DE GRUYTER J. Math. Cryptol. 2017; 11 (2):117–130

Research Article

Yao Lu, Liqiang Peng and Santanu Sarkar*

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

DOI: 10.1515/jmc-2016-0025
Received May 7, 2016; revised January 18, 2017; accepted April 23, 2017

Abstract: In this paper we study an RSA variant with moduli of the form $N = p^r q^l$ ($r > l \geq 2$). This variant was mentioned by Boneh, Durfee and Howgrave-Graham [2]. Later Lim, Kim, Yie and Lee [11] showed that this variant is much faster than the standard RSA moduli in the step of decryption procedure. There are two proposals of RSA variants when $N = p^r q^l$. In the first proposal, the encryption exponent e and the decryption exponent d satisfy $ed \equiv 1 \bmod p^{r-1}q^{l-1}(p-1)(q-1)$, whereas in the second proposal $ed \equiv 1 \bmod (p-1)(q-1)$. We prove that for the first case if $d < N^{1-(3r+l)(r+l)^{-2}}$, one can factor N in polynomial time. We also show that polynomial time factorization is possible if $d < N^{(7-2\sqrt{7})/(3(r+l))}$ for the second case. Finally, we study the case when few bits of one prime are known to the attacker for this variant of RSA. We show that given $\min(\frac{l}{r+l}, \frac{2(r-l)}{r+l}) \log_2 p$ least significant bits of one prime, one can factor N in polynomial time.

Keywords: Coppersmith's method, lattices, RSA, RSA variants

MSC 2010: 94A60

Communicated by: Simon Blackburn

1 Introduction

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

At Crypto 1997, Takagi [18] proposed an RSA-type cryptosystems using n-adic expansion. One important variant of RSA is multi-power RSA [19], proposed by Takagi in 1998. In multi-power RSA, the RSA modulus N is of the form $N = p^r q$, where $r \geq 2$. Compared to 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 [21], which uses the modulus $N = p^2 q$.

At 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 $p^r q^{r+1}$, $p^{r-1}q^{r+1}$ or $p^{r-2}q^{r+2}$ is optimal under the assumption that the sum of exponents is fixed. For example, they claimed that 8192-bit RSA will be fifteen times faster than standard RSA if one takes $N = p^2 q^3$. In Crypto 1999, Boneh, Durfee and Howgrave-Graham [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 $r, l \geq 2$. Therefore, it is important to investigate the safety parameters of their algorithm.

Yao Lu: The University of Tokyo, Tokyo, Japan, e-mail: lywhhit@gmail.com
Liqiang Peng: Institute of Information Engineering, Chinese Academy of Sciences, Beijing, P. R. China, e-mail: pengliqiang@iie.ac.cn
***Corresponding author: Santanu Sarkar:** Indian Institute of Technology, Madras, India, e-mail: sarkar.santanu.bir@gmail.com

1.1 Related works

The security of this variant of RSA, like that of 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 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 this bound to $\frac{1}{2}$. Note that for the above results, the unknown bits are within one consecutive block. The case of $n(n \geq 2)$ 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. In 2013, Lu, Zhang and Lin [12] considered the case of n $(n \geq 2)$ blocks.

To speed up decryption, the small secret exponent d is often used in some cryptographic applications. However, it is well known that the RSA scheme is easily broken if the secret exponent d is too small (known as *small secret exponent attack*). In 1990, by utilizing the continued fraction method, Wiener [20] 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 to $N^{0.292}$. Recently, in [6], Herrmann and May gave an elementary proof for the Boneh–Durfee's bound, and in [9], Kunihiro, Shinohara and Izu also investigated this problem. However, $N^{0.292}$ is still the best bound at present.

For the case of multi-power RSA, there exists two variants. In the first variant, $ed \equiv 1 \bmod p^{r-1}(p-1)(q-1)$ while in the second variant, $ed \equiv 1 \bmod (p-1)(q-1)$. For the first variant, in 1999, Takagi [19] showed that when the secret exponent $d \leq N^{1/(2(r+1))}$, one can factorize N. Later in 2004, May [14] improved Takagi's bound to $N^{\max\{r(r+1)^{-2},(r-1)^2(r+1)^{-2}\}}$. Recently, Sarkar [17] used a lattice-based method to improve the previous bounds when $r \leq 5$. In [13], the authors further improved May's bound to $N^{r(r-1)(r+1)^{-2}}$, which is better than May's result when $r > 2$. For the second variant, in 2008, Itoh, Kunihiro and Kurosawa [8] showed that d can be recovered from if $d < N^{(2-\sqrt{2})/(r+1)}$.

1.2 Our contributions

In this paper,[1] we analyze the security of RSA-type schemes with moduli $N = p^r q^l$, where $r > l \geq 2$ and $\gcd(r, l) = 1$. Admittedly, RSA-type schemes with moduli $N = p^r q^l$ have very limited application. However, as rightly mentioned in [4] a significant fraction of cryptography is still based on RSA and so it is important to study these RSA-type moduli. Throughout the paper, we assume that $q < p < 2q$, which means $p \approx q$.

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 and d satisfy $ed \equiv 1 \bmod p^{r-1}q^{l-1}(p-1)(q-1)$. In the second variant, e and d satisfy $ed \equiv 1 \bmod (p-1)(q-1)$. For these two variants, we give the analysis respectively.

For the equation $ed \equiv 1 \bmod p^{r-1}q^{l-1}(p-1)(q-1)$, we solve a small solution d of the modular equation $ex - 1 \equiv 0 \bmod p^{r-1}q^{l-1}$. We introduce a new technique to select more helpful polynomials which are used to construct a 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 the result of [13].

[1] This is a thoroughly revised and extended version of the paper "Cryptanalysis of an RSA variant with moduli $N = p^r q^l$" that has been presented at WCC 2015, April 13–17, 2015, Paris, France. There is no formal proceedings for WCC 2015. Section 4.3 of this paper is the additional contribution that was not appeared in the workshop version.

For the equation $ed \equiv 1 \bmod (p-1)(q-1)$, we solve a small solution (k, p, q) of the modular equation $x(y-1)(z-1)+1 = 0 \bmod e$, 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 and show that when

$$d < N^{\frac{7-2\sqrt{7}}{3(r+l)}},$$

the small solution (k, p, q) can be found. Note that when $l = 1$, our result is exactly the general bound of [8].

