Implicit-Key Attack on the RSA Cryptosystem

Mengce Zheng(✉) and Honggang Hu

Key Laboratory of Electromagnetic Space Information, CAS,
University of Science and Technology of China, Hefei, China
{mczheng,hghu2005}@ustc.edu.cn

Abstract. In this paper, we address the security evaluation issue of the RSA cryptosystem with implicitly related private keys. We formulate the attack scenario and propose a novel implicit-key attack using the lattice-based method. When given public information (N_1, e_1), (N_2, e_2) and the amount of shared bits of the private keys d_1 and d_2, one can conduct the implicit-key attack to factor N_1, N_2 in polynomial time under a certain condition. We show that the RSA cryptosystem is more insecure when taking the implicitly related keys into consideration. The experimental results are provided to verify the validity of our proposed attack.

Keywords: RSA cryptosystem · Cryptanalysis · Implicit-key attack · Lattice · Coppersmith's techniques

1 Introduction

The RSA cryptosystem [14] is the most attractive one in public key cryptography and plays an important role in the field of cybersecurity. The main mathematical equation is $ed \equiv 1 \bmod \varphi(N)$, where e, d, N and $\varphi(N)$ are described as follows. $N = pq$ is the product of two large primes of the same bit-size. The respective public and private keys e, d are also called public/encryption and private/decryption exponents. $\varphi(N) = (p-1)(q-1)$ is Euler's totient function of N. To encrypt an integer m, one computes $c = m^e \bmod N$. To decrypt a ciphertext c, one needs to compute $c^d \bmod N$.

The security of the RSA cryptosystem has been investigated in [1,12]. Since Coppersmith [4] introduced the lattice-based method, its variations have been widely used for attacking the RSA cryptosystem such as [2,5–7,10,16]. Among the various attacks, the partial key exposure attack and the implicit factoring problem are two attractive ones.

In 2005, Ernst et al. [7] presented several concrete attacks that work up to full size exponents. This attack type was first studied by Boneh, Durfee, and Frankel in [3]. In other words, partial key exposure attack can be seemed as the problem of attacking RSA with an oracle providing explicit information about d. In 2009, May and Ritzenhofen [13] proposed a new approach to factor RSA modulus with an oracle providing implicit information about p. To be specific,

F. Liu et al. (Eds.): SciSec 2019, LNCS 11933, pp. 354–362, 2019.
https://doi.org/10.1007/978-3-030-34637-9_26

for RSA moduli $N_1 = p_1q_1$ and $N_2 = p_2q_2$ with α-bit q_i and p_1, p_2 share at least t many least significant bits (LSBs), it has been proved that if $t > 2(\alpha + 2)$, one can find q_1 and q_2. Thus, N_1 and N_2 can be factored easily. Other cases such as shared most significant bits (MSBs), shared middle bits [8] and some improved methods [15] were proposed afterwards.

Inspired by the partial key exposure attack and the implicit factoring problem with existing drawbacks, we concentrate on a weaker setting, where some implicit information about the private keys is given. We informally formulate the following scenario related to the implicit-key attack. Let (N_1, e_1, d_1) and (N_2, e_2, d_2) be two different RSA key pairs with N_1, N_2 of the same bit-size. Suppose we know some implicit information about the private keys, i.e. the amount of shared MSBs and LSBs of d_1 and d_2. The goal is to factor N_1 and N_2 in polynomial time from the knowledge of the implicitly related private keys.

It is opposed to the previous cryptanalyses dealing with only one RSA key pair. Our work can cover other similar works and make further improvements. Once RSA instances are generated with imperfect randomness or backdoored keys, one may encounter such attack scenario. Though such implicit-key attack may not directly influence the security of the RSA cryptosystem. We consider the following issues for which our theoretical study may be interesting. One is to deeply disclose the vulnerability of RSA with weaker conditions. Moreover, we want to investigate how one can further extend previous attacks, where partial key exposure and implicit hint are combined.

We adapt the Jochemsz-May strategy [10] as a main mathematical tool to solve the common root of multivariate equations. To achieve theoretical effects, the lattice-based method relies on the following heuristic assumption. One can obtain algebraically independent polynomials by the lattice-based method, and then efficiently solve the common root by the Gröbner basis computation. This heuristic assumption always holds in the simulated experiments like previous works in the literature. 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.

