Partial Key Exposure Attack on RSA – Improvements for Limited Lattice Dimensions

Santanu Sarkar, Sourav Sen Gupta, and Subhamoy Maitra

Applied Statistics Unit, Indian Statistical Institute,
203 B T Road, Kolkata 700 108, India
sarkar.santanu.bir@gmail.com, sg.sourav@gmail.com, subho@isical.ac.in

Abstract. Consider the RSA public key cryptosystem with the parameters $N = pq$, $q < p < 2q$, public encryption exponent e and private decryption exponent d. In this paper, cryptanalysis of RSA is studied given that some amount of the Most Significant Bits (MSBs) of d is exposed. In Eurocrypt 2005, a lattice based attack on this problem was proposed by Ernst, Jochemsz, May and de Weger. In this paper, we present a variant of their method which provides better experimental results depending on practical lattice parameters and the values of d. We also propose a sublattice structure that improves the experimental results significantly for smaller decryption exponents.

Keywords: Cryptanalysis, Factorization, Lattice Reduction, Public Key Cryptosystem, RSA, Sublattice.

1 Introduction

The RSA [15] public key cryptosystem can be briefly described as follows:

- primes p, q, (generally considered of same bit size, i.e., $q < p < 2q$);
- $N = pq$, $\phi(N) = (p-1)(q-1)$;
- e, d are such that $ed = 1 + k\phi(N)$, $k \geq 1$;
- N, e are public and plaintext $M \in \mathbb{Z}_N$ is encrypted as $C \equiv M^e \bmod N$;
- secret key d needed to decrypt ciphertext $C \in \mathbb{Z}_N$ as $M \equiv C^d \bmod N$.

One important model of cryptanalysis in the field of RSA is side channel attacks such as fault attacks, timing attacks, power analysis etc. [3,12,13], by which an adversary may obtain some bits of the private key d.

Boneh et al. [3] studied how many bits of d need to be known to factor the RSA modulus N. The constraint in [3] was the upper bound on e, that had been $\sqrt{N}$. The idea of [3] has been improved by Blömer and May [2] where the bound on e was increased upto $N^{0.725}$. Then the work by Ernst et al. [8] improved the result for full size public exponent e. Sarkar and Maitra [16] extended the work of [8] by guessing few bits of one prime. Recently, the work by Aono [1] improved the results of [8] when some portion of Least Significant Bits (LSBs) of d are exposed and $d < N^{0.5}$. In this paper, we propose a variant of the idea presented in [8] to make the results more practical when some portion of Most Significant

G. Gong and K.C. Gupta (Eds.): INDOCRYPT 2010, LNCS 6498, pp. 2–16, 2010.

Bits (MSBs) of d are exposed and $d < N^{0.6875}$. One may argue that exposing LSBs and MSBs pose two different scenarios. But if we compare the two methods by the total number of bits of d that one needs to know for cryptanalysis, our method improves the results of [8] for a larger range of d than [1].

In this paper we consider the case when some MSBs of d are exposed. So one can write $d = d_0 + d_1$ where the attacker knows d_0. Attacker can also find an approximation $k_0 = \lfloor \frac{ed_0-1}{N} \rfloor$ of k. Let $k_1 = k - k_0$ and d be of bitsize $\delta \log_2 N$. Ernst et al. [8] considered the polynomial $f_1(x, y, z) = ed_0 - 1 + ex - Ny - yz$ and it is clear that (d_1, k, s) is a root of f_1 where $s = 1 - p - q$. Further, the approximation k_0 of k has been used to consider the polynomial $f_2(x, y, z) = ed_0 - 1 - k_0N + ex - Ny - yz - k_0z$. In this case, (d_1, k_1, s) is the root of f_2. If one can find the root of either f_1 or f_2, N can be factored.

Given $(\delta - \gamma) \log_2 N$ many MSBs of d, one can find the root of f_1 or f_2 in poly$(\log N)$ time if any of the following holds [8][1]:

$$\gamma < \tfrac{5}{6} - \tfrac{1}{3}\sqrt{1 + 6\delta},$$
$$\gamma < \tfrac{3}{16} \text{ and } \delta \leq \tfrac{11}{16},$$
$$\gamma < \tfrac{1}{3} + \tfrac{1}{3}\delta - \tfrac{1}{3}\sqrt{4\delta^2 + 2\delta - 2} \text{ and } \delta \geq \tfrac{11}{16}.$$

In this paper we consider the polynomial $f_e(x, y) = 1 + (k_0 + x)(N + y)$ over $\mathbb{Z}_e$ where the terms k_0, k_1, d_0, d_1, s are same as mentioned before. Clearly (k_1, s) is the root of f_e. If one gets s, then immediately the factorization of N is possible. We use Coppersmith's [5] method for roots of modular polynomials to find such a root. As predicted in [8, Section 5], this leads to lattices of smaller dimension and hence better practical results for fixed lattice parameters within certain range of d. However, [8] does not precisely analyze this situation, and so we provide a comprehensive treatment of such an analysis in Section 2.

Though the theoretical bounds on γ, as given in [8], work for $d < \phi(N)$, the experimental results could only be achieved for the range $d \leq N^{0.7}$ with lattice dimension upto 50. Our experimental results are better than that of [8] for $d \leq N^{0.64}$ with smaller lattice dimensions. The results explaining experimental advantage are presented in Section 4.

Although the practical attacks are mounted using lattices with small dimension, where the lattice parameters are generally predetermined, the results of this kind are often compared in asymptotic sense in literature. In this direction, we show that the root of f_e can be obtained in poly$(\log N)$ time if $\lambda < \frac{3}{16}$, where $\lambda = \max\{\gamma, \delta - \frac{1}{2}\}$. Our results are as good as [8] for $N^{0.4590} \leq d < N^{0.6875}$ in terms of asymptotic bound.

The reader may note that our results are not better than those in [8] if we consider the asymptotic performance. It is only better in practical experimental scenario, where we obtain results of the same quality as in [8] by using lattices with comparatively smaller dimension. The reason is that we use Coppersmith's [5] idea for the modular polynomial, while [8] used Coron's [6] version for ease of presentation. Thus, the lattice dimension obtained in [8] was a cubic in a certain parameter while we obtain a quadratic, hence smaller. This idea has

[1] The terms δ, β in [8] are denoted as γ, δ respectively in our analysis.

already been pointed out in [8] itself, but rigorous analysis for limited lattice dimensions is studied here.

