Cryptanalysis of an RSA variant with Moduli N = p r q

Yao Lu, Liqiang Peng, Santanu Sarkar

Cryptanalysis of an RSA variant with Moduli $N = p^r q^l$

Yao Lu[1], Liqiang Peng[1], and Santanu Sarkar[2]

[1] State Key Laboratory of Information Security (SKLOIS)
Institute of Information Engineering (IIE)
Chinese Academy of Sciences (CAS)
Beijing 100 093, China
lywhhit@gmail.com, pengliqiang@iie.ac.cn
[2] Indian Institute of Technology Madras, Sardar Patel Road,
Chennai 600 036, India
sarkar.santanu.bir@gmail.com

Abstract. We study an RSA variant with moduli of the form $N = p^r q^l$ with $r > l \geq 2$. This variant was mentioned by Boneh et al. (Crypto 1999). Later Kim et al. (Indocrypt 2000) showed that this variant is much faster than standard RSA moduli in the decryption process. In this paper, for the first time, we give some cryptanalysis results on this RSA variant. Our analysis show that in some cases, this cryptosystem can be totally broken.

Keywords: Coppersmith's method, Lattices, RSA, RSA variants.

1 Introduction

Since the RSA public key cryptosystem has been proposed, this public key scheme is possibly the most studied topic in cryptology world and has been widely used. To achieve high efficiency in the decryption phase, many variants of RSA have been proposed.

Among them, an important one is so-called multi-power RSA [18], proposed by Takagi in 1998. In multi-power RSA, the RSA moduli N is of the form $N = p^r q$ where $r \geq 2$. Compared to the standard RSA, it is more efficient in both key generation and decryption. Besides, moduli of this type has been applied in many cryptographic designs, e.g., the Okamoto-Uchiyama cryptosystem [15], or better known via EPOC and ESIGN [4], which uses the modulus $N = p^2 q$.

In Indocrypt 2000, Lim, Kim, Yie and Lee [11] extended Takagi's cryptosystem to include moduli of the form $N = p^r q^l$, where $r, l \geq 2$. They showed that the choice of either $p^{r+1}q^r$, $p^{r+1}q^{r-1}$ or $p^{r+2}q^{r-2}$ gives optimal efficiency under the assumption that the sum of exponents are fixed. For example, they claimed that 8192-bit RSA will be 15 times faster than the standard RSA if one takes $N = p^2 q^3$. In Crypto 1999, Boneh et al. [2] also mentioned as an open problem to factor $p^r q^l$ using lattice-based approach.

Surprisingly, there had been very little research into the security RSA-type schemes with moduli $N = p^r q^l$ for $r, l \geq 2$.

1.1 Related Works

The security of this variant of RSA, like that of the standard RSA, is based on the hardness of factoring large integers. Until now there is no known polynomial-time algorithm to factorize large numbers except quantum algorithms. However, in a real-world implementation, partial information regarding the secret prime p can be leaked by side-channel attacks (known as *factoring with known bits problem*), hence it is crucial to study how this affects the factoring problem. In fact, there have been a number of results in this direction.

- For the case of the standard RSA with modulus $N = pq$: In 1985, Rivest and Shamir [16] first studied this problem, they designed an algorithm to factor N given $\frac{2}{3}$-fraction of the bits of p. In 1996, Coppersmith [3] improved their bound to $\frac{1}{2}$. Note that for the above results, the unknown bits are within one consecutive block. The case of $n(n \geq 2)$ unknown blocks was first considered by Herrmann and May in [5];
- For the case of multi-power RSA with moduli $N = p^r q$ $(r \geq 2)$: In 1999, Boneh, Durfee and Howgrave-Graham [2] showed that N can be recovered efficiently given $\frac{1}{r+1}$-fraction of the most significant bits (MSBs) of p. Recently, Lu, Zhang and Lin [12] considered the case of $n(n \geq 2)$ unknown blocks.

