Mathematics

https://www.aimspress.com/journal/Math

AIMS Mathematics, 11(2): 4902–4934.
DOI: 10.3934/math.2026201
Received: 31 October 2025
Revised: 16 January 2026
Accepted: 22 January 2026
Published: 27 February 2026

Research article

Extending LSB-based partial key exposure to RSA with special-structured primes

Priscilla Kyle Payne[1], **Wan Nur Aqlili Ruzai**[1,2,*], **Amir Hamzah Abd Ghafar**[2,3], **Muhammad Asyraf Asbullah**[2,4,*] **and Muhammad Rezal Kamel Ariffin**[2,3]

[1] School of Distance Education, Universiti Sains Malaysia, Penang 11800, Malaysia

[2] Malaysia Cryptology Technology and Management Centre, c/o Universiti Putra Malaysia, 43400 UPM, Serdang, Selangor, Malaysia

[3] Department of Mathematics and Statistics, Faculty of Science, Universiti Putra Malaysia, 43400 UPM Serdang, Selangor, Malaysia

[4] Centre for Foundation Studies in Science of Universiti Putra Malaysia, Universiti Putra Malaysia, 43400 UPM Serdang, Selangor, Malaysia

* **Correspondence:** Email: aqliliruzai@usm.my, ma_asyraf@upm.edu.my.

Abstract: The Rivest–Shamir–Adleman (RSA) cryptosystem remains one of the most widely used public-key mechanisms, with its security depending on the computational difficulty of factoring a large composite modulus N generated from two primes. Previous studies have shown that RSA becomes vulnerable when its prime factors follow special algebraic structures or when partial information about their least significant bits (LSBs) is exposed. Earlier work demonstrated that primes close to perfect powers allow efficient reconstruction of the modulus when several LSBs of both primes are known. In this paper, we extended this line of research by examining three additional near-square prime structures in which the primes are slightly different, either positively or negatively shifted from their base-power forms. For each structure, we obtained analytical bounds that relate the difference to the square-root proximity of the modulus, and we presented polynomial-time algorithms that recover the prime factors when only a small number of their LSBs are leaked. Numerical experiments confirmed the practicality of the proposed methods. Our results broaden the class of RSA moduli susceptible to LSB-based partial key-exposure attacks and highlight the importance of strengthened key-generation strategies to avoid such structured primes.

Keywords: RSA cryptosystem; RSA cryptanalysis; integer factorization problem; least significant bits; partial key exposure attack
Mathematics Subject Classification: 11T71, 68P25, 94A60

1. Introduction

The Rivest–Shamir–Adleman (RSA) cryptosystem, introduced in 1978 by Rivest et al. [1], remains one of the most widely used public-key mechanisms for securing digital communication. Its security relies on the computational difficulty of factoring a large composite modulus $N = pq$, where p and q are distinct primes, as discussed by Crandall and Pomerance [2]. Although factoring a general RSA modulus is hard with current algorithms, numerous studies have shown that RSA can become vulnerable when its prime factors follow special algebraic structures or when partial information about the secret key is exposed.

Early works by Pollard [3] and Lenstra [4] demonstrated that primes with particular structural properties could be exploited to accelerate factorization. Coppersmith [5] further showed that partial knowledge of prime bits, especially least significant bits (LSBs), can significantly weaken RSA. Building on this line of research, Ghafar et al. [6, 7] proposed attacks on RSA moduli generated from base-power expressions, while Ruzai et al. [8] analyzed multiple sign configurations of near-square primes and demonstrated their susceptibility to structural weaknesses.

Another significant source of RSA vulnerability arises from partial key exposure attacks, where the adversary obtains only a portion of the private exponent d or partial information about the primes. Foundational works such as Rivest and Shamir [9], Coppersmith [10], Hermann and May [11], Heninger and Shacham [12], and Blömer and May [13] developed influential methods for recovering RSA keys from partial information. Subsequent advances extended these attacks to larger key sizes and more general leakage models, including full-size exponent scenarios, as shown by Ernst et al. [14] and Patsakis [15]. More recent theoretical studies by Takayasu and Kunihiro [16], Suzuki et al. [17], and Feng et al. [18] demonstrate that recovering RSA private keys from limited information remains an active research area, although its effectiveness typically decreases when only a small number of bits are leaked.

