Further Improvement of Factoring RSA Moduli with Implicit Hint

Liqiang Peng[1,2], Lei Hu[1,2], Jun Xu[1,2], Zhangjie Huang[1,2], and Yonghong Xie[1,2]

[1] State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100093, China
[2] Data Assurance and Communication Security Research Center, Chinese Academy of Sciences, Beijing 100093, China
{lqpeng,hu,jxu,zjhuang,xyxie}@is.ac.cn

Abstract. We investigate the problem of factoring RSA moduli with implicit hint, which was firstly proposed by May and Ritzenhofen in 2009 where unknown prime factors of several RSA moduli shared some number of least significant bits (LSBs) and was considered by Faugère et al. in 2010 where some most significant bits (MSBs) were shared between the primes. In this paper, we further consider this factorization with implicit hint problem, present a method to deal with the case when the number of shared LSBs or MSBs is not large enough to satisfy the bound proposed by May et al. and Faugère et al. by making use of a result from Herrmann and May for solving linear equations modulo unknown divisors, and finally get a better lower bound on the the number of shared LSBs or MSBs. To the best of our knowledge, our lower bound is better than all known results and we can theoretically deal with the implicit factorization for the case of balanced RSA moduli.

Keywords: RSA modulus, factorization with implicit hint, Coppersmith's technique.

1 Introduction

Factoring large integers efficiently is a problem of most concern in algorithmic number theory and also in practical cryptographic applications since the RSA public key cryptosystem based on the factorization problem has been widely used. However, due to practical reasons, e.g., for achieving high implementation efficiency, specific RSA parameters 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 (Chinese-remainder-theorem-exponent) attack [12] and so on. Recently, Lenstra et al. [13] and Bernstein et al. [3] utilized the weakness of pseudo random number generators to successfully factor some RSA moduli which are used in the real world. Hence, the problem of factoring RSA moduli with some specific hint is worthy of investigation.

In the PKC'2009 conference, May and Ritzenhofen proposed an efficient method to factor RSA moduli with an implicit hint [16]. More precisely, for two

D. Pointcheval and D. Vergnaud (Eds.): AFRICACRYPT 2014, LNCS 8469, pp. 165–177, 2014.

n-bit RSA moduli $N_1 = p_1q_1$ and $N_2 = p_2q_2$ where p_1 and p_2 share tn least significant bits (LSBs) and q_1 and q_2 are (αn)-bit prime integers, it has been proved in [16] that if $tn \geq 2\alpha n+3$, then (q_1, q_2) is a shortest vector in a two-dimensional lattice and it can be found by a lattice basis reduction algorithm. Thus, the two RSA moduli can be factored. May et al. [16] also gave a heuristic generalization for the factorization of multiple RSA moduli $N_1 = p_1q_1, \cdots, N_k = p_kq_k$, where the number of shared LSBs, tn, is at least $\frac{k}{k-1}\alpha n$. Shortly later, Faugère et al. [7] made an extension analysis to deal with the case that $p_1, \cdots, p_k$ share most significant bits (MSBs) or bits in the middle.

In 2011, Sarkar and Maitra [19] transformed the factorization with implicit hint problem to the approximate integer common divisor problem [10,5], and lower bounds on the number of LSBs or MSBs required to be shared is improved in theory and experimentally [19]. Sarkar and Maitra used Coppersmith's lattice-based technique to find out the desired roots of modular equation, and the lower bound they obtained was improved to

$$\begin{cases} t > \max\{\alpha, \frac{\alpha k^2-(2\alpha+1)k+1+\sqrt{k^2+2\alpha^2k-\alpha^2k^2-2k+1}}{k^2-3k+2}\}, \text{for } k > 2, \\ t > 2\alpha - \alpha^2, \text{for } k = 2. \end{cases}$$

Based on this result, Lu et al. [15] modified the polynomials in the construction of the lattice and the bound was further improved as $1 - (1-\alpha)^{\frac{k}{k-1}}$.

In this paper, we firstly reconsider the problem of factoring RSA moduli with primes sharing LSBs, which has been discussed by May et al. [16]. As it has been shown in [16], if there are enough shared LSBs, 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 LSBs are not enough to ensure that the desired factorization is included in the output of the L^3 algorithm. The idea 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, then we transform the modular equation system to a modular equation with unknown modulus by applying the Chinese remainder theorem, and finally, we solve this modular equation by a method of Herrmann and May in [8]. Note that, our method does not require the constraint that $t \geq \alpha$ in [16,7,19,15], which means for multiple RSA moduli we can for the first time theoretically deal with the implicit factorization for the case of balanced RSA moduli (i.e., p_i and q_i have the same bitlength). The factorization of RSA moduli with primes sharing MSBs is also revisited in this paper.

