# Improved Results on Factoring General RSA Moduli with Known Bits

Mengce Zheng

University of Science and Technology of China, China
mengce.zheng@gmail.com

**Abstract.** We revisit the *factoring with known bits problem* on general RSA moduli in the forms of $N = p^r q^s$ for $r, s \geq 1$, where two primes $p$ and $q$ are of the same bit-size. The relevant moduli are inclusive of $pq$, $p^r q$ for $r > 1$, and $p^r q^s$ for $r, s > 1$, which are used in the standard RSA scheme and other RSA-type variants. Previous works acquired the results mainly by solving univariate *modular* equations.
In contrast, we investigate how to efficiently factor $N = p^r q^s$ with given leakage of the primes by the *integer method* using the lattice-based technique in this paper. More precisely, factoring general RSA moduli with known most significant bits (MSBs) of the primes can be reduced to solving bivariate integer equations, which was first proposed by Coppersmith to factor $N = pq$ with known high bits. Our results provide a unifying solution to the factoring with known bits problem on general RSA moduli. Furthermore, we reveal that there exists an improved factoring attack via the integer method for particular RSA moduli like $p^3 q^2$ and $p^5 q^3$.

**Keywords:** Factorization · General RSA moduli · Known bits · Integer method · Lattice-based technique

## 1 Introduction

### 1.1 Background

RSA [28] is a famous public key cryptosystem and has been widely used for secure data transmission. In its standard scheme, the modulus $N$ is the product of two large primes, namely $p$ and $q$ of the same bit-size. The public and private keys are defined as $(N, e)$ and $(p, q, d)$ respectively, where $e$ is a randomly chosen integer such that $e, \varphi(N)$ are coprime for Euler's totient $\varphi(N) = (p-1)(q-1)$ and $d$ satisfies the key equation $ed \equiv 1 \pmod{\varphi(N)}$. The message string is transformed into an integer $m$ and then encrypted as $c = m^e \pmod{N}$. The decryption phase computes $c^d \pmod{N}$. Since $e$ and $d$ are exponents in above computation, they are sometimes called the public and private exponents.

In order to speed up the decryption phase when utilizing RSA in some constrained environments like smart cards, some variants with modified moduli have been proposed. These modified moduli are designed as $N = p^r q$ for $r > 1$, or $N = p^r q^s$ for $r, s > 1$, or $N = \prod_{i=1}^{r} p_i$ for $r \geq 3$. Similarly, the primes appearing in each modulus are suggested to share the same bit-size.

The security of above RSA variants is also related to *integer factoring problem* like the standard one. The well-known algorithm for factorizing large composite integers is Number Field Sieve (NFS) [18], which works in sub-exponential time. In practice, some partial information leaked by side channel attacks (e.g. [12,17]) can be used to enhance the factoring attack. The so-called partial information is usually referred to some known bits of the primes.

Therefore, it is interesting to investigate polynomial-time factorization of such moduli used in RSA and its variants with some known bits of the primes. It is designated as *factoring with known bits problem* and has been widely analyzed in the literature. We review previous related works below.

**Factoring $N = pq$ with known bits.** This problem was first studied by Rivest and Shamir [27] in 1985. They used integer programming to construct an algorithm that factors $N$ when given 2/3-fraction of the bits of $p$. Later in 1996, Coppersmith [6] showed that $N$ can be factored when half of the bits of $p$ are known. The main technique is to solve small roots of modular/integer polynomial equations using lattice reduction algorithm i.e. the LLL algorithm [19]. This lattice-based technique is also named Coppersmith's technique [4,5] in the literature.

Since previous analysis assumed that the remaining unknown bits of $p$ are located in one consecutive block, an algorithm proposed by Herrmann and May [13] extended to the case of $n$ unknown discrete blocks. They showed that $\ln 2 \approx 70\%$ known bits of $p$ are sufficient to attain the factorization. However, the running time is polynomial in $\log N$ but exponential in $n$. It means that the running time becomes polynomial for $n = \mathcal{O}(\log\log N)$.

**Factoring $N = p^r q$ with known bits.** This RSA variant using a modulus in the form of $N = p^r q$ was suggested by Takagi [29] in 1998. Later, Boneh, Durfee and Howgrave-Graham [2] considered the factorization of $N$ by applying Coppersmith's technique. They showed that exposing a fraction $1/(r + 1)$ of the bits of $p$ is sufficient to factor $N$ in polynomial time. Furthermore, when $r$ increases to $r \approx \log p$, one only needs to know a constant number of the bits of $p$, which can be recovered by exhaustive search, hence the running time of the factorization becomes polynomial. The result implies that one should not use Takagi's RSA variant with a large $r$.

Inspired by [13], Lu, Zhang and Lin [23] proposed lattice-based analysis to extend the number of unknown blocks in prime $p$ to an arbitrary number $n$. The result showed that knowing a fraction $\ln(r + 1)/r$ of the bits of $p$ is already enough to factor $N$. However, the running time is polynomial only for $n = \mathcal{O}(\log\log N)$.

**Factoring $N = p^r q^s$ with known bits.** In 2000, Lim et al. [21] extended general RSA moduli $N = p^r q$ to the form of $N = p^r q^s$. The advantage is that the decryption phase is much faster than that in Takagi's RSA variant. On the other hand, how to generalize the lattice-based factoring method to $N = p^r q^s$ for $r$ and $s$ of almost the same size was considered as an open problem in [2]. Lim et al. also analyzed the security of the extended RSA variant with $N = p^r q^{r+1}$ by a modified lattice-based factoring method. When $r$ satisfies $r \geq \log(pq)$, i.e. $r \geq 2 \log p$, $N = p^r q^{r+1}$ can be factored in polynomial time.

In 2016, Coron et al. [9] proposed an algorithm to factor $N = p^r q^s$ in polynomial time when $r$ is greater than $\log^3 p$. They first aimed to find an appropriate decomposition of $r$ and $s$ and then applied Coppersmith's technique to factor $N$. Later this result was improved to $r \geq \log p$ by Coron and Zeitoun [10,11]. To be specific, there exist two positive integers $a$ and $b$ such that $a \cdot s - b \cdot r = 1$, which lead to the decomposition of $N^a = (p^a q^b)^r q$. It is much simpler to factor $N^a = (p^a q^b)^r q$ using the algorithm in [2] to recover $p$ and $q$.

Lu, Peng and Sarkar [22] studied how to factor $N = p^r q^s$ with partial known bits of $p$ or of $pq$. Knowing a fraction $\min\{s/(r+s), 2(r-s)/(r+s)\}$ of the bits of $p$ is sufficient to factor $N$ in polynomial time. The attack then was generalized to the case of $n$ unknown blocks for an arbitrary number $n$ similar to that mentioned above.

**Factoring $\boldsymbol{N = \prod_{i=1}^{r} p_i}$ with known bits.** This variant modifies modulus $N$ to be $p_1 p_2 \cdots p_r$ for $r \geq 3$. It was patented by Compaq [3], using a modulus in the form of $N = p_1 p_2 p_3$. The advantage is the efficiency when using Chinese Remainder Theorem in its decryption phase. The asymptotic speedup over standard RSA is approximately $r^2/4$. Moreover, small private exponent attack and partial key exposure attack are less effective as $r$ increases. On the other hand, $r$ should not be extremely large because of Elliptic Curve Method (ECM) [20].

