Recursive Lattice Reduction[*]

Thomas Plantard and Willy Susilo

Centre for Computer and Information Security Research
School of Computer Science and Software Engineering
University of Wollongong, Australia
{thomaspl,wsusilo}@uow.edu.au

Abstract. Lattice reduction is known to be a very powerful tool in modern cryptanalysis. In the literature, there are many lattice reduction algorithms that have been proposed with various time complexity (from quadratic to subexponential). These algorithms can be utilized to find a short vector of a lattice with a small norm. Over time, shorter vector will be found by incorporating these methods. In this paper, we take a different approach by presenting a methodology that can be applied to any lattice reduction algorithms, with the implication that enables us to find a shorter vector (i.e. a smaller solution) while requiring shorter computation time. Instead of applying a lattice reduction algorithm to a complete lattice, we work on a sublattice with a smaller dimension chosen in the function of the lattice reduction algorithm that is being used. This way, the lattice reduction algorithm will be fully utilized and hence, it will produce a better solution. Furthermore, as the dimension of the lattice becomes smaller, the time complexity will be better. Hence, our methodology provides us with a new direction to build a lattice that is resistant to lattice reduction attacks. Moreover, based on this methodology, we also propose a recursive method for producing an optimal approach for lattice reduction with optimal computational time, regardless of the lattice reduction algorithm used. We evaluate our technique by applying it to break the lattice challenge[1] by producing the shortest vector known so far. Our results outperform the existing known results and hence, our results achieve the record in the lattice challenge problem.

Keywords: Geometry of numbers, Lattice reduction, Hermite factor, Recursive reduction.

1 Introduction

Lattice reduction algorithms have been proposed to solve or approximate the shortest vector problem. In the literature, it has been demonstrated that many cryptosystems can be cryptanalyzed successfully using lattice reduction algorithms. Some of the historical examples include the following[2].

[*] This work is partially supported by The Department of Prime Minister and Cabinet's Research Support for Counter-Terrorism: PR06-0006.

[1] http://latticechallenge.org/

[2] We will refer to [51] for a more specific survey.

J.A. Garay and R. De Prisco (Eds.): SCN 2010, LNCS 6280, pp. 329–344, 2010.

Knapsack Cryptosystems
In 1978, Merkle and Hellman [43] proposed the first public key cryptosystem based on an NP-hard problem, namely the knapsack problem. This is the first practical public key cryptosystem which is the concrete construction of the proposed seminal notion of public key cryptography by Diffie and Hellman [26]. Unfortunately, Merkle-Hellman's first proposition was attacked severely and broken using two different methods: the first attack on the trapdoor itself that was proposed by Shamir [67,68] and the second attack on the knapsack problem using lattice theory that was proposed by Adleman [2]. In 1985, Lagarias and Odlyzko [39] proposed a general attack against knapsack cryptosystems using a lattice reduction algorithm. Their attack does not incorporate the weakness on the trapdoor itself, rather than only using the fact that the knapsack problems produced are generally weaker that a random one. This result was subsequently improved in [25,24,66,58]. Nevertheless, some improvements of knapsack cryptosystems were also proposed (e.g. [16,57]). We refer the reader to [56] for these two faces of knapsack cryptology. Despite many variants that have been proposed in the literature, as mentioned in [55], the knapsack cryptosystem proposed by Okamoto, Tanaka and Uchiyama in 2000 [57] seems to be the only remaining secure knapsack cryptosystem.

SVP-based Cryptosystems
In 1996, Ajtai and Dwork [8] proposed the first lattice cryptosystem where its security is based on a variant of the Shortest Vector Problem (SVP). This cryptosystem received wide attention due to a surprising security proof based on worst-case assumptions. Nonetheless, this cryptosystem is merely a theoretical proposition and it cannot be used in practice. Furthermore, Nguyen and Stern presented a heuristic attack against this cryptosystem [54]. Until then, this initial proposition has been improved [29,14,38] and this result has inspired other cryptosystems based on SVP [60,61,6]. These cryptosystems are based on SVP and are naturally concerned by lattice reduction algorithm.

CVP-based Cryptosystems
There exists a heuristic way introduced by Kannan [36] to solve CVP (Closest Vector Problem) using a lattice reduction algorithm that was originally proposed to solve SVP, namely the *embedding method*. Instead of solving CVP, we solve SVP in a different lattice. Finding the closest vector of v in $\mathcal{L}_B$ can be done by solving the shortest vector of $\mathcal{L}_{B'}$ with $B' = \begin{pmatrix} B & 0 \\ v & 1 \end{pmatrix}$. This method has been successfully applied by Nguyen [50] to develop his first attack against GGH cryptosystem and it seems practically the best way to attack a CVP-based cryptosystem. In 1997, Goldreich, Goldwasser and Halevi (GGH) [30] proposed an efficient way to use lattice theory to build a cryptosystem inspired by McEliece cryptosystem [42] and based on the Closest Vector Problem (CVP). Their practical proposition of a cryptosystem was attacked and broken severely by Nguyen in 1999 [50]. Nonetheless, the general idea is still viable. Until then, the other propositions were made using the same principle [27,44,59].

After the first Nguyen's attack [50], utilization of the initial GGH proposition requires lattices with a big dimension (> 500), to ensure its security. Consequently, the computation of the closest vector even with a "good basis" becomes very expensive. In 2000, Fischlin and Seifert [27] proposed a very intuitive way to build lattices with good basis which can solve the closest vector problem. They used a tensor product of lattice to obtain a divide and conquer approach to solve CVP. In 2001, Micciancio [44] proposed some major improvements of the speed and the security of GGH. In this scheme, the public key uses a Hermite Normal Form (HNF) for the “bad” basis. The HNF basis is better to answer the inclusion question and it also seems to be more difficult to transform to a “good basis” compared to another basis. In 2003, Paeng, Jung and Ha [59] proposed to use some lattices built on a polynomial ring. However, in 2007, Han, Kim, and Yeom [32] used lattice reduction to cryptanalyze this scheme. Their attack can successfully recover the secret key even in a huge dimension (> 1000) and make the Paeng-Jung-Ha scheme [59] unusable. However, there exists a secure (and yet ‘unbroken’) cryptosystem using polynomial representation, namely the NTRU cryptosystem, for N^{th} degree truncated polynomial ring units. NTRU was originally proposed in 1998 by Hoffstein, Pipher and Silverman [34]. Even this cryptosystem was not modelled initially as a GGH-type cryptosystem, it can actually be represented as one. This has been useful specially for analyzing its security [23].