Factoring RSA moduli $N = p^r q^l$ with partial known bits. In the conclusion of Boneh, Durfee and Howgrave-Graham's paper [2], the authors raised a question that whether one can generalize the factoring with partial known bits to 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 least significant bits (LSBs) of p in order to factor N in polynomial time. Independently, Coron, Faugère, Renault and Zeitoun [4] also studied this problem. We give a comparison with their method and give an improvement for certain parameters. Besides, we also extend to the case of the arbitrary number n $(n \geq 2)$ of unknown blocks.

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 (2.53 GHz, 1.9 GB RAM Windows 7). And the experimental results demonstrate that the performance of our algorithms is effective.

2 Preliminaries

Consider w linearly independent vectors $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_w \in \mathbb{Z}^n$. The set

$$\mathcal{L} = \left\{ \boldsymbol{b} : \boldsymbol{b} = \sum_{i=1}^{w} c_i \boldsymbol{b}_i,\ c_1, \ldots, c_w \in \mathbb{Z} \right\}$$

is called an w-dimensional lattice with basis $B = \{\boldsymbol{b}_1, \ldots, \boldsymbol{b}_w\}$. A lattice is of full rank when $w = n$ and in this paper we only use such lattices. The determinant of L is defined as $\det(\mathcal{L}) = \det(M)$, where the rows of M are the vectors from B. When $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_w \in \mathbb{Z}^n$, the lattice $\mathcal{L}$ is called an integer lattice.

In 1982, Lenstra, Lenstra and Lovász [10] proposed a polynomial time algorithm (known as LLL-Algorithm); let us first state the LLL-Algorithm.

Lemma 2.1 (LLL Algorithm). *Let $\mathcal{L}$ be a lattice of dimension w. Within polynomial time, LLL-Algorithm outputs a set of reduced basis vectors $\boldsymbol{v}_i$, $1 \leqslant i \leqslant w$, that satisfies*

$$\|\boldsymbol{v}_1\| \leqslant \|\boldsymbol{v}_2\| \leqslant \cdots \leqslant \|\boldsymbol{v}_i\| \leqslant 2^{\frac{w(w-1)}{4(w+1-i)}} \det(\mathcal{L})^{\frac{1}{w+1-i}}.$$

Let $g(x_1, \ldots, x_k) = \sum_{i_1,\ldots,i_k} a_{i_1,\ldots,i_k} x_1^{i_1} \cdots x_k^{i_k}$. We define the norm of g by the Euclidean norm of its coefficient vector:

$$\|g\|^2 = \sum_{i_1,\ldots,i_k} a_{i_1,\ldots,i_k}^2.$$

Also we need the following result due to Howgrave-Graham [7].

Lemma 2.2 (Howgrave-Graham). *Let $g(x_1, \ldots, x_k) \in \mathbb{Z}[x_1, \ldots, x_k]$ be an integer polynomial that consists of at most w monomials. Suppose that*

(i) $g(y_1, \ldots, y_k) = 0 \bmod e^m$ *for* $|y_1| \leqslant X_1, \ldots, |y_k| \leqslant X_k$,

(ii) $\|g(x_1X_1, \ldots, x_kX_k)\| < \frac{e^m}{\sqrt{w}}$.

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

Suppose we have w $(> k)$ polynomials $b_1, \ldots, b_w$ in the variables $x_1, \ldots, x_k$ such that

$$b_1(y_1, \ldots, y_k) = \cdots = b_w(y_1, \ldots, y_k) = 0 \bmod e^m$$

with

$$|y_1| \leq X_1, \ldots, |y_k| \leq X_k.$$

Now we construct a lattice $\mathcal{L}$ with the coefficient vectors of $b_1(x_1X_1, \dots, x_kX_k), \dots, b_w(x_1X_1, \dots, x_kX_k)$. After lattice reduction, we get k polynomials $v_1(x_1, \dots, x_k), \dots, v_k(x_1, \dots, x_k)$ such that

$$v_1(y_1, \dots, y_k) = \cdots = v_k(y_1, \dots, y_k) = 0 \bmod e^m$$

which correspond to first k vectors of the reduced basis. Also by the property of the LLL-Algorithm, we have

$$\|v_1(x_1X_1, \dots, x_kX_k)\| \le \cdots \le \|v_k(x_1X_1, \dots, x_kX_k)\| \le 2^{\frac{w(w-1)}{4(w+1-k)}} \det(\mathcal{L})^{\frac{1}{w+1-k}}.$$

Hence by Lemma 2.2, if

$$2^{\frac{w(w-1)}{4(w+1-k)}} \det(\mathcal{L})^{\frac{1}{w+1-k}} < \frac{e^m}{\sqrt{w}},$$

then we have $v_1(y_1, \dots, y_k) = \cdots = v_k(y_1, \dots, k_k) = 0$. Next we want to find $y_1, \dots, y_k$ from $v_1, \dots, v_k$.

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

Assumption 2.3. 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 2.4. *Let N be a sufficiently large composite integer (of unknown factorization) with a divisor p^r ($p \ge N^\beta$ and an integer $r \ge 1$). Let $f(x_1, \dots, x_n) \in \mathbb{Z}[x_1, \dots, x_n]$ be a linear polynomial in n variables. Under Assumption 2.3, we can find all the solutions $(x_1^0, \dots, x_n^0)$ of the equation $f(x_1, \dots, x_n) = 0 \bmod p$ with $|x_1^0| \le N^{\gamma_1}, \dots, |x_n^0| \le N^{\gamma_n}$ if*

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

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

3 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.

