# On Deterministic Polynomial-Time Equivalence of Computing the CRT-RSA Secret Keys and Factoring* 

Subhamoy Maitra and Santanu Sarkar<br>Applied Statistics Unit, Indian Statistical Institute, 203 B T Road, Kolkata 700 108, India. \{subho, santanu_r\}@isical.ac.in


#### Abstract

Let $N=p q$ be the product of two large primes. Consider CRT-RSA with the public encryption exponent $e$ and private decryption exponents $d_{p}, d_{q}$. It is well known that given any one of $d_{p}$ or $d_{q}$ (or both) one can factorize $N$ in probabilistic poly $(\log N)$ time with success probability almost equal to 1 . Though this serves all the practical purposes, from theoretical point of view, this is not a deterministic polynomial time algorithm. In this paper, we present a lattice based deterministic poly $(\log N)$ time algorithm that uses both $d_{p}, d_{q}$ (in addition to the public information $e, N$ ) to factorize $N$ for certain ranges of $d_{p}, d_{q}$. We like to stress that proving the equivalence for all the values of $d_{p}, d_{q}$ may be a nontrivial task.


Keywords: CRT-RSA, Cryptanalysis, Factorization, LLL Algorithm, RSA.

## 1 Introduction

RSA [17] is one of the most popular cryptosystems in the history of cryptology. Let us first briefly describe the idea of RSA as follows:

- primes $p, q$, with $q<p<2 q$;
$-N=p q, \phi(N)=(p-1)(q-1)$;
$-e, d$ are such that $e d=1+k \phi(N), k \geq 1$;
- $N, e$ are publicly available and plaintext $M$ is encrypted as $C \equiv M^{e} \bmod N$;
- the secret key $d$ is required to decrypt the ciphertext as $M \equiv C^{d} \bmod N$.

The study of RSA is one of the most attractive areas in cryptology research as evident from many excellent works (one may refer to $[1,10,15]$ and the references therein for detailed information). The paper [17] itself presents a probabilistic polynomial time algorithm that on input $N, e, d$ provides the factorization of $N$; this is based on the technique provided by [16]. One may also have a look at [18, Page 197]. Recently in [14, 7] it has been proved that given $N, e, d$, one can factor $N$ in deterministic poly $(\log N)$ time provided $e d \leq N^{2}$.

Speeding up RSA encryption and decryption is of serious interest and for large $N$ as both $e, d$ cannot be small at the same time. For fast encryption, it is possible to use smaller e, e.g.,

[^0]the value as small as $2^{16}+1$ is widely believed to be a good candidate. For fast decryption, the value of $d$ needs to be small. However, Wiener [19] showed that for $d<\frac{1}{3} N^{\frac{1}{4}}, N$ can be factorized easily. Later, Boneh-Durfee [2] increased this bound up to $d<N^{0.292}$. Thus use of smaller $d$ is in general not recommended. In this direction, an alternative approach has been proposed by Wiener [19] exploiting the Chinese Remainder Theorem (CRT) for faster decryption. The idea is as follows:

- the public exponent $e$ and the private CRT exponents $d_{p}$ and $d_{q}$ are used satisfying $e d_{p} \equiv 1 \bmod (p-1)$ and $e d_{q} \equiv 1 \bmod (q-1)$;
- the encryption is same as standard RSA;
- to decrypt a ciphertext $C$ one needs to compute $M_{1} \equiv C^{d_{p}} \bmod p$ and $M_{2} \equiv C^{d_{q}} \bmod q$;
- using CRT, one can get the plaintext $M \in \mathbb{Z}_{N}$ such that $M \equiv M_{1} \bmod p$ and $M \equiv$ $M_{2} \bmod q$.
This variant of RSA is popularly known as CRT-RSA. One may refer to [12] and the references therein for state of the art analysis on CRT-RSA.

Let us now outline the organization of this paper. Some preliminaries required in this area are discussed in Sections 1.1 and 1.2. A lattice based technique is used in Section 2 to show that one can factorize $N$ in deterministic polynomial time from the knowledge of $N, e, d_{p}, d_{q}$ for certain ranges of $d_{p}, d_{q}$. Section 3 concludes the paper.

### 1.1 Discussion on the known Probabilistic Polynomial time algorithm

Given $N, e$ and any one of $d_{p}, d_{q}$ (or both), there exists a well known solution to factorize $N$ in probabilistic poly $(\log N)$ time with probability almost 1 . An important work in this direction shows that with the availability of decryption oracle under a fault model, one can factorize $N$ in poly $(\log N)$ time [3, Section 2.2] and the idea has been improved by A. Lenstra [3, Section 2.2, Reference 16].

