Joseph H. Silverman (Ed.)

LNCS 2146

Cryptography and Lattices

International Conference, CaLC 2001
Providence, RI, USA, March 2001
Revised Papers

Springer

Joseph H. Silverman (Ed.)

Cryptography and Lattices

International Conference, CaLC 2001
Providence, RI, USA, March 29-30, 2001
Revised Papers

Springer

Series Editors

Gerhard Goos, Karlsruhe University, Germany
Juris Hartmanis, Cornell University, NY, USA
Jan van Leeuwen, Utrecht University, The Netherlands

Volume Editor

Joseph H. Silverman
Brown University, Mathematics Department - Box 1917
Providence, RI 02912, USA
E-mail: jhs@math.brown.edu

Cataloging-in-Publication Data applied for

Die Deutsche Bibliothek - CIP-Einheitsaufnahme

Cryptography and lattices : international conference, revised papers / CaLC 2001, Providence, RI, USA, March 29 - 30, 2001. Joseph H. Silverman (ed.). - Berlin ; Heidelberg ; New York ; Barcelona ; Hong Kong ; London ; Milan ; Paris ; Singapore ; Tokyo : Springer, 2001
(Lecture notes in computer science ; Vol. 2146)
ISBN 3-540-42488-1

CR Subject Classification (1998): E.3, F.2.1, F.2.2, G.1, I.1.2, G.2, K.4.4

ISSN 0302-9743
ISBN 3-540-42488-1 Springer-Verlag Berlin Heidelberg New York

Springer-Verlag Berlin Heidelberg New York
a member of BertelsmannSpringer Science+Business Media GmbH

http://www.springer.de

Printed in Germany

Typesetting: Camera-ready by author, date conversion by Christian Grosche, Hamburg
Printed on acid-free paper SPIN 10840224 06/3142 5 4 3 2 1 0

Preface

These are the proceedings of CaLC 2001, the first conference devoted to cryptography and lattices. We have long believed that the importance of lattices and lattice reduction in cryptography, both for cryptographic construction and cryptographic analysis, merits a gathering devoted to this topic. The enthusiastic response that we received from the program committee, the invited speakers, the many people who submitted papers, and the 90 registered participants amply confirmed the widespread interest in lattices and their cryptographic applications.

We thank everyone whose involvement made CaLC such a successful event; in particular we thank Natalie Johnson, Larry Larrivee, Doreen Pappas, and the Brown University Mathematics Department for their assistance and support.

March 2001 Jeffrey Hoffstein, Jill Pipher, Joseph Silverman

Organization

CaLC 2001 was organized by the Department of Mathematics at Brown University. The program chairs express their thanks to the program commitee and the additional external referees for their help in selecting the papers for CaLC 2001. The program chairs would also like to thank NTRU Cryptosystems for providing financial support for the conference.

Program Commitee

- Don Coppersmith <dcopper@us.ibm.com>
  IBM Research
- Jeffrey Hoffstein (co-chair) <jhoff@math.brown.edu>, <jhoff@ntru.com>
  Brown University and NTRU Cryptosystems
- Arjen Lenstra <arjen.lenstra@citicorp.com>
  Citibank, USA
- Phong Nguyen <Phong.Nguyen@ens.fr>
  ENS
- Andrew Odlyzko <amo@research.att.com>
  AT&T Labs Research
- Joseph H. Silverman (co-chair) <jhs@math.brown.edu>, <jhs@ntru.com>
  Brown University and NTRU Cryptosystems

External Referees

Ali Akhavi, Glenn Durfee, Nick Howgrave-Graham, Daniele Micciancio

Sponsoring Institutions

NTRU Cryptosystems, Inc., Burlington, MA <www.ntru.com>

Table of Contents

An Overview of the Sieve Algorithm for the Shortest Lattice Vector Problem

Miklós Ajtai, Ravi Kumar, and Dandapani Sivakumar

IBM Almaden Research Center
650 Harry Road, San Jose, CA 95120
{ajtai,ravi,siva}@almaden.ibm.com

We present an overview of a randomized $2^{O(n)}$ time algorithm to compute a shortest non-zero vector in an n-dimensional rational lattice. The complete details of this algorithm can be found in [2].

A lattice is a discrete additive subgroup of $\mathbf{R}^n$. One way to specify a lattice is through a basis. A basis $B = \{b_1, \ldots, b_n\}$ is a set of linearly independent vectors in $\mathbf{R}^n$. The lattice generated by a basis B is $L = L(B) = \{\sum_{i=1}^{n} c_i b_i \mid c_i \in \mathbf{Z}\}$. The shortest lattice vector problem (SVP) is the problem of finding a shortest non-zero vector (under some norm, usually ℓ_2) in L. The α-approximate version of SVP is to find a non-zero lattice vector whose length is at most α times the length of a shortest non-zero lattice vector.