The rest of the paper is organized as follows. We provide the basic knowledge of lattice reduction theory and the condition for finding the common root in Sect. 2. In Sect. 3, we formulate the concrete attack scenario and present the implicit-key attack. In Sect. 4, we provide the experimental results with more details. Finally, concluding remarks are given in Sect. 5.

2 Preliminaries

In this section, we briefly introduce the LLL algorithm [11] and Coppersmith's techniques (also stated as Howgrave-Graham's lemma [9]). Then, we provide the condition for finding the common root and simply mention the running time. One can refer to [12] for more details about the lattice-based 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 their integer linear combinations. Thus, the lattice $\mathcal{L}$ can be written

as $\mathcal{L}(\boldsymbol{b}_1, \ldots, \boldsymbol{b}_m) = \{\sum_{i=1}^{m} z_i \boldsymbol{b}_i | z_i \in \mathbb{Z}\}$. For $i = 1, \ldots, m$, we regard each basis vector $\boldsymbol{b}_i$ as a row vector, which generates so-called $m \times n$ basis matrix B. The determinant of $\mathcal{L}$ is calculated as $\det(\mathcal{L}) = \sqrt{\det(BB^T)}$. We usually consider a full-rank lattice for $m = n$ and hence $\det(\mathcal{L}) = |\det(B)|$.

The LLL algorithm proposed by Lenstra, Lenstra, and Lovász [11] is practically used for finding approximately non-zero short lattice vectors due to its efficient running results. We provide the following substratal lemma about its outputs.

Lemma 1. *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 $(\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_m)$ in polynomial time. For $1 \leq i \leq m$, the first i many reduced basis vectors satisfy*

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

The following lemma presented by Howgrave-Graham [9] gives a criterion for judging whether the desired small root of a modular equation is also a 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 as $\|g(x_1, \ldots, x_n)\|^2 := \sum |a_{i_1,\ldots,i_n}|^2$.

Lemma 2. *Let $g(x_1, \ldots, x_n) \in \mathbb{Z}[x_1, \ldots, x_n]$ be an n-variate integer polynomial, which is a sum of at most m monomials. Suppose that (1) $g(\tilde{x}_1, \ldots, \tilde{x}_n) \equiv 0 \bmod R$, where $|\tilde{x}_1| < X_1, \ldots, |\tilde{x}_n| < X_n$, and (2) $\|g(x_1X_1, \ldots, x_nX_n)\| < R/\sqrt{m}$. Then $g(\tilde{x}_1, \ldots, \tilde{x}_n) = 0$ holds over the integers.*

Applying the lattice-based method, we can combine Lemma 1 with Lemma 2 to solve modular/integer polynomials. Once having the first l reduced vectors, one can solve the unknown variables for $2^{\frac{m(m-1)}{4(m+1-l)}} \det(\mathcal{L})^{\frac{1}{m+1-l}} < R/\sqrt{m}$, which can be further reduced to $\det(\mathcal{L}) \leq R^{m-\epsilon}$ with an error term ϵ, or a simplified condition $\det(\mathcal{L}) < R^m$. We can construct an upper/lower triangular basis matrix by the lattice-based method. The lattice determinant can be calculated as $\det(\mathcal{L}) = R^{u_R} \prod_{i=1}^{n} X_i^{u_i}$, where u_i denotes the exponent sum of each X_i or R that appear on the diagonal in the corresponding basis matrix. Hence, the condition $\det(\mathcal{L}) < R^m$ can be rewritten as $R^{u_R} \prod_{i=1}^{n} X_i^{u_i} < R^m$.

We sketch the lattice-based method and derive the crucial condition for finding small roots of integer polynomials. Lattice-based attacks using Coppersmith's techniques start with an integer/modular equation in some unknown parameters of given RSA instances. To carry out the proposed implicit-key attack, we aim to find a suitable root of a five-variate integer polynomial $f(x_1, x_2, x_3, x_4, x_5)$.

