithm [SE94] could solve almost all problems in dimension $n = 66$ by combining block reduction with some sort of pruning. The new algorithm PRUNED SUBSETSUM prunes the enumeration of short lattice vectors by the volume heuristic. It solves for $n = 74, 82$ a substantial fraction of all random subset sum problems of arbitrary density.

In the following table, every row with entries n, b corresponds to 20 random input bases (3) that are generated as follows. Pick random integers $a_1, \ldots, a_n$ in the interval $[1, 2^b]$, pick a random subset $I \subset \{1, \ldots, n\}$ of size $n/2$ and put $s = \sum_{i \in I} a_i$. To solve the corresponding subset sum problem $\sum_{i=1}^{n} a_i x_i = s$ we apply the algorithm PRUNED SUBSET SUM to the lattice basis (3) with $q = n/2$. The numbers in columns S, S1, S2, S3, S4 denote the total number of successes, and the number of successes in stages 1, 2, 3, 4.

		# successes					average time in minutes				
n	b	S	S1	S2	S3	S4	stage 1	stage 2	stage 3	stage 4	total
74	26	20	19	1	0	0	0.08	0.00			0.08
74	34	20	5	13	2	0	0.13	0.23	0.05		0.40
74	42	19	1	4	13	1	0.15	1.05	0.62	0.07	1.88
74	50	17	0	0	11	6	0.20	1.43	4.08	2.75	8.47
74	58	15	0	0	4	11	0.23	2.12	10.98	14.67	28.00
74	66	12	0	0	6	6	0.28	2.72	21.82	27.67	52.48
74	74	12	0	0	6	6	0.32	3.68	27.28	31.72	63.00
74	82	20	0	0	19	1	0.38	5:70	11.42	0.37	17.87
74	90	20	0	8	12	0	0.45	4.07	2.17		6.68
74	98	20	0	15	5	0	0.55	3.00	0.38		3.93
82	34	20	2	16	2	0	0.17	0.37	0.05		0.06
82	42	20	0	7	13	0	0.20	1.30	0.88		2.38
82	50	18	0	1	13	4	0.27	1.77	4.78	1.32	8.13
82	58	8	0	0	4	4	0.28	2.85	10.25	26.65	40.05
82	66	5	0	0	0	5	0.33	3.20	30.25	65.93	99.73
82	74	4	0	0	1	3	0.37	3.73	60.67	171.37	236.13
82	82	5	0	0	1	4	0.45	4.73	104.87	172.53	282.06
82	90	14	0	0	9	5	0.53	6.00	60.95	73.72	141.20
82	98	20	0	0	17	3	0.61	7.55	33.90	7.65	49.72
82	106	20	0	3	17	0	0.68	9.87	9.28		19.83

PRUNED SUBSET SUM is remarkably efficient for densities less than 0.9408 where the shortest lattice vector most likely yields a solution, see lines $n = 74, b \geq 82$ and $n = 82, b \geq 90$. This gives new hope that shortest, or near shortest lattice vectors can be found in polynomial time.

Random subset sum problems with $n = 82$ and density 1 are harder than the Chor–Rivest scheme with $n = 103$ and density 1.271. Here stage 4 of the

algorithm PRUNED SUBSET SUM is necessary for the generation of solutions. Only 5 out of 20 problems for $n = 82, b = 82$ are solved in 282 minutes. The Chor–Rivest problems are easier because the problem solution yields a shortest lattice vector with no further vector being nearly as short.

References

[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.

[DA89] I. B. Damgård: A Design Principle for Hash Functions; Advances in Cryptology, Proc. Crypto 89, Springer LNCS 435 (1990), 416–427.

[H94] H.H. Hörner: Verbesserte Gitterbasenreduktion; getestet am Chor–Rivest Kryptosystem und an allgemeinen Rucksack–Problemen. Diplomarbeit, Universität Frankfurt (August 1994).

[JG94] A. Joux and L. Granboulan: A Practical Attack against Knapsack based Hash Functions; Prodeedings EUROCRYPT'94, Springer LNCS (1994).

[JS94] A. Joux and J. Stern: Lattice Reduction: a Toolbox for the Cryptanalyst, TR DGA/CELAR, ENS (1994).

[KA87] R. Kannan: Minkowski's convex body theorem and integer programming; Math. Oper. Res. 12 (1987), 415–440.

[KR94] M. Kaib and H. Ritter: Block Reduction with Respect to Arbitrary Norms; TR U. Frankfurt (1994).

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

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

[MO90] J.E. Mazo and A.M. Odlyzko: Lattice Points in high–dimensional spheres; Monatsh. Math. 110 (1990), 47–61.

[RK88] S. Radziszowski and D. Kreher: Solving subset sum problems with the L^3 algorithm; J. Combin. Math. Combin. Comput. 3 (1988), 49–63.

[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.