

## RESEARCH ARTICLE

# Solving Generalized Bivariate Integer Equations and Its Application to Factoring With Known Bits

MENGCE ZHENG<sup>1,2</sup>, (Member, IEEE), ZHIGANG CHEN<sup>1</sup>, AND YAOHUI WU<sup>1</sup>

<sup>1</sup>College of Information and Intelligence Engineering, Zhejiang Wanli University, Ningbo 315100, China

<sup>2</sup>School of Information Science and Technology, University of Science and Technology of China, Hefei 230027, China

Corresponding author: Mengce Zheng (mengce.zheng@gmail.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 62002335, in part by the Ningbo Natural Science Foundation under Grant 2021J1174 and Grant 2019A610076, and in part by the Zhejiang Provincial Natural Science Foundation of China under Grant LGF22F020001.

**ABSTRACT** In this paper, we propose two improved theorems for addressing generalized bivariate integer equations using the lattice-based method. We examine the application of these theorems to the problem of factoring general RSA (Rivest–Shamir–Adleman) moduli of the form  $N = p^r q^s$  where  $r, s \geq 1$  and  $p, q$  are prime numbers. These moduli, which are commonly used in the RSA cryptosystem and its variants, have previously been subjected to attacks primarily through the solution of univariate modular equations. In contrast, we investigate the possibility of factoring  $N = p^r q^s$  using leaked most significant bits (MSBs) or least significant bits (LSBs) of the prime numbers by solving generalized bivariate integer equations. We determine the minimum amount of known bits required for implementing the proposed factoring attacks and establish a unifying attack strategy. Furthermore, our results are verified through numerical computer experiments.


**INDEX TERMS** Bivariate integer equation, factorization, lattice, RSA.

## I. INTRODUCTION

Consider given a monomial set  $M$  with respect to two variables  $x, y$  and an integer polynomial  $f(x, y) := \sum_{x^i y^j \in M} c_{ij} x^i y^j$  for  $c_{ij} \in \mathbb{Z}$ , we are interested in solving all possible roots  $(x', y')$  satisfying  $f(x', y') = 0$  in polynomial time and further maximizing the upper bounds on  $x'$  and  $y'$ . Solving bivariate integer polynomial equations stemmed from Coppersmith's lattice-based analyses [1], [2] on RSA (Rivest–Shamir–Adleman) cryptosystem [3] and was later studied by [4], [5], [6], [7]. Coppersmith [2] and Coron [4], [7] studied several basic cases and efficient solving methods. Blömer and May [5] proposed an approach to analyze more situations and presented a useful theorem with concrete lattice constructions. The advantage is that solving a certain integer polynomial  $f(x, y)$  can be formulated just in terms of its monomials. Moreover, Jochemsz and May [6] presented a generic approach for extracting possible roots of modular and

integer multivariate polynomials. However, it is less efficient for several multivariate polynomials of specific structures.

RSA [3] is a widely used public-key cryptosystem for secure data transmission in cyberspace. The standard modulus  $N = pq$  is the product of two large primes of the same bit-size, namely  $p$  and  $q$ . To speed up the decryption phase when utilizing RSA in the constrained environments like smart cards, some variants with modified moduli such as  $N = p^r q$  for  $r > 1$  and  $N = p^r q^s$  for  $r, s > 1$  have been proposed. Similarly, the primes appearing in each modulus are suggested to share the same bit-size. The cryptosystem security is related to the integer factoring problem. A well-known algorithm for factorizing large composite integers is Number Field Sieve (NFS) [8], which runs in sub-exponential time. In practice, some partial information leaked by side-channel attacks (e.g. [9], [10]) can be used to enhance the factoring attacks by solving multivariate polynomial equations. The so-called partial information is usually referred to as some known bits of the primes. We further investigate polynomial-time factorization of such

The associate editor coordinating the review of this manuscript and approving it for publication was Pedro R. M. Inácio .

RSA moduli with some known bits of the primes, which is designated as the factoring with known bits problem.

Rivest and Shamir [11] first studied the factoring with known bits problem. They used integer programming to factor  $N$  when given  $2/3$ -fraction of  $p$ . Later Coppersmith [2] showed that it can be done when  $1/2$ -fraction of  $p$  are known. The main technique is to solve modular/integer equations using lattice reduction algorithms, i.e., the LLL (Lenstra-Lenstra-Lovász) algorithm [12]. This lattice-based idea is also named Coppersmith's technique in the literature. A fast RSA variant using modified moduli  $N = p^r q$  was suggested by Takagi [13]. Later, Boneh, Durfee, and Howgrave-Graham [14] demonstrated that exposing  $1/(r+1)$ -fraction 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  $p$  and it can be recovered by exhaustive search. Hence, the running time of the factorization becomes polynomial, which implies that one should not use Takagi's RSA variant with a large  $r$ .

Lim et al. [15] 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. How to generalize lattice-based factoring attacks on  $N = p^r q^s$  for  $r$  and  $s$  of almost same bit-size was considered as an open problem in [14]. 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 attack.  $N = p^r q^{r+1}$  can be factored in polynomial time when  $r \geq \log(pq)$ , i.e.,  $r \geq 2 \log p$ . In 2016, Coron et al. [16] factored  $N = p^r q^s$  in polynomial time when  $r > \log^3 p$ . They first aimed to find an appropriate decomposition of  $r$  and  $s$  and then applied Coppersmith's technique to factor  $N$ . This result was later improved to  $r \geq \log p$  by Coron and Zeitoun [17]. To be specific, we have two positive integers  $a, b$  satisfying  $as - br = 1$ , 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 [14] to recover  $p$  and  $q$ . Lu et al. [18] studied how to factor  $N = p^r q^s$  with partial known bits of  $p$  or of  $pq$ . They demonstrated that knowing  $\min\{s/(r+s), 2(r-s)/(r+s)\}$ -fraction of  $p$  is sufficient to factor  $N$ . Wang et al. [19] showed further improvement on required known bits of  $p$  or  $q$  for factoring  $N = p^r q^s$ .

We revisit and handle the factoring with known bits problem by solving generalized bivariate integer polynomial equations based on the lattice-based technique. Instead of solving modular equations (i.e., the modular method for short), we handle the problem by solving integer equations (i.e., the integer method for short). Previous factoring attacks such as [16], [18], [20], [21], and [22] on general RSA moduli with known bits other than Coppersmith's original work [23] are based on the modular method. Conversely, we further exploit the power of the integer method to present a unifying attack strategy on factoring  $N = p^r q^s$  with known bits.

The subsequent analyses restrict our attack scenarios 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. We have the following reasonable preconditions on the integer exponents  $r$  and  $s$  to simplify our analyses.

- We know  $r$  and  $s$ , otherwise an exhaustive search in time  $\mathcal{O}(\log^2 N)$  recovers them.
- We have  $1 \leq s \leq r \ll \log p$ , otherwise we can exchange  $p$  and  $q$ .
- We have  $\gcd(r, s) = 1$ , otherwise we try to factor another  $N^* = p^{r^*} q^{s^*}$  for  $r^* = r/\gcd(r, s)$  and  $s^* = s/\gcd(r, s)$ .

More precisely, we aim to factor  $N = p^r q^s$  for  $r \geq s \geq 1$  with some known MSBs denoted by  $P$  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. In the proposed integer method, we aim to solve several integer equations like  $(P+x)^r(Q+y)^s - N = 0$  when performing factoring attacks on  $N = p^r q^s$  with  $P$  and  $Q$ . Firstly, we show that most previous results can be obtained through the integer method. In fact, the modular method is preferable when  $s$  is small (down to 1) or  $s$  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 the most suitable  $(r, s)$  pairs for various  $r$ 's and  $s$ 's when using the integer method.