RSA

In 1996, Coppersmith [20] presented some methods to attack RSA [62] in a special case using lattice reduction. Similar method has been proposed to attack RSA with low exponent [11], RSA with short padding [22] and to factor the RSA public key with or without knowing any partial information [21,12].

Our Contribution

In this paper, we present a methodology that can be applied to *any* lattice reduction algorithms. Our methodology will enable us to find a shorter vector (i.e. a smaller solution) while requiring shorter computation time. The idea of our methodology is as follows. Instead of applying a lattice reduction algorithm to a complete lattice, we work on a sublattice with a smaller dimension obtained in the function of the lattice reduction algorithm that is being used. This way, the lattice reduction algorithm will be fully utilized and hence, it will produce a better solution. Furthermore, as the dimension of the lattice becomes smaller, the time complexity will be better. Hence, our methodology provides us with new direction to build a lattice that is resistant to lattice reduction attacks. Moreover, we also propose a recursive method for producing an optimal approach for lattice reduction with optimal computational time, regardless of the lattice reduction algorithm used. We evaluate our technique by applying it to break the lattice challenge by producing the shortest vector known so far. Our results outperform the existing known results and hence, our results achieve the record in the lattice challenge problem.

Organization of the Paper

The paper is organized as follows. In the next section, we will recall definitions, properties, problems and algorithms of lattice theory required throughout this paper. In Section 3, we will present our methodology. In Section 4, we will present our recursive reduction, followed by analysis of practical tests in Section 5. Finally, Section 6 concludes the paper by showing some results and future works.

2 Lattice Theory

In this section, we will review some concepts of the lattice theory that will be used throughout this paper. For a more complex account, we refer the readers to [45,46].

2.1 Basics of Lattice Theory

The lattice theory, also known as the geometry of numbers, has been introduced by Minkowski in 1896 [49]. A complete discussion on the basic of lattice theory can be found from [15,41,19].

Definition 1 (Lattice). *A lattice $\mathcal{L}$ is a discrete sub-group of $\mathbb{R}^n$, or equivalently the set of all the integral combinations of $d \leq n$ linearly independent vectors over $\mathbb{R}$.*

$$\mathcal{L} = \mathbb{Z}\, b_1 + \cdots + \mathbb{Z}\, b_d, \quad b_i \in \mathbb{R}^n.$$

$B = (b_1, ..., b_d)$ is called a basis of $\mathcal{L}$ and d, the dimension of $\mathcal{L}$, noted $dim(\mathcal{L})$.
We will refer $\mathcal{L}_B$ as a lattice of basis B.

Definition 2 (Full-rank Lattice). *Let $\mathcal{L} \subset \mathbb{R}^n$ be a lattice. If its dimension d is equal to n then the lattice $\mathcal{L}$ is called full-rank.*

Theorem 1 (Determinant). *Let $\mathcal{L}$ be a lattice. There exists a real value, denoted as $\det(\mathcal{L})$, such that for any basis B of $\mathcal{L}$, we have $\det(\mathcal{L}) = \sqrt{\det\left(BB^T\right)}$. $\det(\mathcal{L})$ is called the determinant of $\mathcal{L}$.*

For a given lattice $\mathcal{L}$, there exists an infinite number of basis. However, the Hermite Normal Form basis (Definition 3) is unique [17].

Definition 3 (HNF). *Let $\mathcal{L}$ be an integer lattice of dimension d and $H \in \mathbb{Z}^{d,n}$ be a basis of $\mathcal{L}$. H is a Hermite Normal Form basis of $\mathcal{L}$ if and only if*

$$\forall 1 \leq i, j \leq d \quad H_{i,j} \begin{cases} = 0 & \textit{if } i > j \\ \geq 0 & \textit{if } i \leq j \\ < H_{j,j} & \textit{if } i < j \end{cases}$$

The HNF basis can be computed from a given basis in a polynomial time [37]. For efficient solutions, we refer the readers to [47].

The lattice theory problem is based on distance minimization. The natural norm used in lattice theory is the euclidean norm.

Definition 4 (Euclidean norm). *Let w be a vector of $\mathbb{R}^n$. The euclidean norm is the function $\|.\|$ defined by $\|w\| = \sqrt{< w, w >} = \sqrt{ww^T} = \sqrt{\sum_{i=1}^n w_i^2}$.*

Using a norm, we can define some other invariants that are crucial in lattice theory.

Definition 5 (Successive Minima). *Let $\mathcal{L}$ be a lattice and an integer i. The i^{th} Successive Minima, denoted as λ_i, is the smallest real number such that there exist i non zero linear independent vector $v_1, \ldots, v_i \in \mathcal{L}$ with $\|v_1\|, \ldots, \|v_i\| \leq \lambda_i$.*
If $i = 1$, to find such v_1 is called the Shortest Vector Problem (SVP).

The determinant and the successive minima of a lattice are connected by an important theorem as follows.

Theorem 2 (Minkowski [49]). *Let $d \in \mathbb{N}^+$. There exists a smallest real, γ_d, such that for any lattice $\mathcal{L}$ of dimension d, $\lambda_1(\mathcal{L}) \leq \sqrt{\gamma_d} \det(\mathcal{L})^{1/d}$. γ_d is called the Hermite Constant.*

The exact value of the Hermite constant is only known for $1 \leq d \leq 8$ and $d = 24$. However, some upper bound are known $\gamma_d \leq 1 + \frac{d}{4}$. We refer to [18] for a better numerical upper bound and to [48,19] for a lower and an upper asymptotical bounds.

2.2 Lattice Reduction Algorithms

Theorem 3 (Ajtai [4]). *SVP is NP-Hard under randomized reductions.*

In 2007, Hanrot and Stehle [33] gave the best known deterministic algorithm to solve SVP in time $O(d^{\frac{d}{2e}})$ where d is the dimension: they used the algorithm proposed by Kannan in 1983 [35] and improved the anlaysis of the worst-case complexity. In 2007, Blömer and Naewe [10] proposed the best known probabilistic algorithm to solve SVP in time $(2 + \frac{1}{\epsilon})^d$. It is an improvement of the initial proposition of Ajtai, Kumar, and Sivakumar in 2001 [9].

