Theoretical Computer Science 1004 (2024) 114634

Contents lists available at ScienceDirect

Theoretical Computer Science

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

# Revisiting RSA-polynomial problem and semiprime factorization ☆

Mengce Zheng

*College of Information and Intelligence Engineering, Zhejiang Wanli University, Ningbo, China*

ARTICLE INFO

Communicated by G. Yang

*Keywords:*
Bivariate integer polynomial
Cryptographic hard problem
Factorization
Lattice
RSA

ABSTRACT

This paper focuses on the RSA-polynomial problem, a cryptographic hard problem that has been recently proposed and studied in, along with its various applications. We revisit this problem and conduct a refined analysis to address an ambiguous condition that was previously introduced in the context of RSA-polynomial based semiprime factorization. By deriving an accurate attack condition, we are able to identify weak cases of the RSA-polynomial problem and expand the vulnerable bound. To facilitate this, we propose two optimized factoring attacks that leverage improved lattice-based theorems for solving bivariate integer polynomials of a specific form. The validity and effectiveness of our proposed factoring attacks are verified through both theoretical analysis and experimental results. Additionally, we examine the RSA-polynomial based commitment scheme and identify deficiencies that compromise its reliability. To address the limitations, we propose enhancements to the commitment phase of the scheme.

## 1. Introduction

Recently, Bagherpour [1] introduced the RSA-polynomial problem, a new cryptographic challenge related to RSA (Rivest-Shamir-Adleman) [2]. The author's demonstration established the equivalence in difficulty between solving the RSA-polynomial problem and solving the factoring problem. Moreover, the author presented a methodology for factoring semiprimes by leveraging the RSA-polynomial problem alongside lattice basis reduction techniques. Additionally, a novel commitment scheme based on the RSA-polynomial problem was introduced. The proposed commitment scheme surpasses previous schemes in terms of performance and eliminates the need for group exponentiation. Specifically, the computational cost of the RSA-polynomial based commitment scheme is lower compared to well-known commitment schemes such as [3–5].

The RSA-polynomial problem, as introduced in [1], is based on bivariate integer polynomials of a specific form, defined as follows.

**Definition 1** *([1])*. Suppose $e_0$, $e_1$ and $e_2$ are arbitrary integers and $F(x,y) = 8x^2 - e_0x + e_1 + y(8x + e_2)$ is a bivariate polynomial. A pair of integers $(x_0, y_0)$ satisfying $F(x_0, y_0) = 0$ is called a trivial integer root (or non-trivial integer root) of $F(x,y)$ if $|8x_0 + e_2| = 1$ (or $|8x_0 + e_2| \neq 1$), respectively.

The author analyzed the solutions of $F(x,y)$ by embedding a semiprime $n$ with two non-trivial factors $p$ and $q$ of the same bit-length. The following lemma presents this embedding.

☆ This article belongs to Section A: Algorithms, automata, complexity and games, Edited by Paul Spirakis.

*E-mail address:* mengce.zheng@gmail.com.

https://doi.org/10.1016/j.tcs.2024.114634
Received 14 November 2023; Received in revised form 11 April 2024; Accepted 13 May 2024

Available online 16 May 2024

**Lemma 1** *([1]). Let $n$ be a semiprime with non-trivial factors $p$, $q$ of the same bit-length. Suppose $r_1 = p \pmod 8$, $r_2 = q \pmod 8$, $r = n \pmod 8$, $k = (n-r)/8$ and $d = (r - r_1 r_2)/8$. Let $s$ be an arbitrary integer satisfying $1 \le s < (p - r_1)/8$. Then, consider the following equations:*

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

*Consider the bivariate polynomial $F(x, y) = 8x^2 - e_0 x + e_1 + y(8x + e_2)$, where $e_0 = 8c + r_2 - r_1$, $e_1 = f$ and $e_2 = 8s + r_1$. For any $y < (e_0^2 - 32e_1)/(16e_0 + 32e_2)$, $F(x, y)$ possesses two roots. Moreover, it possesses only one non-trivial integer root $(x_0, y_0)$ that satisfies $p = 8(s + x_0) + r_1$ and $q = 8(s + c - x_0 - y_0) + r_2$.*

Note that throughout our paper, we explicitly mention and utilize the existence of two prime factors of the same bit-length, although it is implicitly mentioned in [1], to ensure clear and precise descriptions.

Using the aforementioned findings, the author introduced the concept of RSA-polynomials and the corresponding RSA-polynomial problem.

**Definition 2** *([1]).* Let $e_0$, $e_1$ and $e_2$ be arbitrary integers. A bivariate polynomial $F(x, y)$ is referred to as an RSA-polynomial if $F(x, y) = 8x^2 - e_0 x + e_1 + y(8x + e_2)$ and possesses only one non-trivial integer root.

The RSA-polynomial algorithm [1, Algorithm 1] is restated in Algorithm 1 and illustrates a method to generate RSA-polynomials using an RSA modulus with two distinct primes of the same bit-length.

**Algorithm 1** RSA-polynomial algorithm.

**Require:** An RSA modulus $n = pq$, $r_1 = p \pmod 8$, $r_2 = q \pmod 8$, a random integer $s \in [1, (p - r_1)/8 - 1]$.
**Ensure:** An RSA-polynomial $F(x, y)$ satisfying Definition 2.
1: $r \leftarrow n \pmod 8$
2: $k \leftarrow (n - r)/8$
3: $d \leftarrow (r - r_1 r_2)/8$
4: $f \leftarrow (k - 8s^2 - (r_1 + r_2)s + d) \pmod{(8s + r_1)}$
5: $c \leftarrow (k - 8s^2 - (r_1 + r_2)s + d - f)/(8s + r_1)$
6: $e_0 = 8c + r_2 - r_1$
7: $e_1 = f$
8: $e_2 = 8s + r_1$
9: **return** $F(x, y) = 8x^2 - e_0 x + e_1 + y(8x + e_2)$.

RSA polynomials offer a solution for minimizing the storage requirements of systems. Instead of storing the factors of a semiprime number $n = pq$, the RSA-polynomial algorithm can be utilized to generate an RSA polynomial $F(x, y) = 8x^2 - e_0 x + e_1 + y(8x + e_2)$ with a non-trivial integer root $(x_0, y_0)$, ensuring that $x_0$ is sufficiently small, thereby reducing storage costs. Subsequently, we store $x_0$ and broadcast $F(x, y)$. By utilizing $x_0$ and $F(x, y)$, and without engaging in any time-consuming mathematical operations, the non-trivial factors are computed as $p = 8x_0 + e_2$ and $q = e_0 + e_2 - 8(x_0 + y_0)$.

The literature on breaking RSA and its connection to the hardness of factoring has been extensively discussed in [6–8]. The author emphasizes the RSA-polynomial problem and its hardness as follows. For more detailed information and proofs, we refer readers to [1].

**Problem 1** *([1]).* Consider an RSA-polynomial $F(x, y) = 8x^2 - e_0 x + e_1 + y(8x + e_2)$. The problem is to find the non-trivial integer root of $F(x, y)$ given $F(x, y)$.

**Theorem 1** *([1]). Solving the RSA-polynomial problem is at least as difficult as solving the factoring problem.*

Furthermore, utilizing the RSA-polynomial problem and the lattice basis reduction algorithm, the author proposes a strategy [1, Theorem 5] for factoring semiprimes, building upon the works of Coppersmith [9,10].

**Theorem 2** *([1]). Let $F(x, y) = 8x^2 - e_0 x + e_1 + y(8x + e_2)$ be an RSA-polynomial with the non-trivial integer root $(x_0, y_0)$. Suppose $|x_0| \le X$, $|y_0| \le Y$ and $W = \max\{8X^2, |e_0|X, |e_1|, 8XY, |e_2|Y\}$. If $X^2 Y < W^{1/2}/2^{15/2}$, then the non-trivial integer root of $F(x, y)$ can be computed in polynomial time.*

*Our contribution.* In this work, we revisit and address the problem of semiprime factorization based on RSA-polynomials by considering specific forms of bivariate polynomials. Our approach leverages advanced and refined lattice-based techniques and constructions, as presented in [11] and [12]. We enhance the existing RSA-polynomial based semiprime factoring attacks by thoroughly analyzing the unclear condition $X^2 Y < W^{1/2}/2^{15/2}$ and optimizing the effectiveness of factoring attacks.

Furthermore, we identify certain cases where Theorem 1 is not applicable and uncover potential vulnerabilities in Problem 1. As a result, we demonstrate that the proposed RSA-polynomial based commitment scheme in [1] does not meet the hiding property, enabling us to recover both the message and its corresponding opening value.

To support our claims, we conduct a series of numerical experiments based on examples generated using Algorithm 1. Through these experiments, we validate the efficacy and reliability of our proposed factoring attacks. The results obtained from these computer experiments serve as compelling evidence for the effectiveness of our approach.

*Organization.* The remaining content of this paper is structured as follows. In Section 2, we provide a comprehensive review of essential definitions and fundamental theorems that are crucial to our approach in solving bivariate polynomials of a specific form. Section 3 presents two improved theorems specifically used for RSA-polynomial based semiprime factorization. We address the previously ambiguous condition $X^2Y < W^{1/2}/2^{15/2}$ from [1] and introduce two novel lattice-based factoring attacks based on these enhanced theorems. To validate the practicality and efficiency of our factoring attacks, we conduct extensive experiments in Section 4. The experimental results are analyzed and discussed, shedding light on the feasibility of our proposed approach. Finally, in Section 5, we summarize the key findings of this paper.

## 2. Preliminaries

We provide an overview of the fundamental concepts and definitions involved in solving bivariate integer polynomials and introduce a significant theorem. We then present a parameterized theorem specifically tailored for solving bivariate integer polynomials with certain Newton polygons. To streamline the analysis in this paper, we omit a detailed discussion of lattice conceptions. For further details, refer to [10,13,11,12,14–16]. We first provide the formal definitions of asymptotic notations.

**Definition 3** *([17]).* Three asymptotic notations $O$, $\Theta$, and $o$ are defined as follows:

- $O$-notation describes an asymptotic upper bound. For a given function $g(n)$, we denote by $f(n) = O(g(n))$ such that there exist positive constants $c$ and $n_0$ satisfying $0 \le f(n) \le cg(n)$ for all $n \ge n_0$.
- $\Theta$-notation describes an asymptotic tight bound. For a given function $g(n)$, we denote by $f(n) = \Theta(g(n))$ such that there exist positive constants $c_1$, $c_2$, and $n_0$ satisfying $0 \le c_1 g(n) \le f(n) \le c_2 g(n)$ for all $n \ge n_0$.
- $o$-notation is provided to denote an upper bound that is not asymptotically tight. For a given function $g(n)$, we denote by $f(n) = o(g(n))$ such that for any positive constant $c > 0$, there exists a constant $n_0 > 0$ satisfying $0 \le f(n) < cg(n)$ for all $n \ge n_0$.