Our results are extensions of Coppersmith's work [23] via the integer method, as well as a refinement of previous solutions to the factoring with known bits problem. A direct application 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. [15] considering optimal efficiency for a roughly fixed sum of the exponents. We show that some moduli like  $p^3 q^2$  and  $p^5 q^3$  are more vulnerable to the integer method. Furthermore, a unifying condition on the desired amount of the prime leakage is derived. Informally speaking, knowing a fraction

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

of  $p$  is sufficient to factor  $N = p^r q^s$  for primes  $p, q$  of the same bit-size and coprime integers  $r > s$ .

The rest is organized as follows. We review basic definitions and a crucial theorem employed in the integer method in Section II. Subsequently, two improved theorems are developed for the factoring with known bits problem. We propose several factoring attacks using known MSBs in both primes (i.e.,  $P$  and  $Q$ ) or in only one prime (i.e.,  $P$  or  $Q$ ) in Section III. In Section IV, the theoretical results are compared and discussed in detail to obtain a unifying attack strategy. We conduct validation experiments for practical attacks and provide experimental results in Section V. Section VI concludes the paper.

**II. SOLVING GENERALIZED BIVARIATE INTEGER EQUATIONS**

We first review basic definitions involved in the integer method and then state a crucial theorem. After that, we propose two improved theorems for solving specific bivariate integer equations in our attack scenarios. We note that the detailed lattice conception is not mentioned to simplify the analysis in this paper. More information can be found in [2], [5], [6], and [24].

An irreducible integer polynomial  $f(x, y)$  implies that we must have  $|g(x, y)| = |h(x, y)| = 1$  if  $f(x, y)$  can be expressed as the product of two integer polynomials  $g(x, y)$  and  $h(x, y)$ . There exists an index set for any monomial set  $M$  in variables  $x$  and  $y$ , which is  $\mathcal{I}_M := \{(i, j) \in \mathbb{N}^2 : x^i y^j \in M\}$ . Its corresponding convex hull is  $\text{ch}\{(i, j) \in \mathbb{N}^2 : x^i y^j \in M\}$  and the Newton polygon for  $f(x, y)$  is

$$N(f) := \text{ch}\{(i, j) \in \mathbb{N}^2 : c_{ij} \neq 0\}. \tag{2}$$

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. Its  $l_p$ -norm is defined as

$$\|f(x, y)\|_p = \left(\sum |c_{ij}|^p\right)^{1/p}. \tag{3}$$

The  $l_\infty$ -norm is involved in the literature of solving integer polynomials such as [4], [5], and [7]. We point out that it can be directly deduced from the 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 definitions to guarantee that one can extract the roots of a given bivariate integer polynomial.

*Definition 1 [5]:* 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 we have  $g(x, y) = h(x, y) \cdot f(x, y)$  for some polynomial  $h(x, y)$ , then  $h(x, y)$  is defined over  $S$ .

*Definition 2 [5]:* Let  $\mathcal{I}_A$  and  $\mathcal{I}_B$  be two index sets. The Minkowski sum  $\mathcal{I}_A + \mathcal{I}_B$  is defined as

$$\begin{aligned} \mathcal{I}_A + \mathcal{I}_B \\ = \{(a_1, a_2) + (b_1, b_2) : (a_1, a_2) \in \mathcal{I}_A, (b_1, b_2) \in \mathcal{I}_B\}. \end{aligned} \tag{4}$$

The first property of Definition 1 can be satisfied by  $M$  in a straightforward manner for a given integer polynomial  $f(x, y)$  and a given set  $S$ , i.e.,  $M$  such that  $\mathcal{I}_M = N(f) + \mathcal{I}_S$ . It usually leads to monomial sets  $S$  and  $M$  also satisfying the second property, i.e.,  $S$  and  $M$  are admissible for  $f(x, y)$ .

*Lemma 1 [5]:* Assume that the Newton polygon  $N(f)$  of  $f(x, y)$  is  $\{(i, j) \in \mathbb{N}^2 : 0 \leq i \leq a, 0 \leq j \leq b\}$  for positive integers  $a$  and  $b$ . Then monomial sets  $S$  and  $M$  that

correspond to two respective index sets

$$\begin{aligned} \mathcal{I}_S &= \{(i, j) \in \mathbb{N}^2 : 0 \leq i \leq \gamma k, 0 \leq j \leq k\}, \\ \mathcal{I}_M &= \{(i, j) \in \mathbb{N}^2 : 0 \leq i \leq \gamma k + a, 0 \leq j \leq k + b\} \end{aligned} \tag{5}$$

are admissible for  $f(x, y)$ , where  $k \in \mathbb{N}$  controls low order error terms and  $\gamma > 0$  optimizes the solving bound.

*Lemma 2 [5]:* Assume that the Newton polygon  $N(f)$  of  $f(x, y)$  is  $\{(i, j) \in \mathbb{N}^2 : 0 \leq i \leq cj/d, 0 \leq j \leq d\}$  for positive integers  $c$  and  $d$ . Then monomial sets  $S$  and  $M$  that correspond to two respective index sets

$$\begin{aligned} \mathcal{I}_S &= \{(i, j) \in \mathbb{N}^2 : 0 \leq i \leq \gamma k, 0 \leq j \leq k\} \\ &\cup \{(\gamma k + i, j) \in \mathbb{N}^2 : 0 \leq i \leq cj/d, 0 \leq j \leq k\}, \\ \mathcal{I}_M &= \{(i, j) \in \mathbb{N}^2 : 0 \leq i \leq \gamma k, 0 \leq j \leq k + d\} \\ &\cup \{(\gamma k + i, j) \in \mathbb{N}^2 : 0 \leq i \leq cj/d, 0 \leq j \leq k + d\} \end{aligned} \tag{6}$$

are admissible for  $f(x, y)$ , where  $k \in \mathbb{N}$  controls low order error terms and  $\gamma > 0$  optimizes the solving bound.

See [5, Lemma 7] for the proofs. Blömer-May theorem for extracting possible roots of bivariate integer polynomials is stated as follows.

*Theorem 1 [5]:* Let  $f(x, y)$  be an irreducible integer polynomial in  $x, y$  with degree at most  $d_x, d_y \geq 1$ , respectively. Let  $X, Y$  denote the upper bounds on possible root  $(x', y')$ ,  $W$  denote  $\|f(xX, yY)\|_\infty$ , and  $S, M$  such that  $S \subseteq M$  be two admissible monomial sets for  $f(x, y)$ . Set

$$s := |S|, \quad m := |M|, \quad s_x := \sum_{x^i y^j \in M \setminus S} i, \quad s_y := \sum_{x^i y^j \in M \setminus S} j. \tag{7}$$

All possible  $(x', y')$  satisfying  $f(x', y') = 0$  can be extracted in time polynomial in  $m, d_x, d_y$ , and  $\log W$  provided  $X^{s_x} Y^{s_y} < W^s$ , assuming that  $(m - s)^2 = \mathcal{O}(sd_x d_y)$  is satisfied.

We omit low order terms since the increasing factor of running time is a constant and one may refer to [5, Section 5] for a detailed lattice-based proof. However, Theorem 1 cannot directly apply to factoring general RSA moduli with known bits. We embody Blömer-May theorem in two improved theorems for solving generalized integer polynomials.