As SVP is NP-hard, a relaxation factor has been introduced in the initial SVP to be able to propose and evaluate the quality of polynomial algorithms.

Definition 6 (Hermite-SVP). *Let $\mathcal{L}$ be a lattice of dimension d and $\alpha \in R^+$ be a real positive number. Then, the Hermite-SVP is to find a vector $u \in \mathcal{L}$ such that $0 < \|u\| \leq \alpha \det(\mathcal{L})^{1/d}$. α is called the Hermite Factor.*

Theorem 2 ensures a solution for Hermite-SVP if $\alpha \geq \sqrt{\gamma_d}$.

In 1982 Lenstra, Lenstra and Lovasz [40] proposed a powerful polynomial algorithm, known as the LLL algorithm, which solve Hermite-SVP for a Hermite factor $\alpha_{LLL} = \left(\frac{4}{3}\right)^{\frac{d-1}{4}}$. However, in practice LLL seems to be much more efficient [53]. In addition, a lot of improvements have been proposed on LLL to obtain a better approximation factor and/or a better time complexity. For the recent result on LLL, refer to [52,65].

In 1987, Schnorr [63,64] proposed a method which can be seen as a generalization of LLL, known as LLL with deep insertion (DEEP) and Block Korkin-Zolotarev (BKZ). BKZ allows some exponential computations but only on some block. The length k of the block itself is a parameter. LLL can been seen as BKZ with block length of $k = 2$, whereas the Kannan method can be seen as a BKZ with block length of $k = d$. BKZ$-k$ solves Hermite-SVP for $\alpha_{BKZ-k} = \sqrt{\gamma_k}^{1+\sqrt{\frac{d-1}{k-1}}}$ in theory but the BKZ variant used in practice are difficult to evaluate. Theoretically, DEEP has no best upper bound (cf. LLL), $\alpha_{DEEP-k} = \left(\frac{4}{3}\right)^{\frac{d-1}{4}}$.

BKZ is a very powerful way to attack a cryptosystem and it can be extended to provide a level of security with the block length needed to break a cryptosystem. In [50], Nguyen has successfully broken GGH cryptosystem of dimension $200, 250, 300$ using a BKZ-20 and a GGH cryptosystem of dimension 350 using a BKZ-60.

In a recent work [28], Gama and Nguyen presented some tests showing that all of the existing methods seem to solve Hermite-SVP with an average hermite factor α of the form

$$\alpha = bc^d. \tag{1}$$

They also showed that the difference between the theoretical and the practical Hermite factor is huge. We review here those estimations of $\alpha = bc^d$ for different lattice reduction algorithms:

- For LLL, we have $c \sim 1.0219 < 1.0754$.
- For BKZ-20, we have $c \sim 1.0128 < 1.0337$.
- For BKZ-28, we have $c \sim 1.0109 < 1.0282$.
- For DEEP-50, we have $c \sim 1.011 < 1.0754$.

To compare all of these methods practically, Buchmann, Lindner and Rückert [13] proposed a benchmark of lattices created following the paper of Ajtai [3]. Those lattices, denoted as $\mathcal{L}_{m,n,q}$, are characterized using 3 parameters m, n, q. Their basis are as follows

$$\begin{pmatrix} qI & 0 \\ A & I \end{pmatrix}$$

with I the identity matrix and $A \in \mathbb{Z}^{m-n,n}$ a random matrix. The dimension is $\dim(\mathcal{L}_{m,n,q}) = m$ and the determinant is $\det(\mathcal{L}_{m,n,q}) = q^n$.

Those lattices are created such that there exists a vector $v \in \mathcal{L}$ such that $0 < \|v\| \leq \sqrt{m}$. Finding such a vector is the goal of the challenge. However, to find a vector with a norm strictly smaller that q is already difficult[3]. The results of the shortest vector respecting this second condition are presented in the challenge web page http://latticechallenge.org/. There exists a challenge for each dimension with interval 25 between 200 and 2000. However, solutions are accepted only for challenges bigger that 500 (which correspond more to useful dimensions for cryptography).

Remark 1 (Random Lattice). The lattice proposed in this challenge can not be considered as random lattice. Ajtai lattices [3] are lattices for which the solution of SVP implies a solution of SVP in all lattices of a certain smaller dimension. This means that the lattice reduction algorithm solving SVP on those lattice can solve even the worst case of SVP lattices.

Random lattice is a complex notion [5,31,7]. Goldstein and Mayer's characterization of random lattices [31] allows to create random lattices for experiment for example [53]. We will use the same method in our practical section (Section 5) to evaluate our method in the case of random lattices. Practically to respect those criteria, we will create random lattices as $\mathcal{L}_{m,n,q}$ with $n = 1$ and q prime.

3 A Methodology for Lattice Reduction

In this section, we do not propose an algorithm for lattice reduction but rather a methodology applicable to all lattice reduction algorithms with the impact of improving quality and timing of those algorithms.

[3] We note that there exist already some obvious vectors with this norm.

Methodology

Let $\mathcal{A}$ be an algorithm solving Hermite-SVP for a lattice $\mathcal{L}$ of dimension d with a Hermite Factor $\alpha = bc^d$. This means that this algorithm $\mathcal{A}$ will find a vector $v \in \mathcal{L}$ such that $0 < \|v\| \leq bc^d \det(\mathcal{L})^{1/d}$.

The main idea of this methodology can be explained simply in three points as follows.

1. Find d' such that $bc^{d'} \det(\mathcal{L})^{1/d'}$ is minimal.
2. Choose a sublattice $\mathcal{L}' \subseteq \mathcal{L}$ such that $\det(\mathcal{L}') \leq \det(\mathcal{L})$ and $dim(\mathcal{L}') = d'$.
3. Apply $\mathcal{A}$ on $\mathcal{L}'$.

This simple methodology provides several advantages as follows.

a) The quality of the result will be better as $\alpha \det(\mathcal{L})^{1/d'}$ will be smaller.
b) The time and space used will be smaller as $d' \leq d$.

How to Choose a Sublattice $\mathcal{L}' \subseteq \mathcal{L}$

Prior to explaining how to pick d', firstly we develop a simple way to build a sublattice $\mathcal{L}' \subseteq \mathcal{L}$ such that $det(\mathcal{L}') \leq \det(\mathcal{L})$ and $\dim(\mathcal{L}') = d'$. For simplicity, we deal here with a full-rank integer lattice. Nonetheless, this method can be easily modified to accommodate non full-rank and/or non integer lattices.

