Theoretical Computer Science 1070 (2026) 115837

Contents lists available at ScienceDirect

Theoretical Computer Science

journal homepage: www.elsevier.com/locate/tcs

A more complete cryptanalysis of the RSA-polynomial problem

Mengce Zheng [a,*], Abderrahmane Nitaj [b]

[a] College of Information and Intelligence Engineering, Zhejiang Wanli University, Ningbo, China
[b] Normandie Univ, UNICAEN, CNRS, LMNO, 14000, Caen, France

ARTICLE INFO

Section Editor: Seth Pettie
Handling Editor: Katie Harris

2020 MSC:
94A60

Keywords:
Cryptographic hard problem
Lattice-based techniques
RSA-polynomial
Semiprime factorization
Unbalanced primes

ABSTRACT

In 2023, Bagherpour proposed a fast variant of the RSA cryptosystem. The security of the new variant is based on a hard problem, called the RSA-polynomial problem, strengthening the traditional security of RSA. This paper investigates the RSA-polynomial problem, and examines its applications. In particular, we further optimize an ambiguous attack boundary initially introduced for RSA-polynomial based semiprime factorization. While this attack boundary for balanced primes has been recently improved by Zheng, our work demonstrates that more powerful and efficient lattice-based attacks exist even if the underlying primes are unbalanced. By conducting a more complete cryptanalysis, we further uncover additional vulnerable cases of the RSA-polynomial problem, thereby broadening the exploitable range by roughly 30.9% compared to previous result. To this end, we propose two exact factoring attacks, i.e., one based on solving bivariate integer polynomial equations, and the other on solving univariate modular polynomial equations using Coppersmith's lattice-based techniques. Theoretical analyses that produce state-of-the-art results, corroborated by experimental findings, validate the correctness and efficacy of our proposed attacks. Moreover, we expose certain shortcomings in the existing RSA-polynomial based commitment and propose a refined alternative.

1. Introduction

The RSA-polynomial problem, introduced by Bagherpour [1] in 2023, represents a new advancement in cryptographic research, particularly within the domain of the RSA (Rivest-Shamir-Adleman) cryptosystem [2]. In his work, Bagherpour claimed that the RSA-polynomial problem is computationally as difficult as semiprime factorization. Furthermore, he proposed a method to factor semiprimes by exploiting the structure of the RSA-polynomial framework in conjunction with lattice-based techniques. This work introduced a new commitment scheme that, by eliminating the need for group exponentiation, achieves superior performance compared to earlier ones. The RSA-polynomial problem is formulated in terms of bivariate integer polynomials as follows.

Definition 1 ([1]). Suppose e_0, e_1, and e_2 are integers, and consider the bivariate polynomial

$$F(x, y) = 8x^2 - e_0x + e_1 + y(8x + e_2).$$

A pair (x_0, y_0) is called its *trivial integer root* if $|8x_0 + e_2| = 1$, and its *non-trivial integer root* if $|8x_0 + e_2| \neq 1$.

Bagherpour further conducted a deeper investigation into the structure of the roots of $F(x, y)$ with a semiprime $N = pq$. We next introduce the integers e_0, e_1, e_2 linked explicitly to the RSA modulus $N = pq$. Lemma 1 captures their key properties and sets the stage for our small–root analysis in Sections 2 and 3.

* Corresponding author.
E-mail addresses: mengce.zheng@gmail.com (M. Zheng), abderrahmane.nitaj@unicaen.fr (A. Nitaj).

https://doi.org/10.1016/j.tcs.2026.115837
Received 28 May 2025; Received in revised form 15 February 2026; Accepted 17 February 2026
Available online 19 February 2026

Lemma 1 ([1]). *Let $N = pq$ be a semiprime with $p, q > 2$. Define $r_1 = p \bmod 8$, $r_2 = q \bmod 8$, $r = N \bmod 8$, $k = (N - r)/8$, and $d = (r - r_1 r_2)/8$. For an integer s such that $1 \le s < (p - r_1)/8$, set*

$$f = \left(k - 8s^2 - (r_1 + r_2)s + d\right) \bmod (8s + r_1), \quad c = \frac{k - 8s^2 - (r_1 + r_2)s + d - f}{8s + r_1}.$$

Then, by defining

$$e_0 = 8c + r_2 - r_1, \quad e_1 = f, \quad e_2 = 8s + r_1,$$

derive the bivariate polynomial

$$F(x, y) = 8x^2 - e_0 x + e_1 + y(8x + e_2).$$

This polynomial $F(x, y)$ admits an integer root (x_0, y_0) satisfying

$$p = 8(s + x_0) + r_1 \quad \text{and} \quad q = 8(s + c - x_0 - y_0) + r_2.$$

Drawing on the above findings, the notion of RSA-polynomials and the associated RSA-polynomial problem is introduced.

Definition 2 ([1]). Let e_0, e_1, e_2 be the integers defined in Lemma 1. Then consider the bivariate polynomial

$$F(x, y) = 8x^2 - e_0 x + e_1 + y(8x + e_2),$$

and define it to be an *RSA-polynomial* if it admits exactly one non-trivial integer root.

Remark 1. Henceforth we fix the integers e_0, e_1, e_2 as in Lemma 1 throughout the paper. Note that while Bagherpour [1] defines an RSA-polynomial specifically based on the uniqueness of this root within a certain bound for y (i.e., $y < \frac{e_0^2 - 32e_1}{16e_0 + 32e_2}$), our cryptanalysis focuses on the general feasibility of recovering (x_0, y_0) from N using the structural properties of the polynomial $F(x, y)$. Considering that the uniqueness bound is a formal requirement for the RSA-polynomial definition, we later observe and confirm that it is consistently satisfied in practical cases, as demonstrated in Section 4.

Algorithm 1 restates the RSA-polynomial generation from [1], which constructs an RSA-polynomial from an RSA modulus $N = pq$.

Algorithm 1 RSA-polynomial generation.

Require: An RSA modulus $N = pq$, residues $r_1 = p \bmod 8$, $r_2 = q \bmod 8$, and an integer $s \in \left[1, (p - r_1)/8 - 1\right]$.
Ensure: An RSA-polynomial $F(x, y)$ as per Definition 2.
1: $r = N \bmod 8$, $k = (N - r)/8$, $d = (r - r_1 r_2)/8$
2: $f = \left(k - 8s^2 - (r_1 + r_2)s + d\right) \bmod (8s + r_1)$
3: $c = (k - 8s^2 - (r_1 + r_2)s + d - f)/(8s + r_1)$
4: $e_0 = 8c + r_2 - r_1$, $e_1 = f$, $e_2 = 8s + r_1$
5: **return** $F(x, y) = 8x^2 - e_0 x + e_1 + y(8x + e_2)$.

RSA-polynomials offer an attractive approach for reducing storage overhead in cryptographic systems. Rather than storing the complete prime factors of a semiprime $N = pq$, one may simply retain the small integer x_0 extracted from the RSA-polynomial $F(x, y)$ and broadcast the polynomial. The prime factors are then recoverable by computing

$$p = 8x_0 + e_2 \quad \text{and} \quad q = e_0 + e_2 - 8(x_0 + y_0),$$

thereby avoiding any computationally intensive procedures.

The intrinsic connection between the difficulty of compromising RSA and the intractability of factoring semiprimes has been thoroughly investigated in previous work like [3–6]. In this context, the RSA-polynomial problem is formally defined as follows. Finding its unique non-trivial integer root for a given RSA-polynomial $F(x, y) = 8x^2 - e_0 x + e_1 + y(8x + e_2)$. Additionally, Bagherpour [1] claimed that resolving the RSA-polynomial problem is at least as difficult as factoring RSA semiprimes.

Moreover, Bagherpour [1, Theorem 5] introduced an effective method for factoring semiprimes, building upon Coppersmith's lattice-based techniques [7] and the RSA-polynomial problem.

Theorem 1 ([1]). *Suppose given an RSA-polynomial*

$$F(x, y) = 8x^2 - e_0 x + e_1 + y(8x + e_2),$$

with its unique non-trivial integer root (x_0, y_0). Assume that $|x_0| \le X$, $|y_0| \le Y$, and define

$$W = \max\{8X^2, |e_0|X, |e_1|, 8XY, |e_2|Y\}.$$

If

$$X^2 Y < \frac{W^{1/2}}{2^{15/2}},$$

then (x_0, y_0) can be recovered in polynomial time.

The above original attack was presented by using a 5-dimensional lattice along with an unclear attack condition. Recently, Zheng [8] revisited such semiprime factorization strategy under the balanced prime assumption (i.e., where prime factors have equal bit-length ℓ). By employing refined lattice-based techniques as introduced in [9,10], Zheng improved the attack by clarifying the condition $X^2Y < W^{1/2}/2^{15/2}$ and demonstrating that a factoring attack is feasible when x_0 is sufficiently small. This is summarized in the following result.

Theorem 2 ([8]). *Suppose given an RSA-polynomial*

$$F(x, y) = 8x^2 - e_0x + e_1 + y(8x + e_2),$$

where $N = pq$ is an RSA modulus and the same bit-length of p, q is ℓ. Suppose $F(x, y)$ has a non-trivial integer root (x_0, y_0) with x_0 of bit-length ξ. If

$$\xi < \frac{(3 - \sqrt{5})\ell}{2},$$

then (x_0, y_0) can be recovered in polynomial time and hence N can be factored.

Zheng's investigation applies higher-dimensional lattice, exposing additional scenarios where Bagherpour's claim no longer holds, thereby uncovering new weaknesses within the RSA-polynomial problem. Furthermore, the RSA-polynomial based commitment introduced in [1] is found to violate the hiding property, rendering it susceptible to attacks that enable the extraction of the committed message and its corresponding opening value.

Remark 2. The original work on the RSA-polynomial problem [1] did not impose any explicit constraints on the bit-lengths of p and q. In contrast, Zheng's analysis [8] considered the generic case where p and q have equal bit-lengths. However, the use of unbalanced primes may impact the computational complexity of solving the RSA-polynomial problem.

1.1. Our contribution

While Zheng [8] focused on RSA-polynomial attacks for moduli with balanced primes, we extend lattice-based attacks to the RSA-polynomial problem involving unbalanced prime factors. In particular, we propose better attacks on the RSA-polynomial problem under the assumption of unbalanced primes. Therefore, a more complete cryptanalysis is conducted in this work.

Several variants of RSA have been proposed where the prime factors of $N = pq$ are unbalanced (see [11,12]). When possible, it is preferable to analyze an attack for both balanced and unbalanced instances. In this work, we initiate a method that works for both instances. Moreover, it has a better result than the former results in [1,8] where only the balanced instance is considered.

Additionally, although many analyses of RSA and its hardness assume roughly balanced primes $p \approx q$, in practice prime generation or implementation constraints may lead to a size–imbalance $q < p$ or $p < q$. This imbalance can reduce the effective security margin and opens the question of how small–root or lattice methods might exploit the imbalance. In this work we extend prior balanced–case analyses to such regime and investigate the parameter region in which a refined small–root attack is effective.

As noted in [8], factoring attacks become viable when x_0 is sufficiently small. Following a similar bound analysis for the parameters X, Y, and W as in the balanced case, we derive new bounds when the RSA modulus $N = pq$ is constructed from primes p and q with differing bit-lengths ℓ_p and ℓ_q. Specifically, we first establish upper bounds for all involved parameters including $|e_0|$, $|e_1|$, and $|e_2|$, and then deduce the corresponding upper bounds on X, Y, and W, which are subsequently employed in our lattice-based attacks.