First, we need to estimate the upper bounds X_i as mentioned in Lemma 2. Moreover, we define the largest size of an individual term in $f(x_1, x_2, x_3, x_4, x_5)$ as $X_\infty = \|f(x_1X_1, x_2X_2, x_3X_3, x_4X_4, x_5X_5)\|_\infty$ that is related to the definition of a sufficient large modulus R. Then, a lattice basis matrix is constructed using the shift polynomials defined in two monomial sets S and T. Based on the Jochemsz-May strategy, the solvable condition reduces to $X_1^{s_1} X_2^{s_2} X_3^{s_3} X_4^{s_4} X_5^{s_5} < X_\infty^{s_g}$ for $s_j = \sum_{T \setminus S} i_j$ and $s_g = |S|$ in our proposed implicit-key attack. More details about the concrete lattice construction for a given specific polynomial will be described in Sect. 3.

Under the above condition, we can compute the first l reduced basis vectors using the LLL algorithm and then obtain the equations $f_1, \ldots, f_l$ that all share the same root over the integers. Next, we use the Gröbner basis computation to extract the common root. The running time depends on the time of reducing the basis matrix and extracting the common root. For conducting the implicit-key attack on concrete RSA instances, both of them can be done in polynomial time.

3 Implicit-Key Attack

We describe the implicit-key attack by providing the concrete construction for two RSA instances (N_1, e_1, d_1) and (N_2, e_2, d_2). Consider a general case when e_1, e_2 are of arbitrary bit-size and d_1, d_2 share some MSBs and LSBs leaving one different block in the middle. Unless otherwise noted, N in this paper denotes the greater one of N_1, N_2 and $\log_2 N$ denotes their bit-size (suppose two RSA moduli are of the same bit-size). Our main result is stated as follows.

Theorem 1. *Let $N_1 = p_1 q_1, N_2 = p_2 q_2$ be two different RSA moduli of the same bit-size, and p_1, q_1, p_2, q_2 be primes of the same bit-size. Let e_1, d_1, e_2, d_2 satisfy $e_1 d_1 \equiv 1 \bmod \varphi(N_1)$ and $e_2 d_2 \equiv 1 \bmod \varphi(N_2)$, such that $e_1 = N^{\alpha_1}, e_2 = N^{\alpha_2}$ and $d_1, d_2 \approx N^{\delta}$. Suppose that d_1 and d_2 share $\beta_1 \log_2 N$ MSBs and $\beta_2 \log_2 N$ LSBs. Then N_1, N_2 can be factored in polynomial time if*

$$\delta < \frac{(\alpha + \beta - 1)(1 + 10\tau + 20\tau^2) - 10\tau^2 - 30\tau^3}{4 + 30\tau + 40\tau^2} - \frac{\alpha}{2} + 1,$$

where $\alpha = \alpha_1 + \alpha_2$, $\beta = \beta_1 + \beta_2$ and τ is the only positive root of

$$120x^4 + 180x^3 + (86 - 20\alpha - 20\beta)x^2 + (16 - 8\alpha - 8\beta)x - \alpha - \beta + 1 = 0.$$

Proof. From the main equation of the RSA cryptosystem, namely $ed \equiv 1 \bmod \varphi(N)$, we have $e_1 d_1 = k_1(N_1 + 1 - p_1 - q_1) + 1$ and $e_2 d_2 = k_2(N_2 + 1 - p_2 - q_2) + 1$ for two unknown positive integers k_1 and k_2. Multiplying the above equations by e_2 and e_1 respectively and then subtracting, we have

$$e_1 e_2 (d_1 - d_2) = e_2 k_1 (N_1 + 1 - p_1 - q_1) + e_2 - e_1 k_2 (N_2 + 1 - p_2 - q_2) - e_1. \quad (1)$$