Table 1 lists a comparison of our result with the previous results in [16], [7], [19] and [15], where

$$F(\alpha, k) = \begin{cases} \frac{\alpha k^2-(2\alpha+1)k+1+\sqrt{k^2+2\alpha^2k-\alpha^2k^2-2k+1}}{k^2-3k+2}, \text{for } k > 2, \\ 2\alpha - \alpha^2, \text{for } k = 2, \end{cases}$$

$$G(\alpha, k) = \frac{k}{k-1}(\alpha - 1 + (1-\alpha)^{\frac{k+1}{k}} + (k+1)(1-(1-\alpha)^{\frac{1}{k}})(1-\alpha)),$$

Table 1. Comparison with existing results on t

	[16]	[7]	[19]	[15]	this paper
LSB	$\frac{k}{k-1}\alpha$	-	$F(a,k)$	$1-(1-\alpha)^{\frac{k}{k-1}}$	$G(\alpha,k)$
MSB	-	$\frac{k}{k-1}\alpha+\frac{6}{n}$	$F(a,k)$	$1-(1-\alpha)^{\frac{k}{k-1}}$	$G(\alpha,k)$

and the curves of $G(\alpha,k)$ and $1-(1-\alpha)^{\frac{k}{k-1}}$ as functions on α can be seen in Figures 1 and 2 in Sections 3 and 4 which show $G(\alpha,k) < 1-(1-\alpha)^{\frac{k}{k-1}}$. To the best of our knowledge, our lower bound on the number of the shared bits is theoretically better than all known results and experimental results also show this improvement.

2 Preliminaries

Let $w_1, w_2, \cdots, w_k$ be k linearly independent vectors in $\mathbb{R}^n$. They span a k-dimensional lattice L which is the set of all integer linear combinations, $c_1w_1 + \cdots + c_kw_k$, of $w_1, \cdots, w_k$, where $c_1, \cdots, c_k \in \mathbb{Z}$. The vectors $w_1, \cdots, w_k$ form a basis of the lattice L. Any lattice of dimension larger than 1 has infinitely many bases [18].

Calculating the shortest vectors in a lattice is known to be an NP-hard problem under randomized reductions [2]. However, some approximations of shortest vectors in a lattice can be found out in polynomial time and the famous L^3 lattice basis reduction algorithm is invented thirty years ago for attending such a goal [14,18], and since then lattice becomes a fundamental tool to analyze the security of public key cryptosystems.

Lemma 1. *(L^3, [14,18]) Let L be a lattice of dimension k. Applying the L^3 algorithm to L, the outputted reduced basis vectors $v_1, \cdots, v_k$ satisfy that*

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

Lattices are used to find small roots of univariate modular equations and bivariate equations [6], and this strategy is now usually called Coppersmith's technique. In [11], Jochemsz and May extended the technique and gave a general result to find roots of multivariate polynomials.

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

$$\|g(x_1, \cdots, x_k)\| = \Big(\sum_{(i_1,\cdots,i_k)} a_{i_1,\cdots,i_k}^2 \Big)^{1/2}.$$

The following lemma due to Howgrave-Graham [9] gives a sufficient condition under which roots of a modular equation also satisfy an integer equation.

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*

$$1. g(y_1, \cdots, y_k) \equiv 0 \pmod{p^m} \text{ for } |y_1| \leq X_1, \cdots, |y_k| \leq X_k, \text{ and}$$
$$2. \|g(x_1X_1, \cdots, x_kX_k)\| < \frac{p^m}{\sqrt{w}}$$

Then $g(y_1, \cdots, y_k) = 0$ *holds over 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. *The common roots of the polynomials yielded by lattice based constructions can be efficiently computed by using numerical method, symbolic method or exploiting the special structure of these polynomials.*

In our analysis, we will use the following theorem proposed by Herrmann and May in [8]. Based on Coppersmith's technique, they gave upper bounds on the size of solutions of a bivariate linear equation modulo an unknown divisor of a known composite integer.

Theorem 1. *(Herrmann and May, [8]) Let* $\epsilon > 0$, N *be a sufficiently large composite integer with an unknown divisor* $p \geq N^\beta$, *and* $f(x_1, x_2) \in \mathbb{Z}[x_1, x_2]$ *be a bivariate linear polynomial. Under Assumption 1, one can find all solutions* (y_1, y_2) *of the equation* $f(x_1, x_2) = 0 \pmod{p}$ *with* $|y_1| \leq N^\gamma$ *and* $|y_2| \leq N^\delta$ *if*

$$\gamma + \delta \leq 3\beta - 2 + 2(1-\beta)^{\frac{3}{2}} - \epsilon. \tag{1}$$

The above theorem 1 has been extended to a modular equation with $k \geq 3$ variables [8].