To speed up the decryption, the small secret exponent d is often used in some cryptographic applications. However, it is well known that the standard RSA scheme can easily be broken if the secret exponent d is too small (known as *small secret exponent attack*). In 1990, by utilizing continued fraction method, Wiener [19] showed that the standard RSA scheme can be broken when $d \leq N^{0.25}$. Later, in 1999, Boneh and Durfee [1] improved Wiener's bound up to $N^{0.292}$. Recently, in [6], Herrmann et al. gave an elementary proof for the Boneh-Durfee's bound, and in [9], Kunihiro et al. also investigated this problem. However, $N^{0.292}$ is still the best bound at present.

There are two variants of multi-power RSA. In the first variant, $ed \equiv 1 \mod p^{r-1}(p-1)(q-1)$ while in the second variant, $ed \equiv 1 \mod (p-1)(q-1)$. For the first variant, in 1999, Takagi [18] showed that when the secret exponent $d \leq N^{\frac{1}{2(r+1)}}$, one can factor N. Later in 2004, May [14] improved Takagi's bound to $N^{\max\{\frac{r}{(r+1)^2}, \frac{(r-1)^2}{(r+1)^2}\}}$. Recently, Sarkar [17] used lattice-based method to improve the previous bounds when $r \leq 5$. In [13], the authors further improved May's bound to $N^{\frac{r(r-1)}{(r+1)^2}}$, which is better than May's result when $r > 2$. For the second variant, in 2008, Itoh et al. [8] showed that d can be recovered if $d < N^{\frac{2-\sqrt{2}}{r+1}}$.

1.2 Our Contributions

In this paper, we give some analysis on the security of RSA-type schemes with moduli $N = p^r q^l$, where $r > l \geq 2$ and $\gcd(r, l) = 1$. Throughout the paper, we assume that $q < p < 2q$, which means $p \approx q$.

Factoring RSA Moduli $N = p^r q^l$ with Partial Known Bits. In the conclusion of Boneh et al.'s paper [2], authors raised a question regarding lattice-based factorization of the integers of the form $N = p^r q^l$. In this paper, we answered this question firmly that we only need a $\min\{\frac{l}{r+l}, \frac{2(r-l)}{r+l}\}$-fraction of LSBs (or MSBs) of p in order to factor N in polynomial-time. For $\min\{\frac{r+l}{l}, \frac{r+l}{2(r-l)}\} = \Omega\big(\frac{\log p}{\log\log p}\big)$, one only has to guess $\mathcal{O}(\log\log N)$ bits of p, which can be done in polynomial-time. Besides, we also extend to the case of the arbitrary number n ($n \geq 2$) of unknown blocks.

Small Secret Exponent Attacks on RSA-type Schemes with Moduli $N = p^r q^l$. Considering the form of the moduli $N = p^r q^l$, there are also two variants of encryption and decryption phases. In the first variant, (e, d) satisfy that $ed \equiv 1 \mod p^{r-1}q^{l-1}(p-1)(q-1)$. In the second variant, (e, d) satisfy that $ed \equiv 1 \mod (p-1)(q-1)$. For these two variants, we give the analysis respectively.

For $ed \equiv 1 \mod p^{r-1}q^{l-1}(p-1)(q-1)$, we solve small solution d of modular equation $ex - 1 \equiv 0 \mod p^{r-1}q^{l-1}$. We introduce a new technique to select more helpful polynomials which are used to construct the desired lattice. We show that when

$$d < N^{1-\frac{3r+l}{(r+l)^2}},$$

one can recover d in polynomial-time. Note that, when $l = 1$ our result is the same as [13].

For $ed \equiv 1 \mod (p-1)(q-1)$, we solve small solution (k, p, q) of modular equation $x(y-1)(z-1) + 1 = 0 \mod e$, where $k = \frac{ed-1}{(p-1)(q-1)}$. By utilizing the property $p^r q^l = N$, we give a method of lattice construction which is similar to Itoh et al's idea [8] and show that when

$$d < N^{\frac{1}{2(r+l)}},$$

the small solution (k, p, q) can be found. Note that, when $l = 1$ our result is a little weaker than the general bound of [8]; when $r = 1, l = 1$ our result is exactly Wiener's result [19].

Experimental Results. To verify the correctness of our above attacks, we have performed the experiments in Magma 2.11 computer algebra system on a PC with Intel(R) Core(TM) Duo CPU (2.53GHz, 1.9GB RAM Windows 7) and carried out the L^3 algorithm.