Consider we know $d_1, d_2 \approx N^{\delta}$ sharing $\beta_1 \log_2 N$ MSBs and $\beta_2 \log_2 N$ LSBs. Hence, it implies that $d_1 = d_{\mathrm{MSB}} 2^{(\delta - \beta_1)\log_2 N} + \bar{d}_1 2^{\beta_2 \log_2 N} + d_{\mathrm{LSB}}$ and $d_2 = d_{\mathrm{MSB}} 2^{(\delta - \beta_1)\log_2 N} + \bar{d}_2 2^{\beta_2 \log_2 N} + d_{\mathrm{LSB}}$, where d_{MSB} and d_{LSB} are shared MSBs and LSBs, $\bar{d}_1$ and $\bar{d}_2$ are different values in the middle block. Substituting d_1 and d_2 into (1), it can be rewritten as

$$e_1 e_2 (\bar{d}_2 - \bar{d}_1) N^{\beta_2} + e_2 k_1 (N_1 + 1 - p_1 - q_1) - e_1 k_2 (N_2 + 1 - p_2 - q_2) + e_2 - e_1 = 0.$$

The known values are $a_1 = e_1 e_2 N^{\beta_2}$, $a_2 = e_2(N_1 + 1)$, $a_3 = -e_1(N_2 + 1)$, $a_4 = -e_2$, $a_5 = e_1$, and $a_6 = e_2 - e_1$. The unknown variables are $x_1 = \bar{d}_2 - \bar{d}_1$, $x_2 = k_1$, $x_3 = k_2$, $x_4 = p_1 + q_1$, and $x_5 = p_2 + q_2$. We aim to find a suitable root of $f(x_1, x_2, x_3, x_4, x_5) := a_1 x_1 + a_2 x_2 + a_3 x_3 + a_4 x_2 x_4 + a_5 x_3 x_5 + a_6$.

If e_1 and e_2 have a nontrivial great common divisor, one can do the division to make the polynomial irreducible. Suppose we know $e_1 = N^{\alpha_1}$ and $e_2 = N^{\alpha_2}$. The upper bounds X_i are estimated as follows. $X_1 = N^{\delta-\beta}$ for $\beta = \beta_1 + \beta_2$, $X_2 = N^{\alpha_1+\delta-1}$, $X_3 = N^{\alpha_2+\delta-1}$, and $X_4 = X_5 = N^{1/2}$. The maximal coefficient X_∞ can be easily calculated as $X_\infty \approx N^{\alpha+\delta}$ for $\alpha = \alpha_1 + \alpha_2$.

We follow the Jochemsz-May strategy [10] and use extra shifts of x_4 and x_5 for solving $f(x_1, x_2, x_3, x_4, x_5)$. Define two monomial sets S and T for two integers $s \geq 1$ and $t \geq 0$.

$$S = \bigcup_{0 \leq j_4, j_5 \leq t} \left\{ x_1^{i_1} x_2^{i_2} x_3^{i_3} x_4^{i_4+j_4} x_5^{i_5+j_5} \middle| x_1^{i_1} x_2^{i_2} x_3^{i_3} x_4^{i_4} x_5^{i_5} \text{ is a monomial of } f^{s-1} \right\},$$

$$T = \bigcup_{0 \leq j_4, j_5 \leq t} \left\{ x_1^{i_1} x_2^{i_2} x_3^{i_3} x_4^{i_4+j_4} x_5^{i_5+j_5} \middle| x_1^{i_1} x_2^{i_2} x_3^{i_3} x_4^{i_4} x_5^{i_5} \text{ is a monomial of } f^{s} \right\}.$$

Through the expansion of f^{s-1} and f^s, we know the relation of $x_1^{i_1} x_2^{i_2} x_3^{i_3} x_4^{i_4} x_5^{i_5}$ in S and T to their exponents i_1, i_2, i_3, i_4, i_5, respectively.

Let $R = X_\infty X_1^{s-1} X_2^{s-1} X_3^{s-1} X_4^{s-1+t} X_5^{s-1+t}$, we define $f' = a_6^{-1} f \bmod R$ and the shift polynomials below,

$$g_{i_1,i_2,i_3,i_4,i_5} : x_1^{i_1} x_2^{i_2} x_3^{i_3} x_4^{i_4} x_5^{i_5} f' X_1^{s-1-i_1} X_2^{s-1-i_2} X_3^{s-1-i_3} X_4^{s-1+t-i_4} X_5^{s-1+t-i_5}, \quad \text{for } x_1^{i_1} x_2^{i_2} x_3^{i_3} x_4^{i_4} x_5^{i_5} \in S,$$

$$g'_{i_1,i_2,i_3,i_4,i_5} : x_1^{i_1} x_2^{i_2} x_3^{i_3} x_4^{i_4} x_5^{i_5} R, \quad \text{for } x_1^{i_1} x_2^{i_2} x_3^{i_3} x_4^{i_4} x_5^{i_5} \in T \backslash S.$$