3.1 The first variant

At first, we study the first variant of encryption and decryption phases: e and d satisfy

$$ed \equiv 1 \bmod p^{r-1}q^{l-1}(p-1)(q-1).$$

Theorem 3.1. *For every $\epsilon > 0$, 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 let d be the private key exponent satisfying $ed \equiv 1 \bmod \phi(N)$. Suppose that*

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

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:

$$ed - 1 = kp^{r-1}q^{l-1}(p-1)(q-1) \quad \text{for some } k \in \mathbb{N}.$$

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 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 the 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 $i = 0, \dots, m$ 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 ($X = N^{\gamma}$) be the upper bound on the desired root d. We built a lattice $\mathcal{L}$ of dimension $d = m + 1$ 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 $\det(\mathcal{L}) = X^s N^{s_N}$. We can calculate s as

$$s = \sum_{i=0}^{m} i = \frac{m(m+1)}{2}.$$

The computation of s_N is somewhat complicated. At first, we have $t_1 < t_2$. Otherwise, since $r > l$, we have

$$\left\lceil \frac{(r-1)(t_1-i)}{r} \right\rceil \ge \left\lceil \frac{(l-1)(t_2-i)}{l} \right\rceil$$

for $i = 0, \dots, t_1$, 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

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

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

$$l(r-1)t_1 > r(l-1)t_2 \tag{3.1}$$

Notice that for $i = 0, 1, \dots, \Delta - 1$, we have

$$\left\lceil \frac{(r-1)(t_1-i)}{r} \right\rceil > \left\lceil \frac{(l-1)(t_2-i)}{l} \right\rceil;$$

however, for $i = \Delta, \Delta + 1, \dots, t_2$, we have

$$\left\lceil \frac{(r-1)(t_1-i)}{r} \right\rceil < \left\lceil \frac{(l-1)(t_2-i)}{l} \right\rceil.$$

Then we can calculate s_N as

$$\begin{aligned} s_N &= \sum_{i=0}^{\Delta-1} \left\lceil \frac{(r-1)(t_1-i)}{r} \right\rceil + \sum_{i=\Delta}^{t_2} \left\lceil \frac{(l-1)(t_2-i)}{l} \right\rceil \\ &= \frac{(r-1)(2t_1\Delta - \Delta^2)}{2r} + \frac{(l-1)(t_2-\Delta)^2}{2l} + \frac{\Delta(r-1)}{2r} + \frac{(t_2-\Delta)(l-1)}{2l} + \sum_{i=0}^{t_2} c_i. \end{aligned}$$

Here we rewrite

$$\left\lceil \frac{(r-1)(t_1-i)}{r} \right\rceil = \frac{(r-1)(t_1-i)}{r} + c_i$$

for $i = 0, \dots, \Delta - 1$, and

$$\left\lceil \frac{(l-1)(t_2-i)}{l} \right\rceil = \frac{(l-1)(t_2-i)}{l} + c_i$$

for $i = \Delta, \dots, t_2$, where $c_i \in [0, 1)$.

Furthermore, we rewrite

$$\Delta = \frac{l(r-1)t_1 - r(l-1)t_2}{r-l} + c',$$

where $c' \in [0, 1)$; we have that

$$s_N = \frac{(r-1)(l(r-1)t_1^2 - 2r(l-1)t_1t_2 + r(l-1)t_2^2)}{2r(r-l)} + \frac{c'(r-l) - c'^2(r-l) + l(r-1)t_1}{2rl} + \sum_{i=0}^{t_2} c_i.$$

To obtain a polynomial with short coefficients that contains all small roots over integer, we apply the LLL-Basis Reduction Algorithm to the lattice $\mathcal{L}$. Lemma 2.1 gives us an upper bound on the norm of the shortest vector in the LLL-reduced basis; if the bound is smaller than the bound given in Lemma 2.2, we can obtain the desired polynomial. We require the following condition:

$$2^{\frac{\omega-1}{4}} \sqrt{\omega} \det(\mathcal{L})^{\frac{1}{\omega}} < N^{\frac{(r-1)t_1+(l-1)t_2}{r+l}},$$

where $\omega = m + 1$. When plug in the value for $\det(\mathcal{L})$ and ω, we have that

$$2^{\frac{m(m+1)}{4}} (m+1)^{\frac{m+1}{2}} X^{\frac{m(m+1)}{2}} < N^{\frac{(m+1)((r-1)t_1+(l-1)t_2)}{r+l} - \frac{(r-1)(l(r-1)t_1^2-2r(l-1)t_1t_2+r(l-1)t_2^2)}{2r(r-l)} - \frac{c'(r-l)-c'^2(r-l)+l(r-1)t_1}{2rl} - \sum_{i=0}^{t_2} c_i}.$$

To obtain the asymptotic bound, we let m grow to infinity. Note that for sufficiently large N the powers of 2 and $m + 1$ are negligible. Thus we only consider the exponent of N. Then we obtain that

$$X < N^{\frac{2(r-1)\tau_1+2(l-1)\tau_2}{r+l} - \frac{(r-1)(l(r-1)\tau_1^2-2r(l-1)\tau_1\tau_2+r(l-1)\tau_2^2)}{r(r-l)}} \cdot N^{\frac{(r-1)(l(r-1)\tau_1^2-2r(l-1)\tau_1\tau_2+r(l-1)\tau_2^2)}{(m+1)r(r-l)} - \frac{c'(r-l)-c'^2(r-l)}{m(m+1)rl} - \frac{l(r-1)\tau_1}{(m+1)rl} - \frac{2\sum_{i=0}^{t_2} c_i}{m(m+1)}}, \tag{3.2}$$

where $t_1 = \tau_1 m$ and $t_2 = \tau_2 m$.

Now we have to decide the optimized values of τ_1 and τ_2. We consider the exponent of N as a function $h(\tau_1, \tau_2)$:

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

Using $h'_{\tau_1}(\tau_1, \tau_2) = 0$ and $h'_{\tau_2}(\tau_1, \tau_2) = 0$, we have

$$l(r-1)(r+l)\tau_1 - r(l-1)(r+l)\tau_2 + r(l-r) = 0,$$
$$(r-1)(r+l)\tau_1 - (r-1)(r+l)\tau_2 + r - l = 0.$$

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 into equation (3.1), we note that the condition is satisfied. Moreover, since $\sum_{i=0}^{t_2} c_i < t_2 + 1$, $\frac{1}{m^2} \le \frac{1}{m}$ and $c' - c'^2 < \frac{1}{4}$, inequality (3.2) can be reduced into,

$$X < N^{1-\frac{3r+l}{(r+l)^2}} \cdot N^{-\frac{(15l+1)r^4-(2l^2-10l)r^3-(l^3-6l^2+8l)r^2+(2l^3-12l^2+6l)r+l^4-4l^3+l^2}{4(m+1)(r-l)(r+l)^3}}.$$

We appropriate the terms $m + 1$ by m, and obtain

$$X < N^{1-\frac{3r+l}{(r+l)^2}-\frac{(15l+1)r^4-(2l^2-10l)r^3-(l^3-6l^2+8l)r^2+(2l^3-12l^2+6l)r+l^4-4l^3+l^2}{4m(r-l)(r+l)^3}}.$$

We can express how m depends on the error term ϵ:

$$m \ge \frac{(15l+1)r^4 - (2l^2-10l)r^3 - (l^3-6l^2+8l)r^2 + (2l^3-12l^2+6l)r + l^4 - 4l^3 + l^2}{4\epsilon(r-l)(r+l)^3}.$$

This concludes the proof of Theorem 3.1. □

Table 1 lists some theoretical and experimental results with 1000-bit N. In all experiments, we obtained an univariate integer equation with desired integer solution d. Thus we can obtain d.