2 Useful Results

Let us first state the L^3 Algorithm[10].

Lemma 1 (L^3). *Let $\mathcal{L}$ be a lattice of dimension w. Within polynomial-time, L^3-algorithm outputs a set of reduced basis vectors v_i, $1 \leqslant i \leqslant w$ that satisfies*

$$||v_1|| \leqslant ||v_2|| \leqslant \cdots \leqslant ||v_i|| \leqslant 2^{\frac{w(w-1)}{4(w+1-i)}} \det(\mathcal{L})^{\frac{1}{w+1-i}}$$

Let $g(x_1, \cdots, x_k) = \sum_{i_1,\cdots,i_k} a_{i_1,\cdots,i_k} x_1^{i_1} \cdots x_k^{i_k}$. We define the norm of g by the Euclidean norm of its coefficient vector: $||g||^2 = \sum_{i_1,\cdots,i_k} a_{i_1,\cdots,i_k}^2$. Also we need the following result due to Howgrave-Graham [7].

Lemma 2 (Howgrave-Graham). *Let $g(x_1, \cdots, x_k) \in \mathbb{Z}[x_1, \cdots, x_k]$ be an integer polynomial that consists of at most w monomials. Suppose that*

1. *$g(y_1, \cdots, y_k) = 0 \bmod p^m$ for $| y_1 | \leqslant X_1, \cdots, | y_k | \leqslant X_k$ and*
2. *$||g(x_1X_1, \cdots, x_kX_k)|| < \frac{p^m}{\sqrt{w}}$*

Then $g(y_1, \cdots, y_k) = 0$ holds over integers.

Although our technique works in practice as noted from the experiments we perform, we need heuristic assumption for theoretical results.

Assumption 1 *The lattice-based construction yields algebraically independent polynomials. The common roots of these polynomials can be efficiently computed using the* Gröbner basis technique.

We also use the following Theorem [13].

Theorem 1. *Let N be a sufficiently large composite integer (of unknown factorization) with a divisor p^r ($p \geq N^\beta$ and an integer $r \geq 1$). Let $f(x_1, \ldots, x_n) \in \mathbb{Z}[x_1, \ldots, x_n]$ be a linear polynomial in n variables. Under Assumption 1, we can find all the solutions $(x_1^0, \ldots, x_n^0)$ of the equation $f(x_1, \ldots, x_n) = 0 \bmod p$ with $|x_1^0| \leq N^{\gamma_1}, \ldots, |x_n^0| \leq N^{\gamma_n}$ if*

$$\sum_{i=1}^{n} \gamma_i < \frac{1}{r}\left(1 - (1 - r\beta)^{\frac{n+1}{n}} - (n+1)(1 - r\beta)\left(1 - \sqrt[n]{1 - r\beta}\right)\right)$$

The running time of the algorithm is polynomial in $\log N$ but exponential in n.

3 Factoring RSA Moduli $N = p^r q^l$ with Partial Known Bits

In this section, we assume that we are given the number of k LSBs of p: $\tilde{p}$ ($\tilde{p} = p \bmod 2^k$). Our goal is to determinate the minimal amount of bits of p that one has to know in order to factor N in polynomial-time. Following we give two methods to solve this problem.

The Attack Modulo p The above problem can be reduced to solving modular univariate polynomial equation

$$f(x) = \tilde{p} + 2^k x = 0 \bmod p$$

We can apply Theorem 1 with $n = 1$, $\beta = \frac{1}{r+l}$. Therefore, we can find all the roots y if $|y| \leq N^{\frac{r}{(r+l)^2}}$. When $l = 1$, the bound is $N^{\frac{r}{(r+l)^2}} = N^{\frac{r}{(r+1)^2}} = p^{\frac{r}{r+1}}$, which is exactly the same as [2]. Since $N^{\frac{r}{(r+l)^2}} = p^{\frac{r}{r+l}}$, attacker has to guess $(1 - \frac{r}{r+l}) \log_2 p = \frac{l}{r+l} \log_2 p$ LSBs of p. Thus the total complexity to factor $N = p^r q^l$ is $\exp\left(\frac{l}{r+l} \log_2 p\right) \cdot P(\log N)$ where P is a polynomial. Thus when $\frac{r+l}{l} = \Omega\left(\frac{\log p}{\log \log p}\right)$, the running time of our attack is polynomial.