The lattice $\mathcal{L}$ is constructed by the coefficient vectors of g_{i_1,i_2,i_3,i_4,i_5} and g'_{i_1,i_2,i_3,i_4,i_5} with $x_i X_i$ substituting for each x_i. We have $u_R = |T \backslash S|$, $u_j = \sum_T i_j$ and $m = |T|$. More precisely, the diagonal elements of g_{i_1,i_2,i_3,i_4,i_5} is equal to R/X_∞ and $u_j = \sum_S i_j + \sum_{T \backslash S} i_j$. So $R^{u_R} \prod_{i=1}^5 X_i^{u_i} < R^m$ implies $R^{u_R} (R/X_\infty)^{s_g} \prod_{i=1}^5 X_i^{s_i} < R^{u_R + s_g}$ for $s_j = \sum_{T \backslash S} i_j$ and $s_g = |S|$, which can be reduced to

$$X_1^{s_1} X_2^{s_2} X_3^{s_3} X_4^{s_4} X_5^{s_5} < X_\infty^{s_g}. \tag{2}$$

We now calculate s_j for $j = 1, \ldots, 5$ and s_g by above definitions. Taking $t = \tau s$ for $\tau \geq 0$ and omitting the lower term for simplicity, we obtain

$$s_g = s_1 = \frac{1}{120}(1 + 10\tau + 20\tau^2)s^5, \ s_2 = s_3 = \frac{1}{120}(2 + 15\tau + 20\tau^2)s^5,$$
$$s_4 = s_5 = \frac{1}{120}(1 + 10\tau + 30\tau^2 + 30\tau^3)s^5.$$

We substitute the values of X_j, s_j and X_∞, s_g into the condition (2) and obtain $1 + 10\tau + 30\tau^2 + 30\tau^3 + (1 + 10\tau + 20\tau^2)(\delta - \beta) + (2 + 15\tau + 20\tau^2)(\alpha + 2\delta - 2) < (1 + 10\tau + 20\tau^2)(\alpha + \delta)$. It leads to

$$\delta < \frac{(\alpha + \beta - 1)(1 + 10\tau + 20\tau^2) - 10\tau^2 - 30\tau^3}{4 + 30\tau + 40\tau^2} - \frac{\alpha}{2} + 1.$$

As α and β are already given, the value of the right side can be maximized by an optimal value of τ. It is easy to see that τ is the only positive root of

$$120x^4 + 180x^3 + (86 - 20\alpha - 20\beta)x^2 + (16 - 8\alpha - 8\beta)x - \alpha - \beta + 1 = 0.$$

We can obtain four integer polynomials f_1, f_2, f_3 and f_4 apart from f by the proposed implicit-key attack. Moreover, f, f_1, f_2, f_3 and f_4 share the common root $(\bar{d}_2 - \bar{d}_1, k_1, k_2, p_1 + q_1, p_2 + q_2)$ over the integers. Thus, we can extract $p_1 + q_1$ and $p_2 + q_2$ that directly lead to the factorization of N_1 and N_2. □

If e_1 and e_2 are of full bit-size, i.e. $\alpha = 2$, we immediately know τ is the only positive root of $120x^4 + 180x^3 + (46 - 20\beta)x^2 - 8\beta x - \beta - 1 = 0$. Therefore, we show that N_1, N_2 can be factored in polynomial time for $\beta = \beta_1 + \beta_2$ if

$$\delta < \frac{(\beta + 1)(1 + 10\tau + 20\tau^2) - 10\tau^2 - 30\tau^3}{4 + 30\tau + 40\tau^2}. \tag{3}$$

We illustrate the above condition (3) with respect to various β's in Fig. 1. It is oblivious that we achieve higher insecure bound on δ as β increases, which means that the RSA cryptosystem with implicitly related keys is more vulnerable.