By applying advanced lattice-based techniques to solve the bivariate polynomial equation $F(x, y)$, we obtain new attack results for the RSA-polynomial problem with unbalanced primes. Moreover, we introduce an enhanced strategy based on solving univariate modular polynomials, which proves to be more efficient than the bivariate approach. A crucial observation is that, since

$$p = 8x_0 + e_2 \quad \text{and} \quad q = e_0 + e_2 - 8(x_0 + y_0),$$

one can extract $z_0 = x_0$ from the congruence $8z + e_2 \equiv 0 \pmod{p}$ or recover $z_0 = -(x_0 + y_0)$ from $8z + e_0 + e_2 \equiv 0 \pmod{q}$. Our improved attack is summarized as follows. Let

$$F(x, y) = 8x^2 - e_0x + e_1 + y(8x + e_2)$$

be an RSA-polynomial associated with an RSA modulus $N = pq$, where p and q have bit-lengths ℓ_p and ℓ_q, respectively. Suppose $F(x, y)$ has a non-trivial integer root (x_0, y_0) with x_0 of bit-length ξ. Then (x_0, y_0) can be recovered in polynomial time and hence N is factored into $N = pq$ if

$$\xi < \frac{\ell_p^2}{\ell_p + \ell_q},$$

which is optimal when l_p is much larger than l_q. Moreover, when p and q are balanced, the former bound becomes $\xi < \frac{1}{2}l_p$. This is better than the bound of Zheng presented in Theorem 2.

1.2. Organization

The rest of this paper is structured as follows. Section 2 establishes fundamental definitions and key lemmas that support our lattice-based solving strategy, while also estimating upper bounds on the unknown parameters relevant to the semiprime factorization. In Section 3, we introduce two new attacks for semiprime factorization with unbalanced primes and propose a refined RSA-polynomial based commitment scheme. Section 4 presents numerical experiments to validate the feasibility and efficiency of our proposed attacks. Section 5 summarizes our main results and concludes the paper.

2. Preliminaries

We provide a concise exposition of the foundational concepts that form the basis of our lattice-based solving strategy. Additionally, we introduce a crucial solvability condition frequently utilized in lattice-based cryptanalysis. For a more comprehensive discussion, we refer the reader to [10,13–16]. Furthermore, we derive upper bounds on relevant parameters when dealing with unbalanced prime factors in the RSA-polynomial framework. For clarity, we assume an RSA modulus $N = pq$, where p and q have respective bit-lengths ℓ_p and ℓ_q, and consider parameters such as e_0, e_1, and e_2 to be distinct powers of 2.

2.1. Lattice-based solving strategy

We first recall the basic concepts used in our cryptanalysis, including the LLL algorithm [17], Coppersmith's lattice-based techniques [7], and Howgrave-Graham's refined criterion [18]. In addition, we introduce a key condition that guarantees a modular polynomial's root is also a root over the integers.

A *lattice* Λ is defined as the set of all integer linear combinations of linearly independent vectors $\vec{b}_1, \vec{b}_2, \dots, \vec{b}_\omega \in \mathbb{R}^n$, i.e.,

$$\Lambda = \left\{ \sum_{i=1}^{\omega} z_i \vec{b}_i \, : \, z_i \in \mathbb{Z} \right\}.$$

The *lattice determinant*, denoted by $\det(\Lambda)$, is given by $\sqrt{\det(BB^{\mathsf{T}})}$ with its transpose B^{T}, where the rows of the basis matrix B are the vectors $\vec{b}_i$. For a full-rank lattice with $\omega = n$, we have $\det(\Lambda) = |\det(B)|$.

The LLL algorithm [17] is a cornerstone for efficiently obtaining a reduced basis with relatively short lattice vectors. In particular, the algorithm produces a reduced basis $(\vec{v}_1, \vec{v}_2, \dots, \vec{v}_\omega)$ satisfying the following bound:

Lemma 2 ([13]). *Let $\Lambda \subset \mathbb{R}^n$ be an ω-dimensional lattice. Then the LLL algorithm returns a reduced basis $(\vec{v}_1, \vec{v}_2, \dots, \vec{v}_\omega)$ such that*

$$\|\vec{v}_i\| \leq 2^{\frac{\omega(\omega-1)}{4(\omega+1-i)}} \det(\Lambda)^{\frac{1}{\omega+1-i}}, \quad \text{for } i = 1, 2, \dots, \omega.$$

The algorithm runs in time polynomial in ω and the logarithm of the maximum norm of the input vector component.

An essential tool for the lattice-based method is the following lemma by Howgrave-Graham, which provides a criterion to lift a modular root to an integer root. Let

$$h(x_1, x_2, \dots, x_n) = \sum c_{i_1, i_2, \dots, i_n} x_1^{i_1} x_2^{i_2} \cdots x_n^{i_n}$$

be an integer polynomial with norm defined by

$$\|h\| := \sqrt{\sum c_{i_1, i_2, \dots, i_n}^2}.$$

Lemma 3 ([18]). *Let $h(x_1, x_2, \dots, x_n) \in \mathbb{Z}[x_1, x_2, \dots, x_n]$ be a polynomial containing at most ω monomials. Suppose $R, X_1, X_2, \dots, X_n$ are certain positive integers. If*

$$h(x_1^\star, x_2^\star, \dots, x_n^\star) \equiv 0 \pmod{R} \quad \text{and} \quad \|h(X_1 x_1, X_2 x_2, \dots, X_n x_n)\| < \frac{R}{\sqrt{\omega}},$$

with $|x_i^\star| \leq X_i$ for $i = 1, \dots, n$, then it follows that

$$h(x_1^\star, x_2^\star, \dots, x_n^\star) = 0$$

over the integers.

By connecting the bounds derived from Lemma 2 with Howgrave-Graham's Lemma 3, the modular (or integer) polynomial equations can be efficiently solved. Suppose reduced vectors $\vec{v}_1, \dots, \vec{v}_k$ have been obtained, satisfying the bound

$$\|\vec{v}_i\| \leq 2^{\frac{\omega(\omega-1)}{4(\omega+1-i)}} \det(\Lambda)^{\frac{1}{\omega+1-i}}, \quad \text{for } i = 1, \dots, k.$$

A solution can be achieved when the following condition holds:

$$2^{\frac{\omega(\omega-1)}{4(\omega+1-k)}} \det(\Lambda)^{\frac{1}{\omega+1-k}} < \frac{R}{\sqrt{\omega}},$$

which is rearranged as

$$\det(\Lambda) < 2^{-\frac{\omega(\omega-1)}{4}} \omega^{-\frac{\omega+1-k}{2}} R^{\omega+1-k}. \tag{1}$$

In practice, since $k < \omega \ll R$, this condition implies $\det(\Lambda) < R^{\omega-\epsilon}$ for some tiny ϵ, and asymptotically one often requires that

$$\det(\Lambda) < R^{\omega}.$$

This condition is central to the effective application of lattice-based techniques in solving polynomial equations.

Coppersmith's lattice-based method has found widespread application in cryptanalysis (e.g., attacks on RSA and its variants [14–16,19–21]) and in analyzing various cryptographic hard problems [8,22–26]. The lattice-based solving strategy typically involves five stages below:

Target Polynomial: Given an irreducible n-variate polynomial $f(x_1, \dots, x_n) \in \mathbb{Z}[x_1, \dots, x_n]$ such that $f(x_1^\star, \dots, x_n^\star) \equiv 0 \pmod{u}$, and given bounds $|x_i^\star| \le X_i$, the target is to efficiently recover the root $(x_1^\star, \dots, x_n^\star)$. Establishing appropriate bounds X_i is crucial for deriving a solvable condition. (Furthermore, the number of given polynomials can be extended to more than one and the given known modulus can be extended to an unknown modulus with its known multiple u.)

Shift Construction: From the target polynomial $f(x_1, \dots, x_n)$, construct *shift polynomials* $g_k(x_1, \dots, x_n)$ for $1 \le k \le \omega$ such that each $g_k(x_1^\star, \dots, x_n^\star) \equiv 0 \pmod{u^m}$ using a well-chosen positive integer m. A standard construction is

$$g_k(x_1, \dots, x_n) := x_1^{i_{k1}} \cdots x_n^{i_{kn}} f^{j_k} u^{m-j_k},$$

where the exponents $(i_{k1}, \dots, i_{kn}, j_k)$ belong to an appropriate index set $\mathcal{I} = \{(i_{k1}, \dots, i_{kn}, j_k) : i_{k1}, \dots, i_{kn}, j_k \in \mathbb{Z},\ i_{k1}, \dots, i_{kn} \ge 0,\ 0 \le j_k \le m,\ 1 \le k \le \omega\}$. It is related to a monomial set $\mathcal{M} = \{x_1^{i_{k1}} \cdots x_n^{i_{kn}} : 1 \le k \le \omega\}$. These shift polynomials shall share a common root $(x_1^\star, \dots, x_n^\star)$ modulo $R = u^m$ by construction. (Furthermore, Jochemsz-May strategy [10] suggests to use extra shifts on some variables x_i's with positive integers t_i's.)

Lattice Generation: Using a proper ordering of monomials and polynomials, form a lattice Λ by representing the coefficient vectors of the scaled shift polynomials $g_k(X_1 x_1, \dots, X_n x_n)$ as the rows $\vec{b}_1, \dots, \vec{b}_\omega$ of a basis matrix B. Moreover, B can be full-rank and triangular under suitable arranging orders.

Basis Reduction: Perform the LLL algorithm to B to obtain several reduced vectors $\vec{v}_1, \dots, \vec{v}_k$ with $k \ge n$. These vectors correspond to integer polynomials $h_i(x_1, \dots, x_n)$, each satisfying $h_i(x_1^\star, \dots, x_n^\star) \equiv 0 \pmod{R}$ by construction. Success is ensured when the parameters satisfy condition (1). (Furthermore, how to make $\det(\Lambda)$ as small as possible compared to R and how to include helpful polynomials as many as possible are the most challenging issues.)

Root Extraction: If $h_i(x_1, \dots, x_n)$ for $1 \le i \le k$ are algebraically independent, these simultaneous integer equations can be solved through Gröbner basis approach [27] or resultant computation. The common root $(x_1^\star, \dots, x_n^\star)$ is finally recovered. While the assumption of algebraic independence for $n \ge 2$ in lattice-based cryptanalysis is heuristic, numerical experiments generally support its validity.

The generation of an efficient lattice and the subsequent extraction of the common root are critical to the overall strategy. Numerous studies (e.g., [9,10,21,28,29]) have focused on optimizing these steps. While the extraction step relies on the assumption of algebraic independence, it is generally accepted in the literature, and our experiments further corroborate its validity.

2.2. Upper bounds on relevant parameters

We derive upper bounds on the relevant parameters involved in the RSA-polynomial problem. To this end, the asymptotic notations are first defined as follows.

Definition 3 ([30]). The asymptotic O, Θ, and o notations with functions $f(n)$ and $g(n)$ are defined as follows:

- The O-notation denotes an asymptotic upper bound: $f(n) = O(g(n))$ indicates $0 \le f(n) \le c_0\, g(n)$ for all $n \ge n_0$ if there exist positive c_0 and n_0.
- The Θ-notation denotes an asymptotic tight bound: $f(n) = \Theta(g(n))$ indicates $0 \le c_1\, g(n) \le f(n) \le c_2\, g(n)$ for all $n \ge n_0$ if there exist positive c_1, c_2, and n_0.
- The o-notation denotes a non-tight asymptotic upper bound: $f(n) = o(g(n))$ indicates $0 \le f(n) < c_0\, g(n)$ for all $n \ge n_0$ if there exists $n_0 > 0$ for every positive c_0.

In our setting, we consider an RSA modulus $N = pq$ with p an ℓ_p-bit prime and q an ℓ_q-bit prime. Other parameters (e.g., e_0, e_1, e_2) are modeled as powers of 2. The Θ-notation is mainly used to establish tight bounds, ensuring that some associated constants remain asymptotically negligible compared to N. This formulation aids in specifying X, Y, and W, where $|x_0| \le X$, $|y_0| \le Y$, and

$$W = \max\{8X^2, |e_0|X, |e_1|, 8XY, |e_2|Y\}.$$

To determine these bounds, we first state the following lemma, which is based on Lemma 1 and the analysis in [8]. In the attack scenarios considered, we assume that $x_0 < s$.