Hinek [14] studied its related factoring problem and presented the following attack by directly applying the lattice-based factoring method proposed in [2]. For any $s \in [2, r]$, given a modulus $N$, along with $r - s$ known primes of all $r$ many prime factors, a fraction $(s-1)/s$ of the bits of one remaining unknown prime, a fraction $(s-2)/(s-1)$ of the bits of another one remaining unknown prime, …, and $1/2$ of the bits of one of the last two remaining unknown primes, then $N$ can be factored in time polynomial in $r$ and $\log N$. Besides, factoring $N = p_1 p_2 \cdots p_r$ with small prime difference, which can be viewed as knowing some MSBs of the primes, has been studied in [30,31].

### 1.2 Our Contributions

In this paper, we revisit the factoring with known bits problem by solving several distinct equations with the help of lattice-based technique. Instead of solving modular equations (or the *modular method* for short), we handle the problem by solving integer equations (or the *integer method* for short). Previous factoring attacks on general RSA moduli with known bits other than Coppersmith's original work [4] are based on the modular method. Thus, we further exploit the power of the integer method to present a unifying condition on factoring $N = p^r q^s$ with known bits. We want to point out that optimizing the solution to above factoring with known bits problem on general RSA moduli is mainly of theoretical interest.

The subsequent analysis restricts our attack scenario when given some MSBs in each prime leaving behind one consecutive unknown block. Though the description of our attack scenario is uncomplicated, we have many integer equations to solve in different cases. Without loss of generality, we have the following reasonable assumptions on the integer exponents $r$ and $s$ to simplify our analysis.

- They are known, otherwise we can recover them by exhaustive search in time $\mathcal{O}(\log^2 N)$.
- We have $1 \leq s \leq r \ll \log p$, otherwise we can exchange $p$ and $q$.
- $\gcd(r, s) = 1$, otherwise we can factor $N' = p^{r'} q^{s'}$ for $r' = \frac{r}{\gcd(r,s)}$ and $s' = \frac{s}{\gcd(r,s)}$.

More precisely, we aim to factor $N = p^r q^s$ for $r \geq s \geq 1$ with some known MSBs namely $P$ and $Q$ respectively, where $r$ and $s$ are two known coprime integers. The LSBs case is skipped since it is similar to the MSBs case. Previous factoring attacks [2,9,13,22,23] on general RSA moduli with known bits except Coppersmith's original work [4] are based on the modular method. The difference is that distinct solvable equations are used in two methods. When performing factoring attacks on $N = pq$ with $P$ and $Q$, the modular method aims to solve $P + x = 0 \pmod{p}$ while $(P + x)(Q + y) - N = 0$ is considered in the integer method.

Firstly, we show that all the previous results can be obtained through the integer method. In fact, the modular method is preferable when $s$ is small (down to 1) and $s$ is large (up to $r - 1$) because of the efficiency. Secondly, we observe that the least amount of known MSBs to factor $N$ depends on the relation of $r$ and $s$. To be specific, we identify some particular $(r, s)$ pairs for $s$ is medium (e.g. $s = \frac{r+1}{2}$ for odd integers $r$) while the integer method surpasses the modular method.

In other words, our results can be seen as an extension of Coppersmith's work [4] via the integer method, as well as a refinement of previous solutions to the factoring with known bits problem. With respect to solvable integer equations, we provide the concrete choices for several RSA moduli with $1 \le r, s \le 7$ in Table 1. A direct application of our results is to factor RSA moduli in the forms of $p^{r+1}q^r$, $p^{r+1}q^{r-1}$ and $p^{r+2}q^{r-2}$ with known bits. Such RSA moduli were suggested by Lim et al. [21] considering optimal efficiency for a roughly fixed sum of the exponents. It is clear that $p^3q^2$ and $p^5q^3$ are more vulnerable to the integer method.

**Table 1.** The choices of solvable integer equations for several RSA moduli

| Solvable Integer Equations | Applicable RSA Moduli |
|---|---|
| $(P + x)^r y - N = 0$ | $pq$, $p^2q$, $p^3q$, $p^4q$, $p^5q$, $p^5q^2$, $p^6q$, $p^7q$, $p^7q^2$, $p^7q^3$ |
| $(P + x)^r(Q + y)^s - N = 0$ | $p^3q^2$, $p^5q^3$, $p^7q^4$ |
| $(PQ + x)^s y - N = 0$ | $p^4q^3$, $p^5q^4$, $p^6q^5$, $p^7q^5$, $p^7q^6$ |

We provide a unifying condition on the desired amount of the prime leakage. Informally speaking, knowing a fraction

$$\min\left\{\frac{s}{r+s}, \frac{\sqrt{rs}}{r+s-1+\sqrt{rs}}, \frac{2(r-s)}{r+s}\right\}$$

of the bits of $p$ is sufficient to factor $N = p^r q^s$ for coprime integers $r > s$ in polynomial time. Our asymptotic improvement and previous analytic results are showed in Figure 1. The concrete results with respect to different attack scenarios are showed in Table 2. More comparison and discussion of the integer and modular methods are depicted in detail in Sect. 4.

**Table 2.** The desired fractions of known bits with respect to relevant solvable integer equations

| The Fractions of Known Bits | Solvable Integer Equations | Restrictions |
|---|---|---|
| $s/(r + s)$ | $(P + x)^r y - N = 0$ | $1 \le s \le r$ |
| $\sqrt{rs}/(r + s - 1 + \sqrt{rs})$ | $(P + x)^r(Q + y)^s - N = 0$ | $1 \le s \le r$ |
| $2(r - s)/(r + s)$ | $(PQ + x)^s y - N = 0$ | $1 \le s < r < 3s$ |

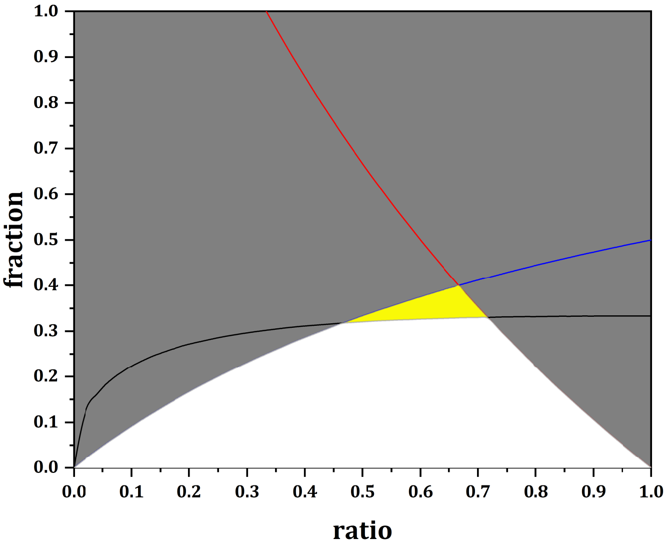

**Fig. 1.** The horizontal and vertical axes denote the ratio $s/r$ and requisite fraction of known bits, respectively. The grey area indicates known attack results that are bounded by previous analysis $\min\{s/(r+s),\ 2(r-s)/(r+s)\}$. The yellow area shows our improvement that is vulnerable to the integer method, which finally leads to a unifying solution to the factoring with known bits problem on general RSA moduli.

