# RSA Cryptanalysis with Increased Bounds on the Secret Exponent using Less Lattice Dimension

Santanu Sarkar, Subhamoy Maitra and Sumanta Sarkar

Indian Statistical Institute, 203 B T Road, Kolkata 700 108, India
{santanu_r, subho, sumanta_r}@isical.ac.in

**Abstract.** We consider RSA with $N = pq$, $q < p < 2q$, public encryption exponent $e$ and private decryption exponent $d$. Boneh and Durfee (Eurocrypt 1999, IEEE-IT 2000) used Coppersmith's method (Journal of Cryptology, 1997) to factorize $N$ using $e$ when $d < N^{0.292}$, the theoretical bound. However, the experimental bound that has been reached so far is only $N^{0.280}$ for 1000 bits integers (and less for higher number of bits). The basic idea relied on LLL algorithm, but the experimental bounds were constrained by large lattice dimensions. In this paper we present theoretical results and experimental evidences to extend the bound of $d$ for which RSA is weak. This requires the knowledge of a few most significant bits of $p$ (alternatively these bits need to be searched exhaustively). We provide experimental results to highlight that the problem can be solved with low lattice dimensions in practice. Our results outperform the existing experimental results by increasing the bounds of $d$ and also we provide clear evidence that RSA with 1000 bit $N$ and $d$ of the order of $N^{0.3}$ can be cryptanalysed in practice from the knowledge of $N, e$.

**Keywords:** Cryptanalysis, Factorization, Lattice, LLL Algorithm, RSA, Weak Keys.

## 1 Introduction

RSA [13] is one of the most popular cryptosystems in the history of cryptology. Here, we use the standard notations in RSA as follows: primes $p, q$, with $q < p < 2q$; $N = pq$, $\phi(N) = (p-1)(q-1)$; $e, d$ are such that $ed = 1 + k\phi(N)$, $k \geq 1$; $N, e$ are available in public and the message $M$ is encrypted as $C = M^e \bmod N$; the secret key $d$ is required to decrypt the message as $M = C^d \bmod N$.

Wiener [17] showed that if one uses $d < \frac{1}{3}N^{0.25}$, then RSA is insecure. Boneh and Durfee [3] extended this bound up to $d < N^{0.292}$ using Coppersmith's technique [6]. There exist considerable amount of references in the literature where the bound on $d$ is increased till $O(N^{0.5})$ depending on different constraints on the differences of primes or the values of $d, e$ (see [2, 18, 12] and the references therein). However, there is no general result and experimental evidence where the bound for $d$ can be increased exceeding $O(N^{\delta})$, where for example $\delta = 0.3$. Here we present ideas to achieve better bounds on $\delta$ such that $N$ can be factorized from the knowledge of $e$ when $d$ is $O(N^{\delta})$. This is to note that in [15], it has been clearly pointed out that Wiener's method cannot be extended with good efficiency beyond $d$ of the order of $N^{0.25}$. In [8], RSA cryptanalysis has been studied following the idea of [6], where it was considered that some bits of $d$ are known.

In this paper we concentrate on the existing techniques [3, 4, 1] with the idea that a few MSBs of the prime $p$ is known. We consider that some estimate $p_0$ of $p$ is known such that

$|p - p_0| < N^{\gamma}$, $\gamma \leq \frac{1}{2}$. That is $(\frac{1}{2} - \gamma) \log_2 N$ many MSBs of $p$ are known. The other way of interpreting it is that one may need to try for $N^{\frac{1}{2}-\gamma}$ many possible options to guess the MSBs of $p$. With this idea, we find that it is possible to exceed the bound of $d$ over the works of Boneh-Durfee [3, 4] and Blömer-May [1] with low lattice dimensions as used in [3, 4, 1].

The idea of [3] uses the full rank lattice for attacking this problem. Later in [4], sub-lattices have been used for better results. This idea has been further extended in [1]. The main idea used in [3, 4, 1] and in this work relies on three important parts: (i) reduction of lattice or sub-lattice, (ii) calculation of resultant, (iii) finding roots of the resultant polynomial. The idea of using sub-lattices (with lesser lattice dimension than the full rank lattice) instead of full rank lattice provides improvements in time complexity during the first step, i.e., if sub-lattice is used instead of lattice the requirement of time is less. This we detail in experimental results. However, we have observed that the calculation of resultant needs significantly more time than the first step irrespective of using lattice or sub-lattice. This shows that though the idea of sub-lattices [4, 1] improved the bound on $d$ than in [3] theoretically, there is not much improvement in experimental results due to the overhead in calculating the resultant. This has been pointed out in [1, Section 6] too.

The outline of the paper is as follows. In Section 1.1, we briefly discuss some background materials. Next, in Section 2 we present our strategy on a theoretical framework which is in the line of [3]. Section 3 describes the complete experimental details with comparison of existing works.

### 1.1 Preliminaries

Now we briefly present some basics on basis reduction in lattice (see [3, 6] and the references therein for more details). Consider that $u_1, \ldots, u_w \in Z^n$ are linearly independent vectors with $w \leq n$. A Lattice, spanned by $< u_1, \ldots, u_w >$, is the set of all linear combinations of $u_1, \ldots, u_w$, i.e., $w$ is the dimension of the lattice. A lattice is called full rank when $w = n$. Let $L$ be a lattice spanned by linearly independent vectors $u_1, \ldots, u_w$, where $u_1, \ldots, u_w \in Z^n$. By $u_1^*, \ldots, u_w^*$, we denote the vectors obtained by applying the Gram-Schmidt process to the vectors $u_1, \ldots, u_w$. It is known that given a basis $u_1, \ldots, u_w$ of a lattice $L$, LLL algorithm can find a new basis $b_1, \ldots, b_w$ of $L$ with the following properties.

- $\| b_i^* \|^2 \leq 2 \| b_{i+1}^* \|^2$, for $1 \leq i < w$
- 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 $j$.
- $\| b_1 \| \leq 2^{\frac{w}{2}} \det(L)^{\frac{1}{w}}$, $\| b_2 \| \leq 2^{\frac{w}{2}} \det(L)^{\frac{1}{w-1}}$.

The determinant of $L$ is defined as $det(L) = \prod_{i=1}^{w} ||u_i^*||$, where $||.||$ denotes the Euclidean norm on vectors.

Let us now explain the issue of solving the small inverse problem as presented in [3]. Let $d < N^{\delta}$. We assume $e$ is same order of magnitude as $N$. As $e$ gets reduced, the Boneh-Durfee technique [3] works better. Thus for the worst case scenario, one can assume $d < e^{\delta}$. It has been noticed that $ed = 1 \bmod \frac{\phi(N)}{2}$. So $ed + k(\frac{N+1}{2} + s) = 1$, where $k \in Z$, $s = -\frac{p+q}{2}$, i.e., $k(\frac{N+1}{2} + s) - 1 = 0 \bmod e$. Let $f(x, y) = x(N + 1 - y) - 1$. We have to find $x_0, y_0$ such that $f(x_0, y_0) \equiv 0(\text{mod}e)$, where, $|x_0| < e^{\delta}$ and $|y_0| < e^{0.5}$. To find the roots, the modular equation is transformed to an equation over integers by the idea of Coppersmith [6]. Given a polynomial $g(x, y) = \sum a_{i,j}x^i y^j$, we define the norm as $\| g(x, y) \|^2 = \sum a_{i,j}^2$.