		$\dim(\mathcal{L}) = 20$		$\dim(\mathcal{L}) = 40$	
(r, l)	theoretical	experimental	time (in seconds)	experimental	time (in seconds)
$(3, 2)$	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

Table 1. The first variant: experimental results for small d.

3.2 The second variant

In the following we study the second variant of encryption and decryption phases: e and d satisfy

$$ed \equiv 1 \bmod (p-1)(q-1).$$

Theorem 3.2. *For every $\epsilon > 0$, 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 let d be the private key exponent satisfying $ed \equiv 1 \bmod (p-1)(q-1)$. Suppose that*

$$d < N^{\frac{7-2\sqrt{7}}{3(r+l)}-\epsilon}.$$

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 follows:

$$k = \frac{ed-1}{(p-1)(q-1)} < \frac{2ed}{pq} < 2e^{1+\frac{\delta}{\alpha}-\frac{2}{\alpha(r+l)}}.$$

Usually, α is chosen as $\frac{2}{r+l}$. In this case, we have $p, q \simeq e^{\frac{1}{2}}$ and $k \simeq e^{\frac{r+l}{2}\delta}$. Let X $(X = e^{\frac{r+l}{2}\delta})$, Y $(Y = e^{\frac{1}{2}})$ and Z $(Z = e^{\frac{1}{2}})$ be the upper bounds of desired roots (p, q, k). In order to get desired solution, we define a list G of polynomials sharing the desired root modulo e^m,