### 1.3 Organization

The rest of this paper is organized as follows. In Sect. 2, we review some basic definitions and a useful theorem employed in the integer method. We develop two parameterized theorems for the factoring with known bits problem on general RSA moduli. In Sect. 3, we propose several attacks when using known MSBs in both primes (i.e. $P$ and $Q$) or only one prime (i.e. $P$ or $Q$). In Sect. 4, we compare known results and ours in detail to obtain a unifying condition. We conclude the paper in Sect. 5.

## 2 Preliminaries

In this section, we first review several basic definitions of the integer method and then state a crucial theorem. After that, we propose two theorems for solving specific integer polynomials in our attack scenario. We note that the detailed lattice conception is not mentioned in order to simplify the analysis in this paper. More information can be found in [6,15,16,25,26]. The integer method stemmed from Coppersmith's work [4] and was later studied by [1,7,8,16].

We start with a set $M$ of monomials in the variables $x$ and $y$. A polynomial $f(x,y)$ is *defined over* $M$ or is called a *polynomial over* $M$ iff $f(x,y)$ can be written as

$$f(x,y) = \sum_{x^i y^j \in M} c_{ij} x^i y^j$$

for $c_{ij} \in \mathbb{Z}$. An integer polynomial $f(x,y) \in \mathbb{Z}[x,y]$ is called *irreducible* if $f(x,y) = g(x,y) \cdot h(x,y)$ with $g(x,y), h(x,y) \in \mathbb{Z}[x,y]$ implies that we have either $g(x,y) = \pm 1$ or $h(x,y) = \pm 1$. Additionally, the *greatest common divisor* of all coefficients of an irreducible polynomial must be 1.

More generally, there exists an index set (or a point set) in the Euclidean plane $\mathbb{R}^2$ for any set $M$ of monomials in two variables $x$ and $y$. The *index set* for $M$ is defined as

$$\mathcal{I}_M := \{(i,j) \in \mathbb{N}^2 : x^i y^j \in M\}.$$

The *convex hull* of the index set $\mathcal{I}_M$ is defined as $\mathrm{conv}\{(i,j) \in \mathbb{N}^2 : x^i y^j \in M\}$. Furthermore, we define a *convex set* $N(g)$ in the Euclidean plane for a polynomial $g(x,y) = \sum c_{ij} x^i y^j$ with $c_{ij} \in \mathbb{R}$ as

$$N(g) := \mathrm{conv}\{(i,j) \in \mathbb{N}^2 : c_{ij} \neq 0\},$$

which is also called the *Newton polygon* of $g(x,y)$.

It is important to identify the Newton polygon of an integer polynomial as well as its polynomial norm when we try to solve bivariate integer polynomials. The definition of the polynomial norm is given. Let $f(x,y) = \sum c_{ij} x^i y^j \in \mathbb{Z}[x,y]$ be an integer polynomial. The *$l_p$-norm* of $f(x,y)$ is defined as

$$\|f(x,y)\|_p = \left(\sum |c_{ij}|^p\right)^{1/p}.$$

The $l_\infty$-norm is involved in the literature of handling integer polynomials such as [1,7,8], We point out that it can be directly deduced from above definition as

$$\|f(x,y)\|_\infty = \max\{|c_{ij}|\}$$

for $f(x,y) = \sum c_{ij} x^i y^j$. We provide the following definition to guarantee that one can extract the roots of a given bivariate integer polynomial.

**Definition 1.** *Let $f(x,y)$ be a bivariate integer polynomial and $S, M$ be two finite non-empty monomial sets in the variables $x$ and $y$. The sets $S, M$ are called* admissible *for $f(x,y)$ iff*

1. *For every monomial $\alpha \in S$, the polynomial $\alpha \cdot f(x,y)$ is defined over $M$.*
2. *For every polynomial $g(x,y)$ defined over $M$, if $g(x,y) = h(x,y) \cdot f(x,y)$ for some polynomial $h(x,y)$, then $h(x,y)$ is defined over $S$.*

Before we show how to generate monomial sets $S$ and $M$ under the constraint that $S, M$ are admissible for $f(x,y)$, we give the definition of Minkowski sum of two index sets. Let $\mathcal{I}_A$ and $\mathcal{I}_B$ be two index sets. The *Minkowski sum* $\mathcal{I}_A + \mathcal{I}_B$ is defined as

$$\mathcal{I}_A + \mathcal{I}_B = \{(a_1,a_2) + (b_1,b_2) : (a_1,a_2) \in \mathcal{I}_A, (b_1,b_2) \in \mathcal{I}_B\}.$$

For a certain integer polynomial $f(x,y)$ and a chosen $S$, there is a natural choice for $M$ in order to guarantee the first property in Definition 1. That is, we choose $M$ such that $\mathcal{I}_M = N(f) + \mathcal{I}_S$. Actually, this choice will usually lead to monomial sets $S$ and $M$ that also satisfy the second property, i.e. $S, M$ are admissible for $f(x,y)$, which can be summarized in the following lemmas.

**Lemma 1.** *Assume that the Newton polygon $N(f)$ of polynomial $f(x,y)$ is $\{(i,j) \in \mathbb{N}^2 : 0 \leq i \leq a, 0 \leq j \leq b\}$ for positive integers $a$ and $b$. Then we use monomial sets $S$ and $M$ that correspond to two respective index sets*

$$\begin{aligned}\mathcal{I}_S =&\{(i,j) \in \mathbb{N}^2 : 0 \leq i \leq \gamma k, 0 \leq j \leq k\},\\ \mathcal{I}_M =&\{(i,j) \in \mathbb{N}^2 : 0 \leq i \leq \gamma k + a, 0 \leq j \leq k+b\},\end{aligned}$$

*where $k \in \mathbb{N}$ controls the size of low order error terms and $\gamma > 0$ optimizes the final condition. This construction lead to admissible sets $S$ and $M$ for $f(x,y)$.*

**Lemma 2.** *Assume that the Newton polygon $N(f)$ of polynomial $f(x,y)$ is $\{(i,j) \in \mathbb{N}^2 : 0 \leq i \leq \frac{c}{d}j, 0 \leq j \leq d\}$ for positive integers $c$ and $d$. Then we use monomial sets $S$ and $M$ that correspond to two respective index sets*

$$\begin{aligned}\mathcal{I}_S =& \{(i,j) \in \mathbb{N}^2 : 0 \leq i \leq \gamma k, 0 \leq j \leq k\}\\ &\cup \{(\gamma k + i, j) \in \mathbb{N}^2 : 0 \leq i \leq \frac{c}{d}j, 0 \leq j \leq k\},\\ \mathcal{I}_M =& \{(i,j) \in \mathbb{N}^2 : 0 \leq i \leq \gamma k, 0 \leq j \leq k+d\}\\ &\cup \{(\gamma k + i, j) \in \mathbb{N}^2 : 0 \leq i \leq \frac{c}{d}j, 0 \leq j \leq k+d\},\end{aligned}$$

*where $k \in \mathbb{N}$ controls the size of low order error terms and $\gamma > 0$ optimizes the final condition. This construction lead to admissible sets $S$ and $M$ for $f(x,y)$.*

See [1, Lemma 7] for the proofs. Now we state Blömer-May theorem from [1] for finding the roots of bivariate integer polynomials.