SVP has a rich history. Gauss and Hermite studied an equivalent of SVP in the context of minimizing quadratic forms [4, 7]. Dirichlet formulated SVP under the guise of diophantine approximations. Using the convex body theorem, Minkowski gave an existential bound on the shortest vector in a lattice [13].

Though the extended Euclidean GCD algorithm can be used to solve SVP in two dimensions, the first algorithmic breakthrough in n dimensions was obtained in a celebrated result of Lenstra, Lenstra, and Lovász [10], who gave an algorithm (the LLL algorithm) that computes a $2^{n/2}$-approximate shortest vector in polynomial time. This was improved in a generalization of the LLL algorithm by Schnorr [14], who obtained a hierarchy of algorithms that provide a uniform trade-off between the running time and the approximation factor. This algorithm runs in $n^{O(1)}k^{O(k)}$ steps to solve a $k^{O(n/k)}$-approximate SVP. For instance, a polynomial-time version of this algorithm improves the approximation factor obtained by the LLL algorithm to $2^{n(\log\log n)^2/\log n}$. Kannan [8] obtained a $2^{O(n \log n)}$ time algorithm for the exact SVP. The constant in the exponent of this algorithm was improved to about $1/2$ by Helfrich [6]. Recently, Kumar and Sivakumar solved the decision version of n^3-approximate SVP in $2^{O(n)}$ time [9].

On the hardness front, SVP for the L_∞ norm was shown to be NP-complete by van Emde Boas [3]. Ajtai [1] proved that SVP under the ℓ_2 norm is NP-hard under randomized reductions. Micciancio [12] showed that the α-approximate SVP remains NP-hard for any $\alpha < \sqrt{2}$. Lagarias, Lenstra, and Schnorr [11] showed that n-approximate SVP is unlikely to be NP-hard. Goldreich and Goldwasser [5] showed that $\sqrt{n/\log n}$-approximate SVP is unlikely to be NP-hard.

We sketch a randomized $2^{O(n)}$ algorithm for SVP (in ℓ_2 norm) for a lattice L in $\mathbf{R}^n$. In fact, in $2^{O(n)}$ time, our algorithm can find all α-approximate shortest vectors for any constant $\alpha \geq 1$.

J.H. Silverman (Ed.): CaLC 2001, LNCS 2146, pp. 1–3, 2001.

We make a few simplifying assumptions about the lattice L: (1) the length of shortest vector is at least 1 and at most 2 — this can be realized by appropriately scaling L; (2) the length of the longest vector in the basis is at most $2^{O(n)}$ — this can be realized by appropriate applications of the LLL algorithm.

We create a large (sides of exponential length) parallelepiped $\mathcal{P}$ that is fairly close to being a cube. Then we uniformly sample a large number of lattice points $z_1, \ldots, z_N$, $N = 2^{O(n)}$, from $\mathcal{P} \cap L$, and to each sample z_i, we add a uniform perturbation vector y_i of expected length $O(1)$ to obtain a sequence of points $x_1, \ldots, x_N$. For each perturbed lattice point x_i, we will keep track of two lattice points: its "true identity" z_i, and an "approximator" a_i, initially set to 0.

Then, we use the following sieve procedure — given sufficiently many points in $\mathbf{R}^n$ of length at most R, identify a small set of "representatives" from the set of points and a large set of "survivors" such that for every survivor point, there is a representative at distance at most $R/2$. We repeatedly apply the sieve to the vectors $x_i - a_i$; for each survivor $x_i - a_i$ with representative $x_j - a_j$, we know that the distance between $x_i - a_i$ and $x_j - a_j$ is about half the distance between x_i and a_i. Therefore, $a_i + x_j - a_j$ is a better approximation to x_i, and since x_j is close to its true identity z_j, we define the new approximator for x_i to be $a_i + z_j - a_j$. In these steps, once the true identity of a point is revealed, we will not use it in the future. We repeat this process until the distance between the points and their approximators are bounded by another constant. Finally, if x_i still survives and has an approximator a_i, output the lattice point $w_i = z_i - a_i$. Since both z_i and a_i are close to x_i, with high probability, the length of w_i is bounded by a constant. We will denote this process as the basic algorithm.

