International Journal on Emerging Technologies 1(1): 67-70(2010) ISSN : 0975-8364

On lattice based cryptographic sampling : An algorithmic approach

Sunderlal, Santosh Kumar Yadav and Kuldeep Bhardwaj***

Professor and Pro-Vice Chancellor, Dr. B.R. Ambedkar University, Agra (UP) INDIA
**Dept. of Mathematics, Kalindi College, University of Delhi (UP) INDIA*
***Dr. B.R. Ambedkar University Agra (UP) INDIA*

(Received 23 Nov., 2009, Accepted 26 Feb., 2010)

ABSTRACT : In this paper we propose a practical lattice based reduction by sampling to avoid any dependence on Schnorr's Geometric Series Assumption. It is a generalization of Schnorr's RSR algorithm. It is also well defined for bases where this algorithm is not applicable. It demonstrates that the sampling reduction can significantly reduce the length of the base vectors. We also propose a practical sampling reduction algorithm for lattice bases based on work by Schnorr. We report the empirical behaviour of these algorithms.

Keywords : Sampling algorithm, Best Bound, NTRU Reduction, lattice bases reduction

I. INTRODUCTION

Lattice bases reduction has been establish as a powerful tool of cryptanalysis. Several cryptosystems have been proposed over the last two decades that are based on the hardness of some lattice problems. The key sizes that need to be selected in order for the system to be secure depend on the efficiency of the best algorithm for computing short vectors in lattices. In 2003 Schnorr presented Random Sampling Reduction (RSR) [1]. It is s new algorithm for computing short vectors in lattices. We assume the Geometrical Series Assumption (GSA), RSR asymptotically outperforms Block Korkine Zolotarev (BKZ) reduction algorithm [6]. However RSR is not a practical algorithm since the choice of parameter in RSR depends on the GSA. The motive of this paper is to present sampling Reduction (SR) as a practical algorithm based on RSR. The experiments to demonstrate that the shortest vector found by SR is significantly shorter than the shortest vector found by BKZ. We also propose two generalization of SR to generate lattice bases with more short vectors On this paper we describe successful attack on low dimensional NTRU lattice bases that require smaller BKZ parameters than previous attacks that used BKZ only.

II. NOTATIONS AND DEFINITIONS

We consider the Euclidean metric on R^d. A lattice L is a discrete subgroup of R^d, its dimension is dim (L): = dim $(L \otimes_R R)$. The first minimum of L is $\lambda_1(L)$: = min $\{||x|| \mid 0 \neq x \in L\}$.

For any n-dimensional lattice L, $n > 1$, there are ordered bases $B = [b_1, ..., b_n] \in R^{d \times n}$ such that $L = L(B)$: = $\{v|v = Bx$ for some $x \in Z^n\}$. Given an ordered basis B, the set of all bases of $L(B)$ is $\{BU \mid U \in Z^{n \times n}$ and det $U = \pm 1\}$. We consider integer coefficient lattices only whence $B \in Z^{d \times n}$.

Let $B = \hat{B}R$ be the Gram-Schmidt decomposition of B, *i.e.*, the columns $\hat{b}_j$ of $\hat{B} \in Q^{d \times n}$ are pairwise perpendicular and $R = (\mu_{i,j}) \in Q^{n \times n}$ is unit upper triangular. Let $\pi_i : R^d \rightarrow \text{lin}\{b_1, ..., b_{i-1}\}^{\perp}$ be the orthogonal projection onto the orthogonal space of the first i-1 base vectors. We denote $L_{i,b}(B) = L([\pi_i(b_i), ..., \pi_i(b_{\min(i+\beta-1, n)})])$.

We also consider a generating system of L and parameters (δ, β) with $1/2 < \delta < 1$ and $2 < \beta \in N$, the BKZ algorithm [6] computes a (δ, β)-BKZ reduced basis of L.A(δ, β)-BKZ reduced basis B satisfies $|\mu_{i,j}| \leq 1/2$ for all $1 \leq i < j \leq n$, (size condition) $\delta \|\hat{b}_i\|^2 \leq \lambda_1(L_{i,\beta}(B))$ for all $1 \leq i \leq n$.

(BKZ condition)

In BKZ reduction we obtain the Gram-Schmidt coefficient matrix R as well as $\|\hat{b}_i\|^2$ for $i = 1, ..., n.L^3$ reduction is the special case of BKZ reduction with $\beta = 2$.