If an integer polynomial $f(x, y)$ cannot be factored further, it is considered irreducible. In such cases, if $f(x, y)$ can be written as the product of two other integer polynomials $g(x, y)$ and $h(x, y)$, both $g(x, y)$ and $h(x, y)$ must have absolute values equal to 1. To establish an index set for any monomial set $M$ containing variables $x$ and $y$, we define $\mathcal{I}_M$ as the collection of $(i, j) \in \mathbb{N}^2$ satisfying $x^i y^j \in M$. The convex hull associated with $\mathcal{I}_M$ is denoted as $\mathsf{ch}\{(i, j) \in \mathbb{N}^2 : x^i y^j \in M\}$. Additionally, the Newton polygon for $f(x, y)$ is defined as:

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

When solving bivariate integer polynomials, it is crucial to identify the Newton polygon of an integer polynomial and its polynomial norm. The definition of the polynomial norm is provided as follows. The $\ell_p$-norm of an integer polynomial $f(x, y) = \sum c_{ij} x^i y^j \in \mathbb{Z}[x, y]$ is $\|f(x, y)\|_p = \left(\sum |c_{ij}|^p\right)^{1/p}$, which is commonly used for solving bivariate integer polynomials, as seen in [18,11,19]. Notably, it can be derived directly from the above definition as $|f(x, y)|_\infty = \max\{|c_{ij}|\}$ for $f(x, y) = \sum c_{ij} x^i y^j$. To ensure the extraction of roots from a given bivariate integer polynomial, we provide the following definitions.

**Definition 4** *([11]).* Let $f(x, y)$ be a bivariate integer polynomial, and let $S$ and $M$ be two finite non-empty monomial sets in the variables $x$ and $y$. The sets $S$ and $M$ are called *admissible* for $f(x, y)$ if the following conditions hold:

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

**Definition 5** *([11]).* The Minkowski sum $\mathcal{I}_A + \mathcal{I}_B$ of two index sets $\mathcal{I}_A$ and $\mathcal{I}_B$ is defined as $\mathcal{I}_A + \mathcal{I}_B = \{(a_1, a_2) + (b_1, b_2) : (a_1, a_2) \in \mathcal{I}_A, (b_1, b_2) \in \mathcal{I}_B\}$.

It is straightforward to satisfy the first condition of Definition 4 by choosing $M$ such that $\mathcal{I}_M = \mathsf{N}(f) + \mathcal{I}_S$, where $\mathsf{N}(f)$ is the Newton polygon of the given integer polynomial $f(x, y)$ and $S$ is a given set. In almost all cases, this choice also satisfies the second condition of Definition 4, making $S$ and $M$ admissible for $f(x, y)$. We introduce the following lemma from [11], which establishes the admissibility of specific monomial sets $S$ and $M$ when the Newton polygon $\mathsf{N}(f)$ of $f(x, y)$ is an extended rectangle.

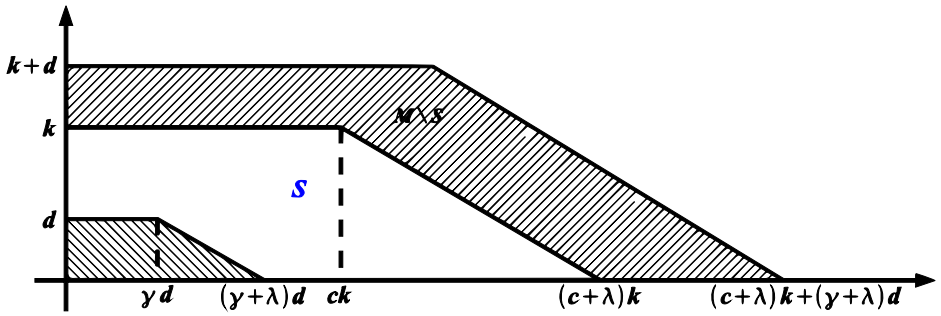


**Fig. 1.** The graphical extended rectangle construction and its corresponding Newton polygon indicated by the shaded area in the lower left corner.

**Lemma 2** *([11]). Assume that the Newton polygon* $\mathsf{N}(f)$ *of* $f(x,y)$ *is an extended rectangle, defined as* $\{(i,j)\in\mathbb{N}^2 : 0\le i\le \gamma d+\lambda(d-j),\ 0\le j\le d\}$ *for a positive integer* $d$ *and two positive real numbers* $\gamma$ *and* $\lambda$*. Then, the monomial sets* $S$ *and* $M$ *corresponding to the following index sets are admissible for* $f(x,y)$*:*

$$\mathcal{I}_S=\{(i,j)\in\mathbb{N}^2 : 0\le i\le ck+\lambda(k-j),\ 0\le j\le k\},$$
$$\mathcal{I}_M=\{(i,j)\in\mathbb{N}^2 : 0\le i\le ck+\gamma d+\lambda(k+d-j),\ 0\le j\le k+d\},$$

*where* $k\in\mathbb{N}$ *controls low-order error terms, and* $c>0$ *optimizes the solving bound.*

Fig. 1 illustrates the construction of an extended rectangle as described in [11], and further details and proofs can be found in [11, Lemma 7].

The Blömer-May theorem, as presented in [11], offers a strategy for extracting potential roots of bivariate integer polynomials.

**Theorem 3** *([11]). Consider an irreducible integer polynomial* $f(x,y)$ *in variables* $x$ *and* $y$*, where the degrees of* $x$ *and* $y$ *are at most* $d_x$ *and* $d_y$ *respectively. Let* $X$ *and* $Y$ *be the upper bounds on the potential root* $(x',y')$*,* $W$ *denote* $|f(xX,yY)|_\infty$*, and let* $S$ *and* $M$ *be two admissible monomial sets for* $f(x,y)$ *with* $S\subseteq M$*. Set* $s=|S|$ *as the cardinality of* $S$*,* $m=|M|$ *as the cardinality of* $M$*,* $s_x=\sum_{x^iy^j\in M\setminus S} i$ *as the sum of the* $i$ *values for monomials* $x^iy^j$ *in* $M\setminus S$*, and* $s_y=\sum_{x^iy^j\in M\setminus S} j$ *as the sum of the* $j$ *values for monomials* $x^iy^j$ *in* $M\setminus S$*. Then all potential* $(x',y')$ *satisfying* $f(x',y')=0$ *can be extracted in time polynomial in* $m$*,* $d_x$*,* $d_y$*, and* $\log W$ *provided* $X^{s_x}Y^{s_y}<W^s$*, under the assumption that* $(m-s)^2=O(sd_xd_y)$*.*

Note that the specific lattice-based proof for Theorem 3 is not included in this paper, but readers can refer to [11, Section 5] for a detailed explanation. The subsequent sections of this paper will further explore the application of Theorem 3 in the context of RSA-polynomial based semiprime factorization, providing insights into the efficiency and effectiveness of the proposed method.

While the Blömer-May theorem cannot be directly applied to RSA-polynomial based semiprime factorization, we present a parameterized theorem that encapsulates its essence and is tailored for solving bivariate integer polynomials with a specific Newton polygon structure.

**Theorem 4.** *Consider a bivariate integer polynomial* $f(x,y)=\sum c_{ij}x^iy^j$*, whose Newton polygon is an extended rectangle, i.e.,* $\mathsf{N}(f)=(i,j)\in\mathbb{N}^2 : 0\le i\le \gamma d+\lambda(d-j),\ 0\le j\le d$*, where* $d$ *is a positive integer and* $\gamma$ *and* $\lambda$ *are positive real numbers. The upper bounds for the roots* $(x',y')$ *are denoted as* $X$ *and* $Y$*, and* $W$ *is defined as* $|f(xX,yY)|_\infty$*. Under the condition that*

$$X^{c^2+2c\gamma+2c\lambda+\gamma\lambda+\lambda^2}Y^{2c+\gamma+\lambda}<W^{\frac{2c+\lambda}{d}}, \tag{1}$$

*where* $c>0$ *is an optimizing parameter, all potential roots* $(x',y')$ *of* $f(x,y)$ *can be solved in time polynomial in* $\log W$*.*

**Proof.** Note that $f(x,y)$ is an irreducible polynomial with a Newton polygon $\mathsf{N}(f)=\{(i,j)\in\mathbb{N}^2 : 0\le i\le \gamma d+\lambda(d-j),\ 0\le j\le d\}$. We can construct two admissible sets $S$ and $M$ according to Lemma 2, where

$$S=\{x^iy^j : 0\le i\le ck+\lambda(k-j),\ 0\le j\le k\},$$
$$M=\{x^iy^j : 0\le i\le ck+\gamma d+\lambda(k+d-j),\ 0\le j\le k+d\},$$

with $k\in\mathbb{N}$ being a sufficiently large number and $c>0$ being an optimizing parameter. Next, we calculate the values of $s$, $m$, $s_x$, and $s_y$ as stated in Theorem 3.

$$s=\sum_{j=0}^{k}\sum_{i=0}^{ck+\lambda(k-j)}1=\frac{2c+\lambda}{2}k^2+o(k^2),$$

$$m = \sum_{j=0}^{k+d} \sum_{i=0}^{ck+\gamma d+\lambda(k+d-j)} 1 = \frac{2c+\lambda}{2}k^2 + o(k^2),$$

$$s_x = \sum_{j=0}^{k+d} \sum_{i=0}^{ck+\gamma d+\lambda(k+d-j)} i - \sum_{j=0}^{k} \sum_{i=0}^{ck+\lambda(k-j)} i = \frac{d(c^2+2c\gamma+2c\lambda+\gamma\lambda+\lambda^2)}{2}k^2 + o(k^2),$$

$$s_y = \sum_{j=0}^{k+d} \sum_{i=0}^{ck+\gamma d+\lambda(k+d-j)} j - \sum_{j=0}^{k} \sum_{i=0}^{ck+\lambda(k-j)} j = \frac{d(2c+\gamma+\lambda)}{2}k^2 + o(k^2).$$

By substituting these values into the inequality $X^{s_x}Y^{s_y} < W^s$ and omitting lower order terms $o(k^2)$ for simplicity, we obtain

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

which can be simplified to

$$X^{c^2+2c\gamma+2c\lambda+\gamma\lambda+\lambda^2} Y^{2c+\gamma+\lambda} < W^{\frac{2c+\lambda}{d}}.$$

Furthermore, we have $d_x = (\gamma+\lambda)d$ and $d_y = d$, which implies $(m-s)^2 = O(sd_xd_y) = O(k^2)$. The time complexity is mainly determined by $\log W$ since $d_x,\ d_y \ll \log W$, and we set $k = \Theta(\log W)$. Therefore, the running time is polynomial in terms of $\log W$. □

Besides, Jochemsz and May [12] described a lattice-based method for finding small modular and integer roots of multivariate polynomials. In [12, Appendix B], a similar result for solving bivariate integer polynomials with the same Newton polygon was presented. The result is stated as follows.