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

$$X^{b\gamma^2 + 2a\gamma} Y^{2b\gamma + a} < W^{2\gamma} \tag{8}$$

for an optimizing parameter  $\gamma > 0$ . Furthermore, by setting  $X = N^{\delta_1}, Y = N^{\delta_2}, W = N^\alpha$ , the above exponential inequality leads to a bound  $(\alpha - a\delta_1 - b\delta_2)^2 - ab\delta_1\delta_2 > 0$  for  $\gamma = (\alpha - a\delta_1 - b\delta_2)/(b\delta_1)$ .

*Proof:* Note that  $f(x, y)$  is an irreducible polynomial of Newton polygon  $N(f) = \{(i, j) \in \mathbb{N}^2 : 0 \leq i \leq a, 0 \leq j \leq b\}$ .

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

$$\begin{aligned} S &= \{x^i y^j : 0 \leq i \leq \gamma k, 0 \leq j \leq k\}, \\ M &= \{x^i y^j : 0 \leq i \leq \gamma k + a, 0 \leq j \leq k + b\}, \end{aligned} \quad (9)$$

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.

$$s = \sum_{j=0}^k \sum_{i=0}^{\gamma k} 1 = \gamma k^2 + o(k^2), \quad (10)$$

$$m = \sum_{j=0}^{k+b} \sum_{i=0}^{\gamma k+a} 1 = \gamma k^2 + o(k^2), \quad (11)$$

$$s_x = \sum_{j=0}^{k+b} \sum_{i=0}^{\gamma k+a} i - \sum_{j=0}^k \sum_{i=0}^{\gamma k} i = \frac{b\gamma^2 + 2a\gamma}{2} k^2 + o(k^2), \quad (12)$$

$$s_y = \sum_{j=0}^{k+b} \sum_{i=0}^{\gamma k+a} j - \sum_{j=0}^k \sum_{i=0}^{\gamma k} j = \frac{2b\gamma + a}{2} k^2 + o(k^2). \quad (13)$$

Substituting them in  $X^{s_x} Y^{s_y} < W^s$  (omitting lower order terms  $o(k^2)$  for simplicity) gives

$$X^{\frac{b\gamma^2 + 2a\gamma}{2} k^2} Y^{\frac{2b\gamma + a}{2} k^2} < W^{\gamma k^2}, \quad (14)$$

which leads to

$$X^{b\gamma^2 + 2a\gamma} Y^{2b\gamma + a} < W^{2\gamma}. \quad (15)$$

Additionally, we have  $d_x = a, d_y = b$  and hence  $(m-s)^2 = \mathcal{O}(s d_x d_y) = \mathcal{O}(k^2)$  is satisfied. The time complexity is mainly dominated by  $\log W$  since  $a, b \ll \log W$  and  $k = \log W$ . Thus, the running time is a polynomial regarding  $\log W$ .

Moreover, by setting  $X = N^{\delta_1}, Y = N^{\delta_2}, W = N^\alpha$ , we obtain  $(b\gamma^2 + 2a\gamma)\delta_1 + (2b\gamma + a)\delta_2 < 2\gamma\alpha$  if considering the exponents over  $N$ . We have  $b\delta_1\gamma^2 + 2(a\delta_1 + b\delta_2 - \alpha)\gamma + a\delta_2 < 0$ , which reduces to  $(\alpha - a\delta_1 - b\delta_2)^2 - ab\delta_1\delta_2 > 0$  for  $\gamma = (\alpha - a\delta_1 - b\delta_2)/(b\delta_1)$ .  $\square$

**Theorem 3:** Given  $f(x, y) = (x + \tilde{x})^c y^d - N$ , where  $c, d$  are two positive integers,  $N$  is a known composite integer, and  $\tilde{x}$  is an approximations of  $x$ . Let  $X, Y$  denote the upper bounds on roots  $(x', y')$  and set  $W := \|f(xX, yY)\|_\infty$ . All possible  $(x', y')$  satisfying  $f(x', y') = 0$  can be extracted in time polynomial in  $\log W$  if

$$X^{(d\gamma+c)^2} Y^{2d(d\gamma+c)} < W^{2d\gamma+c} \quad (16)$$

for an optimizing parameter  $\gamma > 0$ . Furthermore, by setting  $X = N^{\delta_1}, Y = N^{\delta_2}, W = N^\alpha$ , the above exponential inequality leads to a bound  $(\alpha - d\delta_2)^2 - c\alpha\delta_1 > 0$  for  $\gamma = (\alpha - c\delta_1 - d\delta_2)/(d\delta_1)$ .

*Proof:* Note that  $f(x, y)$  is an irreducible polynomial of Newton polygon  $N(f) = \{(i, j) \in \mathbb{N}^2 : 0 \leq i \leq cj/d, 0 \leq j \leq d\}$ . We can construct two admissible sets  $S$  and  $M$  such that

$S \subseteq M$  according to Lemma 2,

$$\begin{aligned} S &= \{x^i y^j : 0 \leq i \leq \gamma k, 0 \leq j \leq k\} \\ &\cup \{x^{\gamma k+i} y^j : 0 \leq i \leq cj/d, 0 \leq j \leq k\}, \\ M &= \{x^i y^j : 0 \leq i \leq \gamma k, 0 \leq j \leq k + d\} \\ &\cup \{x^{\gamma k+i} y^j : 0 \leq i \leq cj/d, 0 \leq j \leq k + d\}, \end{aligned} \quad (17)$$

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.

$$s = \sum_{j=0}^k \sum_{i=0}^{\gamma k} 1 + \sum_{j=0}^k \sum_{i=0}^{cj/d} 1 = (\gamma + \frac{c}{2d})k^2 + o(k^2), \quad (18)$$

$$m = \sum_{j=0}^{k+d} \sum_{i=0}^{\gamma k} 1 + \sum_{j=0}^{k+d} \sum_{i=0}^{cj/d} 1 = (\gamma + \frac{c}{2d})k^2 + o(k^2), \quad (19)$$

$$s_x = \sum_{j=0}^{k+d} \sum_{i=0}^{\gamma k} i + \sum_{j=0}^{k+d} \sum_{i=0}^{cj/d} (\gamma k + i) \quad (20)$$

$$- \sum_{j=0}^k \sum_{i=0}^{\gamma k} i - \sum_{j=0}^k \sum_{i=0}^{cj/d} (\gamma k + i) \quad (21)$$

$$= \frac{(d\gamma + c)^2}{2d} k^2 + o(k^2), \quad (22)$$

$$s_y = \sum_{j=0}^{k+d} \sum_{i=0}^{\gamma k} j + \sum_{j=0}^{k+d} \sum_{i=0}^{cj/d} j - \sum_{j=0}^k \sum_{i=0}^{\gamma k} j - \sum_{j=0}^k \sum_{i=0}^{cj/d} j \quad (23)$$

$$= (d\gamma + c)k^2 + o(k^2). \quad (24)$$

Substituting them in  $X^{s_x} Y^{s_y} < W^s$  gives

$$X^{\frac{(d\gamma+c)^2}{2d} k^2} Y^{(d\gamma+c)k^2} < W^{(\gamma + \frac{c}{2d})k^2}, \quad (25)$$

which reduces to

$$X^{(d\gamma+c)^2} Y^{2d(d\gamma+c)} < W^{2d\gamma+c}. \quad (26)$$

Furthermore, we have  $d_x = c, d_y = d$ , and hence  $(m-s)^2 = \mathcal{O}(s d_x d_y) = \mathcal{O}(k^2)$  is satisfied. The time complexity is mainly dominated by  $\log W$  since  $a, b \ll \log W$  and  $k = \log W$ . Thus, the running time is a polynomial regarding  $\log W$ .