**Theorem 1.** *[9] Let $g(x, y)$ be a polynomial which is a sum of $\omega$ many monomials. Suppose $g(x_0, y_0) = 0 \bmod e^m$ for some positive integer $m$, where $|x_0| < X$ and $|y_0| < Y$. If $\| g(xX, yY) \| < \frac{e^m}{\sqrt{\omega}}$, then $g(x_0, y_0) = 0$ holds over integers.*

Following [3], one can define the polynomials $g_{i,k}(x, y) = x^i f^k(x, y)e^{m-k}$ and $h_{j,k}(x, y) = y^j f^k(x, y)e^{m-k}$ for a given positive integer $m$ and $k = 0, \ldots, m$, $i = 0, \ldots, m - k$ and $j = 0, \ldots, t$ for some positive integer $t$. Now consider the lattice $L_B$ spanned by the coefficient vectors of the polynomials $g_{i,k}(xX, yY)$ and $h_{j,k}(xX, yY)$. One can check that the basis vectors $B(m, t)$ of the lattice $L_B$ form a triangular matrix $M_B$.

Now we present the definition of geometrically progressive matrices following [4].

**Definition 1.** *Let $M$ be an $(a+1)b \times (a+1)b$ matrix. The pair $(i, j)$ corresponds to $(bi+j)$-th column of $M$. Similarly a pair $(k, l)$ can be used to index $(bk + l)$-th row of $M$.*

*Let $C, D, c_0, c_1, c_2, c_3, c_4, \beta$ be real numbers with $C, D, \beta \geq 1$. A matrix $M$ is said to be geometrically progressive with parameters $(C, D, c_0, c_1, c_2, c_3, c_4, \beta)$ if the following conditions hold for all $i, k$ in $[0, \ldots, a]$ and for all $j, l$, in $[1, \ldots, b]$:*

- $|M(i, j, k, l)| \leq C \cdot D^{c_0+c_1 i+c_2 j+c_3 k+c_4 l}$,
- $M(k, l, k, l) = D^{c_0+c_1 k+c_2 l+c_3 k+c_4 l}$,
- $M(i, j, k, l) = 0$ *whenever* $i > k$ *or* $j > l$.
- $\beta c_1 + c_3 \geq 0$ *and* $\beta c_2 + c_4 \geq 0$.

**Theorem 2.** *[4, Theorem 5.1] Let $M$ be an $(a + 1)b \times (a + 1)b$ geometrically progressive matrix with parameters $(C, D, c_0, c_1, c_2, c_3, c_4, \beta)$, and let $B$ be a real number. Define*

$$S_B = \{(k, l) \in 0, \ldots a \times 1, \ldots b | M(k, l, k, l) \leq B\}$$

*and set $w = |S_B|$. If $L$ is the lattice defined by rows $(k, l) \in S_B$ of $M$, then*

$$det(L) \leq ((a + 1)b)^{w/2}(1 + C)^{w^2} M(k, l, k, l).$$

Blomer-May [1] referred to the coefficient vectors of the polynomials $g_{i,k}(xX, yY)$ as the $X$ block. The $X$ block is further divided into $(m+1)$ many blocks named as $X_l$ for $l = 0, \ldots, m$, where the block $X_l$ consists of the $l + 1$ many coefficient vectors of $g_{i,k}$ with $i + k = l$. Fixing $l$, each of these $l + 1$ vectors is denoted as $X_{l,k}$, $0 \leq k \leq l$ (the $k$-th vector in the $X_l$ block). That is, $X_{l,k}$ is the coefficient vector of $g_{l-k,k}$.

Further, a $Y_j$ block is defined as the block of all $m+1$ coefficient vectors of the polynomials that are shifted by $y^j$. The $k$-th vector in $Y_j$ block is called $Y_{j,k}$, which is the coefficient vector of $h_{j,k}$.

All column vectors with label $x^l y^j$, $l \geq j$ form a column block named $X^{(l)}$. Similarly the column block $Y^{(l)}$ contains all column vectors labeled with $x^i y^{i+l}$. Then a new lattice $L_M$ is presented in [1] as follows.

- Lattice parameters $m$ and $t$ are chosen to build a lattice basis $B(m,t)$.
- In the block $Y_t$ of $B(m,t)$, every vector is removed except for the last vector $Y_{t,m}$. In the block $Y_{(t-1)}$ of $B(m,t)$, every vector is removed except for the last two vectors $Y_{t-1,m}, Y_{t-1,m-1}$. This continues upto the block $Y_1$, where every vector is removed except the last $t$ vectors $Y_{1,m}, Y_{1,m-1}, \ldots, Y_{1,m-t+1}$.
- Every vector in the block $X$ is removed except for the vectors in the $t+1$ many blocks $X_{m-t}, \ldots, X_m$.
- The columns need to be deleted in such a manner that the resulting basis is again triangular. All column blocks $X^{(0)}, X^{(1)}, \ldots, X^{(m-t-1)}$ are removed. Moreover, in the column block $Y^{(l)}$, $1 \le l \le t$, the columns labeled with $x^i y^{i+1}$, $0 \le i \le m-t+l$, are removed.

Below, we give an example of the lattice basis for the parameter choice $m = 3, t = 1$. For this $m, t$, the basis vectors $B(3,1)$ of the lattice $L_B$ form a triangular matrix $M_B$ as follows. In case of [3], the product of the diagonal elements gives $det(M_B)$.

$$
\begin{array}{c|c|cc|ccc|cccc|cccc|}
 & \downarrow & \downarrow & \downarrow & & & & & & & & \downarrow & \downarrow & \downarrow & \\
 & 1 & x & xy & x^2 & x^2y & x^2y^2 & x^3 & x^3y & x^3y^2 & x^3y^3 & y & xy^2 & x^2y^3 & x^3y^4 \\
\hline
\rightarrow e^3 & e^3 & & & & & & & & & & & & & \\
\hline
\rightarrow xe^3 & & e^3X & & & & & & & & & & & & \\
\rightarrow fe^2 & - & - & e^2XY & & & & & & & & & & & \\
\hline
x^2e^3 & & & & e^3X^2 & & & & & & & & & & \\
xfe^2 & & - & & - & e^2X^2Y & & & & & & & & & \\
f^2e & - & - & - & - & - & eX^2Y^2 & & & & & & & & \\
\hline
x^3e^3 & & & & & & & e^3X^3 & & & & & & & \\
x^2fe^2 & & & & - & & & - & e^2x^3Y & & & & & & \\
xf^2e & & - & & - & - & & - & - & eX^3Y^2 & & & & & \\
f^3 & - & - & - & - & - & - & - & - & - & X^3Y^3 & & & & \\
\hline
\hline
\Rightarrow ye^3 & & & & & & & & & & & e^3Y & & & \\
\Rightarrow yfe^2 & & & - & & & & & & & & - & e^2XY^2 & & \\
\Rightarrow yf^2e & & & - & & - & - & & & & & - & - & eX^2Y^3 & \\
yf^3 & & & - & & & - & & - & - & - & - & - & - & X^3Y^4 \\
\hline
\end{array}
$$

In case of [4], the rows marked by ⇒ are removed. In that case, the sub-matrix of $M_B$ is not a square matrix and the determinant is calculated following [4, Theorem 5.1] (presented above in Theorem 2 also). The work of [1] removes the rows marked by ⇒ a well as →. This sub-matrix of $M_B$ is again not a square one, but the columns marked by ↓ are also removed to get a square matrix (see [1, Theorems 2, 3] for more details).

It has been demonstrated in [3] that for $\delta < 0.284$, one can find $m, t$ such that $N$ can be factored using the LLL algorithm. Further the idea was improved to extend this bound upto 0.292 by using non-triangular lattice bases [4]. Improvements towards implementation have been studied in [1] by significantly reducing the lattice dimension for same $m, t$. They could achieve the bound of $\delta$ till $N^{0.290}$ theoretically which is less than the Boneh-Durfee bound of $N^{0.292}$, but the results of [1] were more efficient in practice.

Our idea in this paper is to extend the bounds of [3, 4, 1] further with small lattice dimension given a few MSBs of $p$ (which can also be searched exhaustively). One may note that given the constraint $q < p < 2q$, a few bits of $p, q$ can be known in polynomial time (e.g., around 7 bits for 1024 bit $N$ and 9 bits for 2048 bit $N$ following the work of [16]). This will indeed reduce the search effort further.