Theorem 2. *(Herrmann and May, [8]) Let* $\epsilon > 0$, N *be a sufficiently large composite integer with an unknown divisor* $p \geq N^\beta$, $f(x_1, \cdots, x_k) \in \mathbb{Z}[x_1, \cdots, x_k]$ *be a monic linear polynomial in* k *variables. Under Assumption 1, one can find all solutions* $(y_1, \cdots, y_k)$ *of the equation* $f(x_1, \cdots, x_k) = 0 \pmod{p}$ *with* $|y_1| \leq N^{\gamma_1}, \cdots, |y_k| \leq N^{\gamma_k}$ *if*

$$\sum_{i=1}^{k} \gamma_i \leq 1 - (1-\beta)^{\frac{k+1}{k}} - (k+1)(1 - \sqrt[k]{1-\beta})(1-\beta) - \epsilon. \tag{2}$$

More details about the theorems can be referred to [8]. Note that, in our experiments, the equations obtained by calculation of the resultant or finding a Gröbner basis are not univariate polynomials, however we can exploit the structure of these polynomials to solve out the desired small roots.

3 Factoring Two RSA Moduli with Implicitly Common LSBs

Recall in the implicit factoring of two RSA moduli in [16], there are two different n-bit RSA moduli $N_1 = p_1q_1$ and $N_2 = p_2q_2$, where p_1 and p_2 satisfy that $p_1 \equiv p_2 (\mathrm{mod}\, 2^{tn})$, where $0 < t < \log_{N_i} p_i$ for $i = 1, 2$.

Since $p_1 \equiv p_2 (\mathrm{mod}\, 2^{tn})$, we let $p_1 = p + 2^{tn}\widetilde{p_1}$ and $p_2 = p + 2^{tn}\widetilde{p_2}$. We have

$$\begin{aligned}(p + 2^{tn}\widetilde{p_1})q_1 &= N_1,\\ (p + 2^{tn}\widetilde{p_2})q_2 &= N_2,\end{aligned}$$

which means

$$\begin{aligned}pq_1 &= N_1 \ (\mathrm{mod}\ 2^{tn}),\\ pq_2 &= N_2 \ (\mathrm{mod}\ 2^{tn}).\end{aligned}$$

Moreover, we get the following linear equation

$$(N_1^{-1}N_2)q_1 - q_2 \equiv 0 \ (\mathrm{mod}\, 2^{tn}), \tag{3}$$

where N_1^{-1} is the inverse of N_1 modulo 2^{tn}.

In [16], the authors have proved that the vector (q_1, q_2) is the shortest vector of the two-dimensional lattice L_1 generated by the row vectors of the following matrix

$$\begin{pmatrix} 1 & N_1^{-1}N_2 \\ 0 & 2^{tn} \end{pmatrix} \tag{4}$$

when q_1 and q_2 are both (αn)-bit numbers and $tn > 2(\alpha n + 1)$, where $\alpha \approx 1 - \log_{N_i} p_i$ for $i = 1, 2$. Note that $t < \log_{N_i} p_i \approx 1 - \alpha$. Once q_1 and q_2 are obtained by the L^3 algorithm in polynomial time, N_1 and N_2 are factored.

However, when $tn \leq 2(\alpha n + 1)$ the vector (q_1, q_2) is not the shortest vector of L_1, which means (q_1, q_2) is generally not included in the outputted basis (λ_1, λ_2) of the L^3 algorithm. Write the vector (q_1, q_2) as a linear combination of λ_1 and λ_2. Below we present a method to find out the linear combination by solving linear equations modulo unknown RSA factors. Once the linear combination is found, a better bound on t than that in [16] is obtained.

Let $\lambda_1 = (l_{11}, l_{12})$ and $\lambda_2 = (l_{21}, l_{22})$ be the basis vectors of L_1 obtained from the L^3 algorithm. Then we have a rough estimation on the l_{ij}, with overwhelming probability, the minima of a lattice are all asymptotically close to the Gaussian heuristic [1], hence we have $\|\lambda_1\| \approx \|\lambda_2\| \approx \sqrt{\frac{2}{2\pi e}}\det(L)^{\frac{1}{2}}$. Thus, the sizes of $l_{11}, l_{12}, l_{21}, l_{22}$ can be estimated from $\det(L_1)^{\frac{1}{2}} = 2^{\frac{tn}{2}}$.