$$g_{i,j,k,b}(x, y, z) = x^i y^j z^k f(x, y, z)^b e^{m-b}.$$

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 $m$
        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 $G$
    for $u = 0$ to $m$ do
        for $j = 0$ to $s$ do
                append $g_{0,j,0,u}$ to $G$
                for $i = l - 1$ in $1$ do
                    append $g_{0,r+j,i,u}$ to $G$
        for $k = 1$ to $t$ do
            for $j = r - 1$ to 0 do
                append $g_{0,j,k,u}$ to $G$
return $G$
```

where each occurrence of $y^r z^l$ is replaced by N since $N = p^r q^l$, m, s, t are non-negative integers.

Then we construct a lattice $\mathcal{L}_1$ which is spanned by the coefficient vectors of $g_{i,j,k,l}(xX, yY, zZ)$. By some calculations, the determinant of $\mathcal{L}_1$ is $\det(\mathcal{L}_1) = X^{S_x} Y^{S_y} Z^{S_z} e^{S_e}$, where

$$\begin{aligned}
S_x &= \Big(\frac{2r + 2l + 3\tau r + 3\sigma l}{6}\Big)m^3 + \Big(\frac{r + 2l + \tau r + \sigma l}{2}\Big)m^2 + \Big(\frac{r + 4l}{6}\Big)m, \\
S_y &= \Big(\frac{l + 3\sigma l + 3\sigma^2 l}{6}\Big)m^3 + \Big(\frac{r^2 + 2\tau r^2 + 2rl + 4\sigma rl - 3r + 2l - 2\tau r - 4\sigma r + 2\sigma^2 l + 4\sigma l}{4}\Big)m^2 \\
&\quad + \Big(\frac{3r^2 + 6\tau r^2 + 18rl + 12\sigma rl - 21r + 4l - 6\tau r - 12\sigma r + 6\sigma l}{12}\Big)m - (r - rl), \\
S_z &= \Big(\frac{r + 3\tau r + 3\tau^2 r}{6}\Big)m^3 + \Big(\frac{l^2 + 2\sigma l^2 + 2r - l + 2\tau^2 r + 4\tau r - 2\sigma l}{4}\Big)m^2 \\
&\quad + \Big(\frac{9l^2 + 6\sigma l^2 + 4r - 9l + 6\tau r - 6\sigma l}{12}\Big)m + \Big(\frac{l^2 - l}{2}\Big), \\
S_e &= \Big(\frac{2r + 2l + 3\tau r + 3\sigma l}{6}\Big)m^3 + \Big(\frac{r + 2l + \tau r + \sigma l}{2}\Big)m^2 + \Big(\frac{r + 4l}{6}\Big)m,
\end{aligned}$$

with $s = \sigma m$ and $t = \tau m$. On the other hand,

$$\dim(\mathcal{L}_1) = \Big(\frac{r + l + 2\tau r + 2\sigma l}{2}\Big)m^2 + \Big(\frac{r + 3l + 2\tau r + 2\sigma l}{2}\Big)m + l.$$

Since there are three unknown variables, based on Lemma 2.1 and Lemma 2.2, one can obtain three polynomial equations which share the roots (k, p, q) over integers when

$$2^{\frac{\dim(\mathcal{L}_1)(\dim(\mathcal{L}_1)-1)}{4(\dim(\mathcal{L}_1)-2)}} (X^{S_x} Y^{S_y} Z^{S_z} e^{S_e})^{\frac{1}{\dim(\mathcal{L}_1)-2}} < \frac{e^m}{\sqrt{\dim(\mathcal{L}_1)}}.$$

Putting the upper bounds and the value of $\dim(\mathcal{L}_1)$ into the above inequality and neglecting the terms that do not depend on N, we obtain that

$$e^{\frac{(r+l)\delta}{2}} < e^{\frac{m(\dim(\mathcal{L}_1)-2)-\frac{1}{2}(S_y+S_z)-S_e}{S_x}},$$

or equivalently,

$$\frac{(r + l)\delta}{2} < \frac{m(\dim(\mathcal{L}_1) - 2) - \frac{1}{2}(S_y + S_z) - S_e}{S_x}.$$

Setting $\sigma = \tau$, moreover, since $m < m^2$, $0 \le \tau \le 1$ and $l < r$, the left side of the above inequality can be bounded by

$$\begin{aligned}
&\frac{\frac{(r+l)(1+3\tau-3\tau^2)}{12}m^3 - \frac{(2\tau+1)r^2+(2\tau^2-6\tau+2l+4\tau l-1)r-3l-2\tau l+2\tau l^2+l^2+2\tau^2 l}{8}m^2}{\frac{(r+l)(2+3\tau)}{6}m^3 + \frac{r+2l+\tau r+\tau l}{2}m^2 + \frac{r+4l}{6}m} \\
&\qquad - \frac{\frac{(6\tau+3)r^2-(13+12\tau-18l-12\tau l)r+9l^2+6\tau l^2-17l+48}{24}m + \frac{(2r+l)(l-1)}{4}}{\frac{(r+l)(2+3\tau)}{6}m^3 + \frac{r+2l+\tau r+\tau l}{2}m^2 + \frac{r+4l}{6}m} \\
&< \frac{\frac{(r+l)(1+3\tau-3\tau^2)}{12}m^3 - \frac{(2\tau+1)r^2+(2\tau^2-6\tau+2l+4\tau l-1)r-3l-2\tau l+2\tau l^2+l^2+2\tau^2 l}{8}m^2}{\frac{(r+l)(2+3\tau)}{6}m^3} \\
&= \frac{1 + 3\tau - 3\tau^2}{2(2 + 3\tau)} - \frac{3((2\tau + 1)r^2 + (2\tau^2 - 6\tau + 2l + 4\tau l - 1)r - 3l - 2\tau l + 2\tau l^2 + l^2 + 2\tau^2 l)}{4(r + l)(2 + 3\tau)m} \\
&< \frac{1 + 3\tau - 3\tau^2}{2(2 + 3\tau)} - \frac{3(r^2 + 2rl + l^2 - 7r - 5l)}{20(r + l)m}.
\end{aligned}$$

			$\dim(\mathcal{L}_1) = 81$		$\dim(\mathcal{L}_1) = 148$	
(r, l)	**$\log_2 N$**	**theoretical d**	**experimental d**	**time of L^3 (in seconds)**	**experimental d**	**time of L^3 (in seconds)**
$(3, 2)$	2000	200 bits	29 bits	35.350	71 bits	2573.002
$(3, 2)$	3000	300 bits	47 bits	103.600	110 bits	5197.392

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

Putting an optimized value for τ, which is $\tau = \frac{\sqrt{7}-2}{3}$, into the above inequality, we obtain

$$\frac{7-2\sqrt{7}}{6} - \frac{3(r^2+2rl+l^2-7r-5l)}{20(r+l)m}.$$

Then we have

$$\delta < \frac{7-2\sqrt{7}}{3(r+l)} - \frac{3(r^2+2rl+l^2-7r-5l)}{10(r+l)^2 m}.$$

The relation between the error term ϵ and m can be expressed as

$$m \geq \frac{3(r^2+2rl+l^2-7r-5l)}{10(r+l)^2 \epsilon}.$$

This concludes the proof of Theorem 3.2. □

Table 2 lists some theoretical and experimental results. In all experiments, we obtained several integer equations which share desired roots and successfully obtained the roots by using Gröbner basis technique.

4 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: $\tilde{p} = p \bmod 2^k$. 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. Below we present two methods to solve this problem.

4.1 The attack modulo p

The above problem can be reduced to solve modular univariate polynomial equation

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