**Theorem 5** *([12]). Given a bivariate integer polynomial $f(x,y) = \sum c_{ij}x^iy^j$, where its Newton polygon is an extended rectangle, i.e., $\mathsf{N}(f) = \{(i,j) \in \mathbb{N}^2 : 0 \le i \le \gamma d + \lambda(d-j),\ 0 \le j \le d\}$, with $d$ being a positive integer and $\gamma,\ \lambda$ being two positive real numbers. The upper bounds for the roots $(x', y')$ are denoted as $X$ and $Y$, and $W$ is defined as $|f(xX, yY)|_\infty$. Under the condition that*

$$X^{3\gamma^2+3\gamma\lambda+\lambda^2+4\gamma\tau+2\lambda\tau+\tau^2} Y^{3\gamma+\lambda+2\tau} < W^{\frac{2\gamma+\lambda+2\tau}{d}}, \tag{2}$$

*where $\tau \ge 0$ is an optimizing parameter, all potential roots $(x', y')$ of $f(x,y)$ can be solved in time polynomial in $\log W$.*

The proof of this theorem follows a similar manner as the one of Theorem 4, with the only difference being the definitions of the two admissible sets $S$ and $M$.

$$S = \{x^iy^j : 0 \le i \le \gamma d(m-1) + \lambda(d(m-1)-j) + \tau dm,\ 0 \le j \le d(m-1)\},$$

$$M = \{x^iy^j : 0 \le i \le \gamma dm + \lambda(dm-j) + \tau dm,\ 0 \le j \le dm\},$$

where $m \in \mathbb{N}$ is a sufficiently large number and $\tau \ge 0$ is an optimizing parameter. Similarly, we calculate the values of $|S|$, $|M|$, $s_x$, and $s_y$ as follows.

$$|S| = \sum_{j=0}^{d(m-1)} \sum_{i=0}^{\gamma d(m-1)+\lambda(d(m-1)-j)+\tau dm} 1 = \frac{d^2(2\gamma+\lambda+2\tau)}{2}m^2 + o(m^2),$$

$$|M| = \sum_{j=0}^{dm} \sum_{i=0}^{\gamma dm+\lambda(dm-j)+\tau dm} 1 = \frac{d^2(2\gamma+\lambda+2\tau)}{2}m^2 + o(m^2),$$

$$s_x = \sum_{j=0}^{dm} \sum_{i=0}^{\gamma dm+\lambda(dm-j)+\tau dm} i - \sum_{j=0}^{d(m-1)} \sum_{i=0}^{\gamma d(m-1)+\lambda(d(m-1)-j)+\tau dm} i = \frac{d^3(3\gamma^2+3\gamma\lambda+\lambda^2+4\gamma\tau+2\lambda\tau+\tau^2)}{2}m^2 + o(m^2),$$

$$s_y = \sum_{j=0}^{dm} \sum_{i=0}^{\gamma dm+\lambda(dm-j)+\tau dm} j - \sum_{j=0}^{d(m-1)} \sum_{i=0}^{\gamma d(m-1)+\lambda(d(m-1)-j)+\tau dm} j = \frac{d^3(3\gamma+\lambda+2\tau)}{2}m^2 + o(m^2).$$

By substituting these values into the inequality $X^{s_x}Y^{s_y} < W^{|S|}$ and omitting lower order terms $o(m^2)$ for simplicity, we obtain

$$X^{\frac{d^3(3\gamma^2+3\gamma\lambda+\lambda^2+4\gamma\tau+2\lambda\tau+\tau^2)}{2}m^2} Y^{\frac{d^3(3\gamma+\lambda+2\tau)}{2}m^2} < W^{\frac{d^2(2\gamma+\lambda+2\tau)}{2}m^2},$$

which is further reduced to

$$X^{3\gamma^2+3\gamma\lambda+\lambda^2+4\gamma\tau+2\lambda\tau+\tau^2} Y^{3\gamma+\lambda+2\tau} < W^{\frac{2\gamma+\lambda+2\tau}{d}}.$$

For completeness, we provide the details on how to construct the lattices for Theorem 4 and Theorem 5 by the shift polynomials and the basis matrix in Appendix A.1 and Appendix A.2, respectively.

## 3. Refined analysis with optimized factoring attacks

We carefully examine Theorem 2 and refine its ambiguous condition $X^2Y < W^{1/2}/2^{15/2}$, which provides a clearer and more precise requirement for the RSA-polynomial based factorization. This refinement enhances the performance of the lattice-based method significantly. In order to further optimize the factoring attacks, we introduce two improved lattice constructions based on Theorem 4 and Theorem 5.

By using the proposed factoring attacks, we demonstrate that Problem 1 is not a cryptographic hard problem in all cases. This finding affects the validity of Theorem 1. Additionally, we conduct a careful analysis of the commitment scheme proposed in [1]. We identify and address several flaws present in the original design. By fixing these issues, we improve the security and reliability of the commitment scheme significantly.

### 3.1. Revisiting RSA-polynomial based semiprime factorization

Before delving into the examination of Theorem 2, it is necessary to establish the bounds on various parameters present in the RSA-polynomial. For the sake of simplicity, we consider an RSA modulus $n = pq$ using two $\ell$-bit primes as a base quantity, while representing other parameters such as $e_0, e_1, e_2$ as distinct powers of 2. Moreover, we employ $\Theta$-notation to provide precise bounds on the involved parameters, ensuring $c_1, c_2$ occurred in Definition 3 remain within negligible constants relative to $n$. This facilitates the derivation of $X$, $Y$, and $W$, as used in Theorem 4 and Theorem 5.

To determine the parameter bounds, we introduce Lemma 3 based on Lemma 1. The following lemma provides three distinct cases, depending on the relationship between $s$ and $x_0$.

**Lemma 3.** *Let $F(x, y) = 8x^2 - e_0x + e_1 + y(8x + e_2)$ be an RSA-polynomial using an RSA modulus $n = pq$ with two primes of the same bit-length $\ell$. Suppose that $s$ and $x_0$ have bit-lengths of $\epsilon$ and $\xi$ respectively, implying $s = \Theta(2^\epsilon)$ and $x_0 = \Theta(2^\xi)$, other related parameters are bounded as follows.*

**Case 1.** *If $\xi < \epsilon = \ell - 3$, indicating that $s > x_0$ and $s$ dominates $p$, the parameter bounds are as follows:*

$$|k| = \Theta(2^{2\ell-3}),\ |d| = \Theta(1),\ |f| = O(2^\ell),\ |c| = \Theta(2^{\ell-3}),$$
$$|e_0| = \Theta(2^\ell),\ |e_1| = O(2^\ell),\ |e_2| = \Theta(2^\ell),\ |x_0| = \Theta(2^\xi),\ |y_0| = \Theta(2^\xi).$$

**Case 2.** *If $\epsilon = \xi = \ell - 4$, indicating that $s$ and $x_0$ have the same bit-length, the parameter bounds are as follows:*

$$|k| = \Theta(2^{2\ell-3}),\ |d| = \Theta(1),\ |f| = O(2^{\ell-1}),\ |c| = \Theta(2^{\ell-2}),$$
$$|e_0| = \Theta(2^{\ell+1}),\ |e_1| = O(2^{\ell-1}),\ |e_2| = \Theta(2^{\ell-1}),\ |x_0| = \Theta(2^{\ell-4}),\ |y_0| = \Theta(2^{\ell-4}).$$

**Case 3.** *If $\epsilon < \xi = \ell - 3$, indicating that $s < x_0$ and $x_0$ dominates $p$, the parameter bounds are as follows:*

$$|k| = \Theta(2^{2\ell-3}),\ |d| = \Theta(1),\ |f| = O(2^{\epsilon+3}),\ |c| = \Theta(2^{2\ell-\epsilon-6}),$$
$$|e_0| = \Theta(2^{2\ell-\epsilon-3}),\ |e_1| = O(2^{\epsilon+3}),\ |e_2| = \Theta(2^{\epsilon+3}),\ |x_0| = \Theta(2^{\ell-3}),\ |y_0| = \Theta(2^{2\ell-\epsilon-6}).$$

Lemma 3 divides the analysis into the above three cases, capturing the distinct relationships between $s$ and $x_0$ and providing clear parameter bounds for each scenario.

**Proof.** According to Lemma 1, we can determine the values of $r_1$, $r_2$, and $r$ based on the RSA-polynomial. These values are chosen from the small integers 1, 3, 5, or 7. Consequently, we have $|k| = (n - r)/8 = \Theta(pq/8) = \Theta(2^{2\ell-3})$ and $|d| = |(r - r_1r_2)|/8 = \Theta(1)$. Since $s = \Theta(2^\epsilon)$ and $x_0 = \Theta(2^\xi)$ with the known relation $p = 8(s + x_0) + r_1$ from Lemma 1, we can categorize the analysis into three cases based on the distinct relationships between $s$ and $x_0$.

Considering the definition of $f = (k - 8s^2 - (r_1 + r_2)s + d) \pmod{(8s + r_1)}$, we find that $|f| = O(8s + r_1) = O(2^{\epsilon+3})$. Similarly, we obtain $|c| = \Theta(\max\{k/(8s), s\})$ for $c = (k - 8s^2 - (r_1 + r_2)s + d - f)/(8s + r_1)$. As $s$ ranges from 1 to $(p - r_1)/8 - 1$, we conclude that $\epsilon \le \ell - 3$ and thus $\epsilon \le 2\ell - \epsilon - 6$. Consequently, we have $|c| = \Theta(\max\{2^{2\ell-\epsilon-6}, 2^\epsilon\}) = \Theta(2^{2\ell-\epsilon-6})$.[1]

Regarding $e_0 = 8c + r_2 - r_1$, $e_1 = f$, and $e_2 = 8s + r_1$, it directly follows that $|e_0| = \Theta(2^{2\ell-\epsilon-3})$, $|e_1| = O(2^{\epsilon+3})$, and $|e_2| = \Theta(2^{\epsilon+3})$. Finally, we aim to bound $y_0$ based on $F(x_0, y_0) = 8x_0^2 - e_0x_0 + e_1 + y_0(8x_0 + e_2) = 0$. We have $|y_0| = |(e_0x_0 - 8x_0^2 - e_1)/(8x_0 + e_2)|$, and its value depends on the relationship between $s$ and $x_0$. We can divide the analysis into the following three cases.