Motivated by these findings, this paper extends the study of LSB-based weaknesses in RSA by analyzing three additional near-square prime structures that are not addressed in previous work. We show that when the perturbation terms r_x and r_y corresponding to known LSBs of the primes are sufficiently small, the modulus N can be factored in polynomial time across all three sign configurations. Our results broaden the class of RSA moduli that are vulnerable under partial key exposure and underscore the importance of robust, structure-resistant prime generation in RSA implementations.

The novelty of this work arises from several aspects. First, we show that recovery of the RSA prime factors is still possible when only a very small number of LSBs are known and fewer than required in previous LSB-based approaches proposed by Ghafar et al. [6, 7] and Ruzai et al. [8]. Second, unlike MSB or Coppersmith-type attacks, our method does not depend on reconstructing polynomial equations or using lattice reduction, and therefore remains purely deterministic. In this work, we consider RSA primes constructed from near-square base-power expressions. Specifically, the primes take the form

$$p = a^m \pm r_x, \quad q = b^m \pm r_y,$$

where $a, b \in \mathbb{Z}^+$, $m \geq 2$ is an even integer, and r_x, r_y are small integers corresponding to the known LSBs of p and q, respectively. Third, we obtain deviation bounds that handle all mixed-sign perturbation structures of the form $a^m r_x$ and $b^m r_y$, completing the study of near-square RSA primes and demonstrating new structural weaknesses not covered in earlier works.

Although near-square prime structures have been previously studied in the plus-plus case $(a^m + r_x)(b^m + r_y)$, the remaining sign patterns have not been fully analyzed. Here, the expression $(a^m + r_x)(b^m + r_y)$ represents an RSA modulus $N = pq$, where

$$p = a^m + r_x \quad \text{and} \quad q = b^m + r_y$$

after primality testing. Each sign configuration produces a distinct deviation behavior around $\sqrt{N}$ and therefore leads to different recovery bounds. In particular, the minus-minus case reduces both base values simultaneously, while the mixed-sign cases shift the modulus asymmetrically, resulting in different dominant terms in the deviation expression. By analyzing all three unexplored configurations, our work completes the study of near-square RSA primes constructed from small perturbations and reveals that all such structures, regardless of sign, remain vulnerable under partial LSB leakage.

2. Preliminaries

In this section, we will revisit some key findings that will be useful for our proposed attack. These insights come from Ghafar et al. [6,7]. Ghafar et al. [7] demonstrated that a prime number of the form

$$p = (a^m + r_x),$$

where $m \geq 2$ is even and r is a small value, can be closely approximated by $a^{\frac{m}{2}}$ with only a small error term. This leads us to the following result.

Lemma 1. *Let $a, r \in \mathbb{Z}^+$ and $m \geq 2$ be an even number. If*

$$\sqrt{a^m + r} = a^{m/2} + \epsilon,$$

then

$$\epsilon < \frac{r}{2a^{m/2}}.$$

Proof. Refer to Lemma 1 of Ghafar et al. [7]. □

Suppose that the RSA modulus is constructed from primes generated from expressions of the form

$$p = (a^m \pm r_x) \quad \text{and} \quad q = (b^m \pm r_y).$$

Ghafar et al. [7] established upper and lower bounds for the difference between $\sqrt{N}$ and $(ab)^{\frac{m}{2}}$.

Lemma 2. *Let $a, b \in \mathbb{Z}^+$ and $m \geq 2$ be an even number such that*

$$a < b < (2a^m + 1)^{\frac{1}{m}}.$$

Suppose

$$N = (a^m + r_x)(b^m + r_y),$$

where

$$r_x \leq r_y < N^{\gamma}.$$

If

$$r_x < (2a)^{m/2} \quad \textit{and} \quad r_y < (2b)^{m/2},$$

then

$$(r_x r_y)^{1/2} < N^{1/2} - (ab)^{m/2} < \frac{r_y}{2} + 2^{\frac{m}{2}-1} r_x + 1.$$

Proof. Refer to Lemma 2 of Ghafar et al. [7]. □

This inequality provides a bounded search interval for $(ab)^{\frac{m}{2}}$, which is essential for the recovery of the prime factors. To formalize the conditions under which side-channel or partial-information attacks can be exploited, Ghafar et al. [7] introduced precise definitions for the LSBs of primes and for what constitutes a "sufficiently small" perturbation.