Lemma 4. *Suppose given an RSA-polynomial*

$$F(x,y) = 8x^2 - e_0x + e_1 + y(8x + e_2),$$

where $N = pq$ is an RSA modulus and p, q have bit-lengths ℓ_p, ℓ_q respectively. Suppose that $x_0 < s$ and that x_0 has bit-length ξ, so that $x_0 = \Theta(2^\xi)$ and $s = \Theta(2^{\ell_p - 3})$ with $\xi < \ell_p - 3$. Then, the following bounds hold:

$$|k| = \Theta(2^{\ell_p+\ell_q-3}), \quad |d| = \Theta(1),$$

$$|f| = O(2^{\ell_p}), \quad |c| = \Theta(2^{\max\{\ell_p,\ell_q\}-3}),$$

$$|e_0| = \Theta(2^{\max\{\ell_p,\ell_q\}}), \quad |e_1| = O(2^{\ell_p}), \quad |e_2| = \Theta(2^{\ell_p}),$$

$$|x_0| = \Theta(2^\xi), \quad |y_0| = \Theta(2^{\max\{\ell_q-\ell_p,0\}+\xi}).$$

Proof. From Lemma 1, the parameters r_1, r_2, and r are chosen from the small set $\{1,3,5,7\}$. Therefore,

$$|k| = \frac{N-r}{8} = \Theta\Big(\frac{pq}{8}\Big) = \Theta(2^{\ell_p+\ell_q-3})$$

and

$$|d| = \frac{|r - r_1r_2|}{8} = \Theta(1).$$

Since $p = 8(s + x_0) + r_1$, with $x_0 = \Theta(2^\xi)$ and $s = \Theta(2^{\ell_p-3})$, we deduce that

$$|f| = O(8s + r_1) = O(2^{\ell_p}).$$

Moreover, noting that

$$c = \frac{k - 8s^2 - (r_1 + r_2)s + d - f}{8s + r_1},$$

and given $s = \Theta(2^{\ell_p-3})$, we have

$$|c| = \Theta(\max\{2^{\ell_q-3}, 2^{\ell_p-3}\}) = \Theta(2^{\max\{\ell_p,\ell_q\}-3}).$$

Then, using the definitions $e_0 = 8c + r_2 - r_1$, $e_1 = f$, and $e_2 = 8s + r_1$, it follows that

$$|e_0| = \Theta(8c) = \Theta(2^{\max\{\ell_p,\ell_q\}}), \quad |e_1| = O(2^{\ell_p}), \quad |e_2| = \Theta(8s) = \Theta(2^{\ell_p}).$$

Finally, since $|x_0| = \Theta(2^\xi)$ is given and the relation

$$F(x_0, y_0) = 8x_0^2 - e_0x_0 + e_1 + y_0(8x_0 + e_2) = 0$$

implies

$$|y_0| = \Theta\Big(\frac{\max\{|e_0x_0|,\ |8x_0^2|,\ |e_1|\}}{e_2}\Big).$$

We have

$$|e_0x_0| = \Theta(2^{\max\{\ell_p,\ell_q\}+\xi}), \quad |8x_0^2| = \Theta(2^{2\xi+3}), \quad |e_1| = O(2^{\ell_p}).$$

Since $\xi < \ell_p - 3$, it follows that $2\xi + 3 < \ell_p + \xi \le \max\{\ell_p, \ell_q\} + \xi$, so that

$$|y_0| = \Theta\Big(\frac{|e_0x_0|}{e_2}\Big) = \Theta(2^{\max\{\ell_p,\ell_q\}+\xi-\ell_p}) = \Theta(2^{\max\{\ell_q-\ell_p,0\}+\xi}).$$

This completes the proof. □

We now follow Theorem 1 to define the parameters X, Y, and W using the established bounds on the unknowns. Here, $|x_0| \le X$, $|y_0| \le Y$, and

$$W = \max\{8X^2, |e_0|X, |e_1|, 8XY, |e_2|Y\}.$$

Theorem 3. *Assume that $\xi < \ell_p - 3$, so that $x_0 < s$. Then, the bounds are given by*

$$X = 2^\xi, \quad Y = 2^{\max\{\ell_q-\ell_p,0\}+\xi}, \quad W = 2^{\max\{\ell_p,\ell_q\}+\xi}.$$

Proof. From Lemma 4, we have $X = 2^\xi$ and $Y = 2^{\max\{\ell_q-\ell_p,0\}+\xi}$. Thus, by the definition of W, we obtain

$$W = \max\{8X^2, |e_0|X, |e_1|, 8XY, |e_2|Y\} = \max\{2^{2\xi+3},\ 2^{\max\{\ell_p,\ell_q\}+\xi},\ 2^{\ell_p},\ 2^{\max\{\ell_q-\ell_p,0\}+2\xi+3},\ 2^{\ell_p+\max\{\ell_q-\ell_p,0\}+\xi}\}.$$

Analyzing these terms, we have

$$W = \max\{2^{\max\{\ell_p,\ell_q\}+\xi}, 2^{\max\{\ell_q-\ell_p,0\}+2\xi+3}\}.$$

Hence, $W = 2^{\max\{\ell_p,\ell_q\}+\xi}$ since $\xi - \ell_p + 3 < 0$ and $\max\{\ell_q - \ell_p, 0\} + 2\xi + 3 = \max\{\ell_p, \ell_q\} - \ell_p + 2\xi + 3 < \max\{\ell_p, \ell_q\} + \xi$. This completes the proof. □

3. Better attacks on RSA-polynomial problem with unbalanced primes

We extend Theorem 1 to the case of unbalanced primes, thereby formulating better attack conditions for semiprime factorization. This advancement significantly enhances the effectiveness of lattice-based attacks. To further optimize the factorization process, we propose two distinct strategies: one using bivariate integer polynomials and the other using univariate modular polynomials. Before introducing these lattice constructions, we first establish a basic result for attacking the RSA-polynomial problem with unbalanced primes. We immediately derive a corollary corresponding to the original attack in [1] by combining Theorem 1 with Theorem 3.

Corollary 1. *Suppose given an RSA-polynomial*

$$F(x,y) = 8x^2 - e_0x + e_1 + y(8x + e_2),$$

where $N = pq$ is an RSA modulus and p, q have bit-lengths ℓ_p, ℓ_q respectively. Suppose $F(x,y)$ has a non-trivial integer root (x_0, y_0) and the bit-length of x_0 is ξ with $\xi < \ell_p - 3$. Then (x_0, y_0) can be recovered in polynomial time and hence N is factored into $N = pq$ if

$$\xi < \begin{cases} \ell_p/5 - 3, & \text{if } \ell_p \geq \ell_q, \\ (2\ell_p - \ell_q)/5 - 3, & \text{if } \ell_q/2 + 15/2 < \ell_p < \ell_q. \end{cases}$$

Proof. Substitute $X = 2^{\xi}$, $Y = 2^{\max\{\ell_q - \ell_p, 0\} + \xi}$, and $W = 2^{\max\{\ell_p, \ell_q\} + \xi}$ from Theorem 3 into the condition

$$X^2Y < \frac{W^{1/2}}{2^{15/2}}$$

in Theorem 1. Expressing the inequality in terms of exponents yields

$$2\xi + \max\{\ell_q - \ell_p, 0\} + \xi < \frac{\max\{\ell_p, \ell_q\} + \xi}{2} - \frac{15}{2}.$$

Rearranging this inequality leads to

$$\xi < \frac{\max\{\ell_p, \ell_q\} - 2\max\{\ell_q - \ell_p, 0\}}{5} - 3. \tag{2}$$

We consider the following two cases:

- When $\ell_p \geq \ell_q$, we have $\max\{\ell_p, \ell_q\} = \ell_p$ and $\max\{\ell_q - \ell_p, 0\} = 0$. Thus, (2) simplifies to
$$\xi < \frac{\ell_p}{5} - 3.$$
- When $\ell_p < \ell_q$, we obtain $\max\{\ell_p, \ell_q\} = \ell_q$ and $\max\{\ell_q - \ell_p, 0\} = \ell_q - \ell_p$. Then, (2) implies
$$\xi < \frac{2\ell_p - \ell_q}{5} - 3.$$
Additionally, since $\xi > 0$, it is necessary that $0 < 2\ell_p - \ell_q - 15$, or equivalently, $\ell_q/2 + 15/2 < \ell_p$.

This completes the proof. □

3.1. Lattice-based attacks using bivariate integer polynomials

We begin by presenting an attack strategy that relies on solving bivariate integer polynomials. In this approach, we directly apply the RSA-polynomial whose small root relates to the factorization of the RSA modulus N. The main result is stated below.

Theorem 4. *Suppose given an RSA-polynomial*

$$F(x,y) = 8x^2 - e_0x + e_1 + y(8x + e_2),$$

where $N = pq$ is an RSA modulus and p, q have bit-lengths ℓ_p, ℓ_q respectively. Suppose $F(x,y)$ has a non-trivial integer root (x_0, y_0) and the bit-length of x_0 is ξ with $\xi < \ell_p - 3$. Then (x_0, y_0) can be recovered in polynomial time and hence N is factored into $N = pq$ if

$$\xi < \begin{cases} \dfrac{(3-\sqrt{5})\ell_p}{2}, & \text{if } \ell_p \geq \ell_q, \\ \dfrac{2\ell_p + \ell_q - \sqrt{4\ell_p\ell_q + \ell_q^2}}{2}, & \text{if } \ell_p < \ell_q. \end{cases} \tag{3}$$

Proof. We start from the given bivariate polynomial $F(x,y) = 8x^2 - e_0x + e_1 + y(8x + e_2)$ with small root (x_0, y_0). We follow a strategy similar to that in [8] and use two positive integers m and t (to be optimized later). The specific monomial sets are

$$\mathcal{M}_F = \{x^i y^j : 0 \leq i \leq m + t - j,\ 0 \leq j \leq m\},$$

Table 1
A toy example for $m = 1$ and $t = 1$.

$g_{[i,j]}$ \ $x^i y^j$	1	x	y	x^2	xy	y^2	x^3	x^2y	xy^2	x^4	x^3y	x^2y^2
$g_{[0,0]}$	X^2Y	$*$	$*$	$*$	$*$	0	0	0	0	0	0	0
$g_{[1,0]}$	0	X^2Y	0	$*$	$*$	0	$*$	$*$	0	0	0	0
$g_{[0,1]}$	0	0	X^2Y	0	$*$	$*$	0	$*$	$*$	0	0	0
$g_{[2,0]}$	0	0	0	X^2Y	0	0	$*$	$*$	0	$*$	$*$	0
$g_{[1,1]}$	0	0	0	0	X^2Y	0	0	$*$	$*$	0	$*$	$*$
$g_{[0,2]}$	0	0	0	0	0	Y^2R	0	0	0	0	0	0
$g_{[3,0]}$	0	0	0	0	0	0	X^3R	0	0	0	0	0
$g_{[2,1]}$	0	0	0	0	0	0	0	X^2YR	0	0	0	0
$g_{[1,2]}$	0	0	0	0	0	0	0	0	XY^2R	0	0	0
$g_{[4,0]}$	0	0	0	0	0	0	0	0	0	X^4R	0	0
$g_{[3,1]}$	0	0	0	0	0	0	0	0	0	0	X^3YR	0
$g_{[2,2]}$	0	0	0	0	0	0	0	0	0	0	0	X^2Y^2R

$$\mathcal{M} = \{x^i y^j : 0 \le i \le m + t + 2 - j,\ 0 \le j \le m + 1\}.$$

X and Y are respective upper bounds on $|x_0|$ and $|y_0|$. We have

$$W = \max\{8X^2, |e_0|X, |e_1|, 8XY, |e_2|Y\}.$$

We assume that $\gcd(e_1, WXY) = 1$. Since the maximum degrees of x and y in $\mathcal{M}_F$ are $m + t$ and m, respectively, we define

$$R = WX^{m+t}Y^m.$$

For convenience in the lattice-based strategy, the constant term of $F(x, y)$ (i.e., $c_{00} = e_1$) should be normalized to 1. Thus, we consider the modular polynomial

$$f(x, y) \equiv c_{00}^{-1} F(x, y) \equiv c_{20}x^2 + c_{11}xy + c_{10}x + c_{01}y + 1 \pmod{R},$$

where

$$c_{20} \equiv 8e_1^{-1} \pmod{R},$$

$$c_{11} \equiv 8e_1^{-1} \pmod{R},$$

$$c_{10} \equiv -e_0e_1^{-1} \pmod{R},$$

$$c_{01} \equiv e_2e_1^{-1} \pmod{R}.$$

Using the sets $\mathcal{M}_F$ and $\mathcal{M} \setminus \mathcal{M}_F$, the shift polynomials are

$$g_{[i,j]}(x, y) := \begin{cases} x^i y^j f(x, y) X^{m+t-i} Y^{m-j}, & \text{if } x^i y^j \in \mathcal{M}_F, \\ x^i y^j R, & \text{if } x^i y^j \in \mathcal{M} \setminus \mathcal{M}_F. \end{cases}$$