[1] Fortunately, we obtain this bound even if the special case where $k$ and $8s^2$ have the same bit-length occurs (i.e., when $s$ has bit-length of $\ell - 3$). The dominant component of the numerator of $|c|$ is $|k - 8s^2| = \Theta(|(n - r)/8 - (p - r_1)^2/8|) = \Theta(|n - p^2|/8) = \Theta(2^{2\ell-3})$ with overwhelming probability. Therefore, we deduce that $|c| = \Theta(|k - 8s^2|/(8s)) = \Theta(2^{2\ell-\epsilon-6})$.

**Case 1.** Suppose $\xi < \epsilon = \ell - 3$, which implies $s > x_0$ and $s$ dominates $p$. In this case, we can determine the following bounds:

$$|k| = \Theta(2^{2\ell-3}),\ |d| = \Theta(1),\ |f| = O(2^{\epsilon+3}) = O(2^{\ell}),\ |c| = \Theta(2^{2\ell-\epsilon-6}) = \Theta(2^{\ell-3}),$$
$$|e_0| = \Theta(2^{2\ell-\epsilon-3}) = \Theta(2^{\ell}),\ |e_1| = O(2^{\epsilon+3}) = O(2^{\ell}),\ |e_2| = \Theta(2^{\epsilon+3}) = \Theta(2^{\ell}),\ |x_0| = \Theta(2^{\xi}).$$

As $e_2 = 8s + r_1 > 8x_0$, we can deduce that $|y_0| = |(e_0x_0 - 8x_0^2 - e_1)/(8x_0 + e_2)| = \Theta(\max\{e_0x_0/e_2, 8x_0^2/e_2, e_1/e_2\})$. Thus, we have

$$|y_0| = \Theta(\max\{2^{2\ell-2\epsilon+\xi-6}, 2^{2\xi-\epsilon}, 1\}) = \Theta(\max\{2^{\xi}, 2^{2\xi-\epsilon}\}) = \Theta(2^{\xi}).$$

**Case 2.** Suppose $\epsilon = \xi = \ell - 4$, which implies $s$ and $x_0$ are of the same bit-length. In this case, we can determine the following bounds:

$$|k| = \Theta(2^{2\ell-3}),\ |d| = \Theta(1),\ |f| = O(2^{\epsilon+3}) = O(2^{\ell-1}),\ |c| = \Theta(2^{2\ell-\epsilon-6}) = \Theta(2^{\ell-2}),$$
$$|e_0| = \Theta(2^{2\ell-\epsilon-3}) = \Theta(2^{\ell+1}),\ |e_1| = O(2^{\epsilon+3}) = O(2^{\ell-1}),\ |e_2| = \Theta(2^{\epsilon+3}) = \Theta(2^{\ell-1}),\ |x_0| = \Theta(2^{\xi}) = \Theta(2^{\ell-4}).$$

As $e_2 = 8s + r_1$ and $8x_0$ are of the same bit-length, we can deduce that $|y_0| = |(e_0x_0 - 8x_0^2 - e_1)/(8x_0 + e_2)| = \Theta(\max\{e_0/8, x_0, e_1/(8x_0)\})$. Thus, we have

$$|y_0| = \Theta(\max\{2^{2\ell-\epsilon-6}, 2^{\xi}, 1\}) = \Theta(\max\{2^{\ell-2}, 2^{\ell-4}\}) = \Theta(2^{\ell-2}).$$

**Case 3.** Suppose $\epsilon < \xi = \ell - 3$, which implies $s < x_0$ and $x_0$ dominates $p$. In this case, we can determine the following bounds:

$$|k| = \Theta(2^{2\ell-3}),\ |d| = \Theta(1),\ |f| = O(2^{\epsilon+3}),\ |c| = \Theta(2^{2\ell-\epsilon-6}),$$
$$|e_0| = \Theta(2^{2\ell-\epsilon-3}),\ |e_1| = O(2^{\epsilon+3}),\ |e_2| = \Theta(2^{\epsilon+3}),\ |x_0| = \Theta(2^{\xi}) = \Theta(2^{\ell-3}).$$

As $e_2 = 8s + r_1 < 8x_0$, we can deduce that $|y_0| = |(e_0x_0 - 8x_0^2 - e_1)/(8x_0 + e_2)| = \Theta(\max\{e_0/8, x_0, e_1/(8x_0)\})$. Thus, we have

$$|y_0| = \Theta(\max\{2^{2\ell-\epsilon-6}, 2^{\xi}, 1\}) = \Theta(\max\{2^{2\ell-\epsilon-6}, 2^{\ell-3}\}) = \Theta(2^{2\ell-\epsilon-6})$$

since $2\ell - \epsilon - 6 > \ell - 3$, which is deduced from $\epsilon < \ell - 3$.

This completes the proof. □

In order to apply the RSA-polynomial problem to semiprime factorization, we follow Theorem 2 and determine the values of $X$, $Y$, and $W$, where $|x_0| \le X$, $|y_0| \le Y$ and $W = \max\{8X^2, |e_0|X, |e_1|, 8XY, |e_2|Y\}$.

**Theorem 6.** *The values of $X$, $Y$, $W$ depend on the distinct relationships between $s$ and $x_0$ and are determined as follows.*

**Case 1.** *Suppose $\xi < \epsilon = \ell - 3$, which implies $s > x_0$ and $s$ dominates $p$. In this case, we have $X = 2^{\xi}$, $Y = 2^{\xi}$, and $W = 2^{\ell+\xi}$.*
**Case 2.** *Suppose $\epsilon = \xi = \ell - 4$, which implies $s$ and $x_0$ are of the same bit-length. In this case, we have $X = 2^{\ell-4}$, $Y = 2^{\ell-4}$, and $W = 2^{2\ell-3}$.*
**Case 3.** *Suppose $\epsilon < \xi = \ell - 3$, which implies $s < x_0$ and $x_0$ dominates $p$. In this case, we have $X = 2^{\ell-3}$, $Y = 2^{2\ell-\epsilon-6}$, and $W = 2^{3\ell-\epsilon-6}$.*

**Proof.** We can substitute the values of $X$ and $Y$ into the definition to calculate $W$. Let us consider the three cases based on the distinct relationships between $s$ and $x_0$.

**Case 1.** Suppose $\xi < \epsilon = \ell - 3$, which implies $s > x_0$ and $s$ dominates $p$. From Lemma 3, we have $X = Y = 2^{\xi}$, and we can calculate $W$ as follows:

$$W = \max\{8X^2, |e_0|X, |e_1|, 8XY, |e_2|Y\} = \max\{2^{2\xi+3}, 2^{\ell+\xi}, 2^{\ell}, 2^{2\xi+3}, 2^{\ell+\xi}\} = 2^{\ell+\xi}$$

since $\ell + \xi > 2\xi + 3$.
**Case 2.** Suppose $\epsilon = \xi = \ell - 4$, which implies $s$ and $x_0$ are of the same bit-length. From Lemma 3, we have $X = Y = 2^{\ell-4}$, and we can calculate $W$ as follows:

$$W = \max\{8X^2, |e_0|X, |e_1|, 8XY, |e_2|Y\} = \max\{2^{2\ell-5}, 2^{2\ell-3}, 2^{\ell-1}, 2^{2\ell-5}, 2^{2\ell-5}\} = 2^{2\ell-3}.$$

**Case 3.** Suppose $\epsilon < \xi = \ell - 3$, which implies $s < x_0$ and $x_0$ dominates $p$. From Lemma 3, we have $X = 2^{\ell-3}$ and $Y = 2^{2\ell-\epsilon-6}$, and we can calculate $W$ as follows:

$$W = \max\{8X^2, |e_0|X, |e_1|, 8XY, |e_2|Y\} = \max\{2^{2\ell-3}, 2^{3\ell-\epsilon-6}, 2^{\epsilon+3}, 2^{3\ell-\epsilon-6}, 2^{2\ell-3}\} = 2^{3\ell-\epsilon-6}$$

since $3\ell - \epsilon - 6 > 2\ell - 3$ and $3\ell - \epsilon - 6 > \epsilon + 3$.

This completes the proof. □

Now we examine Theorem 2 with the given unclear condition $X^2Y < W^{1/2}2^{15/2}$ and present a refined analysis in the following proposition.

**Proposition 1.** *Consider the RSA-polynomial $F(x,y) = 8x^2 - e_0x + e_1 + y(8x + e_2)$ using an RSA modulus $n = pq$ with two primes of the same bit-length $\ell$. Assuming $s$ and $x_0$ have bit-lengths of $\epsilon$ and $\xi$ respectively, the condition $X^2Y < W^{1/2}2^{15/2}$ in Theorem 2 is refined to $\xi < \ell/5 - 3$. Moreover, the probability for conducting a successful factoring attack is $2^{-4\ell/5}$.*

**Proof.** We examine the unclear condition $X^2Y < W^{1/2}2^{15/2}$ in the three cases mentioned previously.

**Case 1.** In this case, we have $\xi < \epsilon = \ell - 3$ and $X = 2^{\xi}$, $Y = 2^{\xi}$, $W = 2^{\ell+\xi}$. Substituting these values into the condition, $X^2Y < W^{1/2}2^{15/2}$ becomes $2^{2\xi}2^{\xi} < 2^{(\ell+\xi)/2}/2^{15/2}$. Simplifying further, we get $2^{5\xi-\ell+15} < 1$, which implies $\xi < \ell/5 - 3$.

**Case 2.** In this case, we have $\epsilon = \xi = \ell - 4$ and $X = 2^{\ell-4}$, $Y = 2^{\ell-4}$, $W = 2^{2\ell-3}$. Substituting these values into the condition, $X^2Y < W^{1/2}2^{15/2}$ becomes $2^{2\ell-8}2^{\ell-4} < 2^{(2\ell-3)/2}/2^{15/2}$. Simplifying further, we get $2^{2\ell-3} < 1$, which implies $\ell < 3/2$.