Note that if the basic algorithm stops with a non-zero w_i, we already have a constant factor approximation algorithm for SVP. To ensure that w_i is non-zero with good probability and to obtain the shortest vector, we make the following argument. Let u denote a shortest vector in L. Let $w = w_i$ be a lattice point of constant length that is output by the procedure above. Let x be a sample point from which w was obtained, and let $z \in L$ be the true identity of x. Since the perturbations are small, we can argue that the probability (conditioned on x being a sample) that one of $z \pm u$ is the true identity of x is at least $2^{-O(n)}$ times the probability that z is the true identity of x. Furthermore — and this is the crucial point — the basic algorithm is oblivious to the true identity of x. Using this fact, we will argue that for some w, $w + u$ has at least $2^{-O(n)}$ times the probability of w to be the output of the basic algorithm. Since the number of lattice points in the ball of constant radius around the origin is at most $2^{O(n)}$, we obtain that there is at least one $w \in L$ whose probability of being output is at least $2^{-O(n)}$ and $w + u$ has the probability of being output at least $2^{-O(n)}$. Therefore, by repeating the basic algorithm $2^{O(n)}$ times we can ensure that with high probability both w and $w + u$ are output. Thus, the final algorithm is the following: repeat the basic algorithm $2^{O(n)}$ times, take all possible pairwise differences of the points output by the basic algorithm, and output the shortest of these vectors.

More details of this algorithm can be found in [2].

References

1. M. Ajtai. The shortest vector problem in L_2 is NP-hard for randomized reductions. *Proc. 30th ACM Symposium on Theory of Computing*, pp. 10–19, 1998.
2. M. Ajtai, R. Kumar, and D. Sivakumar. A sieve algorithm for the shortest lattice vector problem. *Proc. 33rd ACM Symposium on Theory of Computing*, 2001. To appear.
3. P. van Emde Boas. Another NP-complete partition problem and the complexity of computing short vectors in lattices. *Mathematics Department, University of Amsterdam*, TR 81-04, 1981.
4. C. F. Gauss. *Disquisitiones Arithmeticae*. English edition, (Translated by A. A. Clarke) Springer-Verlag, 1966.
5. O. Goldreich and S. Goldwasser. On the limits of nonapproximability of lattice problems. *Journal of Computer and System Sciences*, 60(3):540–563, 2000.
6. B. Helfrich. Algorithms to construct Minkowski reduced and Hermite reduced bases. *Theoretical Computer Science*, 41:125–139, 1985.
7. C. Hermite. Second letter to Jacobi, Oeuvres, I, *Journal für Mathematik*, 40:122–135, 1905.
8. R. Kannan. Minkowski's convex body theorem and integer programming. *Mathematics of Operations Research*, 12:415–440, 1987. Preliminary version in *ACM Symposium on Theory of Computing* 1983.
9. R. Kumar and D. Sivakumar. On polynomial approximations to the shortest lattice vector length. *Proc. 12th Symposium on Discrete Algorithms*, 2001.
10. A. K. Lenstra, H. W. Lenstra, and L. Lovász. Factoring polynomials with rational coefficients. *Mathematische Annalen*, 261:515–534, 1982.
11. J. C. Lagarias, H. W. Lenstra, and C. P. Schnorr. Korkine-Zolotarev bases and successive minima of a lattice and its reciprocal lattice. *Combinatorica*, 10:333–348, 1990.
12. D. Micciancio. The shortest vector in a lattice is hard to approximate to within some constant. *Proc. 39th IEEE Symposium on Foundations of Computer Science*, pp. 92–98, 1998.
13. H. Minkowski. *Geometrie der Zahlen*. Leipzig, Teubner, 1990.
14. C. P. Schnorr. A hierarchy of polynomial time basis reduction algorithms. *Theoretical Computer Science*, 53:201–224, 1987.

Low Secret Exponent RSA Revisited

Johannes Blömer and Alexander May

Department of Mathematics and Computer Science
University of Paderborn, 33095 Paderborn, Germany
{bloemer,alexx}@uni-paderborn.de

Abstract. We present a lattice attack on low exponent RSA with short secret exponent $d = N^{\delta}$ for every $\delta < 0.29$. The attack is a variation of an approach by Boneh and Durfee [4] based on lattice reduction techniques and Coppersmith's method for finding small roots of modular polynomial equations. Although our results are slightly worse than the results of Boneh and Durfee they have several interesting features. We partially analyze the structure of the lattices we are using. For most $\delta < 0.29$ our method requires lattices of smaller dimension than the approach by Boneh and Durfee. Hence, we get a more practical attack on low exponent RSA. We demonstrate this by experiments, where $\delta > 0.265$.
Our method, as well as the method by Boneh and Durfee, is heuristic, since the method is based on Coppersmith's approach for bivariate polynomials. Coppersmith [6] pointed out that this heuristic must fail in some cases. We argue in this paper, that a (practically not interesting) variant of the Boneh/Durfee attack proposed in [4] always fails. Many authors have already stressed the necessity for rigorous proofs of Coppersmith's method in the multivariate case. This is even more evident in light of these results.
Keywords: Low secret exponent RSA, cryptanalysis, Coppersmith's method, lattice reduction.