Without loss of generality, consider that $d_{p}$ is available. One can pick any random integer $W$ in $[2, N-1]$. If $\operatorname{gcd}(W, N) \neq 1$, then we already have one of the factors. Else, we consider $\operatorname{gcd}\left(W^{e d_{p}-1}-1, N\right)$. First note that $p$ divides $W^{e d_{p}-1}-1$. This is because, $e d_{p} \equiv 1 \bmod (p-1)$, i.e., $e d_{p}-1=k(p-1)$ for some positive integer $k$ and hence $W^{e d_{p}-1}-1=W^{k(p-1)}-1$ is divisible by $p$. Thus if $q$ does not divide $W^{e d_{p}-1}-1$ then $\operatorname{gcd}\left(W^{e d_{p}-1}-1, N\right)=p$ (this happens with probability almost equal to 1 ). If $q$ too divides $W^{e d_{p}-1}-1$, then $\operatorname{gcd}\left(W^{e d_{p}-1}-1, N\right)=N$ and the factorization is not possible (this happens with a very low probability).

Thus, when both $d_{p}, d_{q}$ are available, one can calculate both $\operatorname{gcd}\left(W^{e d_{p}-1}-1, N\right)$ and $\operatorname{gcd}\left(W^{e d_{q}-1}-1, N\right)$. If both of them are $N$ (which happens with a very low probability) then the factorization is not possible by this method.

Given $e, d_{p}, d_{q}$ and $N$, let us define,

$$
\begin{aligned}
T_{e, d_{p}, d_{q}, N}= & \{W \in[2, N-1] \mid \operatorname{gcd}(W, N)=1, \\
& \left.\operatorname{gcd}\left(W^{e d_{p}-1}-1, N\right)=N \text { and } \operatorname{gcd}\left(W^{e d_{q}-1}-1, N\right)=N\right\}, \\
T_{e, d_{p}, N}= & \left\{W \in[2, N-1] \mid \operatorname{gcd}(W, N)=1, \operatorname{gcd}\left(W^{e d_{p}-1}-1, N\right)=N\right\} \text { and } \\
T_{e, d_{q}, N}= & \left\{W \in[2, N-1] \mid \operatorname{gcd}(W, N)=1, \operatorname{gcd}\left(W^{e d_{q}-1}-1, N\right)=N\right\} .
\end{aligned}
$$

It is easy to note that $T_{e, d_{p}, d_{q}, N}=T_{e, d_{p}, N} \cap T_{e, d_{q}, N}$.
Let us now provide some examples in Table 1. Looking at Table 1, it is clear that while $\left|T_{e, d_{p}, d_{q}, N}\right|$ is quite large for one prime-pair, it is very small for the other.

| $p$ | $q$ | $e$ | $d_{p}$ | $d_{q}$ | $\left\|T_{e, d_{p}, N}\right\| \mid$ | $T_{e, d_{q}, N} \mid$ | $\left\|T_{e, d_{p}, d_{q}, N}\right\|$ |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| 1021 | 1601 | 77 | 53 | 1413 | 81599 | 543999 | 2199 |
| 1021 | 1601 | 179 | 359 | 1019 | 20399 | 95999 | 1199 |
| 1021 | 1601 | 1999 | 199 | 1199 | 203999 | 31999 | 3999 |
| 1021 | 1601 | 10019 | 479 | 779 | 101999 | 95999 | 5999 |
| 1229 | 1987 | 77 | 925 | 1367 | 2455 | 3971 | 3 |
| 1229 | 1987 | 5791 | 95 | 1213 | 2455 | 3971 | 3 |
| 1229 | 1987 | 7793 | 601 | 605 | 2455 | 7943 | 7 |
| 1229 | 1987 | 121121 | 501 | 1271 | 2455 | 3971 | 3 |

Table 1. Cardinality of $T_{e, d_{p}, d_{q}, N}$ : some toy examples.

We like to present the following technical result in this direction.
Proposition 1. Consider CRT-RSA with $N=p q$, encryption exponent e and decryption exponents $d_{p}$ and $d_{q}$. Let $g_{1}=\operatorname{gcd}(p-1, q-1), g_{p}=\operatorname{gcd}\left(e d_{p}-1, q-1\right), g_{q}=\operatorname{gcd}\left(e d_{q}-1, p-1\right)$ and $g_{e}=\operatorname{gcd}\left(e d_{p}-1, e d_{q}-1\right)$. Then $\left|T_{e, d_{p}, N}\right|=g_{p}(p-1)-1,\left|T_{e, d_{q}, N}\right|=g_{q}(q-1)-1$ and $\left|T_{e, d_{p}, d_{q}, N}\right|=g_{p} g_{q}-1$. Further, $g_{1}^{2}-1 \leq\left|T_{e, d_{p}, d_{q}, N}\right| \leq g_{e}^{2}-1$.