It is time consuming to handle large lattice dimensions and that is the constraint in extending the value of $\delta$ using Boneh-Durfee [3] and related techniques [4, 1]. Also it is not very clear how large lattice dimensions can be handled efficiently in a parallel environment. In our case, it is very easy to distribute the work in different machines independently for different choices of the MSBs, given the small lattice dimension to work with. Advantage of our ideas in comparison with [3, 4, 1] is presented in Section 3.

## 2 Reduction in Lattice Dimension

We start working in the direction of [3, Section 4].

**Theorem 3.** *Let $N = pq$, where $p$ and $q$ are primes of same bitsize. Let $d = N^{\delta}$. Suppose, $p_0 \geq \sqrt{N}$ be an approximation of $p$ with $|p - p_0| < N^{\gamma}$, $\gamma \leq \frac{1}{2}$. We show that, RSA is insecure if $\delta < \frac{\gamma+3-2\sqrt{\gamma(\gamma+3)}}{3}$.*

*Proof.* We assume $e = N$ as for $e < N$ one can get better upper bound on $\delta$ (similar to the approach of [3, Page 9]).

Let $q_0 = \frac{N}{p_0}$. We have $ed = 1 + k\phi(N) = 1 + k(N + 1 - p - q) = 1 + k(N + 1 - p_0 - q_0 - (p + q - p_0 - q_0)) = 1 + x(A + y)$, where $x = k < d = N^{\delta} = e^{\delta}, A = N + 1 - p_0 - q_0, y = -(p + q - p_0 - q_0)$. As $p > \sqrt{N}$ and as we assume $p_0 \geq \sqrt{N}$ too, we have $|y| < N^{\gamma} = e^{\gamma}$.

We have to find $x_0, y_0$ such that $1 + x_0(A + y_0) \equiv 0 \bmod e$, where $|x_0| < e^{\delta}$ and $|y_0| < e^{\gamma}$. Let $X = e^{\delta}, Y = e^{\gamma}$. Note that we consider the same $X$ as in [3, Section 4], but our $Y$ is generalized as $Y$ has been taken as $e^{\frac{1}{2}}$ in [3, Section 4].

One may refer to [3, Section 4] for $det_x = e^{m(m+1)(m+2)/3} X^{m(m+1)(m+2)/3} Y^{m(m+1)(m+2)/6}$ and $det_y = e^{tm(m+1)/2} X^{tm(m+1)/2} Y^{t(m+1)(m+t+1)/2}$. Plugging in the values of $X$ and $Y$ (note that our $Y$ is different than [3, Section 4]), we obtain, $det_x = e^{m^3(\frac{1}{3}+\frac{\delta}{3}+\frac{\gamma}{6})+o(m^3)}$, $det_y = e^{tm^2(\frac{1}{2}+\frac{\delta}{2}+\frac{\gamma}{2})+t^2m\frac{\gamma}{2}+o(tm^2)}$. Now $det(L) = det_x det_y$ and we need to satisfy $det(L) < e^{mw}$, where $w = (m+1)(m+2)/2 + t(m+1)$, the dimension of $L$. To satisfy $det(L) < e^{mw}$, we need $m^3(\frac{1}{3}+\frac{\delta}{3}+\frac{\gamma}{6}) + tm^2(\frac{1}{2}+\frac{\delta}{2}+\frac{\gamma}{2}) + t^2m\frac{\gamma}{2} < (tm + \frac{m^2}{2})m$. This leads to $m^2(-\frac{1}{6}+\frac{\delta}{3}+\frac{\gamma}{6}) + tm(-\frac{1}{2}+\frac{\delta}{2}+\frac{\gamma}{2}) + t^2\frac{\gamma}{2} < 0$. After fixing an $m$, the left hand side is minimized at $t = \frac{\frac{1}{2}-\frac{\delta}{2}-\frac{\gamma}{2}}{\gamma}$. Putting this value we have, $m^2(-\frac{1}{6}+\frac{\delta}{3}+\frac{\gamma}{6}) + \frac{m^2(-\frac{1}{2}+\frac{\delta}{2}+\frac{\gamma}{2})(\frac{1}{2}-\frac{\delta}{2}-\frac{\gamma}{2})}{\gamma} - \frac{(\frac{1}{2}-\frac{\delta}{2}-\frac{\gamma}{2})^2m^2}{\gamma^2}\frac{\gamma}{2} < 0$, simplying $(-\frac{1}{6}+\frac{\delta}{3}+\frac{\gamma}{6}) + \frac{(-\frac{1}{2}+\frac{\delta}{2}+\frac{\gamma}{2})(\frac{1}{2}-\frac{\delta}{2}-\frac{\gamma}{2})}{\gamma} - \frac{(\frac{1}{2}-\frac{\delta}{2}-\frac{\gamma}{2})^2}{\gamma^2}\frac{\gamma}{2} < 0$. Hence, $\delta < \frac{2\gamma+6-\sqrt{(2\gamma+6)^2+12(\gamma^2+2\gamma-3)}}{6}$ and simplifying we get $\delta < \frac{\gamma+3-2\sqrt{\gamma(\gamma+3)}}{3}$.

Similar to the idea presented in [3, Section 4], if the first two elements (polynomials $P_1(x, y)$, $P_2(x, y)$) of the reduced basis out of the LLL algorithm are algebraically independent (i.e., nonzero resultant $res(P_1, P_2)$ which is a polynomial of $y$, say), then we will get $x_0, y_0$ correctly which will in turn provide the factorization of $N$ making RSA insecure. (This actually happens with a high probability in practice as we have also checked by experimentation.)

□

First note that the result presented in Theorem 3 gives the same upper bound 0.284 presented in [3] when $\gamma = \frac{1}{2}$. In the above theorem we use lattice of full rank. We can improve the bounds on $\delta$ if we use the techniques based on sub-lattice [4, 1]. These ideas are discussed in Section 2.1.

Based on Theorem 3, one can design

- a probabilistic polynomial time algorithm $\mathcal{A}$, which will take
- $N, e, p_0$ as inputs
- and will provide correct $p$ if
  - $|p - p_0| < N^{\gamma}$,
  - $\delta < \frac{2\gamma+6-\sqrt{(2\gamma+6)^2+12(\gamma^2+2\gamma-3)}}{6}$
  - and the resultant polynomial $res(P_1, P_2)$ on $y$ is nonzero with integer solution (in practice the integer solution is correct with a high probability).

It is important to study the performance of the algorithm $\mathcal{A}$, based on different values of $m, t$. Our main improvement is achieved due to the lesser bound on $Y$. Once again, we like to reiterate that we consider the same $X$ as in [3, Section 4], but our $Y = e^{\gamma}$ is smaller than the $Y$ considered as $e^{\frac{1}{2}}$ in [3, Section 4].

The knowledge of MSBs of $p_0$ can be interpreted as following also. One can use $\mathcal{A}$ for $p_0 = \sqrt{N}$ to $\sqrt{2N}$ (as for $q < p < 2q$, we have $\sqrt{N} < p < \sqrt{2N}$) at an interval of $N^{\gamma}$, i.e., one needs to try for $(\sqrt{2}-1)N^{\frac{1}{2}-\gamma}$ many steps. Each step will require $\pi(N)$ time complexity when $\pi(N)$ is the running time for $\mathcal{A}$. It is important to mention here that proper choice of $m, t$ are required to get the results and we will compare this with [3, Section 6, Page 9] and [1, Table 1, Section 6, Page 18].

For $\gamma = 0.477$, we find that $\delta < 0.30044$, and for $\gamma = 0.478$, we find that $\delta < 0.29975$. Thus, for $\delta = 0.3$, it is enough to consider $\gamma = 0.477$, which gives $\frac{1}{2} - \gamma = 0.023$. Hence, theoretically speaking, we either need to know $(\frac{1}{2}-\gamma) \times \log_2 N$ many bits of $p_0$ or $N^{\frac{1}{2}-\gamma}$ many invocations of $\mathcal{A}$ are required. Thus, $N$ can be factorized in $O(N^{0.023}\pi(N))$ time complexity with the knowledge of $e$ when $d$ is $O(N^{0.3})$. For 1000 bit integers, our strategy will require either the knowledge of 23 bits or $2^{23}$ invocations of $\mathcal{A}$. See Table 1 later in Section 2.1 for more detailed results. However, these are only theoretical estimates and we will present the exact experimental details in Section 3.