Table 1. Factoring N with partial known bits of p.

(r,l)	$\log_2 N$	$\log_2 p$	attack modulo p				attack modulo pq			
			theo.	expt.	dim(L)	time (in sec.)	theo.	expt.	dim(L)	time (in sec.)
$(3,2)$	2500	500	200	260	21	19.095	200	260	21	760.661
$(3,2)$	2500	500	200	230	41	832.983	200	230	41	42447.935
$(5,2)$	3500	500	143	260	21	21.856	429	–	21	–
$(5,2)$	3500	500	143	200	41	1205.591	429	497	41	86495.347
$(5,4)$	4500	500	223	330	21	32.245	112	260	21	4018.133
$(5,4)$	4500	500	223	280	41	1413.463	112	230	41	163533.305

The Attack Modulo pq We rewrite N as $N = (pq)^l p^{r-l}$. Suppose we know k LSBs of pq. Let $c \equiv \tilde{p}\tilde{q} \bmod 2^k$. Now we reduce the above problem to solving a modular univariate polynomial equation

$$f(x) = c + 2^k x = 0 \bmod pq$$

Applying Theorem 1 with $n = 1$, $\beta = \frac{2}{r+l}$. Thus we can find all the roots y if $|y| \leq N^{\frac{4l}{(r+l)^2}}$. After we get the value of pq, we can calculate $p^{r-l} = \frac{N}{(pq)^l}$. Then we can get p. Since $N^{\frac{4l}{(r+l)^2}} = (pq)^{\frac{2l}{r+l}}$, attacker has to guess$(1 - \frac{2l}{r+l}) \log_2 pq = \frac{r-l}{r+l} \log_2 pq = \frac{2(r-l)}{r+l} \log_2 p$ many bits. Thus the total complexity to factor $N = p^r q^l$ is $\exp\left(\frac{2(r-l)}{r+l} \log_2 p\right) \cdot P(\log N)$ where P is a polynomial. Thus when $\frac{r+l}{2(r-l)} = \Omega\big(\frac{\log p}{\log \log p}\big)$, the running time of our attack is polynomial.

Note that our first attack (modulo p) is superior to our second attack (modulo pq) in the case $2r > 3l$. Our second attack performs better for the cases that r, s are approximately the same size i.e. $p^{r+1}q^r$, $p^{r+1}q^{r-1}$.

The analysis for the MSBs case is similar to the LSBs case, so we omit the details here. Let $\frac{r+l}{l} = (\log p)^\epsilon$.

Experimental Results. In Table 1, we list the theoretical and experimental results. In the case of $(5, 2)$, we failed to give an experimental data with a 21-dimensional lattice. Note that, since the size of the entries of lattice which is used in the attack modulo pq are roughly as N, the running time of L^3 of attack modulo pq is much longer than attack modulo p.

3.1 Extend to More Unknown Blocks

We also consider the scenario that the number of unknown blocks in the secret prime p is n (n is large).

Theorem 2. *Let $N = p^r q^l$ where p and q are of equal length. Suppose a $\frac{l}{r} \ln(\frac{r+l}{l})$-fraction of the bits are known for n blocks in p (n is large), then under Assumption 1, we can recover p. The running time of the algorithm is polynomial in $\log N$ but exponential in n.*

Proof. We can reduce the above problem to solving the following multivariate linear polynomial equation

$$f(x_1, x_2, \ldots, x_n) = a_0 + a_1 x_1 + a_2 x_2 + \cdots + a_n x_n = 0 \bmod p$$

where $a_k = 2^l$ if the k-th unknown blocks starts in the l-th bit position. Moreover, if n goes to infinity, from Theorem 1, we have $\lim_{n\to\infty} \left(\frac{1}{r}\left(1-(1-r\beta)^{\frac{n+1}{n}} - (n+1)(1-r\beta)\left(1-\sqrt[n]{1-r\beta}\right)\right)\right) = \beta + \frac{(1-r\beta)\ln(1-r\beta)}{r}$. It shows that if n is very large, we can recover p regardless of n. Conversely, once a $\left(1-\frac{1}{r\beta}\right)\ln(1-r\beta)$ -fraction of the bits from p together with their positions are given, we are able to recover the missing bits. Since $p \approx q$, i.e. $\beta = \frac{1}{r+l}$, we need a $\left(1-\frac{1}{r\beta}\right)\ln(1-r\beta) = \left(1-\frac{r+l}{r}\right)\ln(1-\frac{r}{r+l}) = -\frac{l}{r}\ln(\frac{l}{r+l}) = \frac{l}{r}\ln(\frac{r+l}{l})$-fraction of known bits from p. □