We can apply Theorem 2.4 with $n = 1$, $\beta = \frac{1}{r+l}$. Therefore, we can find all root y if

$$|y| \leq N^{\frac{r}{(r+l)^2}}.$$

When $l = 1$, the bound

$$N^{\frac{r}{(r+l)^2}} = N^{\frac{r}{(r+1)^2}} = p^{\frac{r}{r+1}}.$$

This bound is exactly the same as in [2]. As $N^{\frac{r}{(r+l)^2}} = p^{\frac{r}{r+l}}$, the 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 $2^{(\frac{l}{r+l}\log_2 p)\cdot P(\log N)}$, where P is a polynomial. This method is very suitable for the case of $r \gg l$.

4.2 The attack modulo pq

Let us start with the following lemma.

Lemma 4.1. *For a given integer k, consider the modular function $f(x) = x^w \bmod 2^k$ whose domain is the set $\{1, 3, \dots, 2^k - 1\}$. When w is odd and $x_0^w \equiv a \bmod 2^k$, then one can get x_0.*

Proof. Since the domain of $f(x)$ is $\{1, 3, \dots, 2^k - 1\}$, the range of $f(x)$ is also $\{1, 3, \dots, 2^k - 1\}$. On the other hand, assume that $x_1, x_2 \in \{1, 3, \dots, 2^k - 1\}$ and $x_1^w \equiv x_2^w \pmod{2^k}$. Then we can obtain that $2^k \mid x_1^w - x_2^w$, namely $2^k \mid (x_1 - x_2)(x_1^{w-1} + x_1^{w-2}x_2 + \cdots + x_2^{w-1})$. Since x_1, x_2, w are odd integers, $x_1^{w-1} + x_1^{w-2}x_2 + \cdots + x_2^{w-1}$ is odd and $x_1 - x_2 \in \{-2^k + 2, 2^k - 2\}$. Then one can get that $x_1 = x_2$, namely $f(x)$ is bijective.

Above all, the solution x_0 is unique and it can be obtained as

$$x_0 \equiv a^{w^{-1} \bmod 2^{k-1}} \bmod 2^k.$$

This concludes the proof of Lemma 4.1. □

We rewrite N by $N = (pq)^l p^{r-l}$. Notice that at least one of r and l must be odd; we may assume without loss of generality that l is odd. Suppose that we have k LSBs of p and let us denote it as $\tilde{p}$. So $\tilde{p} = p \bmod 2^k$. Thus $q^l = N(\tilde{p}^r)^{-1} \bmod N$. Then by Lemma 4.1 we can calculate the number of k LSBs of q: $\tilde{q} = q \bmod 2^k$. Using $\tilde{p}$ and $\tilde{q}$, we can get the number of k LSBs of pq: $c = \tilde{p}\tilde{q} \bmod 2^k$. Now we reduce the above problem to solve a modular univariate polynomial equation

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

Now apply Theorem 2.4 with $n = 1$, $\beta = \frac{2}{r+l}$. Then we can find 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 $N^{\frac{4l}{(r+l)^2}} = (pq)^{\frac{2l}{r+l}}$, the attacker has to guess

$$\left(1 - \frac{2l}{r+l}\right)\log_2 pq = \frac{r-l}{r+l}\log_2 pq = \frac{2(r-l)}{r+l}\log_2 p$$

LSBs of p. Thus the total complexity to the factor $N = p^r q^l$ is $2^{(\frac{2(r-l)}{r+l}\log_2 p)\cdot P(\log N)}$, where P is a polynomial. This method is very suitable for the case of $r \simeq l$.

Comparison between the two methods. In the first method, the attacker has to guess $\frac{l}{r+l}\log p$ bits whereas in the second method it is required to guess $\frac{2(r-l)}{r+l}\log p$ bits. Since $\frac{l}{r+l} < \frac{2(r-l)}{r+l}$ if $2r > 3l$, our first attack (modulo p) is superior to our second attack (modulo pq) in the case $2r > 3l$.

We present our bounds $\min(\frac{l}{r+l}, \frac{2(r-l)}{r+l})$ in Figure 1. In Table 3, we give some experimental results of the above two methods.

			attack modulo p				attack modulo pq			
(r, l)	$\log_2 N$	$\log_2 p$	theo.	expt.	dim.	time (sec.)	theo.	expt.	dim.	time (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

Table 3. Factoring N with partial known bits of p.

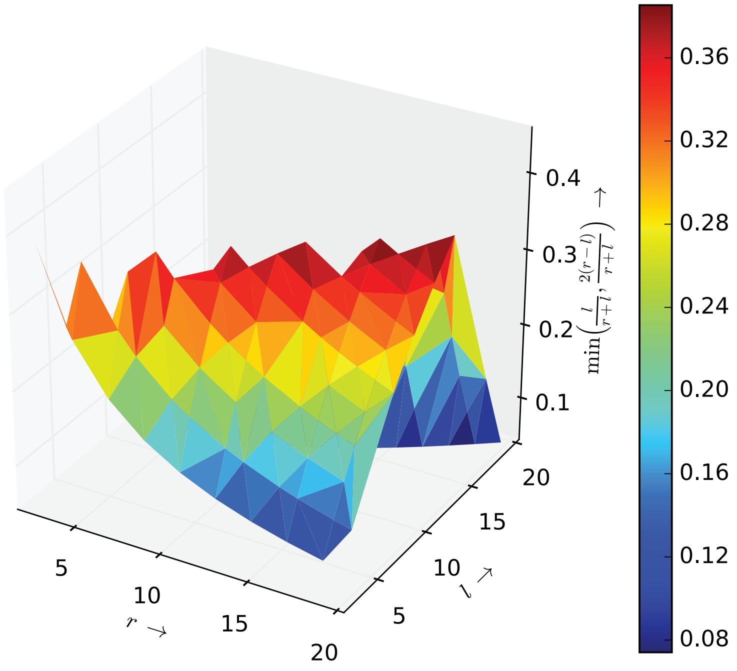

Figure 1. Our bounds for different r, l.

4.3 Comparison with the work of Coron, Faugère, Renault and Zeitoun

Independently, Coron, Faugère, Renault and Zeitoun [4] also studied this problem; they showed that $N = p^r q^l$ can be factored in polynomial time when r or l is at least $(\log p)^3$. In the following remark, we will briefly discuss their idea. Moreover, based on an observation of the short vectors in a two-dimensional lattice which has been introduced in [4], we further improved Coron–Faugère–Renault–Zeitoun's bound for the moduli with form of $N = p^r q^l$, where $r = 2k + 1$, $l = k + 1$ and $k \in \mathbb{Z}^+$.

In [4, p. 5], for the modulus $N = p^r q^l$, r and l are first expressed as $r = u \cdot \alpha + a$ and $l = u \cdot \beta + b$, where the integers u, α, β, a, b should satisfy certain conditions. To find such integers, it is required to apply the LLL-Algorithm on the two-dimensional lattice which is spanned by the row vectors of the following matrix:

$$\begin{pmatrix} \lfloor r^{\frac{1}{3}} \rfloor & -l \\ 0 & r \end{pmatrix}.$$

After lattice reduction, suppose that the short vector is $v = (\lfloor r^{\frac{1}{3}} \rfloor \cdot \alpha, -l \cdot \alpha + r \cdot \beta)$ for some $\beta \in \mathbb{Z}$. Now if $\beta = 0$ or $\lfloor \frac{r}{\alpha} \rfloor \leq \frac{l}{\beta}$, u is taken as $\lfloor \frac{r}{\alpha} \rfloor$. On the other hand if $\beta \neq 0$ and $\lfloor \frac{r}{\alpha} \rfloor > \frac{l}{\beta}$, u is set as $\lceil \frac{r}{\alpha} \rceil$. Finally, a is taken as $r - u\alpha$ and b is taken as $l - u\beta$. It has been proved in [4, Lemma 1] that either both $a, b \geq 0$ or $a, b \leq 0$.

