Implicit Factorization of RSA Moduli Revisited (Short Paper)

Liqiang Peng[1,2,3], Lei Hu[1,2(✉)], Yao Lu[1,4], Zhangjie Huang[1,2], and Jun Xu[1,2]

[1] State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China
{pengliqiang,hulei,huangzhangjie,xujun}@iie.ac.cn
[2] Data Assurance and Communication Security Research Center, Chinese Academy of Sciences, Beijing, China
[3] University of Chinese Academy of Sciences, Beijing, China
[4] The University of Tokyo, Tokyo, Japan

Abstract. In this paper, we revisit the problem of factoring RSA moduli with implicit hint, where primes of two RSA moduli share some number of middle bits. Suppose that for two n-bit RSA moduli $N_1 = p_1q_1$ and $N_2 = p_2q_2$, q_1 and q_2 are (αn)-bit primes, p_1 and p_2 share tn bits at positions from t_1n to $t_2n = (t_1 + t)n$. Faugère et al. (PKC 2010) showed that when $t \geq 4\alpha$, one can factor N_1 and N_2 in polynomial time. In this paper, we improve this bound to $t > 4\alpha - 3\alpha^2$ by presenting a new method of solving a homogeneous linear equation modulo unknown divisors. Our method is verified by experiments.

Keywords: RSA modulus · Factorization with implicit hint · Coppersmith's technique · Middle bit

1 Introduction

How to efficiently factor integers which are composed of large primes is one of the most concern problems in algorithmic number theory. However, for now it does not exist any polynomial time algorithm. Therefore, many cryptosystems based on the difficulty of factorization problem are designed. Since its invention [18], the RSA public key cryptosystem is the most studied scheme in cryptology and has been widely used in practical applications due to its effective encryption and decryption. From the work of Coron and May [7], it has been proved that recovering the private key of the RSA cryptosystem and factoring the moduli are determinately equivalent in polynomial time.

However, there still exist many weaknesses in the RSA cryptosystem. For example, to achieve high efficiency in the decryption phase, small decryption exponents are often adopted and the security of such an RSA cryptosystem may be threatened by cryptanalysis such as small private exponent attack [4,20], small CRT-exponent attack [11] and so on. Moreover, the pseudo random number

K. Tanaka and Y. Suga (Eds.): IWSEC 2015, LNCS 9241, pp. 67–76, 2015.
DOI: 10.1007/978-3-319-22425-1_5

generators which are used in the key generation algorithm in the RSA cryptosystem may also threaten the security. Recently, Lenstra et al. [13] and Bernstein et al. [3] discovered this weakness and successfully factor some RSA moduli which are used in the real world. Hence, along this direction many researchers have paid many attentions to factoring RSA moduli with some specific hints.

Implicit Factorization. For the convenience of describing the problem of implicit factorization, we begin with a simple example. Assume that there are two n-bit RSA moduli $N_1 = p_1q_1$ and $N_2 = p_2q_2$, where q_1, q_2 are (αn)-bit prime integers.

In PKC 2009, May and Ritzenhofen [16] firstly proposed an efficient method to factor the RSA moduli if p_1 and p_2 share a large number of the least significant bits (LSBs). It has been rigorously proved in [16], if $tn \geq \alpha n + 3$, then (q_1, q_2) is the shortest vector in a related two-dimensional lattice. Once (q_1, q_2) is found by some lattice basis reduction method, the two RSA moduli are factored. May and Ritzenhofen also heuristically generalize their method to deal with implicit factorization of multiple RSA moduli.

Shortly later, Faugère et al. [8] analyzed the problem of implicit factorization where the primes share most significant bits (MSBs) or bits in the middle. According to Faugère et al.'s work, when p_1 and p_2 share $tn \geq 2\alpha n + 3$ MSBs, (q_1, q_2) can be found from a two-dimensional lattice. In the case of tn bits shared in the middle of the binary expressions of p_1 and p_2, they gave a heuristic bound that for the case of $tn \geq 4\alpha n + 7$, and q_1 and q_2 can be recovered from a three-dimensional lattice.