4 Small Secret Exponent Attacks on RSA-type Schemes with Moduli $N = p^r q^l$.

In this section we consider the situation when the secret exponent d is small.

4.1 The First Variant

At first, we study the first variant of encryption and decryption phases: (e, d) satisfy that $ed \equiv 1 \mod p^{r-1}q^{l-1}(p-1)(q-1)$.

Theorem 3. *Let $N = p^r q^l$, where r, l $(r > l)$ are two known positive integers and p, q are primes of the same bit-size. Let e be the public key exponent and d be the private key exponent, satisfying $ed \equiv 1 \bmod \phi(N)$. Suppose $d < N^{1-\frac{3r+l}{(r+l)^2}}$ Then N can be factored in polynomial-time.*

Proof. Since $\phi(N) = p^{r-1}q^{l-1}(p-1)(q-1)$, we have the following equation $ed-1 = kp^{r-1}q^{l-1}(p-1)(q-1)$ for some $k \in \mathbb{N}$. Then we want to find the root $x_0 = d$ of the polynomial $f_1(x) = ex - 1 \bmod p^{r-1}q^{l-1}$. Multiplying the inverse of e modulo N, we can obtain the following equation

$$f(x) = E - x \bmod p^{r-1}q^{l-1}$$

where E denotes the inverse of e modulo N. Note that $N(N \equiv 0 \bmod p^r q^l)$ is a known multiple of unknown $p^{r-1}q^{l-1}$.

Since $r > l$, we define the following collection of polynomials:

$$g_i(x) := f^i(x) N^{\max\{0, \lceil \frac{(r-1)(t_1-i)}{r} \rceil, \lceil \frac{(l-1)(t_2-i)}{l} \rceil\}}$$

for $i = 0, \ldots, m$ and positive integer parameters m, t_1 and t_2 with $t_1 = \tau_1 m$, $t_2 = \tau_2 m$ $(0 \le \tau_1, \tau_2 < 1)$, which will be optimized later.

Note that for all i, $g_i(d) \equiv 0 \bmod (p^{(r-1)t_1}q^{(l-1)t_2})$.

Let $X(= N^{\gamma})$ be the upper bound on the desired root d. We built a lattice $\mathcal{L}$ of dimension $\omega = \dim(\mathcal{L}) = m+1$ using the coefficient vectors of $g_i(xX)$ as basis vectors. We sorted the polynomials according to the ascending order of g, i.e., $g_i < g_j$ if $i < j$.

From the triangular matrix of the lattice basis, we can compute the determinant as the product of the entries on the diagonal as $\det(\mathcal{L}) = X^s N^{s_N}$. Now $s = \sum_{i=0}^{m} i = \frac{m(m+1)}{2} = \frac{m^2}{2} + o(m^2)$. The computation of s_N is somewhat complicated. At first, we have $t_1 < t_2$. Otherwise, since $r > l$, we have $\lceil \frac{(r-1)(t_1-i)}{r} \rceil \geq \lceil \frac{(l-1)(t_2-i)}{l} \rceil$ for $i = 0, \ldots, t_1$, in this case, we only consider the exponents of p. Therefore, we let $t_1 < t_2$ to consider the exponents of p and q at the same time.