1 Introduction

In this paper we consider the problem of breaking the RSA cryptosystem for short secret keys. An RSA public key is a pair (N, e) where $N = pq$ is a product of two n-bit primes. The corresponding secret key d is chosen such that it satisfies the equation

$$ed \equiv 1 \pmod{\tfrac{1}{2}\phi(N)},$$

where $\phi(N) = (p-1)(q-1)$.

The first result showing that RSA is insecure, if the secret key is too small, is due to Wiener. In 1990, Wiener [20] showed that $d < \frac{1}{3}N^{0.25}$ leads to a polynomial time attack on the RSA system. Wiener's method is based on continued fractions. Basically, Wiener showed that d is the denominator of some convergent of the continued fraction expansion of e/N. A variant of Euclid's algorithm computes the continued fraction expansion of a number. Since N, e both are public, this shows that d can be computed efficiently from the public key (N, e).

J.H. Silverman (Ed.): CaLC 2001, LNCS 2146, pp. 4–19, 2001.

Recently, Boneh and Durfee [4] proposed an attack on RSA, that shows that RSA is insecure provided $d < N^{0.292}$. Unlike Wiener's attack, the attack by Boneh and Durfee is a heuristic. It builds upon Coppersmith's result for finding small solutions of modular polynomial equations [6]. Coppersmith's method for the univariate case is rigorous but the proposed generalization for the multivariate case is a heuristic. More precisely, Boneh and Durfee show that for a small secret key d, the number $s = -\frac{p+q}{2}$ can be found as a small solution to some modular bivariate polynomial equation. Once s is known, one can immediately solve the equations $s = -\frac{p+q}{2}$ and $N = pq$ for the unknowns p and q. Using Coppersmith's method, which in turn is based on the famous L^3-lattice reduction algorithm, Boneh and Durfee reduce the problem of finding s to finding a *common root* of two bivariate polynomials $f(x,y), g(x,y)$ over the integers. As proposed by Coppersmith, finding a common root of f, g is done by first computing the resultant $r(y)$ of f, g with respect to the variable x. Provided $r \not\equiv 0$, the parameter s, and hence the factorization, can be found by computing the roots (over $\mathbb{Z}$) of r. Unfortunately, this method, as well as any other method based on Coppersmith's approach for multivariate polynomials[1], fails if the resultant r is identically 0. As it has never been proved that $r \not\equiv 0$, the Boneh/Durfee approach is heuristic.

In this paper we study the method by Boneh and Durfee in more detail. In Section 4, we propose a new lattice for cryptanalysing low secret exponent RSA with $d < N^{0.290}$. The new approach uses the same heuristical assumption as Boneh/Durfee. Although the new attack does not improve the bound $d < N^{0.292}$ of Boneh and Durfee [4], it has several advantages. First, the lattice dimension is reduced. Therefore, in practice we are able to get closer to the theoretical bounds. Second, the new lattice basis is triangular. This leads to rather simple proofs. Third, the new lattice basis takes advantage of special properties of the lattice vectors. We believe that some of our structural results in Section 4 can be applied to other applications of Coppersmith's method as well.

Actually, Boneh and Durfee present three different variations of the Coppersmith methodology to break RSA versions with small secret exponent d. The first one works for $d < N^{1/4}$, hence this variant basically reproduces Wiener's result. The second variation of Boneh and Durfee works for $d < N^{0.284}$. Finally they have a method that works for d up to $N^{0.292}$.

We made the experimental observation, that the first method of Boneh and Durfee, supposed to work for $d < N^{1/4}$ always failed. In fact, in all experiments the resultant r mentioned above was identically zero. Although one cannot recover the factorization by resultant computation, we show that RSA with secret key $d < \frac{1}{3}N^{1/4}$ can be broken using lattice reduction in dimension 2. In fact, we show that for an appropriately chosen lattice, a shortest vector in the lattice immediately reveals the secret key d.