Related Works. Since the problem of implicit factorization has been proposed, it attracts a lot of attentions. Sarkar and Maitra [19] combined the implicit factorization and approximate integer common divisor problem, and by solving modular equations, they obtained the same bound of [8,16] for both LSBs case and MSBs case. Then Kurosawa and Ueda [12] reconsidered the method of [16] and gave a more tighter bound on the numbers of shared LSBs. In 2014, Peng et al. [17] and Lu et al. [15] used two different methods to improve the bound for both LSBs case and MSBs case. The intriguing point is that these two completely different methods obtained the same bounds on the numbers of shared LSBs or MSBs and it is worth to do further investigation to find the internal relations. However, all the above mentioned methods do not work for the case that the primes share middle bits.

Our Contribution. Recall the work of [17], Peng et al. firstly used a low dimensional lattice which is exactly considered in [8,16] to obtain a reduced basis, then they represented the desired vector as a linear equation of the reduced basis, they solved out the linear equation by using Coppersmith's technique, and finally obtain an improved bound.

In this paper, inspired by the idea of [17], for the first time we optimize the bound on the number of shared bits in the middle position. As it has been shown

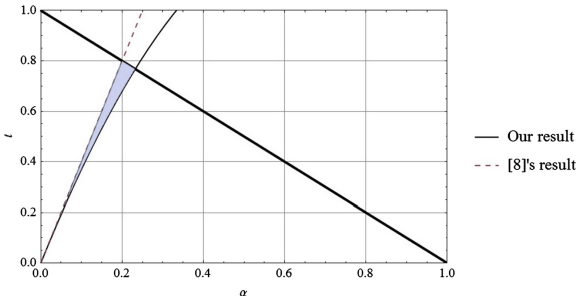

Fig. 1. Comparison with previous ranges on t with respect to α. Since $t \leq 1 - \alpha$, any valid range is under the thick solid diagonal line. Here the dotted line denotes the lower bound on t in [8] and the thin solid line denotes that in this paper. The grey shaded area is a new improvement presented in this paper.

in [8], if there are enough shared middle bits, the desired factorization can be directly obtained from the L^3 lattice basis reduction algorithm. We present a method to deal with the case where the shared middle bits are not enough to ensure that the desired factorization is included in the output of the L^3 algorithm. The starting point is that we represent the vector which we desire to find out as an integer linear combination of the reduced basis vectors of the lattice and obtain a modular equation system with three modular equations and three unknown variables. Then we transform the first two modular equations of the system to a modular equation by applying the Chinese remainder theorem and reduce one of the unknown variables by elimination with the last equation. Finally, we can obtain a homogeneous linear equation with two unknown variables modulo an unknown divisor of a known composite large integer. Once the small root of the modular equation has been solved out, the desired vectors can be recovered, which means the bound on the number of shared middle bits can be improved. Ignoring the small constant which is dependent on the bitlength n, the previous bound $t \geq 4\alpha$ can be improved to $t > 4\alpha - 3\alpha^2$. To the best of our knowledge, our lower bound on the number of the shared middle bits is the first improvement on the implicit factorization problem of middle bits case and experimental results also show this improvement.

An explicit description on our improvement is illustrated in Fig. 1.

The rest of this paper is organized as follows. Preliminaries on lattices are given in Sect. 2. In Sect. 3, we give a brief description of previous work of implicit factorization for middle bits case. Section 4 is our improvement and the experimental results. Finally, Sect. 5 is the conclusion.

2 Preliminaries

Consider the linear independent vectors $w_1, w_2, \cdots, w_k \in \mathbb{R}^n$. Then the lattice L spanned by $w_1, \cdots, w_k$ is the set of all integer linear combinations of $w_1, \cdots, w_k$. The number of vectors, namely k, is the dimension of L and the

vectors $w_1, \cdots, w_k$ is a basis of L. Any lattice of dimension larger than 1 has infinitely many bases.

For a lattice, calculating its shortest vector is known to be NP-hard problem under randomized reductions [2]. However, since the L^3 lattice basis reduction algorithm which can output an approximation of shortest vector in polynomial time has been introduced in [14], lattice becomes a fundamental tool to analyze the security of public key cryptosystem.