Moreover, by setting  $X = N^{\delta_1}, Y = N^{\delta_2}, W = N^\alpha$ , we obtain  $(d\gamma + c)^2\delta_1 + 2d(d\gamma + c)\delta_2 < (2d\gamma + c)\alpha$  if considering the exponents over  $N$ . We have  $d^2\delta_1\gamma^2 + 2d(c\delta_1 + d\delta_2 - \alpha)\gamma + c^2\delta_1 + 2cd\delta_2 - c\alpha < 0$ , which reduces to  $(\alpha - d\delta_2)^2 - c\alpha\delta_1 > 0$  for  $\gamma = (\alpha - c\delta_1 - d\delta_2)/(d\delta_1)$ .  $\square$

### III. APPLICATION TO FACTORING WITH KNOWN BITS

We propose several attacks to factor  $N$  with known MSBs, namely  $P$  and  $Q$ . Let us first specify the attack scenarios. 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  $X = Y = N^\eta$ , we aim to efficiently recover  $p$  and  $q$  leading to the factorization of  $N$  under minimal requirements 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 obtain the attack results by applying above improved theorems via the integer method. To do so, we should derive some integer equations from the above attack scenarios. The suitable integer equations are divided into two parts as follows. The first part is involved with two approximations that consists of solving  $(P+x)^r(Q+y)^s - N = 0$  and  $(PQ+x)^s y - N = 0$ . The second part is related to only one approximation, which consists of solving  $(P+x)^r y - N = 0$ . Before presenting the analyses, 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  $r, s \geq 1$  and primes  $p, q$  are of the same bit-size. Given an MSBs approximation  $P$  of  $p$  for  $|p - P| < N^\eta$ , the rounding integer  $Q := \lfloor (N/P^r)^{1/s} \rfloor$  is an MSBs approximation of  $q$  satisfying  $|q - Q| < N^\eta$ .

*Proof:* Because  $r, s$  are negligible compared to  $p$  and  $q$ , we assume  $p, q$  and  $P$  are roughly equal to  $N^{1/(r+s)}$  and thus  $Q$  is also roughly equal to  $N^{1/(r+s)}$ . To bound  $|q - Q|$ , we first bound the value of  $|q^s - Q^s|$  since we have

$$|q - Q| = \frac{|q^s - Q^s|}{q^{s-1} + q^{s-2}Q + \dots + Q^{s-1}} \approx \frac{|q^s - Q^s|}{sN^{\frac{s-1}{r+s}}}. \quad (27)$$

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

$$|q^s - Q^s| \approx |q^s - \frac{N}{P^r}| = \frac{q^s |P^r - p^r|}{P^r} \approx |P^r - p^r| N^{\frac{s-r}{r+s}}. \quad (28)$$

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

$$|P^r - p^r| = |P - p|(P^{r-1} + P^{r-2}p + \dots + p^{r-1}) < rN^{\frac{r-1}{r+s} + \eta}. \quad (29)$$

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

$$|q - Q| \approx \frac{|q^s - Q^s|}{N^{\frac{s-1}{r+s}}} \approx \frac{|P^r - p^r| N^{\frac{s-r}{r+s}}}{N^{\frac{s-1}{r+s}}} < \frac{N^{\frac{r-1}{r+s} + \eta} N^{\frac{s-r}{r+s}}}{N^{\frac{s-1}{r+s}}} = N^\eta, \quad (30)$$

which terminates the proof.  $\square$

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

### A. USING TWO APPROXIMATIONS

We present the results in theorems derived from solving bivariate integer equations. More concretely, we try to solve  $(P+x)^r(Q+y)^s - N = 0$  and  $(PQ+x)^s y - N = 0$  to obtain the solution to the factoring with known bits problem. We have 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  $N = p^r q^s$ .

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

$$\frac{\sqrt{rs}}{r+s-1+\sqrt{rs}} \quad (31)$$

of MSBs 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

$$X^{s\gamma^2+2r\gamma} Y^{2s\gamma+r} < W^{2\gamma}. \quad (32)$$

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

$$\begin{aligned} W &= \|f(xX, yY)\|_\infty \\ &= \max\{|P^{r-1}XQ^s|, |P^rQ^{s-1}Y|, |P^rQ^s - N|\} \\ &= N^{\frac{r+s-1}{r+s} + \eta}. \end{aligned} \quad (33)$$

Considering the exponents in the condition, it leads to

$$\eta(s\gamma^2 + 2r\gamma + 2s\gamma + r) < 2\gamma \left( \frac{r+s-1}{r+s} + \eta \right), \quad (34)$$

which further reduces to

$$\eta < \frac{2(r+s-1)\gamma}{(r+s)(s\gamma^2 + 2(r+s-1)\gamma + r)}. \quad (35)$$

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

$$\eta < \frac{r+s-1}{(r+s)(r+s-1+\sqrt{rs})}. \quad (36)$$

A fraction  $1 - (r+s)\eta$  is required, which implies that at least a fraction

$$1 - (r+s) \frac{r+s-1}{(r+s)(r+s-1+\sqrt{rs})} = \frac{\sqrt{rs}}{r+s-1+\sqrt{rs}} \quad (37)$$

of  $p$  and  $q$  is required. The time complexity is polynomial in  $\log W$ , and it is also polynomial in  $\log N$ .  $\square$

We have another integer equation  $(PQ+x)^s y - N = 0$  based on 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.

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

$$\frac{2(r-s)}{r+s} \quad (38)$$

of MSBs 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

$$X^{(\gamma+s)^2} Y^{2(\gamma+s)} < W^{2\gamma+s}. \quad (39)$$

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

$$W = \|f(xX, yY)\|_{\infty} = \max\{|(PQ)^s Y|, |N|\} = N. \quad (40)$$

From the condition, we have

$$(\gamma + s)^2 \left( \frac{1}{r+s} + \eta \right) + \frac{2(r-s)}{r+s}(\gamma + s) < 2\gamma + s, \quad (41)$$

which reduces to

$$\eta < \frac{-\gamma^2 + 2s\gamma + 2s^2 - rs}{(r+s)(\gamma + s)^2}. \quad (42)$$

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

$$\eta < \frac{3s-r}{(r+s)^2}. \quad (43)$$

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

$$1 - (r+s) \frac{3s-r}{(r+s)^2} = \frac{2(r-s)}{r+s} \quad (44)$$

of  $p$  is required to recover  $p$  and then factor  $N$ . The time complexity is polynomial in  $\log W$ , and thus is polynomial in  $\log N$ .  $\square$

Besides, we can solve  $(PQ+x)^s y^{r-s} - N = 0$  and  $(PQ+x)^s (P+y)^{r-s} - N = 0$ . 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  $X = N^{\frac{1}{r+s}+\eta}$ ,  $Y = N^{\frac{r-s}{r+s}}$ , and  $W = N$ . Setting  $\gamma = 1/2$  in the proof to obtain

$$\eta < \frac{3s-r}{(r+s)^2}. \quad (45)$$

It reduces to the same result that we require a fraction at least

$$1 - (r+s) \frac{3s-r}{(r+s)^2} = \frac{2(r-s)}{r+s}. \quad (46)$$

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 implies that we need at least a fraction

$$\frac{\sqrt{s^2(r-s)^2 + 8s(r-s)(r-1)^2 + s(r-s)}}{\sqrt{s^2(r-s)^2 + 8s(r-s)(r-1)^2 + s(r-s) + 2(r-1)^2}} \quad (47)$$

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

### B. 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 Section III-A. But we observe that  $W$  decreases when taking both  $P$  and  $Q$  into consideration and it may weaken the bound on  $\eta$ . Therefore, we try to explore the factoring attacks only with the help of  $P$  or  $Q$ . More concretely, we try to solve  $(P+x)^r y - N = 0$  without the knowledge of  $Q$ .