The first general way is to build a lattice generated by

$$U = \begin{pmatrix} u_1 & 0 & \cdots & 0 & 0 \\ 0 & u_2 & \cdots & 0 & 0 \\ \vdots & & \ddots & & \vdots \\ 0 & 0 & \cdots & u_{d-1} & 0 \\ 0 & 0 & \cdots & 0 & u_d \end{pmatrix}$$

where $u_i = \{0, 1\}$ and $\sum_{i=1}^{d} u_i = d'$. To create $\mathcal{L}'$, we intersect[4] $\mathcal{L}$ and $\mathcal{L}_U$, $\mathcal{L}' = \mathcal{L} \cap \mathcal{L}_U$.

As $\det(\mathcal{L}_U) = 1$ and $\dim(\mathcal{L}_U) = d'$, we obtain $\dim(\mathcal{L}') = d'$ and $\det(\mathcal{L}') \leq \det(\mathcal{L})$[5].

Another simple and practical way is to build the Hermite Normal Form basis of $\mathcal{L}$, and use only the d' last vectors of $\mathcal{L}$ as a basis of $\mathcal{L}'$. Moreover, as all of those vectors will start with zero, we can eventually use only the d' last columns to accelerate some computations. We will need to re-transform the vector to the correct length once the reduction is completed. This method can be generalized using any permutation of the Hermite Normal Form Basis.

How to Find an Optimal d' for a Given Lattice Reduction Algorithm

In this situation, suppose we have a given algorithm $\mathcal{A}$ solving Hermite-SVP for a lattice $\mathcal{L}$ of dimension d with a Hermite Factor $\alpha = bc^d$. This means that this algorithm $\mathcal{A}$ will find a vector $v \in \mathcal{L}$ such that $0 < \|v\| \leq bc^d \det(\mathcal{L})^{1/d}$.

[4] We refer to [45], for the polynomial technique to intersect two lattices.

[5] $\det(\mathcal{L}')$ will be a factor of $\det(\mathcal{L}) \times \det(\mathcal{L}_U) = \det(\mathcal{L})$.

We would like to find d' such that $bc^{d'}\det(\mathcal{L})^{1/d'}$ is minimal or equivalently $\log(b) + d'\log(c) + \frac{\log(\det(\mathcal{L}))}{d'}$ is minimal. To find the minimum value of the function $f(d') = \log(b) + d'\log(c) + \frac{\log(\det(\mathcal{L}))}{d'}$, we compute its derivative $f'(d') = \log c - \frac{\log\det(\mathcal{L})}{d'^2}$ and find d' such that $f'(d') = 0$.

$$\begin{aligned}\log(c) - \frac{\log(\det(\mathcal{L}))}{d'^2} &= 0\\ \log(c) &= \frac{\log(\det(\mathcal{L}))}{d'^2}\\ d'^2 &= \frac{\log(\det(\mathcal{L}))}{\log(c)}\end{aligned}$$

Finally, we obtain that the best evaluated d' as

$$d' = \sqrt{\frac{\log(\det(\mathcal{L}))}{\log(c)}}. \qquad (2)$$

How to Choose Optimally d' to Find a Vector with a Given Norm

In this situation, suppose we have a given lattice in which we want to find a short vector v with a given norm $\|v\|$. Hence, we need an algorithm and a sublattice such that $\|v\| = bc^{d'}\det(\mathcal{L})^{1/d'}$.

In this case, we want to maximize c and therefore, we will use the quicker lattice reduction algorithm.

$$\begin{aligned}c^{d'} &= \frac{\|v\|}{b\det(\mathcal{L})^{1/d'}}\\ c &= \left(\frac{\|v\|}{b\det(\mathcal{L})^{1/d'}}\right)^{1/d'}. \qquad (3)\end{aligned}$$

We can apply an equivalent method.

$$\log(c) = \frac{\log(\|v\|) - \log(b)}{d'} - \frac{\log(\det(\mathcal{L}))}{d'^2}$$

To find the maximum of the function $g(d') = \frac{\log(\|v\|)}{d'} - \frac{\log(\det(\mathcal{L}))}{d'^2}$, we need to evaluate its derivative as follows

$$\begin{aligned}g'(d') &= -\frac{\log(\|v\|) - \log(b)}{d'^2} + 2d\frac{\log(\det(\mathcal{L}))}{d'^4}\\ g'(d') &= -\frac{\log(\|v\|) - \log(b)}{d'^2} + 2\frac{\log(\det(\mathcal{L}))}{d'^3}\end{aligned}$$

and find d' such that $g'(d') = 0$.

$$-\frac{\log(\|v\|) - \log(b)}{d'^2} + 2\frac{\log(\det(\mathcal{L}))}{d'^3} = 0$$

$$\frac{\log(\|v\|) - \log(b)}{d'^2} = 2\frac{\log(\det(\mathcal{L}))}{d'^3}$$
$$\log(\|v\|) - \log(b) = 2\frac{\log(\det(\mathcal{L}))}{d'}$$

Finally, we obtain the best evaluated d' for a maximal c as follows

$$d' = 2\left(\frac{\log(\det(\mathcal{L}))}{\log(\|v\|) - \log(b)}\right). \tag{4}$$

Practically, we can ignore $\log(b)$.

This result is important as it demonstrates which are the most difficult lattices for a given bound. Consequently, this result has a great impact on lattice based cryptography.

4 Recursive Lattice Reduction

When attempting to reduce lattices using the two previous methods, some knowledge on the lattices are required, as well as some knowledge on the lattice reduction algorithm. Nevertheless, these knowledge may be imprecise or even missing. For instance, a small error on c can have a huge implication on d'. Henceforth, in this section, we will propose a new technique that incorporates sublattices without requiring any prior knowledge.

Let $\mathcal{A}$ be an algorithm solving Hermite-SVP for a lattice $\mathcal{L}$. We use a *recursive reduction method* as follows.
The main idea of this technique can be explained as follows.