Therefore, for all (i, j) it follows that

$$g_{[i,j]}(x_0, y_0) \equiv 0 \pmod{R}.$$

Next, the coefficient vectors of the scaled polynomials $g_{[i,j]}(Xx, Yy)$ are used to construct a lattice Λ with basis matrix B. We arrange the rows (polynomials) according to a polynomial order $\prec_{\mathsf{p}}$ and the columns (monomials) according to a monomial order $\prec_{\mathsf{m}}$, where the ordering prioritizes the terms in $\mathcal{M}_F$ over those in $\mathcal{M} \setminus \mathcal{M}_F$. For instance, one may define $\prec_{\mathsf{p}}$ using the ascending graded lexicographic order. To be specific, $g_{[i,j]}$ comes ahead for smaller grade $i + j$, or for greater i with the same grade, and $\prec_{\mathsf{m}}$ is defined similarly.

For illustration, we set $m = 1$ and $t = 1$. Table 1 provides a toy example of the resulting lattice basis matrix B, where "$*$" denotes nonzero off-diagonal entries.

The lattice determinant is expressed as

$$\det(\Lambda) = R^{s_R} X^{s_X} Y^{s_Y},$$

where the exponents s_R, s_X, and s_Y are positive integers that can be computed using $s(\sigma_v)$ for $v = X$, Y, or R. Here σ_v represent the respective degrees of X, Y, or R appeared in the leading terms $X^{\sigma_X} Y^{\sigma_Y} R^{\sigma_R}$ of each scaled shift polynomials.

$$\begin{aligned} s(\sigma_v) &= \sum_{x^i y^j \in \mathcal{M}_F} \sigma_v + \sum_{x^i y^j \in \mathcal{M} \setminus \mathcal{M}_F} \sigma_v \\ &= \sum_{j=0}^{m} \sum_{i=0}^{m+t-j} \sigma_v^{\mathcal{M}_F} + \sum_{j=0}^{m+1} \sum_{i=0}^{m+t+2-j} \sigma_v^{\mathcal{M} \setminus \mathcal{M}_F} - \sum_{j=0}^{m} \sum_{i=0}^{m+t-j} \sigma_v^{\mathcal{M} \setminus \mathcal{M}_F}. \end{aligned}$$

More precisely, since the respective leading terms are $X^{m+t}Y^m$ for $x^iy^j \in \mathcal{M}_F$ and X^iY^jR for $x^iy^j \in \mathcal{M} \setminus \mathcal{M}_F$, we obtain

$$s_R = s(\sigma_R) = \sum_{j=0}^{m}\sum_{i=0}^{m+t-j} 0 + \sum_{j=0}^{m+1}\sum_{i=0}^{m+t+2-j} 1 - \sum_{j=0}^{m}\sum_{i=0}^{m+t-j} 1 = 2m+t+4,$$

$$s_X = s(\sigma_X) = \sum_{j=0}^{m}\sum_{i=0}^{m+t-j} (m+t) + \sum_{j=0}^{m+1}\sum_{i=0}^{m+t+2-j} i - \sum_{j=0}^{m}\sum_{i=0}^{m+t-j} i$$

$$= \frac{m^3}{2} + \frac{3m^2t}{2} + \frac{5m^2}{2} + mt^2 + \frac{9mt}{2} + 5m + \frac{3t^2}{2} + \frac{9t}{2} + 4,$$

$$s_Y = s(\sigma_Y) = \sum_{j=0}^{m}\sum_{i=0}^{m+t-j} m + \sum_{j=0}^{m+1}\sum_{i=0}^{m+t+2-j} j - \sum_{j=0}^{m}\sum_{i=0}^{m+t-j} j$$

$$= \frac{m^3}{2} + m^2t + \frac{5m^2}{2} + 2mt + 4m + t + 2.$$

The lattice dimension is given by

$$\omega = s(1) = \sum_{j=0}^{m}\sum_{i=0}^{m+t-j} 1 + \sum_{j=0}^{m+1}\sum_{i=0}^{m+t+2-j} 1 - \sum_{j=0}^{m}\sum_{i=0}^{m+t-j} 1 = \frac{m^2}{2} + mt + \frac{7m}{2} + 2t + 5.$$

We obtain several reduced integer polynomials $h_i(x,y)$, $1 \le i \le \omega$ after performing the LLL algorithm. If the two shortest vectors $h_1(x,y)$ and $h_2(x,y)$ satisfy those bounds from Lemmas 2 and 3, then we must have

$$\det(\Lambda) < R^{\omega-1}2^{-\frac{\omega(\omega-1)}{4}}\omega^{-\frac{\omega-1}{2}}.$$

Since the factor $2^{-\frac{\omega(\omega-1)}{4}}\omega^{-\frac{\omega-1}{2}}$ is negligible for large ω, it follows that $\det(\Lambda) < R^{\omega-1}$. Substituting the expression for $\det(\Lambda)$ and for R, we obtain

$$R^{s_R}X^{s_X}Y^{s_Y} < R^{\omega-1},$$

which implies

$$X^{s_X}Y^{s_Y} < (WX^{m+t}Y^m)^{\omega-s_R-1}.$$

Equivalently,

$$W^{s_R-\omega+1}X^{s_X-(m+t)(\omega-s_R-1)}Y^{s_Y-m(\omega-s_R-1)} < 1.$$

For simplicity, set $t = \tau m$ with $\tau > 0$ and ignore lower-order terms. Then, the dominant parts of the exponents become

$$s_R - \omega + 1 = -\frac{1+2\tau}{2}m^2 + o(m^2),$$

$$s_X - (m+t)(\omega - s_R - 1) = \frac{2+4\tau+\tau^2}{2}m^2 + o(m^2),$$

$$s_Y - m(\omega - s_R - 1) = (1+\tau)m^2 + o(m^2),$$

for W, X, and Y respectively. Substituting $X = 2^{\xi}$, $Y = 2^{\max\{\ell_q-\ell_p,0\}+\xi}$, and $W = 2^{\max\{\ell_p,\ell_q\}+\xi}$ from Theorem 3 and converting to exponents yields

$$-\frac{1+2\tau}{2}(\max\{\ell_p,\ell_q\}+\xi) + \frac{2+4\tau+\tau^2}{2}\xi + (1+\tau)(\max\{\ell_q-\ell_p,0\}+\xi) < 0.$$

Simplifying, we obtain

$$\frac{3+4\tau+\tau^2}{2}\xi < \frac{1+2\tau}{2}\max\{\ell_p,\ell_q\} - (1+\tau)\max\{\ell_q-\ell_p,0\},$$

which can be rearranged as

$$\xi < \frac{-\max\{\ell_q-\ell_p,0\} + (1+2\tau)\ell_p}{3+4\tau+\tau^2}. \tag{4}$$

We consider the following two cases:

- When $\ell_p \ge \ell_q$, we have $\max\{\ell_q-\ell_p,0\} = 0$, so that

$$\xi < \frac{(1+2\tau)\ell_p}{3+4\tau+\tau^2}.$$

Choosing $\tau = (\sqrt{5}-1)/2$ maximizes the right-hand side, yielding

$$\xi < \frac{(3-\sqrt{5})\ell_p}{2}. \tag{5}$$

- When $\ell_p < \ell_q$, we have $\max\{\ell_q - \ell_p, 0\} = \ell_q - \ell_p$, which leads to

$$\xi < \frac{(2+2\tau)\ell_p - \ell_q}{3+4\tau+\tau^2}.$$

After optimizing by setting

$$\tau = \frac{\ell_q - 2\ell_p + \sqrt{4\ell_p\ell_q + \ell_q^2}}{2\ell_p},$$

we obtain

$$\xi < \frac{2\ell_p + \ell_q - \sqrt{4\ell_p\ell_q + \ell_q^2}}{2}. \tag{6}$$

Note that we ensure that $\tau > \frac{\ell_q - 2\ell_p + \ell_q}{2\ell_p} > 0$. Moreover, we require $\xi > 0$ and $2\ell_p + \ell_q > \sqrt{4\ell_p\ell_q + \ell_q^2}$ is always true.

Thus, the overall attack boundary on ξ is given by

$$\xi < \begin{cases} \dfrac{(3-\sqrt{5})\ell_p}{2}, & \text{if } \ell_p \geq \ell_q, \\ \dfrac{2\ell_p + \ell_q - \sqrt{4\ell_p\ell_q + \ell_q^2}}{2}, & \text{if } \ell_p < \ell_q. \end{cases}$$

The overall running time is polynomial in $\log W$ and hence polynomial in ℓ_p and ℓ_q. Once (x_0, y_0) is recovered, the prime factors are obtained via

$$p = 8x_0 + e_2 \quad \text{and} \quad q = e_0 + e_2 - 8(x_0 + y_0).$$

This completes the proof. □

Remark 3. Our lattice-based attack on the RSA-polynomial problem with unbalanced primes covers the previous result of [8] as stated in Theorem 2 and further improves upon it, thus providing a more complete cryptanalysis of the RSA-polynomial problem.

3.2. More efficient attacks using univariate modular polynomials

Motivated by the ideas in [1,8], one may ask whether a more efficient attack can be achieved by using univariate modular polynomials rather than bivariate integer polynomials. We answer this question affirmatively with the following result.

Theorem 5. *Suppose given an RSA-polynomial*

$$F(x, y) = 8x^2 - e_0 x + e_1 + y(8x + e_2),$$

where $N = pq$ is an RSA modulus and p, q have bit-lengths ℓ_p, ℓ_q respectively. Suppose $F(x, y)$ has a non-trivial integer root (x_0, y_0) and the bit-length of x_0 is ξ with $\xi < \ell_p - 3$. Then (x_0, y_0) can be recovered in polynomial time and hence N is factored into $N = pq$ if

$$\xi < \frac{\ell_p^2}{\ell_p + \ell_q}. \tag{7}$$

Proof. Our key observation is that we have

$$p = 8x_0 + e_2 \quad \text{and} \quad q = e_0 + e_2 - 8(x_0 + y_0)$$

in the RSA-polynomial problem. Thus, one may extract x_0 by solving the univariate modular equation

$$8z + e_2 \equiv 0 \pmod{p},$$

or equivalently, extract $-(x_0 + y_0)$ by solving

$$8z + e_0 + e_2 \equiv 0 \pmod{q}.$$

This requires one to solve a univariate modular polynomial in the form of $G(z) := c_1 z + c_0 \pmod{u}$ with known c_0 and c_1. We are given N, exactly a multiple of u for unknown $u = p$ or $u = q$. Moreover, we shall ensure that the univariate polynomial to be solved is a monic polynomial, so we multiply $G(z)$ by $c_1^{-1} \bmod N$, i.e., $8^{-1} \bmod N$. Thus, we derive the following modular polynomial

$$f(z) \equiv (8^{-1} \bmod N) \cdot G(z) \equiv z + (8^{-1} c_0 \bmod N) \pmod{u}.$$

To find the small root z_0, we employ a strategy similar to that in [14,29]. We define the family of shift polynomials

$$g_k(z) := f^k(z) N^{\max\{t-k,0\}}, \quad k = 0, \ldots, m$$

Table 2
A toy example for $m = 9$ and $t = 6$.

g_k \ z^k	1	z	z^2	z^3	z^4	z^5	z^6	z^7	z^8	z^9
g_0	N^6	0	0	0	0	0	0	0	0	0
g_1	*	N^5Z	0	0	0	0	0	0	0	0
g_2	*	*	N^4Z^2	0	0	0	0	0	0	0
g_3	*	*	*	N^3Z^3	0	0	0	0	0	0
g_4	*	*	*	*	N^2Z^4	0	0	0	0	0
g_5	*	*	*	*	*	NZ^5	0	0	0	0
g_6	*	*	*	*	*	*	Z^6	0	0	0
g_7	*	*	*	*	*	*	*	Z^7	0	0
g_8	*	*	*	*	*	*	*	*	Z^8	0
g_9	*	*	*	*	*	*	*	*	*	Z^9

for a positive integer m and an integer t with $t = \tau m$ $(0 \le \tau < 1)$. Let $\mathcal{M} = \{z^k : 0 \le k \le m\}$ be the corresponding monomial set. By construction, for all k we have

$$g_k(z_0) \equiv 0 \pmod{R},$$

where $R = u^t$.