*Remark 1.* If we do not neglect the lower order terms in Theorem 3, then to satisfy $det(L) < e^{mw}$ we need

$$m(m+1)(\frac{m+2}{3}+\frac{t}{2})\delta+(m+1)(\frac{m(m+2)}{6}+\frac{t(m+t+1)}{2})\gamma < m(m+1)(\frac{m+2}{6}+\frac{t}{2}). \quad (1)$$

The value of $\gamma$ is always $\frac{1}{2}$ in the analysis of [3]. However, due to the knowledge of a few bits, we can have $\gamma < \frac{1}{2}$, and thus it is possible to get extended bound on $\delta$ in our case. This is the reason we get improved bounds on $d$ from Theorem 3 than the idea of [3] for same lattice dimension. Later in Section 2.1 where we exploit the ideas related to sub-lattice, the similar technique is used and hence following Theorems 4, 5, we get better bounds on $d$ than what presented in [4, 1] respectively.

### 2.1 Exploiting the Sub-lattice based Techniques

Boneh and Durfee showed how they improve their result $\delta < 0.284$ [3] to $\delta < 0.292$ [4] using the sub-lattice technique. We will now follow the idea of [4]. This idea has also been followed in [18, Section 6].

**Theorem 4.** *Let $N = pq$, where $p$ and $q$ are primes of same bitsize. Let $d = N^{\delta}$. Suppose, $p_0 \geq \sqrt{N}$ be an approximation of $p$ with $|p - p_0| < N^{\gamma}$, $\gamma \leq \frac{1}{2}$. We show that, RSA is insecure if $1 - 2\gamma < \delta < 1 - \sqrt{\gamma}$.*

*Proof.* This proof is similar to the proof of Theorem 3, till the calculation of $det_x$. However, $det_y$ will be different here than in the proof of Theorem 3.

Let $M_{By}$ be the portion of the matrix $M_B$ with rows corresponding to the $y$ shifts and columns corresponding to the variables of the form $x^u y^v$, for $v > u$. In this case, $M_{By}$ is a geometrically progressive matrix with parameter choice $(m^{2m}, e, m, \delta + \gamma, \gamma - 1, -1, 1, b)$ for some $b$. One may note that the first three conditions of Definition 1 hold. To satisfy the fourth condition, the parameter $b$ should satisfy $b(\delta + \gamma) - 1 \geq 0$ and $b(\gamma - 1) + 1 \geq 0$ together and thus we get the constraint $\delta > 1 - 2\gamma$, which in turn gives a possible value of $b$ as $b = \frac{2}{2-2\gamma}$. Similar to the idea of [4], we also get the optimal choice for $t$ as twice the value of $t$ in Theorem 3, i.e., $t = \frac{1-\delta-\gamma}{\gamma}m$. Following Definition 1, we have $M_{By}(k, l, k, l) = e^{m+(\delta+\gamma-1)k+\gamma l}$. Denote $S_B$ (as in Theorem 2) by $S$ when $B = e^m$. By our choice of $t$, we have $(k, l) \in S$ iff $l \leq \frac{1-\delta-\gamma}{\gamma}k$. Neglecting the lower order terms, $|S| = \frac{1-\delta-\gamma}{2\gamma}m^2$. Thus $w = \frac{(m+1)(m+2)}{2} + |S| = \frac{m^2}{2} + |S| = (\frac{1}{2} + \frac{1-\delta-\gamma}{2\gamma})m^2$ (neglecting lower order terms) $= \frac{1-\delta}{2\gamma}m^2$. Following the similar idea as in [4] and going through similar calculation in [18, Section 6] for the sub-lattice, we get $det_y = e^{\frac{1}{12}\frac{9-4(\delta+\frac{1}{2}+\gamma)^2}{2\gamma}m^3}$. Then the condition $det(L) = det_x det_y < e^{mw}$ gives the bound $\delta < 1 - \sqrt{\gamma}$. □

The result presented in Theorem 4 gives the same upper bound 0.292 presented in [4] when $\gamma = \frac{1}{2}$. Next we present an approach following [1, Section 4].

**Theorem 5.** *Let $N = pq$, where $p$ and $q$ are primes of same bitsize. Let $d = N^{\delta}$. Suppose, $p_0 \geq \sqrt{N}$ be an approximation of $p$ with $|p - p_0| < N^{\gamma}$, $\gamma \leq \frac{1}{2}$. We show that, RSA is insecure if $\delta < \frac{\sqrt{16\gamma^2 - 4\gamma + 4} - (6\gamma - 2)}{5}$.*

*Proof.* This proof is again similar to the proof of Theorem 3, but both $det_x$ and $det_y$ will be different here than in the proof of Theorem 3. Given that certain rows and columns of $M_B$ will be removed following the idea of [1], the diagonal elements of the new matrix will be
$X^m e^m, X^m Y e^{m-1}, \ldots, X^m Y^m,$
$X^{m-1} e^m, X^{m-1} Y e^{m-1}, \ldots, X^{m-1} Y^{m-1} e,$
$\ldots,$
$X^{m-t} e^m, X^{m-t} Y e^{m-1}, \ldots, X^{m-t} Y^{m-t} e^t$
for $x$-shifts (i.e., they will contribute to $det_x$) and
$X^m Y^{m+t},$

$X^mY^{m+t-1}, X^{m-1}Y^{m+t-2}e,$
$\ldots,$
$X^mY^{m+1}, X^{m-1}Y^me, \ldots X^{m-t+1}Y^{m-t+2}e^{t-1},$
for $y$-shifts (i.e., they will contribute to $det_y$).

Multiplying the diagonal elements and neglecting the lower order terms, we need the condition

$$X^{tm^2 - \frac{mt^2}{2} + \frac{t^3}{6}} Y^{\frac{tm^2}{2} + \frac{t^3}{6}} < e^{\frac{tm^2}{2}}.$$

Putting the values of $X = e^\delta, Y = e^\gamma, t = \tau m$, we have the required condition

$$(\frac{\delta}{6} + \frac{\gamma}{6})\tau^2 - \frac{1}{2}\delta\tau + (\delta + \frac{\gamma}{2} - \frac{1}{2}) < 0.$$

The left hand side is minimum when $\tau = \frac{\delta}{\frac{2}{3}(\delta+\gamma)}$. Putting this value of $\tau$, in the previous inequality we get the bound on $\delta$. □

The result presented in Theorem 5 gives the same upper bound 0.290 presented in [1] when $\gamma = \frac{1}{2}$.

In Table 1 we present the corresponding values of $\gamma$ for which the values of $\delta$ can be reached. The value of $\frac{1}{2} - \gamma$ gives the proportion of bits we need to know or search exhaustively. We start listing the results from $\delta = 0.285$, as already there is theoretical result available for $\delta = 0.284$ using full rank lattice [3] (the theoretical result achieving $\delta = 0.292$ is presented using sub-lattice in [4]).

| $\delta$ | $\gamma$ | $\gamma$ | $\gamma$ |
|---|---|---|---|
|  | Theorem 3 | Theorem 4 | Theorem 5 |
| 0.285 | 0.49962 | 0.5 | 0.5 |
| 0.290 | 0.49224 | 0.5 | 0.49985 |
| 0.295 | 0.48491 | 0.49703 | 0.49284 |
| 0.300 | 0.47763 | 0.48999 | 0.48589 |

**Table 1.** Theoretical estimate of $\gamma$ following Theorems 3, 4, 5 to reach the corresponding bounds on $\delta$.

It is clear from the Table 1, that from theoretical point of view, the best efficiency is achieved in Theorem 4, followed by Theorem 5 and Theorem 3.