**Case 3.** In this case, we have $\epsilon < \xi = \ell - 3$ and $X = 2^{\ell-3}$, $Y = 2^{2\ell-\epsilon-6}$, $W = 2^{3\ell-\epsilon-6}$. Substituting these values into the condition, $X^2Y < W^{1/2}2^{15/2}$ becomes $2^{2\ell-6}2^{2\ell-\epsilon-6} < 2^{(3\ell-\epsilon-6)/2}/2^{15/2}$. Simplifying further, we get $2^{5\ell-\epsilon-3} < 1$, which implies $\epsilon > 5\ell - 3$.

From the analysis, it is evident that only **Case 1** is valid, while **Case 2** and **Case 3** are invalid. Therefore, the refined condition for $X^2Y < W^{1/2}2^{15/2}$ is $\xi < \ell/5 - 3$. When $\xi < \ell/5 - 3$, the values of $x_0$ ranging from 1 to $2^{\ell/5-3} - 1$ are vulnerable. Furthermore, since $s$ is chosen from 1 to $(p - r_1)/8 - 1$ as shown in Algorithm 1, we can deduce that $x_0$ can be chosen from 1 to $(p - r_1)/8 - 1$ (i.e., $2^{\ell-3} - 1$). Consequently, the probability of conducting a successful factoring attack is approximately

$$\frac{2^{\ell/5-3} - 1}{2^{\ell-3} - 1} \approxeq 2^{-4\ell/5}.$$

This completes the proof. □

### 3.2. Optimizing RSA-polynomial based semiprime factoring

We aim to enhance and optimize the performance of semiprime factorization by utilizing more powerful tools, namely Theorem 4 and Theorem 5. As discussed in the proof of Proposition 1, we focus on the valid case where $\xi < \epsilon = \ell - 3$ and $X = 2^{\xi}$, $Y = 2^{\xi}$, $W = 2^{\ell+\xi}$ and propose the following theorems.

**Theorem 7.** *Consider an RSA-polynomial $F(x,y) = 8x^2 - e_0x + e_1 + y(8x + e_2)$, where $n = pq$ is an RSA modulus with two primes of the same bit-length $\ell$, and assume it possesses a non-trivial integer root $(x_0, y_0)$. Assuming $x_0$ has bit-length of $\xi$, then the non-trivial integer root can be computed, leading to the polynomial-time factorization of $n$ if $\xi < (3 - \sqrt{5})\ell/2$.*

**Proof.** Let $F(x,y) = 8x^2 - e_0x + e_1 + y(8x + e_2)$ satisfy Lemma 1 and Definition 2. We aim to apply Theorem 4 with $d = \gamma = \lambda = 1$ in (1) and obtain

$$X^{c^2+4c+2}Y^{2c+2} < W^{2c+1}.$$

By substituting $X = 2^{\xi}$, $Y = 2^{\xi}$, $W = 2^{\ell+\xi}$ and dealing with the exponents over 2, we have

$$\xi(c^2 + 4c + 2) + \xi(2c + 2) < (\ell + \xi)(2c + 1),$$

which simplifies to

$$\xi < \frac{2c + 1}{c^2 + 4c + 3}\ell.$$

By setting $c = (\sqrt{5} - 1)/2$ to maximize the right side, we obtain

$$\xi < \frac{3 - \sqrt{5}}{2}\ell. \tag{3}$$

The executing time is a polynomial regarding $\log W$, as well as a polynomial regarding $\log n$. After extracting the integer root $(x_0, y_0)$, the primes $p$ and $q$ can be computed as $p = 8x_0 + e_2$ and $q = e_0 + e_2 - 8(x_0 + y_0)$, respectively. This completes the proof. □

**Theorem 8.** *Consider an RSA-polynomial $F(x,y) = 8x^2 - e_0x + e_1 + y(8x + e_2)$, where $n = pq$ is an RSA modulus with two primes of the same bit-length $\ell$, and assume it possesses a non-trivial integer root $(x_0, y_0)$. Assuming $x_0$ has bit-length of $\xi$, then the non-trivial integer root can be computed, leading to the polynomial-time factorization of $n$ if $\xi < 3\ell/8$.*

**Proof.** Let $F(x,y) = 8x^2 - e_0x + e_1 + y(8x + e_2)$ satisfy Lemma 1 and Definition 2. We aim to apply Theorem 5 with $d = \gamma = \lambda = 1$ in (2) and obtain

$$X^{\tau^2+6\tau+7}Y^{2\tau+4} < W^{2\tau+3}.$$

By substituting $X = 2^{\xi}$, $Y = 2^{\xi}$, $W = 2^{\ell+\xi}$ and considering the exponents over 2, we have

$$\xi(\tau^2 + 6\tau + 7) + \xi(2\tau + 4) < (\ell + \xi)(2\tau + 3),$$

which simplifies to

$$\xi < \frac{2\tau + 3}{\tau^2 + 6\tau + 8}\ell.$$

By setting $\tau = 0$ to maximize the right side, we obtain

$$\xi < \frac{3}{8}\ell. \tag{4}$$

The executing time is a polynomial regarding $\log W$, as well as a polynomial regarding $\log n$. After extracting the integer root $(x_0, y_0)$, the primes $p$ and $q$ can be computed as $p = 8x_0 + e_2$ and $q = e_0 + e_2 - 8(x_0 + y_0)$, respectively. This completes the proof. □

As observed, the factoring bound has improved from $\xi < \ell/5 - 3$ to $\xi < \max\{(3 - \sqrt{5})\ell/2, 3\ell/8\}$, i.e., $\xi < 0.3819\ell$, which corresponds to an increment of approximately 90%. Moreover, we demonstrate that Problem 1 has several weak cases where the factorization of the RSA modulus can be efficiently computed. Likewise, the probability of conducting a successful factoring attack is approximately $2^{(3-\sqrt{5})\ell/2}/2^{\ell-3} \approxeq 2^{(1-\sqrt{5})\ell/2} \approxeq 2^{-0.618\ell}$.

For completeness, we present an improved approach to factor RSA semiprimes in Algorithm 2, which relies on the following definition.

**Definition 6.** Let $\mathsf{Rem}_r$ be defined 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 $\mathsf{Rem}_7 = \{(1,7),(7,1),(3,5),(5,3)\}$.

The following algorithm outputs the non-trivial factors of a semiprime when it generates an RSA-polynomial satisfying Theorem 7 or Theorem 8. By randomly choosing a lucky $s$ for a given semiprime $n$, we can successfully find the factorization of $n$ with its prime factors. We have made two improvements: modifying the selection of a random $s$ to make the factoring attack more efficient and applying Theorem 7 or Theorem 8 instead of the original Theorem 2 to increase the success rate. Our improved RSA-polynomial based factoring algorithm serves as a candidate for the lattice-based modulus factorization problem, in addition to [9,20–26,16].

**Algorithm 2** Improved RSA-polynomial based factoring algorithm.

**Require:** A semiprime $n$ whose bit-length is $2\ell$.
**Ensure:** The factorization of $n = pq$ with its primes $p$ and $q$.
$r \leftarrow n \pmod 8$
$s \leftarrow$ a random $(\ell - 3)$-bit integer
**for** $(r_1, r_2) \in \mathsf{Rem}_r$ **do** ▷ $\mathsf{Rem}_r$ is given in Definition 6
  $F(x,y) = 8x^2 - e_0x + e_1 + y(8x + e_2) \leftarrow$ Algorithm 1 with the input $(n, r_1, r_2, s)$
  $(x_0, y_0) \leftarrow$ Theorem 7 or Theorem 8 with the input $F(x,y)$
  **if** $x_0$ and $y_0$ exist as integers **then**
    $p \leftarrow 8x_0 + e_2$
    $q \leftarrow e_0 + e_2 - 8(x_0 + y_0)$
    **if** $p \times q = n$ **then**
      **return** $p$ and $q$ ▷ The factorization of $n = pq$ is found
    **else**
      go back to Step 2
    **end if**
  **else**
    go back to Step 2
  **end if**
**end for**

Note that while the improved RSA-polynomial based factoring algorithm increases the success rate of factoring semiprimes, it does not guarantee success for all instances. There may still be cases where the algorithm fails to find a non-trivial root. Additionally, the algorithm may take different amounts of time to run depending on the specific parameters of the RSA modulus and the random choices made during execution. Nonetheless, the improvements presented in Theorem 7 and Theorem 8 provide a more efficient approach to factoring semiprimes than Theorem 1 using the same RSA-polynomial.

### 3.3. Breaking RSA-polynomial based commitment scheme

In [1], the RSA-polynomial based commitment scheme was introduced, with a focus on the commitment phase and verifying phase. The commitment phase involves the following steps for committing a message $m \in [1, 2^{\ell-1} - 1]$, where $\ell \geq \kappa$ and $\kappa$ is the security parameter.

1. Generate two random primes $p$ and $q$ of the same bit-length $\ell$, such that $n = pq$ is a secure semiprime.
2. Compute $n = pq$, $r_1 = p \pmod 8$, and $r_2 = q \pmod 8$.
3. Compute $s = (p - r_1)/8 - m$.
4. Execute Algorithm 1 with the input $(n, r_1, r_2, s)$ to obtain an RSA-polynomial $F(x, y) = 8x^2 - e_0 x + e_1 + y(8x + e_2)$.
5. Compute $o = (8m^2 - e_0 m + e_1)/(8m + e_2)$.
6. Send $(n,\ F(x, y))$ as the witness to the receiver.
7. Save $o$ as the opening value.

The author claims that the RSA-polynomial based commitment scheme satisfies the hiding property, implying that a polynomial-time malicious receiver cannot compute $m$ or $o$ using the known witness. However, as we have demonstrated, it is possible for a polynomial-time malicious receiver to break this commitment scheme when committing a short message $m < 2^{(3-\sqrt{5})\ell/2}$. In such cases, the message $m$ and its corresponding opening value $o$, which are considered as a non-trivial root of the RSA-polynomial, can be recovered by solving the polynomial. This leads us to the following proposition.