1. Choose d sublattices $\mathcal{L}_i$ such that $\mathcal{L}_1 \subset \cdots \subset \mathcal{L}_i \cdots \subset \mathcal{L}_d = \mathcal{L}$ and $dim(\mathcal{L}_i) = i$.
2. $\mathcal{L}_1$ is already reduced.
3. To reduce each $\mathcal{L}_{i+1}$, apply $\mathcal{A}$ to $\mathcal{L}_{i+1} \cup \mathcal{L}_i$ where $\mathcal{L}_i$ has already been reduced[6].

This technique incorporates the work that has been performed to reduce the previous sublattice, and hence, it simplifies the reduction of the lattice. This will allow reduction on $\mathcal{L}_{d'}$ where d' is optimal, by trying all the possible dimensions without any extra timing cost. We will demonstrate this in our practical test in Section 5 where this method will improve the time complexity, using time computation whenever appropriate.

5 Practical Test

In this section, we present some tests that have been conducted using NTL5.5.2 and GMP4.3.1 libraries [69,1]. We used random lattices that are built according to the definition [31] as used in [53,28] (Remark 1). However, there is no definition for a random basis of a given lattice.

Figure 1 demonstrates some results using 'random' bases that are built with some techniques similar to [30][7]. Each curve represents an average of 10 tests. The determinant is always equivalent to have a proper comparison; only the dimension changes.

[6] Lattice reduction on a lattice corresponds to lattice reduction on its basis.

[7] We multiplied the hermite normal form basis of the lattice by a random unimodular matrix.

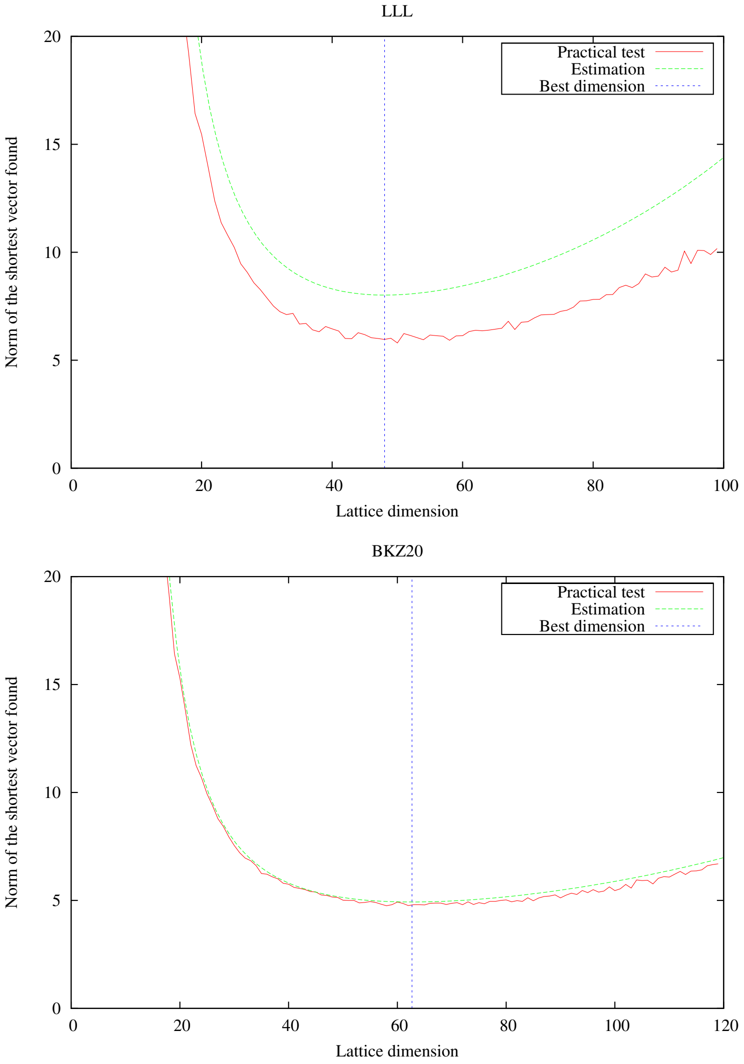

Fig. 1. Result and estimation for LLL and BKZ20

We observed that even if the estimation is not always correct due to the absence of b in its computation, the estimation of the best dimension to use is correct.

After conducting some tests, we observed that instead of using ‘random’ bases, HNF bases (Definition 3) produce better timing results and avoid the problem to produce a worse result in a bigger dimension as shown in Figure 1. However, the use of HNF basis will still be consistent with the estimation of the best dimension to use (Figure 2).

Figure 2 shows that the recursive reduction will also be consistent with the best dimension to use. Figure 2 clearly demonstrates that the results obtained with the three

