Defence Science Journal, Vol. 62, No. 2, March 2012, pp. 122-126, DOI: 10.14429/dsj.62.1716

On Deterministic Polynomial-time Equivalence of Computing the CRT-RSA Secret Keys and Factoring

Subhamoy Maitra and Santanu Sarkar

Indian Statistical Institute, Kolkata-700 108, India
E-mail: subho@isical.ac.in

ABSTRACT

Let $N = pq$ be the product of two large primes. Consider Chinese remainder theorem-Rivest, Shamir, Adleman (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 factorise 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 factorise 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, factorisation, LLL algorithm, cryptosystems

1. INTRODUCTION

RSA[17] is one of the most popular cryptosystems in the history of cryptology. Let us briefly describe the idea of 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 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[1,10,15]. Rivest[17], *et al.* itself presents a probabilistic polynomial time algorithm that on input N, e, d provides the factorisation of N; this is based on the technique provided by Miller[16,18]. It has been proved[7,14] that given N, e, d, one can factor N in deterministic poly(log N) time provided $ed \leq N^2$.

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., 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 factorised 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 $ed_p \equiv 1 \bmod (p - 1)$ and $ed_q \equiv 1 \bmod (q - 1)$,
- 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 Jochemsz & May[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 section 1.1 and 1.2. A lattice-based technique was used to show that one can factorise 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 Probabilistic Polynomial Time Algorithm

Given N, e and any one of d_p, d_q (or both), there exists a well known solution to factorise 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 factorise N in poly(log N) time [3,Section 2,2] and the idea has been improved by Lenstra[13].

Without loss of generality, consider that d_p is available. One can pick any random integer W in $[2, N - 1]$. If $\gcd(W, N) \neq 1$, then we already have one of the factors. Else, we consider gcd ($W^{ed_p-1} - 1$, N). First note that p divides $W^{ed_p-1} - 1$. This is because, $ed_p \equiv 1 \bmod (p - 1)$, i.e., $ed_p - 1 = k(p - 1)$ for some positive integer k and hence $W^{ed_p-1} - 1 = W^{k(p-1)} - 1$ is divisible by p. Thus if q does not divide $W^{ed_p-1} - 1$ then

Note: 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 International Workshop on Coding and Cryptography, 10-15 May 2009, Ullensvang, Norway.

Received 11 November 2011, online published 13 March 2012

gcd($W^{ed_p-1}-1$, N) = p (this happens with probability almost equal to 1). If q too divides $W^{ed_p-1}-1$, then gcd($W^{ed_p-1}-1$, N) = N and the factorisation is not possible (this happens with a very low probability).

Thus, when both d_p, d_q are available, one can calculate both gcd($W^{ed_p-1}-1$, N) and gcd($W^{ed_q-1}-1$, N). If both of them are N (which happens with a very low probability) then the factorisation is not possible by this method.

Given e, d_p, d_q and N, let us define,

$T_{e,d_p,d_q,N} = \{W \in [2, N-1] \mid \gcd(W, N)=1, \gcd(W^{ed_p-1}-1, N) = N \text{ and } \gcd(W^{ed_q-1}-1, N) = N\}$

$T_{e,dp,N} = \{W \in [2, N-1] \mid \gcd(W, N)=1, \gcd(W^{ed_p-1}-1, N) = N\}$ and

$T_{e,d_q,N} = \{W \in [2, N-1] \mid \gcd(W, N)=1, \gcd(W^{ed_q-1}-1, N) = N\}$.

Table 1. Cardinality of $T_{e,d_p,d_q,N}$: some toy examples

p	q	e	d_p	d_q	$\lvert T_{e,d_p,N}\rvert$	$\lvert T_{e,d_q,N}\rvert$	$\lvert T_{e,d_p,d_q,N}\rvert$
1021	1601	77	53	1413	81599	543999	27199
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

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. It is clear that while $|T_{e,d_p,d_q,N}|$ is quite large for one prime-pair, it is very small for the other.

Proposition 1

Consider CRT-RSA with $N = pq$, encryption exponent e and decryption exponents d_p and d_q. Let g_1 = gcd($p-1$, $q-1$), g_p=gcd(ed_p-1, $q-1$), g_q = gcd(ed_q-1, $p-1$) and g_e = gcd(ed_p-1, ed_q-1). Then $|T_{e,d_p,N}| = g_p(p-1)-1$, $|T_{e,d_q,N}| = g_q(q-1)-1$ and $|T_{e,d_p,d_q,N}| = g_p g_q - 1$. Further, $g_1^2 - 1 \le |T_{e,d_p,d_q,N}| \le g_e^2 - 1$.