**Proposition 2.** *Given the commitment phase of RSA-polynomial based commitment scheme, as described in [1] with a predetermined parameter $\ell$, suppose that a message $m$ is of bit-length $\xi$, then $m$ and the opening value $o$ can be computed in polynomial time if $\xi < (3 - \sqrt{5})\ell/2$.*

The proof of Proposition 2 is a direct consequence of Theorem 7, which we omit here. Furthermore, the upper bound $2^{\ell-1} - 1$ for $m$ is also infeasible because $s = (p - r_1)/8 - m$ might be a negative integer, which conflicts with the condition $1 \leq s \leq (p - r_1)/8 - 1$ in Algorithm 1. Hence, to ensure that $s$ remains a positive integer, the maximum bit-length of $m$ must be $\ell - 4$. Taking these factors into consideration, we can refine the range of $m$ to $m \in [2^{(3-\sqrt{5})\ell/2-1}, 2^{\ell-4} - 1]$.

## 4. Experimental results

In order to verify the validity and effectiveness of our proposed factoring attacks, which rely on Theorem 7 and Theorem 8, we conducted a series of numerical experiments. These experiments were carried out on a computer running a 64-bit Windows 10 operating system with Ubuntu 22.04 installed on WSL 2. The system was equipped with a CPU operating at 2.80 GHz and 16 GB of RAM. To perform the experiments, the LLL algorithm [27], which is readily available in the `SageMath` [28] software platform, was used for lattice reduction.

To test the proposed RSA-polynomial based factoring attacks, we randomly generated two prime numbers of bit-length $\ell$. Then, we generated a random value $s$ with a bit-length $\xi$, which represents the unknown variable $x_0$. Subsequently, we calculated the values of $e_0$, $e_1$, and $e_2$ using Algorithm 1, with input parameters $n$, $s$, $r_1$ and $r_2$. By following this process, we constructed RSA-polynomials $F(x, y)$ that satisfied Definition 2.

During the experiments, we carefully selected parameters $k$ and $c$ for Theorem 7 (or $m$ and $\tau$ for Theorem 8) to control the lattice settings and perform the proposed factoring attacks. Each simulated experiment instance was tested five times, ensuring the successful extraction of the desired integer root. We collected numerous polynomial equations that satisfied the solvable requirements and used the resultant computation approach to extract the desired integer root. We have compiled the results from our factoring attacks in Table 1, showing the outcomes of our experiments.

The column labeled '$\xi_e$' in Table 1 presents the experimental number of bits required for a successful factoring attack. The column labeled '$\xi_t$' corresponds to the theoretical number of bits required for executing the proposed factoring attacks, as specified in Theorem 7 and Theorem 8. The 'Theorem' column indicates the particular theorem employed in our simulated RSA-polynomial instances. The related lattice parameters are given in '$k$' and '$c$' columns for using Theorem 4, or '$m$' and '$\tau$' columns for using Theorem 5, which result in the lattice dimension denoted by '$\omega$'. The running time of each experiment is recorded in the 'Time' column, measured in seconds.

In each experiment, we gathered a sufficient number of integer polynomials that shared a common root. From this collection, we carefully selected a subset to solve the shared root. By obtaining the values of $x_0$ and $y_0$, we calculated $p = 8x_0 + e_2$ and $q = e_0 + e_2 - 8(x_0 + y_0)$, which led to the factorization of $n = pq$. Through these experiments, we successfully verified the validity and effectiveness of the proposed factoring attacks. However, when comparing the values of $\xi_e$ with $\xi_t$ in Table 1, we observed that the experimental results fall short of the theoretical ones by several bits. This discrepancy may be attributed to the limitation of computing resources, which prevented us from achieving a sufficiently large lattice dimension.

Based on the observations from Table 1, we conclude that Theorem 7 slightly outperforms Theorem 8 in terms of efficiency. Specifically, the factoring attacks based on Theorem 7 yield identical experimental results compared to those based on Theorem 8 when applying lower-dimensional lattices (30 vs. 35). This phenomenon is also compatible with the theoretical comparison between

**Table 1**
Experimental results of RSA-polynomial based factoring attacks with various settings.

| $\ell$ | $\xi_e$ | $\xi_t$ | Theorem | $k$ | $c$ | $m$ | $\tau$ | $\omega$ | Time |
|---|---|---|---|---|---|---|---|---|---|
| 512 | 164 | 196 | Theorem 7 | 3 | 2/3 | - | - | 30 | 0.57 s |
| 512 | 168 | 196 | Theorem 7 | 4 | 1/2 | - | - | 39 | 2.55 s |
| 512 | 177 | 196 | Theorem 7 | 6 | 2/3 | - | - | 76 | 76.26 s |
| 1024 | 328 | 391 | Theorem 7 | 3 | 2/3 | - | - | 30 | 1.53 s |
| 1024 | 336 | 391 | Theorem 7 | 4 | 1/2 | - | - | 39 | 7.84 s |
| 1024 | 353 | 391 | Theorem 7 | 6 | 2/3 | - | - | 76 | 291.87 s |
| 2048 | 656 | 782 | Theorem 7 | 3 | 2/3 | - | - | 30 | 4.68 s |
| 2048 | 672 | 782 | Theorem 7 | 4 | 1/2 | - | - | 39 | 26.29 s |
| 2048 | 693 | 782 | Theorem 7 | 5 | 3/5 | - | - | 56 | 67.17 s |
| 512 | 164 | 192 | Theorem 8 | - | - | 4 | 0 | 35 | 1.51 s |
| 512 | 174 | 192 | Theorem 8 | - | - | 6 | 0 | 70 | 52.28 s |
| 512 | 180 | 192 | Theorem 8 | - | - | 9 | 0 | 145 | 3096.38 s |
| 1024 | 328 | 384 | Theorem 8 | - | - | 4 | 0 | 35 | 4.71 s |
| 1024 | 348 | 384 | Theorem 8 | - | - | 6 | 0 | 70 | 163.83 s |
| 1024 | 356 | 384 | Theorem 8 | - | - | 8 | 0 | 117 | 4499.55 s |
| 2048 | 655 | 768 | Theorem 8 | - | - | 4 | 0 | 35 | 17.97 s |
| 2048 | 696 | 768 | Theorem 8 | - | - | 6 | 0 | 70 | 626.35 s |
| 2048 | 721 | 768 | Theorem 8 | - | - | 9 | 0 | 145 | 37 171.04 s |

Theorem 7 and Theorem 8. Therefore, we recommend using the factoring attack based on Theorem 7 for the most practical efficiency considerations. Furthermore, we provide the following example to aid in numerical understanding.

**Example 1.** We present a numerical example to illustrate the RSA-polynomial based semiprime factoring attack using Theorem 7. In this example, we set two prime numbers $p$ and $q$ to be 256 bits in length, indicating that $\ell = 256$. Assuming that Algorithm 1 is executed with a significantly large value of $s$, resulting in a small value of $x_0$ with bit-length of 69. The example instance is presented below.

$$n = 11869019022398882792869704924422803005755873694436907378545478090442472070037754\backslash$$
$$9773490026013454042986799996665544792746421978860683499484905575675797780 81,$$
$$e_0 = 3265922621583208002220893453194353450541585055549787640966395335905379852312,$$
$$e_1 = 7887209561259071501257582781452160932649574432983170745873909329280277102 3899,$$
$$e_2 = 1073243036376340472105078295831763128101379473504450853462037633079357630 10959.$$

We proceed by deriving the RSA-polynomial $F(x, y) = 8x^2 - e_0x + e_1 + y(8x + e_2)$ with the known parameters $e_0$, $e_1$, and $e_2$. To conduct the proposed factoring attack using Theorem 7, we set $k = 1$ and $c = 1$ and hence construct a 12-dimensional lattice. After less than one second, we successfully extract the non-trivial root $(x_0, y_0)$ that satisfies the RSA-polynomial equation mentioned above. The obtained root is presented below.

$$x_0 = 322245836007467197621,$$
$$y_0 = 9806073087426374675.$$

Thus, we compute $p = 8x_0 + e_2$ and $q = e_0 + e_2 - 8(x_0 + y_0)$ as follows.

$$p = 1073243036376340472105078295831763128101379473504450853487817299959955005 91927,$$
$$q = 1105902262592172552127287230363706662606795324059948729845137433710819942 84903.$$

It is easy to check that $n = pq$ does hold. Therefore, Theorem 7 is successfully applied to semiprime factorization using the RSA-polynomial. Moreover, we confirm that it is possible for a polynomial-time malicious attacker to break RSA-polynomial based commitment scheme when committing a short message.

Interestingly, this numerical example can also be resolved using Theorem 8 with $m = 2$ and $\tau = 0$ (where the parameters even yield the same lattice basis matrix). The lattice construction details including the underlying lattice basis matrix are provided in Appendix A.3.

## 5. Conclusion

We have revisited the RSA-polynomial problem and its applications, aiming to provide a more thorough analysis. Specifically, we conducted a refined analysis of the unclear condition presented in [1, Theorem 5] and derived a precise attack condition by

incorporating several given parameters. Additionally, we proposed two optimized factoring attacks based on improved theorems that efficiently solve bivariate integer polynomials of a specific form. Our findings revealed previously unidentified weak cases of Problem 1 and expanded the vulnerable bound by an impressive 90% increment. To validate the efficacy of our proposed factoring attacks, we provided both theoretical analysis and experimental results.

Moreover, we highlighted certain deficiencies in the related applications, i.e., the RSA-polynomial based semiprime factorization and commitment scheme, as presented in [1]. To address these issues, we employed sharper lattice-based techniques to optimize RSA-polynomial based factoring attacks and enhance the RSA-polynomial based commitment scheme.

## CRediT authorship contribution statement

**Mengce Zheng:** Conceptualization, Data curation, Funding acquisition, Investigation, Methodology, Validation, Writing – original draft, Writing – review & editing.

## Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Data availability

Data will be made available on request.

## Acknowledgements

The author would like to thank the anonymous reviewers for their detailed comments and useful suggestions, which improved this paper in terms of both technical and editorial quality. This work was supported by the National Natural Science Foundation of China, grant number 62002335, Ningbo Natural Science Foundation, grant number 2021J174 and Ningbo Young Science and Technology Talent Cultivation Program, grant number 2023QL007.

## Appendix A. Details on lattice constructions

The lattice constructions primarily revolve around obtaining basis vectors for a lattice basis matrix. These vectors stem from coefficient vectors of shift polynomials related to two monomial sets $S$ and $M$. Here, we illustrate the process of generating shift polynomials and organizing coefficient vectors. Consequently, we can establish an upper triangular basis matrix for specific chosen parameters. It is worth noting that the lattice constructions for Theorem 4 and Theorem 5 are nearly identical except for monomial sets $S$, $M$ and the definition of a crucial modulus $R$.