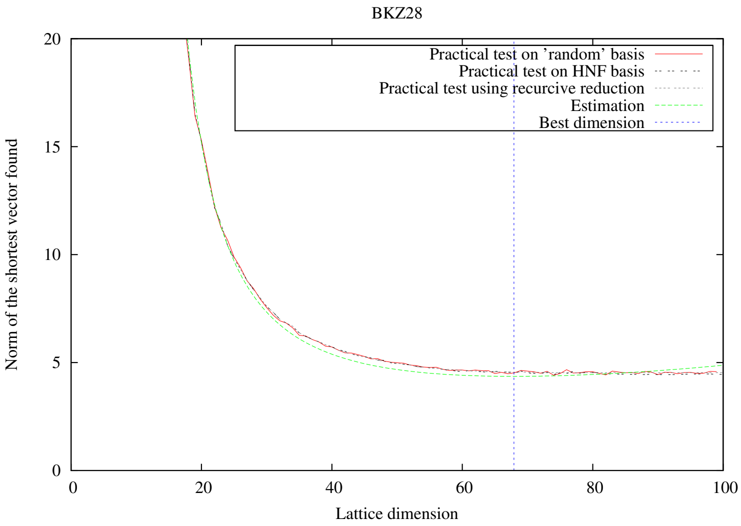

Fig. 2. Result and estimation for BKZ28

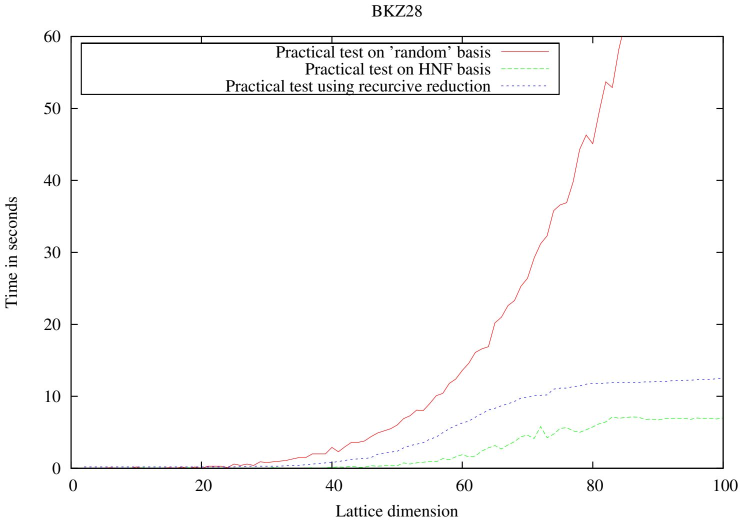

Fig. 3. Time in seconds for BKZ28

different methods are close to each other. However, the main difference is in the timing as shown in Figure 3.

Figure 3 demonstrates that the cost of the recursive method is just a bit higher than the HNF basis if the estimation is correct. However, if the estimation is not precise or unknown, then the non-recursive method will have to do a complete lattice reduction again. In contrast, the recursive method does not need to do so.

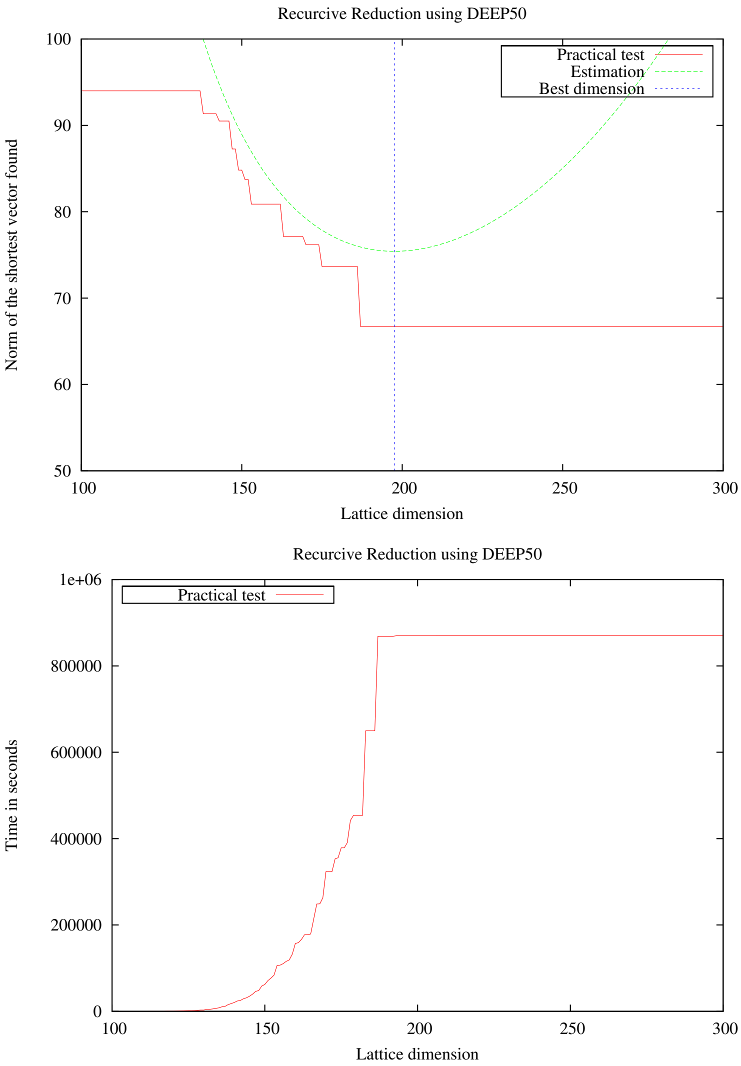

Fig. 4. Result, estimation and time in seconds for DEEP50 with recursive reduction on challenge-650

To finish this section, we present a test performed on lattice of the challenge http://latticechallenge.org/.

Figure 4 demonstrates that the time grows for the recursive reduction only when the quality is improved, and therefore computation is optimally used.

It appears at the end of these practical tests that the recursive technique clearly outperforms the classic application of lattice reduction algorithms, even with a good knowledge of the lattice reduction tool itself. It really simplifies lattice reduction operations as it can be used even if the dimension is not optimal.

6 Conclusion

In this paper, we presented further analysis of lattice reduction algorithms, by presenting a methodology. This methodology offers different consequences, namely better utilization of these algorithms and better level of security of cryptosystems based on or connected to lattices. Using our recursive lattice reduction, we obtained new results in http://latticechallenge.org/ that outperform the previously known results.

Table 1 presents the results we have performed so far on the lattice challenge and outperformed all the previous one. This has been possible only due do the recursive reduction as we have used some reduction techniques (DEEP60,DEEP70, BKZ30, . . .) where the estimation is unknown and very difficult to produce with the precision required for such huge lattices.

Table 1. Lattice Challenge Result

Challenge	Previous Best Result	Recurcive Reduction Result
500	25.8457	25.2587
525	35.6651	30.7409
550	39.7995	38.2884
575	50.7149	42.7083
600	57.2975	52.0096
625	61.8061	59.4138
650	69.4478	66.7158
675	82.6015	80.0937
700	89.4315	89.3924
725	103.7208	100.8960
750	-	-

References

1. The GNU multiple precision arithmetic libraiary
2. Adleman, L.M.: On breaking generalized knapsack public key cryptosystems (abstract). In: STOC, pp. 402–412 (1983)
3. Ajtai, M.: Generating hard instances of lattice problems (extended abstract). In: Twenty-Eighth Annual ACM Symposium on the Theory of Computing (STOC 1996), pp. 99–108 (1996)
4. Ajtai, M.: The shortest vector problem in l_2 is NP-hard for randomized reductions (extended abstract). In: Thirtieth Annual ACM Symposium on the Theory of Computing (STOC 1998), pp. 10–19 (1998)
5. Ajtai, M.: Random lattices and a conjectured 0 - 1 law about their polynomial time computable properties. In: FOCS, pp. 733–742 (2002)
6. Ajtai, M.: Representing hard lattices with o(n log n) bits. In: STOC, pp. 94–103 (2005)
7. Ajtai, M.: Generating random lattices according to the invariant distribution (2006)
8. Ajtai, M., Dwork, C.: A public-key cryptosystem with worst-case/average-case equivalence. In: Twenty-Ninth Annual ACM Symposium on the Theory of Computing (STOC 1997), pp. 284–293 (1997)