## 3 Complete Experimental Details

We have implemented the program in SAGE 2.10.1 over Linux Ubuntu 7.04 on a computer with Dual CORE Intel(R) Pentium(R) D CPU 2.80GHz, 1 GB RAM and 2 MB Cache. While comparing our results to the existing results [3, 4, 1], we will present higher bounds on $d$ indeed.

In Section 3.1, we present the results related to the algorithms in [3, 4, 1] on our platform. Then we discuss the implementation results in Section 3.2 related to Theorem 3, where the full rank lattice is exploited. The results related to sub-lattice (i.e., experimental results related to Theorems 4, 5) are presented in Section 3.3.

### 3.1 Existing Experimental Results

First we restate the results of [3, Table in Section 6] in Table 2 with the running time on our experimental setup, which will provide a clear idea of our improvement while presenting our results.

| $N$ | $\delta$ | $m$ | $t$ | lattice dimension | running time [3] | running time on our machine |
|---|---|---|---|---|---|---|
| 1000 bits | 0.265 | 5 | 3 | 39 | 45 minutes | 23 seconds |
| 3000 bits | 0.265 | 5 | 3 | 39 | 5 hours | 128 seconds |
| 10000 bits | 0.255 | 3 | 1 | 14 | 2 hours | 7 seconds |

**Table 2.** Running time estimate of [3] on our setup.

In Table 3 we present the results of [4, 1] to give an idea of lattice dimensions required for certain $\delta$ values. The time estimates are presented as in the corresponding papers [4, 1].

| $N$ | $\delta$ | $m$ | $t$ | lattice dimension | running time | Reference |
|---|---|---|---|---|---|---|
| 1000 bits | 0.270 | 6 | 2 | 21 | 19 minutes | [1] |
| 1000 bits | 0.274 | 8 | 3 | 36 | 300 minutes | [1] |
| 1000 bits | 0.2765 | 10 | 4 | 55 | 26 hours | [1] |
| 1000 bits | 0.278 | 11 | 5 | 72 | 6 days | [1] |
| 1000 bits | 0.280 | 7 | 3 | 45 | 14 hours | [4] |
| 2000 bits | 0.265 | 4 | 2 | 15 | 6 minutes | [1] |
| 2000 bits | 0.275 | 7 | 3 | 45 | 65 hours | [4] |
| 4000 bits | 0.265 | 5 | 2 | 25 | 14 hours | [4] |
| 6000 bits | 0.265 | 4 | 2 | 15 | 100 minutes | [1] |
| 6000 bits | 0.269 | 5 | 2 | 18 | 8 hours | [1] |
| 10000 bits | 0.255 | 3 | 1 | 11 | 90 minutes | [4] |

**Table 3.** Results from [4, 1].

We have already pointed out that for practical experiments, the resultant calculation takes more time than lattice reduction. It is not clear from the experimental results in [3, 4] whether the resultant calculation time has been considered. In [1, Section 6], it has been clearly commented that only lattice reduction time has been presented.

Let us denote the lattice reduction time by $T_i^l$, the resultant calculation time by $T_i^r$ and the solution time by $T_i^s$ (in seconds).

We have implemented the algorithms of [3, 4, 1] to study all the time requirements in detail. Below we present the running time for 1000, 2000 and 4000 bits $N$ and for $\delta = 0.26$, where $m = 7, t = 3$. It is clear from the experimental results in Table 4 that the resultant calculation takes more time than lattice reduction. The experiments are for a single run in each case.

| $N$ | $w$ | Reference | $T_l$ | $T_r$ | $T_s$ |
|---|---|---|---|---|---|
| 1000 bits | 60 | [3] | 76.29 | 411.42 | 3.90 |
| 1000 bits | 45 | [4] | 61.17 | 410.72 | 4.20 |
| 1000 bits | 32 | [1] | 29.58 | 333.76 | 2.46 |
| 2000 bits | 60 | [3] | 347.56 | 1139.58 | 11.03 |
| 2000 bits | 45 | [4] | 283.69 | 1147.34 | 13.17 |
| 2000 bits | 32 | [1] | 148.71 | 938.79 | 8.02 |
| 4000 bits | 60 | [3] | 1358.30 | 2461.80 | 37.89 |
| 4000 bits | 45 | [4] | 1041.47 | 2465.66 | 58.23 |
| 4000 bits | 32 | [1] | 605.03 | 2028.11 | 32.96 |

**Table 4.** Experimental Results for execution of the algorithms presented in $[3, 4, 1]$ on 1000, 2000 and 4000 bits $N$ give our platform, when $m = 7, t = 3$, $\delta = 0.26$.

Note that, given $m, t$, the lattice dimension $w$ can be calculated directly as $w = (m + 1)(m + 2)/2 + t(m + 1)$ for [3] (Theorem 3 in our approach) and $w = (m + 1)(t + 1)$ for [1] (Theorem 5 in our approach). However, the there is no exact formula for calculating $w$ from $m, t$ for the case of [4] (Theorem 4 in our approach). Thus, in this section, the value of $w$ is presented as found experimentally following Theorem 4 in our approach and we present the maximum value of $w$, when more than one runs are executed.

### 3.2 Our Experimental Results: Using Complete Lattice

In this section we concentrate on the experimental results following our Theorem 3.

In Table 5, we present our results for 1000-bit $N$ that shows that the bound on $\delta$ can be increased much further than the work of $[3, 4, 1]$ as listed in Tables 2, 3. While considering the primes, we take $q < p < 2q$, i.e., $p, q$ are of same bit size. Further we select $p, q$ randomly with the constraint that $p - q > n^{0.45}$ and $2q - p > n^{0.45}$ so that the ideas in the direction of $[18, 12]$ do not work efficiently. The cryptanalytic strategy of [18] works well when $p - q$ is bounded and in the same direction, the method of [12] works well when $2q - p$ is bounded.

By $\tau$-bit $N$ we mean that $p, q$ are of $\frac{\tau}{2}$ bits each. The column with $\delta$ provides that we consider the first $d$ which is greater than or equal to $\lceil N^{\delta} \rceil$ such that $d$ is coprime to $\phi(N)$. After fixing $\delta$, the size of $N$ (which is 1000 bits for the Table 5) and the lattice parameters $(m, t, w)$, we go for 100 runs for each case (except for $m = 7, t = 3$, when we go for 10 runs). For each run, we calculate the minimum number of bits (for $i$-th run, call this variable $v_i$) that need to be known to factorize $N$ from the knowledge of $N, e$ using the idea of Theorem 3. We present the minimum $\min(v_i)$, maximum $\max(v_i)$, average $\overline{v}$ and standard deviation $\sigma(v)$ of the data set $v_i$, for $i = 1, \ldots, 100$. Further, consider that the time required for each run is $T_i$ second(s), while considering $v_i$ many bits are known.