Proof. We have $g_{p}=\operatorname{gcd}\left(e d_{p}-1, q-1\right)$. Then there exists a subgroup $S_{q}$ of order $g_{p}$ in $\mathbb{Z}_{q}^{*}$ such that for any $w \in S_{q}$, we have $q \mid w^{g_{p}}-1$. Now consider any $w_{1} \in \mathbb{Z}_{p}^{*}$ and $w_{2}$ from $S_{q}$. By CRT, there exists a unique $W \in \mathbb{Z}_{N}^{*}$ such that $W \equiv w_{1} \bmod p$ and $W \equiv w_{2} \bmod q$ and vice versa. Thus the number of such $W^{\prime}$ 's is $g_{p}(p-1)$. It is evident that for all these $W$ 's, we have $\operatorname{gcd}(W, N)=1$ and $N \mid W^{e d_{p}-1}-1$. We can also observe that any $W \in T_{e, d_{p}, N}$ can be obtained in this way. Discarding the case $W=1$, we get $\left|T_{e, d_{p}, N}\right|=g_{p}(p-1)-1$.

Similarly, we have $g_{q}=\operatorname{gcd}\left(e d_{q}-1, p-1\right)$. Then there exists a subgroup $S_{p}$ of order $g_{q}$ in $\mathbb{Z}_{p}^{*}$ such that for any $w \in S_{p}$, we have $p \mid w^{g_{q}}-1$. In the same manner, we get $\left|T_{e, d_{q}, N}\right|=$ $g_{q}(q-1)-1$.

Now consider any $w_{1} \in S_{p}$ and $w_{2} \in S_{q}$. By CRT, there exists a unique $W \in \mathbb{Z}_{N}^{*}$ such that $W \equiv w_{1} \bmod p$ and $W \equiv w_{2} \bmod q$ and vice versa. Thus the number of such $W$ 's is $g_{p} g_{q}$. It is evident that for all these $W^{\prime}$ 's, we have $\operatorname{gcd}(W, N)=1, N \mid W^{e d_{p}-1}-1$ and $N \mid W^{e d_{q}-1}-1$. One may observe that any $W \in T_{e, d_{p}, d_{q}, N}$ can be obtained in this manner. Discarding the case $W=1$, we get $\left|T_{e, d_{p}, d_{q}, N}\right|=g_{p} g_{q}-1$.

Consider $e d_{p}-1=k(p-1)$ and $e d_{q}-1=l(q-1)$. Then we get $\left|T_{e, d_{p}, d_{q}, N}\right| \geq g_{1}^{2}-1$, as $g_{1}$ divides both $g_{p}$ and $g_{q}$. Since $g_{e}=\operatorname{gcd}\left(e d_{p}-1, e d_{q}-1\right)=\operatorname{gcd}(k(p-1), l(q-1))$, each of $g_{p}, g_{q}$ divides $g_{e}$. Thus the bounds on $\left|T_{e, d_{p}, d_{q}, N}\right|$ follow.

Given $e, N, d_{p}, d_{q}$, one can get $g_{e}$ easily, and thus the upper bound of $\left|T_{e, d_{p}, d_{q}, N}\right|$ is immediately known. If $g_{e}$ is bounded by poly $(\log N)$, then it is enough to try $g_{e}^{2}$ many distinct $W$ 's to factorize $N$ in poly $(\log N)$ time. However, from Proposition 1, it is clear that $\left|T_{e, d_{p}, d_{q}, N}\right|$
may not be bounded by poly $(\log N)$ as $g_{p}, g_{q}$ may not be bounded by poly $(\log N)$ in all the cases. Thus we have the following question, where an affirmative answer will transform the probabilistic algorithm to a deterministic one.

- Is it possible to identify a $W \in[2, N-1] \backslash T_{e, d_{p}, d_{q}, N}$ in poly $(\log N)$ time?

To our knowledge, an affirmative answer to the above question is not known. Thus, from theoretical point of view, getting a deterministic polynomial time algorithm for factorizing $N$ with the knowledge of $N, e, d_{p}, d_{q}$ is important. We solve it using lattice based technique.

### 1.2 Preliminaries on Lattices

Let us present some basics on lattice reduction techniques. Consider the linearly independent vectors $u_{1}, \ldots, u_{\omega} \in \mathbb{Z}^{n}$, where $\omega \leq n$. A lattice, spanned by $\left\{u_{1}, \ldots, u_{\omega}\right\}$, is the set of all linear combinations of $u_{1}, \ldots, u_{\omega}$, i.e., $\omega$ is the dimension of the lattice. A lattice is called full rank when $\omega=n$. Let $L$ be a lattice spanned by the linearly independent vectors $u_{1}, \ldots, u_{\omega}$, where $u_{1}, \ldots, u_{\omega} \in \mathbb{Z}^{n}$. By $u_{1}^{*}, \ldots, u_{\omega}^{*}$, we denote the vectors obtained by applying the GramSchmidt process to the vectors $u_{1}, \ldots, u_{\omega}$.