9. Ajtai, M., Kumar, R., Sivakumar, D.: A sieve algorithm for the shortest lattice vector problem. In: 33rd Annual ACM Symposium on Theory of Computing (STOC 2001), pp. 601–610 (2001)
10. Blömer, J., Naewe, S.: Sampling methods for shortest vectors, closest vectors and successive minima. In: Arge, L., Cachin, C., Jurdziński, T., Tarlecki, A. (eds.) ICALP 2007. LNCS, vol. 4596, pp. 65–77. Springer, Heidelberg (2007)
11. Boneh, D.: Twenty years of attacks on the rsa cryptosystem. Notices of the American Mathematical Society (AMS) 46(2), 203–213 (1999)
12. Boneh, D., Durfee, G., Howgrave-Graham, N.: Factoring $n = p^r q$ for large r. In: Wiener, M. (ed.) CRYPTO 1999. LNCS, vol. 1666, pp. 326–337. Springer, Heidelberg (1999)
13. Buchmann, J., Lindner, R., Rückert, M.: Explicit hard instances of the shortest vector problem. In: Buchmann, J., Ding, J. (eds.) PQCrypto 2008. LNCS, vol. 5299, pp. 79–94. Springer, Heidelberg (2008)
14. Cai, J.-Y., Cusick, T.W.: A lattice-based public-key cryptosystem. In: Tavares, S., Meijer, H. (eds.) SAC 1998. LNCS, vol. 1556, pp. 219–233. Springer, Heidelberg (1999)
15. Cassels, J.W.S.: An Introduction to the Geometry of Numbers. Springer, Heidelberg (1959)
16. Chor, B., Rivest, R.L.: A knapsack-type public key cryptosystem based on arithmetic in finite fields. IEEE Transactions on Information Theory 34(5), 901–909 (1988)
17. Cohen, H.: A course in computational algebraic number theory. Graduate Texts in Mathematics, vol. 138. Springer, Heidelberg (1993)
18. Cohn, H., Elkies, N.: New upper bounds on sphere packings i. Annals of Mathematics 157(2), 689–714 (2003)
19. Conway, J.H., Sloane, N.J.A.: Sphere Packings, Lattices and Groups. Springer, Heidelberg (1988)
20. Coppersmith, D.: Finding a small root of a univariate modular equation. In: Maurer, U.M. (ed.) EUROCRYPT 1996. LNCS, vol. 1070, pp. 155–165. Springer, Heidelberg (1996)
21. Coppersmith, D.: Small solutions to polynomial equations, and low exponent rsa vulnerabilities. J. Cryptology 10(4), 233–260 (1997)
22. Coppersmith, D.: Finding small solutions to small degree polynomials. In: Silverman, J.H. (ed.) CaLC 2001. LNCS, vol. 2146, pp. 20–31. Springer, Heidelberg (2001)
23. Coppersmith, D., Shamir, A.: Lattice attacks on ntru. In: Fumy, W. (ed.) EUROCRYPT 1997. LNCS, vol. 1233, pp. 52–61. Springer, Heidelberg (1997)
24. Coster, M.J., Joux, A., LaMacchia, B.A., Odlyzko, A.M., Schnorr, C.-P., Stern, J.: Improved low-density subset sum algorithms. Computational Complexity 2, 111–128 (1992)
25. Coster, M.J., LaMacchia, B.A., Odlyzko, A.M.: An iproved low-denisty subset sum algorithm. In: Davies, D.W. (ed.) EUROCRYPT 1991. LNCS, vol. 547, pp. 54–67. Springer, Heidelberg (1991)
26. Diffie, W., Hellman, M.E.: New directions in cryptography. IEEE Transactions on Information Theory IT-22(6), 644–654 (1976)
27. Fischlin, R., Seifert, J.-P.: Tensor-based trapdoors for cvp and their application to public key cryptography. In: IMA Int. Conf., pp. 244–257 (1999)
28. Gama, N., Nguyen, P.Q.: Predicting lattice reduction. In: Smart, N.P. (ed.) EUROCRYPT 2008. LNCS, vol. 4965, pp. 31–51. Springer, Heidelberg (2008)
29. Goldreich, O., Goldwasser, S., Halevi, S.: Eliminating decryption errors in the ajtai-dwork cryptosystem. In: Kaliski Jr., B.S. (ed.) CRYPTO 1997. LNCS, vol. 1294, pp. 105–111. Springer, Heidelberg (1997)
30. Goldreich, O., Goldwasser, S., Halevi, S.: Public-key cryptosystems from lattice reductions problems. In: Kaliski Jr., B.S. (ed.) CRYPTO 1997. LNCS, vol. 1294, pp. 112–131. Springer, Heidelberg (1997)
31. Goldstein, D., Mayer, A.: On the equidistribution of Hecke points. Forum Mathematicum 15, 165–189 (2003)