As the solution time is almost negligible compared to the other two, for the results in this subsection we consider $T_i = T_i^l + T_i^s$. The average of $T_i$ over 100 data is also presented as $\overline{T}$, which is in second(s).

| $m = 3, t = 1, w = 14$ | | | | | |
|---|---|---|---|---|---|
| $\delta$ | $\min(v_i)$ | $\max(v_i)$ | $\overline{v}$ | $\sigma(v)$ | $\overline{T}$ |
| 0.280 | 30 | 33 | 31.3 | 0.900 | 0.252 |
| 0.285 | 37 | 41 | 39.0 | 1.096 | 0.327 |
| 0.290 | 45 | 48 | 46.1 | 0.700 | 0.325 |
| 0.295 | 52 | 55 | 53.6 | 0.800 | 0.332 |
| 0.300 | 60 | 62 | 61.0 | 0.775 | 0.328 |

| $m = 3, t = 2, w = 18$ | | | | | |
|---|---|---|---|---|---|
| $\delta$ | $\min(v_i)$ | $\max(v_i)$ | $\overline{v}$ | $\sigma(v)$ | $\overline{T}$ |
| 0.280 | 29 | 34 | 31.41 | 1.040 | 0.286 |
| 0.285 | 38 | 41 | 39.04 | 0.811 | 0.362 |
| 0.290 | 44 | 50 | 46.36 | 0.986 | 0.365 |
| 0.295 | 51 | 57 | 53.81 | 1.102 | 0.366 |
| 0.300 | 59 | 64 | 61.13 | 1.064 | 0.368 |

| $m = 5, t = 1, w = 27$ | | | | | |
|---|---|---|---|---|---|
| $\delta$ | $\min(v_i)$ | $\max(v_i)$ | $\overline{v}$ | $\sigma(v)$ | $\overline{T}$ |
| 0.280 | 18 | 23 | 20.96 | 1.076 | 20.380 |
| 0.285 | 27 | 32 | 29.0 | 0.970 | 20.422 |
| 0.290 | 34 | 39 | 37.17 | 1.105 | 20.493 |
| 0.295 | 44 | 48 | 45.35 | 0.910 | 20.565 |
| 0.300 | 51 | 57 | 53.45 | 0.994 | 21.534 |

| $m = 5, t = 2, w = 33$ | | | | | |
|---|---|---|---|---|---|
| $\delta$ | $\min(v_i)$ | $\max(v_i)$ | $\overline{v}$ | $\sigma(v)$ | $\overline{T}$ |
| 0.280 | 15 | 19 | 16.75 | 1.014 | 23.613 |
| 0.285 | 22 | 27 | 24.37 | 0.976 | 24.040 |
| 0.290 | 29 | 34 | 31.56 | 0.962 | 24.876 |
| 0.295 | 36 | 41 | 38.72 | 1.021 | 25.114 |
| 0.300 | 44 | 48 | 45.99 | 0.995 | 25.267 |

| $m = 5, t = 3, w = 39$ | | | | | |
|---|---|---|---|---|---|
| $\delta$ | $\min(v_i)$ | $\max(v_i)$ | $\overline{v}$ | $\sigma(v)$ | $\overline{T}$ |
| 0.280 | 15 | 19 | 16.61 | 0.915 | 24.158 |
| 0.285 | 22 | 27 | 24.17 | 0.960 | 24.606 |
| 0.290 | 30 | 34 | 31.47 | 1.044 | 25.484 |
| 0.295 | 37 | 41 | 38.72 | 0.991 | 25.677 |
| 0.300 | 44 | 48 | 45.82 | 0.993 | 25.851 |

| $m = 7, t = 3, w = 60$ | | | | | |
|---|---|---|---|---|---|
| $\delta$ | $\min(v_i)$ | $\max(v_i)$ | $\overline{v}$ | $\sigma(v)$ | $\overline{T}$ |
| 0.280 | 8 | 10 | 8.7 | 0.640 | 481.016 |
| 0.285 | 15 | 17 | 16.5 | 1.118 | 483.016 |
| 0.290 | 23 | 25 | 23.9 | 0.831 | 487.354 |
| 0.295 | 30 | 33 | 31.5 | 1.204 | 499.738 |
| 0.300 | 37 | 40 | 38.59 | 0.917 | 520.45 |

**Table 5.** Our results for 1000 bits $N$ and different $m, t, w$.

The data in Table 5 clearly presents the improvements over the works of [3, 4, 1] for achieving higher bounds on $\delta$ for 1000 bits $N$.

We present detailed results for 1000 bits $N$ in Table 5 and for 10000 bits $N$ in Table 7. A few results are provided for 2000, 4000 and 6000 bits $N$ in Table 6. We consider the $\delta$ values which are higher than the results achieved in [4, 1] (see also Table 3). To have a comparison, we take same values of $m, t$ as used in the highest values for $\delta$ in [3, 4, 1].

| $N$ | $\delta$ | $m$ | $t$ | $w$ | $\min(v_i)$ | $\max(v_i)$ | $\overline{v}$ | $\overline{T}$ |
|---|---|---|---|---|---|---|---|---|
| 2000 bits | 0.280 | 7 | 3 | 60 | 15 | 17 | 16 | 1341.61 |
| 4000 bits | 0.270 | 5 | 2 | 33 | 0 | 0 | 0 | 183.49 |
| 6000 bits | 0.270 | 5 | 2 | 33 | 0 | 0 | 0 | 305.21 |

**Table 6.** Our Results for 2000, 4000 and 6000 bits $N$.

For 2000 bits $N$, we need to search 16 bits on an average and out of that 9 bits can be known using the technique of [16, Section 3.3]. Thus, the total work can be completed in a day with two computers with our specifications. We could reach the bound $\delta = 0.270$ (this is a better bound than what described in [3, 4, 1]) for 4000 and 6000 bits $N$, without the knowledge of any bit in $p_0$.

Next we present our experimental results for 10000 bits $N$ in Table 7.

| $\delta$ | $\min(v_i)$ | $\max(v_i)$ | $\overline{v}$ | $\sigma(v)$ | $\overline{T}$ | $\delta$ | $\min(v_i)$ | $\max(v_i)$ | $\overline{v}$ | $\sigma(v)$ | $\overline{T}$ |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0.260 | 0 | 0 | 0 | 0 | 7.38 | 0.275 | 212 | 215 | 213.5 | 0.81 | 7.82 |
| 0.261 | 0 | 0 | 0 | 0 | 7.21 | 0.280 | 293 | 295 | 293.8 | 0.75 | 7.95 |
| 0.262 | 2 | 4 | 3 | 0.63 | 7.12 | 0.285 | 369 | 370 | 369.4 | 0.488 | 10.61 |
| 0.263 | 18 | 21 | 19.6 | 1.11 | 7.12 | 0.290 | 442 | 443 | 442.7 | 0.44 | 10.62 |
| 0.265 | 51 | 53 | 51.79 | 0.60 | 7.19 | 0.295 | 516 | 519 | 517.1 | 0.96 | 10.74 |
| 0.270 | 131 | 133 | 132.1 | 1.22 | 7.54 | 0.300 | 589 | 591 | 589.9 | 0.68 | 10.79 |

**Table 7.** Our results for 10000 bits $N$ and different $m = 3, t = 1, w = 14$.

Refer to the result of [4] in this direction (see also Table 3) where in such a case the bound of $\delta = 0.255$ could be achieved. However, we could reach the bound $\delta = 0.261$ without the knowledge of any bit in $p_0$. Moreover, from the results in Table 7, it is clear that the bound of $\delta = 0.263$ can be achieved easily in practice by searching around 20 MSBs of $p$. Considering that at least 10 bits will be available using the technique of [16], the other 10 bits can be exhaustively searched in less than two hours on a computer with our specification.

### 3.3 Our Experimental Results: Using Sub-Lattice

In this section, we compare the experimental results related to Theorems 3, 4, 5. For comparison, we go for 10 runs for each case as presented in Table 9. One may easily note from Table 9 that the lattice reduction time reduces when sub-lattice is used instead of full rank lattice.

Below we present the results using the lattice parameters $m = 11, t = 5$. Using the technique of Theorem 3, we get the lattice dimension $w = 138$ and exploiting the strategy using sub-lattice following Theorem 4, we get $w = 108$ at maximum. Among the techniques we discuss in this paper, the minimum sub-lattice dimension is $w = 72$, using the idea of Theorem 5. We present the implementation results for this in Table 8. Due to longer time requirement, we present the result of one run in each case.

| $m = 11, t = 5, w = 72$ (Theorem 5) | | | | |
|---|---|---|---|---|
| $\delta$ | $v_i$ | $T_l$ | $T_r$ | $T_s$ |
| 0.280 | 3 | 1185.81 | 16741.77 | 46.92 |
| 0.285 | 10 | 1425.12 | 16954.72 | 39.11 |
| 0.290 | 18 | 1630.63 | 17107.79 | 40.23 |
| 0.295 | 25 | 1687.21 | 17135.02 | 59.61 |
| 0.300 | 33 | 1770.66 | 17222.95 | 59.38 |

**Table 8.** Experimental results following Theorem 5.

*Remark 2.* We like to estimate the best possible scenario in these attacks.