Lemma 1. *(L^3, [14]) Let L be a lattice of dimension k. Applying the L^3 algorithm to L, the output reduced basis vectors $v_1, \cdots, v_k$ satisfy that*

$$\|v_1\| \leq \|v_2\| \leq \cdots \leq \|v_i\| \leq 2^{\frac{k(k-i)}{4(k+1-i)}} det(L)^{\frac{1}{k+1-i}}, \text{for any } 1 \leq i \leq k.$$

In [6], a strategy which is usually called Coppersmith's technique has been discussed. It used lattice-based method to find small integer roots of modular equation with one variable, and of integer equation with two variables. In [10], Jochemsz and May extended the results and gave a general method to find roots of multivariate polynomials.

Given a polynomial $g(x_1, \cdots, x_k) = \sum_{i_1, \cdots, i_k} a_{i_1, \cdots, i_k} x_1^{i_1} \cdots x_k^{i_k}$, we define

$$\|g(x_1, \cdots, x_k)\|^2 = \sum_{i_1, \cdots, i_k} a_{i_1, \cdots, i_k}^2$$

The following lemma due to Howgrave-Graham's result [9] gives a sufficient condition under which modular roots are still satisfied for integer equations.

Lemma 2. *(Howgrave-Graham, [9]) Let $g(x_1, \cdots, x_k) \in \mathbb{Z}[x_1, \cdots, x_k]$ be an integer polynomial with at most w monomials. Suppose that*

1. $g(y_1, \cdots, y_k) \equiv 0 \pmod{p^m}$ for $|y_1| \leq X_1, \cdots, |y_k| \leq X_k$, and
2. $\|g(x_1 X_1, \cdots, x_k X_k)\| < \frac{p^m}{\sqrt{w}}$

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

Lattice based approaches of solving small roots of a modular or integer equation are first to construct a lattice from the polynomial of the equation, then by lattice basis reduction algorithm obtain new short lattice vectors which correspond to new polynomials with small norms and with the same roots as the original polynomial. These approaches usually rely on the following heuristic assumption.

Assumption 1. *Lattice based constructions always yield algebraically independent polynomials, and the common roots of these polynomials can be efficiently computed by using numerical or symbolic methods.*

Gaussian Heuristic. In [1] a claim states that with overwhelming probability, the minima $\lambda_i(\mathcal{L})$ of a random n-dimensional lattice $\mathcal{L}$ are all asymptotically close to the Gaussian heuristic, that is, for all $1 \leq i \leq n$,

$$\frac{\lambda_i(\mathcal{L})}{\det(\mathcal{L})^{\frac{1}{n}}} \approx \sqrt{\frac{n}{2\pi e}},$$

where the minima $\lambda_i(\mathcal{L})$ denotes the i-th minimum of lattice $\mathcal{L}$, which means it is the radius of the smallest zero-centered ball containing at least i linearly independent lattice vectors.

Note that for our attack, the low-dimensional lattice we constructed is not a random lattice, however, according to our practical experiments, the lengths of the vectors of the lattice basis outputted from the L^3 algorithm to that specific lattice are indeed asymptotically close to the Gaussian heuristic. Based on this observation on the lengths of our reduced basis vectors, we give the following attacks and we also do experiments to verify our attacks.

3 Previous Method of Factoring Two RSA Moduli with Implicitly Common Middle Bits

Let $N_1 = p_1q_1$ and $N_2 = p_2q_2$ be two given RSA moduli of n bits, where q_1 and q_2 are (αn)-bit primes and p_1 and p_2 are primes that share tn bits at position from t_1n to $t_2n = (t_1 + t)n$. For convenience, we write N_1 and N_2 as follows:

$$N_1 = p_1q_1 = (p_{1_2}2^{t_2n} + p2^{t_1n} + p_{1_0})q_1,$$
$$N_2 = p_2q_2 = (p_{2_2}2^{t_2n} + p2^{t_1n} + p_{2_0})q_2.$$