The determinant of $L$ is defined as $\operatorname{det}(L)=\prod_{i=1}^{w}\left\|u_{i}^{*}\right\|$, where $\|$.$\| denotes the Euclidean$ norm on vectors. Given a polynomial $g(x, y)=\sum a_{i, j} x^{i} y^{j}$, we define the Euclidean norm as $\|g(x, y)\|=\sqrt{\sum_{i, j} a_{i, j}^{2}}$ and infinity norm as $\|g(x, y)\|_{\infty}=\max _{i, j}\left|a_{i, j}\right|$.

It is known that given a basis $u_{1}, \ldots, u_{\omega}$ of a lattice $L$, the LLL algorithm [13] can find a new basis $b_{1}, \ldots, b_{\omega}$ of $L$ with the following properties.

- $\left\|b_{i}^{*}\right\|^{2} \leq 2\left\|b_{i+1}^{*}\right\|^{2}$, for $1 \leq i<\omega$.
- For all $i$, if $b_{i}=b_{i}^{*}+\sum_{j=1}^{i-1} \mu_{i, j} b_{j}^{*}$ then $\left|\mu_{i, j}\right| \leq \frac{1}{2}$ for all $j$.
$-\left\|b_{i}\right\| \leq 2^{\frac{\omega(\omega-1)+(i-1)(i-2)}{4(\omega-i+1)}} \operatorname{det}(L)^{\frac{1}{\omega-i+1}}$ for $i=1, \ldots, \omega$.
In [4], deterministic polynomial time algorithms have been presented to find small integer roots of (i) polynomials in a single variable $\bmod N$, and of (ii) polynomials in two variables over the integers. The idea of [4] extends to more than two variables also, but in that event, the method becomes heuristic.

In [5], a simpler algorithm than [4] has been presented in this direction, but it was asymptotically less efficient. Later, in [6], a simpler idea than [4] has been presented with the same asymptotic bound as in [4]. Both the works [5, 6] depend on the result of [8].

The result of [14], in finding the deterministic polynomial time algorithm to factorize $N$ from the knowledge of $e, d$, uses the techniques presented in [4,5]. Further, the work of [7] exploits the technique presented in [9].

## 2 Deterministic Polynomial Time Algorithm

In this section we consider that both $d_{p}, d_{q}$ are known apart from the public information $N, e$. We start with the following lemma. In the following results, we consider $p \approx N^{\gamma_{1}}$ as the bit size of $p$ can be correctly estimated in $\log N$ many attempts.

Lemma 1. Let $e=N^{\alpha}, d_{p} \leq N^{\delta_{1}}, d_{q} \leq N^{\delta_{2}}$. Suppose $p>q$ and $p \approx N^{\gamma_{1}}$. If both $d_{p}, d_{q}$ are known then one can factor $N$ in deterministic poly $(\log N)$ time if $2 \alpha+\delta_{1}+\delta_{2} \leq 2-\gamma_{1}$.

Proof. We have $e d_{p}-1=k(p-1), e d_{q}-1=l(q-1)$ for some positive integers $k, l$. So, $k l=\frac{\left(e d_{p}-1\right)\left(e d_{q}-1\right)}{(p-1)(q-1)}$. Let, $A=\frac{\left(e d_{p}-1\right)\left(e d_{q}-1\right)}{N}$. Now, $|k l-A|=\left(e d_{p}-1\right)\left(e d_{q}-1\right) \frac{N-(p-1(q-1)}{N(p-1)(q-1)} \approx$ $\frac{e d_{p} e d_{q}(p+q)}{N^{2}} \leq N^{2 \alpha+\delta_{1}+\delta_{2}+\gamma_{1}-2}$ (neglecting the small constant). So, as long as, $2 \alpha+\delta_{1}+\delta_{2} \leq$ $2-\gamma_{1}$, we have $k l=\lceil A\rceil$. After finding $k l$, one gets $(p-1)(q-1)$ and hence $p+q$ can be obtained immediately, which factorizes $N$.

In the next result, we use the idea of [4].
Theorem 1. Let $e=N^{\alpha}, d_{p} \leq N^{\delta_{1}}, d_{q} \leq N^{\delta_{2}}$. Suppose $p$ is estimated as $N^{\gamma_{1}}$. Further consider that an approximation $p_{0}$ of $p$ is known such that $\left|p-p_{0}\right|<N^{\beta}$.