Let (q_1, q_2) be represented as $(q_1, q_2) = x_1\lambda_1 + x_2\lambda_2$ with integral coefficients x_1 and x_2. Then we get two modular equations modulo unknown prime numbers

$$\begin{cases} x_1 l_{11} + x_2 l_{21} = q_1 \equiv 0 \ (\mathrm{mod}\ q_1), \\ x_1 l_{12} + x_2 l_{22} = q_2 \equiv 0 \ (\mathrm{mod}\ q_2). \end{cases} \tag{5}$$

Since $l_{11}, l_{12}, l_{21}, l_{22}$ have roughly the same size, the desired coefficients x_1 and x_2 can be roughly estimated as $\frac{q_j}{2l_{ij}}$ for any i and j.

Using the Chinese remainder theorem, from (5) we get an equation with the form of

$$ax_1 + bx_2 \equiv 0 \,(\mathrm{mod}\ q_1 q_2), \tag{6}$$

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

Since $q_1 \approx q_2 \approx 2^{\alpha n}$, we have $q_1 q_2 \approx (N_1 N_2)^{\alpha}$. By Theorem 1, we can find all solutions (y_1, y_2) of equation (6) with $|y_1| \leq (N_1 N_2)^{\delta_1} \approx 2^{2\delta_1 n}$ and $|y_2| \leq (N_1 N_2)^{\delta_2} \approx 2^{2\delta_2 n}$ if

$$\delta_1 + \delta_2 \leq 3\alpha - 2 + 2(1-\alpha)^{\frac{3}{2}} - \epsilon.$$

When $\delta_1 \approx \delta_2$, we have

$$2\delta_1 \approx 2\delta_2 \leq 3\alpha - 2 + 2(1-\alpha)^{\frac{3}{2}} - \epsilon. \tag{7}$$

From (5), there is a good possibility that the desired solution of (5) can be estimated with $\frac{q_1}{2l_{11}} \approx 2^{(\alpha - \frac{t}{2})n}$. Hence, when

$$\alpha - \frac{t}{2} \leq 3\alpha - 2 + 2(1-\alpha)^{\frac{3}{2}} - \epsilon,$$

or equivalently,

$$t \geq 4 - 4\alpha - 4(1-\alpha)^{\frac{3}{2}} + \epsilon,$$

the desired solution can be solved out.

Comparing with the works of [16], [19] and [15], we can get the following Figure 1.

Experimental Results:
We have implemented the experiment program in Sage 5.12 computer algebra system on a PC with Intel(R) Core(TM) Duo CPU(2.53GHz, 1.9GB RAM ubuntu 13.10) and carried out the L^2 algorithm [17]. In all experiments, we obtained several integer equations with desired roots (y_1, y_2) over $\mathbb{Z}$ and found that these equations had a common factor with the form of $ax_1 + bx_2$. In these situations, $ay_1 + by_2$ always equals to 0 and $\gcd(y_1, y_2)$ is small. Hence, the solution (y_1, y_2) can be solved out.

The following Table 2 lists some theoretical and experimental results on factoring two 1024-bit RSA moduli with shared LSBs.

4 Extending to Factoring Multiple RSA Moduli with Implicitly Common LSBs

In the case of multiple RSA moduli with implicit common LSBs, let $N_i = p_i q_i$, $i = 1, 2, \cdots, k$, be k different n-bit RSA moduli and p_i share tn least significant

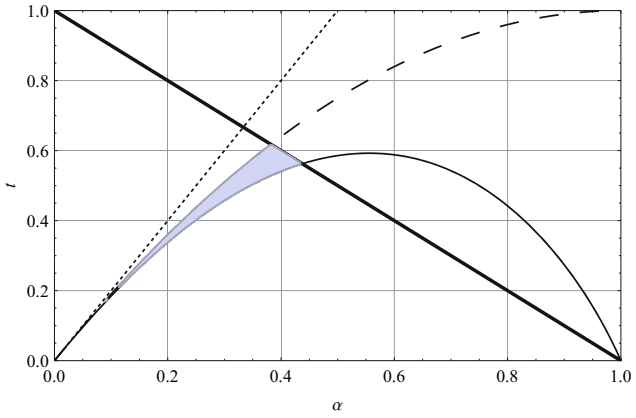


Fig. 1. Comparison with previous ranges on t with respect to α. Since $t \leq 1 - \alpha$, any valid range is under the thick solid diagonal line. Here the dotted line denotes the lower bound on t in [16], the dashed line denotes that in [19] and [15], and the thin solid line denotes that in this paper. The grey shaded area is a new improvement presented in this paper.