Further, in Section 3, we propose the construction of a sublattice by deleting certain rows of the above mentioned lattice. This provides significantly improved results for smaller decryption exponents. Once again, experimental evidences in Section 4 support our claim.

1.1 Preliminaries

Let us start with some basic concepts on lattice reduction techniques. Consider a set of linearly independent vectors $u_1, \ldots, u_\omega \in \mathbb{Z}^n$, with $\omega \leq n$. The lattice L, spanned by $\{u_1, \ldots, u_\omega\}$, is the set of all integer linear combinations of the vectors $u_1, \ldots, u_\omega$. The number of vectors ω is the dimension of the lattice. Such a lattice is called full rank when $\omega = n$. By $u_1^*, \ldots, u_\omega^*$, we denote the vectors obtained by applying the Gram-Schmidt process [4, Page 81] to $u_1, \ldots, u_\omega$. The determinant of L is defined as $\det(L) = \prod_{i=1}^{\omega} ||u_i^*||$, where $||.||$ denotes the Euclidean norm on vectors. Given a bivariate polynomial $g(x, y) = \sum a_{i,j} x^i y^j$, the Euclidean norm is defined as $\| g(x, y) \| = \sqrt{\sum_{i,j} a_{i,j}^2}$ and the infinity norm is defined as $\| g(x, y) \|_\infty = \max_{i,j} |a_{i,j}|$. We shall follow these notation in this paper.

Fact 1. *Given a basis $u_1, \ldots, u_\omega$ of a lattice L, the LLL algorithm [14] generates a new basis $b_1, \ldots, b_\omega$ of L with the following properties.*

1. $\| b_i^* \|^2 \leq 2 \| b_{i+1}^* \|^2$, *for* $1 \leq i < \omega$.
2. *For all i, if $b_i = b_i^* + \sum_{j=1}^{i-1} \mu_{i,j} b_j^*$ then $|\mu_{i,j}| \leq \frac{1}{2}$ for all j.*
3. $\| b_i \| \leq 2^{\frac{\omega(\omega-1)+(i-1)(i-2)}{4(\omega-i+1)}} det(L)^{\frac{1}{\omega-i+1}}$ *for* $i = 1, \ldots, \omega$.

Here $b_1^*, \ldots, b_\omega^*$ denote the vectors obtained by applying Gram-Schmidt process to $b_1, \ldots, b_\omega$.

In [5], Coppersmith discusses lattice based techniques to find small integer roots of univariate polynomials $\bmod n$, and of bivariate polynomials over the integers. The idea of [5] can also be extended to more than two variables, but the method becomes heuristic. Lemma 1 is relevant to the idea of [5] for finding roots of bivariate polynomials over integers.

Lemma 1. *Let $g(x_1, x_2)$ be a polynomial which is the sum of ω many monomials. Suppose $g(y_1, y_2) \equiv 0 \bmod n$, where $|y_1| < Y_1$ and $|y_2| < Y_2$. If $\| g(x_1 Y_1, x_2 Y_2) \| < \frac{n}{\sqrt{\omega}}$, then $g(y_1, y_2) = 0$ holds over integers.*

We apply Gröbner Basis based techniques to solve for the roots of bivariate polynomials. Though our technique works in practice as noted from the experiments we perform, theoretically this may not always happen. Thus we formally state the following heuristic assumption, that we will require for our theoretical results.

Assumption 1. *Suppose that one constructs a lattice using the idea of Coppersmith [5] in order to find the root of a bivariate modular equation. Further, consider that the lattice reduction is executed using the LLL algorithm. Let the polynomials corresponding to the first two basis vectors of the lattice after LLL reduction be $\{f_1, f_2\}$ and they share the common root of the form $(x_1^{(0)}, x_2^{(0)})$. If J be the ideal generated by $\{f_1, f_2\}$, then one can efficiently collect the root by computing the Gröbner Basis of J.*

Note that the time complexity of the Gröbner Basis computation is in general double-exponential in the degree of the polynomials [7].

2 The Lattice Based Technique

We start with the following theorem.

Theorem 1. *Consider the RSA equation $ed \equiv 1 \pmod{\phi(N)}$. Let $d = N^{\delta}$ and e be $\Theta(N)$. Suppose we know an integer d_0 such that $|d - d_0| < N^{\gamma}$. Then, under Assumption 1, one can factor N in poly$(\log N)$ time when*

$$\lambda < \frac{\frac{1}{12}m^3 - \frac{13}{12}m + \frac{1}{4}m^2t + \frac{1}{4}mt}{\frac{1}{2}m^3 + m^2 + \frac{1}{2}m + \frac{1}{2}t^2 + \frac{1}{2}t + m^2t + \frac{1}{2}mt^2 + \frac{3}{2}mt}$$

where $\lambda = \max\{\gamma, \delta - \frac{1}{2}\}$ and m, t are non-negative integers.

Proof. From the RSA equation, we have $ed = 1 + k(N + 1 - p - q)$. When MSBs of d are known, we can write $d = d_0 + d_1$, where d_0, corresponding the MSBs of d, is known to the attacker, but d_1 is not. The attacker can also calculate $k_0 = \lfloor \frac{ed_0 - 1}{N} \rfloor$ as an approximation of k and set $k_1 = k - k_0$.

We can write $ed = 1 + (k_0 + k_1)(N + s)$, where $s = 1 - p - q$. Thus $1 + k_0N + k_0s + k_1N + k_1s \equiv 0 \pmod{e}$ and we are interested in finding the solution $(x_0, y_0) = (k_1, s)$ of

$$f_e(x, y) = \overline{1 + k_0N} + k_0y + Nx + xy$$

in $\mathbb{Z}_e$. Note that we are considering the polynomial $f_e(x, y)$ reduced modulo e, and hence the modified constant term $\overline{1 + k_0N}$ is actually equivalent to $1 + k_0N - ed_0$, which is much smaller than the original. This helps in reducing the bit size of some elements in the matrix corresponding to the lattice we describe below.