- First suppose that both $a, b \geq 0$. Now N can be expressed as $N = p^r q^l = p^{u\alpha + a} q^{u\beta + b} = P^u Q$, where $P = p^\alpha q^\beta$ and $Q = p^a q^b$. It has been proved in [4, p. 18] that to factor $N = P^r Q$ in polynomial time, the attacker has to guess $\frac{c}{u+c}$ many bits of P to find P, where $Q < P^c$. Thus if $a, b \geq 0$, it is required to guess $\frac{c}{u+c} \log P$ many bits of P. Here we can take $c = \frac{a+b}{\alpha+\beta}$ as $P \approx p^{\alpha+\beta}$ and $Q \approx p^{a+b}$. Thus in this case the attacker has to guess $\frac{a+b}{(\alpha+\beta)u+a+b} \cdot (\alpha + \beta) \log p$ many bits.
- Next suppose that $a, b \leq 0$. Now express $N = \frac{P^u}{Q}$, where $P = p^\alpha q^\beta$ and $Q = p^{-a} q^{-b}$. In this case it has been proved in [4, p. 8] that the attacker has to search over $[0, 2Q^{\frac{1}{u}}]$. So the required guess in this case will be approximately $\frac{-(a+b)}{u} \log p$ bits.

Although in most of the cases the bounds of [4] may found the optimal expressions of $N = p^r q^l$, for some values of r, l they could not give the best bound. For example, based on Coron–Faugère–Renault–Zeitoun's method, the modulus N of the form $p^{2k+1} q^{k+1}$, $k \geq 2$, should be expresses as $N = P^k Q$, where $P = p^2 q$ and $Q = pq$; however, when we express N in the form $\frac{P^{k+1}}{Q}$, where $P = p^2 q$ and $Q = p$, the less number of known bits is required to factor N.

More specifically, for the modulus of the form $N = p^{2k+1}q^{k+1}$, it is required in [4] to apply the LLL-Algorithm on the lattice $\mathcal{L}$ which is spanned by the row vectors of the following matrix:

$$\begin{pmatrix} \lfloor (2k+1)^{\frac{1}{3}} \rfloor & -(k+1) \\ 0 & 2k+1 \end{pmatrix}.$$

It is easily checked that $\lambda_1(\mathcal{L}) = (2\lfloor (2k+1)^{\frac{1}{3}} \rfloor, -1)$ and $\lambda_2(\mathcal{L}) = (\lfloor (2k+1)^{\frac{1}{3}} \rfloor, k)$, where the minima $\lambda_i(\mathcal{L})$ denotes the i-th minimum of lattice $\mathcal{L}$.

According to $\lambda_1(\mathcal{L})$, we have that $\alpha = 2, \beta = 1$. Furthermore, since $\lfloor \frac{2k+1}{\alpha} \rfloor = k \le \frac{k+1}{\beta}$, based on Coron–Faugère–Renault–Zeitoun's method [4], u is taken as $\lfloor \frac{2k+1}{\alpha} \rfloor = k$. Furthermore, the modulus N should be expressed as $P^k Q$, where $P = p^2 q$ and $Q = pq$. Moreover, for the second shortest vector $\lambda_2(\mathcal{L})$, the modulus N will be expressed as $\frac{P^{2k+1}}{Q}$, where $P = pq$ and $Q = q^k$.

Then for the first expression of N, it is required to guess $\frac{6}{3k+2} \log p$ bits. And for the second expression, the number of required known bits is $\frac{k}{2k+1} \log p$ bits of p.

Based on our two methods of Section 3.1 and Section 3.2, the number of known LSBs of p which is required to factor $N = p^{2k+1}q^k$ is

$$\min\left(\frac{k+1}{2k+1+k+1}, \frac{2(2k+1-(k+1))}{2k+1+k+1}\right) = \frac{k+1}{3k+2}.$$

However, when we express N as $\frac{P^{k+1}}{Q}$, where $P = p^2 q$ and $Q = p$, in this case the attacker has to search over $[0, 2p^{\frac{1}{k+1}}]$. Namely, the required guess in this case will be approximately $\frac{1}{k+1} \log p$ bits.

Actually, there does not exist any vector in the two-dimensional $\mathcal{L}$ which will express $N = p^{2k+1}q^k$ as $\frac{P^{k+1}}{Q}$, where $P = p^2 q$ and $Q = p$. Since according to Coron–Faugère–Renault–Zeitoun's method [4], if one wants to express $N = p^{2k+1}q^k$ as $\frac{P^{k+1}}{Q}$, where $P = p^2 q$ and $Q = p$, one should have that $\alpha = 2$, $\beta = 1$ and $u = k+1$. However, for $\alpha = 2$ and $\beta = 1$, we have $\lfloor \frac{2k+1}{\alpha} \rfloor \le \frac{k+1}{\beta}$; then u should be taken as $\lfloor \frac{2k+1}{\alpha} \rfloor = k$, which contradicts $u = k+1$.

Thus in general, the Coron–Faugère–Renault–Zeitoun approach cannot give optimal u, α, β. For $r \le 20$ and $2 \le l < r$, we search exhaustively to find optimal u, α, β. Optimal bounds are presented in Figure 2.

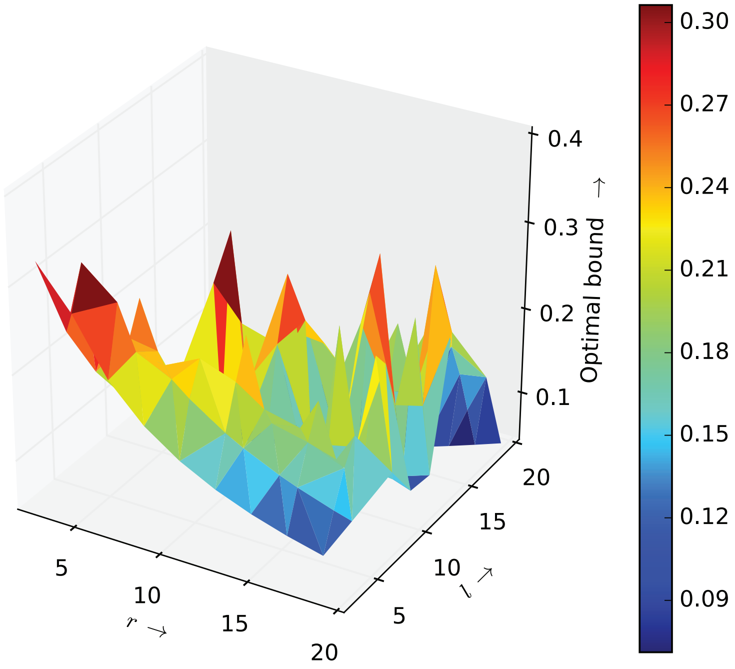

Figure 2. Optimal bound for some values of r, l.

4.4 Extend to more unknown blocks

We also consider the case of the number of n ($n \geq 2$) unknown blocks.

Theorem 4.2. *Let $N = p^r q^l$, where p and q are of equal length. Suppose that a $\frac{l}{r}\ln(\frac{r+l}{l})$-fraction of the bits is known for n blocks in p (n is large). Then, under Assumption 2.3, 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 solve the following multivariate linear polynomial equation:

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

where $a_k = 2^l$ if the k-th unknown blocks start on the l-th bit position. Moreover, if n goes to infinity, from Theorem 2.4, we have

$$\lim_{n\to\infty}\left(\frac{1}{r}\left(1 - (1 - r\beta)^{\frac{n+1}{n}} - (n+1)(1 - r\beta)(1 - \sqrt[n]{1 - r\beta})\right)\right) = \beta + \frac{(1 - r\beta)\ln(1 - r\beta)}{r}.$$

It shows that if n is very large, we can recover p regardless of n. Conversely, once a $(1 - \frac{1}{r\beta})\ln(1 - r\beta)$ portion of the bits from p together with their positions are given, we are able to recover the missing bits. Suppose that $|p| = |q|$, i.e. $\beta = \frac{1}{r+l}$. Then we need a

