SCIENCE CHINA
Information Sciences

• **RESEARCH PAPER** •
Special Focus on Security of Cyberspace

November 2015, Vol. 58 110103:1–110103:8
doi: 10.1007/s11432-015-5409-4

Cryptanalysis of Prime Power RSA with two private exponents

ZHENG MengCe[1,2] & HU HongGang[1,2*]

[1] *Key Laboratory of Electromagnetic Space Information, Chinese Academy of Sciences, Hefei* 230027*, China;*
[2] *School of Information Science and Technology, University of Science and Technology of China, Hefei* 230027*, China*

Received May 30, 2015; accepted September 6, 2015; published online September 28, 2015

Abstract In this paper, we consider a variant of RSA schemes called Prime Power RSA with modulus $N = p^r q$ for $r \geqslant 2$, where p, q are of the same bit-size. May showed that when private exponent $d < N^{\frac{r}{(r+1)^2}}$ or $d < N^{\left(\frac{r-1}{r+1}\right)^2}$, N can be factored in polynomial time in PKC 2004. Later in 2014, Sarkar improved the bound for $r \leqslant 5$. We propose a new cryptanalytic method to attack this RSA variant when given two pairs of public and private exponents, namely (e_1, d_1) and (e_2, d_2) with the same modulus N. Suppose that we know $d_1 < N^{\delta_1}$ and $d_2 < N^{\delta_2}$. Our results show that when $\delta_1 \delta_2 < \left(\frac{r-1}{r+1}\right)^3$, Prime Power RSA is insecure.

Keywords cryptanalysis, Prime Power RSA, two private exponents, LLL algorithm, Coppersmith's techniques

Citation Zheng M C, Hu H G. Cryptanalysis of Prime Power RSA with two private exponents. Sci China Inf Sci, 2015, 58: 110103(8), doi: 10.1007/s11432-015-5409-4

1 Introduction

The famous RSA cryptosystem [1] plays an important role in the area of information security due to its popularity. Since Coppersmith introduced a new lattice-based method of finding small roots of modular and integer equations [2,3], many researchers have studied its vulnerability in various cases such as small public exponent [2–4], small private exponent [5–8], and partial key exposure [9–13]. Opposed to cryptanalysis of the original RSA, several variants of RSA schemes have been proposed for efficient encryption and decryption, or for higher security.

In this paper, we concentrate on the variant with modulus $N = p^r q$ for $r \geqslant 2$, where p and q are two primes of the same bit-size. We call this variant Prime Power RSA. It was introduced in [14] by Takagi in Crypto 1998. He showed that the decryption process of Prime Power RSA is much faster than the variant using Chinese remainder theorem.

There are two different types of Prime Power RSA according to the definition of e, d, where e denotes the public exponent and d denotes the private exponent.

Type i $ed \equiv 1 \bmod \phi(N)$, where $\phi(N) = p^{r-1}(p-1)(q-1)$.

Type ii $ed \equiv 1 \bmod \tilde{\phi}(N)$, where $\tilde{\phi}(N) = (p-1)(q-1)$.

[*] Corresponding author (email: hghu2005@ustc.edu.cn)

Boneh et al. [15] studied the factorization of N for leaking some bits of p, so that Prime Power RSA is insecure. When one knows $\frac{1}{r+1}$ fraction of the most significant bits (MSBs) of p, it is sufficient to factor N in polynomial time. Also, small private exponent attack has been proposed. For **Type i**, Takagi [14] showed that Wiener's continued fraction attack is effective for small private exponent $d < N^{\frac{1}{2(r+1)}}$. May [16] improved the bound to $d < N^{\frac{r}{(r+1)^2}}$ or $d < N^{\left(\frac{r-1}{r+1}\right)^2}$ by lattice-based method in PKC 2004. Later in 2014, Sarkar [17] improved the bound for $r \leqslant 5$. For **Type ii**, Itoh [18] et al. showed that Prime Power RSA is insecure for $d < N^{\frac{2-\sqrt{2}}{r+1}}$.