Then we reduce the equations by modulo 2^{t_2n}, one can obtain two equations with 5 unknown variables $p, p_{1_0}, p_{2_0}, q_1, q_2$:

$$N_1 \equiv (p2^{t_1} + p_{1_0})q_1 \bmod 2^{t_2n}$$
$$N_2 \equiv (p2^{t_1} + p_{2_0})q_2 \bmod 2^{t_2n}.$$

Faugère et al. transformed the problem of factoring N_1 and N_2 to finding short vectors of a three-dimensional lattice, more precisely, a lattice L defined by the row vectors of the following matrix

$$\begin{pmatrix} K & 0 & N_2 \\ 0 & K & -N_1 \\ 0 & 0 & 2^{t_2n} \end{pmatrix},$$

where $K = 2^{(\alpha+t_1)n}$.

Clearly, the vector $v = (q_1K, q_2K, r)$ with r being the unique remainder in $(-2^{t_2n-1}, 2^{t_2n-1}]$ of $q_1N_2 - q_2N_1$ modulo 2^{t_2n} is in L. Due to the work of [8], v is the shortest vector in L when

$$tn \geq 4\alpha n + 7.$$

Then one can obtain the primes (q_1, q_2) by a lattice basis reduction algorithm. Note that, for large n, we simplify the bound as $t \geq 4\alpha$ by ignoring the small constant $\frac{7}{n}$.

4 Our Improvement

In this section, we propose a method to deal with the failure case of $t < 4\alpha$ in the previous section and improve the lower bound on t.

Note that, when $t < 4\alpha$ the vector (q_1K, q_2K, r) is not the shortest vector of L, which means (q_1K, q_2K, r) is generally not included in the outputted basis of the L^3 algorithm.

To facilitate the description, we denote $\lambda_1 = (l_{11}, l_{12}, l_{13}), \lambda_2 = (l_{21}, l_{22}, l_{23})$ and $\lambda_3 = (l_{31}, l_{32}, l_{33})$ as the basis vectors of L_1 obtained from the L^3 algorithm. With overwhelming probability, the minima of a lattice are all asymptotically close to the Gaussian heuristic, hence we have that $\|\lambda_1\| \approx \|\lambda_2\| \approx \|\lambda_3\| \approx \det(L)^{\frac{1}{3}}$. Thus, the sizes of l_{ij} can be estimated as $\det(L)^{\frac{1}{3}} = 2^{\frac{2\alpha+3t_1+t}{3}n}$.

Write the vector (q_1K, q_2K, r) as a linear combination of λ_1, λ_2 and λ_3 with integral coefficients x_0, y_0, z_0, namely $(q_1K, q_2K, r) = x_0\lambda_1 + y_0\lambda_2 + z_0\lambda_3$. Moreover, the entry r is $q_1N_2 - q_2N_1 \bmod 2^{t_2n} = q_1q_2(p_{2_0} - p_{1_0}) \bmod 2^{t_2n}$ in $(-2^{t_2n-1}, 2^{t_2n-1}]$ and $|q_1q_2(p_{2_0} - p_{1_0})|$ is less than $2^{(2\alpha+t_1)n}$. Hence, when $t \geq 2\alpha$, we have that $r = q_1q_2(p_{2_0} - p_{1_0})$.

Then we get three modular equations modulo unknown prime numbers:

$$\begin{cases} x_0l_{11} + y_0l_{21} + z_0l_{31} = q_1K \equiv 0 \pmod{q_1}, \\ x_0l_{12} + y_0l_{22} + z_0l_{32} = q_2K \equiv 0 \pmod{q_2}, \\ x_0l_{13} + y_0l_{23} + z_0l_{33} = q_1q_2(p_{2_0} - p_{1_0}) \equiv 0 \pmod{q_1q_2} \end{cases} \tag{1}$$

Since $|l_{ij}| \approx 2^{\frac{2\alpha+3t_1+t}{3}n}$, the desired solutions of (1) can be estimated roughly by $x_0, y_0, z_0 \approx \frac{q_jK}{l_{ij}} \approx 2^{\frac{4\alpha-t}{3}n}$.