Let Z be an upper bound for $|z_0|$, so that $|z_0| \le Z$. The coefficient vectors of the scaled polynomials $g_k(Zz)$ are used to generate a basis matrix B and then a lattice Λ. We order the polynomials and monomials in the ascending order (i.e., $g_k \prec_{\mathsf{p}} g_{k'}$ and $z^k \prec_{\mathsf{m}} z^{k'}$ if $k < k'$). For illustration, assume $m = 9$ and $t = 6$. Table 2 provides a toy example of the corresponding lattice basis matrix.

The lattice determinant is given by

$$\det(\Lambda) = N^{s_N} Z^{s_Z},$$

where the exponents s_N, s_Z are positive integers that can be computed using $s(\sigma_v)$ for $v = N$ or Z. Here σ_v represent the respective degrees on N or Z appeared in the leading terms $N^{\sigma_N} Z^{\sigma_Z}$ of each scaled shift polynomials.

$$s(\sigma_v) = \sum_{z^k \in \mathcal{M}} \sigma_v = \sum_{k=0}^{m} \sigma_v^{\mathcal{M}}.$$

More precisely, since the leading terms are $N^{\max\{t-k,0\}} Z^k$ for $z^k \in \mathcal{M}$ and $t < m$, we obtain

$$s_N = s(\sigma_N) = \sum_{k=0}^{m} \max\{t-k, 0\} = \sum_{k=0}^{t} (t-k) = \frac{t^2}{2} + \frac{t}{2},$$

$$s_Z = s(\sigma_Z) = \sum_{k=0}^{m} k = \frac{m^2}{2} + \frac{m}{2}.$$

The lattice dimension is

$$\omega = s(1) = \sum_{k=0}^{m} 1 = m + 1.$$

We obtain a reduced basis yielding a polynomial $h_1(z)$ with shortest norm after performing the LLL algorithm. According to (1), we require

$$\det(\Lambda) < R^{\omega} 2^{-\frac{\omega(\omega-1)}{4}} \omega^{-\frac{\omega}{2}},$$

so that (ignoring negligible factors)

$$\det(\Lambda) < R^{\omega}.$$

Substituting $\det(\Lambda) = N^{s_N} Z^{s_Z}$ and $R = u^t$, we have

$$N^{s_N} Z^{s_Z} < u^{t\omega}.$$

Recalling $t = \tau m$ for simplicity and omitting the lower-order terms, the respective dominant parts in s_N, s_Z, and $t\omega$ are

$$s_N = \frac{\tau^2}{2} m^2 + o(m^2),$$

$$s_Z = \frac{1}{2} m^2 + o(m^2),$$

$$t\omega = \tau m^2 + o(m^2).$$

Thus, we obtain

$$N^{\frac{\tau^2}{2}m^2} Z^{\frac{1}{2}m^2} u^{-\tau m^2} < 1. \tag{8}$$

Depending on whether $u = p$ or $u = q$ is considered in solving $f(z) \equiv 0 \pmod{u}$, we can divide the further analysis and hence the attack boundary of ξ into two cases:

- When we choose $u = p$ and try to solve $z_0 = x_0$ from $8z + e_2 \equiv 0 \pmod{p}$. We have

  $$N = 2^{\ell_p + \ell_q}, \quad Z = X = 2^{\xi}, \quad u = p = 2^{\ell_p}.$$

  Simplifying the above inequality (8), we obtain

  $$\frac{\tau^2}{2}(\ell_p + \ell_q) + \frac{1}{2}\xi - \tau\ell_p < 0,$$

  which results in

  $$\xi < 2\tau\ell_p - \tau^2(\ell_p + \ell_q).$$

  Setting $\tau = \frac{\ell_p}{\ell_p + \ell_q}$ to maximize the right-hand side, it follows that

  $$\xi < \frac{\ell_p^2}{\ell_p + \ell_q}. \tag{9}$$

- When we choose $u = q$ and try to solve $z_0 = -(x_0 + y_0)$ from $8z + e_0 + e_2 \equiv 0 \pmod{q}$. We have

  $$N = 2^{\ell_p + \ell_q}, \quad Z = \max\{X, Y\} = 2^{\max\{\ell_q - \ell_p, 0\} + \xi}, \quad u = q = 2^{\ell_q}.$$

  Simplifying the above inequality (8), we obtain

  $$\frac{\tau^2}{2}(\ell_p + \ell_q) + \frac{1}{2}(\max\{\ell_q - \ell_p, 0\} + \xi) - \tau\ell_q < 0,$$

  which results in

  $$\xi < 2\tau\ell_q - \tau^2(\ell_p + \ell_q) - \max\{\ell_q - \ell_p, 0\}.$$

  Setting $\tau = \frac{\ell_q}{\ell_p + \ell_q}$ to maximize the right-hand side, it follows that

  $$\xi < \frac{\ell_q^2}{\ell_p + \ell_q} - \max\{\ell_q - \ell_p, 0\}. \tag{10}$$

  More concretely, this leads to

  $$\xi < \begin{cases} \dfrac{\ell_q^2}{\ell_p + \ell_q}, & \text{if } \ell_p \geq \ell_q, \\ \dfrac{\ell_q^2}{\ell_p + \ell_q} - (\ell_q - \ell_p) = \dfrac{\ell_p^2}{\ell_p + \ell_q}, & \text{if } \ell_p < \ell_q. \end{cases}$$

  We finally derive the attack boundary

  $$\xi < \frac{\min\{\ell_p, \ell_q\}^2}{\ell_p + \ell_q}. \tag{11}$$

  Combining both cases, we conclude that solving $8z + e_2 \equiv 0 \pmod{p}$ is always superior and the attack boundary is

  $$\xi < \frac{\ell_p^2}{\ell_p + \ell_q}.$$

The overall running time is polynomial in $\log N$ and hence polynomial in ℓ_p and ℓ_q. Once the small root z_0 is recovered, the primes can be computed as

$$p = 8z_0 + e_2 \quad \text{and} \quad q = \frac{N}{p}.$$

This completes the proof. □

Remark 4. For a more precise attack boundary, one may write

$$\xi \leq \frac{\ell_p^2}{\ell_p + \ell_q} - \epsilon,$$

for a negligible error term ϵ. The corresponding time complexity is estimated as $O(\epsilon^{-7} \log^2 N)$ (see [14,29] for further details).

Remark 5. The proposed lattice-based attacks are non-heuristic. In the univariate modular case (Theorem 5), the root recovery is guaranteed by Coppersmith's method [7]. For the bivariate integer case (Theorem 4), it is still non-heuristic as Coppersmith's method is guaranteed to yield an algebraically independent integer polynomial (see Section 10 in [7] for further explanations). Note that the root-extraction step for an n-variate modular equation with $n \geq 2$ is considered heuristic in the literature.

3.3. Comparison and discussion

We now compare the two proposed attacks and discuss the resulting semiprime factoring algorithm. It should be clarified that for a given $N = pq$, Algorithm 2 generates a candidate RSA-polynomial using the construction in Algorithm 1. For the purpose of factorization, we do not strictly require $F(x, y)$ to meet the formal uniqueness bound of an RSA-polynomial; rather, we only require that the root (x_0, y_0) (or z_0) remains within the recoverable boundary ξ. Based on Theorems 4 and 5, the attack boundaries are given respectively by

$$\xi < \begin{cases} \dfrac{(3-\sqrt{5})\ell_p}{2}, & \ell_p \geq \ell_q, \\ \dfrac{2\ell_p + \ell_q - \sqrt{4\ell_p\ell_q + \ell_q^2}}{2}, & \ell_p < \ell_q, \end{cases}$$

and

$$\xi < \frac{\ell_p^2}{\ell_p + \ell_q}.$$

We show that the univariate approach is always superior by comparing the two cases.

- When $\ell_p \geq \ell_q$, define

$$\Delta_{\ell_p \geq \ell_q} := \frac{\ell_p^2}{\ell_p + \ell_q} - \frac{(3-\sqrt{5})\ell_p}{2}.$$

  Rewriting, we have

$$\Delta_{\ell_p \geq \ell_q} = \frac{((\sqrt{5}-1)\ell_p - (3-\sqrt{5})\ell_q)\ell_p}{2(\ell_p + \ell_q)}.$$

  Since $\sqrt{5} - 1 > 3 - \sqrt{5} > 0$ and $\ell_p \geq \ell_q$, it follows that $(\sqrt{5}-1)\ell_p - (3-\sqrt{5})\ell_q > 0$ and hence $\Delta_{\ell_p \geq \ell_q} > 0$.
- When $\ell_p < \ell_q$, define

$$\Delta_{\ell_p < \ell_q} := \frac{\ell_p^2}{\ell_p + \ell_q} - \frac{2\ell_p + \ell_q - \sqrt{4\ell_p\ell_q + \ell_q^2}}{2}.$$

  One can rewrite this as

$$\Delta_{\ell_p < \ell_q} = \frac{(\ell_p + \ell_q)\sqrt{(4\ell_p + \ell_q)\ell_q} - (3\ell_p + \ell_q)\ell_q}{2(\ell_p + \ell_q)}.$$

  Observing that

$$(\ell_p + \ell_q)\sqrt{(4\ell_p + \ell_q)\ell_q} - (3\ell_p + \ell_q)\ell_q = \frac{(\ell_p + \ell_q)^2(4\ell_p + \ell_q)\ell_q - (3\ell_p + \ell_q)^2\ell_q^2}{(\ell_p + \ell_q)\sqrt{(4\ell_p + \ell_q)\ell_q} + (3\ell_p + \ell_q)\ell_q},$$

  and noting that the numerator simplifies to $4\ell_p^3\ell_q > 0$, we conclude that $\Delta_{\ell_p < \ell_q} > 0$.

Thus, the attack boundary derived from Theorem 5

$$\xi < \frac{\ell_p^2}{\ell_p + \ell_q}$$

is strictly better than that of the bivariate approach. Moreover, comparing the underlying lattice constructions reveals that the lattice dimension in Theorem 4 is $\Theta(m^2)$, whereas in Theorem 5 it is $\Theta(m)$. In addition, the lattice determinants satisfy

$$\det(\Lambda) = \begin{cases} R^{s_R} X^{s_X} Y^{s_Y}, & \text{with } s_R = \Theta(m),\ s_X, s_Y = \Theta(m^3), \\ N^{s_N} Z^{s_Z}, & \text{with } s_N, s_Z = \Theta(m^2), \end{cases}$$

which again favors the univariate approach.