Proof

We have g_p = gcd($ed_p - 1$, $q - 1$). 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 $q|w^{g_p} - 1$. 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's is $g_p(p-1)$. It is evident that for all these W's, we have gcd(W, N) = 1 and $N|W^{ed_p-1} - 1$. 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 $|T_{e,d_p,N}| = g_p(p-1) - 1$.

Similarly, we have g_q = gcd($ed_q - 1$, $p - 1$). 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 $p|w^{g_q} - 1$. In the same manner, we get $|T_{e,d_q,N}| = g_q(q-1) - 1$.

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's, we have gcd(W, N) = 1, $N|W^{ed_p-1} - 1$ and $N|W^{ed_q-1} - 1$. 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 $|T_{e,d_p,d_q,N}| = g_p g_q - 1$.

Consider $ed_p - 1 = k(p-1)$ and $ed_q - 1 = l(q-1)$. Then we get $|T_{e,d_p,d_q,N}| \ge g_1^2 - 1$, as g_1 divides both g_p and g_q. Since g_e = gcd($ed_p - 1$, $ed_q - 1$) = gcd($k(p-1)$, $l(q-1)$), each of g_p, g_q divides g_e. Thus the bounds on $|T_{e,d_p,d_q,N}|$ follow.

Given e, N, d_p, d_q, one can get g_e easily, and thus the upper bound of $|T_{e,d_p,d_q,N}|$ 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 factorise N in poly(log N) time. However, from proposition 1, it is clear that $|T_{e,d_p,d_q,N}|$ 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 $W \in [2, N-1] \setminus T_{e,d_p,d_q,N}$ 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 factorising 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 $\omega \le n$. A lattice, spanned by $\{u_1,\ldots,u_\omega\}$, is the set of all linear combinations of $u_1,\ldots,u_\omega$, i.e., ω 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 $u_1^*,\ldots\ldots,u_w^*$, we denote the vectors obtained by applying the Gram-Schmidt process to the vectors $u_1,\ldots,u_\omega$.

The determinant of L is defined as $\det(L) = \prod_{i=1}^{w} \| u_i^* \|$, 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 $\|g(x,y)\|_\infty = \max_{i,j} |a_{i,j}|$.

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.

– $\|b_i^*\|^2 \le 2\|b_{i+1}^*\|^2$, for $1 \le i < \omega$.

– For all i, if $b_i = b_i^* + \sum_{j=1}^{i-1} \mu_{i,j} b_j^*$ then $|\mu_{i,j}| \le \frac{1}{2}$ for all j.

– $\|b_i\| \le 2^{\frac{\omega(\omega-1)+(i-1)(i-2)}{4(\omega-i+1)}} \det(L)^{\frac{1}{\omega-i+1}}$ for $i = 1,\ldots,\omega$.

Deterministic polynomial time algorithms has been presented by Coppersmith[4] to find small integer roots of (i) polynomials in a single variable mod N, and of (ii) polynomials in two variables over the integers. The idea of Coppersmith[4] extends to more than two variables also, but in that event, the method becomes heuristic.

A simpler algorithm by Coron[5], than Coppersmith[4] has been presented in this direction, but it was asymptotically less efficient. Later, a simpler idea by Coron[6] than Coppersmith[4] has been presented with the same asymptotic bound as in Coppersmith[4]. Both the works of Coron[5,6] depends on the result of Howgrave-Graham[8].

The results of May[14], in finding the deterministic polynomial time algorithm to factorise N from the knowledge of e, d, uses the techniques presented by Coppersmith[4] & Coron[5]. Further, the work of Coron and May[7] exploits the technique presented in Howgrave-Graham[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 $e = N^{\alpha}$, $d_p \le N^{\delta_1}$, $d_q \le N^{\delta_2}$. 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 $2\alpha + \delta_1 + \delta_2 \le 2 - \gamma_1$.

Proof

We have $ed_p - 1 = k(p-1)$, $ed_q - 1 = l(q-1)$ for some positive integers k, l.