Following results by Blömer and May [2, Proof of Theorem 6], and Ernst et al. [8, Section 2], it can be shown that $|k_1| < 4N^{\lambda}$, for $\lambda = \max\{\gamma, \delta - \frac{1}{2}\}$. We also have $|s| \leq 2N^{0.5}$, by definition. Now, let us take $X = N^{\lambda}$ and $Y = N^{0.5}$. One may note that X, Y are the upper bounds of the roots $(x_0, y_0) = (k_1, s)$ of $f_e(x, y)$, neglecting the respective small constants 4 and 2 respectively. For integers $m, t \geq 0$, we define two sets of polynomials

$$g_{i,j}(x, y) = x^i f_e^j(x, y) e^{m-j} \quad \text{where } j = 0, \ldots, m,\ i = 0, \ldots, m - j + t,$$
$$h_{i,j}(x, y) = y^i f_e^j(x, y) e^{m-j} \quad \text{where } j = 0, \ldots, m,\ i = 1, \ldots, m - j.$$

Note that $g_{i,j}(k_1, s) \equiv 0 \pmod{e^m}$ and $h_{i,j}(k_1, s) \equiv 0 \pmod{e^m}$. We call $g_{i,j}$ the x-shift and $h_{i,j}$ the y-shift polynomials, as per construction.

Next, we form a lattice L by taking the coefficient vectors of the shift polynomials $g_{i,j}(xX, yY)$ and $h_{i,j}(xX, yY)$ as basis. One can verify that the dimension of the lattice L is $\omega = (m+1)^2 + t(m+1)$. The matrix L_M, containing the basis vectors of L, is lower triangular and has diagonal entries of the form

$$\begin{aligned} &X^{i+j}Y^je^{m-j} \text{ for } j = 0, \ldots, m \text{ and } i = 0, \ldots, m-j+t, \text{ and} \\ &X^jY^{i+j}e^{m-j} \text{ for } j = 0, \ldots, m \text{ and } i = 1, \ldots, m-j, \end{aligned}$$

coming from $g_{i,j}$ and $h_{i,j}$ respectively. Thus, one can calculate the determinant of L as

$$\det(L) = \left[\prod_{j=0}^{m}\prod_{i=0}^{m-j+t} X^{i+j}Y^je^{m-j}\right]\left[\prod_{j=0}^{m}\prod_{i=1}^{m-j} X^jY^{i+j}e^{m-j}\right] = X^{s_1}Y^{s_2}e^{s_3}$$

where

$$\begin{aligned} s_1 &= \frac{1}{2}m^3 + m^2 + \frac{1}{2}m + \frac{1}{2}t^2 + \frac{1}{2}t + m^2t + \frac{1}{2}mt^2 + \frac{3}{2}mt, \\ s_2 &= \frac{1}{2}m^3 + m^2 + \frac{1}{2}m + \frac{1}{2}m^2t + \frac{1}{2}mt, \text{ and} \\ s_3 &= \frac{2}{3}m^3 + \frac{1}{2}m^2t + \frac{3}{2}m^2 + \frac{1}{2}mt + \frac{5}{6}m. \end{aligned}$$

To utilize Gröbner basis techniques and Assumption 1, we need two polynomials $f_1(x, y)$, $f_2(x, y)$ which share the roots (k_1, s) over integers. From Lemma 1 and Fact 1, we know that one can find such $f_1(x, y)$, $f_2(x, y)$ using LLL lattice reduction algorithm over L when

$$2^{\frac{\omega(\omega-1)}{4}} (det(L))^{\frac{1}{\omega-1}} < \frac{e^m}{\sqrt{\omega}}.$$

Now, for large N, e we have $(det(L))^{\frac{1}{\omega-1}}, e$ is much larger than $2^{\frac{\omega(\omega-1)}{4}}$, $\sqrt{\omega}$. Hence we approximate the required condition by $det(L) < e^{m(\omega-1)}$. Given the values of $det(L)$ and ω obtained above, we get the required condition as $X^{s_1}Y^{s_2}e^{s_3} < e^{m\left((m+1)^2+t(m+1)-1\right)}$, i.e., $X^{s_1}Y^{s_2} < e^{s_0}$, where

$$\begin{aligned} s_0 &= m\left((m+1)^2 + t(m+1) - 1\right) - s_3 \\ &= \frac{1}{3}m^3 + \frac{1}{2}m^2 - \frac{5}{6}m + \frac{1}{2}m^2t + \frac{1}{2}mt. \end{aligned}$$

Now putting the values of the bounds $X = N^\lambda, Y = N^{0.5}$ in $X^{s_1}Y^{s_2} < e^{s_0}$, and considering e to be $\Theta(N)$, we get the condition as

$$\begin{aligned} &\lambda\left(\frac{m^3}{2} + m^2 + \frac{m}{2} + \frac{t^2}{2} + \frac{t}{2} + m^2t + \frac{mt^2}{2} + \frac{3mt}{2}\right) + \\ &\frac{1}{2}\left(\frac{m^3}{2} + m^2 + \frac{m}{2} + \frac{m^2t}{2} + \frac{mt}{2}\right) < \frac{m^3}{3} + \frac{m^2}{2} - \frac{5m}{6} + \frac{m^2t}{2} + \frac{mt}{2}. \qquad (1) \end{aligned}$$

From Equation (1) we get the required bound for λ as follows:

$$\lambda < \frac{\frac{1}{12}m^3 - \frac{13}{12}m + \frac{1}{4}m^2t + \frac{1}{4}mt}{\frac{1}{2}m^3 + m^2 + \frac{1}{2}m + \frac{1}{2}t^2 + \frac{1}{2}t + m^2t + \frac{1}{2}mt^2 + \frac{3}{2}mt}.$$

Now, one can find the root (k_1, s) from f_1, f_2 under Assumption 1. The claimed time complexity of $\text{poly}(\log N)$ can be achieved because

- the time complexity of the LLL lattice reduction is $\text{poly}(\log N)$; and
- given a fixed lattice dimension of small size, we get constant degree polynomials and the Gröbner Basis calculation is in general double-exponential in the degree of the polynomial.