$$\left(1 - \frac{1}{r\beta}\right)\ln(1 - r\beta) = \left(1 - \frac{r+l}{r}\right)\ln\left(1 - \frac{r}{r+l}\right) = -\frac{l}{r}\ln\left(\frac{l}{r+l}\right) = \frac{l}{r}\ln\left(\frac{r+l}{l}\right)$$

portion of known bits from p. □

Note that for $l = 1$, this is exactly the result of [12].

5 Conclusion

In this paper, we have considered the RSA variant with moduli of the form $N = p^r q^l$, where $r > l \geq 2$, and we have given some cryptanalytic results for this kind of RSA variant. For the small secret exponent attacks, we have two cases of encryption and decryption exponents: $ed \equiv 1 \bmod p^{r-1}q^{l-1}(p-1)(q-1)$ and $ed \equiv 1 \bmod (p-1)(q-1)$. For these two cases, we have given the lattice-based attacks and obtained the upper bounds of decryption exponents d such that d can be recovered in polynomial time. Then we have presented the partial known bits attacks and successfully factored $N = p^r q^l$ when least significant bits of one prime are known.

Funding: Yao Lu is supported by Project CREST, JST and Liqiang Peng is supported by the National Key Basic Research Program of China (Grant 2013CB834203) and the National Natural Science Foundation of China (Grant 61472417).

References

[1] D. Boneh and G. Durfee, Cryptanalysis of RSA with private key d less than $N^{0.292}$, *IEEE Trans. Inform. Theory* **46** (2000), no. 4, 1339–1349.

[2] D. Boneh, G. Durfee and N. Howgrave-Graham, Factoring $N = p^r q$ for large r, in: *Advances in Cryptology – CRYPTO 1999*, Lecture Notes in Comput. Sci. 1666, Springer, Berlin (1999), 787–787.

[3] D. Coppersmith, Small solutions to polynomial equations, and low exponent RSA vulnerabilities, *J. Cryptologyy* **10** (1997), no. 4, 233–260.

[4] J. S. Coron, J. C. Faugère, G. Renault and R. Zeitoun, Factoring $N = p^r q^s$ for large r and s, in: *Topics in Cryptology – CT-RSA 2016*, Lecture Notes in Comput. Sci. 9610, Springer, Berlin (2016), 448–464; https://eprint.iacr.org/2015/071.

[5] M. Herrmann and A. May, Solving linear equations modulo divisors: On factoring given any bits, in: *Advances in Cryptology – ASIACRYPT 2008*, Lecture Notes in Comput. Sci. 5350, Springer, Berlin (2008), 406–424.

[6] M. Herrmann and A. May, Maximizing small root bounds by linearization and applications to small secret exponent RSA, in: *Public Key Cryptography – PKC 2010*, Lecture Notes in Comput. Sci. 6056, Springer, Berlin (2010), 53–69.

[7] N. Howgrave-Graham, Finding small roots of univariate modular equations revisited, in: *Crytography and Coding – IMACC 1997*, Lecture Notes in Comput. Sci. 1355, Springer, Berlin (1997), 131–142.

[8] K. Itoh, N. Kunihiro and K. Kurosawa, Small secret key attack on a variant of RSA (due to Takagi), in: *Topics in Cryptology – CT-RSA 2008*, Lecture Notes in Comput. Sci. 4964, Springer, Berlin (2008), 387–406.

[9] N. Kunihiro, N. Shinohara and T. Izu, A unified framework for small secret exponent attack on RSA, in: *Selected Areas in Cryptography – SAC 2011*, Lecture Notes in Comput. Sci. 7118, Springer, Berlin (2012), 260–277.

[10] A. K. Lenstra, H. W. Lenstra and L. Lovász, Factoring polynomials with rational coefficients, *Math. Ann.* **261** (1982), no. 4, 515–534.

[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*, Lecture Notes in Comput. Sci. 1977, Springer, Berlin (2000), 283–294.

[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 – ACISP 2013*, Lecture Notes in Comput. Sci. 7959, Springer, Berlin (2013), 57–71.

[13] Y. Lu, R. Zhang, L. Peng and D. Lin, Solving linear equations modulo unknown divisors: revisited, in: *Advances in Cryptology – ASIACRYPT 2015*, Lecture Notes in Comput. Sci. 9452, Springer, Berlin (2015), 189–213; https://eprint.iacr.org/2014/343.

[14] A. May, Secret exponent attacks on RSA-type schemes with moduli $N = p^r q$, in: *Public Key Cryptography – PKC 2004*, Lecture Notes in Comput. Sci. 2947, Springer, Berlin (2004), 218–230.

[15] T. Okamoto and S. Uchiyama, A new public-key cryptosystem as secure as factoring, in: *Advances in Cryptology – EUROCRYPT 1998*, Lecture Notes in Comput. Sci. 1403, Springer, Berlin (1998), 308–318.

[16] R. Rivest and A. Shamir, Efficient factoring based on partial information, in: *Advances in Cryptology – EUROCRYPT 1985*, Lecture Notes in Comput. Sci. 219, Springer, Berlin (1986), 31–34.

[17] S. Sarkar, Small secret exponent attack on RSA variant with modulus $N = p^r q$, *Des. Codes Cryptogr.* **73** (2014), no. 2, 383–392.

[18] T. Takagi, Fast RSA-type cryptosystems using n-adic expansion, in: *Advances in Cryptology – CRYPTO 1997*, Lecture Notes in Comput. Sci. 1294, Springer, Berlin (1997), 372–384.

[19] T. Takagi, Fast RSA-type cryptosystem modulo $p^k q$, in: *Advances in Cryptology – CRYPTO 1998*, Lecture Notes in Comput. Sci. 1462, Springer, Berlin (1998), 318–326.

[20] M. J. Wiener, Cryptanalysis of short RSA secret exponents, *IEEE Trans. Inform. Theory* **36** (1990), no. 3, 553–558.

[21] The EPOC and the ESIGN Algorithms, IEEE P1363: Protocols from other families of Public-Key algorithms, 1998, http://grouper.ieee.org/groups/1363/StudyGroup/NewFam.html.