So, $kl = \frac{(ed_p-1)(ed_q-1)}{(p-1)(q-1)}$

Let $A = \frac{(ed_p-1)(ed_q-1)}{N}$

Now

$$|kl - A| = (ed_p - 1)(ed_q - 1)\frac{N-(p-1)(q-1)}{N(p-1)(q-1)}$$
$$\approx \frac{ed_p ed_q (p+q)}{N^2} \le N^{2\alpha+\delta_1+\delta_2+\gamma_1-2}$$

(neglecting the small constant).

So, as long as, $2\alpha + \delta_1 + \delta_2 \le 2 - \gamma_1$, we have $kl = \lceil A \rceil$. After finding kl, one gets $(p-1)(q-1)$ and hence $p+q$ can be obtained immediately, which factorises N. In the next result, we use the idea of Coppersmith[4].

Theorem 1

Let $e = N^{\alpha}$, $d_p \le N^{\delta_1}$, $d_q \le N^{\delta_2}$. Suppose p is estimated as N^{γ_1}. Further consider that an approximation p_0 of p is known such that $|p - p_0| < N^{\beta}$.

Let $k_0 = \left\lfloor \frac{ed_p}{p_0} \right\rfloor, q_0 = \left\lfloor \frac{N}{p_0} \right\rfloor, l_0 = \left\lfloor \frac{ed_q}{q_0} \right\rfloor$ and

$g = \gcd(N-1, ed_q - 1 + l_0 - l_0 N, ed_p - 1 + k_0 - k_0 N) = N^{\eta}$

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 $1 + 3\gamma_1 - 2\beta - \delta_1 - \alpha - \eta \ge 0$.

Proof

We have $ed_p = 1 + k(p-1)$ and $ed_q = 1 + l(q-1)$. So $k = \frac{ed_p - 1}{p-1}$. We also have $k_0 = \frac{ed_p}{p_0}$.

Then,

$$|k - k_0| = \left|\frac{ed_p - 1}{p-1} - \frac{ed_p}{p_0}\right| \approx \left|\frac{ed_p}{p} - \frac{ed_p}{p_0}\right| = \frac{ed_p|p-p_0|}{pp_0} \le N^{\alpha+\delta_1+\beta-2\gamma_1}$$

Considering $q_0 = \frac{N}{p_0}$, it can be shown that $|q - q_0| < N^{1+\beta-2\gamma_1}$, neglecting the small constant. Assume, $q = N^{\gamma_2}$, where $\gamma_2 = 1 - \gamma_1$. So if we take $l_0 = \frac{ed_q}{p_0}$.

then

$$|l - l_0| = \left|\frac{ed_q - 1}{q-1} - \frac{ed_q}{q_0}\right| \approx \left|\frac{ed_q}{q} - \frac{ed_q}{q_0}\right|$$
$$= \frac{ed_q|q-q_0|}{qq_0} \le N^{\alpha+\delta_2+1+\beta-2\gamma_1-2\gamma_2} = N^{\alpha+\delta_2+\beta-1}$$

Let $k_1 = k - k_0$ and $l_1 = l - l_0$. We have $ed_p + k - 1 = kp$. So $ed_p + k_0 + k_1 - 1 = (k_0 + k_1)p$. Similarly, $ed_q + l_0 + l_1 - 1 = (l_0 + l_1)q$. Now multiplying these equations, we get

$$(ed_p - 1 + k_0)(ed_q - 1 + l_0) + k_1(ed_q - 1 + l_0)$$
$$+ l_1(ed_p - 1 + k_0) + k_1 l_1 = (k_0 + k_1)p(l_0 + l_1)q$$

Now if we substitute k_1, l_1 by x, y respectively, then

$$(ed_p - 1 + k_0)(ed_p - 1 + l_0) + x(ed_q - 1 + l_0)$$
$$+ y(ed_p - 1 + k_0) + xy = (k_0 + x)p(l_0 + y)q$$

Hence we have to find the solution k_1, l_1 of

$$(ed_p - 1 + k_0)(ed_q - 1 + l_0) + x(ed_q - 1 + l_0)$$
$$+ y(ed_p - 1 + k_0) + xy = (k_0 + x)p(l_0 + y)q$$

i.e., we have to find the roots of $f'(x, y) = 0$, where

$$f'(x, y) = (1 - N)xy + x(ed_q - 1 + l_0 - l_0 N)$$
$$+ y(ed_p - 1 + k_0 - k_0 N) + (ed_p - 1 + k_0)(ed_q - 1 + l_0) - k_0 l_0 N.$$

We have

$g = \gcd(1 - N, ed_q - 1 + l_0 - l_0 N, ed_p - 1 + k_0 - k_0 N) = N^{\eta}$.