This completes the proof of Theorem 1. □

Let us illustrate the lattice generation technique for $m = 3, t = 0$. We use the shift polynomials e^3, xe^3, ye^3, fe^2, x^2e^3, xfe^2, x^2e^3, xfe^2, x^3e^3, x^2fe^2, y^2e^3, yfe^2, f^2e, xf^2e, y^3e^3, y^2fe^2, yf^2e, f^3 and build the following lattice L with the basis elements coming from the coefficients of these shift polynomials, as discussed before. In this case, the lattice dimension turns to be $(m+1)^2+t+mt = 16$. The '$-$' marked places contain non-zero elements, but we do not write those as those elements do not contribute in the calculation of the determinant.

poly	1	x	y	xy	x^2	x^2y	x^3	x^3y	y^2	xy^2	x^2y^2	x^3y^2	y^3	xy^3	x^2y^3	x^3y^3
e^3	e^3															
xe^3		Xe^3														
ye^3			Ye^3													
fe^2	$-$	$-$	$-$	XYe^2												
x^2e^3					X^2e^3											
xfe^2		$-$		$-$	$-$	X^2Ye^2										
x^3e^3							X^3e^3									
x^2fe^2					$-$	$-$	$-$	X^3Ye^2								
y^2e^3									Y^2e^3							
yfe^2			$-$	$-$					$-$	XY^2e^2						
f^2e	$-$	$-$	$-$	$-$	$-$	$-$			$-$	$-$	X^2Y^2e					
xf^2e		$-$		$-$	$-$	$-$	$-$	$-$		$-$	$-$	X^3Y^2e				
y^3e^3													Y^3e^3			
y^2fe^2									$-$	$-$			$-$	XY^3e^2		
yf^2e			$-$	$-$		$-$			$-$	$-$	$-$		$-$	$-$	X^2Y^3e	
f^3	$-$	$-$	$-$	$-$	$-$	$-$	$-$	$-$	$-$	$-$	$-$	$-$	$-$	$-$	$-$	X^3Y^3

The technique of Ernst et al. [8] as well as our strategy explained in the proof of Theorem 1 fall under the generalized strategy presented in Jochemsz and May [10].

In [8], Ernst et al. present two methods for lattice based cryptanalysis of RSA with partial key exposure. In Method I, dimension of the proposed lattice is $\omega_1 = (\frac{m^2}{2} + \frac{5m}{2} + 3)t + \frac{m^3}{6} + \frac{3}{2}m^2 + \frac{13}{3}m + 4$ and the technique will be successful for

$$\gamma < \frac{\left(\frac{1}{12} - \frac{1}{6}\delta\right)m^3 + \frac{1}{4}m^2t - \frac{1}{4}mt^2 + \left(\frac{1}{2} - \delta\right)m^2 + \frac{1}{2}mt - \frac{1}{2}t^2 + \left(\frac{5}{12} - \frac{17}{6}\delta\right)m - \frac{1}{2}t - 2\delta - \frac{1}{2}}{\frac{1}{6}m^3 + \frac{1}{2}m^2t + m^2 + \frac{3}{2}mt + \frac{17}{6}m + t + 1} \qquad (2)$$

In case of Method II of [8], dimension of the corresponding lattice is $\omega_2 = (\frac{1}{2}m^2 + \frac{5}{2}m + 3)t + \frac{1}{3}m^3 + \frac{5}{2}m^2 + \frac{37}{6}m + 5$ and the required condition for success, with $\delta \leq \frac{11}{16}$, is

$$\gamma < \frac{\frac{1}{12}m^3 + \frac{1}{4}m^2t + \frac{1}{4}m^2 + \frac{3}{4}mt - \frac{1}{3}m + \frac{1}{2}t - 1}{\frac{1}{2}m^3 + m^2t + \frac{1}{2}mt^2 + \frac{5}{2}m^2 + \frac{7}{2}mt + t^2 + 6m + 4t + 3} \tag{3}$$

In case of our method, the corresponding lattice dimension of $\omega = mt + t + m^2 + 2m + 1$ produces equivalent results if $\lambda = \max\{\gamma, \delta - \frac{1}{2}\}$, and

$$\lambda < \frac{\frac{1}{12}m^3 - \frac{13}{12}m + \frac{1}{4}m^2t + \frac{1}{4}mt}{\frac{1}{2}m^3 + m^2 + \frac{1}{2}m + \frac{1}{2}t^2 + \frac{1}{2}t + m^2t + \frac{1}{2}mt^2 + \frac{3}{2}mt}. \tag{4}$$

At this point, let us present some numerical values of m, t, as in Table 1, that clearly show that theoretical bound presented in Theorem 1 is better than that of Ernst et al. [8] for similar lattice dimension. Larger values of γ in our case indicate that we need to know less amount of MSBs of the decryption exponent d. Moreover, the negative values of γ in case of Method I of [8] suggests that it is not theoretically possible to get desired results for the corresponding values of (m, t).

Table 1. Comparison of our theoretical results with that of [8] for some specific m, t

δ	Our			Method I of [8]			Method II of [8]		
	γ	(m, t)	LD	γ	(m, t)	LD	γ	(m, t)	LD
0.45	0.158	(10, 4)	165	0.082	(6, 2)	192	0.118	(5,1)	168
0.45	0.160	(11, 4)	192	0.099	(7, 3)	300	0.132	(5,2)	196
0.5	0.143	(7, 3)	88	-0.007	(4, 2)	98	0.107	(4,1)	112
0.55	0.162	(11, 5)	204	-0.012	(7, 1)	210	0.126	(6,1)	240

In view of the above data, our method proves to be considerably efficient in terms of the lattice dimension as well. One can observe that our method offers same or better values of γ compared to [8, Method I] or [8, Method II] with a considerably lower lattice dimension. The reason, as already mentioned in the Introduction, is that we use Coppersmith's [5] idea for solving the modular polynomial, while [8] used Coron's [6] version. Thus, the lattice dimension they obtained was a cubic in m whereas we obtain a quadratic in m (as t is linear in m).

Note that the maximum bit size of an entry corresponding to x shift is $X^{m+t}N^m$ and the maximum bit size of an entry corresponding to y shift is $Y^m e^m$ in our lattice. These bounds are of the same (or lower) size as those in case of the lattice constructed by Ernst et al. [8] in most of the cases. Hence, a smaller lattice dimension in our case will automatically imply better efficiency. It is worth noticing that comparatively smaller lattice dimension for same values of m, t allows us to tune these parameters to higher values and obtain better results at the same cost.