However, there is one limitation that previous work considers one pair of public and private exponents of Prime Power RSA. Hence, we are interested in a new situation, where many instances are given with a common modulus N. To be specific, we investigate the security of Prime Power RSA with two private exponents for **Type i**.

From the main equation of **Type i**, namely $ed \equiv 1 \bmod \phi(N)$, where $\phi(N) = p^{r-1}(p-1)(q-1)$, we have

$$\begin{cases} e_1d_1 = k_1p^{r-1}(p-1)(q-1)+1, \\ e_2d_2 = k_2p^{r-1}(p-1)(q-1)+1, \end{cases}$$

for some positive integers k_1, k_2. Let e_1', e_2' be the inverse of e_1, e_2 modulo N respectively. Then we have

$$\begin{cases} e_1e_1' = l_1N+1, \\ e_2e_2' = l_2N+1, \end{cases}$$

for some positive integers l_1, l_2. If e_1' or e_2' does not exist, we can obtain the factorization of N by computing the greatest common divisors $\mathrm{GCD}(e_1, N)$ and $\mathrm{GCD}(e_2, N)$. They must be non-trivial divisors of N, namely p^t, $p^{t'}q$ $(1 \leqslant t, t' \leqslant r)$ or q. Combining them together leads to

$$\begin{cases} d_1 - e_1' = (e_1'k_1(p-1)(q-1) - l_1d_1pq)p^{r-1}, \\ d_2 - e_2' = (e_2'k_2(p-1)(q-1) - l_2d_2pq)p^{r-1}. \end{cases}$$

It finally reduces to

$$\begin{cases} d_1 - e_1' = 0 \bmod p^{r-1}, \\ d_2 - e_2' = 0 \bmod p^{r-1}. \end{cases} \tag{1}$$

We note that our method is also available for partial key exposure attacks. The attacks can be divided into two cases. When given MSBs d' with $d = d' + \tilde{d}$, we have $e\tilde{d} + ed' - 1 \equiv 0 \bmod \phi(N)$. By using the above technique, we obtain $\tilde{d} + e'(ed'-1) = 0 \bmod p^{r-1}$ for e' denoting the inverse of e modulo N. When given LSBs d' with $d = \tilde{d}M + d'$, we have $eM\tilde{d} + ed' - 1 \equiv 0 \bmod \phi(N)$. There also exists $\tilde{d} + e'(ed'-1) = 0 \bmod p^{r-1}$ for e' denoting the inverse of eM modulo N. Since we have two private exponents for the same modulus N, gathering them together gives us

$$\begin{cases} \tilde{d}_1 + a_1 = 0 \bmod p^{r-1}, \\ \tilde{d}_2 + a_2 = 0 \bmod p^{r-1}, \end{cases} \tag{2}$$

for $a_i = e_i'(e_id_i' - 1)$, where d_i' denotes known MSBs (or LSBs) and e_i' denotes the inverse of e_i (or e_iM_i) modulo N for $i = 1, 2$.

We apply Takayasu and Kunihiro's better lattice constructions [19] to solve bivariate linear equations modulo an unknown divisor. We now estimate the unknown divisor in our attacks. It is easy to see that $p, q \approx N^{\frac{1}{r+1}}$. So, we have $p^{r-1} \approx N^{\frac{r-1}{r+1}}$. To achieve the theoretical results, our method relies on the following heuristic assumption.

Assumption 1. Algebraically independent polynomials can be obtained by our lattice-based method, and the common root can be efficiently solved by the Gröbner basis computations.

Our main results are stated in the following theorems that will be proved in Section 3 afterwards. We want to point out that the theoretical results stated below are asymptotic since we require the dimension of the corresponding lattice to be preferably large.