The attack boundary improves from $\xi < \frac{(3-\sqrt{5})\ell_p}{2}$ to $\xi < \frac{\ell_p^2}{\ell_p+\ell_q}$. In the balanced case $\ell_p = \ell_q$, our new attack boundary becomes $\xi < \frac{\ell_p}{2}$ with an improvement of approximately 30.9%. This finding suggests that the RSA-polynomial problem admits considerably more

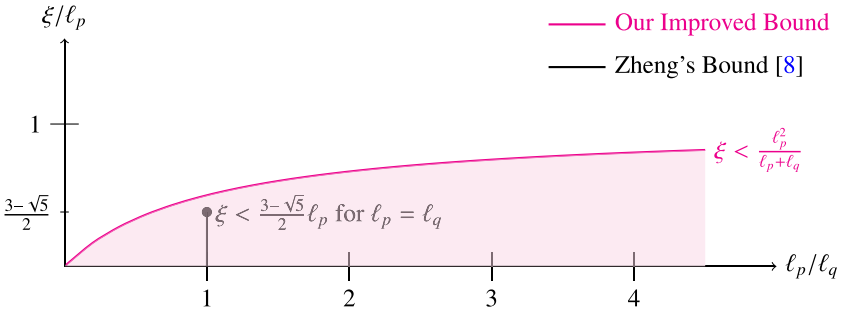

Fig. 1. Attack region for the RSA-polynomial problem: horizontal axis shows ratio ℓ_p/ℓ_q, vertical axis shows the small-root bound ξ over ℓ_p.

weak instances where N can be efficiently factored. In particular, the success probability of semiprime factorization is approximately estimated as

$$\Pr = 2^{\frac{\ell_p^2}{\ell_p+\ell_q}} \Big/ 2^{\ell_p-3} \approx 2^{-\frac{\ell_p\ell_q}{\ell_p+\ell_q}}.$$

Fig. 1 illustrates the vulnerable parameter space in which our attack succeeds, compared to the balanced–primes case. The prior method of Zheng [8] applies only when $\ell_p = \ell_q$, corresponding to a single line in the plot. In contrast, our method covers a substantially larger region in the plane of ℓ_p/ℓ_q versus ξ/ℓ_p, reflecting unbalanced configurations. The shaded area in the figure highlights this expanded attack surface when p and q differ in magnitude.

We present a new RSA-polynomial based semiprime factorization below in Algorithm 2. Before that, we require the definition of Rem_r, the set of possible couples $(r_1, r_2) \equiv (p \bmod 8, q \bmod 8)$ in Algorithm 1, corresponding to $r \equiv r_1 r_2 \equiv N \bmod 8$ for $r = 1, 3, 5, 7$.

Definition 4. Define Rem_r as follows: $\mathsf{Rem}_1 = \{(1,1),(3,3),(5,5),(7,7)\}$, $\mathsf{Rem}_3 = \{(1,3),(3,1),(7,5),(5,7)\}$, $\mathsf{Rem}_5 = \{(1,5),(5,1),(7,3),(3,7)\}$, and $\mathsf{Rem}_7 = \{(1,7),(7,1),(3,5),(5,3)\}$.

Algorithm 2 shall output the prime factors p, q of a semiprime $N = pq$ by generating an RSA-polynomial satisfying Theorem 5. For a given semiprime N, one randomly selects an appropriate parameter s, Algorithm 2 is capable of recovering the prime factors of N efficiently. Two major enhancements have been incorporated: first, the range for the random choice of s has been broadened to strengthen the overall attack; and second, Theorem 5 is applied to significantly improve both the success probability and attack efficiency. Consequently, our new RSA-polynomial based semiprime factorization presents a promising approach to lattice-based factorization. It complements existing approaches such as those in [8,22,31–35].

Algorithm 2 New RSA-polynomial based semiprime factorization.

Require: A semiprime $N = pq$ with primes of bit-lengths ℓ_p and ℓ_q.
Ensure: The factorization $N = pq$.
1: $r = N \bmod 8$
2: $s \leftarrow$ a random $(\ell_p - 3)$-bit integer
3: **for all** $(r_1, r_2) \in \mathsf{Rem}_r$ **do** ▷ See Definition 4
4: Construct $F(x, y) = 8x^2 - e_0 x + e_1 + y(8x + e_2)$ via Algorithm 1 with inputs (N, r_1, r_2, s).
5: Extract z_0 by solving $8z + e_2 \equiv 0 \pmod{p}$ using Theorem 5.
6: **if** z_0 is found as an integer **then**
7: $p = 8z_0 + e_2$
8: $q = \lfloor N/p \rceil$ ▷ $\lfloor \bullet \rceil$ denotes rounding
9: **if** $p \times q = N$ **then**
10: **return** p and q
11: **else**
12: **continue** with a new random s (return to Step 2)
13: **end if**
14: **else**
15: **continue** with a new random s (return to Step 2)
16: **end if**
17: **end for**

It should be noted that even though the enhanced RSA-polynomial based semiprime factorization significantly optimizes the success probability, a successful factorization is not guaranteed in every instance. Its running time may vary with the specific parameters of N and the random choices of s. Nonetheless, Algorithm 2 outperforms previous algorithms in [1] and [8] in terms of both success rate and efficiency.

In recent decades, many methods have been proposed that try to factor an RSA modulus $N = pq$. Some of such attacks apply when the decryption exponent d is small, as in [13,14,36–39], and others assume that a fraction of the decryption key is known, as in [40–43], or that a fraction of one of the private factors of N is known, as in [44–46]. We notice that in the Bagherpour variant of RSA, the private exponent d is not small, and there is no known approximation of it, nor of one of the prime factors. As a consequence, none of the former attacks can be applied to the Bagherpour variant of RSA.

There are several general factoring methods that can be adapted to factor an RSA modulus such as the number field sieve (NFS) [47] (also the general number field sieve (GNFS) [48]), and the elliptic curve method of factorization (ECM) [49]. While ECM works for small integers, NFS can factor RSA moduli of moderate size such as RSA-250, an RSA modulus of 831 bit-size [50]. Since the size of the RSA moduli used for concrete applications is at least 2048 bits, the classical factoring methods are ineffective to solve the factorization problem.

3.4. Refined RSA-polynomial based commitment scheme

The commitment scheme based on RSA-polynomials was initially proposed in [1] and later improved in [8]. In the original scheme, a message $\mathsf{m} \in [1, 2^{\ell-1} - 1]$ is committed via the following procedure:

1. Two random primes p and q of equal bit-length ℓ are generated, yielding a semiprime $N = pq$.
2. The values $r_1 = p \bmod 8$ and $r_2 = q \bmod 8$ are computed alongside the modulus N.
3. A parameter s is determined by the formula
$$s = \frac{p - r_1}{8} - \mathsf{m}.$$
4. By running Algorithm 1 with (N, r_1, r_2, s), an RSA-polynomial is produced
$$F(x, y) = 8x^2 - e_0 x + e_1 + y(8x + e_2).$$
5. The opening value o is then computed as
$$o = \frac{8\mathsf{m}^2 - e_0\mathsf{m} + e_1}{8\mathsf{m} + e_2}.$$
6. The pair $(N, F(x, y))$ is transmitted to the receiver as the commitment witness.
7. The value o is securely stored as the decommitment information.

As shown in [8], a polynomial-time adversary can break this scheme when m is short. Hence, the message range is refined to $\mathsf{m} \in [2^{(3-\sqrt{5})\ell/2-1}, 2^{\ell-4} - 1]$. However, our analysis shows that even some messages outside this range (along with their corresponding opening values) can be recovered via Theorem 5, which is clearly stated below.

Proposition 1. *For an RSA-polynomial based commitment scheme with two ℓ_p, ℓ_q-bit primes and a ξ-bit message m, then both m and o can be efficiently extracted, provided that*
$$\xi < \frac{\ell_p^2}{\ell_p + \ell_q}.$$

Since Proposition 1 directly follows from Theorem 5, we omit the detailed proof. Consequently, we further refine the message space to
$$\mathsf{m} \in \left[2^{\frac{\ell_p^2}{\ell_p+\ell_q}-1}, 2^{\ell_p-4} - 1\right].$$

For completeness, our refined commitment scheme operates as follows:

Initialization: The sender first broadcasts the system parameters and detailed instructions for both the commitment and verification phases.

Commitment: The sender proceeds commitment on a message $\mathsf{m} \in [2^{\ell/2-1}, 2^{\ell-4} - 1]$ as follows:

1. Two random primes p and q of equal bit-length ℓ are generated, yielding a semiprime $N = pq$.
2. The values $r_1 = p \bmod 8$ and $r_2 = q \bmod 8$ are computed alongside the modulus N.
3. A parameter s is determined by the formula
$$s = \frac{p - r_1}{8} - \mathsf{m}.$$

4. By running Algorithm 1 with (N, r_1, r_2, s), an RSA-polynomial is produced
$$F(x, y) = 8x^2 - e_0x + e_1 + y(8x + e_2).$$
5. The opening value o is then computed as
$$o = \frac{8\mathsf{m}^2 - e_0\mathsf{m} + e_1}{8\mathsf{m} + e_2}.$$
6. The pair $(N, F(x, y))$ is transmitted to the receiver as the commitment witness.
7. The value o is securely stored as the decommitment information.
8. The sender forwards the pair (m, o) to the receiver for subsequent verification.

Verification: The receiver performs the following checks upon receiving (m, o):

1. Calculate $p = 8\mathsf{m} + e_2$.
2. Determine $q = e_0 + e_2 - 8(\mathsf{m} + o)$.
3. Accept the commitment if all of the following conditions are met:
   - The pair (m, o) is verified to be the unique non-trivial integer root of $F(x, y)$.
   - The computed p and q each has a bit-length greater than the security parameter.
   - The computed p and q satisfy $N = pq$ and N is a non-trivial semiprime.
4. Otherwise, the commitment is rejected.

The correctness, hiding, and binding properties can be verified in a manner similar to [1]. We provide the proofs for security properties of the refined commitment scheme as follows.

Theorem 6. *The refined commitment scheme satisfies the correctness property of commitment schemes.*

Proof. Assume that the sender S and receiver R follow the above refined commitment scheme honestly. Let $(N, F(x, y), o)$ be a commitment to message m, where
$$F(x, y) = 8x^2 - e_0\,x + e_1 + y\,(8x + e_2).$$
By construction the scheme sets
$$s = \frac{p - r_1}{8} - \mathsf{m}, \quad o = \frac{8\mathsf{m}^2 - e_0\mathsf{m} + e_1}{8\mathsf{m} + e_2},$$
so that the pair (m, o) is a non-trivial integer root of $F(x, y)$. From Lemma 1 we have
$$p = 8(s + \mathsf{m}) + r_1, \quad q = 8\big(s + c - \mathsf{m} - o\big) + r_2,$$
where c, r_1, r_2 come from the scheme parameter setup. Since
$$e_2 = 8s + r_1, \quad e_0 = 8c + r_2 - r_1,$$
we deduce
$$p = 8\mathsf{m} + e_2, \quad q = e_2 + e_0 - 8(\mathsf{m} + o).$$
Hence $p \cdot q = N$, (m, o) is the designated integer root of $F(x, y)$, and m satisfies the verification–phase requirements. It follows that the receiver R will accept m as the committed message. □

Theorem 7. *The refined commitment scheme satisfies the hiding property of commitment schemes.*

Proof. Consider a commitment $(N, F(x, y), o)$ produced by the sender for message m. Suppose a polynomial-time adversarial receiver R' attempts to compute either m or o given only the witness $(N, F(x, y))$. Since the commitment is valid, the pair (m, o) is the non-trivial integer root of the polynomial $F(x, y)$. By the above analysis, computing (m, o) from $(N, F(x, y))$ is at least as hard as factoring N. Therefore, with overwhelming probability, R' cannot compute (m, o) given only $(N, F(x, y))$. This establishes that the message m remains hidden from R'. □

Theorem 8. *The refined commitment scheme satisfies the binding property of commitment schemes.*

Proof. Suppose there are two openings for the same commitment instance: $(N, F(x, y), o)$ corresponding to message m, and $(N, F(x, y), o')$ corresponding to message m'. In each case, the pair (m, o) and (m', o') is a non-trivial integer root of the same polynomial $F(x, y)$. By Lemma 1 and injectivity, it follows that $\mathsf{m} = \mathsf{m}'$ and $o = o'$. Hence a sender cannot open a single commitment to two distinct messages. Therefore the scheme satisfies the binding property. □

Table 3
Experimental results on RSA-polynomial based semiprime factorization.