Table 2. Theoretical and experimental results of factoring 1024-bit RSA moduli with LSBs. Here dim denotes the dimension of the lattice.

k	bitsize of (p_i, q_i), i.e., $((1-\alpha)\log_2 N_i, \alpha\log_2 N_i)$	no. of shared LSBs in p_i ([19])				no. of shared LSBs in p_i (this paper)			
		theo.	expt.	dim	time (sec)	theo.	expt.	dim	time (sec)
2	(874, 150)	278	–	–	–	267	278	190	1880.10
2	(824, 200)	361	–	–	–	340	357	190	1899.21
2	(774, 250)	439	–	–	–	405	412	190	2814.84
2	(724, 300)	513	–	–	–	461	470	190	2964.74

bits. Let q_i be of (αn)-bit. Write the moduli as

$$N_1 = (p + 2^{tn}\widetilde{p_1})q_1,$$
$$\cdots$$
$$N_k = (p + 2^{tn}\widetilde{p_k})q_k.$$

Then,

$$N_1 \equiv pq_1 \pmod{2^{tn}},$$
$$\cdots$$
$$N_k \equiv pq_k \pmod{2^{tn}}.$$

Similarly as in the analysis in the previous section for the case $k = 2$, we have $\frac{N_1}{q_1} \equiv \frac{N_i}{q_i} \pmod{2^{tn}}$, for $i = 2, 3, \cdots, k$. Since the modular equation $N_1^{-1}N_i q_1 - q_i \equiv 0 \pmod{2^{tn}}$ holds, we get a vector $(q_1, q_2, \cdots, q_k)$ in a k-dimensional lattice

L_2 which is generated by the row vectors of the following matrix

$$\begin{pmatrix} 1 & N_1^{-1}N_2 & N_1^{-1}N_3 & \cdots & N_1^{-1}N_k \\ 0 & 2^{tn} & 0 & \cdots & 0 \\ 0 & 0 & 2^{tn} & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 2^{tn} \end{pmatrix}.$$

In [16], it is proved that when $t \geq \frac{k}{k-1}\alpha$, with a good possibility $(q_1, q_2, \cdots, q_k)$ is a shortest vector in L_2 and it can be found out by applying the L^3 algorithm to L_2. However, when $t < \frac{k}{k-1}\alpha$, $(q_1, q_2, \cdots, q_k)$ is not included in the L^3 reduced basis $\{\lambda_1, \cdots, \lambda_k\}$ of L_2. Similarly as in the previous section, we represent $(q_1, q_2, \cdots, q_k)$ as a linear combination of $\lambda_1, \cdots, \lambda_k$, i.e., $(q_1, \cdots, q_k) = x_1\lambda_1 + x_2\lambda_2 + \cdots + x_k\lambda_k$, where $x_1, \cdots, x_k$ are integers.

Hence, we have the following modular equation system

$$\begin{cases} x_1 l_{11} + x_2 l_{21} + \cdots + x_k l_{k1} = q_1 \equiv 0 \pmod{q_1}, \\ \cdots \\ x_1 l_{1k} + x_2 l_{2k} + \cdots + x_k l_{kk} = q_k \equiv 0 \pmod{q_k}, \end{cases} \tag{8}$$

where $\lambda_i = (l_{i1}, l_{i2}, \cdots, l_{ik})$, $i = 1, 2, \cdots, k$.

The lengths of the output vectors of the L^3 algorithm can be estimated based on the Gaussian heuristic and experimental experience of the L^3 algorithm. We roughly estimate the sizes of $|\lambda_1|, \cdots, |\lambda_k|$ and the entries of λ_i as $\det(L_2)^{\frac{1}{k}} = 2^{\frac{nt(k-1)}{k}}$ and the solution of (8) as $|x_i| \approx \frac{q_i}{kl_{ij}} \approx 2^{\alpha n - \frac{nt(k-1)}{k} - \log_2 k} \leq 2^{\alpha n - \frac{nt(k-1)}{k}}$.

Similarly as in the previous section, we can obtain an equation with the form of

$$a_1x_1 + a_2x_2 + \cdots + a_kx_k \equiv 0 \pmod{q_1q_2\cdots q_k} \tag{9}$$

from equation system (8) by using the Chinese remainder theorem, where a_i is an integer satisfying $a_i \equiv l_{ij} \pmod{N_j}$ for $1 \leq j \leq k$ and it can be calculated from the l_{ij} and N_j.

For this linear polynomial equation in k variables modulo the unknown integer $q_1q_2\cdots q_k \approx (N_1N_2\cdots N_k)^\alpha$, by Theorem 2 the variables with $|x_i| \leq (N_1N_2\cdots N_k)^{\delta_i} \approx 2^{k\delta_i n}$, $i = 1, 2, \cdots, k$, can be solved out if

$$\sum_{i=1}^{k} \delta_i \leq 1 - (1-\alpha)^{\frac{k+1}{k}} - (k+1)(1 - \sqrt[k]{1-\alpha})(1-\alpha) - \epsilon,$$

or equivalently,

$$k\delta_i \leq 1 - (1-\alpha)^{\frac{k+1}{k}} - (k+1)(1 - \sqrt[k]{1-\alpha})(1-\alpha) - \epsilon$$

when $\delta_1 \approx \delta_2 \approx \cdots \approx \delta_k$.