### A.1. Lattice construction for Theorem 4

Suppose that the upper bounds $X$ and $Y$ on $x$, $y$ along with the maximal norm $W$ are derived. We use the monomial sets $S$ and $M$ presented in Theorem 4 with chosen $k$ and $c$ to generate shift polynomials.

$$S = \{x^i y^j : 0 \le i \le ck + \lambda(k-j),\ 0 \le j \le k\},$$
$$M = \{x^i y^j : 0 \le i \le ck + \gamma d + \lambda(k+d-j),\ 0 \le j \le k+d\}.$$

Before that, we require the constant term of $f(x,y)$, i.e., $c_{00}$ to be 1. Thus, we should define a modular polynomial $\bar{f}(x,y) = c_{00}^{-1} f(x,y) \pmod R$, where $R = WX^{(c+\lambda)k}Y^k$. The shift polynomials $g_{[i,j]}(x,y)$ according to $S$ and $M$ are defined as follows:

$$g_{[i,j]}(x,y) := \begin{cases} \dfrac{x^i y^j \bar{f} \cdot R}{WX^i Y^j}, & x^i y^j \in S, \\ x^i y^j \cdot R, & x^i y^j \in M \setminus S. \end{cases}$$

The coefficient vectors of $g_{[i,j]}(xX, yY)$ are used in the construction of lattice basis matrix.

We further specify the polynomial order $\prec_{\mathsf{p}}$ (resp. the monomial order $\prec_{\mathsf{m}}$) related to the row order (resp. the column order) of the lattice basis matrix. The general principle is to place the polynomials $g_{[i,j]}$ (resp. the monomials $x^i y^j$) regarding $S$ before those regarding $M \setminus S$. The polynomial order $\prec_{\mathsf{p}}$ is defined as $g_{[i,j]} \prec_{\mathsf{p}} g_{[i',j']}$ if $i+j < i'+j'$ or $i+j = i'+j'$ with $i > j$. The monomial order $\prec_{\mathsf{m}}$ is defined as $x^i y^j \prec_{\mathsf{m}} x^{i'} y^{j'}$ if $i+j < i'+j'$ or $i+j = i'+j'$ with $i > j$. Assuming given $\bar{f}(x,y) = c_{20}x^2 + c_{11}xy + c_{10}x + c_{01}y + 1$,

**Table A.2**
A toy example of lattice basis matrix for $k=1$, $c=1$, and $\bar{f}(x,y)=c_{20}x^2+c_{11}xy+c_{10}x+c_{01}y+1$.

| | 1 | $x$ | $y$ | $x^2$ | $xy$ | $y^2$ | $x^3$ | $x^2y$ | $xy^2$ | $x^4$ | $x^3y$ | $x^2y^2$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $g_{[0,0]}$ | $R/W$ | – | – | – | – | | | | | | | |
| $g_{[1,0]}$ | | $R/W$ | | – | – | | – | – | | | | |
| $g_{[0,1]}$ | | | $R/W$ | | – | – | | – | – | | | |
| $g_{[2,0]}$ | | | | $R/W$ | | | – | – | | – | – | |
| $g_{[1,1]}$ | | | | | $R/W$ | | | – | – | | – | – |
| $g_{[0,2]}$ | | | | | | $Y^2R$ | | | | | | |
| $g_{[3,0]}$ | | | | | | | $X^3R$ | | | | | |
| $g_{[2,1]}$ | | | | | | | | $X^2YR$ | | | | |
| $g_{[1,2]}$ | | | | | | | | | $XY^2R$ | | | |
| $g_{[4,0]}$ | | | | | | | | | | $X^4R$ | | |
| $g_{[3,1]}$ | | | | | | | | | | | $X^3YR$ | |
| $g_{[2,2]}$ | | | | | | | | | | | | $X^2Y^2R$ |

**Table A.3**
A toy example of lattice basis matrix for $m=2$, $\tau=0$, and $\bar{f}(x,y)=c_{20}x^2+c_{11}xy+c_{10}x+c_{01}y+1$.

| | 1 | $x$ | $y$ | $x^2$ | $xy$ | $y^2$ | $x^3$ | $x^2y$ | $xy^2$ | $x^4$ | $x^3y$ | $x^2y^2$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $g_{[0,0]}$ | $R/W$ | – | – | – | – | | | | | | | |
| $g_{[1,0]}$ | | $R/W$ | | – | – | | – | – | | | | |
| $g_{[0,1]}$ | | | $R/W$ | | – | – | | – | – | | | |
| $g_{[2,0]}$ | | | | $R/W$ | | | – | – | | – | – | |
| $g_{[1,1]}$ | | | | | $R/W$ | | | – | – | | – | – |
| $g_{[0,2]}$ | | | | | | $Y^2R$ | | | | | | |
| $g_{[3,0]}$ | | | | | | | $X^3R$ | | | | | |
| $g_{[2,1]}$ | | | | | | | | $X^2YR$ | | | | |
| $g_{[1,2]}$ | | | | | | | | | $XY^2R$ | | | |
| $g_{[4,0]}$ | | | | | | | | | | $X^4R$ | | |
| $g_{[3,1]}$ | | | | | | | | | | | $X^3YR$ | |
| $g_{[2,2]}$ | | | | | | | | | | | | $X^2Y^2R$ |

which implies $d=\gamma=\lambda=1$, we set $k=1$ and $c=1$. Table A.2 shows a toy example of lattice basis matrix, where symbols "–" denote the non-zero off-diagonal entries and other off-diagonal entries are 0.

### *A.2. Lattice construction for Theorem 5*

Suppose that the upper bounds $X$ and $Y$ on $x,\ y$ along with the maximal norm $W$ are derived. We use the monomial sets $S$ and $M$ presented in Theorem 5 with chosen $m$ and $\tau$ to generate shift polynomials.

$$S=\{x^iy^j : 0\le i\le \gamma d(m-1)+\lambda(d(m-1)-j)+\tau dm,\ 0\le j\le d(m-1)\},$$

$$M=\{x^iy^j : 0\le i\le \gamma dm+\lambda(dm-j)+\tau dm,\ 0\le j\le dm\}.$$

Before that, we require the constant term of $f(x,y)$, i.e., $c_{00}$ to be 1. Thus, we should define a modular polynomial $\bar{f}(x,y)=c_{00}^{-1}f(x,y) \pmod R$, where $R=WX^{(\gamma+\lambda)d(m-1)+\tau dm}Y^{d(m-1)}$. The shift polynomials $g_{[i,j]}(x,y)$ according to $S$ and $M$ are defined as follows:

$$g_{[i,j]}(x,y):=\begin{cases}\dfrac{x^iy^j\bar{f}\cdot R}{WX^iY^j}, & x^iy^j\in S,\\ x^iy^j\cdot R, & x^iy^j\in M\setminus S.\end{cases}$$

The coefficient vectors of $g_{[i,j]}(xX,yY)$ are used in the construction of lattice basis matrix.

We further specify the polynomial order $\prec_{\mathsf{p}}$ (resp. the monomial order $\prec_{\mathsf{m}}$) related to the row order (resp. the column order) of the lattice basis matrix. The general principle is to place the polynomials $g_{[i,j]}$ (resp. the monomials $x^iy^j$) regarding $S$ before those regarding $M\setminus S$. The polynomial order $\prec_{\mathsf{p}}$ is defined as $g_{[i,j]}\prec_{\mathsf{p}} g_{[i',j']}$ if $i+j<i'+j'$ or $i+j=i'+j'$ with $i>j$. The monomial order $\prec_{\mathsf{m}}$ is defined as $x^iy^j\prec_{\mathsf{m}} x^{i'}y^{j'}$ if $i+j<i'+j'$ or $i+j=i'+j'$ with $i>j$. Assuming given $\bar{f}(x,y)=c_{20}x^2+c_{11}xy+c_{10}x+c_{01}y+1$, which implies $d=\gamma=\lambda=1$, we set $m=2$ and $\tau=0$. Table A.3 shows a toy example of lattice basis matrix, where symbols "–" denote the non-zero off-diagonal entries and other off-diagonal entries are 0.

One may observe that the above toy examples in Table A.2 and Table A.3 are identical. This arises from the fact that the respective monomial sets $S$ and $M$ exhibit surprising sameness when considering $\bar{f}(x,y)=c_{20}x^2+c_{11}xy+c_{10}x+c_{01}y+1$, with parameters set to $k=1,c=1$ and $m=2,\tau=0$, respectively. However, it is important to note that in most cases, the lattice basis matrices differ.

### A.3. Lattice construction for numerical example

From Example 1, we obtain the RSA-polynomial $F(x,y) = 8x^2 - e_0x + e_1 + y(8x + e_2)$ with known parameters $e_0$, $e_1$, and $e_2$. Additionally, we have $X = Y = 2^{\xi} = 2^{69}$ and $W = 2^{\ell+\xi} = 2^{325}$ since $\ell = 256$ and $\xi = 69$. Consequently, we compute $R = WX^2Y = 2^{532}$. Next, we derive $\overline{F}(x,y) = (e_1)^{-1}F(x,y) \pmod R$[2] with its constant term set to 1, as follows:

$$\overline{F}(x,y) = 1 + C_1x + C_2y + C_3x^2 + C_4xy,$$

where $C_1 = -e_0 \cdot (e_1)^{-1} \pmod R$, $C_2 = e_2 \cdot (e_1)^{-1} \pmod R$, and $C_3 = C_4 = 8 \cdot (e_1)^{-1} \pmod R$. The known parameters $e_0$, $e_1$, and $e_2$ in Example 1 are listed below.

$$e_0 = 326592262158320800222089345319435345054158505554978764096639533590537985 2312,$$

$$e_1 = 788720956125907150125758278145216093264957443298317074587390932928027710 23899,$$

$$e_2 = 107324303637634047210507829583176312810137947350445085346203763307935763010959.$$

Therefore, we calculate $C_1$, $C_2$, $C_3$, and $C_4$ as follows:

$$\begin{aligned}
C_1 = &-522134236407960308712709680902984815594910027136139225051939141609271542599927 27\backslash\\
&172745839076079948801638109766163027033059067034882540617794165566829949736919 21\backslash\\
&970046809884532902078932366632308171961285885291284900913711997673228865992,\\
C_2 = &171583040447807565826517335161939107366605510189739584090424278744404318103563 82\backslash\\
&074309912986174826068473516686930346354800255255668377886585436763435819034107 42\backslash\\
&747262094058115066217697357690179150371663616991706864188087250519303678146 9,\\
C_3 = C_4 = &127898740272014753949541626068752169668104412942271055405542762290447839161275 79\backslash\\
&682492768719781954087875211948579322065581738889617898647868508499374294437111 28.
\end{aligned}$$