Since we have not found examples where the other two variants for $d < N^{0.284}$ and $d < N^{0.292}$ described by Boneh and Durfee fail, this observation in no way invalidates the results of Boneh and Durfee. On the other hand, this is

[1] This includes among others [1, 4, 8, 12].

to our knowledge the first case mentioned in literature, that an application of Coppersmith's approach fails in general. Some authors [6, 14] already pointed out that the heuristic must fail in some cases, but no general failure has been reported for real applications of the method.

Although we are not quite able to rigorously analyze the Boneh and Durfee method for $d < N^{1/4}$, in Section 5 we prove several results that almost completely explain the behavior observed in experiments. Many authors already stressed the necessity of a rigorous analysis of methods based on Coppersmith's approach in the multivariate case. This is even more evident in light of our results.

In Section 6 we give experimental results for our new attack on RSA with short secret key d. We carried out cryptanalysis of secret keys up to $d \leq N^{0.278}$. We also compared our experimental results with the experimental results of Boneh and Durfee. In [3], they only provided examples with $d \leq N^{0.265}$. In all cases we considered, our method was faster.

2 The Boneh-Durfee Lattice

In this section we review the lattice attack by Boneh and Durfee on low exponent RSA. For an introduction into lattice theory and lattice basis reduction, we refer to the textbooks [9, 17]. Descriptions of Wiener's RSA attack and the method of Coppersmith can be found in [6, 20]. For a good overview of RSA attacks, we refer to a survey article of Boneh [2].

Let $d < e^{\delta}$. We assume that the size of e is in the order of the size of N. If e is smaller, the attack of Boneh and Durfee becomes even more effective (see [4], section 5).

All known attacks on RSA with short secret exponent focus on the identity

$$ed = 1 \bmod \frac{\phi(N)}{2} \quad \Leftrightarrow \quad ed + k\left(\frac{N+1}{2} + s\right) = 1, \tag{1}$$

where $k \in \mathbb{Z}$, $s = -\frac{p+q}{2}$ and d are unknown quantities. Since $e < \frac{\phi(N)}{2}$, we obtain $k < d$. Boneh and Durfee [4] look at equation (1) modulo e.

$$k\left(\frac{N+1}{2} + s\right) - 1 = 0 \bmod e$$

They define the polynomial

$$f(x, y) = x(A + y) - 1$$

with $A = \frac{N+1}{2}$. Let $X = e^{\delta}$ and $Y = e^{0.5}$. We know, that f has a root $(x_0, y_0) = (k, s)$ modulo e, that satisfies $|x_0| < X$ and $|y_0| < Y$. To transform the modular equation into an equation over the integers, Boneh/Durfee use a theorem of Howgrave-Graham [11]. Given a polynomial $p(x, y) = \sum_{i,j} a_{i,j} x^i y^j$, we define the norm $\|p(x, y)\|^2 = \sum_{i,j} a_{i,j}^2$.

Theorem 1 (Howgrave-Graham [11]). *Let $p(x,y)$ be a polynomial which is a sum of at most w monomials. Suppose that $p(x_0,y_0) = 0 \bmod e^m$ for some positive integer m, where $|x_0| < X$ and $|y_0| < Y$. If $\|p(xX,yY)\| < e^m/\sqrt{w}$, then $p(x_0,y_0) = 0$ holds over the integers.*

Next, Boneh and Durfee define polynomials

$$g_{i,k}(x,y) = x^i f^k(x,y) e^{m-k} \quad \text{and} \quad h_{j,k}(x,y) = y^j f^k(x,y) e^{m-k}$$

for a given positive integer m.

In the sequel, the polynomials $g_{i,k}$ are referred to as x-shifts and analogously the polynomials $h_{j,k}$ are referred to as y-shifts. By construction, the point (x_0,y_0) is a root of all these polynomials modulo e^m. Thus, we can apply Howgrave's theorem and search for a small norm linear combination of polynomials $g_{i,k}(xX,yY)$ and $h_{j,k}(xX,yY)$. This is done by using the L^3 lattice reduction algorithm. The goal is to construct a lattice that is guaranteed to contain a vector shorter than $e^m/\sqrt{w}$.

Boneh and Durfee suggest to build the lattice spanned by the coefficient vectors of the polynomials $g_{i,k}, h_{j,k}$ for certain parameters i, j and k. For each $k = 0,\ldots,m$, they use the x-shifts $g_{i,k}(xX,yY)$ for $i = 0,\ldots,m-k$. Additionally, they use the y-shifts $h_{j,k}$ for $j = 0,\ldots,t$ for some parameter t.

In the sequel, we call the lattice constructed by Boneh and Durfee the lattice L_{BD}. The basis for L_{BD} is denoted by B_{BD}. The lattice L_{BD} is spanned by the row vectors of B_{BD}. Since the lattice depends on the parameters m and t, we sometimes refer to the parameters by $B_{BD}(m,t)$ to clarify notation.