Hence, when

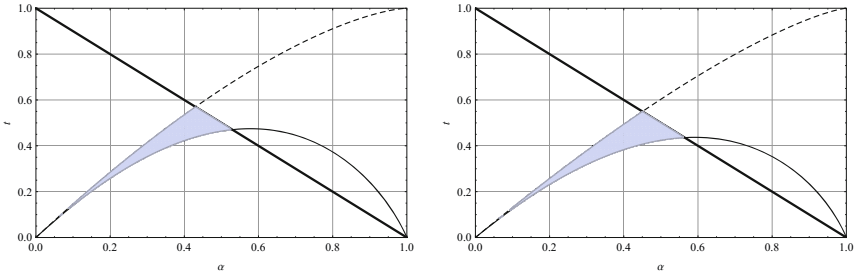
$$\alpha - \frac{t(k-1)}{k} \leq 1 - (1-\alpha)^{\frac{k+1}{k}} - (k+1)(1 - \sqrt[k]{1-\alpha})(1-\alpha) - \epsilon,$$

or namely,

$$t \geq \frac{k}{k-1}(\alpha - 1 + (1-\alpha)^{\frac{k+1}{k}} + (k+1)(1-(1-\alpha)^{\frac{1}{k}})(1-\alpha)) + \epsilon, \qquad (10)$$

the desired solution can be solved out.

To the best of our knowledge, the previous best theoretical bound on t is given in [15]: $t \geq 1-(1-\alpha)^{\frac{k}{k-1}}$. We make a comparison between our theoretical bound (10) and this bound, see Figure 2 for the cases of $k = 3$ and $k = 4$. We shall note that when $k \geq 3$, there exists t satisfying $t \leq 1-\alpha$ and the inequality (10), which removes the requirement that $t \geq \alpha$ in [16,7,19,15] and means for multiple RSA moduli we can for the first time theoretically deal with the implicit factorization for the case of balanced RSA moduli (i.e., p_i and q_i have the same bitlength and $\alpha = \frac{1}{2}$).



(a) $k = 3$ (b) $k = 4$

Fig. 2. The comparison between the bound (10) and the known best bound in [15]. As in Figure 1, any valid range is under the thick solid diagonal line. Here the dashed line denotes the lower bound on t in [15], the thin solid line denotes that in this paper, the grey shaded area on the figure is a new improvement presented in this paper.

Experimental Results:
We have implemented the program in Sage 5.12 computer algebra system on a PC with Intel(R) Core(TM) Duo CPU (2.53GHz, 1.9GB RAM ubuntu 13.10).

In all experiments for the case $k = 3$ and 1000-bit RSA moduli, we obtained several integer equations with desired roots (y_1, y_2, y_3) over $\mathbb{Z}$. To find out the roots, we used the technique of calculation of resultants and we always obtained a homogeneous equation of the form of $c_1x_2^4+c_2x_2^3x_3+c_3x_2^2x_3^2+c_4x_2x_3^3+c_5x_3^4 = 0$ which has the desired roots. Then we transformed these homogeneous bivariate equations to univariate equations over $\mathbb{Q}$ and obtained the ratio of $\frac{y_2}{y_3}$ by solving univariate equations. Similarly as in the experiments in the previous section, the common divisor of the desired roots is always small, hence we can obtain the desired roots (y_1, y_2, y_3). See Table 3 for the comparison with the previous bounds on t.

Table 3. For 1000-bit RSA moduli, theoretical and experimental bounds on t

k	bitsize of q_i	[16]		[19]		this paper	
		theo.	expt.	theo.	expt.	theo.	expt.
3	250	375	378	352	367	309	350
3	300	450	452	416	431	354	420
3	350	525	527	478	499	392	440
3	400	600	–	539	562	423	480

We notice that, when k is increasing, the lower bound on t will decrease, however, the dimension of the lattice constructed for solving the roots of the polynomials will be also increase. Due to the restriction of our computing ability, it is hard to evaluate the experimental results for larger k.

5 Factoring RSA Moduli with Implicitly Common MSBs

In [7], Faugère et al. extended May et al.'s results to factoring RSA moduli with primes implicitly sharing most significant bits (MSBs). Below we briefly recall Faugère et al.'s work.

Given two n-bit RSA moduli, $N_1 = p_1q_1$ and $N_2 = p_2q_2$, where q_1 and q_2 are (αn)-bit primes and p_1 and p_2 share tn MSBs, namely $|p_1 - p_2| \leq 2^{n-\alpha n-tn+1}$.