We follow the lattice construction detailed in Appendix A.1 and explicitly display two monomial sets $S$ and $M$ for $k = c = 1$.

$$S = \{1, x, y, x^2, xy\}, \quad M = \{1, x, y, x^2, xy, y^2, x^3, xy^2, x^2y, x^4, x^3y, x^2y^2\}.$$

Furthermore, we have $M \setminus S = \{y^2, x^3, xy^2, x^2y, x^4, x^3y, x^2y^2\}$. The shift polynomials $g_{[i,j]}(x,y)$ according to $S$ and $M$ are listed as follows:

$$\begin{aligned}
g_{[0,0]}(x,y) &:= \frac{\overline{F} \cdot R}{W},\\
g_{[1,0]}(x,y) &:= \frac{x\overline{F} \cdot R}{WX},\\
g_{[0,1]}(x,y) &:= \frac{y\overline{F} \cdot R}{WY},\\
g_{[2,0]}(x,y) &:= \frac{x^2\overline{F} \cdot R}{WX^2Y},\\
g_{[1,1]}(x,y) &:= \frac{xy\overline{F} \cdot R}{WXY},\\
g_{[0,2]}(x,y) &:= y^2 \cdot R,\\
g_{[3,0]}(x,y) &:= x^3 \cdot R,\\
g_{[2,1]}(x,y) &:= x^2y \cdot R,\\
g_{[1,2]}(x,y) &:= xy^2 \cdot R,\\
g_{[4,0]}(x,y) &:= x^4 \cdot R,\\
g_{[3,1]}(x,y) &:= x^3y \cdot R,\\
g_{[2,2]}(x,y) &:= x^2y^2 \cdot R.
\end{aligned}$$

The coefficient vectors of $g_{[i,j]}(xX, yY)$ in the above order are used to construct the lattice basis matrix. Table A.4 illustrates the lattice basis matrix of the given numerical example, where blank entries denote 0.

[2] If $(e_1)^{-1} \pmod R$ does not exist, we can gradually increase $W$ to make $R$ coprime to $e_1$.

**Table A.4**
The lattice basis matrix of Example 1 for $\overline{F}(x, y) = 1 + C_1 x + C_2 y + C_3 x^2 + C_4 xy$.

| 1 | $x$ | $y$ | $x^2$ | $xy$ | $y^2$ | $x^3$ | $x^2y$ | $xy^2$ | $x^4$ | $x^3y$ | $x^2y^2$ |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $R/W$ | $C_1XR/W$ | $C_2YR/W$ | $C_3X^2R/W$ | $C_4XYR/W$ | | | | | | | |
| | $R/W$ | | $C_1XR/W$ | $C_2YR/W$ | | $C_3X^2R/W$ | $C_4XYR/W$ | | | | |
| | | $R/W$ | | $C_1XR/W$ | $C_2YR/W$ | | $C_3X^2R/W$ | $C_4XYR/W$ | | | |
| | | | $R/W$ | | | $C_1XR/W$ | $C_2YR/W$ | | $C_3X^2R/W$ | $C_4XYR/W$ | |
| | | | | $R/W$ | | | $C_1XR/W$ | $C_2YR/W$ | | $C_3X^2R/W$ | $C_4XYR/W$ |
| | | | | | $Y^2R$ | | | | | | |
| | | | | | | $X^3R$ | | | | | |
| | | | | | | | $X^2YR$ | | | | |
| | | | | | | | | $XY^2R$ | | | |
| | | | | | | | | | $X^4R$ | | |
| | | | | | | | | | | $X^3YR$ | |
| | | | | | | | | | | | $X^2Y^2R$ |

## References


[1] B. Bagherpour, A bivariate polynomial-based cryptographic hard problem and its applications, Des. Codes Cryptogr. 91 (8) (2023) 2723–2735, https://doi.org/10.1007/S10623-023-01229-1.

[2] R.L. Rivest, A. Shamir, L.M. 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] T.P. Pedersen, Non-interactive and information-theoretic secure verifiable secret sharing, in: J. Feigenbaum (Ed.), Advances in Cryptology - CRYPTO '91, 11th Annual International Cryptology Conference, Proceedings, Santa Barbara, California, USA, August 11-15, 1991, in: Lecture Notes in Computer Science, vol. 576, Springer, 1991, pp. 129–140, https://doi.org/10.1007/3-540-46766-1_9.

[4] G.D. Crescenzo, J. Katz, R. Ostrovsky, A.D. Smith, Efficient and non-interactive non-malleable commitment, in: B. Pfitzmann (Ed.), Advances in Cryptology - EUROCRYPT 2001, International Conference on the Theory and Application of Cryptographic Techniques, Proceeding, Innsbruck, Austria, May 6-10, 2001, in: Lecture Notes in Computer Science, vol. 2045, Springer, 2001, pp. 40–59, https://doi.org/10.1007/3-540-44987-6_4.

[5] E. Bresson, D. Catalano, D. Pointcheval, A simple public-key cryptosystem with a double trapdoor decryption mechanism and its applications, in: C. Laih (Ed.), Advances in Cryptology - ASIACRYPT 2003, 9th International Conference on the Theory and Application of Cryptology and Information Security, Proceedings, Taipei, Taiwan, November 30 - December 4, 2003, in: Lecture Notes in Computer Science, vol. 2894, Springer, 2003, pp. 37–54, https://doi.org/10.1007/978-3-540-40061-5_3.

[6] 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, Proceeding, Espoo, Finland, May 31 - June 4, 1998, in: Lecture Notes in Computer Science, vol. 1403, Springer, 1998, pp. 59–71, https://doi.org/10.1007/BFb0054117.

[7] D. Aggarwal, U.M. 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, Proceedings, Cologne, Germany, April 26–30, 2009, in: Lecture Notes in Computer Science, vol. 5479, Springer, 2009, pp. 36–53, https://doi.org/10.1007/978-3-642-01001-9_2.

[8] D.R.L. 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.

[9] D. Coppersmith, Finding a small root of a bivariate integer equation, factoring with high bits known, in: U.M. Maurer (Ed.), Advances in Cryptology - EUROCRYPT '96, International Conference on the Theory and Application of Cryptographic Techniques, Proceeding, Saragossa, Spain, May 12-16, 1996, in: Lecture Notes in Computer Science, vol. 1070, Springer, 1996, pp. 178–189.

[10] 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.

[11] 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, Proceedings, Aarhus, Denmark, May 22-26, 2005, in: Lecture Notes in Computer Science, vol. 3494, Springer, 2005, pp. 251–267.

[12] 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, Proceedings, Shanghai, China, December 3–7, 2006, in: Lecture Notes in Computer Science, vol. 4284, Springer, 2006, pp. 267–282.

[13] A. May, New RSA vulnerabilities using lattice reduction methods, Ph.D. thesis, University of Paderborn, Paderborn, Germany, 2003.

[14] E. Jochemsz, A. May, A polynomial time attack on RSA with private crt-exponents smaller than $N^{0.073}$, in: A. Menezes (Ed.), Advances in Cryptology - CRYPTO 2007, 27th Annual International Cryptology Conference, Proceedings, Santa Barbara, CA, USA, August 19-23, 2007, in: Lecture Notes in Computer Science, vol. 4622, Springer, 2007, pp. 395–411, https://doi.org/10.1007/978-3-540-74143-5_22.

[15] J. Coron, A. Kirichenko, M. Tibouchi, A note on the bivariate Coppersmith theorem, J. Cryptol. 26 (2) (2013) 246–250, https://doi.org/10.1007/s00145-012-9121-x.

[16] 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.

[17] T.H. Cormen, C.E. Leiserson, R.L. Rivest, C. Stein, Introduction to Algorithms, fourth edition, The MIT Press, 2022, https://mitpress.mit.edu/9780262046305/introduction-to-algorithms/.

[18] J. Coron, Finding small roots of bivariate integer polynomial equations revisited, in: C. Cachin, J. Camenisch (Eds.), Advances in Cryptology - EUROCRYPT 2004, International Conference on the Theory and Applications of Cryptographic Techniques, Proceedings, Interlaken, Switzerland, May 2-6, 2004, in: Lecture Notes in Computer Science, vol. 3027, Springer, 2004, pp. 492–505.

[19] J. Coron, Finding small roots of bivariate integer polynomial equations: a direct approach, in: A. Menezes (Ed.), Advances in Cryptology - CRYPTO 2007, 27th Annual International Cryptology Conference, Proceedings, Santa Barbara, CA, USA, August 19-23, 2007, in: Lecture Notes in Computer Science, vol. 4622, Springer, 2007, pp. 379–394.

[20] 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, Proceedings, Santa Barbara, California, USA, August 15-19, 1999, in: Lecture Notes in Computer Science, vol. 1666, Springer, 1999, pp. 326–337.

[21] 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, Proceedings, Melbourne, Australia, December 7-11, 2008, in: Lecture Notes in Computer Science, vol. 5350, Springer, 2008, pp. 406–424.

[22] Y. Lu, R. Zhang, D. Lin, Factoring multi-power RSA modulus $N = p^r q$ with partial known bits, in: C. Boyd, L. Simpson (Eds.), Information Security and Privacy - 18th Australasian Conference, Proceedings, ACISP 2013, Brisbane, Australia, July 1-3, 2013, in: Lecture Notes in Computer Science, vol. 7959, Springer, 2013, pp. 57–71.
[23] Y. Lu, L. Peng, S. Sarkar, Cryptanalysis of an RSA variant with moduli $N = p^r q^l$, J. Math. Cryptol. 11 (2) (2017) 117, https://doi.org/10.1515/jmc-2016-0025.
[24] 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, Proceedings, San Francisco, CA, USA, April 16-20, 2018, in: Lecture Notes in Computer Science, vol. 10808, Springer, 2018, pp. 65–79.
[25] 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.
[26] 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, Proceedings, CANS 2019, Fuzhou, China, October 25-27, 2019, in: Lecture Notes in Computer Science, vol. 11829, Springer, 2019, pp. 525–537.
[27] A.K. Lenstra, H.W. Lenstra, L. Lovász, Factoring polynomials with rational coefficients, Math. Ann. 261 (4) (1982) 515–534, https://doi.org/10.1007/BF01457454.
[28] The Sage Developers, SageMath, the Sage Mathematics Software System (Version 9 (5), https://www.sagemath.org, 2023.