Theorem 1. Let $N = p^r q$ for $r \geqslant 2$ be a known RSA modulus, where p and q are two primes of the same bit-size. Let e_1, d_1, e_2, d_2 satisfy $e_1 d_1 \equiv 1 \bmod \phi(N)$ and $e_2 d_2 \equiv 1 \bmod \phi(N)$, such that $e_1, e_2 \approx N$, $d_1 < N^{\delta_1}$, and $d_2 < N^{\delta_2}$. Then under Assumption 1, N can be factored in polynomial time if

$$\delta_1 \delta_2 < \left(\frac{r-1}{r+1}\right)^3 .$$

Theorem 2. Let $N = p^r q$ for $r \geqslant 2$ be a known RSA modulus, where p and q are two primes of the same bit-size. Let e_1, d_1, e_2, d_2 satisfy $e_1 d_1 \equiv 1 \bmod \phi(N)$ and $e_2 d_2 \equiv 1 \bmod \phi(N)$, such that $e_1, e_2 \approx N$ and some MSBs or LSBs of d_1, d_2 are given. Suppose that for $i = 1, 2$, given MSBs d_i' with $d_i = d_i' + \tilde{d}_i$ and $\tilde{d}_i < N^{\gamma_i}$, or given LSBs d_i' with $d_i = \tilde{d}_i M + d_i'$ and $\tilde{d}_i < N^{\gamma_i}$. Then under Assumption 1, N can be factored in polynomial time if

$$\gamma_1 \gamma_2 < \left(\frac{r-1}{r+1}\right)^3 .$$

The rest of the paper is organized as follows: We review basic results of lattice reduction theory in Section 2. In Section 3, we describe the method of solving simultaneous equations (1) and (2). In Section 4, we provide the experimental results to verify our attacks. The paper concludes in Section 5.

2 Preliminaries

In this section, we introduce lattice and the LLL algorithm (Lenstra-Lenstra-Lovász algorithm). Then we refer to Coppersmith's techniques and the generalized reformulation summarized by Howgrave-Graham's lemma. Finally, we provide the condition for finding the common root and simply mention the running time of our method.

A lattice $\mathcal{L}$ spanned by linearly independent vectors $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_m \in \mathbb{R}^n$ is the set of all integer linear combinations of $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_m$. $(\boldsymbol{b}_1, \ldots, \boldsymbol{b}_m)$ is called a basis of $\mathcal{L}$ and m is known as the dimension. We usually consider a full-rank lattice when $m = n$. $\mathcal{L}$ can be denoted by

$$\mathcal{L}(\boldsymbol{b}_1, \ldots, \boldsymbol{b}_m) = \left\{ \sum_{i=1}^{m} z_i \boldsymbol{b}_i | z_i \in \mathbb{Z} \right\}.$$

For $i = 1, \ldots, m$, we regard each basis vector $\boldsymbol{b}_i$ as a row vector, hence $(\boldsymbol{b}_1, \ldots, \boldsymbol{b}_m)$ generates the $m \times n$ basis matrix B. Thus, this lattice can also be written as $\mathcal{L}(B)$. The determinant of $\mathcal{L}$ is calculated as $\det(\mathcal{L}) = \sqrt{\det(BB^T)}$. We have $\det(\mathcal{L}) = |\det(B)|$ when $\mathcal{L}$ is full-rank (B is a square matrix). It can be easily inferred that different bases of the same lattice do not change its determinant. Therefore we provide another definition $\det(\mathcal{L}) = \prod_{i=1}^{m} \|\boldsymbol{b}_i^*\|$, where $\boldsymbol{b}_1^*, \ldots, \boldsymbol{b}_m^*$ are derived from Gram-Schmidt orthogonalization to a basis $(\boldsymbol{b}_1, \ldots, \boldsymbol{b}_m)$, and $\|\cdot\|$ denotes the Euclidean norm of a vector.

The LLL algorithm proposed by Lenstra et al. [20] is practically used for finding approximate non-zero short lattice vectors due to its efficient running results computed in polynomial time. We provide the following substratal lemma about the outputs of the LLL algorithm.