- The bound of $\delta = 0.285$ has never been achieved so far for 1000 bit $N$. Using $m = 7, t = 3, w = 32$, we find from Table 9 that around 17 bits need to be known to cryptanalyze RSA. In [16, Section 3.3], experimental results are known that a few bits of $p, q$ can be extracted in polynomial time (around 7 bits for 1024 bit $N$). Thus, using the 7-bits available from the technique of [16], only 9-bits need to be known. Given each run requires around 354 seconds, we need only a day with 2 machines.
- Now we present the estimate for $m = 11, t = 5$ following Table 8. Here the lattice dimension is 72 only. Our estimate is less than 6 hours to reduce the lattice in the machine we have referred (i.e. around 4 lattice reductions in a day). The requirements of bits will be around 33 for $\delta = 0.3$. Given that 7 bits will be available from the idea of [16], we need $2^{16}$ many CPUs to complete the task in 256 days (less than a year).

## 4 Conclusion

In this paper we show that the techniques of $[3, 4, 1]$ can be modified to have higher bounds on $d$ with low lattice dimensions. First of all, our idea provides theoretical extension of the bound of $N^{0.292}$ given the knowledge of some MSBs of $p$, which can also be managed by exhaustive search. We use the same lattice dimensions as presented in $[3, 4, 1]$ to have larger values of $d$ for which RSA can be attacked given $N, e$. Our experimental results outperform the results of $[3, 4, 1]$ for 1000, 2000, 4000, 6000 and 10000 bits $N$. We justify that for 1000 bits $N$, RSA can be crypanalyzed in practice when $d$ is of the order of $N^{0.3}$.

## References

1. J. Blömer and A. May. Low secret exponent RSA revised. CaLC 2001, LNCS 2146, pp. 4–19, 2001.
2. J. Blö mer and A. May. A generalized Wiener attack on RSA. PKC 2004, LNCS 2947, pp. 1–13, 2004.
3. D. Boneh and G. Durfee. Cryptanalysis of RSA with private key $d$ less than $N^{0.292}$. Eurocrypt 1999, LNCS 1592, pp. 1–11, 1999.
4. D. Boneh and G. Durfee. Cryptanalysis of RSA with private key $d$ less than $N^{0.292}$. IEEE Trans. on Information Theory, 46(4):1339–1349, 2000.
5. H. Cohen. A Course in Computational Algebraic Number Theory. Springer Verlag, 1996
6. D. Coppersmith. Small solutions to polynomial equations and low exponent vulnerabilities. Journal of Cryptology, 10(4):223–260, 1997.
7. A. Duejella. Continued fractions and RSA with small secret exponent. Tatra Mt. Math. Publ, vol. 29, pp. 101–112, 2004.
8. M. Ernst, E. Jochemsz, A. May and B. de Weger. Partial key exposure attacks on RSA up to full size exponents. Eurocrypt 2005, LNCS 3494, pp. 371–386, 2005.
9. N. Howgrave-Graham. Finding small roots of univariate modular equations revisited. Proceedings of Cryptography and Coding, LNCS 1355, pp. 131-142, 1997.
10. C. Jutla. On finding small solutions of modular multivariate polynomial equations. Proc. of Eurocrypt 1998.
11. A.Lenstra, H.Lenstra, and L.Lovasz. Factoring polynomials with rational coefficients. Mathematische Annalen, vol. 261, pp. 515-534, 1982.
12. S. Maitra and S. Sarkar. Revisiting Wiener's Attack – New Weak Keys in RSA. Information Security Conference (ISC 2008), to be held in Taipei, Taiwan, September 15–18, 2008.

13. R. L. Rivest, A. Shamir and L. Adleman. A method for obtaining digital signatures and public key cryptosystems. Communications of ACM, 21(2):158–164, Feb. 1978.
14. D. R. Stinson. Cryptography – Theory and Practice. 2nd Edition, Chapman & Hall/CRC, 2002.
15. R. Steinfeld, S. Contini, J. Pieprzyk and H. Wang. Converse results to the Wiener attack on RSA. PKC 2005, LNCS 3386, pp. 184–198, 2005.
16. H. -M. Sun, M. -E. Wu and Y. -H. Chen. Estimating the prime-factors of an RSA modulus and an extension of the Wiener attack. ACNS 2007, LNCS 4521, pp. 116–128, 2007.
17. M. Wiener. Cryptanalysis of short RSA secret exponents. IEEE Transactions on Information Theory, 36(3):553–558, 1990.
18. B. de Weger. Cryptanalysis of RSA with small prime difference. Applicable Algebra in Engineering, Communication and Computing, 13(1):17–28, 2002.

$m = 3, t = 1, w = 14$ (Theorem 3)

| $\delta$ | $\min(v_i)$ | $\max(v_i)$ | $\overline{v}$ | $\sigma(v)$ | $\overline{T}_l$ | $\overline{T}_r$ | $\overline{T}_s$ |
|---|---|---|---|---|---|---|---|
| 0.280 | 30 | 32 | 31.2 | 0.600 | 0.072 | 0.167 | 0.016 |
| 0.285 | 38 | 40 | 39.2 | 0.600 | 0.077 | 0.242 | 0.038 |
| 0.290 | 46 | 48 | 47.0 | 0.894 | 0.077 | 0.241 | 0.039 |
| 0.295 | 53 | 56 | 53.9 | 1.044 | 0.080 | 0.240 | 0.038 |
| 0.300 | 59 | 63 | 61.3 | 1.110 | 0.081 | 0.242 | 0.038 |

$m = 3, t = 1, w = 12$ (Theorem 4)

| $\delta$ | $\min(v_i)$ | $\max(v_i)$ | $\overline{v}$ | $\sigma(v)$ | $\overline{T}_l$ | $\overline{T}_r$ | $\overline{T}_s$ |
|---|---|---|---|---|---|---|---|
| 0.280 | 31 | 34 | 32.1 | 1.136 | 0.056 | 0.164 | 0.016 |
| 0.285 | 38 | 40 | 39.1 | 0.831 | 0.057 | 0.242 | 0.038 |
| 0.290 | 45 | 47 | 46.3 | 0.640 | 0.056 | 0.231 | 0.037 |
| 0.295 | 53 | 55 | 54.1 | 0.700 | 0.059 | 0.244 | 0.041 |
| 0.300 | 60 | 62 | 61.4 | 0.800 | 0.061 | 0.242 | 0.038 |

$m = 3, t = 1, w = 8$ (Theorem 5)

| $\delta$ | $\min(v_i)$ | $\max(v_i)$ | $\overline{v}$ | $\sigma(v)$ | $\overline{T}_l$ | $\overline{T}_r$ | $\overline{T}_s$ |
|---|---|---|---|---|---|---|---|
| 0.280 | 27 | 31 | 29.9 | 0.860 | 0.037 | 0.165 | 0.016 |
| 0.285 | 36 | 40 | 38.0 | 0.843 | 0.038 | 0.166 | 0.016 |
| 0.290 | 43 | 48 | 45.8 | 0.941 | 0.039 | 0.166 | 0.016 |
| 0.295 | 52 | 56 | 54.1 | 0.840 | 0.040 | 0.166 | 0.015 |
| 0.300 | 59 | 64 | 62.1 | 0.990 | 0.041 | 0.166 | 0.015 |