In this paper $B = [b_1, ..., b_n]$ denotes a (δ, β)-BKZ reduced ordered lattices bases with Gram-Schmidt decomposition. All lattice points belong to the n-dimensional lattice $L = L(B)$. B is updated in the course of the reduction, L stays always the same.

III. SAMPLING REDUCTION ALGORITHM

Sampling reduction operates on a generating system G of an n-dimensional lattice L. Sampling reduction applies (δ, β)-BKZ reduction to G and obtain the BKZ reduced bases B. The following lemma illustrates to terminate *SR*.

Lemma. The recursion depth x of SR $(G, \gamma, u_{\max}, \delta, \beta)$ is bound by $x \leq (n-1)\log_{\gamma}(\delta - 1/4)$.

Proof: SR operates on (δ, β)-BKZ reduced and thus δ-L^3 reduced bases. Therefore, $\|b_1\| \leq (\delta - 1/4)^{(1-n)/2} \lambda_1(L)$ [2]. BKZ reduction never increases the length of the first vector in the generating system. Each recursion decreases the length of the first base vector by a factor at most $\sqrt{\gamma} < 1$, and b_1 cannot be shorter than $\lambda_1(L)$. Hence, $\gamma^x(\delta - 1/4)^{1-n} \geq 1$.

The input variable $u_{max} \in N$ limits the amount of work SR spends on sampling vector.

Algorithm : Sampling Reduction (SR)

Input: Generating system G of L, reduction factor γ, search space parameter $u_{\max}$, BKZ parameters (δ, β).

Output: (B, reason) where B is a (δ, β)-BKZ reduced basis of L and reason indicates why the algorithm terminates.

Procedure SR (G, γ, $u_{\max}$, δ, β)
(B, b, R) $\leftarrow$ BKZ (G, δ, β)
/*$B = \hat{B}R, b = (\|\hat{b}_i\|^2, ..., \|\hat{b}_n\|^2)$*/
if – BESTBOUND (b, $u_{\max}$, γ) > $u_{\max}$ **then**
return (B, "success probability too small")
else
for $l = 1, ..., 2^{u_{\max}}$ **do**
$v \rightarrow$ SAMPLE (B, R, l)
if $\|v\|^2 \le \gamma \|b_1\|^2$ **then**
return SR ($[v, b_1, ..., b_n]$, γ, $u_{\max}$, δ, β)
end if
end for
return (B, "search space exhausted")
end if
end procedure

IV. SAMPLING ALGORITHM

The **Sample** is to generate lattice points that are likely to be short. Because of $\|v\|^2 = \sum_{i=1}^{n} v_i^2 \|\hat{b}_i\|^2$, it is plausible to expect that a lattice point v is short if all Gram-Schmidt coefficients v_i are small. **Sample** enumerates lattice points with all $|v_i| \le 1$.

To be precise, let $2^{u-1} < 1 \le 2^u$. Then v = Sample (B, R, l) = $\sum_{i=1}^{n} v_i \hat{b}_i$ satisfies

$$v_i \in \begin{cases} \left(-\frac{1}{2}, \frac{1}{2}\right), & \text{for } 1 \le i < n-u, \\ (-1,1], & \text{for } n-u \le i < n \text{ (SC)} \\ \{1\}, & \text{for } i = n \end{cases}$$

Let $i \in \{1, ..., n\}$. The choice of $v = \sum x_j b_j = \sum v_j \hat{b}_j$ does not affect $v_{i+1}, ..., v_n$ since R is unit upper triangular. Therefore, Sample computes (x_i, v_i) by iteration based on $x_n = v_n = 1$. Assume the coefficients (x_{i+1}, v_{i+1}), ..., (x_n, v_n) are already fixed. Then Sample determines the unique $x' \in Z$ with $\pi_i (x'b_1 + \Sigma_{J=i+1} x_j b_j) = v' \hat{b}_i = \sum_{J=i+1}^{n} v_J \hat{b}_J$ and $v' \in (-1/2, 1/2)$. Sampling chooses (x_i, v_i) = (x', v') if l div 2^{n-i-1} is even. Else (x_i, v_i) becomes also unique ($x' \pm 1$, $v' \pm 1$) s.t. $v_i \in (-1, -1/2] \cup [1/2, 1)$.