As it is generally studied in cryptanalytic materials, we also obtain the asymptotic bounds for our technique as follows.

Corollary 1. *Consider the RSA equation* $ed \equiv 1 \pmod{\phi(N)}$*. Let* $d = N^{\delta}$ *and* e *be* $\Theta(N)$*. Suppose we know an integer* d_0 *such that* $|d - d_0| < N^{\gamma}$*. Then one can factor* N *in poly*$(\log N)$ *time under Assumption 1 when* $\lambda < \frac{3}{16}$*, where* $\lambda = \max\{\gamma, \delta - \frac{1}{2}\}$*.*

Proof. Putting $t = \tau m$ and neglecting $o(m^3)$ terms in Equation (1), we get

$$\frac{1}{2}\tau^2\lambda + \left(\lambda - \frac{1}{4}\right)\tau + \left(\frac{1}{2}\lambda - \frac{1}{12}\right) < 0.$$

Substituting the optimal value of $\tau = \frac{1}{\lambda}\left(\frac{1}{4} - \lambda\right)$, we get the required condition as $\lambda < \frac{3}{16}$. □

The corresponding asymptotic bounds for the methods proposed by Ernst et al. [8] are

- Method I: $\gamma < \frac{5}{6} - \frac{1}{3}\sqrt{1 + 6\delta}$,
- Method II (1st result): $\gamma < \frac{3}{16}$ and $\delta \leq \frac{11}{16}$,
- Method II (2nd result): $\gamma < \frac{1}{3} + \frac{1}{3}\delta - \frac{1}{3}\sqrt{4\delta^2 + 2\delta - 2}$ and $\delta \geq \frac{11}{16}$.

On the other hand, cryptanalysis using our method is possible when $\lambda < \frac{3}{16}$, with $\lambda = \max\{\gamma, \delta - \frac{1}{2}\}$. As $\lambda < \frac{3}{16}$, we have $\gamma < \frac{3}{16}$ and $\delta - \frac{1}{2} < \frac{3}{16}$, that is, $\delta < \frac{11}{16}$.

Thus our result and Method II (1st result) are of same quality in terms of asymptotic bound when $\delta < \frac{11}{16} = 0.6875$. However, when $\delta < 0.4590$, then the bound on γ using Method I of [8] is $\geq \frac{3}{16}$, and our result is worse than that of [8] in this case. Hence, our asymptotic results are of the same quality as the work of Ernst et al. [8] for $N^{0.4590} \leq d < N^{0.6875}$.

But in experimental situations, our result is better than that of [8] for $d \leq N^{0.64}$. These experimental advantages are detailed in Section 4.

3 Further Improvement Using Sublattice

From the experimental results of Ernst et al. [8, Method II], one may note that for small values of δ (e.g., $\delta = 0.3$), the experimental results are better than the theoretical bounds. This happens in case of our experiments as well. Our method suggests the theoretical bound

$$\lambda < \frac{\frac{1}{12}m^3 - \frac{13}{12}m + \frac{1}{4}m^2t + \frac{1}{4}mt}{\frac{1}{2}m^3 + m^2 + \frac{1}{2}m + \frac{1}{2}t^2 + \frac{1}{2}t + m^2t + \frac{1}{2}mt^2 + \frac{3}{2}mt}.$$

When, $t = 0$, we have $\lambda < \frac{\frac{1}{12}m^3 - \frac{13}{12}m}{\frac{1}{2}m^3 + m^2 + \frac{1}{2}m} < \frac{1}{6} \approx 0.167$, for all m. But the experimental evidences for $t = 0$ in the range $\delta = 0.3$ and $\delta = 0.35$ are clearly better. This is because, for these parameters, the shortest vectors may belong to some sub-lattice. However, the theoretical calculation in [8] as well as in our Theorem 1 cannot capture that. Further, identifying such optimal sub-lattice seems to be difficult as pointed out by Jochemsz and May [11, Section 7.1]. In this section, we propose a strategy to obtain better experimental results using a special structure of the sublattice.

Our strategy: Recall our construction of the lattice L in Section 2. The rows of the matrix L_M corresponding to L came from the coefficients of $g_{i,j}(xX, yY)$ and $h_{i,j}(xX, yY)$, where

$$g_{i,j}(x,y) = x^i f_e^j(x,y)e^{m-j} \quad \text{with } j = 0, \ldots, m,\ i = 0, \ldots, m-j+t,$$
$$h_{i,j}(x,y) = y^i f_e^j(x,y)e^{m-j} \quad \text{with } j = 0, \ldots, m,\ i = 1, \ldots, m-j.$$

The strategy for constructing a sublattice is to keep the x-shift portion of L_M unchanged and judiciously delete a few rows from the y-shift portion of L_M to produce a new matrix $L_M^{'}$. We propose deleting the rows generated by $h_{i,j} = y^i f_e^j e^{m-j}$, where $j = 0, \ldots, m$ and $i = 2, \ldots, m-j$. In other words, the new matrix $L_M^{'}$ can be constructed from the shift polynomials $g_{i,j}(xX, yY)$ and $h_{i,j}(xX, yY)$, where

$$g_{i,j}(x,y) = x^i f_e^j(x,y)e^{m-j} \quad \text{with } j = 0, \ldots, m,\ i = 0, \ldots, m-j+t,$$
$$h_{i,j}(x,y) = y f_e^j(x,y)e^{m-j} \quad \text{with } j = 0, \ldots, m-1.$$

Let $L^{'}$ be the lattice defined by $L_M^{'}$. As all the rows of $L_M^{'}$ come from L_M, $L^{'}$ is a sublattice of L and we propose $L^{'}$ to be our chosen sublattice.

One may easily calculate that the number of rows of the sublattice is $\omega_R^{'} = \frac{1}{2}(m+1)(m+2) + m + t(m+1)$. Hence, we obtain a substantial reduction of $\frac{1}{2}m(m-1)$ in terms of lattice dimension, which makes the LLL operation considerably faster. Experiments show that applying LLL to $L^{'}$ (with lower lattice dimension) yield results of same quality as those in case of L as shown in Table 4 in Section 4.