32. Han, D., Kim, M.-H., Yeom, Y.: Cryptanalysis of the paeng-jung-ha cryptosystem from pkc 2003. In: Okamoto, T., Wang, X. (eds.) PKC 2007. LNCS, vol. 4450, pp. 107–117. Springer, Heidelberg (2007)
33. Hanrot, G., Stehle, D.: Improved analysis of Kannan's shortest lattice vector algorithm. In: Menezes, A. (ed.) CRYPTO 2007. LNCS, vol. 4622, pp. 170–186. Springer, Heidelberg (2007)
34. Hoffstein, J., Pipher, J., Silverman, J.H.: NTRU: A ring-based public key cryptosystem. In: Buhler, J.P. (ed.) ANTS 1998. LNCS, vol. 1423, pp. 267–288. Springer, Heidelberg (1998)
35. Kannan, R.: Improved algorithms for integer programming and related lattice problems. In: Proceedings of the Fifteenth Annual ACM Symposium on Theory of Computing, Boston, Massachusetts, pp. 193–206 (April 1983)
36. Kannan, R.: Minkowski's convex body theorem and integer programming. Math. Oper. Res. 12(3), 415–440 (1987)
37. Kannan, R., Bachem, A.: Polynomial algorithms for computing the Smith and Hermite normal forms of an integer matrix. SIAM Journal of Computing 8(4), 499–507 (1979)
38. Kawachi, A., Tanaka, K., Xagawa, K.: Multi-bit cryptosystems based on lattice problems. In: Okamoto, T., Wang, X. (eds.) PKC 2007. LNCS, vol. 4450, pp. 315–329. Springer, Heidelberg (2007)
39. Lagarias, J.C., Odlyzko, A.M.: Solving low-density subset sum problems. Journal of the ACM 32(1), 229–246 (1985)
40. Lenstra, A.K., Lenstra, H.W., Lovász, L.: Factoring polynomials with rational coefficients. In: Mathematische Annalen, vol. 261, pp. 513–534. Springer, Heidelberg (1982)
41. Lovász, L.: An Algorithmic Theory of Numbers, Graphs and Convexity. CBMS-NSF Regional Conference Series in Applied Mathematics, vol. 50. SIAM Publications, Philadelphia (1986)
42. McEliece, R.J.: A public-key cryptosystem based on algebraic coding theory. Deep Space Network Progress Report 44, 114–116 (1978)
43. Merkle, R.C., Hellman, M.E.: Hiding information and signatures in trapdoor knapsacks. IEEE Transactions on Information Theory IT-24(5), 525–530 (1978)
44. Micciancio, D.: Improving lattice based cryptosystems using the Hermite normal form. In: Silverman, J.H. (ed.) CaLC 2001. LNCS, vol. 2146, pp. 126–145. Springer, Heidelberg (2001)
45. Micciancio, D., Goldwasser, S.: Complexity of Lattice Problems, A Cryptographic Perspective. Kluwer Academic Publishers, Dordrecht (2002)
46. Micciancio, D., Regev, O.: Lattice-based cryptography. In: Bernstein, D.J., Buchmann, J., Dahmen, E. (eds.) Post-quantum Cryprography. Springer, Heidelberg (2008)
47. Micciancio, D., Warinschi, B.: A linear space algorithm for computing the Hermite normal form. In: International Symposium on Symbolic Algebraic Computation (ISSAC 2001), pp. 231–236 (2001)
48. Milnor, J., Husemoller, D.: Symmetric bilinear forms. Springer, Heidelberg (1973)
49. Minkowski, H.: Geometrie der Zahlen. B. G. Teubner, Leipzig (1896)
50. Nguyen, P.Q.: Cryptanalysis of the Goldreich-Goldwasser-Halevi cryptosystem from crypto 1997. In: Wiener, M. (ed.) CRYPTO 1999. LNCS, vol. 1666, pp. 288–304. Springer, Heidelberg (1999)
51. Nguyen, P.Q.: Public-Key Cryptanalysis. Contemporary Mathematics. AMS–RSME (2008)
52. Nguyen, P.Q., Stehlé, D.: Floating-point LLL revisited. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 215–233. Springer, Heidelberg (2005)
53. Nguyen, P.Q., Stehlé, D.: LLL on the average. In: Hess, F., Pauli, S., Pohst, M. (eds.) ANTS 2006. LNCS, vol. 4076, pp. 238–256. Springer, Heidelberg (2006)
54. Nguyen, P.Q., Stern, J.: Cryptanalysis of the ajtai-dwork cryptosystem. In: Krawczyk, H. (ed.) CRYPTO 1998. LNCS, vol. 1462, pp. 223–242. Springer, Heidelberg (1998)

55. Nguyen, P.Q., Stern, J.: Adapting density attacks to low-weight knapsacks. In: Roy, B. (ed.) ASIACRYPT 2005. LNCS, vol. 3788, pp. 41–58. Springer, Heidelberg (2005)
56. Odlyzko, A.M.: The rise and fall of knapsack cryptosystems. Cryptology and Computational Number Theory 42, 75–88 (1990)
57. Okamoto, T., Tanaka, K., Uchiyama, S.: Quantum public-key cryptosystems. In: Bellare, M. (ed.) CRYPTO 2000. LNCS, vol. 1880, pp. 147–165. Springer, Heidelberg (2000)
58. Omura, K., Tanaka, K.: Density attack to the knapsack cryptosystems with enumerative source encoding. IEICE Trans Fundam. Electron Commun. Comput. Sci. 87(6), 1564–1569 (2004)
59. Paeng, S.-H., Jung, B.E., Ha, K.-C.: A lattice based public key cryptosystem using polynomial representations. In: Desmedt, Y.G. (ed.) PKC 2003. LNCS, vol. 2567, pp. 292–308. Springer, Heidelberg (2003)
60. Regev, O.: Improved inapproximability of lattice and coding problems with preprocessing. In: IEEE Conference on Computational Complexity, pp. 363–370 (2003)
61. Regev, O.: On lattices, learning with errors, random linear codes, and cryptography. In: STOC, pp. 84–93 (2005)
62. Rivest, R., Shamir, A., Adleman, L.: A method for obtaining digital signatures and public-key cryptosystems. Communications of the ACM 21(2), 120–126 (1978)
63. Schnorr, C.-P.: A hierarchy of polynomial time lattice basis reduction algorithms. Theoretical Computer Science 53(2-3), 201–224 (1987)
64. Schnorr, C.-P.: A more efficient algorithm for lattice basis reduction. Journal of Algorithms 9(1), 47–62 (1988)
65. Schnorr, C.-P.: Fast LLL-type lattice reduction. Information and Computation 204(1), 1–25 (2006)
66. Schnorr, C.-P., Hörner, H.H.: Attacking the chor-rivest cryptosystem by improved lattice reduction. In: Guillou, L.C., Quisquater, J.-J. (eds.) EUROCRYPT 1995. LNCS, vol. 921, pp. 1–12. Springer, Heidelberg (1995)
67. Shamir, A.: A polynomial time algorithm for breaking the basic merkle-hellman cryptosystem. In: McCurley, K.S., Ziegler, C.D. (eds.) CRYPTO 1982. LNCS, vol. 1440, pp. 279–288. Springer, Heidelberg (1999)
68. Shamir, A.: A polynomial-time algorithm for breaking the basic merkle-hellman cryptosystem. IEEE Transactions on Information Theory 30(5), 699–704 (1984)
69. Shoup, V.: NTL: Number theory library