Thus, $\{1, ..., 2^{u_{\max}}\} \rightarrow L(B) : l \mapsto$ **Sample** (B, R, l) is an enumeration of all points in $L(B)$ subject to (SC) with $u = u_{\max}$. Inspection of the following algorithm shows the computation of Sample requires $2n$ vector updates and assignments, *i.e.*, $O(n^2)$ arithmetic operations.

Algorithm :

Input : Unit upper triangular matrix $R = [r_1, ..., r_n] \in Q^{n \times n}$, lattice basis $B = [b_1, ..., b_n] \in Z^{n \times n}$ with Gram-Schmidt decomposition $B = \hat{B}R$, $1 \le l \le 2^{n-1}$.

Output: $u \in L(B)$ subject to (SC).

Procedure: Sample (B, R, l)
$v \leftarrow b_n$, $v = (v_1 ..., v_n)^t \leftarrow r_n$
for $i = n-1, n-2, ..., 1$ **do**
$x \leftarrow \left\lceil v_i - \frac{1}{2} \right\rceil$
if l **mod** 2 = 1 **then**
/*$-\frac{1}{2} \le v_i - x \le \frac{1}{2}$*/
if $v_i - x \le 0$ **then**
$x \leftarrow x - 1$ /*$\frac{1}{2} < v_i - x \le 1$*/
else
$x \leftarrow x + 1$ /*$-1 < v_i - x \le -\frac{1}{2}$*/
end if
end if
$l \leftarrow 1$ **div** 2
$v \leftarrow v - xb_i$, $v \leftarrow v - xr_i$
/*$v_i \leftarrow v_i - x$*/
end for
return v
end procedure

V. BEST BOUND ALGORITHM

The vectors computed by SAMPLE are likely to be short but they are of course not necessarily shorter than b_1. The algorithm BESTBOUND yields as estimate how many samples are required in the search space $V_t := \{v_1, ..., v_{2-t}\}$ if we want to guarantee a success probability Pr [min $\{\|v\|^2 \; v \in V_t\} \le \gamma \|b_1\|^2] \ge 1/2$.

Let $l \in_R \{1, ..., 2^u\}$ be by SAMPLE (B, R, l) = $\sum_{i=1} v_i \hat{b}_i$ are statistically indistinguishable from independent random variables with uniform distribution on the intervals defined by (SC).

BESTBOUND is supposed to return a lower bound for (the $\log_2$ of) the probability that SAMPLE returns a vector shorter than $\sqrt{\gamma}\|b_1\|$. The algorithm is based on the following idea: The sampling of a lattice point v is a random experiment. We consider some event ($S_{q,k}$) parameterized by

$q \in [0, 1]$ and $1 \le k < n - u_{max} < n$. The probability of $(S_{q,k})$ is strictly increasing in q. Let $0 \le q_\gamma \le 1$ be maximal s.t. the conditional expected length $E[\|u\|^2 | (S_{q,k}] \le \gamma \|b_1\|^2$. Then the success probability is

$$Pr[\| v \|^2 < \gamma \| b_1 \|^2] > \Pr\{\| v \|^2 < E \| v \|^2 \mid (S_{q,k})] \mid (S_{q,k})] = \frac{1}{2} \Pr[(S_{q,k})]$$

Best Bound computes

$$\max \left\{ \left| \log_2 \left(\frac{1}{2} \Pr[(S_{q_\gamma,k})] \right) \right| \mid k = 1, ..., n - u_{\max} \right\} .$$

Consequently, if SR samples at least $2^{-BestBond(b,u_{\max},\gamma)}$ lattice points then the probability to find a sufficiently short vector is at least ½.

The event $(S_{q,k})$. Consider the random experiment $v = \text{Sample}(B, R, l) = \sum_{i=1}^{n} v_i \hat{b}_i, l \in_R \{1, ..., 2^{u_{\max}}\}$. $\sigma \in$ Sym $(\{1, ..., n\})$ describes the sorting of the firs n-$u_{\max} - 1$ elements of b in non-increasing order, *i.e.*,

$$\| \hat{b}_{\sigma(1)} \|^2 \ge ... \ge \| \hat{b}_{\sigma(n-u_{\max}-1)} \|^2 \text{ and}$$

$$\sigma(i) = i \text{ for } i \ge n - u_{\max} \quad ...(1)$$

Let $q \in [0, 1]$ and $1 \le k < n - u_{max}$. $(S_{q,k})$ denotes the event

$$v_{\sigma(i)}^2 \le \frac{1}{4} q^{k-i} \text{ for } i = 1, ..., k - 1.(S_{q,k})$$