Let us illustrate the strategy for choosing a sublattice in case of $m = 3, t = 0$. Please refer back to Section 2 for our original lattice having 16 rows. Here, following our strategy, we delete rows 9, 11, 12 from top and obtain the following sublattice. The reduction in number of rows in this case is $\frac{1}{2}m(m-1) = 3$, as expected. This reduction produces considerably better results in practice as higher values of m can be used. For example, number of rows reduces to 43 from 64 in case of $m = 7, t = 0$.

poly	1	x	y	xy	x^2	x^2y	x^3	x^3y	y^2	xy^2	x^2y^2	x^3y^2	y^3	xy^3	x^2y^3	x^3y^3
e^3	e^3															
xe^3		Xe^3														
ye^3			Ye^3													
fe^2	–	–	–	XYe^2												
x^2e^3					X^2e^3											
xfe^2		–		–	–	X^2Ye^2										
x^3e^3							X^3e^3									
x^2fe^2					–	–	–	X^3Ye^2								
yfe^2			–	–					–	XY^2e^2						
f^2e	–	–	–	–	–	–			–	–	X^2Y^2e					
xf^2e		–		–	–	–	–	–		–	–	X^3Y^2e				
yf^2e			–	–		–			–	–	–		–	–	X^2Y^3e	
f^3	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	X^3Y^3

It is also worth noting that this reduction in dimension allows us some extra x-shifts by increasing the value of t, which improve our results even further.

Note that our choice of sublattice is purely heuristic at this point and it will be interesting if one can furnish the theoretical justification for this strategy. We have noted that the idea of [9] cannot be immediately exploited to theoretically capture the sublattice structure.

The main motivation of exploring the idea of sublattice is the observation that experimental results perform better than theoretical bounds. This happens for low values of d. During experimentation, we indeed observed that improved results are obtained for $d = N^{0.3}, N^{0.35}$ using sublattices. However, for $d \geq N^{0.4}$, we could not achieve any improvement using the sublattice based technique over our lattice based technique.

4 Experimental Results

We have implemented the code in SAGE 4.1 on a Linux Ubuntu 8.10, Dual CORE Intel(R) Pentium(R) D CPU 1.83 GHz, 2 GB RAM, 2 MB Cache machine. Let us present two examples to explain our improvements.

Example 1. We consider 500 bits p, q, i.e., 1000 bits $N = pq$. The exponent e is of 1000 bits and d is of 300 bits. The details of p, q, e, d are available in Appendix A. The idea of [8, Method I] has been implemented on our platform and we get the following comparison which shows that our method is more efficient. By LD, we mean the Lattice Dimension.

Method	m, t, LD	MSBs of d to be known	Time (seconds)
Method I of [8]	2, 2, 40	95	30.22
Our (Lattice)	5, 0, 36	75	6.15
Method I of [8]	3, 1, 50	75	451.42
Our (Lattice)	6, 0, 49	66	26.72
Method I of [8]	4, 2, 98	66	9101.23
Our (Lattice)	7, 0, 64	63	104.57
Our (Sublattice)	7, 0, 43	63	49.66

For lattice dimension 98, using [8, Method I], successful result could not be achieved when 63 MSBs are available. □

Example 2. We take the same p, q as in Example 1, and consider 1000-bit e and 600-bit d. The details of p, q, e, d are given in Appendix A. We implemented the idea of [8, Method II] on our platform to get the following comparison, which shows the efficiency of our method.

Method	m, t, LD	MSBs of d to be known	Time (seconds)
Method II of [8]	2, 2, 50	491	82.47
Method II of [8]	3, 1, 70	477	618.86
Our method	5, 0, 36	467	12.34
Our method	6, 0, 49	459	67.02
Our method	6, 1, 56	451	197.68

In these cases, we have checked that our sublattice based technique does not provide any improvement over the general method. This is because there are probably no sublattice structures to improve the bound of γ. □

Table 2. Experimental results for Method I (left) and Method II (right) of [8] for 1000 bit N in our implementation. LLL time is presented in seconds.

δ	γ	γ (expt.), $m = 1$			γ (expt.), $m = 2$		
	asym.	$t = 0$	$t = 1$	$t = 2$	$t = 0$	$t = 1$	$t = 2$
0.30	0.28	0.194	0.195	0.199	0.209	0.209	0.210
0.35	0.25	0.136	0.148	0.153	0.142	0.159	0.158
0.40	0.22	0.097	0.117	0.114	0.096	0.140	0.139
0.45	0.19	0.048	0.100	0.098	0.047	0.117	0.117
0.50	0.17	0	0.083	0.083	0	0.098	0.111
0.55	0.14	0	0.081	0.083	0	0.086	0.108
0.60	0.12	0	0.045	0.048	0	0.061	0.105
0.638	0.10	0	0	0	0	0.013	0.069
0.65	0.10	0	0	0	0	0	0.055
0.70	0.07	0	0	0	0	0	0
0.75	0.05	0	0	0	0	0	0
0.80	0.03	0	0	0	0	0	0
0.85	0.01	0	0	0	0	0	0
Lattice dim.		10	16	22	20	30	40
LLL time		<1	1	3	5	11	29

δ	γ	γ (expt.), $m = 1$				γ (expt.), $m = 2$		
	asym.	$t = 0$	$t = 1$	$t = 2$	$t = 3$	$t = 0$	$t = 1$	$t = 2$
0.30	0.19	0.197	0.197	0.198	0.194	0.192	0.193	0.201
0.35	0.19	0.147	0.147	0.146	0.143	0.158	0.159	0.158
0.40	0.19	0.116	0.119	0.120	0.124	0.139	0.140	0.140
0.45	0.19	0.101	0.109	0.117	0.115	0.120	0.129	0.135
0.50	0.19	0.084	0.111	0.120	0.118	0.109	0.123	0.133
0.55	0.19	0.081	0.110	0.116	0.118	0.109	0.122	0.134
0.60	0.19	0.052	0.109	0.115	0.121	0.112	0.124	0.132
0.638	0.19	0	0.074	0.078	0.082	0.076	0.110	0.123
0.65	0.19	0	0.058	0.058	0.060	0.060	0.096	0.106
0.70	0.18	0	0	0	0	0	0.048	0.051
0.75	0.14	0	0	0	0	0	0	0
0.80	0.11	0	0	0	0	0	0	0
0.85	0.08	0	0	0	0	0	0	0
0.90	0.05	0	0	0	0	0	0	0
0.95	0.03	0	0	0	0	0	0	0
Lattice dim.		14	20	26	32	30	40	50
LLL time		< 1	2	4	37	9	50	415

We present the results for 1000 bit N here because RSA moduli of this order are used in practice. We also detail the comparison of results of our method with that of [8] for 256 bit N in Appendix B. We experiment with Methods I, II of [8] on our platform as results for 1000 bits N are not available in the paper [8]. These results are presented in Table 2. We present the experimental results of our lattice based technique for 1000 bit N in Table 3.