Lemma 1 (LLL)**.** Let $\mathcal{L}$ be a lattice spanned by a basis $(\boldsymbol{b}_1, \boldsymbol{b}_2, \ldots, \boldsymbol{b}_m)$. The LLL algorithm outputs a reduced basis $(\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_m)$ of $\mathcal{L}$ in polynomial time, that satisfies

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

for $1 \leqslant i \leqslant m$.

The following lemma presented by Howgrave-Graham [4] gives a criterion for judging whether the desired small root of a modular equation is also root over $\mathbb{Z}$. To a given polynomial $g(x_1, \ldots, x_n) = \sum a_{i_1,\ldots,i_n} x_1^{i_1} \cdots x_n^{i_n}$, its norm is defined by $\|g(x_1, \ldots, x_n)\|^2 := \sum |a_{i_1,\ldots,i_n}|^2$.

Lemma 2 (Howgrave-Graham)**.** Let $g(x_1, \ldots, x_n) \in \mathbb{Z}[x_1, \ldots, x_n]$ be an integer polynomial that is a sum of at most m monomials. Suppose that

1. $g(x_1^{(0)}, \ldots, x_n^{(0)}) \equiv 0 \bmod R$, where $|x_1^{(0)}| < X_1, \ldots, |x_n^{(0)}| < X_n$,

2. $\|g(x_1X_1,\ldots,x_nX_n)\| < \frac{R}{\sqrt{m}}$.

Then it also holds over the integers, namely $g(x_1^{(0)},\ldots,x_n^{(0)}) = 0$.

Thus we can solve the polynomials derived from the LLL algorithm. Consider that we already have the first two basis vectors by the LLL algorithm, the condition for finding the common root over the integers implies that $2^{\frac{m}{4}}\det(\mathcal{L})^{\frac{1}{m-1}} < \frac{R}{\sqrt{m}}$, which leads to $\det(\mathcal{L}) < R^{m-1}2^{-\frac{m(m-1)}{4}}m^{-\frac{m-1}{2}}$. Since we usually have $m \ll R$, an error term ϵ is used on behalf of the small terms except R^m, and then it reduces to $\det(\mathcal{L}) \leqslant R^{m-\epsilon}$.

We obtain a lower triangular basis matrix in our method all the time. The determinant can be simply calculated as $\det(\mathcal{L}) = N^{s_N}X_1^{s_1}X_2^{s_2}$, where each s_i denotes the sum of total exponent of X_i or N that appears on the diagonal. Hence, we give the following condition,

$$N^{s_N}X_1^{s_1}X_2^{s_2} < R^m. \tag{3}$$

The running time of our method depends on the time of reducing basis matrix by the LLL algorithm and extracting the common root by the Gröbner basis computations. Both of them can be done in polynomial time, since the lattice dimension, the degrees of the desired polynomials, the bit-size of the entries of basis matrix, and the coefficients of polynomials are fixed in polynomial form of some parameters for concrete instances.

3 Solving simultaneous modular equations

According to Coppersmith's techniques, the basic idea for finding small roots of modular equations is to translate this problem to finding them over the integers. To do so, we construct a set of polynomials sharing the common root modulo R. Then we begin to search some integer linear combinations of the constructed polynomials' coefficient vectors whose norm is expected to be sufficiently small by the LLL algorithm.

First, we want to solve (1). We define the following shift polynomials for a positive integer s,

$$f_{i_1,i_2}(x_1,x_2) = (x_1-e_1')^{i_1}(x_2-e_2')^{i_2}N^{\max(s-i_1-i_2,0)},$$

where $|x_1| < X_1$ and $|x_2| < X_2$. We now deal with the above bivariate modular equation. As described previously, all polynomials $f_{i_1,i_2}(x_1,x_2)$ share the common root (d_1,d_2) modulo $p^{s(r-1)}$. From [19], we know the optimal condition for choosing the shift polynomials,