The randomness assumption on v_i yields

$$\Pr[(S_{q,k})] = \prod_{i=1}^{k-1} \Pr\left[| v_{\sigma(i)} | \le \frac{1}{2} q^{\frac{k-i}{2}} \right] = q^{\frac{k(k-1)}{4}}$$

The expected length of v. Assume $(S_{q,k})$. For any uniform random variable $x \in (-t, t]$ the expected value of x^2 is $E[x^2] = 1/3t^3$. The v_i are independent random variables uniformly distributed on intervals defined by (SC) and $(S_{q,k})$ whence

$$\begin{aligned} E(b, k, q) : &= E[\|v\|^2 | (S_{q,k})] \\ &= \sum_{i=1}^{n} E\left[v_i^2 \mid (S_{q,k}) \right] \| \hat{b}_i \|^2 \\ &= \sum_{i=1}^{k-1} q^{k-i} \frac{\| \hat{b}_{\sigma(i)} \|^2}{12} + \sum_{i=k}^{n-u_{\max}-1} \frac{\| \hat{b}_{\sigma(i)} \|^2}{12} \\ &\quad + \sum_{i=n-u_{\max}}^{n-1} \frac{\| \hat{b}_i \|^2}{3} + \| \hat{b}_n \|^2 \end{aligned}$$

Algorithm : BestBound

Input : $b = (\| \hat{b}_1 \|^2, ..., \| \hat{b}_n \|^2)$, base 2 logarithm $u_{\max}$ of maximum number of samples, reduction factor γ.

Output:

$t_{\max} = \max \{t \in Z \mid \Pr[\min\{\| v \|^2 \le \gamma \| \hat{b}_1 \|^2 \mid v \in V_t\}] \le 1/2\} \cup \{-\infty\}$

procedure ExpLength (l, k, u, γ)

/* $l = (l_1, ..., l_n)$ */

return

$$\frac{1}{12} \sum_{i=1}^{k-1} q^{k-i} l_i + \frac{1}{12} \sum_{i=k}^{n-u-1} l_i + \frac{1}{3} \sum_{i=n-u}^{n-1} l_i + l_n$$

end procedure

procedure LogSuccessProbBound (l, k, u, γ)

if ExpLength (L, k, u, 1) $\le \gamma \| \hat{b}_1 \|^2$

then

return – 1

else if ExpLength (l, k, u, 0) $\le \gamma \| \hat{b}_1 \|^2$

then return – ∞

end if

$q_\gamma \leftarrow$ RegulaFalsi(ExpLength(l, k, u, q) = $\gamma \| \hat{b}_1 \|^2$, $q \in$ [0, 1]) **return** $\left\lfloor \frac{k(k-1)}{4} \log_2(q_\gamma) - 1 \right\rfloor$ **end procedure**

procedure BestBound (b, $u_{\max}$, γ)

$l \leftarrow (\| \hat{b}_{\sigma(1)} \|^2, ..., \| \hat{b}_{\sigma(n)} \|^2)$

/* permutation σ */

return max {LogSuccessProbBound (l, k, $u_{\max}$, γ) | $k = 1, ..., n - u_{\max}$} **end procedure**

Numerical Representation

Computation of q_γ. The expected length $E(b, k, q)$ is a polynomial in q with non-negative coefficients. Therefore $f : [0, 1] \to R : q \mapsto E(b, k, q) - \gamma \|b_1\|^2$ is a strictly increasing continuous function that has a root if and only if $f(0) \le 0 \le f(1)$. The unique root q_γ can be efficiently determined with the textbook Regula Falsi algorithm [7] provided such a root exists.

If $f(1) < 0$ then the (unconditional) means value $E[\|v\|^2]$ is already short enough and we have $\Pr[\|v\|^2 \le \gamma \|b_1\|^2] \ge 1/2$. On the other hand, if $f(0) > 0$ then our approach dos not yield a positive lower bound on $\Pr[\|v\|^2 \le \gamma \|b_1\|^2]$ for this particular choice of k.