Our results (presented in Table 3) are better than that of [8] (presented in Table 2) for $\delta \leq 0.638$. In these cases the experiments we performed were always successful. Beyond that bound, not every attempt with the lattice dimensions mentioned in Table 2 was successful. However, we successfully reached the range $\delta = 0.64$ in some of our experiments. The results can be further improved with higher lattice dimensions.

In Table 4 we present the improvements using our sublattice based technique for small values of d. Improved results are obtained only for $\delta = 0.3, 0.35$, as we discussed before.

Table 3. Experimental result of our lattice based method for 1000 bit N

δ	γ	$m = 4, t = 0$	$m = 5, t = 0$	$m = 6, t = 0$
	asympt.	expt.	expt.	expt.
0.30	0.19	0.211	0.226	0.232
0.35	0.19	0.178	0.191	0.194
0.40	0.19	0.152	0.162	0.169
0.45	0.19	0.135	0.145	0.154
0.50	0.19	0.124	0.134	0.144
0.55	0.19	0.125	0.134	0.141
0.60	0.19	0.127	0.133	0.141
0.638	0.19	0	0	0.142
Lattice dimension		25	36	49
LLL time (in sec)		3	14	100

Table 4. Experimental result of our sublattice based method for 1000 bit N

δ	$m = 4, t = 0$	$m = 5, t = 0$	$m = 6, t = 0$	$m = 7, t = 0$	$m = 8, t = 0$
0.30	0.211	0.226	0.232	0.235	0.237
0.35	0.178	0.191	0.193	0.196	0.199
Sublattice dimension	19	26	34	43	53
LLL time (in sec)	< 1	3	14	50	140

5 Conclusion

In this paper we consider the partial key exposure attack on RSA and provide better results than what were obtained by Ernst et al. [8], for certain parameters. We present experimental evidences to show how our technique improves those of [8] in the following ways:

- we provide better efficiency at smaller lattice dimensions in practice,
- our method offers similar asymptotic results for certain range of δ,
- we propose a strategy for constructing sublattice to improve the efficiency even further.

We would like to clarify that the practical advantages we obtain over [8] are due to using Coppersmith's techniques (for modular polynomials) instead of Coron's idea (for integer polynomials), as predicted in [8].

Our work puts forward two natural open problems. The first is to improve the range of δ for our improvements over the work of Ernst et al. [8]. The second open problem would be to provide a theoretical model for constructing the sublattice or a formal justification of our heuristic sublattice strategy. This will further improve the bounds of γ within a certain range of δ, as expected from our experimental observations.

Acknowledgments. The authors are grateful to the anonymous reviewers and the shepherd for their invaluable comments and suggestions that helped in improving the technical as well as the editorial quality of the paper.

References

1. Aono, Y.: A New Lattice Construction for Partial Key Exposure Attack for RSA. In: Jarecki, S., Tsudik, G. (eds.) Public Key Cryptography – PKC 2009. LNCS, vol. 5443, pp. 34–53. Springer, Heidelberg (2009)
2. Blömer, J., May, A.: New Partial Key Exposure Attacks on RSA. In: Boneh, D. (ed.) CRYPTO 2003. LNCS, vol. 2729, pp. 27–43. Springer, Heidelberg (2003)
3. Boneh, D., Durfee, G., Frankel, Y.: Exposing an RSA Private Key Given a Small Fraction of its Bits. In: Ohta, K., Pei, D. (eds.) ASIACRYPT 1998. LNCS, vol. 1514, pp. 25–34. Springer, Heidelberg (1998)
4. Cohen, H.: A Course in Computational Algebraic Number Theory. Springer, Heidelberg (1996)
5. Coppersmith, D.: Small Solutions to Polynomial Equations and Low Exponent Vulnerabilities. Journal of Cryptology 10(4), 223–260 (1997)

6. Coron, J.-S.: Finding Small Roots of Bivariate Integer Equations Revisited. In: Cachin, C., Camenisch, J.L. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 492–505. Springer, Heidelberg (2004)
7. Cox, D., Little, J., O'Shea, D.: Ideals, Varieties, and Algorithms: An Introduction to Computational Algebraic Geometry and Commutative Algebra, 3rd edn. Springer, New York (2007)
8. Ernst, M., Jochemsz, E., May, A., de Weger, B.: Partial Key Exposure Attacks on RSA up to Full Size Exponents. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 371–386. Springer, Heidelberg (2005)
9. Herrmann, M., May, A.: Maximizing Small Root Bounds by Linearization and Applications to Small Secret Exponent RSA. In: Nguyen, P.Q., Pointcheval, D. (eds.) Public Key Cryptography – PKC 2010. LNCS, vol. 6056, pp. 53–69. Springer, Heidelberg (2010)
10. Jochemsz, E., May, A.: A Strategy for Finding Roots of Multivariate Polynomials with new Applications in Attacking RSA Variants. In: Lai, X., Chen, K. (eds.) ASIACRYPT 2006. LNCS, vol. 4284, pp. 267–282. Springer, Heidelberg (2006)
11. Jochemsz, E., May, A.: A Polynomial Time Attack on RSA with Private CRT-Exponents Smaller Than $N^{0.073}$. In: Menezes, A. (ed.) CRYPTO 2007. LNCS, vol. 4622, pp. 395–411. Springer, Heidelberg (2007)
12. Kocher, P.: Timing attacks on implementations of Diffie-Hellman, RSA, DSS and other systems. In: Koblitz, N. (ed.) CRYPTO 1996. LNCS, vol. 1109, pp. 104–113. Springer, Heidelberg (1996)
13. Kocher, P., Jaffe, J., Jun, B.: Differential power analysis. In: Wiener, M. (ed.) CRYPTO 1999. LNCS, vol. 1666, pp. 388–397. Springer, Heidelberg (1999)
14. Lenstra, A.K., Lenstra, H.W., Lovász, L.: Factoring Polynomials with Rational Coefficients. Mathematische Annalen 261, 513–534 (1982)
15. Rivest, R.L., Shamir, A., Adleman, L.: A Method for Obtaining Digital Signatures and Public Key Cryptosystems. Communications of ACM 21(2), 158–164 (1978)
16. Sarkar, S., Maitra, S.: Improved Partial Key Exposure Attacks on RSA by Guessing a Few Bits of One of the Prime Factors. In: Lee, P.J., Cheon, J.H. (eds.) ICISC 2008. LNCS, vol. 5461, pp. 37–51. Springer, Heidelberg (2009)