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

$$\frac{s}{r+s} \quad (48)$$

of MSBs 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

$$X^{(\gamma+r)^2} Y^{2(\gamma+r)} < W^{2\gamma+r}, \quad (49)$$

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

$$\eta(\gamma+r)^2 + \frac{2s}{r+s}(\gamma+r) < 2\gamma+r. \quad (50)$$

It reduces to

$$\eta < \frac{2r\gamma + r^2 - rs}{(r+s)(\gamma+r)^2}. \quad (51)$$

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

$$\eta < \frac{r}{(r+s)^2}. \quad (52)$$

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

$$1 - (r+s) \frac{r}{(r+s)^2} = \frac{s}{r+s} \quad (53)$$

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

Similarly, we can solve  $(P+x)^r y^s - N = 0$  via 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 and obtain

$$\eta < \frac{r}{(r+s)^2}. \quad (54)$$

It results in the same result as that in Theorem 6.

When we consider using one approximation  $P$  or  $Q$ , there also exist two integer equations  $(Q+x)^s y - N = 0$  and  $(Q+x)^s y^r - N = 0$ . 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

$$\eta < \frac{s}{(r+s)^2}, \quad (55)$$

which means that a fraction at least

$$1 - (r+s) \frac{s}{(r+s)^2} = \frac{r}{r+s} \quad (56)$$

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

**TABLE 1.** The comparison of our proposed attacks with existing results against schemes using RSA moduli  $N = p^r q^s$ .

Related Work	RSA Modulus $N$	Bound on Known Prime Fraction	Used Method	Restriction
Rivest and Shamir [11]	$pq$	$2/3$	integer programming	–
Coppersmith [2]	$pq$	$1/2$	lattice-based method	–
Herrmann and May [25]	$pq$	$\ln 2 \approx 70\%$	lattice-based method	discrete leaked prime bit-blocks
Boneh et al. [14]	$p^r q$	$1/(r + 1)$	lattice-based method	–
Lu et al. [20]	$p^r q$	$\ln(r + 1)/r$	lattice-based method	discrete leaked prime bit-blocks
Coron and Zeitoun [17]	$p^r q^s$	0	lattice-based method	$r \geq \log p$
Lu et al. [18]	$p^r q^s$	$\min \left\{ \frac{s}{r+s}, \frac{2(r-s)}{r+s} \right\}$	lattice-based method	–
Wang et al. [19]	$p^r q^s$	$(su - rv)/(r + s)$	lattice-based method	$u/r > v/s$
Ours	$p^r q^s$	$\min \left\{ \frac{s}{r+s}, \frac{\sqrt{rs}}{r+s-1+\sqrt{rs}}, \frac{2(r-s)}{r+s} \right\}$	lattice-based method	–

**IV. COMPARISON AND DISCUSSIONS**

We show the comparison of our proposed attacks with existing techniques against schemes using RSA moduli  $N = p^r q^s$  in Table 1. Our work is superior based on the comparison and covers several previous results.

Since the modular method is more efficient and simpler for some specific equations, solving modular equations are preferred when the same or even better attack results can be obtained. However, taking Theorem 4, Theorem 5, and Theorem 6 into consideration, the integer method shows its power for solving a generalized bivariate integer equation  $(P + x)^r(Q + y)^s - N = 0$ , which is involved in Theorem 4.

We compare the required amounts of known MSBs derived from the integer method in Section III to conclude a unifying condition since the fractions of desired known bits differ when solving distinct integer equations. Our theoretical results and the unifying condition to factoring general RSA moduli with known bits are showed in Fig. 1. The respective fractions required for factoring general RSA moduli  $N = p^r q^s$  with known bits and the corresponding solvable integer equations are summarized as follows.

- For the solvable equation  $(P + x)^r(Q + y)^s - N = 0$  with  $r \geq s \geq 1$ , the required fraction given via Theorem 4 is

$$\frac{\sqrt{rs}}{r + s - 1 + \sqrt{rs}}. \tag{57}$$

- For the solvable equations  $(PQ + x)^s y - N = 0$  and  $(PQ + x)^s y^{r-s} - N = 0$  with  $1 \leq s < r < 3s$ , the required fraction given via Theorem 5 is

$$\frac{2(r - s)}{r + s}. \tag{58}$$

- For the solvable equations  $(P + x)^r y - N = 0$  and  $(P + x)^r y^s - N = 0$  with  $r \geq s \geq 1$ , the required fraction given via Theorem 6 is

$$\frac{s}{r + s}. \tag{59}$$

We discuss more the unifying condition. For  $N = pq$  with  $r = s = 1$ , we can apply Theorem 4 and Theorem 6. Our results cover that of [1] 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 [14] and [18]. 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. 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 2. It is showed that the best choice actually depends on both  $r, s$  and their relation.

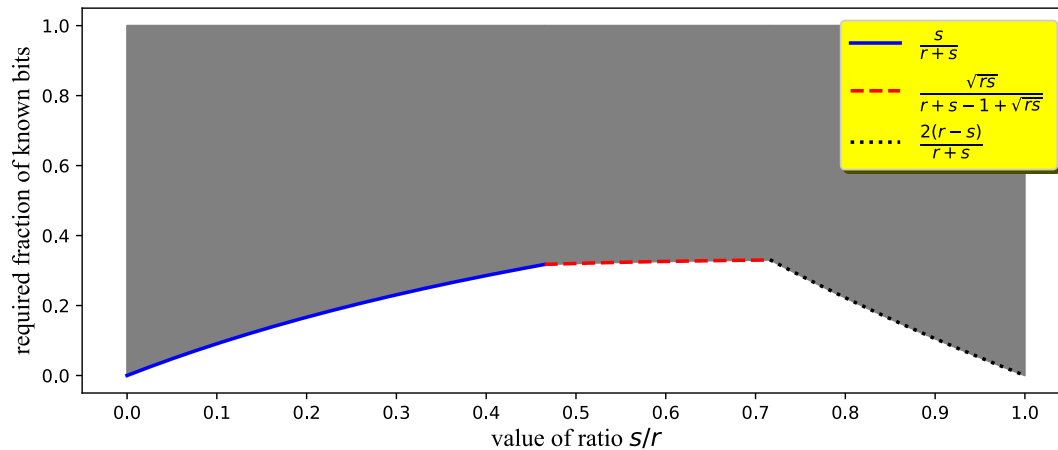
To be concrete, Theorem 4 is preferred for medium  $s$  for a fixed  $r$ . Theorem 6 is more effective for small  $s$  like  $s = 1$  and Theorem 5 works better for large  $s$  like  $s = r - 1$ . Furthermore, we identify the respective applicable ranges of  $s$  along with the most suitable solvable equations for each theorem in Table 3. 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 further define two functions  $\theta(r)$  and  $\xi(r)$  for simplicity since the explicit forms are complicated to express.