$$0 \leqslant \delta_1 i_1 + \delta_2 i_2 \leqslant \frac{r-1}{r+1}s.$$

When we consider the general case that $\delta_1 = \delta_2 = \delta$, there is a more concise condition,

$$0 \leqslant i_1 + i_2 \leqslant \frac{r-1}{r+1}\cdot\frac{s}{\delta}.$$

Afterwards, we begin to search an integer linear combination of all $f_{i_1,i_2}(x_1X_1,x_2X_2)$ by the LLL algorithm and ensure that its norm is sufficiently small in order to meet the conditions in Lemma 2. Here we know that $X_1 = N^{\delta_1}$, $X_2 = N^{\delta_2}$, and $R = p^{r-1} \approx N^{\frac{r-1}{r+1}}$. Then we build lattice $\mathcal{L}$ spanned by the corresponding coefficient vectors. It can also be represented by a square basis matrix whose rows are the polynomials' coefficient vectors. We use the LLL algorithm to find a small norm vector that leads to a small norm polynomial.

We define the monomial order $\prec$ as $x_1^{i_1}x_2^{i_2} \prec x_1^{j_1}x_2^{j_2}$ if $i_1+i_2 < j_1+j_2$ or $i_1+i_2 = j_1+j_2,\ i_1 > j_1$. Two simple examples are shown in Table 1 ($\delta_1 = \delta_2$) and Table 2 ($\delta_1 \neq \delta_2$), where other non-zero off-diagonal entries are denoted by $*$.

For a given parameter s, we can compute the dimension of the full-rank lattice, which is denoted by m. After that, our task is to compute $\det(\mathcal{L})$. Since it is a lower triangular square matrix, we can easily

Table 1 A simple example with $\delta_1 = 0.2$, $\delta_2 = 0.2$, $s = 2$, and $r = 2$

f_{i_1,i_2}	1	x_1	x_2	x_1^2	x_1x_2	x_2^2	x_1^3	$x_1^2x_2$	$x_1x_2^2$	x_2^3
$f_{0,0}$	N^2									
$f_{1,0}$	$*$	NX_1								
$f_{0,1}$	$*$		NX_2							
$f_{2,0}$	$*$	$*$		X_1^2						
$f_{1,1}$	$*$	$*$	$*$		X_1X_2					
$f_{0,2}$	$*$		$*$			X_2^2				
$f_{3,0}$	$*$	$*$		$*$			X_1^3			
$f_{2,1}$	$*$	$*$	$*$	$*$	$*$			$X_1^2X_2$		
$f_{1,2}$	$*$	$*$	$*$		$*$	$*$			$X_1X_2^2$	
$f_{0,3}$	$*$		$*$			$*$				X_2^3

Table 2 A simple example with $\delta_1 = 0.3$, $\delta_2 = 0.4$, $s = 3$, and $r = 3$

f_{i_1,i_2}	1	x_1	x_2	x_1^2	x_1x_2	x_2^2	x_1^3	$x_1^2x_2$	$x_1x_2^2$	x_2^3	x_1^4	$x_1^3x_2$	$x_1^2x_2^2$	$x_1x_2^3$	x_1^5
$f_{0,0}$	N^3														
$f_{1,0}$	$*$	N^2X_1													
$f_{0,1}$	$*$		N^2X_2												
$f_{2,0}$	$*$	$*$		NX_1^2											
$f_{1,1}$	$*$	$*$	$*$		NX_1X_2										
$f_{0,2}$	$*$		$*$			NX_2^2									
$f_{3,0}$	$*$	$*$		$*$			X_1^3								
$f_{2,1}$	$*$	$*$	$*$	$*$	$*$			$X_1^2X_2$							
$f_{1,2}$	$*$	$*$	$*$		$*$	$*$			$X_1X_2^2$						
$f_{0,3}$	$*$		$*$			$*$				X_2^3					
$f_{4,0}$	$*$	$*$		$*$			$*$				X_1^4				
$f_{3,1}$	$*$	$*$	$*$	$*$	$*$		$*$	$*$				$X_1^3X_2$			
$f_{2,2}$	$*$	$*$	$*$	$*$	$*$	$*$		$*$	$*$				$X_1^2X_2^2$		
$f_{1,3}$	$*$	$*$	$*$		$*$	$*$			$*$	$*$				$X_1X_2^3$	
$f_{5,0}$	$*$	$*$		$*$			$*$				$*$				X_1^5