Consider the two-dimensional lattice L_3 which is generated by the row vectors of the following matrix

$$M_3 = \begin{pmatrix} K & 0 & N_2 \\ 0 & K & -N_1 \end{pmatrix}$$

where $K = \lfloor 2^{n-tn+\frac{1}{2}} \rfloor$. It has been proved in [7] that when $tn \geq 2\alpha n + 3$, or for simplicity $t \geq 2\alpha$ for efficiently large n, the vector $(q_1K, q_2K, q_1q_2(p_2-p_1))$ is the shortest vector in L_3. Similarly, when $t \leq 2\alpha$ the vector $(q_1K, q_2K, q_1q_2(p_2-p_1))$ that we wanted is not the shortest vector of L_3 and q_1 and q_2 can not be obtained directly from the basis vectors λ_1 and λ_2 of L_3 which are outputted by applying the L^3 algorithm.

In order to enable our result succinct, we make a rough estimation on the sizes of $\lambda_1 = (l_{11}, l_{12}, l_{13})$ and $\lambda_2 = (l_{21}, l_{22}, l_{23})$ and their entries. Since

$$\det(L_3) = \det(M_3 M_3^T) = K\sqrt{N_1^2 + N_2^2 + K^2} \approx 2^{2n-tn+1},$$

the length of $|\lambda_1|$ and $|\lambda_2|$ can be estimated as $\det(L_3)^{\frac{1}{2}} \approx 2^{n-\frac{tn}{2}+\frac{1}{2}}$, hence the entries can be bounded as $|l_{ij}| \approx 2^{n-\frac{tn}{2}}, i = 1, 2, j = 1, 2, 3$.

Since $(q_1K, q_2K, q_1q_2(p_2 - p_1)) \in L_3$, there exist integers x_1 and x_2 such that $(q_1K, q_2K, q_1q_2(p_2 - p_1)) = x_1\lambda_1 + x_2\lambda_2$. Hence, we obtain a modular equation system

$$\begin{cases} x_1l_{11} + x_2l_{21} = q_1K \equiv 0 \pmod{q_1}, \\ x_1l_{12} + x_2l_{22} = q_2K \equiv 0 \pmod{q_2}. \end{cases} \tag{11}$$

Since $|l_{ij}| \approx 2^{n-\frac{tn}{2}}$, the solutions to (11) can be estimated roughly by $x_i \approx \frac{q_j K}{2l_{ij}} \approx 2^{\alpha n+n-tn-n+\frac{tn}{2}} \approx 2^{\alpha n-\frac{tn}{2}}$.

Using the Chinese remainder theorem, from (11) we get a modular equation with the form of

$$ax_1 + bx_2 \equiv 0 \pmod{q_1 q_2}. \tag{12}$$

On the other hand, since $q_1 q_2 \approx (N_1 N_2)^\alpha$, from Theorem 1 the solution of (12) with $|y_1| < (N_1 N_2)^{\delta_1} \approx 2^{2\delta_1 n}$ and $|y_2| < (N_1 N_2)^{\delta_2} \approx 2^{2\delta_2 n}$ can be found if

$$\delta_1 + \delta_2 \leq 3\alpha - 2 + 2(1-\alpha)^{\frac{3}{2}} - \epsilon.$$

With $\delta_1 \approx \delta_2$, we have

$$2\delta_1 \approx 2\delta_2 \leq 3\alpha - 2 + 2(1-\alpha)^{\frac{3}{2}} - \epsilon.$$

Hence, when

$$\alpha - \frac{t}{2} \leq 2\delta_1 \leq 3\alpha - 2 + 2(1-\alpha)^{\frac{3}{2}} - \epsilon,$$

or equivalently,

$$t \geq 4 - 4\alpha - 4(1-\alpha)^{\frac{3}{2}} + \epsilon,$$

the desired solution can be solved out.

The above method can be extended to factoring multiple RSA moduli with primes implicitly sharing MSBs. In a similar way, we can prove that one can factor k RSA moduli with primes implicitly sharing (tn)-bit MSBs if

$$t \geq \frac{k}{k-1}(\alpha - 1 + (1-\alpha)^{\frac{k+1}{k}} + (k+1)(1-(1-\alpha)^{\frac{1}{k}})(1-\alpha)) + \epsilon.$$

To illustrate our optimization on the lower bounds on t, we list in Table 4 some numerical values for comparison with the results in [16], [7], [19] and [15]. It can be seen that our improvement with previous results increases as α increases.