*Definition 3:* Given a positive integer  $r$ , let  $\theta(r)$  be the unique real root in  $(0, 1)$  of the following equation

$$\frac{\sqrt{xr}}{r + xr - 1 + \sqrt{xr}} = \frac{xr}{r + xr}, \tag{60}$$

which can be explicitly expressed as

$$x = \left( \frac{\sqrt[3]{27r^3 + \sqrt{(729r^6 + 108(r - 1)^3 r^3)}}}{3\sqrt[3]{2r}} - \frac{\sqrt[3]{2}(1 - r)}{\sqrt[3]{27r^3 + \sqrt{(729r^6 + 108(r - 1)^3 r^3)}}} \right)^2. \tag{61}$$



**FIGURE 1.** The horizontal and vertical axes denote the value of ratio  $s/r$  and required fraction of known bits, respectively. The grey area indicates factoring attacks derived by solving generalized bivariate integer equations. It is regarded as a unifying solution to the factoring with known bits problem on general RSA moduli.

**TABLE 2.** The values of respective fractions for several  $(r, s)$  pairs.

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

**TABLE 3.** The respective applicable ranges of  $s$  according to the proposed theorems.

Required Fraction	Applicable Range	Suitable Equation	Theorem
$s/(r + s)$	$1 \leq s \leq \theta(r) \cdot r$	$(P + x)^r y - N = 0$	Theorem 6
$\sqrt{rs}/(r + s - 1 + \sqrt{rs})$	$\theta(r) \cdot r < s \leq \xi(r) \cdot r$	$(P + x)^r (Q + y)^s - N = 0$	Theorem 4
$2(r - s)/(r + s)$	$\xi(r) \cdot r < s < r$	$(PQ + x)^s y - N = 0$	Theorem 5

**TABLE 4.** The values of  $\theta(r)$  and  $\xi(r)$  for various  $r \leq 9$ .

$r$	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

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

$$\frac{\sqrt{xr}}{r + xr - 1 + \sqrt{xr}} = \frac{2(r - xr)}{r + xr}, \quad (62)$$

which cannot be explicitly expressed but can be calculated by numerical methods.

We list the numerical values of  $\theta(r)$  and  $\xi(r)$  for  $r \leq 9$  in Table 4. 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. To explicitly understand our proposed factoring attacks, we list the theoretical required minimum number of prime bits for factoring various moduli  $N = p^r q^s$  using the unifying condition in Table 5. We let both  $p$  and  $q$  be two  $\ell$ -bit primes for  $\ell = 512, 1024,$  and  $2048$  to make the illustration more realistic.

Though our proposed factoring attacks run in polynomial time, we further analyze the attack complexity. As our attacks are derived from the lattice-based method that rely on the LLL algorithm, the attack complexity is mainly dominated



**TABLE 5.** The theoretical required minimum leaked prime bits for factoring various moduli  $N = p^r q^s$  with  $\ell$ -bit primes.

$\ell$	$p^2q$	$p^3q$	$p^3q^2$	$p^4q$	$p^4q^3$	$p^5q$	$p^5q^2$	$p^5q^3$	$p^5q^4$	$p^6q$	$p^6q^5$
512	171	128	195	103	147	86	147	183	114	74	94
1024	342	256	389	205	293	171	293	365	228	147	187
2048	683	512	778	410	586	342	586	730	456	293	373

by the LLL algorithm [12]. We know that it shall terminate in time complexity  $\mathcal{O}(n^6 \log^3 B)$ , where  $n$  denotes the lattice dimension and  $B$  denotes the maximal Euclidean norm of lattice vectors. Assume that we aim to factor  $N = p^r q^s$  with known bits for  $\ell$ -bit primes and the proposed factoring attacks are conducted using an  $n$ -dimensional lattice, the maximal Euclidean norm  $B$  of lattice vectors is approximate  $p^{\frac{3}{\sqrt{n}}}$  due to our lattice construction. Hence, the attack complexity is  $\mathcal{O}(n^6 \log^3 B) \approx \mathcal{O}(n^6 (\sqrt[3]{n} \log p)^3) = \mathcal{O}(n^7 \ell^3)$ . The attack complexity is a roughly estimated upper bound since the LLL algorithm works better in practice.

We want to give a more accurate estimation of the execution time of our proposed factoring attacks based on the attack complexity. Considering the computing power of modern personal computers and the execution time of fundamental operations, we estimate that the complexity  $\mathcal{O}(10^{18})$  can be completed in one second. The execution time (recorded in seconds) and its corresponding attack complexity for conducting factoring attacks using an  $n$ -dimensional lattice are estimated in Table 6. Both  $p$  and  $q$  are assumed to be two 512-bit primes for simplicity. The symbol ‘ $\sim$ ’ indicates that the execution time is at the given magnitude. Based on the observation of the estimated execution time, we would like to use a lattice whose dimension is less than 100 in our experiments for efficient validation.

*Example 1:* We provide the concrete choices for several RSA moduli with  $1 \leq r, s \leq 9$  with respect to solvable integer equations as follows.

- We choose to solve  $(P + x)^r y - N = 0$  for factoring  $pq, p^2q, p^3q, p^4q, p^5q, p^5q^2, p^6q, p^7q, p^7q^2, p^7q^3, p^8q, p^8q^3, p^9q, p^9q^2, p^9q^4$  with known bits.
- We choose to solve  $(P + x)^r (Q + y)^s - N = 0$  for factoring  $p^3q^2, p^5q^3, p^7q^4, p^8q^5, p^9q^5$  with known bits.
- We choose to solve  $(PQ + x)^s y - N = 0$  for factoring  $p^4q^3, p^5q^4, p^6q^5, p^7q^5, p^7q^6, p^8q^7, p^9q^7, p^9q^8$  with known bits.

*Proposition 1:* We provide a unifying attack strategy for factoring general RSA moduli  $N = p^r q^s$  with known bits with respect to a sufficiently large  $s$  (satisfying  $s \ll \log p$ ).

- Solve  $(PQ + x)^s y - N = 0$  for  $0.716r < s < r$ ,
- Solve  $(P + x)^r (Q + y)^s - N = 0$  for  $0.465r < s \leq 0.716r$ ,
- Solve  $(P + x)^r y - N = 0$  for else cases.

### V. VALIDATION EXPERIMENTS

We provide the experimental results to check the validity of our proposed factoring attacks according to Theorem 4, Theorem 5, and Theorem 6, respectively. The experiments were conducted under Windows 10 running on a computer with 3.10GHz CPU and 8 GB RAM. We utilized the LLL algorithm available in SageMath [26]. The RSA instances were generated uniformly at random. To simulate practical factoring attacks on general RSA moduli  $N = p^r q^s$  with known bits, we first randomly generated two  $\ell$ -bit primes. Then we calculated  $N = p^r q^s$  for given parameters  $r$  and  $s$ . The amount of known bits in MSBs of primes  $p$  and  $q$  was assumed as  $u$  and hence the exposed MSBs, i.e.,  $P$  and  $Q$  were computed based on  $p, q$  and  $u$ . Finally, we could construct the above solvable integer equations like  $(P + x)^r y - N = 0, (PQ + x)^s y - N = 0$ , and  $(P + x)^r (Q + y)^s - N = 0$ .

During the experiments, we chose a proper lattice setting for conducting the proposed factoring attacks. We could collect many polynomial equations satisfying our solvable requirements and hence extract the desired root, i.e., the unknown part of the primes and then factor the given RSA moduli. The experimental results of our proposed factoring attacks are shown in Table 7. The ‘ $u$ ’-column provides the experimental number of bits leading to successful factoring attacks. The ‘ $u_t$ ’-column provides the theoretical required number of bits for conducting the proposed factoring attacks as stated in Table 5. The ‘Theorem’-column and ‘Equation’-column provides the specific theorem and solvable integer equation we used for given practical instance in our factoring attacks. The corresponding lattice dimension is denoted by ‘ $n$ ’ and the running time is ‘Time’ (recorded in seconds).