**Theorem 1.** *Let $f(x,y) \in \mathbb{Z}[x,y]$ be an irreducible integer polynomial in two variables $x$ and $y$ with degree at most $d_x, d_y \geq 1$, respectively. Let $X, Y \in \mathbb{N}$ be the upper bounds on roots $(x', y')$ and set $W := \|f(xX, yY)\|_\infty$. Furthermore let $S$, $M$ such that $S \subseteq M$, be two admissible monomial sets for $f(x,y)$. Set*

$$s := |S|, \quad m := |M|, \quad s_x := \sum_{x^i y^j \in M \setminus S} i, \quad s_y := \sum_{x^i y^j \in M \setminus S} j.$$

*All pairs $(x', y') \in \mathbb{Z}^2$ satisfying*

$$f(x', y') = 0 \text{ with } |x'| \leq X,\ |y'| \leq Y$$

*can be found in time polynomial in $m$, $d_x$, $d_y$ and $\log W$ provided*

$$X^{s_x} Y^{s_y} < W^s,$$

*assuming that $(m-s)^2 = \mathcal{O}(s d_x d_y)$ is satisfied.*

In order to provide a concise condition, we omit low order terms since the running time increases only by a constant factor. One may refer to [1, Section 5] for the lattice-based proof.

However, Theorem 1 cannot be directly applied to the factoring with known bits problem on general RSA moduli $p^r q^s$. We embody Blömer-May theorem in our theorems to solve two specific integer polynomials.

**Theorem 2.** *Given $f(x, y) = (x + \tilde{x})^a (y + \tilde{y})^b - N$ for two positive integers $a, b$ with a known composite integer $N$ and two approximations $\tilde{x}, \tilde{y}$. Let $X, Y \in \mathbb{N}$ be the upper bounds on roots $(x', y')$ and set $W := \|f(xX, yY)\|_\infty$. All roots $(x', y') \in \mathbb{Z}^2$ satisfying $f(x', y') = 0$ with $|x'| \leq X$, $|y'| \leq Y$ can be found in time polynomial in $\log W$ if*

$$X^{b\gamma^2+2a\gamma} Y^{2b\gamma+a} < W^{2\gamma}$$

*for an optimizing parameter $\gamma > 0$.*

*Proof.* Note that polynomial $f(x, y)$ is an irreducible polynomial, whose Newton polygon $N(f)$ is

$$\{(i, j) \in \mathbb{N}^2 : 0 \leq i \leq a, 0 \leq j \leq b\}.$$

We can construct two admissible sets $S$ and $M$ such that $S \subseteq M$ according to Lemma 1,

$$\begin{aligned} S =& \{x^i y^j : 0 \leq i \leq \gamma k, 0 \leq j \leq k\}, \\ M =& \{x^i y^j : 0 \leq i \leq \gamma k + a, 0 \leq j \leq k + b\}, \end{aligned}$$

where $k \in \mathbb{N}$ and $\gamma > 0$ is an optimizing parameter. Furthermore, we calculate $s$, $m$, $s_x$ and $s_y$ stated in Theorem 1 as follows.

$$s = \sum_{j=0}^{k} \sum_{i=0}^{\gamma k} 1 = \gamma k^2 + o(k^2), \quad m = \sum_{j=0}^{k+b} \sum_{i=0}^{\gamma k+a} 1 = \gamma k^2 + o(k^2),$$

$$s_x = \sum_{j=0}^{k+b} \sum_{i=0}^{\gamma k+a} i - \sum_{j=0}^{k} \sum_{i=0}^{\gamma k} i = \frac{b\gamma^2 + 2a\gamma}{2} k^2 + o(k^2),$$

$$s_y = \sum_{j=0}^{k+b} \sum_{i=0}^{\gamma k+a} j - \sum_{j=0}^{k} \sum_{i=0}^{\gamma k} j = \frac{2b\gamma + a}{2} k^2 + o(k^2).$$

We substitute these values in $X^{s_x} Y^{s_y} < W^s$ (omitting the lower order terms $o(k^2)$ for simplicity) and obtain

$$X^{\frac{b\gamma^2+2a\gamma}{2} k^2} Y^{\frac{2b\gamma+a}{2} k^2} < W^{\gamma k^2},$$

which leads to

$$X^{b\gamma^2+2a\gamma} Y^{2b\gamma+a} < W^{2\gamma}.$$

Furthermore, we have $d_x = a$ and $d_y = b$. The condition $(m - s)^2 = \mathcal{O}(s d_x d_y) = \mathcal{O}(k^2)$ is satisfied. The time complexity is mainly dominated by $\log W$ since we have $a, b \ll \log W$ and set $k = \log W$. Thus the running time is polynomial in $\log W$. This concludes the proof of the theorem. □

**Theorem 3.** *Given $f(x, y) = (x + \tilde{x})^c y^d - N$ for two positive integers $c, d$ with a known composite integer $N$ and an approximation $\tilde{x}$. Let $X, Y \in \mathbb{N}$ be the upper bounds on roots $(x', y')$ and set $W := \|f(xX, yY)\|_\infty$. All roots $(x', y') \in \mathbb{Z}^2$ satisfying $f(x', y') = 0$ with $|x'| \leq X$, $|y'| \leq Y$ can be found in time polynomial in $\log W$ if*

$$X^{(d\gamma+c)^2} Y^{2d(d\gamma+c)} < W^{2d\gamma+c}$$

*for an optimizing parameter $\gamma > 0$.*

*Proof.* Note that polynomial $f(x,y)$ is an irreducible polynomial, whose Newton polygon $N(f)$ is

$$\{(i,j) \in \mathbb{N}^2 : 0 \le i \le \frac{c}{d}j, 0 \le j \le d\}.$$

We can construct two admissible sets $S$ and $M$ such that $S \subseteq M$ according to Lemma 2,

$$\begin{aligned}
S &= \{x^i y^j : 0 \le i \le \gamma k, 0 \le j \le k\} \\
&\quad \cup \{x^{\gamma k+i} y^j : 0 \le i \le \frac{c}{d}j, 0 \le j \le k\}, \\
M &= \{x^i y^j : 0 \le i \le \gamma k, 0 \le j \le k+d\} \\
&\quad \cup \{x^{\gamma k+i} y^j : 0 \le i \le \frac{c}{d}j, 0 \le j \le k+d\},
\end{aligned}$$

where $k \in \mathbb{N}$ and $\gamma > 0$ is an optimizing parameter. Furthermore, we calculate $s$, $m$, $s_x$ and $s_y$ stated in Theorem 1 as follows.

$$\begin{aligned}
s &= \sum_{j=0}^{k}\sum_{i=0}^{\gamma k} 1 + \sum_{j=0}^{k}\sum_{i=0}^{\frac{c}{d}j} 1 = (\gamma + \frac{c}{2d})k^2 + o(k^2), \\
m &= \sum_{j=0}^{k+d}\sum_{i=0}^{\gamma k} 1 + \sum_{j=0}^{k+d}\sum_{i=0}^{\frac{c}{d}j} 1 = (\gamma + \frac{c}{2d})k^2 + o(k^2), \\
s_x &= \sum_{j=0}^{k+d}\sum_{i=0}^{\gamma k} i + \sum_{j=0}^{k+d}\sum_{i=0}^{\frac{c}{d}j}(\gamma k + i) - \sum_{j=0}^{k}\sum_{i=0}^{\gamma k} i - \sum_{j=0}^{k}\sum_{i=0}^{\frac{c}{d}j}(\gamma k + i) \\
&= \frac{(d\gamma + c)^2}{2d}k^2 + o(k^2), \\
s_y &= \sum_{j=0}^{k+d}\sum_{i=0}^{\gamma k} j + \sum_{j=0}^{k+d}\sum_{i=0}^{\frac{c}{d}j} j - \sum_{j=0}^{k}\sum_{i=0}^{\gamma k} j - \sum_{j=0}^{k}\sum_{i=0}^{\frac{c}{d}j} j \\
&= (d\gamma + c)k^2 + o(k^2).
\end{aligned}$$