Using the Chinese remainder theorem, from the first two equations of (1) we get an equation with the form of

$$ax_0 + by_0 + cz_0 \equiv 0 \pmod{q_1q_2}, \tag{2}$$

where a is an integer satisfying $a \equiv l_{11} \pmod{N_1}$ and $a \equiv l_{12} \pmod{N_2}$, b is an integer satisfying $b \equiv l_{21} \pmod{N_1}$ and $b \equiv l_{22} \pmod{N_2}$ and c is an integer satisfying $c \equiv l_{31} \pmod{N_1}$ and $c \equiv l_{32} \pmod{N_2}$. Clearly, a, b and c can be calculated from $l_{11}, l_{12}, l_{21}, l_{22}, l_{31}, l_{32}, N_1$ and N_2 by the extended Euclidean algorithm.

Then we reduce the common variable z_0 from equation (2) and the third equation of (1) by elimination technique. Therefore, we can obtain a modular equation with the form of

$$a'x_0 + b'y_0 \equiv 0 \pmod{q_1q_2}, \tag{3}$$

In order to recover the integral coefficients x_0 and y_0, we construct a modular equation

$$f(x,y) = a'x + b'y \equiv 0 \,(\mathrm{mod}\ q_1q_2).$$

Since $\gcd(a', N_1N_2)$ is 1, or else we have found a factor of N_1N_2. Therefore, we can use $\widehat{f} = a'^{-1}f(x,y) \bmod N_1N_2$.

Then we select polynomials as follows:

$$g_k(x,y) = y^{m-k}\widehat{f}^k(x,y)(N_1N_2)^{\max\{s-k,0\}}, \text{for } k = 0, 1, \cdots, m,$$

where m and s are integers which will be chosen later. Below we let $s \leq m$ and $\sigma = \frac{s}{m} \in [0,1]$.

Obviously, all the above polynomials have the same roots which are desired integral coefficients (x_0, y_0) modulo $(q_1q_2)^s$ and the solutions can be roughly estimated by $|x_0| \simeq X(:= 2^{\frac{4\alpha-t}{3}n})$ and $|y_0| \simeq Y(:= 2^{\frac{4\alpha-t}{3}n})$, neglecting any small constant because N is relatively large.

Then we construct a matrix, whose row vectors are the coefficient vectors of $g_k(xX, yY)$ with respect to the monomials on x, y. It is easy to check that it is a triangular matrix, and its diagonal entries are

$$X^kY^{m-k}(N_1N_2)^{\max\{s-k,0\}}, \text{ for } k = 0, \cdots, m$$

Let the row vectors of this matrix span a lattice L_1.

By construction, its determinant is easily determined as

$$\det(L_1) = X^{S_x}Y^{S_y}(N_1N_2)^{S_N}$$

where the exponents S_x, S_y, S_N are calculated as follows:

$$\begin{aligned} S_x &= \sum_{k=0}^{m} k = \frac{1}{2}m^2 + o(m^2), \\ S_y &= \sum_{k=0}^{m}(m-k) = \frac{1}{2}m^2 + o(m^2), \\ S_N &= \sum_{k=0}^{s-1}(s-k) = \frac{\sigma^2}{2}m^2 + o(m^2). \end{aligned}$$

On the other hand, the dimension of L_1 is $\dim(L_1) = m + 1$. According to Lemmas 1 and 2, one can obtain polynomial equations which share the root (x_0, y_0) over integers if

$$\det(L_1)^{\frac{1}{\dim(L_1)}} < \gamma(q_1q_2)^s,$$

where γ is a small constant. Now, for large N_1 and N_2, the required condition can be reduced as $\det(L_1)^{\frac{1}{\dim(L_1)}} < (q_1q_2)^s$, namely,

$$X^{\frac{1}{2}m^2+o(m^2)}Y^{\frac{1}{2}m^2+o(m^2)}(N_1N_2)^{\frac{\sigma^2}{2}m^2+o(m^2)} < (q_1q_2)^{\sigma m^2+o(m^2)}$$

Table 1. For 1000-bit RSA moduli, theoretical and experimental results of middle bits problem