It is easy to see, that the basis vectors of lattice L_{BD} form a triangular matrix. We give an example of the lattice basis for the parameter choice $m = 2$ and $t = 1$.

$B_{BD}(2,1) =$

	1	x	xy	x^2	x^2y	x^2y^2	y	xy^2	x^2y^3
e^2	e^2								
xe^2		e^2X							
fe	$-e$	eAX	eXY						
x^2e^2				e^2X^2					
xfe		$-eX$		eAX^2	eX^2Y				
f^2	1	$-2AX$	$-2XY$	A^2X^2	$2AX^2Y$	X^2Y^2			
ye^2							e^2Y		
yfe			$eAXY$				$-eY$	eXY^2	
yf^2			$-2AXY$		A^2X^2Y	$2AX^2Y^2$	Y	$-2XY^2$	X^2Y^3

Boneh and Durfee showed for $\delta < 0.284$, one can find m,t such that an L^3-reduced basis of L_{BD} contains vectors short enough to apply Howgrave's theorem and factor the modulus N. This was improved in the same paper to $\delta < 0.292$ by using non-triangular lattice bases. This is up to now the best

bound for cryptanalysis of low secret exponent RSA. The attack works under the assumption that polynomials obtained from two sufficiently short vectors in the reduced basis have a non-vanishing resultant. Although heuristic, no failure of the method for sufficiently large δ is known.

Boneh and Durfee also argue that using $t = 0$, that is only x-shifts are used to construct a lattice basis, one obtains already an attack working for $\delta < 0.25$. This reproduces Wiener's result. However, experiments show that the method of Boneh and Durfee never works when using only x-shifts. In Section 5, we will explain why this is the case. Of course, this failure of the Boneh/Durfee method in the special case where only x-shifts are used does not affect the method in general. It only points out that one has to be careful when using Coppersmith's heuristic in the multivariate case.

3 Notations

Since the lattice L_{BD} defined in Section 2 is the starting point of our further constructions, we introduce some notations on the rows and columns of the lattice basis B_{BD}.

We refer to the coefficient vectors of the polynomials $g_{i,k}(xX, yY)$ as the X-block. The X-block is further divided into $X_l, l = 0, \ldots, m$, blocks, where the block X_l consist of the $l+1$ coefficient vectors of $g_{i,k}$ with $i + k = l$. These $l+1$ vectors are called $X_{l,k}$, that is the k-th vectors in the X_l block is the coefficient vector of $g_{l-k,k}$.

The coefficient vectors of the polynomials $h_{j,k}$ form the Y-block. We define the Y_j block as the block of all $m+1$ coefficient vectors of polynomials that are shifted by y^j. The k-th vector in the block Y_j is called $Y_{j,k}$, it is identical to the coefficient vector of $h_{j,k}$.

Every column in the basis B_{BD} is labeled by a monomial $x^i y^j$. All column vectors with label $x^l y^j$, $l \geq j$, form the $X^{(l)}$ column block. Analogously, we define the $Y^{(l)}$ column block to consist of all column vectors labeled with $x^i y^{i+l}$.

In the example in Section 2, the horizontol lines divide the basis $B_{BD}(2,1)$ into the blocks X_1, X_2, X_3 and Y_1. Similarly, the vertical lines divide $B_{BD}(2,1)$ into the column blocks $X^{(1)}$, $X^{(2)}$, $X^{(3)}$ and $Y^{(1)}$. In this example, the basis entry in row $Y_{1,2}$ and column x^2y is A^2X^2Y.

4 A New Method for All $\delta < 0.290$

We introduce an alternative method for factoring the modulus N if $d < N^{0.290}$. This does not improve the bound $\delta < 0.292$ given by Boneh and Durfee. However, it has several advantages compared to their approach.

First, our method significantly reduces the lattice dimension as a function of m and t. The practical implication is that we are able to get closer to the theoretical bound. We give experimental results for $\delta > 0.265$. Second, our proofs are simple. As opposed to the Boneh/Durfee lattices for $\delta < 0.292$, the lattice

bases we use in the attack for $\delta < 0.290$ remain triangular. Hence, determinant computations are simple. Third, our construction makes use of structural properties of the underlying polynomials. Thus, it should apply also to other lattice constructions using these polynomials.