compute it by counting the exponential numbers of X_1, X_2, and N on the diagonal, respectively.

$$m = \sum_{\delta_1 i_1 + \delta_2 i_2 = 0}^{\frac{r-1}{r+1}s} 1 = \frac{1}{2\delta_1\delta_2}\left(\frac{r-1}{r+1}s\right)^2 + o(s^2),$$

$$s_N = \sum_{i_1+i_2=0}^{s} (i_1+i_2+1)(s-i_1-i_2) = \frac{1}{6}s^3 + o(s^3),$$

$$s_1 = \sum_{\delta_1 i_1 + \delta_2 i_2 = 0}^{\frac{r-1}{r+1}s} i_1 = \frac{1}{6\delta_1^2\delta_2}\left(\frac{r-1}{r+1}s\right)^3 + o(s^3),$$

$$s_2 = \sum_{\delta_1 i_1 + \delta_2 i_2 = 0}^{\frac{r-1}{r+1}s} i_2 = \frac{1}{6\delta_1\delta_2^2}\left(\frac{r-1}{r+1}s\right)^3 + o(s^3).$$

Consequently, we know $\det(\mathcal{L}) = N^{s_N}X_1^{s_1}X_2^{s_2}$ for $X_1 = N^{\delta_1}$ and $X_2 = N^{\delta_2}$ from above. If the condition for finding the common root holds, the norms of the first two vectors derived from LLL-reduced basis are sufficiently small. Thus we can translate them into the corresponding polynomials g_1 and g_2 having the

Table 3 Comparison with previous results on numerical upper bound of δ for various r

r	$\left(\frac{r-1}{r+1}\right)^{\frac{3}{2}}$	[17]	$\frac{r}{(r+1)^2}$ [16]	$\left(\frac{r-1}{r+1}\right)^2$ [16]
2	0.192	0.395	0.222	0.111
3	0.353	0.410	0.187	0.25
4	0.464	0.437	0.16	0.36
5	0.544	0.464	0.138	0.444
6	0.603	0.489	0.122	0.510
7	0.649	0.512	0.109	0.562
8	0.685	0.532	0.098	0.604

same root and finally solve (d_1, d_2).

Although the polynomials g_1, g_2 derived from the LLL algorithm are linearly independent, they may have a common factor. Assumption 1 ensures that we can solve the common root. Additionally, our experiments confirm this assumption.

Now we estimate δ_1 and δ_2. Then (3) indicates that the following inequality holds,

$$N^{\frac{1}{6}s^3+o(s^3)} \cdot N^{\delta_1\cdot\frac{1}{6\delta_1^2\delta_2}\left(\frac{r-1}{r+1}s\right)^3+o(s^3)} \cdot N^{\delta_2\cdot\frac{1}{6\delta_1\delta_2^2}\left(\frac{r-1}{r+1}s\right)^3+o(s^3)} < N^{\frac{r-1}{r+1}s\cdot\frac{1}{2\delta_1\delta_2}\left(\frac{r-1}{r+1}s\right)^2+o(s^3)}.$$

For taking $s \to \infty$ and omitting the lower term $o(s^3)$, we have

$$\frac{1}{6} + \frac{\delta_1}{6\delta_1^2\delta_2}\left(\frac{r-1}{r+1}\right)^3 + \frac{\delta_2}{6\delta_1\delta_2^2}\left(\frac{r-1}{r+1}\right)^3 < \frac{r-1}{r+1}\cdot\frac{1}{2\delta_1\delta_2}\left(\frac{r-1}{r+1}\right)^2.$$