Table 4. Comparison with previous results on the theoretical bounds on t

k	α	[16]([7])	[19]	[15]	this paper	α	[16]([7])	[19]	[15]	this paper
5	0.20	0.2500	0.2437	0.2434	0.2182	0.30	0.3750	0.3606	0.3597	0.3012
5	0.40	0.5000	0.4740	0.4719	0.3642	0.45	0.5625	0.5292	0.5264	0.3874
5	0.50	–	–	–	0.4045	–	–	–	–	–
10	0.20	0.2222	0.2197	0.2196	0.1962	0.30	0.3333	0.3276	0.3272	0.2725
10	0.40	0.4444	0.4341	0.4331	0.3320	0.45	0.5000	0.4868	0.4853	0.3546
10	0.50	–	–	–	0.3720	–	–	–	–	–
50	0.20	0.2041	0.2037	0.2036	0.1818	0.30	0.3061	0.3052	0.3051	0.2539
50	0.40	0.4082	0.4064	0.4062	0.3112	0.45	0.4592	0.4570	0.4567	0.3335
50	0.50	–	–	–	0.3512	–	–	–	–	–

Experimental Results:
We implemented our analysis in Sage 5.12 computer algebra system on a PC with Intel(R) Core(TM) Duo CPU(2.53GHz, 1.9GB RAM ubuntu 13.10). We present some numerical values for comparison with [19] in Table 5.

Table 5. For 1024-bit RSA moduli, theoretical and experimental results on factoring RSA moduli with implicitly common MSBs

k	bitsize of (p_i, q_i)	no. of shared MSBs in p_i ([19])				no. of shared MSBs in p_i (this paper)			
	$((1-\alpha)\log_2 N_i, \alpha\log_2 N_i)$	theo.	expt.	dim	time (sec)	theo.	expt.	dim	time (sec)
2	(874,150)	278	289	16	1.38	267	278	190	1974.34
2	(824,200)	361	372	16	1.51	340	358	190	2030.92
2	(774,250)	439	453	16	1.78	405	415	190	2940.35
2	(724,300)	513	527	16	2.14	461	474	190	3105.79
3	(874,150)	217	230	56	29.24	203	225	220	5770.99
3	(824,200)	286	304	56	36.28	260	288	220	6719.03
3	(774,250)	352	375	56	51.04	311	343	220	6773.48
3	(724,300)	417	441	56	70.55	356	395	220	7510.86
3	(674,350)	480	505	56	87.18	395	442	220	8403.91
3	(624,400)	540	569	56	117.14	428	483	220	9244.42

6 Conclusion

In this paper, we presented a further method for factoring RSA moduli with implicitly common LSBs or MSBs, and got a more lower bound on the number of the bits shared by the unknown primes of the RSA moduli. Our improvement can deal with some situations where the number of shared LSBs or MSBs does not satisfy the lower bounds proposed by May and Ritzenhofen in [16] and Faugère et al. in [7]. It is nice to see our theoretical bound and experimental results 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 (Grant 61070172), the Strategic Priority Research Program of Chinese Academy of Sciences under Grant XDA06010702, 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. In: Proceedings of the Thirtieth Annual ACM Symposium on Theory of Computing, 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 Transactions on Information Theory 46(4), 1339–1349 (2000)

5. Cohn, H., Heninger, N.: Approximate common divisors via lattices. arXiv preprint arXiv:1108.2714 (2011)
6. Coppersmith, D.: Small solutions to polynomial equations, and low exponent RSA vulnerabilities. Journal of Cryptology 10(4), 233–260 (1997)
7. 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)
8. Herrmann, M., May, A.: Solving linear equations modulo divisors: On factoring given any bits. In: Pieprzyk, J. (ed.) ASIACRYPT 2008. LNCS, vol. 5350, pp. 406–424. Springer, Heidelberg (2008)
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)
10. Howgrave-Graham, N.: Approximate integer common divisors. In: Silverman, J.H. (ed.) CaLC 2001. LNCS, vol. 2146, pp. 51–66. Springer, Heidelberg (2001)
11. 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)
12. 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)
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. Mathematische Annalen 261(4), 515–534 (1982)
15. Lu, Y., Zhang, R., Lin, D.: Improved bounds for the implicit factorization problem. Advances in Mathematics of Communications 7(3), 243–251 (2013)
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. Nguên, P.Q., Stehlé, D.: Floating-point LLL revisited. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 215–233. Springer, Heidelberg (2005)
18. Nguyen, P.Q., Valle, B.: The LLL algorithm: Survey and applications. Springer Publishing Company, Incorporated (2009)
19. Sarkar, S., Maitra, S.: Approximate integer common divisor problem relates to implicit factorization. IEEE Transactions on Information Theory 57(6), 4002–4013 (2011)
20. Wiener, M.J.: Cryptanalysis of short RSA secret exponents. IEEE Transactions on Information Theory 36(3), 553–558 (1990)