Define Δ as

$$\Delta := \lceil \frac{l(r-1)t_1 - r(l-1)t_2}{r-l} \rceil$$

Note that $\Delta < t_1 < t_2$. In order to get $\Delta > 0$, we have to satisfy the following condition

$$l(r-1)t_1 > r(l-1)t_2 \tag{1}$$

Notice that for $i = 0, 1, \ldots, \Delta - 1$, we have $\lceil \frac{(r-1)(t_1-i)}{r} \rceil > \lceil \frac{(l-1)(t_2-i)}{l} \rceil$. However, for $i = \Delta, \Delta+1, \ldots, t_2$, we have $\lceil \frac{(r-1)(t_1-i)}{r} \rceil < \lceil \frac{(l-1)(t_2-i)}{l} \rceil$. Then $s_N = \sum_{i=0}^{\Delta-1} \lceil \frac{(r-1)(t_1-i)}{r} \rceil + \sum_{i=\Delta}^{t_2} \lceil \frac{(l-1)(t_2-i)}{l} \rceil = \frac{(r-1)(2t_1\Delta - \Delta^2)}{2r} + \frac{(l-1)(t_2-\Delta)^2}{2l} + o(m^2)$ $= \frac{(r-1)\left(l(r-1)t_1^2 - 2r(l-1)t_1^2t_2^2 + r(l-1)t_2^2\right)}{2r(r-l)} + o(m^2)$.

To obtain a polynomial with short coefficients that contains all small roots over integer, we apply L^3-basis reduction algorithm to the lattice $\mathcal{L}$. Lemma 1 gives us an upper bound on the norm of the shortest vector in the L^3-reduced basis, if the bound is smaller than the bound given in Lemma 2, we can obtain the desired polynomial. We require $2^{\frac{\omega-1}{4}}\sqrt{\omega}\det(\mathcal{L})^{\frac{1}{\omega}} < N^{\frac{(r-1)t_1+(l-1)t_2}{r+l}}$, where $\omega = m+1$. Putting the values of $\det(\mathcal{L})$ and ω and neglecting the terms that do not depend on N, we obtain $X^{\frac{m^2}{2}+o(m^2)} < N^{\frac{m((r-1)t_1+(l-1)t_2)}{r+l} - \frac{(r-1)\left(l(r-1)t_1^2 - 2r(l-1)t_1^2t_2^2 + r(l-1)t_2^2\right)}{2r(r-l)} + o(m^2)}$. To obtain the asymptotic bound, we let $m \to \infty$. Then we have

$$X < N^{\frac{2(r-1)\tau_1 + 2(l-1)\tau_2}{r+l} - \frac{(r-1)\left(l(r-1)\tau_1^2 - 2r(l-1)\tau_1^2\tau_2^2 + r(l-1)\tau_2^2\right)}{r(r-l)}}.$$

Now we have to decide the optimized values of τ_1 and τ_2. We consider the exponent of N as a function $h(\tau_1, \tau_2) = \frac{2(r-1)\tau_1 + 2(l-1)\tau_2}{r+l} - \frac{(r-1)\left(l(r-1)\tau_1^2 - 2r(l-1)\tau_1^2\tau_2^2 + r(l-1)\tau_2^2\right)}{r(r-l)}$. Using $h'_{\tau_1}(\tau_1, \tau_2) = 0$ and $h'_{\tau_2}(\tau_1, \tau_2) = 0$, we have

$$\begin{aligned} l(r-1)(r+l)\tau_1 - r(l-1)(r+l)\tau_2 + r(l-r) &= 0 \\ (r-1)(r+l)\tau_1 - (r-1)(r+l)\tau_2 + r - l &= 0 \end{aligned}$$

Table 2. The fist variant: experimental results for small d

(r,l)	theo.	$\dim(L)=20$		$\dim(L)=40$	
		expt.	time (in sec.)	expt.	time (in sec.)
$(3,2)$	0.560	0.520	77.751	0.530	4433.798
$(5,2)$	0.653	0.600	64.257	0.620	4177.972
$(4,3)$	0.694	0.650	61.059	0.660	3209.409
$(5,3)$	0.719	0.650	52.120	0.680	2894.411

Solving the above equations, we get

$$\tau_1 = \frac{r(r+l-2)}{(r+l)(r-1)}, \qquad \tau_2 = 1.$$

Putting the values of τ_1 and τ_2 in (1), we note that the condition is satisfied. After some calculations, we can get the required result. □

Experimental Results. Table 2 lists some theoretical and experimental results with 1000-bit N. In all experiments, we obtained an univariate integer equation with desired integer solution d. Hence, the root d can be solved out.