We collected enough integer polynomials having the common root in each experiment. We took some of them to extract the common root and obtained the correct values in the unknown part of the primes, namely  $x' = p - P$  or  $y' = q - Q$ . Thus,  $p = P + x'$  and  $q = Q + y'$  finally led to the factorization of  $N$ . Through the above experiments, we successfully verified the validity of our proposed factoring attacks. However, the experimental results are still several bits away from the theoretical ones when comparing  $u$  with  $u_t$  in Table 7. The reason may be that the lattice dimension is not large enough due to the limitation of computing resources.

From the observation of Table 7, we find that Theorem 4 works more efficient than Theorem 5 and Theorem 6. Besides, Theorem 5 has the worst performance. More

**TABLE 6.** The execution time and its corresponding attack complexity for conducting factoring attacks with 512-bit primes using an  $n$ -dimensional lattice.

$n$	20	30	40	50	70	100	200
Time	$\sim 0.1$ s	$\sim 1$ s	$\sim 10$ s	$\sim 100$ s	$\sim 1000$ s	$\sim 10\,000$ s	$\sim 1\,000\,000$ s
Complexity	$\mathcal{O}(10^{17})$	$\mathcal{O}(10^{18})$	$\mathcal{O}(10^{19})$	$\mathcal{O}(10^{20})$	$\mathcal{O}(10^{21})$	$\mathcal{O}(10^{22})$	$\mathcal{O}(10^{24})$

**TABLE 7.** The experimental results of our proposed factoring attacks.

$\ell$	$r$	$s$	$N$	$u$	$u_t$	Theorem	Solvable Equation	$n$	Time
512	2	1	$p^2q$	223	171	Theorem 6	$(P+x)^r y - N = 0$	45	25 s
1024	2	1	$p^2q$	446	342	Theorem 6	$(P+x)^r y - N = 0$	45	83 s
2048	2	1	$p^2q$	893	683	Theorem 6	$(P+x)^r y - N = 0$	45	323 s
512	3	1	$p^3q$	205	128	Theorem 6	$(P+x)^r y - N = 0$	55	209 s
1024	3	1	$p^3q$	410	256	Theorem 6	$(P+x)^r y - N = 0$	55	837 s
2048	3	1	$p^3q$	819	512	Theorem 6	$(P+x)^r y - N = 0$	55	3004 s
512	3	2	$p^3q^2$	237	195	Theorem 4	$(P+x)^r(Q+y)^s - N = 0$	91	1147 s
1024	3	2	$p^3q^2$	474	389	Theorem 4	$(P+x)^r(Q+y)^s - N = 0$	91	3748 s
2048	3	2	$p^3q^2$	949	778	Theorem 4	$(P+x)^r(Q+y)^s - N = 0$	91	23 254 s
512	4	1	$p^4q$	180	103	Theorem 6	$(P+x)^r y - N = 0$	65	577 s
1024	4	1	$p^4q$	359	205	Theorem 6	$(P+x)^r y - N = 0$	65	2276 s
2048	4	1	$p^4q$	720	410	Theorem 6	$(P+x)^r y - N = 0$	65	10 198 s
512	4	3	$p^4q^3$	304	147	Theorem 5	$(PQ+x)^s y - N = 0$	54	226 s
1024	4	3	$p^4q^3$	608	293	Theorem 5	$(PQ+x)^s y - N = 0$	54	1105 s
2048	4	3	$p^4q^3$	1217	586	Theorem 5	$(PQ+x)^s y - N = 0$	54	4730 s

specifically, the experimental results of the factoring attacks induced by Theorem 4 are closest to the theoretical results. In contrast, the experimental results of the factoring attacks induced by Theorem 5 differ the most from the theoretical results. The running time meets our prediction for it as stated in Table 6. Moreover, we show the following example for numerical understanding.

*Example 2:* We provide a toy example for factoring  $N = p^2q$  with known  $P$  via Theorem 6. Two primes  $p, q$  are set of 128-bit, which implies  $\ell = 128$ . Suppose that we are given known 52-bit MSBs, which implies  $u = 52$ . Note that the theoretical result for recovering the primes is  $u_t = 43$ . The toy RSA instance is listed as follows.

$$\begin{aligned}
 N &= 2676924230283243720261014287455130479504 \\
 &\quad \backslash 4225822158941961472413312864808332757515 \\
 &\quad \backslash 020832236590800828421542339105151967 \\
 P &= 301958494768445181148100139329150517248
 \end{aligned}$$

We then derive the solvable integer equation  $(P+x)^2y - N = 0$  with known parameters  $N$  and  $P$ . We construct a 91-dimensional lattice for conducting the proposed factoring attack via Theorem 6. After nearly 37 seconds, we extract the root  $(x', y')$  satisfying the above equation. The obtained root is listed as follows.

$$\begin{aligned}
 x' &= 35811068498785421000871 \\
 y' &= 293590213774301270676454386402555434447
 \end{aligned}$$

Since we have  $x' = p - P$  and  $y' = q$ , these two primes are computed as follows.

$$\begin{aligned}
 p &= 301958494768445181148100139329150517248 \\
 &\quad + 35811068498785421000871 \\
 &= 301958494768445216959168638114571518119 \\
 q &= 293590213774301270676454386402555434447
 \end{aligned}$$

One may check that  $N = p^2q$  does hold and hence we successfully factor the given modulus  $N$ .

## VI. CONCLUSION

We revisited the factoring with known bits problem on general RSA moduli  $N = p^r q^s$  with  $r, s \geq 1$  for two primes  $p, q$  of the same bit-size. To be specific, we examined the minimum amount of known MSBs of the primes required for factoring and derived the attack results based on solving generalized bivariate integer equations. We established a unifying condition on the required fraction of known MSBs for factoring  $N = p^r q^s$ . Our analysis identified one solution as superior for certain combinations of  $(r, s)$ , such as  $p^3q^2, p^5q^3, p^7q^4, p^8q^5$ , and  $p^9q^5$ , when  $s$  is of medium size relative to  $r$ . Theoretical analysis and experimental results were provided to verify the effectiveness of our proposed factoring attacks.

We demonstrated that the integer method is more powerful as it covers the majority of results derived from the modular method and provides new solvable integer equations

for conducting factoring attacks. We hope that such integer method can be applied to other problems involving the solution of generalized bivariate integer equations and yield even better results.

## ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers whose insightful comments and constructive feedback greatly improved the quality of this manuscript.