Appendix A

Details of Example 1

We consider 500 bits p, q, i.e., 1000 bits $N = pq$. The primes p, q are

291208439798748853010189716174472934796239231793982589697372703133
261175039299368917615319875150921817582610829859036449324861513036
8371602851843098873, and

265778313902463222317152253854708224764165033731853280284472891789
062446101991648433388961946084205843676898740199746957586412762306
7945908748061533789.

The exponents e (1000 bit) and d (300 bit) are respectively

6336592072697636916143183091126456156249373448914681680770114707307345673704169135132242996825746665043587681813217139957921752851107827093765230755870784287989301300080332740805893387704531628007459742805317603659777158458744950023018031526899032667410842692995720687649380410410817145354934370336339, and

1814845758605508107926070456170619634431455166454166878249390985115060612463635344515223099.

Details of Example 2

We take the same p, q as in Example 1, and e (1000 bit), d (600 bit) are

7558665224136375374055690448131132089893236520631697389659469251681205924949978104059590671041756882393215403400574000254028912282454283048431737459758911375883732281559627012563046500414822556109522044727622686588774322310244883084236536267052528517810444320170108273107293503245751392242116148600457, and

3104671629545245370523383160723009827881009018135015555613090899788913464879693929048110577006975062601951688111024769296588117880950937431880308630354291294168946380568110823581913.

Appendix B

Here we present the experimental results for 256 bit N in tabular form to compare our results with that of [8]. In Table 5, we reproduce the results of [8, Fig. 5, 6] when N is of 256 bits. We add one extra row of data containing the run time of the program to show how the implementation of the techniques of [8] works on our platform.

Table 5. Experimental results for the techniques of [8] for 256 bit N. In the table on the left, LLL time A is the data given for Method I in [8, Fig. 6] and LLL time B is the data from our implementation for Method II of [8]. In the table on the right, LLL time A is the data given for Method II in [8, Fig. 6] and LLL time B is the data from our implementation for Method II of [8]. All the LLL times are given in seconds.

δ	γ	γ (expt.), $m = 1$			γ (expt.), $m = 2$		
	asym.	$t = 0$	$t = 1$	$t = 2$	$t = 0$	$t = 1$	$t = 2$
0.30	0.28	0.19	0.19	0.19	0.19	0.21	0.21
0.35	0.25	0.13	0.14	0.14	0.14	0.16	0.16
0.40	0.22	0.09	0.11	0.11	0.09	0.14	0.15
0.45	0.19	0.04	0.10	0.10	0.05	0.12	0.12
0.50	0.17	0	0.08	0.09	0	0.10	0.11
0.55	0.14	0	0.08	0.08	0	0.09	0.11
0.60	0.12	0	0.04	0.04	0	0.06	0.10
0.65	0.10	0	0	0	0	0	0.06
0.70	0.07	0	0	0	0	0	0.01
0.75	0.05	0	0	0	0	0	0
0.80	0.03	0	0	0	0	0	0
0.85	0.01	0	0	0	0	0	0
Lattice dim.		10	16	22	20	30	40
LLL time A		1	2	8	3	25	100
LLL time B		<1	<1	<1	<1	2	4

δ	γ	γ (expt.), $m = 1$				γ (expt.), $m = 2$		
	asym.	$t = 0$	$t = 1$	$t = 2$	$t = 3$	$t = 0$	$t = 1$	$t = 2$
0.30	0.19	0.19	0.20	0.20	0.20	0.19	0.19	0.19
0.35	0.19	0.15	0.16	0.16	0.16	0.16	0.16	0.16
0.40	0.19	0.12	0.12	0.12	0.12	0.14	0.15	0.15
0.45	0.19	0.10	0.11	0.12	0.12	0.12	0.13	0.13
0.50	0.19	0.08	0.11	0.12	0.12	0.12	0.13	0.13
0.55	0.19	0.08	0.11	0.11	0.11	0.11	0.12	0.13
0.60	0.19	0.05	0.11	0.11	0.11	0.11	0.12	0.13
0.65	0.19	0	0.05	0.06	0.06	0.05	0.08	0.10
0.70	0.18	0	0	0	0	0	0.04	0.05
0.75	0.14	0	0	0	0	0	0	0
0.80	0.11	0	0	0	0	0	0	0
0.85	0.08	0	0	0	0	0	0	0
0.90	0.05	0	0	0	0	0	0	0
0.95	0.03	0	0	0	0	0	0	0
LLL time A		1	7	17	32	30	40	50
LLL time B		< 1	< 1	< 1	5	< 1	6	43

Table 6. Experimental results for our method for 256 bit N

δ	γ	$m = 4, t = 0$	$m = 5, t = 0$	$m = 6, t = 0$
	asympt.	expt.	expt.	expt.
0.30	0.19	0.211	0.219	0.227
0.35	0.19	0.172	0.184	0.195
0.40	0.19	0.145	0.160	0.164
0.45	0.19	0.133	0.141	0.156
0.50	0.19	0.121	0.129	0.137
0.55	0.19	0.117	0.133	0.137
0.60	0.19	0.117	0.129	0.145
0.625	0.19	0	0.109	0.137
Lattice dimension		25	36	49
LLL time (in sec)		<1	1	5

Next we present our results when N is of 256 bits in Table 6. One may note that our results are better than that of [8] (presented in Table 5) for $\delta \leq 0.625$. The experimental data till $\delta \leq 0.625$ is presented based on that fact that we are always successful to factorize N in experiments following the idea of Theorem 1.