4.2 The Second Variant

Now we study the second variant of encryption and decryption phases: (e,d) satisfy that $ed \equiv 1 \mod (p-1)(q-1)$. Our result is as follows.

Theorem 4. *Let $N = p^r q^l$, where r, l $(r > l)$ are two known positive integers and p, q are primes of the same bit-size. Let e be the public key exponent and d be the private key exponent, satisfying $ed \equiv 1 \bmod (p-1)(q-1)$. Suppose $d < N^{\frac{1}{2(r+l)}}$. Then N can be factored in polynomial-time.*

Proof. Since $ed - 1 = k(p-1)(q-1)$ for some $k \in \mathbb{N}$, we have the following modular equation

$$f(x,y,z) = x(y-1)(z-1) + 1 \bmod e.$$

Obviously, (k,p,q) is the desired solution. Then we have an estimation on the desired roots. Since $N = p^r q^l$ and p, q are primes of the same bit-size, p and q can be estimated as $N^{\frac{1}{r+l}}$. Letting $e = N^\alpha$, we have $p, q \simeq e^{\frac{1}{\alpha(r+l)}}$. Furthermore, let $d < N^\delta$. Then k can be bounded as $k = \frac{ed-1}{(p-1)(q-1)} < \frac{2ed}{pq} < 2e^{1+\frac{\delta}{\alpha}-\frac{2}{\alpha(r+l)}}$. Usually, α is chosen as $\frac{2}{r+l}$. In this case, we have $p, q \simeq Y = Z = e^{\frac{1}{2}}$, $k \simeq X = e^{\frac{r+l}{2}\delta}$. Then in order to solve out desired solution, we define a list G of polynomials sharing the desired root modulo e^m, $g_{i,j,k,l}(x,y,z) = x^i y^j z^k f(x,y,z)^l e^{m-l}$. To make the matrix triangular whose vectors are corresponding to the coefficients of polynomials, we need to append polynomials to list G as following ordered,

```
G=[]
    for u = 0 to m do:
        for i = 0 to u − 1 do:
            for j = 0 to 1 do: append g_{u−i,j,0,i} to G
            for j = r − 1 to 1 do: append g_{u−i,j,1,i} to G
            for j = l − 1 to 1 do: append g_{u−i,r,j,i} to G
return G,
```

where each occurrence of $y^r z^l$ is replaced by N since $N = p^r q^l$.

Then we construct a lattice L which is spanned by the coefficient vectors of $g_{i,j,k,l}(xX, yY, zZ)$. By some calculations, the determinant of L is $\det(L) = X^{S_x} Y^{S_y} Z^{S_z} e^{S_e}$, where $S_x = \left(\frac{r+l}{3}\right) m^3 + o(m^3), S_y = \left(\frac{l}{6}\right) m^3 + o(m^3), S_z = \left(\frac{r}{6}\right) m^3 + o(m^3), S_e = \left(\frac{r+l}{3}\right) m^3 + o(m^3)$. On the other hand, the $\dim(L) = \left(\frac{r+l}{2}\right) m^2 + o(m^2)$.

Based on the Lemma 1 and Lemma 2, one can obtain polynomial equations which share the roots (k, p, q) when

$$X^{S_x} Y^{S_y} Z^{S_z} e^{S_e} \leq e^{m \dim(L)}.$$

Putting the upper bounds into the above inequality and neglecting the lower order terms of m, we obtain that

$$\left(\frac{r+l}{3}\right)\left(\frac{r+l}{2}\right)\delta + \left(\frac{l}{6}\right)\frac{1}{2} + \left(\frac{r}{6}\right)\frac{1}{2} - \left(\frac{r+l}{6}\right) < 0,$$

namely,

$$\delta < \frac{1}{2(r+l)}.$$

This concludes the proof of Theorem 4. □

Remark. Note that our lattice construction in the proof of Theorem 4 is not optimal. By utilizing the similar idea of Itoh et al. [8], we can select more helpful polynomials to construct our lattice. However, the improvement of theoretical bound is not very large. Moreover, the result of Theorem 4 is relatively intuitive, hence the improved lattice construction and the responding calculations are not further discussed in this paper.