Let $k_{0}=\left\lfloor\frac{e d_{p}}{p_{0}}\right\rfloor, q_{0}=\left\lfloor\frac{N}{p_{0}}\right\rfloor, l_{0}=\left\lfloor\frac{e d_{q}}{q_{0}}\right\rfloor$ and $g=\operatorname{gcd}\left(N-1, e d_{q}-1+l_{0}-l_{0} N, e d_{p}-1+k_{0}-\right.$ $\left.k_{0} N\right)=N^{\eta}$.

If both $d_{p}, d_{q}$ are known then one can factor $N$ in deterministic poly $(\log N)$ time if $\alpha^{2}+$ $\alpha \delta_{1}+2 \alpha \beta+\delta_{1} \beta-2 \alpha \gamma_{1}-\gamma_{1}^{2}+\alpha \delta_{2}+\delta_{1} \delta_{2}+\beta \delta_{2}-2 \gamma_{1} \delta_{2}-2 \beta \eta+2 \gamma \eta-\eta^{2}-\alpha-\delta_{1}+\beta+2 \eta-1<0$ provided $1+3 \gamma_{1}-2 \beta-\delta_{1}-\alpha-\eta \geq 0$.

Proof. We have $e d_{p}=1+k(p-1)$ and $e d_{q}=1+l(q-1)$. So $k=\frac{e d_{p}-1}{p-1}$. We also have $k_{0}=\frac{e d_{p}}{p_{0}}$. Then

$$
\left|k-k_{0}\right|=\left|\frac{e d_{p}-1}{p-1}-\frac{e d_{p}}{p_{0}}\right| \approx\left|\frac{e d_{p}}{p}-\frac{e d_{p}}{p_{0}}\right|=\frac{e d_{p}\left|p-p_{0}\right|}{p p_{0}} \leq N^{\alpha+\delta_{1}+\beta-2 \gamma_{1}} .
$$

Considering $q_{0}=\frac{N}{p_{0}}$, it can be shown that $\left|q-q_{0}\right|<N^{1+\beta-2 \gamma_{1}}$, neglecting the small constant. Assume, $q=N^{\gamma_{2}}$, where $\gamma_{2}=1-\gamma_{1}$. So if we take $l_{0}=\frac{e d_{q}}{q_{0}}$, then $\left|l-l_{0}\right|=\left|\frac{e d_{q}-1}{q-1}-\frac{e d_{q}}{q_{0}}\right| \approx$ $\left|\frac{e d_{q}}{q}-\frac{e d_{q}}{q_{0}}\right|=\frac{e d_{q}\left|q-q_{0}\right|}{q q_{0}} \leq N^{\alpha+\delta_{2}+1+\beta-2 \gamma_{1}-2 \gamma_{2}}=N^{\alpha+\delta_{2}+\beta-1}$.

Let $k_{1}=k-k_{0}$ and $l_{1}=l-l_{0}$. We have $e d_{p}+k-1=k p$. So $e d_{p}+k_{0}+k_{1}-1=$ $\left(k_{0}+k_{1}\right) p$. Similarly, $e d_{q}+l_{0}+l_{1}-1=\left(l_{0}+l_{1}\right) q$. Now multiplying these equations, we get $\left(e d_{p}-1+k_{0}\right)\left(e d_{q}-1+l_{0}\right)+k_{1}\left(e d_{q}-1+l_{0}\right)+l_{1}\left(e d_{p}-1+k_{0}\right)+k_{1} l_{1}=\left(k_{0}+k_{1}\right) p\left(l_{0}+l_{1}\right) q$.

Now if we substitute $k_{1}, l_{1}$ by $x, y$ respectively, then $\left(e d_{p}-1+k_{0}\right)\left(e d_{q}-1+l_{0}\right)+x\left(e d_{q}-\right.$ $\left.1+l_{0}\right)+y\left(e d_{p}-1+k_{0}\right)+x y=\left(k_{0}+x\right) p\left(l_{0}+y\right) q$. Hence we have to find the solution $k_{1}, l_{1}$ of

$$
\left(e d_{p}-1+k_{0}\right)\left(e d_{q}-1+l_{0}\right)+x\left(e d_{q}-1+l_{0}\right)+y\left(e d_{p}-1+k_{0}\right)+x y=\left(k_{0}+x\right) p\left(l_{0}+y\right) q,
$$

i.e., we have to find the roots of $f^{\prime}(x, y)=0$, where $f^{\prime}(x, y)=(1-N) x y+x\left(e d_{q}-1+l_{0}-\right.$ $\left.l_{0} N\right)+y\left(e d_{p}-1+k_{0}-k_{0} N\right)+\left(e d_{p}-1+k_{0}\right)\left(e d_{q}-1+l_{0}\right)-k_{0} l_{0} N$.