Let $f(x, y) = \frac{f'(x,y)}{g}$, $X = N^{\alpha+\delta_1+\beta-2\gamma_1}$ and $Y = N^{\alpha+\delta_2+\beta-1}$. Clearly X, Y are the upper bounds of (k_1, l_1), the root of f.

Thus,

$$W = \|f(xX, yY)\|_\infty \ge \frac{X(ed_q - 1 + l_0 - l_0 N)}{g}$$
$$\approx \frac{XlN}{g} = N^{2\alpha+\delta_1+\delta_2+\beta-\gamma_1-\eta}$$

Then from Coppersmith[4] we need $XY < W^{\frac{2}{3}}$, which implies

$$2\alpha + \delta_1 + \delta_2 + 2\eta < 3 + 4(\gamma_1 - \beta) \quad (1)$$

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 Jochemsz and May[11], 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 (k_1, l_1) over the integers. We define two sets of monomials as follows.

$$S = \bigcup_{0 \le k \le t} \{x^{i+k} y^j : x^i y^j \text{ is a monomial of } f^m\}$$

$$M = \{\text{monomials of } x^i y^j f : x^i y^j \in S\}$$

From Jochemsz and May[11], we know that these polynomials can be found by lattice reduction if $X^{s_1} Y^{s_2} < W^s$ for $s_j = \sum_{x^{i_1} y^{i_2} \in M \setminus S} i_j$

where $s = |S|$, j=1, 2. One can check that

$$s_1 = \tfrac{3}{2}m^2 + \tfrac{7}{2}m + \tfrac{t^2}{2} + \tfrac{5}{2}t + 2mt + 2,$$

$$s_2 = \tfrac{3}{2}m^2 + \tfrac{7}{2}m + t + mt + 2,$$

and $s = (m+1)^2 + mt + t$

Let $t = \tau m$. Neglecting the lower order terms we get that $X^{s_1} Y^{s_2} < W^s$ is satisfied when

$$\left(\tfrac{3}{2} + \tfrac{\tau^2}{2} + 2\tau\right)(\alpha + \delta_1 + \beta - 2\gamma_1) + \left(\tfrac{3}{2} + \tau\right)(\alpha + \delta_2 + \beta - 1) < (1+\tau)(2\alpha + \delta_1 + \delta_2 + \beta - \gamma_1 - \eta)$$

i.e., when

$$\left(\tfrac{\alpha}{2} + \tfrac{\delta_1}{2} + \tfrac{\beta}{2} - \gamma_1\right)\tau^2 + (\alpha + \delta_1 + 2\beta - 3\gamma_1 - 1 + \eta)\tau + \left(\alpha + \tfrac{\delta_1+\delta_2}{2} + 2\beta - 2\gamma_1 - \tfrac{3}{2} + \eta\right) < 0$$

In this case the value of τ 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 τ 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 by Jochemsz and May[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.

Corollary 1

Let $e = N^\alpha$, $d_p < N^{\delta_1}$, $d_q < N^{\delta_2}$.

Let $g = \gcd(N-1, ed_p - 1, ed_q - 1) = N^\eta$.

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 β 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 $|T_{e,d_p,d_q,N}|$ is small, then one can easily prove the deterministic polynomial time equivalence. However, this idea cannot be applied when $|T_{e,d_p,d_q,N}|$ 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, $|T_{e,d_p,d_q,N}|$ is large as $g_1^2 - 1 \leq |T_{e,d_p,d_q,N}|$ 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 Coron[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.