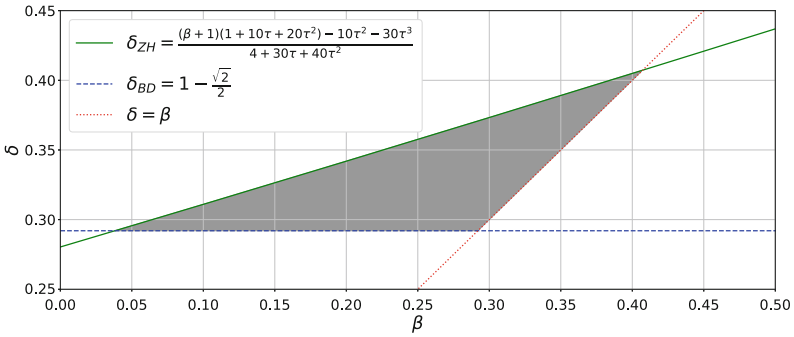

Fig. 1. The comparison of previous result (i.e. $\delta < \delta_{BD}$) and ours (i.e. $\delta < \delta_{ZH}$). The gray region shows our asymptotic improvement using the proposed implicit-key attack.

4 Experimental Results

To achieve the asymptotic bound on δ, the parameter $\tau = t/s$ should be less than 0.2 from our theoretical observation. For the smallest positive integer $t = 1$, s should be at least 6. Therefore, the dimension of the corresponding lattice will be $m = 966$, which seems impossible for our simulated experimental environment. Thus, we always choose $t = 0$ (i.e. $\tau = 0$) in the simulated numerical experiments.

The experiments were carried out by `SageMath` under Windows 10 running on a laptop with Intel Core i7-8550U CPU 1.80 GHz. The numbers for generating the parameters of two RSA instances were chosen at random. During the

experiments, we collected much more polynomials satisfying our requirement and extracted the common root by the Gröbner basis computation.

We would generate 1024-bit moduli in the experiments and all the public exponents appeared are near full bit-size for simplicity. The δ_t-column provides the theoretical bound on δ for fixed β_1 and β_2 (with $\tau = 0$). The δ_e-column provides the experimental bound on δ for the same β_1, β_2 and $\log_2 N = 1024$ in distinct lattice settings. We denote the dimension of the corresponding lattice by m and the running time of the proposed attack is denoted by Time in seconds.

For given two distinct 1024-bit moduli and d_1, d_2 sharing some MSBs and LSBs, we choose $s = 1, 2, 3$ and $t = 0$ to construct the lattices. Hence, we need to reduce 6-dimensional, 21-dimensional and 56-dimensional lattices using the LLL algorithm. The results of the comparison of the theoretical and experimental insecure bounds are showed in Table 1.

Table 1. The theoretical and experimental results of the proposed implicit-key attack

$\log_2 N = 1024$			$s = 1,\ m = 6$		$s = 2,\ m = 21$		$s = 3,\ m = 56$	
β_1	β_2	δ_t	δ_e	Time	δ_e	Time	δ_e	Time
0.043	0.043	0.271	0.259	0.004	0.264	0.623	0.270	47.59
0.064	0.101	0.291	0.280	0.004	0.286	0.621	0.291	47.17
0.107	0.142	0.312	0.300	0.004	0.307	0.682	0.311	37.23
0.150	0.150	0.325	0.315	0.005	0.321	0.522	0.325	32.02

In each experiment, we collected sufficient polynomials sharing the common root over the integers. Then we put several equations into the Gröbner basis computation and finally obtained the correct values of $p_1 + q_1$ and $p_2 + q_2$, which lead to the factorization of N_1 and N_2, respectively. If the Gröbner basis computation did not directly output the desired root, we would first calculate the value of x_3 and then extract the solution of the remaining variables. As the lattice dimension gets larger, the experimental insecure bound becomes higher and the running time gets longer. From Table 1, we observe that $s = 3$ is already enough for performing the implicit-key attack since the experimental result is very close to the theoretical bound.

5 Concluding Remarks

In this paper, we focus on a new attack scenario concerning implicitly related private keys. Our goal is to factor RSA moduli using the implicit information about the related keys. We propose the implicit-key attack based on Coppersmith's techniques, which is applied for solving modular/integer polynomials as a powerful tool.