We substitute these values in $X^{s_x}Y^{s_y} < W^s$ and obtain

$$X^{\frac{(d\gamma+c)^2}{2d}k^2} Y^{(d\gamma+c)k^2} < W^{(\gamma+\frac{c}{2d})k^2},$$

which reduces to

$$X^{(d\gamma+c)^2} Y^{2d(d\gamma+c)} < W^{2d\gamma+c}.$$

Furthermore, we have $d_x = c$ and $d_y = d$. The condition $(m-s)^2 = \mathcal{O}(sd_xd_y) = \mathcal{O}(k^2)$ is satisfied. The time complexity is mainly dominated by $\log W$ since we have $a, b \ll \log W$ and set $k = \log W$. Thus the running time is polynomial in $\log W$. This concludes the proof of the theorem. □

## 3 Factoring General RSA Moduli with Known Bits

In this section, we propose several attacks to factor $N$ with known MSBs, namely $P$ and $Q$. Let us specify the attack scenario clearly. Given $N = p^r q^s$ with $r, s$ and two MSBs approximations $P, Q$, where $p = P + x$ and $q = Q + y$ for unknown variables

$x, y$ that can be bounded by $X = Y = N^{\eta}$, the goal is to efficiently recover $p$ and $q$ that lead to the factorization of $N$ under the minimal sizes of $P$ and $Q$. It means that the size of known MSBs of $p$ (or $q$) is $N^{1/(r+s)-\eta}$, or equivalently $p^{1-(r+s)\eta}$.

We will provide the results by applying the integer method. To do so, we should derive some integer equations from above attack scenario. The suitable integer equations are divided into two parts as follows. The first part is involved with two approximations that consists of solving

$$(P+x)^r(Q+y)^s - N = 0, \quad \text{and} \quad (PQ+x)^s y - N = 0.$$

The second part is related to only one approximation, which consists of solving

$$(P+x)^r y - N = 0.$$

Before presenting the results, we show that known MSBs in one prime can be used to compute some MSBs of the same bit-size in another prime.

**Lemma 3.** *Let $N = p^r q^s$ for $r, s \geq 1$, where $p$ and $q$ are of the same bit-size. Given an MSBs approximation $P$ of $p$ for $|p - P| < N^{\eta}$, the rounding integer $Q := [(N/P^r)^{\frac{1}{s}}]$ is an MSBs approximation of $q$ and satisfies $|q - Q| < N^{\eta}$.*

*Proof.* Because $r, s$ are negligible compared to $p$ and $q$, we can assume that $p, q$ and $P$ are roughly equal to $N^{\frac{1}{r+s}}$ and thus $Q$ is also roughly equal to $N^{\frac{1}{r+s}}$. In order to bound the value of $|q - Q|$, we first bound the value of $|q^s - Q^s|$ since we have

$$|q - Q| = \frac{|q^s - Q^s|}{q^{s-1} + q^{s-2}Q + \cdots + qQ^{s-2} + Q^{s-1}} \approx \frac{|q^s - Q^s|}{sN^{\frac{s-1}{r+s}}}.$$

We define $Q := [(N/P^r)^{\frac{1}{s}}]$ and it leads to $Q^s \approx N/P^r$, which gives

$$|q^s - Q^s| \approx |q^s - \frac{N}{P^r}| = \frac{q^s|P^r - p^r|}{P^r} \approx |P^r - p^r|N^{\frac{s-r}{r+s}}.$$

Now we bound the value of $|P^r - p^r|$, that is

$$|P^r - p^r| = |P - p|(P^{r-1} + P^{r-2}p + \cdots + Pp^{r-2} + p^{r-1}) < rN^{\frac{r-1}{r+s}+\eta}.$$

Combining above results (and omitting negligible $r$ and $s$), we have

$$|q - Q| \approx \frac{|q^s - Q^s|}{N^{\frac{s-1}{r+s}}} \approx \frac{|P^r - p^r|N^{\frac{s-r}{r+s}}}{N^{\frac{s-1}{r+s}}} < \frac{N^{\frac{r-1}{r+s}+\eta}N^{\frac{s-r}{r+s}}}{N^{\frac{s-1}{r+s}}} = N^{\eta},$$

which terminates the proof. □

In the following attacks, we mention the known leakage that always refers to the MSBs approximation $P$ and this implies we know both $P$ and $Q$ from $N$, $r$ and $s$.

### 3.1 Using Two Approximations

We present the results in theorems derived from solving bivariate integer equations. More concretely, we try to solve

$$(P+x)^r(Q+y)^s - N = 0 \quad \text{and} \quad (PQ+x)^s y - N = 0$$

to obtain the solution to the factoring with known bits problem. It is a straightforward option to solve

$$(P+x)^r(Q+y)^s - N = 0,$$

which is based on the observation that we can directly put $p = P + x$ and $q = Q + y$ into $N = p^r q^s$. Our result is stated below.

**Theorem 4.** *Let $N = p^r q^s$ for $r \geq s \geq 1$, where $p$ and $q$ are of the same bit-size. Suppose that a fraction*

$$\frac{\sqrt{rs}}{r+s-1+\sqrt{rs}}$$

*of the bits of $p$ are known, then we can factor $N$ in time polynomial in $\log N$.*

*Proof.* Let

$$f(x,y) = (P+x)^r(Q+y)^s - N$$

and we apply Theorem 2 with $\tilde{x} = P$, $\tilde{y} = Q$, $a = r$ and $b = s$ to obtain

$$X^{s\gamma^2+2r\gamma}Y^{2s\gamma+r} < W^{2\gamma}.$$

We need to figure out the value of $W$ since we know $X = Y = N^\eta$ and $P \approx Q \approx N^{\frac{1}{r+s}}$. Since $r, s \ll \log p$, the binomial coefficients can not exceed $P, Q$ and we have

$$W = \|f(xX, yY)\|_\infty = \max\{|P^{r-1}XQ^s|, |P^rQ^{s-1}Y|, |P^rQ^s - N|\} = N^{\frac{r+s-1}{r+s}+\eta}.$$

Considering the exponents in the condition, it leads to

$$\eta(s\gamma^2 + 2r\gamma + 2s\gamma + r) < 2\gamma\left(\frac{r+s-1}{r+s} + \eta\right),$$

which further reduces to

$$\eta < \frac{2(r+s-1)\gamma}{(r+s)(s\gamma^2 + 2(r+s-1)\gamma + r)}.$$

We set $\gamma = \sqrt{r/s}$ to make the right side reach its maximum and then obtain

$$\eta < \frac{r+s-1}{(r+s)(r+s-1+\sqrt{rs})}.$$

A fraction $1 - (r+s)\eta$ is required to recover $p$ and $q$, so it implies that we require at least a fraction

$$\frac{\sqrt{rs}}{r+s-1+\sqrt{rs}}$$

of the bits of $p$ and $q$. The running time is polynomial in $\log W$, and it is also polynomial in $\log N$. $\square$

In fact, we have a new integer equation

$$(PQ + x)^s y - N = 0$$

from the observation

$$(P + x)^r (Q + y)^s = ((P + x)(Q + y))^s p^{r-s} = (PQ + Qx + Py + xy)^s p^{r-s} = N.$$

Thus, we can apply Theorem 3 for this bivariate integer equation and the result is stated below.