## REFERENCES

- [1] D. Coppersmith, "Finding a small root of a bivariate integer equation; factoring with high bits known," in *Proc. Int. Conf. Theory Appl. Cryptograph. Techn.*, vol. 1070, U. M. Maurer, Ed. Saragossa, Spain: Springer, May 1996, pp. 178–189.
- [2] D. Coppersmith, "Small solutions to polynomial equations, and low exponent RSA vulnerabilities," *J. Cryptol.*, vol. 10, no. 4, pp. 233–260, 1997.
- [3] R. L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Commun. ACM*, vol. 21, no. 2, pp. 120–126, Feb. 1978.
- [4] J. Coron, "Finding small roots of bivariate integer polynomial equations revisited," in *Proc. Int. Conf. Theory Appl. Cryptograph. Techn.*, vol. 3027, C. Cachin and J. Camenisch, Eds. Interlaken, Switzerland: Springer, May 2004, pp. 492–505.
- [5] J. Blömer and A. May, "A tool kit for finding small roots of bivariate polynomials over the integers," in *Proc. 24th Annu. Int. Conf. Theory Appl. Cryptograph. Techn.*, vol. 3494, R. Cramer, Ed. Aarhus, Denmark: Springer, May 2005, pp. 251–267.
- [6] E. Jochemsz and A. May, "A strategy for finding roots of multivariate polynomials with new applications in attacking RSA variants," in *Proc. 12th Int. Conf. Theory Appl. Cryptol. Inf. Secur.*, vol. 4284, X. Lai and K. Chen, Eds. Shanghai, China: Springer, Dec. 2006, pp. 267–282.
- [7] J. Coron, "Finding small roots of bivariate integer polynomial equations: A direct approach," in *Proc. 27th Annu. Int. Cryptol. Conf.*, vol. 4622, A. Menezes, Ed. Santa Barbara, CA, USA: Springer, Aug. 2007, pp. 379–394.
- [8] A. K. Lenstra, H. W. Lenstra Jr., M. S. Manasse, and J. M. Pollard, "The number field sieve," in *Proc. 22nd Annu. ACM Symp. Theory Comput.*, H. Ortiz, Ed. Baltimore, MD, USA, May 1990, pp. 564–572.
- [9] P. C. Kocher, "Timing attacks on implementations of Diffie–Hellman, RSA, DSS, and other systems," in *Proc. 16th Annu. Int. Cryptol. Conf.*, vol. 1109, N. Koblitz, Ed. Santa Barbara, CA, USA: Springer, Aug. 1996, pp. 104–113.
- [10] J. A. Halderman, "Lest we remember: Cold-boot attacks on encryption keys," *Commun. ACM*, vol. 52, no. 5, pp. 91–98, May 2009.
- [11] R. L. Rivest and A. Shamir, "Efficient factoring based on partial information," in *Proc. Workshop Theory Appl. Cryptograph. Techn.*, vol. 219, F. Pichler, Ed. Linz, Austria: Springer, Apr. 1985, pp. 31–34.
- [12] A. K. Lenstra, H. W. Lenstra Jr., and L. Lovasz, "Factoring polynomials with rational coefficients," *Math. Ann.*, vol. 261, no. 4, pp. 515–534, Dec. 1982.
- [13] T. Takagi, "Fast RSA-type cryptosystem modulo  $p^k q$ ," in *Proc. 18th Annu. Int. Cryptol. Conf.*, vol. 1462, H. Krawczyk, Ed. Santa Barbara, CA, USA: Springer, Aug. 1998, pp. 318–326.
- [14] D. Boneh, G. Durfee, and N. Howgrave-Graham, "Factoring  $N = p^r q$  for large  $r$ ," in *Proc. 19th Annu. Int. Cryptol. Conf.*, vol. 1666, M. J. Wiener, Ed. Santa Barbara, CA, USA: Springer, Aug. 1999, pp. 326–337.
- [15] S. Lim, S. Kim, I. Yie, and H. Lee, "A generalized Takagi–Cryptosystem with a modulus of the form  $p^r q^s$ ," in *Proc. 1st Int. Conf. Cryptol.*, vol. 1977, B. K. Roy and E. Okamoto, Eds. Cham, Switzerland: Springer, Dec. 2000, pp. 283–294.
- [16] J. Coron, J. Faugère, G. Renault, and R. Zeitoun, "Factoring  $N = p^r q^s$  for large  $r$  and  $s$ ," in *Proc. Cryptographers' Track RSA Conf.*, vol. 9610, K. Sako, Ed. San Francisco, CA, USA: Springer, Mar. 2016, pp. 448–464.
- [17] J. Coron and R. Zeitoun, "Improved factorization of  $N = p^r q^s$ ," in *Proc. Cryptographers' Track RSA Conf.*, vol. 10808, N. P. Smart, Ed. San Francisco, CA, USA: Springer, Apr. 2018, pp. 65–79.
- [18] Y. Lu, L. Peng, and S. Sarkar, "Cryptanalysis of an RSA variant with Moduli  $N=p^r q^l$ ," *J. Math. Cryptol.*, vol. 11, no. 2, pp. 117–130, Jun. 2017.
- [19] S. Wang, L. Qu, C. Li, and H. Wang, "Further improvement of factoring  $N = p^r q^l$  with partial known bits," *Adv. Math. Commun.*, vol. 13, no. 1, pp. 121–135, 2019.
- [20] Y. Lu, R. Zhang, and D. Lin, "Factoring multi-power RSA modulus  $N = p^r q$  with partial known bits," in *Proc. Australas. Conf. Inf. Secur. Privacy*, vol. 7959, C. Boyd and L. Simpson, Eds. Brisbane, QLD, Australia: Springer, Jul. 2013, pp. 57–71.
- [21] M. Zheng and H. Hu, "Implicit related-key factorization problem on the RSA cryptosystem," in *Proc. Int. Conf. Cryptol. Netw. Secur.*, vol. 11829, Y. Mu, R. H. Deng, and X. Huang, Eds. Fuzhou, China: Springer, Oct. 2019, pp. 525–537.
- [22] M. Zheng, N. Kunihiro, and H. Hu, "Lattice-based cryptanalysis of RSA with implicitly related keys," *IEICE Trans. Fundam. Electron., Commun. Comput. Sci.*, vol. E103.A, no. 8, pp. 959–968, 2020.
- [23] D. Coppersmith, "Finding a small root of a univariate modular equation," in *Proc. Int. Conf. Theory Appl. Cryptograph. Techn.*, vol. 1070, U. M. Maurer, Ed. Saragossa, Spain: Springer, May 1996, pp. 155–165.
- [24] A. May, "New RSA vulnerabilities using lattice reduction methods," Ph.D. thesis, Fac. Elect. Eng., Comput. Sci. Math., Univ. Paderborn, Paderborn, Germany, 2003.
- [25] M. Herrmann and A. May, "Solving linear equations modulo divisors: On factoring given any bits," in *Proc. Int. Conf. Theory Appl. Cryptol. Inf. Secur.*, vol. 5350, J. Pieprzyk, Ed. Melbourne, VIC, Australia: Springer, Dec. 2008, pp. 406–424.
- [26] The Sage Developers. (2021). *SageMath, the Sage Mathematics Software System (Version 9.0)*. [Online]. Available: <https://www.sagemath.org>



**MENGCE ZHENG** (Member, IEEE) received the B.E. degree in information security and the Ph.D. degree in information and communication engineering from the University of Science and Technology of China, Hefei, China, in 2013 and 2018, respectively. He was a Postdoctoral Researcher with the Key Laboratory of Electromagnetic Space Information, CAS, University of Science and Technology of China, from 2019 to 2020. He is currently an Associate Professor with Zhejiang Wanli University. His research interests include cryptography and information security.



**ZHIGANG CHEN** received the B.S. degree in mathematics, the M.S. degree in computer software and theory, and the Ph.D. degree in cryptography from the Nanjing University of Aeronautics and Astronautics, in 1994, 2004, and 2015, respectively. He is a Professor with Zhejiang Wanli University. He has been working on fully homomorphic encryption, since 2011. His research interests include fully homomorphic encryption and blockchain.



**YAOHUI WU** received the B.E. and M.S. degrees in communication engineering from the Beijing University of Posts and Telecommunications, China, in 2001 and 2004, respectively, and the Ph.D. degree in communication engineering from Ningbo University, in 2018. He is currently an Associate Professor with Zhejiang Wanli University. His research interests include communication systems and security.

• • •