Bitsize of p_i, q_i $(1-\alpha)\log_2 N, \alpha\log_2 N$	Theo. of [8]	Theo. of ours	$\dim(L_1) = 21$		$\dim(L_1) = 41$	
			Expt	Time(sec)	Expt	Time(sec)
900,100	407	370	380	118.015	370	6732.652
850,150	607	533	560	196.863	540	10824.582
800,200	failed	680	710	294.561	690	15249.878

To obtain an asymptotic bound, we assume m goes to infinite and ignore the small terms $o(m^2)$. Putting the bounds X, Y into the above inequality, we obtain that

$$(\frac{4\alpha - t}{3}) \cdot \frac{1}{2} \cdot 2 + 2 \cdot \frac{\sigma^2}{2} < 2\alpha \cdot \sigma$$

For optimization, we let $\sigma = \alpha$, and finally we obtain the following bound on t:

$$t > 4\alpha - 3\alpha^2.$$

Then we can obtain several polynomial equations which share the root (x_0, y_0) over integers. Under Assumption 1, we can successfully collect the desired roots.

Experimental Results. We have implemented the experiment program in Magma 2.10 computer algebra system [5] on a PC with Intel(R) Core(TM) Duo CPU(2.53GHz, 1.9GB RAM Windows 7). In all experiments, we obtained several integer equations with desired roots (x_0, y_0) over $\mathbb{Z}$ and found that these equations had a common factor with the form of $ax + by$. In this situation, $ax_0 + by_0$ always equals to 0 and $\gcd(x_0, y_0)$ is small. Hence, the solution (x_0, y_0) can be solved out.

The following Table 1 lists some theoretical and experimental results on factoring two 1000-bit RSA moduli with shared middle bits.

Note that, in the case of $(800, 200)$, the theoretical bound of [8] is 807 bits larger than 800 bits, which means this case can not be found.

Extension to More RSA Mudoli. We heuristically generalize the above result from two RSA moduli to an arbitrary number of n-bit RSA moduli. By combining modulo equations and reducing common variables, we can similarly improve the previous bound of [8].

The key sketch of our method can be described as follows:

(1) For k RSA moduli, based on the work of [8], we firstly construct a $\frac{k(k+1)}{2}$-dimensional lattice. If the shared middle bits are not enough to ensure that the factorization is included in the output of the L^3 algorithm, we represent the desired vector as an integer linear combination of the reduced basis vectors of the lattice and obtain a modular equation system with $\frac{k(k+1)}{2}$ modular equations and $\frac{k(k+1)}{2}$ unknown variables.

(2) In this step, we reduce the unknown variables by elimination in order. At first, we can respectively obtain two homogeneous linear equations with $\frac{k(k+1)}{2}$ unknown variables modulo q_iq_j, for $1 \leq i, j \leq k$ and $i \neq j$ and reduce one of the unknown variables by elimination of these two equations.

Then we have an equation f_1 modulo q_iq_j. Note that, we can also obtain an equation f_2 modulo q_iq_l and an equation f_3 modulo q_jq_l, where $l = 1, \cdots, k$ and $l \neq i, j$. By applying the Chinese remainder theorem, we can obtain an equation modulo $q_iq_jq_l$ from f_1 and f_2, similarly we can obtain another equation modulo $q_iq_jq_l$ from f_1 and f_3. Then we can reduce one unknown variable and obtain a homogeneous linear equation modulo $q_iq_jq_l$.

(3) Based on this order, we can finally obtain a homogeneous linear equation modulo $q_1q_2 \cdots q_k$ and the number of unknown variables is

$$\frac{k(k+1)}{2} - 1 - 1 - 2 - \cdots - (k-2) = 2k - 2.$$

By solving this modular equation, we can obtain an improved bound.

5 Conclusion

In this paper, we revisited the problem of implicit factorization and we for the first time improved the bound of implicit factorization on the number of the middle bits that the primes share. Our method is to recover the coordinates of the expression of the desired vectors with respect to some reduced lattice basis. It is nice to see our theoretical bound and experimental results are both have an improvement on existing results.