We have $g=\operatorname{gcd}\left(1-N, e d_{q}-1+l_{0}-l_{0} N, e d_{p}-1+k_{0}-k_{0} N\right)=N^{\eta}$.
Let $f(x, y)=\frac{f^{\prime}(x, y)}{g}, X=N^{\alpha+\delta_{1}+\beta-2 \gamma_{1}}$ and $Y=N^{\alpha+\delta_{2}+\beta-1}$. Clearly $X, Y$ are the upper bounds of $\left(k_{1}, l_{1}\right)$, the root of $f$. Thus, $W=\|f(x X, y Y)\|_{\infty} \geq \frac{X\left(e d_{q}-1+l_{0}-l_{0} N\right)}{g} \approx \frac{X l N}{g}=$ $N^{2 \alpha+\delta_{1}+\delta_{2}+\beta-\gamma_{1}-\eta}$. Then from [4] we need $X Y<W^{\frac{2}{3}}$, which implies

$$
\begin{equation*}
2 \alpha+\delta_{1}+\delta_{2}+2 \eta<3+4\left(\gamma_{1}-\beta\right) . \tag{1}
\end{equation*}
$$

If one of the variables $x, y$ is significantly smaller than the other, we give some extra shifts on $x$ or $y$. Without loss of generality, let us assume that $k_{1}$ is significantly smaller than $l_{1}$. Following the "Extended Strategy" of [11, Page 274], we exploit extra $t$ many shifts of $x$ where $t$ is a non-negative integer. Our aim is to find a polynomial $f_{0}$ that share the root $\left(k_{1}, l_{1}\right)$ over the integers. We define two sets of monomials as follows.

$$
\begin{gathered}
S=\bigcup_{0 \leq k \leq t}\left\{x^{i+k} y^{j}: x^{i} y^{j} \text { is a monomial of } f^{m}\right\}, \\
M=\left\{\text { monomials of } x^{i} y^{j} f: x^{i} y^{j} \in S\right\} .
\end{gathered}
$$

From [11], we know that these polynomials can be found by lattice reduction if $X^{s_{1}} Y^{s_{2}}<$ $W^{s}$ for $s_{j}=\sum_{x^{i_{1} y^{i_{2}} \in M \backslash S}} i_{j}$ where $s=|S|, j=1,2$. One can check that $s_{1}=\frac{3}{2} m^{2}+\frac{7}{2} m+$ $\frac{t^{2}}{2}+\frac{5}{2} t+2 m t+2, s_{2}=\frac{3}{2} m^{2}+\frac{7}{2} m+t+m t+2$, and $s=(m+1)^{2}+m t+t$.

Let $t=\tau m$. Neglecting the lower order terms we get that $X^{s_{1}} Y^{s_{2}}<W^{s}$ is satisfied when $\left(\frac{3}{2}+\frac{\tau^{2}}{2}+2 \tau\right)\left(\alpha+\delta_{1}+\beta-2 \gamma_{1}\right)+\left(\frac{3}{2}+\tau\right)\left(\alpha+\delta_{2}+\beta-1\right)<(1+\tau)\left(2 \alpha+\delta_{1}+\delta_{2}+\beta-\gamma_{1}-\eta\right)$, i.e., when
$\left(\frac{\alpha}{2}+\frac{\delta_{1}}{2}+\frac{\beta}{2}-\gamma_{1}\right) \tau^{2}+\left(\alpha+\delta_{1}+2 \beta-3 \gamma_{1}-1+\eta\right) \tau+\left(\alpha+\frac{\delta_{1}+\delta_{2}}{2}+2 \beta-2 \gamma_{1}-\frac{3}{2}+\eta\right)<0$.
In this case the value of $\tau$ for which the left hand side of the above inequality is minimum is $\tau=\frac{1+3 \gamma_{1}-2 \beta-\delta_{1}-\alpha-\eta}{\alpha+\delta_{1}+\beta-2 \gamma_{1}}$. As $\tau \geq 0$, we need $1+3 \gamma_{1}-2 \beta-\delta_{1}-\alpha-\eta \geq 0$. Putting this optimal value of $\tau$ we get the required condition as $\alpha^{2}+\alpha \delta_{1}+2 \alpha \beta+\delta_{1} \beta-2 \alpha \gamma_{1}-\gamma_{1}^{2}+\alpha \delta_{2}+\delta_{1} \delta_{2}+$ $\beta \delta_{2}-2 \gamma_{1} \delta_{2}-2 \beta \eta+2 \gamma \eta-\eta^{2}-\alpha-\delta_{1}+\beta+2 \eta-1<0$.