**Theorem 5.** *Let $N = p^r q^s$ for $1 \leq s < r < 3s$, where $p$ and $q$ are of the same bit-size. Suppose that a fraction*

$$\frac{2(r-s)}{r+s}$$

*of the bits of $p$ are known, then we can factor $N$ in time polynomial in $\log N$.*

*Proof.* Let

$$f(x, y) = (PQ + x)^s y - N$$

and we apply Theorem 3 with $\tilde{x} = PQ$, $c = s$ and $d = 1$, we have

$$X^{(\gamma+s)^2} Y^{2(\gamma+s)} < W^{2\gamma+s}.$$

We figure out the values of $X$, $Y$ that are $X = N^{\frac{1}{r+s}+\eta}$ and $Y = p^{r-s} = N^{\frac{r-s}{r+s}}$. The value of $W$ is

$$W = \|f(xX, yY)\|_\infty = \max\{|(PQ)^s Y|, |N|\} = N.$$

From the condition, we have

$$(\gamma + s)^2 \left( \frac{1}{r+s} + \eta \right) + \frac{2(r-s)}{r+s}(\gamma + s) < 2\gamma + s,$$

which reduces to

$$\eta < \frac{-\gamma^2 + 2s\gamma + 2s^2 - rs}{(r+s)(\gamma+s)^2}.$$

We set $\gamma = (r - s)/2$ to make the right side reach its maximum and then obtain

$$\eta < \frac{3s - r}{(r+s)^2}.$$

We must have $s < r < 3s$ since $\gamma, \eta > 0$. The solution of $y$ is enough to compute $p$, so a fraction at least

$$1 - (r + s)\frac{3s - r}{(r+s)^2} = \frac{2(r-s)}{r+s}$$

of the bits of $p$ is required to recover $p$ and then factor $N$. The running time is polynomial in $\log W$, and thus is polynomial in $\log N$. □

In addition, we also can solve

$$(PQ+x)^s y^{r-s} - N = 0, \quad \text{and} \quad (PQ+x)^s (P+y)^{r-s} - N = 0.$$

The result of the former equation is the same as Theorem 5. We apply Theorem 3 with $\tilde{x} = PQ$, $c = s$ and $d = r - s$ for $X = N^{\frac{1}{r+s}+\eta}$, $Y = N^{\frac{1}{r+s}}$ and $W = N$. Setting $\gamma = \frac{1}{2}$ in the proof to obtain

$$\eta < \frac{3s-r}{(r+s)^2},$$

which leads to the same result that we require a fraction at least

$$1 - (r+s)\frac{3s-r}{(r+s)^2} = \frac{2(r-s)}{r+s}.$$

The value of $y$ is $p$ and then we can factor $N$.

As for the latter equation, we can apply Theorem 2 with $\tilde{x} = PQ$, $\tilde{y} = P$, $a = s$ and $b = r - s$ for $X = N^{\frac{1}{r+s}+\eta}$, $Y = N^{\eta}$ and $W = N^{\frac{r+s-1}{r+s}+\eta}$. The result is that we need at least a fraction

$$\frac{\sqrt{s^2(r-s)^2 + 8s(r-s)(r-1)^2} + s(r-s)}{\sqrt{s^2(r-s)^2 + 8s(r-s)(r-1)^2} + s(r-s) + 2(r-1)^2}$$

of the bits of $p$ to factor $N$ in time polynomial in $\log N$ for $r > s \geq 1$. However, this result is always inferior to that stated in Theorem 4 and Theorem 5.

For the sake of completeness, we provide the result of solving univariate modular equations instead of bivariate integer equations since we can achieve an acceleration of efficiency. We have a modular equation $PQ + x = 0 \pmod{pq}$, where $x$ is bounded by $X = N^{\frac{1}{r+s}+\eta}$ and $(pq)^s$ is a divisor of $N$. Thus, we apply the modular method summarized in [24] and obtain the same result as that in Theorem 5.

### 3.2 Using One Approximation

We employ both $p = P + x$ and $q = Q + y$ for unknown variables $x, y$ bounded by $X = Y = N^{\eta}$ in Sect. 3.1. But we observe that $W$ decreases when taking both $P$ and $Q$ into consideration and it may weaken the bound on $\eta$.

In this section, we handle the factoring with known bits problem only with the help of $P$ or $Q$. More concretely, we try to solve

$$(P+x)^r y - N = 0$$

without the knowledge of $Q$, whose result is stated below.

**Theorem 6.** *Let $N = p^r q^s$ for $r \geq s \geq 1$, where $p$ and $q$ are of the same bit-size. Suppose that a fraction*

$$\frac{s}{r+s}$$

*of the bits of $p$ are known, then we can factor $N$ in time polynomial in $\log N$.*

*Proof.* Let

$$f(x,y) = (P+x)^r y - N$$

and we apply Theorem 3 with $\tilde{x} = P$, $c = r$ and $d = 1$ to obtain

$$X^{(\gamma+r)^2} Y^{2(\gamma+r)} < W^{2\gamma+r},$$

where the upper bounds are $X = N^{\eta}$, $Y = N^{\frac{s}{r+s}}$ and $W = \|f(xX, yY)\|_{\infty} = N$. Then we have

$$\eta(\gamma+r)^2 + \frac{2s}{r+s}(\gamma+r) < 2\gamma + r.$$

It reduces to

$$\eta < \frac{2r\gamma + r^2 - rs}{(r+s)(\gamma+r)^2}.$$

We set $\gamma = s$ to make the right side reach its maximum and then obtain

$$\eta < \frac{r}{(r+s)^2}.$$

The solution of roots $x, y$ implies the values of $p$ and $q$, respectively. So a fraction at least

$$1 - (r+s)\frac{r}{(r+s)^2} = \frac{s}{r+s}$$

is required to recover $p$ and then factor $N$. The running time is polynomial in $\log W$, and it is also polynomial in $\log N$. □

Similarly, we can solve

$$(P+x)^r y^s - N = 0$$

by Theorem 3 for $\tilde{x} = P$, $c = r$ and $d = s$ with the upper bounds $X = N^{\eta}$, $Y = N^{\frac{1}{r+s}}$ and $W = \|f(xX, yY)\|_{\infty} = N$. We set $\gamma = 1$ in the proof to obtain

$$\eta < \frac{r}{(r+s)^2},$$

which leads to the same result as that in Theorem 6. We also can derive a modular equation $P + x = 0 \pmod{p}$, where $x$ is bounded by $X = N^{\eta}$ and $p^r$ is a divisor of $N$. Thus, we apply the modular method summarized in [24], which leads to the same result as that in Theorem 6.