The proposed implicit-key attack can reveal the vulnerability of the RSA cryptosystem with implicitly related keys. We further verify the validity of the

proposed attack by several numerical experiments. We would like to extend the implicit-key attack for an arbitrary number n of unknown variables. However, it seems less efficient as n gets greater since the running time is exponential in n.

Acknowledgments. The authors would like to thank the anonymous reviewers for their valuable comments and suggestions. This work was partially supported by the National Natural Science Foundation of China (Grant No. 61632013) and Anhui Initiative in Quantum Information Technologies under Grant AHY150400.

References

1. Boneh, D.: Twenty years of attacks on the RSA cryptosystem. Not. AMS **46**(2), 203–213 (1999)
2. Boneh, D., Durfee, G.: Cryptanalysis of RSA with private key d less than $N^{0.292}$. IEEE Trans. Inf. Theory **46**(4), 1339–1349 (2000)
3. Boneh, D., Durfee, G., Frankel, Y.: An attack on RSA given a small fraction of the private key bits. In: Ohta, K., Pei, D. (eds.) ASIACRYPT 1998. LNCS, vol. 1514, pp. 25–34. Springer, Heidelberg (1998). https://doi.org/10.1007/3-540-49649-1_3
4. Coppersmith, D.: Small solutions to polynomial equations, and low exponent RSA vulnerabilities. J. Cryptol. **10**(4), 233–260 (1997)
5. Coron, J.-S.: Finding small roots of bivariate integer polynomial equations revisited. In: Cachin, C., Camenisch, J.L. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 492–505. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-24676-3_29
6. Coron, J.-S.: Finding small roots of bivariate integer polynomial equations: a direct approach. In: Menezes, A. (ed.) CRYPTO 2007. LNCS, vol. 4622, pp. 379–394. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-74143-5_21
7. Ernst, M., Jochemsz, E., May, A., de Weger, B.: Partial key exposure attacks on RSA up to full size exponents. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 371–386. Springer, Heidelberg (2005). https://doi.org/10.1007/11426639_22
8. Faugère, J.-C., Marinier, R., Renault, G.: Implicit factoring with shared most significant and middle bits. In: Nguyen, P.Q., Pointcheval, D. (eds.) PKC 2010. LNCS, vol. 6056, pp. 70–87. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-13013-7_5
9. Howgrave-Graham, N.: Finding small roots of univariate modular equations revisited. In: Darnell, M. (ed.) Cryptography and Coding 1997. LNCS, vol. 1355, pp. 131–142. Springer, Heidelberg (1997). https://doi.org/10.1007/BFb0024458
10. Jochemsz, E., May, A.: A strategy for finding roots of multivariate polynomials with new applications in attacking RSA variants. In: Lai, X., Chen, K. (eds.) ASIACRYPT 2006. LNCS, vol. 4284, pp. 267–282. Springer, Heidelberg (2006). https://doi.org/10.1007/11935230_18
11. Lenstra, A.K., Lenstra, H.W., Lovász, L.: Factoring polynomials with rational coefficients. Math. Ann. **261**(4), 515–534 (1982)
12. May, A.: Using LLL-reduction for solving RSA and factorization problems. In: Nguyen, P.Q., Vallée, B. (eds.) The LLL Algorithm, pp. 315–348. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-02295-1_10

13. May, A., Ritzenhofen, M.: Implicit factoring: on polynomial time factoring given only an implicit hint. In: Jarecki, S., Tsudik, G. (eds.) PKC 2009. LNCS, vol. 5443, pp. 1–14. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-00468-1_1
14. Rivest, R.L., Shamir, A., Adleman, L.: A method for obtaining digital signatures and public-key cryptosystems. Commun. ACM **21**(2), 120–126 (1978)
15. Sarkar, S., Maitra, S.: Approximate integer common divisor problem relates to implicit factorization. IEEE Trans. Inf. Theory **57**(6), 4002–4013 (2011)
16. Zheng, M., Hu, H., Wang, Z.: Generalized cryptanalysis of RSA with small public exponent. Sci. China Inf. Sci. **59**, 032108:1–032108:10 (2016)