The strategy presented in [11] works in polynomial time in $\log N$. As we follow the same strategy, $N$ can be factored from the knowledge of $N, e, d_{p}, d_{q}$ in deterministic polynomial time in $\log N$.

As the condition given in Theorem 1 is quite involved, we present a few numerical values in Table 2.

| $\alpha$ | $\delta_{1}$ | $\delta_{2}$ | $\beta$ | $\gamma_{1}$ | $\eta$ |
| :---: | :---: | :---: | :---: | :---: | :---: |
| 1.01 | 0.5 | 0.5 | 0.44 | 0.5 | 0.1 |
| 1.02 | 0.45 | 0.5 | 0.47 | 0.5 | 0.06 |
| 1.01 | 0.50 | 0.51 | 0.48 | 0.5 | 0.02 |
| 0.97 | 0.51 | 0.51 | 0.5 | 0.5 | 0.02 |
| 1.00 | 0.47 | 0.47 | 0.5 | 0.5 | 0.03 |
| 1.01 | 0.40 | 0.5 | 0.5 | 0.5 | 0.04 |
| 1.01 | 0.35 | 0.5 | 0.5 | 0.5 | 0.06 |

Table 2. Numerical values of $\alpha, \delta_{1}, \delta_{2}, \beta, \gamma_{1}, \eta$ following Theorem 1 for which $N$ can be factored in poly $(\log N)$ time.

Corollary 1. Let $e=N^{\alpha}, d_{p}<N^{\delta_{1}}, d_{q}<N^{\delta_{2}}$. Let $g=\operatorname{gcd}\left(N-1, e d_{p}-1, e d_{q}-1\right)=N^{\eta}$. If $N, e, d_{p}, d_{q}$ are known then $N$ can be factored in deterministic polynomial time in $\log N$ when $2 \alpha+\delta_{1}+\delta_{2}+2 \eta<3$.

Proof. Since in this case we do not consider any approximation of $p, q$, we take $\beta=\gamma$. Putting this value of $\beta$ in Inequality 1, we get the desired result.

For practical purposes, $p, q$ are same bit size and if we consider that no information about the bits of $p$ is known, then we have $\gamma_{1}=\gamma_{2}=\beta=\frac{1}{2}$. In this case, we require $\alpha^{2}+\alpha \delta_{1}+\alpha \delta_{2}+\delta_{1} \delta_{2}-\eta^{2}-\alpha-\frac{1}{2} \delta_{1}-\frac{1}{2} \delta_{2}+2 \eta-\frac{3}{4}<0$ as well as $\frac{3}{2}-\delta_{1}-\alpha-\eta \geq 0$.

As discussed in Section 1.1, if $\left|T_{e, d_{p}, d_{q}, N}\right|$ is small, then one can easily prove the deterministic polynomial time equivalence. However, this idea cannot be applied when $\left|T_{e, d_{p}, d_{q}, N}\right|$ is large. In such an event, our lattice based technique provides a solution for certain ranges of $d_{p}, d_{q}$. In all our experiments we start with large $g_{1}$, e.g., of the order of 100 bits. In such cases, $\left|T_{e, d_{p}, d_{q}, N}\right|$ is large as $g_{1}^{2}-1 \leq\left|T_{e, d_{p}, d_{q}, N}\right|$ following Proposition 1. One may note that the $g_{1}$ in Proposition 1 divides the $g$ in Theorem 1.

Let us now present some experimental results in Table 3. Our experiments are based on the strategy of [5] as it is easier to implement. We have written the programs in SAGE 3.1.1 over Linux Ubuntu 8.04 on a computer with Dual CORE Intel(R) Pentium(R) D 1.83 GHz CPU, 2 GB RAM and 2 MB Cache. We take large primes $p, q$ such that $N$ is of 1000 bits. We like to point out that the experimental results cannot reach the theoretical bounds due to the small lattice dimensions.

| $N$ | $p$ | $q$ | $e$ | $d_{p}$ | $d_{q}$ | $g_{1}$ | LD | m,t) | $\# \mathrm{MSB}_{p}$ | $L^{3}$-time |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| 1000 bit | 500 bit | 500 bit | 1000 bit | 250 bit | 250 bit | 100 bit | 25 | $(3,0)$ | 20 | 93.40 sec |
| 1000 bit | 500 bit | 500 bit | 1000 bit | 203 bit | 313 bit | 100 bit | 30 | $(3,1)$ | 20 | 187.49 sec |
| 1000 bit | 500 bit | 500 bit | 1000 bit | 150 bit | 150 bit | 120 bit | 16 | $(2,0)$ | 0 | 14.84 sec |
| 1000 bit | 500 bit | 500 bit | 1000 bit | 150 bit | 270 bit | 120 bit | 30 | $(3,1)$ | 20 | 180.70 sec |
| 1000 bit | 500 bit | 500 bit | 1000 bit | 330 bit | 330 bit | 80 bit | 25 | $(3,0)$ | 60 | 108.36 sec |
| 1000 bit | 500 bit | 500 bit | 1000 bit | 300 bit | 300 bit | 150 bit | 25 | $(3,0)$ | 70 | 109.18 sec |

Table 3. Experimental results corresponding to Theorem 1. LD is the lattice dimension and $m, t$ are the parameters as explained in the proof of Theorem 1. The number of MSBs of $p$ to be known is denoted by $\# \mathrm{MSB}_{p}$.