It can be simplified to $\frac{1}{\delta_1\delta_2}\left(\frac{r-1}{r+1}\right)^3 > 1$, which further reduces to

$$\delta_1\delta_2 < \left(\frac{r-1}{r+1}\right)^3.$$

We assume that $\delta_1 = \delta_2 = \delta$ for the comparison with previous results, which is showed in Table 3. Therefore, we obtain the asymptotic bound $\delta < \left(\frac{r-1}{r+1}\right)^{\frac{3}{2}}$. From Table 3, we discover that our bound is better for $r \geqslant 4$.

Now we are to solve (2). We define the following shift polynomials for a positive integer s,

$$\tilde{f}_{i_1,i_2}(x_1,x_2) = (x_1+a_1)^{i_1}(x_2+a_2)^{i_2}N^{\max(s-i_1-i_2,0)},$$

where $|x_1| < X_1$ and $|x_2| < X_2$. All polynomials $\tilde{f}_{i_1,i_2}(x_1,x_2)$ share the common root $(\tilde{d}_1,\tilde{d}_2)$ modulo $p^{s(r-1)}$. The basis matrix construction is similar and we skip it. Assuming $X_1 = N^{\gamma_1}$ and $X_2 = N^{\gamma_2}$, we obtain the condition,

$$\gamma_1\gamma_2 < \left(\frac{r-1}{r+1}\right)^3.$$

4 Experimental results

In this section, we state some experimental results to show the performance of our method. We did several experiments to check if the assumption holds. These experiments were performed under Ubuntu 15.04 running on a computer with Intel(R) Core(TM) i5 M 450 CPU 2.40 GHz, 2 GB RAM and 3 MB Cache. We carried out the experiments by using the LLL implementation available in Shoup's NTL library. The numbers used in each experiment were chosen uniformly at random.

We choose $r = 2,3,4,5$ and $s = 1,2$ for the experiments, and the dimension of the lattice is denoted by "ld" in Table 4. "sp" is the number of suitable polynomials outputted by the LLL algorithm. In the

Table 4 Experimental results on δ_1 and δ_2 for various r

r	s	N (bits)	δ_1	δ_2	ld	sp	LLL time (s)
2	1	900	0.100	0.100	10	9	0.096
2	1	900	0.124	0.125	6	4	0.104
3	2	1000	0.260	0.260	10	9	14.881
3	2	1000	0.298	0.298	10	9	15.436
4	2	2000	0.350	0.355	10	9	46.144
4	2	2000	0.397	0.400	10	7	34.178
5	2	1800	0.388	0.392	10	9	19.956
5	2	1800	0.435	0.442	10	9	17.128
5	2	1800	0.480	0.495	6	5	21.756

experiments, we always obtain more than two independent polynomials g_1, g_2 from the output. Thus, we can solve the common root by the Gröbner basis computations.

While the modulus gets larger, we need more time for the LLL algorithm. For the purpose of successfully extracting the common root, one would better put a bit more polynomials into the Gröbner basis computations. Since we apply a lattice of lower dimension to show the performance of our attacks, the results are some bits away from the asymptotic bound. This requires optimized cryptanalysis and more efficient lattice reduction algorithms.

5 Conclusion

We show that we can perform lattice-based attacks on Prime Power RSA with two private exponents. Suppose that we know $e \approx N$, $d < N^{\delta_1}$, and $d < N^{\delta_2}$. If $\delta_1\delta_2 < \left(\frac{r-1}{r+1}\right)^3$ holds, it is insecure.

Our work is an application of Coppersmith's techniques and also Takayasu and Kunihiro's better lattice constructions. We apply it to solve bivariate modular polynomials and further extend our method to partial key exposure attacks. However, the theoretical results are still heuristic unless Assumption 1 is confirmed. So a way to handle this heuristic assumption is still an open problem.