When we consider using one approximation $P$ or $Q$, there exists an integer equation

$$(Q+x)^s y - N = 0, \quad \text{or} \quad (Q+x)^s y^r - N = 0.$$

For completeness, we provide the result but do not discuss it in further comparison since it is a worse choice for $r \geq s$. For example, we apply Theorem 3 to solve

$$(Q+x)^s y - N = 0$$

for $\tilde{x} = Q$, $c = s$ and $d = r$ with $X = N^{\eta}$, $Y = N^{\frac{r}{r+s}}$ and $W = N$. Setting $\gamma = r$, we obtain

$$\eta < \frac{s}{(r+s)^2},$$

which means that a fraction at least

$$1 - (r+s)\frac{s}{(r+s)^2} = \frac{r}{r+s}$$

is required to recover $q$ and then factor $N$.

## 4 Comparison and Discussion

The modular method is more efficient and simpler than the integer method for some equations. So, the algorithms for solving modular equations are preferred when applying Theorem 5 and Theorem 6. However, the integer method shows its power for solving a general integer equation

$$(P+x)^r(Q+y)^s - N = 0,$$

which finally gives Theorem 4.

We compare the required amounts of known MSBs in a unifying condition derived from the integer method in Sect. 3 since the fractions of known bits differ when solving distinct integer equations. We summarize the respective fractions required for factoring general RSA moduli $N = p^r q^s$ with known bits and the corresponding solvable integer equations as follows.

- For the solvable equation

  $$(P+x)^r(Q+y)^s - N = 0 \quad \text{with} \quad r \geq s \geq 1,$$

  the fraction given by Theorem 4 is

  $$\frac{\sqrt{rs}}{r+s-1+\sqrt{rs}}.$$

- For the solvable equations

  $$(PQ+x)^s y - N = 0, \quad (PQ+x)^s y^{r-s} - N = 0 \quad \text{with} \quad 1 \leq s < r < 3s,$$

  the fraction given by Theorem 5 is

  $$\frac{2(r-s)}{r+s}.$$

- For the solvable equations

  $$(P+x)^r y - N = 0, \quad (P+x)^r y^s - N = 0 \quad \text{with} \quad r \geq s \geq 1,$$

  the fraction given by Theorem 6 is

  $$\frac{s}{r+s}.$$

We discuss more about our unifying condition. For the standard RSA modulus $N = pq$ with $r = s = 1$, we can apply Theorem 4 and Theorem 6. Our results cover that of [4] but we can provide more solvable equations. For the modified RSA modulus $N = p^r q$ with $r > 1, s = 1$, we can apply Theorem 6 since the required amount of known MSBs is least. Our results also cover those of [2,22].

However, for general RSA moduli $N = p^r q^s$ with arbitrary $r, s > 1$, we should compare the above three fractions to choose the best one. For example, we show the comparison of the numerical values of the respective fractions for $r = 3, 4, 5, 6$ with various reasonable $s$'s in Table 3. It is showed that the best choice actually depends on the relation of $r$ and $s$.

**Table 3.** The numerical values of the respective fractions for several $(r, s)$ pairs

| $(r, s)$ | $(3, 2)$ | $(4, 3)$ | $(5, 2)$ | $(5, 3)$ | $(5, 4)$ | $(6, 5)$ |
|---|---|---|---|---|---|---|
| Theorem 4 | **0.380** | 0.367 | 0.346 | **0.357** | 0.359 | 0.354 |
| Theorem 5 | 0.4 | **0.286** | 0.858 | 0.5 | **0.223** | **0.182** |
| Theorem 6 | 0.4 | 0.429 | **0.286** | 0.375 | 0.445 | 0.455 |

**Table 4.** The respective applicable ranges and suitable equations according to distinct theorems

| | Applicable Ranges | Most Suitable Equations | Restrictions |
|---|---|---|---|
| Theorem 6 | $1 \leq s \leq \theta(r) \cdot r$ | $P + x = 0 \pmod{p}$ | $1 \leq s \leq r$ |
| Theorem 4 | $\theta(r) \cdot r < s \leq \xi(r) \cdot r$ | $(P + x)^r (Q + y)^s - N = 0$ | $1 \leq s \leq r$ |
| Theorem 5 | $\xi(r) \cdot r < s < r$ | $PQ + x = 0 \pmod{pq}$ | $1 \leq s < r < 3s$ |

To be concrete, Theorem 4 is preferred for medium $s$ for a fixed $r$. Theorem 6 is more effective for small $s$ like $s = 1$ and Theorem 5 works better for large $s$ like $s = r - 1$. Furthermore, we identify the respective applicable ranges of $s$ along with the most suitable solvable equations for each theorem in Table 4. The results also include $s = 1$ that is considered as a special case of Theorem 6 if $\theta(r) < 1$. Additionally, the restrictions on each theorem are always satisfied.

We define two functions $\theta(r)$ and $\xi(r)$ for simplicity since the explicit forms are complicated. $\theta(r)$ is the unique real root in $(0, 1)$ of the following equation

$$\frac{\sqrt{xr}}{r + xr - 1 + \sqrt{xr}} = \frac{xr}{r + xr},$$

and $\xi(r)$ is the unique real root in $(0, 1)$ of the following equation

$$\frac{\sqrt{xr}}{r + xr - 1 + \sqrt{xr}} = \frac{2(r - xr)}{r + xr}.$$

We list the numerical values of $\theta(r)$ and $\xi(r)$ for some $r < 10$ in Table 5. The results are applicable for all reasonable $(r, s)$ pairs if we let the cases when $s = 1$ for $r = 1, 2$ belong to Theorem 6. Finally, we derive a unifying condition for factoring general RSA moduli $N = p^r q^s$ with known bits. For example, we roughly give the applicable ranges of $s$ for a fixed $r < 10$ as follows since coprime integers $r, s \ll \log p$.

**Table 5.** The numerical values of $\theta(r)$ and $\xi(r)$ for various $r < 10$

| $r$ | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|
| $\theta(r)$ | 1 | 0.697 | 0.611 | 0.572 | 0.549 | 0.534 | 0.524 | 0.516 | 0.510 |
| $\xi(r)$ | – | 0.658 | 0.680 | 0.690 | 0.696 | 0.699 | 0.702 | 0.704 | 0.705 |

– If $0.7r < s < r$, we choose to solve $PQ + x = 0 \pmod{pq}$.
– If $0.5r < s \leq 0.7r$, we choose to solve $(P + x)^r (Q + y)^s - N = 0$.
– Else cases, we choose to solve $P + x = 0 \pmod{p}$.

We do not extend the analysis to the case of more than one unknown block since it is heuristic and relies on an unproven assumption. We do not analyze factoring $N = \prod_{i=1}^{r} p_i$ with known bits either. In the case of more than two primes, there seem no other efficient algorithms except the modular method.

## 5 Concluding Remarks

We revisit the factoring with known bits problem on general RSA moduli $N = p^r q^s$ for $r, s \geq 1$, where $p, q$ are two primes of the same bit-size. To be specific, we study the least desired amount of known MSBs of the primes and obtain the results based on solving bivariate integer equations.