## 3 Conclusion

Towards theoretical interest, we have presented a deterministic poly $(\log N)$ time algorithm that can factorize $N$ given $e, d_{p}$ and $d_{q}$ for certain ranges of $d_{p}, d_{q}$. This algorithm is based on lattice reduction techniques.

Acknowledgments: The authors like to thank Dr. A. Venkateswarlu for pointing out Proposition 1 and Mr. Sourav Sen Gupta for presenting detailed comments on this version.

## References

1. D. Boneh. Twenty Years of Attacks on the RSA Cryptosystem. Notices of the AMS, 46(2):203-213, February, 1999.
2. 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.
3. D. Boneh, R. A. DeMillo and R. J. Lipton. On the importance of eliminating errors in cryptographic computations. Journal of Cryptology, 14(2):101-119, 2001.
4. D. Coppersmith. Small Solutions to Polynomial Equations and Low Exponent Vulnerabilities. Journal of Cryptology, 10(4):223-260, 1997.
5. J. -S. Coron. Finding Small Roots of Bivariate Integer Equations Revisited. Eurocrypt 2004, LNCS 3027, pp. 492-505, 2004.
6. J. -S. Coron. Finding Small Roots of Bivariate Integer Equations: a Direct Approach. Crypto 2007, LNCS 4622, pp. 379-394, 2007.
7. J. -S. Coron and A. May. Deterministic polynomial-time equivalence of computing the RSA secret key and factoring. Journal of Cryptology, 20(1):39-50, 2007.
8. N. Howgrave-Graham. Finding Small Roots of Univariate Modular Equations Revisited. Proceedings of Cryptography and Coding, LNCS 1355, pp. 131-142, 1997.
9. N. Howgrave-Graham. Approximate integer common divisors. Proceedings of CaLC'01, LNCS 2146, pp. 51-66, 2001.
10. E. Jochemsz. Cryptanalysis of RSA Variants Using Small Roots of Polynomials. Ph. D. thesis, Technische Universiteit Eindhoven, 2007.
11. E. Jochemsz and A. May. A Strategy for Finding Roots of Multivariate Polynomials with new Applications in Attacking RSA Variants. Asiacrypt 2006, LNCS 4284, pp. 267-282, 2006.
12. E. Jochemsz and A. May. A Polynomial Time Attack on RSA with Private CRT-Exponents Smaller Than $N^{0.073}$. Crypto 2007, LNCS 4622, pp. 395-411, 2007.
13. A. K. Lenstra, H. W. Lenstra and L. Lovász. Factoring Polynomials with Rational Coefficients. Mathematische Annalen, 261:513-534, 1982.
14. A. May. Computing the RSA secret key is deterministic polynomial time equivalent to factoring. Crypto 2004, LNCS 3152, pp. 213-219, 2004.
15. A. May. Using LLL-Reduction for Solving RSA and Factorization Problems: A Survey. LLL+25 Conference in honour of the 25th birthday of the LLL algorithm, 2007. Available at http://www.informatik.tudarmstadt.de/KP/alex.html [last accessed 23 December, 2008].
16. G. L. Miller. Riemann's hypothesis and test of primality. 7th Annual ACM Symposium on the Theory of Computing, pp. 234-239, 1975.
17. 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.
18. D. R. Stinson. Cryptography - Theory and Practice. 2nd Edition, Chapman \& Hall/CRC, 2002.
19. M. Wiener. Cryptanalysis of Short RSA Secret Exponents. IEEE Transactions on Information Theory, 36(3):553-558, 1990.

[^0]:    * This paper is a corrected and revised version of the paper "Deterministic Polynomial-Time Equivalence of Computing the CRT-RSA Secret Keys and Factoring" presented in "WCC 2009, International Workshop on Coding and Cryptography", May 10-15, 2009, Ullensvang, Norway. There was a flaw in the earlier version where we claimed the deterministic polynomial time equivalence for the complete range of $d_{p}, d_{q}$. The corrected proof in this version achieves the result for a certain range of $d_{p}, d_{q}$, but not for the complete range.