Acknowledgements. The authors would like to thank anonymous reviewers for their helpful comments and suggestions. The work of this paper was supported by the National Key Basic Research Program of China (2013CB834203), the National Natural Science Foundation of China (Grants 61472417, 61402469, 61472416 and 61272478), the Strategic Priority Research Program of Chinese Academy of Sciences under Grant XDA06010702 and XDA06010703, and the State Key Laboratory of Information Security, Chinese Academy of Sciences.

References

1. Ajtai, M.: Generating random lattices according to the invariant distribution. Draft of March (2006)
2. Ajtai, M.: The shortest vector problem in L_2 is NP-hard for randomized reductions (extended abstract). In: Vitter, J.S. (ed.) STOC 1998. pp. 10–19. ACM (1998)
3. Bernstein, D.J., Chang, Y.-A., Cheng, C.-M., Chou, L.-P., Heninger, N., Lange, T., van Someren, N.: Factoring RSA keys from certified smart cards: Coppersmith in the wild. In: Sako, K., Sarkar, P. (eds.) ASIACRYPT 2013, Part II. LNCS, vol. 8270, pp. 341–360. Springer, Heidelberg (2013)
4. Boneh, D., Durfee, G.: Cryptanalysis of RSA with private key d less than $N^{0.292}$. IEEE Trans. Inf. Theor. **46**(4), 1339–1349 (2000)

5. Bosma, W., Cannon, J.J., Playoust, C.: The magma algebra system I: the user language. J. Symbolic Comput. **24**(3–4), 235–265 (1997)
6. Coppersmith, D.: Small solutions to polynomial equations, and low exponent RSA vulnerabilities. J. Crypt. **10**(4), 233–260 (1997)
7. Coron, J., May, A.: Deterministic polynomial-time equivalence of computing the RSA secret key and factoring. J. Crypt. **20**(1), 39–50 (2007)
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)
9. Howgrave-Graham, N.: Finding small roots of univariate modular equations revisited. In: Darnell, M.J. (ed.) Cryptography and Coding 1997. LNCS, vol. 1355, pp. 131–142. Springer, Heidelberg (1997)
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)
11. Jochemsz, E., May, A.: A polynomial time attack on RSA with private CRT-exponents smaller than N^0.073. In: Menezes, A. (ed.) CRYPTO 2007. LNCS, vol. 4622, pp. 395–411. Springer, Heidelberg (2007)
12. Kurosawa, K., Ueda, T.: How to factor N_1 and N_2 when $p_1 = p_2 \ mod \ 2^t$. In: Sakiyama, K., Terada, M. (eds.) IWSEC 2013. LNCS, vol. 8231, pp. 217–225. Springer, Heidelberg (2013)
13. Lenstra, A.K., Hughes, J.P., Augier, M., Bos, J.W., Kleinjung, T., Wachter, C.: Public keys. In: Safavi-Naini, R., Canetti, R. (eds.) CRYPTO 2012. LNCS, vol. 7417, pp. 626–642. Springer, Heidelberg (2012)
14. Lenstra, A.K., Lenstra, H.W., Lovász, L.: Factoring polynomials with rational coefficients. Math. Ann. **261**(4), 515–534 (1982)
15. Lu, Y., Peng, L., Zhang, R., Lin, D.: Towards optimal bounds for implicit factorization problem. IACR Crypt. ePrint Arch. **2014**, 825 (2014)
16. 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)
17. Peng, L., Hu, L., Xu, J., Huang, Z., Xie, Y.: Further improvement of factoring RSA moduli with implicit hint. In: Pointcheval, D., Vergnaud, D. (eds.) AFRICACRYPT. LNCS, vol. 8469, pp. 165–177. Springer, Heidelberg (2014)
18. Rivest, R.L., Shamir, A., Adleman, L.M.: A method for obtaining digital signatures and public-key cryptosystems (reprint). Commun. ACM **26**(1), 96–99 (1983)
19. Sarkar, S., Maitra, S.: Approximate integer common divisor problem relates to implicit factorization. IEEE Trans. Inf. Theor. **57**(6), 4002–4013 (2011)
20. Wiener, M.J.: Cryptanalysis of short RSA secret exponents. IEEE Trans. Inf. Theor. **36**(3), 553–558 (1990)