A unifying condition on the fraction of known MSBs is derived for efficiently factoring $N = p^r q^s$. On the one hand, previous works based on the modular method to factor $N = pq$ and $N = p^r q$ are confirmed to remain the best so far. For general moduli $N = p^r q^s$, the modular method is still applicable to $s$ of small or large size. On the other hand, we reveal that the integer method is superior for some particular $(r, s)$ pairs (e.g. $p^3q^2$ and $p^5q^3$) when $s$ is of medium size with respect to $r$, i.e. $s \in (0.5r, 0.7r]$ for $r < 10$.

We show that the integer method is more powerful since it covers the results derived from the modular method and even provides an improved factoring attack for several particular RSA moduli. We hope that the integer method is applicable to other problems that can be reduced to solving integer equations and further give better results.

**Acknowledgments.** The author would like to thank Noboru Kunihiro for helpful discussions and the anonymous referees for their constructive comments.

## References

1. Blömer, J., May, A.: A tool kit for finding small roots of bivariate polynomials over the integers. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 251–267. Springer, Heidelberg (May 2005)
2. Boneh, D., Durfee, G., Howgrave-Graham, N.: Factoring $N = p^r q$ for large $r$. In: Wiener, M.J. (ed.) CRYPTO'99. LNCS, vol. 1666, pp. 326–337. Springer, Heidelberg (Aug 1999)
3. Collins, T., Hopkins, D., Langford, S., Sabin, M.: Public key cryptographic apparatus and method (Dec 1998), U.S. Patent # 5848159
4. Coppersmith, D.: Finding a small root of a bivariate integer equation; factoring with high bits known. In: Maurer, U.M. (ed.) EUROCRYPT'96. LNCS, vol. 1070, pp. 178–189. Springer, Heidelberg (May 1996)
5. Coppersmith, D.: Finding a small root of a univariate modular equation. In: Maurer, U.M. (ed.) EUROCRYPT'96. LNCS, vol. 1070, pp. 155–165. Springer, Heidelberg (May 1996)
6. Coppersmith, D.: Small solutions to polynomial equations, and low exponent RSA vulnerabilities. Journal of Cryptology **10**(4), 233–260 (1997)
7. Coron, J.S.: Finding small roots of bivariate integer polynomial equations revisited. In: Cachin, C., Camenisch, J. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 492–505. Springer, Heidelberg (May 2004)
8. 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 (Aug 2007)

9. Coron, J.S., Faugère, J.C., Renault, G., Zeitoun, R.: Factoring $N = p^r q^s$ for large $r$ and $s$. In: Sako, K. (ed.) CT-RSA 2016. LNCS, vol. 9610, pp. 448–464. Springer, Heidelberg (Feb / Mar 2016)
10. Coron, J.S., Zeitoun, R.: Improved factorization of $N = p^r q^s$. Cryptology ePrint Archive, Report 2016/551 (2016)
11. Coron, J.S., Zeitoun, R.: Improved factorization of $n = p^r q^s$. In: Smart, N.P. (ed.) CT-RSA 2018. LNCS, vol. 10808, pp. 65–79. Springer, Heidelberg (Apr 2018)
12. Halderman, J.A., Schoen, S.D., Heninger, N., Clarkson, W., Paul, W., Calandrino, J.A., Feldman, A.J., Appelbaum, J., Felten, E.W.: Lest we remember: Cold-boot attacks on encryption keys. Commun. ACM **52**(5), 91–98 (May 2009)
13. 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 (Dec 2008)
14. Hinek, M.J.: On the security of multi-prime RSA. Journal of Mathematical Cryptology **2**(2), 117–147 (2008)
15. Howgrave-Graham, N.: Finding small roots of univariate modular equations revisited. In: Darnell, M. (ed.) 6th IMA International Conference on Cryptography and Coding. LNCS, vol. 1355, pp. 131–142. Springer, Heidelberg (Dec 1997)
16. 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 (Dec 2006)
17. Kocher, P.C.: Timing attacks on implementations of Diffie-Hellman, RSA, DSS, and other systems. In: Koblitz, N. (ed.) CRYPTO'96. LNCS, vol. 1109, pp. 104–113. Springer, Heidelberg (Aug 1996)
18. Lenstra, A.K., Lenstra, H.W., Manasse, M.S., Pollard, J.M.: The number field sieve. In: Lenstra, A.K., Lenstra, H.W. (eds.) The development of the number field sieve. pp. 11–42. Springer Berlin Heidelberg, Berlin, Heidelberg (1993)
19. Lenstra, A.K., Lenstra, H.W., Lovász, L.: Factoring polynomials with rational coefficients. Mathematische Annalen **261**(4), 515–534 (Dec 1982)
20. Lenstra, H.W.: Factoring integers with elliptic curves. Annals of Mathematics **126**(3), 649–673 (1987)
21. Lim, S., Kim, S., Yie, I., Lee, H.: A generalized Takagi-cryptosystem with a modulus of the form $p^r q^s$. In: Roy, B.K., Okamoto, E. (eds.) INDOCRYPT 2000. LNCS, vol. 1977, pp. 283–294. Springer, Heidelberg (Dec 2000)
22. Lu, Y., Peng, L., Sarkar, S.: Cryptanalysis of an RSA variant with moduli $N = p^r q^l$. Journal of Mathematical Cryptology **11**(2), 117–130 (2017)
23. Lu, Y., Zhang, R., Lin, D.: Factoring multi-power RSA modulus $N = p^r q$ with partial known bits. In: Boyd, C., Simpson, L. (eds.) ACISP 2013. LNCS, vol. 7959, pp. 57–71. Springer, Heidelberg (Jul 2013)
24. Lu, Y., Zhang, R., Peng, L., Lin, D.: Solving linear equations modulo unknown divisors: Revisited. In: Iwata, T., Cheon, J.H. (eds.) ASIACRYPT 2015, Part I. LNCS, vol. 9452, pp. 189–213. Springer, Heidelberg (Nov / Dec 2015)
25. May, A.: New RSA Vulnerabilities Using Lattice Reduction Methods. Ph.D. thesis, University of Paderborn, Paderborn, Germany (2003)
26. May, A.: Using LLL-reduction for solving RSA and factorization problems. In: Nguyen, P.Q., Valle, B. (eds.) The LLL Algorithm - Survey and Applications, pp. 315–348. ISC, Springer, Heidelberg (2010)
27. Rivest, R.L., Shamir, A.: Efficient factoring based on partial information. In: Pichler, F. (ed.) EUROCRYPT'85. LNCS, vol. 219, pp. 31–34. Springer, Heidelberg (Apr 1986)
28. Rivest, R.L., Shamir, A., Adleman, L.: A method for obtaining digital signatures and public-key cryptosystems. Commun. ACM **21**(2), 120–126 (Feb 1978)
29. Takagi, T.: Fast RSA-type cryptosystem modulo $p^k q$. In: Krawczyk, H. (ed.) CRYPTO'98. LNCS, vol. 1462, pp. 318–326. Springer, Heidelberg (Aug 1998)

30. Zhang, H., Takagi, T.: Attacks on multi-prime RSA with small prime difference. In: Boyd, C., Simpson, L. (eds.) ACISP 2013. LNCS, vol. 7959, pp. 41–56. Springer, Heidelberg (Jul 2013)

31. Zheng, M., Kunihiro, N., Hu, H.: Improved factoring attacks on multi-prime RSA with small prime difference. In: Pieprzyk, J., Suriadi, S. (eds.) ACISP 2017, Part I. LNCS, vol. 10342, pp. 324–342. Springer, Heidelberg (Jul 2017)