Parameters		Theorem 4						Theorem 5					
ℓ_p	ℓ_q	ξ_t (Theor.)	ξ_e (Exper.)	m	t	ω	Time (Sec.)	ξ_t (Theor.)	ξ_e (Exper.)	m	t	ω	Time (Sec.)
128	896	14	9	2	13	66	57.6	16	**14**	65	8	66	72.3
192	832	31	23	2	8	46	12.4	36	**32**	45	8	46	13.5
256	768	53	44	2	5	34	3.7	64	**58**	33	8	34	3.4
320	704	81	79	3	5	45	21.9	100	**94**	44	13	45	17.5
384	640	113	118	3	4	40	11.7	144	**137**	39	14	40	11.3
448	576	152	163	3	2	30	1.7	196	**188**	29	12	30	3.0
512	512	195	226	4	2	39	4.0	256	**249**	38	19	39	4.7
576	448	220	282	4	2	39	4.8	324	**317**	38	21	39	4.8
640	384	244	356	4	2	39	5.7	400	**393**	38	23	39	5.3
704	320	268	432	4	2	39	7.0	484	**478**	38	26	39	5.8
768	256	293	510	4	2	39	7.7	576	**571**	38	28	39	6.5
832	192	317	634	5	3	56	91.5	676	**673**	55	44	56	39.6
896	128	342	751	5	3	56	96.8	784	**781**	55	48	56	42.8

4. Validating experiments

To evaluate the feasibility and effectiveness of our proposed attacks developed based on Theorem 4 and Theorem 5, extensive numerical experiments were carried out. The computational tests were conducted on a system running a 64-bit Windows 10 operating system with Ubuntu 22.04 via WSL 2. The hardware specifications included a 2.80 GHz CPU and 16 GB of RAM. Lattice reduction, which plays a central role in our attack, was performed using the LLL algorithm [17] implemented in the `SageMath` [51] software platform.

To assess the performance of our proposed RSA-polynomial based semiprime factorization, we first generated two random prime numbers p and q with bit-lengths ℓ_p and ℓ_q, respectively. A random integer s was then selected, where the parameter ξ represented the bit-length of the unknown variable x_0. With inputs $(N = pq, s, r_1, r_2)$, Algorithm 1 was applied to compute the values of e_0, e_1, and e_2, thereby constructing an RSA-polynomial $F(x, y)$ that meets the conditions defined in Definition 2.

The parameters m and t in Theorems 4 and 5 were carefully selected to control the lattice dimensions and enhance the efficiency of the attacks. We executed each attack instance multiple times to ensure successful extraction of (x_0, y_0) (or z_0). A set of integer polynomial equations satisfying the necessary solvability condition was collected. The resultant computation was employed to extract the target integer root. The experimental results are summarized in Table 3.

For each attack instance, a collection of integer polynomials sharing a common root was identified. Then, an optimal subset was chosen to recover the root. Once x_0 and y_0 (or z_0 in the univariate case) were successfully obtained, the factorization of $N = pq$ was completed using the relations: $p = 8x_0 + e_2$, $q = e_0 + e_2 - 8(x_0 + y_0)$ for the bivariate case, and $p = 8z_0 + e_2$, $q = N/p$ for the univariate case.

To address the uniqueness bound condition for y mentioned in Remark 1, we performed post-facto verification on all experimental trials. After recovering p and q (and thus x_0 and y_0), we checked the condition $y_0 < \frac{e_0^2 - 32e_1}{16e_0 + 32e_2}$. In every instance, this inequality was satisfied, validating that the generated instances were consistent with the formal definition of RSA-polynomials.

When comparing the experimental bit-length ξ_e with the theoretical bound ξ_t in Table 3, we observe two distinct trends. In most cases, ξ_e falls slightly below ξ_t, likely due to the limited lattice dimension used in our experiments. However, ξ_e surpasses ξ_t for certain parameter settings with $\ell_p \geq 384$ in Theorem 4, suggesting potential room for further optimization, either by refining the attack strategy or by fully exploiting hidden structural properties of the constructed lattice.

Furthermore, additional observations can be drawn from the experimental data. Despite the above phenomenon, a comparison between Theorem 5 and Theorem 4 confirms that the experimental bound of Theorem 5 consistently surpasses that of Theorem 4, aligning with and validating the theoretical analysis. For Theorem 5, the experimental bound falls short of the theoretical bound by less than 9 bits while only utilizing lattices of the dimensions less than 70. This indicates the high efficiency of our proposed attack. Naturally, employing lattices of higher dimensions is expected to further enhance the practical attack effectiveness.

These observations implies that the attack based on Theorem 5 outperforms that of Theorem 4 in terms of efficiency. In computer experiments, when lattices of identical dimensions are employed for predetermined respective attack settings, the univariate approach yields better results within approximate running time. Consequently, we endorse the usage of the univariate-based factoring attack for practical efficiency.

5. Conclusions

We conducted a comprehensive study of the RSA-polynomial problem with the aim of providing a more complete analysis in this work. In particular, we optimized the ambiguous attack boundary presented in [1] and also improved that given in [8]. To do this, we considered an extended attack condition by taking into account unbalanced primes. Furthermore, we proposed two exact lattice-based attacks: one using bivariate integer polynomials and the other using univariate modular polynomials.

Our analysis significantly expands the vulnerable range for small roots of the RSA-polynomial, achieving an improvement of approximately 30.9% over the previous results in [8] concerning balanced primes. Both theoretical analyses and experimental results validate the correctness and efficacy of our proposed semiprime factorization. Moreover, our findings also expose certain limitations in existing applications, particularly in the RSA-polynomial based commitment proposed in Bagherpour [1] and further improved in Zheng [8]. By establishing a sharper attack boundary, our work contributes to the improvement of the associated commitment scheme, enhancing its security analysis.

In summary, our contributions not only extend the scope of known vulnerabilities in the RSA-polynomial problem but also provide significant improvements for both cryptanalysis and cryptographic design.

CRediT authorship contribution statement

Mengce Zheng: Writing – review & editing, Writing – original draft, Validation, Software, Methodology, Investigation, Funding acquisition, Formal analysis, Data curation, Conceptualization; **Abderrahmane Nitaj:** Writing – review & editing, Writing – original draft, Methodology, Investigation, Conceptualization.

Data availability

Data will be made available on request.

Declaration of competing interest

Mengce Zheng reports travel was provided by China Scholarship Council. If there are other authors, they declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

Mengce Zheng gratefully acknowledges the support for academic visiting provided by the China Scholarship Council (CSC No. 202308330390). This work was supported by the National Natural Science Foundation of China, grant number 62002335, and Ningbo Young Science and Technology Talent Cultivation Program, grant number 2023QL007.

References

[1] B. Bagherpour, A bivariate polynomial-based cryptographic hard problem and its applications, Des. Codes Cryptogr. (2023) 1–13. https://doi.org/10.1007/s10623-023-01229-1
[2] R. Rivest, A. Shamir, L. Adleman, A method for obtaining digital signatures and public-key cryptosystems, Commun. ACM 21 (2) (1978) 120–126. https://doi.org/10.1145/359340.359342
[3] D. Boneh, R. Venkatesan, Breaking RSA may not be equivalent to factoring, in: K. Nyberg (Ed.), Advances in Cryptology - EUROCRYPT '98, International Conference on the Theory and Application of Cryptographic Techniques, Espoo, Finland, May 31–June 4, 1998, Proceeding, 1403 of *Lecture Notes in Computer Science*, Springer, 1998, pp. 59–71. https://doi.org/10.1007/BFb0054117
[4] D. Aggarwal, U. Maurer, Breaking RSA generically is equivalent to factoring, in: A. Joux (Ed.), Advances in Cryptology - EUROCRYPT 2009, 28th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Cologne, Germany, April 26–30, 2009. Proceedings, 5479 of *Lecture Notes in Computer Science*, Springer, 2009, pp. 36–53. https://doi.org/10.1007/978-3-642-01001-9_2
[5] D. Aggarwal, U. Maurer, Breaking RSA generically is equivalent to factoring, IEEE Trans. Inf. Theory 62 (11) (2016) 6251–6259. https://doi.org/10.1109/TIT.2016.2594197
[6] D. Brown, Breaking RSA may be as difficult as factoring, J. Cryptol. 29 (1) (2016) 220–241. https://doi.org/10.1007/s00145-014-9192-y
[7] D. Coppersmith, Small solutions to polynomial equations, and low exponent RSA vulnerabilities, J. Cryptol. 10 (4) (1997) 233–260. https://doi.org/10.1007/s001459900030
[8] M. Zheng, Revisiting RSA-polynomial problem and semiprime factorization, Theor. Comput. Sci. 1004 (2024) 114634. https://doi.org/10.1016/J.TCS.2024.114634
[9] J. Blömer, A. May, A tool kit for finding small roots of bivariate polynomials over the integers, in: R. Cramer (Ed.), Advances in Cryptology - EUROCRYPT 2005, 24Th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Aarhus, Denmark, May 22–26, 2005, Proceedings, 3494 of *Lecture Notes in Computer Science*, Springer, 2005, pp. 251–267. https://doi.org/10.1007/11426639_15
[10] E. Jochemsz, A. May, A strategy for finding roots of multivariate polynomials with new applications in attacking RSA variants, in: X. Lai, K. Chen (Eds.), Advances in Cryptology - ASIACRYPT 2006, 12th International Conference on the Theory and Application of Cryptology and Information Security, Shanghai, China, December 3–7, 2006, Proceedings, 4284 of *Lecture Notes in Computer Science*, Springer, 2006, pp. 267–282. https://doi.org/10.1007/11935230_18
[11] A. Shamir, RSA for paranoids, CryptoBytes 1 (1995) 1–4.
[12] D. Boneh, H. Shacham, Fast variants of RSA, CryptoBytes 5 (1) (2002) 1–9.
[13] A. May, New RSA vulnerabilities using lattice reduction methods, Ph.D. thesis, University of Paderborn, 2003. http://ubdata.uni-paderborn.de/ediss/17/2003/may/disserta.pdf.
[14] A. May, Using LLL-reduction for solving RSA and factorization problems, in: P.Q. Nguyen, B. Vallée (Eds.), The LLL Algorithm - Survey and Applications, Information Security and Cryptography, Springer, 2010, pp. 315–348. https://doi.org/10.1007/978-3-642-02295-1_10
[15] A. May, Lattice-based integer factorization - An introduction to coppersmith's method, in: J.W. Bos, M. Stam (Eds.), Computational Cryptography: Algorithmic Aspects of Cryptology, London Mathematical Society Lecture Note Series, Cambridge University Press, 2021, pp. 78–105. https://www.cits.ruhr-uni-bochum.de/imperia/md/content/may/paper/intro_to_coppersmiths_method.pdf.
[16] M. Zheng, H. Kang, Lattice-based cryptanalysis of RSA-type cryptosystems: a bibliometric analysis, Cybersecur. 7 (1) (2024) 74. https://doi.org/10.1186/S42400-024-00289-7
[17] A. Lenstra, H. Lenstra, L. Lovász, Factoring polynomials with rational coefficients, Mathematische Annalen 261 (4) (1982) 515–534.