$m = 5, t = 2, w = 33$ (Theorem 3)

| $\delta$ | $\min(v_i)$ | $\max(v_i)$ | $\overline{v}$ | $\sigma(v)$ | $\overline{T}_l$ | $\overline{T}_r$ | $\overline{T}_s$ |
|---|---|---|---|---|---|---|---|
| 0.280 | 15 | 17 | 16.6 | 0.48 | 2.17 | 21.5 | 0.57 |
| 0.285 | 23 | 25 | 24.2 | 0.74 | 2.39 | 21.59 | 0.57 |
| 0.290 | 32 | 33 | 32.2 | 0.39 | 2.51 | 22.32 | 0.61 |
| 0.295 | 37 | 39 | 38.2 | 0.74 | 2.68 | 22.33 | 0.56 |
| 0.300 | 45 | 47 | 46.6 | 1.01 | 2.84 | 22.45 | 0.67 |

$m = 5, t = 2, w = 27$ (Theorem 4)

| $\delta$ | $\min(v_i)$ | $\max(v_i)$ | $\overline{v}$ | $\sigma(v)$ | $\overline{T}_l$ | $\overline{T}_r$ | $\overline{T}_s$ |
|---|---|---|---|---|---|---|---|
| 0.280 | 16 | 17 | 16.4 | 0.48 | 1.42 | 21.34 | 0.65 |
| 0.285 | 23 | 25 | 24.4 | 1.02 | 1.45 | 21.67 | 0.54 |
| 0.290 | 30 | 32 | 31.6 | 0.79 | 1.84 | 22.45 | 0.51 |
| 0.295 | 39 | 40 | 39.2 | 0.39 | 2.02 | 22.48 | 0.66 |
| 0.300 | 45 | 48 | 46.6 | 1.01 | 2.14 | 22.42 | 0.50 |

$m = 5, t = 2, w = 18$ (Theorem 5)

| $\delta$ | $\min(v_i)$ | $\max(v_i)$ | $\overline{v}$ | $\sigma(v)$ | $\overline{T}_l$ | $\overline{T}_r$ | $\overline{T}_s$ |
|---|---|---|---|---|---|---|---|
| 0.280 | 16 | 18 | 17 | 0.89 | 1.22 | 16.97 | 0.33 |
| 0.285 | 24 | 26 | 25.4 | 0.80 | 1.28 | 16.94 | 0.28 |
| 0.290 | 31 | 33 | 32.2 | 0.74 | 1.31 | 16.76 | 0.32 |
| 0.295 | 39 | 41 | 40 | 0.63 | 1.74 | 16.81 | 0.29 |
| 0.300 | 47 | 49 | 48 | 0.48 | 1.48 | 16.86 | 0.28 |

$m = 5, t = 3, w = 39$ (Theorem 3)

| $\delta$ | $\min(v_i)$ | $\max(v_i)$ | $\overline{v}$ | $\sigma(v)$ | $\overline{T}_l$ | $\overline{T}_r$ | $\overline{T}_s$ |
|---|---|---|---|---|---|---|---|
| 0.280 | 16 | 18 | 16.9 | 0.83 | 2.59 | 21.54 | 0.58 |
| 0.285 | 23 | 26 | 24.8 | 0.87 | 2.93 | 22.08 | 0.56 |
| 0.290 | 31 | 33 | 31.9 | 0.7 | 3.06 | 22.36 | 0.58 |
| 0.295 | 38 | 40 | 38.8 | 0.6 | 3.28 | 22.46 | 0.56 |
| 0.300 | 45 | 47 | 45.6 | 0.66 | 3.41 | 22.42 | 0.59 |

$m = 5, t = 3, w = 27$ (Theorem 4)

| $\delta$ | $\min(v_i)$ | $\max(v_i)$ | $\overline{v}$ | $\sigma(v)$ | $\overline{T}_l$ | $\overline{T}_r$ | $\overline{T}_s$ |
|---|---|---|---|---|---|---|---|
| 0.280 | 16 | 18 | 16.7 | 0.64 | 1.42 | 21.59 | 0.59 |
| 0.285 | 23 | 26 | 24.7 | 0.9 | 1.76 | 21.95 | 0.60 |
| 0.290 | 30 | 33 | 31.5 | 0.81 | 1.89 | 22.26 | 0.62 |
| 0.295 | 38 | 40 | 38.6 | 0.66 | 2.00 | 22.44 | 0.60 |
| 0.300 | 45 | 48 | 46.5 | 0.67 | 2.11 | 22.45 | 0.57 |

$m = 5, t = 3, w = 24$ (Theorem 5)

| $\delta$ | $\min(v_i)$ | $\max(v_i)$ | $\overline{v}$ | $\sigma(v)$ | $\overline{T}_l$ | $\overline{T}_r$ | $\overline{T}_s$ |
|---|---|---|---|---|---|---|---|
| 0.280 | 15 | 17 | 15.9 | 0.7 | 1.56 | 21.50 | 0.59 |
| 0.285 | 23 | 24 | 23.4 | 0.49 | 1.66 | 21.62 | 0.57 |
| 0.290 | 30 | 32 | 30.7 | 0.64 | 1.81 | 22.40 | 0.62 |
| 0.295 | 36 | 38 | 37.8 | 0.60 | 1.93 | 22.36 | 0.59 |
| 0.300 | 44 | 46 | 44.8 | 0.75 | 2.03 | 22.40 | 0.57 |

$m = 7, t = 3, w = 60$ (Theorem 3)

| $\delta$ | $\min(v_i)$ | $\max(v_i)$ | $\overline{v}$ | $\sigma(v)$ | $\overline{T}_l$ | $\overline{T}_r$ | $\overline{T}_s$ |
|---|---|---|---|---|---|---|---|
| 0.280 | 8 | 10 | 8.9 | 0.70 | 34.33 | 449.42 | 5.74 |
| 0.285 | 16 | 19 | 17.0 | 0.89 | 36.99 | 449.31 | 5.66 |
| 0.290 | 23 | 26 | 24.1 | 0.83 | 39.02 | 450.50 | 5.96 |
| 0.295 | 31 | 33 | 31.4 | 0.66 | 46.070 | 456.99 | 5.87 |
| 0.300 | 37 | 41 | 38.7 | 1.19 | 47.91 | 475.73 | 5.81 |

$m = 7, t = 3, w = 48$ (Theorem 4)

| $\delta$ | $\min(v_i)$ | $\max(v_i)$ | $\overline{v}$ | $\sigma(v)$ | $\overline{T}_l$ | $\overline{T}_r$ | $\overline{T}_s$ |
|---|---|---|---|---|---|---|---|
| 0.280 | 8 | 10 | 9 | 0.89 | 26.22 | 450.46 | 6.12 |
| 0.285 | 15 | 18 | 16.4 | 0.80 | 28.48 | 449.73 | 5.59 |
| 0.290 | 23 | 24 | 23.8 | 0.40 | 32.23 | 449.79 | 5.55 |
| 0.295 | 30 | 33 | 31.8 | 0.97 | 36.93 | 465.58 | 5.96 |
| 0.300 | 38 | 40 | 38.6 | 0.66 | 42.44 | 476.71 | 6.09 |

$m = 7, t = 3, w = 32$ (Theorem 5)

| $\delta$ | $\min(v_i)$ | $\max(v_i)$ | $\overline{v}$ | $\sigma(v)$ | $\overline{T}_l$ | $\overline{T}_r$ | $\overline{T}_s$ |
|---|---|---|---|---|---|---|---|
| 0.280 | 9 | 11 | 10.2 | 0.60 | 14.9 | 334.5 | 2.37 |
| 0.285 | 16 | 19 | 17.5 | 0.90 | 16.0 | 335.38 | 2.41 |
| 0.290 | 25 | 26 | 25.4 | 0.49 | 16.68 | 333.98 | 2.44 |
| 0.295 | 32 | 34 | 32.9 | 0.54 | 16.93 | 33.98 | 2.41 |
| 0.300 | 39 | 41 | 40.3 | 0.64 | 18.10 | 334.23 | 2.33 |

**Table 9.** Comparison of Experimental results following Theorems 3, 4, 5