Construction of the new lattice L with basis B

1. Choose lattice parameters m and t and build the Boneh-Durfee lattice basis $B_{BD}(m,t)$ as explained in Section 2.
2. In the Y_t block of the basis B_{BD} remove every vector except for the last vector $Y_{t,m}$, in the Y_{t-1} block remove every vector except for the last two vectors $Y_{t,m-1}$ and $Y_{t,m}$, and so on. Finally, in the Y_1 block remove every vector except for the last t vectors $Y_{m-t+1}, \ldots, Y_m$.
3. Remove every vector in the X-block except for the vectors in the $t+1$ blocks X_{m-t}, X_{m-t+1}, $\ldots$, X_m.
4. Delete columns in such a way that the resulting basis is again triangular. This is, remove all column blocks $X^{(0)}, X^{(1)}, \ldots, X^{(m-t-1)}$. Furthermore in the column block $Y^{(l)}$, $l = 1, \ldots, t$, remove the columns labeled with $x^i y^{i+l}$ for $0 \leq i < m - t + l$.

This construction leads to a triangular basis B of a new lattice L, which will be used in our approach. Since B depends on m and t, we sometimes write $B(m,t)$.

As opposed to Boneh and Durfee, we do not integrate more y-shifts to improve the bound $\delta < 0.284$, instead we remove some x-shifts.

Remark 1. *In our construction, we take the pattern*

$$(p_0, p_1, \ldots, p_t) = (1, 2, \ldots, t+1).$$

That is, we take the last p_i, $0 \leq i < t$ vectors from the Y_{t-i} block and the last p_t X-blocks and delete columns appropriately. The proofs in this section easily generalize to every strictly increasing pattern $(p_0, p_1, \ldots, p_t)$, $p_0 < p_1 < \cdots < p_t$. This includes among others the pattern used by Boneh/Durfee [4] to show the bound $d < N^{0.292}$. We give the proof of this generalization in the full version of the paper.

Applying the construction to the example given in Section 2, we obtain the following lattice basis of L with parameters $m = 2$ and $t = 1$.

$B(2,1) =$

	x	xy	x^2	x^2y	x^2y^2	x^2y^3
xe^2	e^2X					
fe	eAX	eXY				
x^2e^2			e^2X^2			
xfe	$-eX$		eAX^2	eX^2Y		
f^2	$-2AX$	$-2XY$	A^2X^2	$2AX^2Y$	X^2Y^2	
yf^2		$-2AXY$		A^2X^2Y	$2AX^2Y^2$	X^2Y^3

Let $\bar{B}$ be the non-triangular basis we obtain after Step 3 of the construction. That is, $\bar{B}$ consists of the remaining basis vectors of B_{BD} in the construction after removing row vectors but without removing columns. The lattice spanned by the row vectors of $\bar{B}$ is called $L_{\bar{B}}$. We adopt the notations of Section 3 for the rows and columns of B and $\bar{B}$. For example, the row vector $X_{l,k}$ of B is the coefficient vector of $g_{l-k,k}$, where we removed all the entries specified in Step 4 of the construction. In the basis $B(2,1)$ above, the row vector $X_{2,0}$ is the vector $(0,0,e^2X^2,0,0,0)$.

We call a column vector x^iy^j that appears in the basis $\bar{B}$ but not in the basis B a *removed column* of B. The bases B and $\bar{B}$ are constructed using the same coefficient vectors, where in B certain columns are removed. Having a vector $u = \sum_{b\in B} c_b b$ in the span of B, one can compute the corresponding linear combination $\bar{u} = \sum_{b\in\bar{B}} c_b b$ of vectors in $\bar{B}$ with the same coefficients c_b. Hence, the vector dimension of $\bar{u}$ is larger than the vector dimension of u. One can regard the additional vector entries in $\bar{u}$ as a reconstruction of the vector entries of u in the removed columns. Therefore, we call $\bar{u}$ the *reconstruction vector* of u.

The row vectors

$$X_{l,k}, (l = m-t,\ldots,m; k \le l) \quad \text{and} \quad Y_{j,k}, (j = 1,\ldots,t; k = m-t+j,\ldots,m)$$

form the basis B. These vectors are no longer the coefficient vectors of the polynomials $g_{l-k,k}(xX,yY)$ and $h_{j,k}(xX,yY)$, respectively, since we remove columns in Step 4 of the construction. However in order to apply Howgrave's theorem, we must ensure that we construct a linear combination of bivariate polynomials that evaluates to zero modulo e^m at the point $(x_0,y_0) = (k,s)$. Hence, we still have to associate the rows $X_{l,k}$ and $Y_{j,k}$ with the polynomials $g_{l-k,k}$ and $h_{j,k}$. The basis vectors of $\bar{B}$ represent the coefficient vectors of these polynomials. Therefore, after finding a small vector $u = \sum_{b\in B} c_b b$ in L, we compute the reconstruction vector $\bar{u} = \sum_{b\in\bar{B}} c_b b$ in $L_{\bar{B}}$. That is, we reconstruct the entries in the removed columns. Once the reconstruction vectors of two sufficiently short vectors in L are computed, the rest of our method is the same as in the Boneh/Durfee method.