[18] N. Howgrave-Graham, Finding small roots of univariate modular equations revisited, in: M. Darnell (Ed.), Cryptography and Coding, 6th IMA International Conference, Cirencester, UK, December 17–19, 1997, Proceedings, 1355 of *Lecture Notes in Computer Science*, Springer, 1997, pp. 131–142. https://doi.org/10.1007/BFb0024458. https://doi.org/10.1007/BFb0024458

[19] A. May, J. Nowakowski, S. Sarkar, Approximate divisor multiples - factoring with only a third of the secret CRT-exponents, in: O. Dunkelman, S. Dziembowski (Eds.), Advances in Cryptology - EUROCRYPT 2022 - 41st Annual International Conference on the Theory and Applications of Cryptographic Techniques, Trondheim, Norway, May 30–June 3, 2022, Proceedings, Part III, 13277 of *Lecture Notes in Computer Science*, Springer, 2022, pp. 147–167. https://doi.org/10.1007/978-3-031-07082-2_6

[20] Y. Zhou, J. Pol, Y. Yu, F. Standaert, A third is all you need: extended partial key exposure attack on CRT-RSA with additive exponent blinding, in: S. Agrawal, D. Lin (Eds.), Advances in Cryptology - ASIACRYPT 2022 - 28th International Conference on the Theory and Application of Cryptology and Information Security, Taipei, Taiwan, December 5–9, 2022, Proceedings, Part IV, 13794 of *Lecture Notes in Computer Science*, Springer, 2022, pp. 508–536. https://doi.org/10.1007/978-3-031-22972-5_18

[21] K. Ryan, Solving multivariate coppersmith problems with known moduli, IACR Cryptol. ePrint Arch. (2024) 1577. https://eprint.iacr.org/2024/1577.

[22] M. Herrmann, A. May, Solving linear equations modulo divisors: on factoring given any bits, in: J. Pieprzyk (Ed.), Advances in Cryptology - ASIACRYPT 2008, 14th International Conference on the Theory and Application of Cryptology and Information Security, Melbourne, Australia, December 7–11, 2008. Proceedings, 5350 of *Lecture Notes in Computer Science*, Springer, 2008, pp. 406–424. https://doi.org/10.1007/978-3-540-89255-7_25

[23] S. Sarkar, S. Maitra, Approximate integer common divisor problem relates to implicit factorization, IEEE Trans. Inf. Theory 57 (6) (2011) 4002–4013. https://doi.org/10.1109/TIT.2011.2137270

[24] K. Tosu, N. Kunihiro, Optimal bounds for multi-prime Φ-Hiding assumption, in: W. Susilo, Y. Mu, J. Seberry (Eds.), Information Security and Privacy - 17th Australasian Conference, ACISP 2012, Wollongong, NSW, Australia, July 9–11, 2012. Proceedings, 7372 of *Lecture Notes in Computer Science*, Springer, 2012, pp. 1–14. https://doi.org/10.1007/978-3-642-31448-3_1

[25] J. Meers, J. Nowakowski, Solving the hidden number problem for CSIDH and CSURF via automated coppersmith, in: J. Guo, R. Steinfeld (Eds.), Advances in Cryptology - ASIACRYPT 2023 - 29th International Conference on the Theory and Application of Cryptology and Information Security, Guangzhou, China, December 4–8, 2023, Proceedings, Part IV, 14441 of *Lecture Notes in Computer Science*, Springer, 2023, pp. 39–71. https://doi.org/10.1007/978-981-99-8730-6_2

[26] J. Xu, S. Sarkar, L. Hu, H. Wang, Y. Pan, Revisiting modular inversion hidden number problem and its applications, IEEE Trans. Inf. Theory 69 (8) (2023) 5337–5356. https://doi.org/10.1109/TIT.2023.3263485

[27] T. Becker, V. Weispfenning, H. Kredel, Gröbner bases - a computational approach to commutative algebra, 141 of *Graduate texts in mathematics*, Springer, 1993.

[28] A. Takayasu, N. Kunihiro, Better lattice constructions for solving multivariate linear equations modulo unknown divisors, in: C. Boyd, L. Simpson (Eds.), Information Security and Privacy - 18th Australasian Conference, ACISP 2013, Brisbane, Australia, July 1–3, 2013. Proceedings, 7959 of *Lecture Notes in Computer Science*, Springer, 2013, pp. 118–135. https://doi.org/10.1007/978-3-642-39059-3_9

[29] Y. Lu, R. Zhang, L. Peng, D. Lin, Solving linear equations modulo unknown divisors: revisited, in: T. Iwata, J.H. Cheon (Eds.), Advances in Cryptology - ASIACRYPT 2015 - 21st International Conference on the Theory and Application of Cryptology and Information Security, Auckland, New Zealand, November 29–December 3, 2015, Proceedings, Part I, 9452 of *Lecture Notes in Computer Science*, Springer, 2015, pp. 189–213. https://doi.org/10.1007/978-3-662-48797-6_9

[30] T. Cormen, C. Leiserson, R. Rivest, C. Stein, Introduction to Algorithms, Fourth Edition, The MIT Press, 2022. https://mitpress.mit.edu/9780262046305/introduction-to-algorithms/.

[31] D. Boneh, G. Durfee, N. Howgrave-Graham, Factoring $N = p^r q$ for large r, in: M.J. Wiener (Ed.), Advances in Cryptology - CRYPTO '99, 19th Annual International Cryptology Conference, Santa Barbara, California, USA, August 15-19, 1999, Proceedings, 1666 of *Lecture Notes in Computer Science*, Springer, 1999, pp. 326–337. https://doi.org/10.1007/3-540-48405-1_21

[32] J. Coron, R. Zeitoun, Improved factorization of $N = p^r q^s$, in: N.P. Smart (Ed.), Topics in Cryptology - CT-RSA 2018 - the Cryptographers' Track at the RSA Conference 2018, San Francisco, CA, USA, April 16–20, 2018, Proceedings, 10808 of *Lecture Notes in Computer Science*, Springer, 2018, pp. 65–79. https://doi.org/10.1007/978-3-319-76953-0_4

[33] S. Wang, L. Qu, C. Li, H. Wang, Further improvement of factoring $N = p^r q^s$ with partial known bits, Adv. Math. Commun. 13 (1) (2019) 121–135. https://doi.org/10.3934/amc.2019007

[34] M. Zheng, H. Hu, Implicit related-key factorization problem on the RSA cryptosystem, in: Y. Mu, R.H. Deng, X. Huang (Eds.), Cryptology and Network Security - 18th International Conference, CANS 2019, Fuzhou, China, October 25–27, 2019, Proceedings, 11829 of *Lecture Notes in Computer Science*, Springer, 2019, pp. 525–537. https://doi.org/10.1007/978-3-030-31578-8_29. https://doi.org/10.1007/978-3-030-31578-8_29

[35] M. Zheng, Z. Chen, Y. Wu, Solving generalized bivariate integer equations and its application to factoring with known bits, IEEE Access 11 (2023) 34674–34684. https://doi.org/10.1109/ACCESS.2023.3264590

[36] M. Wiener, Cryptanalysis of short RSA secret exponents, IEEE Trans. Inf. Theory 36 (3) (1990) 553–558. https://doi.org/10.1109/18.54902

[37] D. Boneh, G. Durfee, Cryptanalysis of RSA with private key d less than $N^{0.292}$, IEEE Trans. Inf. Theory 46 (4) (2000) 1339–1349. https://doi.org/10.1109/18.850673

[38] M. Zheng, H. Hu, Z. Wang, Generalized cryptanalysis of RSA with small public exponent, Sci. China Inf. Sci. 59 (3) (2016) 32108:1–32108:10. https://doi.org/10.1007/s11432-015-5325-7

[39] J. Blömer, A. May, Low secret exponent RSA revisited, in: J.H. Silverman (Ed.), Cryptography and Lattices, International Conference, CaLC 2001, Providence, RI, USA, March 29–30, 2001, Revised Papers, 2146 of *Lecture Notes in Computer Science*, Springer, 2001, pp. 4–19. https://doi.org/10.1007/3-540-44670-2_2

[40] K. Suzuki, A. Takayasu, N. Kunihiro, Extended partial key exposure attacks on RSA: improvement up to full size decryption exponents, Theor. Comput. Sci. 841 (2020) 62–83. https://doi.org/10.1016/j.tcs.2020.07.004

[41] A. Takayasu, N. Kunihiro, Partial key exposure attacks on RSA: achieving the Boneh-Durfee bound, Theor. Comput. Sci. 761 (2019) 51–77. https://doi.org/10.1016/j.tcs.2018.08.021

[42] M. Ernst, E. Jochemsz, A. May, B. Weger, Partial key exposure attacks on RSA up to full size exponents, in: R. Cramer (Ed.), Advances in Cryptology - EUROCRYPT 2005, 24th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Aarhus, Denmark, May 22–26, 2005, Proceedings, 3494 of *Lecture Notes in Computer Science*, Springer, 2005, pp. 371–386. https://doi.org/10.1007/11426639_22

[43] J. Blömer, A. May, New partial key exposure attacks on RSA, in: D. Boneh (Ed.), Advances in Cryptology - CRYPTO 2003, 23rd Annual International Cryptology Conference, Santa Barbara, California, USA, August 17–21, 2003, Proceedings, 2729 of *Lecture Notes in Computer Science*, Springer, 2003, pp. 27–43. https://doi.org/10.1007/978-3-540-45146-4_2

[44] B. Weger, Cryptanalysis of RSA with small prime difference, Appl. Algebra Eng. Commun. Comput. 13 (1) (2002) 17–28. https://doi.org/10.1007/s002000100088

[45] Y. Feng, A. Nitaj, Y. Pan, Partial prime factor exposure attacks on some RSA variants, Theor. Comput. Sci. 999 (2024) 114549. https://doi.org/10.1016/J.TCS.2024.114549

[46] M. Zheng, N. Kunihiro, H. Hu, Improved factoring attacks on multi-prime RSA with small prime difference, in: J. Pieprzyk, S. Suriadi (Eds.), Information Security and Privacy - 22nd Australasian Conference, ACISP 2017, Auckland, New Zealand, July 3–5, 2017, Proceedings, Part I, 10342 of *Lecture Notes in Computer Science*, Springer, 2017, pp. 324–342. https://doi.org/10.1007/978-3-319-60055-0_17

[47] A.K. Lenstra, H.W. Lenstra, Jr., M.S. Manasse, J.M. Pollard, The number field sieve, in: H. Ortiz (Ed.), Proceedings of the 22nd Annual ACM Symposium on Theory of Computing, May 13–17, 1990, Baltimore, Maryland, USA, ACM, 1990, pp. 564–572. https://doi.org/10.1145/100216.100295

[48] J.A. Buchmann, J. Loho, J. Zayer, An implementation of the general number field sieve, in: D.R. Stinson (Ed.), Advances in Cryptology - CRYPTO '93, 13th Annual International Cryptology Conference, Santa Barbara, California, USA, August 22–26, 1993, Proceedings, 773 of *Lecture Notes in Computer Science*, Springer, 1993, pp. 159–165. https://doi.org/10.1007/3-540-48329-2_14

[49] H.W. Lenstra, Jr, Factoring integers with elliptic curves, Ann. Math. (1987) 126 (3) 649–673 https://doi.org/10.2307/1971363
[50] F. Boudot, P. Gaudry, A. Guillevic, N. Heninger, E. Thomé, P. Zimmermann, Comparing the difficulty of factorization and discrete logarithm: a 240-digit experiment, in: D. Micciancio, T. Ristenpart (Eds.), Advances in Cryptology - CRYPTO 2020 - 40th Annual International Cryptology Conference, CRYPTO 2020, Santa Barbara, CA, USA, August 17–21, 2020, Proceedings, Part II, 12171 of *Lecture Notes in Computer Science*, Springer, 2020, pp. 62–91. https://doi.org/10.1007/978-3-030-56880-1_3
[51] The Sage Developers, SageMath, the Sage Mathematics Software System (Version 10.3), 2025. https://www.sagemath.org.