Experimental Results. Table 3 lists some theoretical and experimental results. In all experiments, we obtained several integer equations which share desired roots and successfully solved out the roots by using Gröbner basis technique.

References

1. D. Boneh and G. Durfee. Cryptanalysis of RSA with private key d less than $N^{0.292}$. *IEEE Transactions on Information Theory*, 46(4):1339–1349, 2000.

Table 3. The second variant: experimental results for small d

(r, l)	$\log_2 N$	theo. d	$\dim(L) = 81$		$\dim(L) = 148$	
			expt. d	time (in sec.)	expt. d	time (in sec.)
$(3, 2)$	2000	200 bits	29 bits	35.350	71 bits	2573.002
$(3, 2)$	3000	300 bits	47 bits	103.600	110 bits	5197.392

2. D. Boneh, G. Durfee, and N. Howgrave-Graham. Factoring $N = p^r q$ for large r. In *Advances in Cryptology–CRYPTO'99*, pages 787–787. Springer, 1999.
3. D. Coppersmith. Small solutions to polynomial equations, and low exponent RSA vulnerabilities. *Journal of Cryptology*, 10(4):233–260, 1997.
4. The EPOC and the ESIGN Algorithms. IEEE P1363: Protocols from Other Families of Public-Key Algorithms. http://grouper.ieee.org/groups/1363/StudyGroup/NewFam.html, 1998.
5. M. Herrmann and A. May. Solving linear equations modulo divisors: On factoring given any bits. *Advances in Cryptology-ASIACRYPT 2008*, pages 406–424, 2008.
6. M. Herrmann and A. May. Maximizing small root bounds by linearization and applications to small secret exponent RSA. *Public Key Cryptography–PKC 2010*, pages 53–69, 2010.
7. N. Howgrave-Graham. Finding small roots of univariate modular equations revisited. *Crytography and Coding*, pages 131–142, 1997.
8. K. Itoh, N. Kunihiro, and K. Kurosawa. Small secret key attack on a variant of RSA (due to Takagi). In *Proceedings of the 2008 The Cryptopgraphers' Track at the RSA conference on Topics in cryptology*, pages 387–406. Springer-Verlag, 2008.
9. N. Kunihiro, N. Shinohara, and T. Izu. A unified framework for small secret exponent attack on RSA. In *Selected Areas in Cryptography*, pages 260–277. Springer, 2012.
10. A. K. Lenstra, H. W. Lenstra, and L. Lovász. Factoring polynomials with rational coefficients. *Mathematische Annalen*, 261(4):515–534, 1982.
11. S. Lim, S. Kim, I. Yie, and H. Lee. A generalized takagi-cryptosystem with a modulus of the form $p^r q^s$. In *Progress in Cryptology-INDOCRYPT 2000*, pages 283–294. Springer, 2000.
12. Y. Lu, R. Zhang, and D. Lin. Factoring multi-power RSA modulus $N = p^r q$ with partial known bits. In *Information Security and Privacy*, pages 57–71. Springer, 2013.
13. Y. Lu, R. Zhang, and D. Lin. New results on solving linear equations modulo unknown divisors and its applications. 2014. http://eprint.iacr.org/.
14. A. May. Secret exponent attacks on RSA-type schemes with moduli $N = p^r q$. *Public Key Cryptography–PKC 2004*, pages 218–230, 2004.
15. T. Okamoto and S. Uchiyama. A new public-key cryptosystem as secure as factoring. *Advances in Cryptography–Eurocrypt'98*, pages 308–318, 1998.
16. R. Rivest and A. Shamir. Efficient factoring based on partial information. In *Advances in Cryptology–EUROCRYPT'85*, pages 31–34. Springer, 1986.
17. S. Sarkar. Small secret exponent attack on RSA variant with modulus $N = p^r q$. *Designs, Codes and Cryptography*, pages 1–10, 2014.
18. T. Takagi. Fast RSA-type cryptosystem modulo $p^k q$. In *Advances in Cryptology–Crypto'98*, pages 318–326. Springer, 1998.
19. M. J. Wiener. Cryptanalysis of short RSA secret exponents. *IEEE Transactions on Information Theory*, 36(3):553–558, 1990.