The optimal bound t. BestBound computes the maximum success probability for all k. The computation of $\Pr[(S_{q\gamma,k})]$ is in our experience fast enough that the cost for computing the probability for all $k = 1, ..., n - u_{\max} - 1$ is negligible.

VI. RESULTS

Followed by Linux system with a 2.4 GHz Pentium 4 processor and 1 GByte RAM. We used a lattice reduction library that is derived from Shoup's NTL [8]. We tested our algorithm with bases in Hermite normal form as proposed

by Micciancio [7] for the public keys in his variant of the GGH cryptosystem. They are derived from base vectors uniformly chosen from a cube whence the generated lattices do not have any special structure. The HNF bases were (0.99 β)-BKZ reduced for various values of β. The resulting bases were input to the Sampling Reduction.

A large part of this improvement is gained in the first iterations. With β = 5, the Sampling Reduction took 1928s, of 2 which 71s were spent on the BKZ updates. With β = 10, the Sampling reduction ran only for 577s but here 190s were spent in the BKZ updates.

It is noticeable that the very first base vectors are much more improved than the remaining base vectors. Most of the time, the effect of the BKZ updates peters to quickly. In particular, $\| \hat{b}_i \|^2$ does not change significantly beyond base column 20. Since $E[\|v\|^2]$ does not change that much if only few $\hat{b}_i$ become smaller it quickly becomes less likely that a sampled vector is shorter than b_1. This is also reflected in our estimates of the success probability's logarithm, the estimates decrease quite rapidly with very recursion.

The value of BestBound actually depends on the choice of $u_{\max}$: If one increments $u_{\max}$ then $E(b, k, q)$ grows by $\frac{1}{4}\| \hat{b}_{n-u_{\max}-1} \|^2$ which means that q_γ and therefore $\Pr[(S_{q\gamma, k})]$ become smaller.

VII. CONCLUSION AND FUTURE TRENDS

In our work we have demonstrated that the Sampling Reduction can significantly reduce the length of the base vectors. We have also proposed two generalizations that further reduce the overall length of the base vectors and that allow the Sampling Reduction to proceed even if jumps in the length of the orthogonalized base vectors disrupt the plain Sampling Reduction. Observing the algorithms and procedure.

We find that our estimates of the success probability are too pessimistic. We plan to test whether it is numerically feasible to calculate the distribution of the length of the sampled vectors directly by convoluting (*via* FFT) the distributions of the coefficients v_i. The result needs to be verified by further experiments in higher dimensions and approach. The implementation can also be performed in $C^{\#}$.

REFERENCES

[1] Schnorr, C.P. Lattice reduction by random sampling and birthday methods. In Alt, H., Habib, M., eds: STACS 2003: *20th Annual Symposium on Theoretical Aspects of Computer Science. Volume 2607 of LNCS, Springer,* pp. 146-156(2003).

[2] A.K. Lenstra, H.W. Lensta, L. Lovasz. Factoring polynomials with rational coefficients. *Math. Ann.* **261:** 515-534(1982).

[3] M. Ajtai, C. Dwork. A public-key cryptosystem with worstcase / average-case equivalence. In: *Proceedings of the 29th Annual Symposium on Theory of Computing (STOC),* ACM Press, 284-293(1997).

[4] D. Micciancio. The shortest vector problem is NP-hard to approximate to within some constant. *SIAM Journal on Computing,* **30:** 2008-2035(2001).

[5] NTRU Cryptosystems, Inc. : Website: http://www.ntru.com, (2004)

[6] C.P. Schnorr, M. Euchnere. Lattice basis reduction: Improved practical algorithms and solving subset sum problems. *Math. Programming,* **66:** 181-199(1994).

[7] W.H. Press, S.A. Tuekolsky, W.T. Vetterling, B.P. Flannery, *Numerical Recipes in C. 2nd edn.* Cambridge University Press (1992).

[8] V. Shoup. NTL-a library for doing number theory. URL, *http://www. shoup. net/ntl/index.html* (2004) Release 5.3.2.

[9] Coppersmith, D., Shamir, A.: Lattice attacks on NTRU. In: *Advances in Cryptology–Eurocrypt' 97,* Vol. 1233 of LNCS, Springer 52-61(1997).

[10] A.K. Lenstra, E.R. Verheul. Selecting cryptographic key sizes. *J. Cryptology* **14:** 255-293(2001).