Acknowledgements

This work was supported partially by National Natural Science Foundation of China (Grant Nos. 61271271, 61522210), 100 Talents Program of Chinese Academy of Sciences, and Fundamental Research Funds for the Central Universities in China (Grant No. WK2101020005).

References

1 Rivest R L, Shamir A, Adleman L. A method for obtaining digital signatures and public-key cryptosystems. Commun ACM, 1978, 21: 120–126

2 Coppersmith D. Finding a small root of a univariate modular equation. In: Proceedings of International Conference on the Theory and Application of Cryptographic Techniques, Saragossa, 1996. 155–165

3 Coppersmith D. Small solutions to polynomial equations, and low exponent RSA vulnerabilities. J Cryptol, 1997, 10: 233–260

4 Howgrave-Graham N. Finding small roots of univariate modular equations revisited. In: Darnell M, ed. Crytography and Coding. Berlin: Springer, 1997. 131–142

5 Wiener M J. Cryptanalysis of short RSA secret exponents. IEEE Trans Inform Theory, 1990, 36: 553–558

6 Boneh D, Durfee G. Cryptanalysis of RSA with private key d less than $N^{0.292}$. In: Proceedings of International Conference on the Theory and Application of Cryptographic Techniques, Prague, 1999. 1–11

7 Boneh D, Durfee G. Cryptanalysis of RSA with private key d less than $N^{0.292}$. IEEE Trans Inform Theory, 2000, 46: 1339–1349

8 Blömer J, May A. Low secret exponent RSA revisited. In: Silverman J H, ed. Cryptography and Lattices. Berlin: Springer, 2001. 4–19

9 Blömer J, May A. New partial key exposure attacks on RSA. In: Proceedings of 23rd Annual International Cryptology Conference, Santa Barbara, 2003. 27–43

10 Ernst M, Jochemsz E, May A, et al. Partial key exposure attacks on RSA up to full size exponents. In: Proceedings of 24th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Aarhus, 2005. 371–386

11 Aono Y. A new lattice construction for partial key exposure attack for RSA. In: Proceedings of 12th International Conference on Practice and Theory in Public Key Cryptography, Irvine, 2009. 34–53

12 Sarkar S. Partial key exposure: generalized framework to attack RSA. In: Proceedings of 12th International Conference on Cryptology in India, Chennai, 2011. 76–92

13 Joye M, Lepoint T. Partial key exposure on RSA with private exponents larger than N. In: Ryan M D, Smyth B, Wang G L, eds. Information Security Practice and Experience. Berlin: Springer, 2012. 369–380

14 Takagi T. Fast RSA-type cryptosystem modulo p^kq. In: Proceedings of 18th Annual International Cryptology Conference, Santa Barbara, 1998. 318–326

15 Boneh D, Durfee G, Howgrave-Graham N. Factoring $N = p^rq$ for large r. In: Proceedings of 19th Annual International Cryptology Conference, Santa Barbara, 1999. 326–337

16 May A. Secret exponent attacks on RSA-type schemes with moduli $N = p^rq$. In: Proceedings of 7th International Workshop on Theory and Practice in Public Key Cryptography, Singapore, 2004. 218–230

17 Sarkar S. Small secret exponent attack on RSA variant with modulus $N = p^rq$. Designs Codes Cryptogr, 2014, 73: 383–392

18 Itoh K, Kunihiro N, Kurosawa K. Small secret key attack on a variant of RSA (due to Takagi). In: Proceedings of the Cryptographers' Track at the RSA Conference, San Francisco, 2008. 387–406

19 Takayasu A, Kunihiro N. Better lattice constructions for solving multivariate linear equations modulo unknown divisors. In: Proceedings of 18th Australasian Conference, ACISP 2013, Brisbane, 2013. 118–135

20 Lenstra A K, Lenstra H W, Lovász L. Factoring polynomials with rational coefficients. Math Ann, 1982, 261: 515–534