In the remainder of this section we show that short vectors u in L lead to short reconstruction vectors $\bar{u}$ in $L_{\bar{B}}$. To prove this, we first show that removed columns of B are small linear combinations of column vectors in B. We give an example for the removed column x^0y^0 in $B(2,1)$. Applying the construction in the following proof of Lemma 2, we see that this column is a linear combination of the columns x^1y^1 and x^2y^2 in B.

$$\begin{pmatrix} 0 \\ -e \\ \hline 0 \\ 0 \\ 1 \\ \hline 0 \end{pmatrix} = -\frac{1}{XY}\begin{pmatrix} 0 \\ eXY \\ \hline 0 \\ 0 \\ -2XY \\ \hline -2AXY \end{pmatrix} - \frac{1}{X^2Y^2}\begin{pmatrix} 0 \\ 0 \\ \hline 0 \\ 0 \\ X^2Y^2 \\ \hline 2AX^2Y^2 \end{pmatrix}$$

Lemma 2. *All removed columns in the column blocks $X^i, i < m - t$, are linear combinations of columns in B. Moreover, in these linear combinations, the coefficient for a column vector in $X^{(l)}, l \geq m - t$, can be bounded by $\frac{1}{(XY)^{l-i}} \cdot c$, where c depends only on m and t.*

Proof: If $x^i y^j$ is a removed column of B, we show that $x^i y^j$ is a linear combination of columns $x^{i+1}y^{j+1}, \ldots, x^m y^{m-i+j}$. If $x^{i+1}y^{i+1}$ is a removed column, we can repeat the argument to show that $x^{i+1}y^{i+1}$ is a linear combination of the remaining columns $x^{i+2}y^{j+2}, \ldots, x^m y^{m-i+j}$. Continuing in this way until all removed columns have been represented as linear combinations of columns in B, proves the lemma. Hence, it suffices to prove the following claim.

Claim 1. *If $x^i y^j$ is a removed column of B, then $x^i y^j$ is a linear combination of the columns $x^{i+1}y^{j+1}$, $x^{i+2}y^{j+2}, \ldots$, $x^m y^{m-i+j}$, where the coefficient of column $x^{i+b}y^{j+b}, b = 1, \ldots, m-i$, is given by*

$$-\frac{1}{(XY)^b}\binom{j+b}{j}.$$

Note, that the coefficient $c_b = \binom{j+b}{j}$ depends only on m and t, since i, j depend on m and t.

We will prove Claim 1 by showing that for each row in $B(m,t)$ the entry of the column $x^i y^j$ in this row is a linear combination of the entries of the columns $x^{i+b}y^{j+b}$ in this row, with the coefficients as in the claim. We prove this for the rows in the X-block and Y-block separately.

Let $X_{l,k}$ be a row in block X_l, where $l \geq m - t$. The coefficients in this row are the coefficients of the polynomial $e^{m-k}x^{l-k}f^k(xX, yY)$. By definition of f this polynomial is

$$e^{m-k}x^{l-k}f^k(xX, yY) = e^{m-k}\sum_{p=0}^{k}\sum_{q=0}^{p}(-1)^{k+p}\binom{k}{p}\binom{p}{q}A^{p-q}X^p Y^q x^{p+l-k}y^q. \tag{2}$$

To obtain the coefficient of $x^{i+b}y^{j+b}$ in $e^{m-k}x^{l-k}f^k(xX, yY)$, we set $p = i - l + k + b$ and $q = j + b$. Hence, this coefficient is given by

$$e^{m-k}(-1)^{i-l+b}\binom{k}{i-l+k+b}\binom{i-l+k+b}{j+b}A^{i-l+k-j}X^{i-l+k+b}Y^{j+b}$$
$$= e^{m-k}A^{i-l+k-j}X^{i-l+k}Y^j(-1)^{i-l}(-1)^b\binom{k}{i-l+k+b}\binom{i-l+k+b}{j+b}(XY)^b.$$

We can ignore the factor $e^{m-k}A^{i-l+k-j}X^{i-l+k}Y^j(-1)^{i-l}$, common to all entries in row $X_{l,k}$ in the columns $x^{i+b}y^{j+b}$. Then Claim 1 restricted to row $X_{l,k}$ reads