Table 2. Numerical values of α, δ_1, δ_2, β, γ_1, η following Theorem 1 for which N can be factored in poly(log N) time

α	δ_1	δ_2	β	γ_1	η
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 3. Experimental results corresponding to Theorem 1

N (bit)	p (bit)	q (bit)	e (bit)	d_p (bit)	d_q (bit)	G_1 (bit)	LD	(m, t)	#MSB$_p$	L^3-time (s)
1000	500	500	1000	250	250	100	25	(3, 0)	20	93.40
1000	500	500	1000	203	313	100	30	(3, 1)	20	187.49
1000	500	500	1000	150	150	120	16	(2, 0)	0	14.84
1000	500	500	1000	150	270	120	30	(3, 1)	20	180.70
1000	500	500	1000	330	330	80	25	(3, 0)	60	108.36
1000	500	500	1000	300	300	150	25	(3, 0)	70	109.18

LD = lattice dimension, m, t are the parameters, and #MSBp = number of MSBs of p

3. CONCLUSION

Towards theoretical interest, we have presented a deterministic poly(log N) time algorithm that can factorise N given e, d_p and d_q for certain ranges of d_p, d_q. This algorithm is based on lattice reduction techniques.

ACKNOWLEDGEMENTS

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. Boneh, D. Twenty years of attacks on the RSA cryptosystem. *Notices of the AMS*, 1999, **46**(2), 203-13.
2. Boneh, D. & Durfee, G. Cryptanalysis of RSA with private key d less than $N^{0.292}$. *IEEE Trans. Inform. Theory*, 2000, **46**(4),1339-349,
3. Boneh, D.; DeMillo, R.A. & Lipton, R.J. On the importance of eliminating errors in cryptographic computations. *Journal of Cryptology*, 2001, **14**(2), 101-19.
4. Coppersmith, D. Small solutions to polynomial equations and low exponent vulnerabilities. *Journal of Cryptology*, 1997, **10**(4), 223-60.
5. Coron, J.S. Finding small roots of bivariate integer equations revisited. *In* the Eurocrypt 2004, Lecture Notes in Computer Science, **3027**, 2004, pp. 492-505.
6. Coron, J.S. Finding small roots of bivariate integer equations: A direct approach. *In* the Crypto 2007, Lecture Notes in Computer Science, **4622**, 2007, pp. 379-94.
7. Coron, J.S. & May, A. Deterministic polynomial-time equivalence of computing the RSA secret key and factoring. *Journal of Cryptology*, 2007, **20**(1), 39-50.
8. Howgrave-Graham, N. Finding small roots of univariate modular equations revisited. *In* the Proceedings of Cryptography and Coding, Lecture Notes in Computer Science, **1355**, 1997, pp.131-42.
9. Howgrave-Graham, N. Approximate integer common divisors. *In* the Proceedings of CaLC'01, Lecture Notes in Computer Science, **2146**, 2001, pp. 51-66.
10. Jochemsz, E. Cryptanalysis of RSA variants using small roots of polynomials. Technische Universiteit Eindhoven, 2007. PhD Thesis.
11. Jochemsz, E. & May, A. A Strategy for finding roots of multivariate polynomials with new applications in attacking RSA variants. *In* the Asiacrypt 2006, Lecture Notes in Computer Science, **4284**, 2006, pp. 267-82.
12. Jochemsz, E. & May, A. A polynomial time attack on RSA with private CRT-exponents smaller than $N^{0.073}$. *In* the Crypto 2007, Lecture Notes in Computer Science, **4622**, 2007, pp. 395-411.
13. Lenstra, K.; Lenstra, H.W. & Lov´asz, L. Factoring polynomials with rational coefficients. *Mathematische Annalen*, 1982, **261**, 513-34.
14. May, A. Computing the RSA secret key is deterministic polynomial time equivalent to factoring. *In* the Crypto 2004, Lecture Notes in Computer Science, **3152**, 2004, pp. 213-19.
15. May, A. Using LLL-reduction for solving RSA and factorisation problems: A survey. *In* the LLL+25 Conference in honour of the 25th birthday of the LLL algorithm, 2007. http://www.informatik.tudarmstadt.de/KP/alex.html [Accessed on 23 December, 2008].
16. Miller, G.L. Riemann's hypothesis and test of primality. *In* the 7th Annual ACM Symposium on the Theory of Computing, 1975, pp. 234–39.
17. Rivest, R.L.; Shamir, A. & Adleman, L. A method for obtaining digital signatures and public key cryptosystems. *Communications of ACM*, 1978, **21**(2), 158-64.
18. Stinson, D.R. Cryptography -Theory and practice. Ed 2nd, Chapman & Hall/CRC, 2002.
19. Wiener, M. Cryptanalysis of short RSA secret exponents. *IEEE Trans. Inform. Theo.*, 1990, **36**(3), 553-58.