Definition 1. *(Ghafar et al. [7]) Let k be a positive integer. We say that the k LSBs of a prime p are available when there exists a small integer r_x, such that*

$$p \equiv r_x \pmod{2^k}, \quad 0 < r_x < 2^k.$$

Definition 2. *(Ghafar et al. [7]) A value is said to be sufficiently small if it is smaller than the largest feasible value that can be searched exhaustively using current computational capabilities. Following the NIST recommendation for the 112-bit security level, any value below* 2^{112} *can be considered sufficiently small.*

Theorem 1. *(Ghafar et al. [7]) Let $a, b \in \mathbb{Z}^+$ and $m \geq 2$ be an even number such that*

$$a < b < (2a^m + 1)^{\frac{1}{m}}.$$

Suppose

$$N = pq = (a^m + r_x)(b^m + r_y)$$

is a valid RSA modulus. Let

$$r_x \equiv p \pmod{2^m}, \quad r_y \equiv q \pmod{2^m},$$

where

$$r_x < 2a^{\frac{m}{2}} \quad \textit{and} \quad r_y < 2b^{\frac{m}{2}}$$

such that

$$\max\{r_x, r_y\} < 2^k.$$

If

$$2^{k-1}\left(2^{\frac{m}{2}} + 1\right)$$

is a sufficiently small value as defined in Definition 2, and k many LSBs of p and q are known, then N can be factored in polynomial time.

Proof. Refer to Theorem 1 of Ghafar et al. [7]. □

The preliminaries above establish the necessary conditions for constructing and analyzing our LSB-based attacks. We now proceed to describe the proposed attacks in detail.

3. Proposed cases for structured RSA moduli

In this section, we consider RSA moduli whose prime factors are constructed from near-square base-power expressions of the form

$$p = a^m \pm r_x, \quad q = b^m \pm r_y,$$

where the perturbation terms r_x and r_y represent the known LSBs of p and q. These expressions describe the candidate values from which the primes are obtained after primality testing as we do not assume that $a^m \pm r_x$ and $b^m \pm r_y$ are themselves prime. The parameters satisfy the following conditions:

- a, b are positive integers satisfying
$$a < b < 2a,$$
to ensure that p and q remain close.
- $m \geq 2$ is an even integer.
- r_x and r_y are small positive integers representing the known LSBs of p and q. In practice, these values satisfy
$$r_x \ll a^{m/2} \text{ and } r_y \ll b^{m/2}.$$

Throughout this work, the exponent m is treated as a fixed small constant (e.g., $m = 2, 4, 6$) and does not scale with the RSA modulus size N. Under these conditions, the RSA modulus takes the general form

$$N = pq = (a^m \pm r_x)(b^m \pm r_y).$$

Therefore, we consider the following three sign configurations:

- **Case I.** $N = (a^m - r_x)(b^m - r_y)$;
- **Case II.** $N = (a^m + r_x)(b^m - r_y)$;
- **Case III.** $N = (a^m - r_x)(b^m + r_y)$.

Although Cases II and III share a mixed-sign structure, they are not interchangeable. In Case II, the positive perturbation is applied to the smaller base value a^m, whereas in Case III, it is applied to the larger base value b^m. This asymmetry produces different dominant terms in the deviation expression

$$\Delta = \sqrt{N} - (ab)^{m/2}$$

leading to distinct upper and lower bounds for each case. Consequently, the corresponding search intervals and recovery conditions differ, and all three configurations must be analyzed separately to complete the study of near-square RSA primes generated from perturbed base-power expressions. These structures represent small perturbations from the base powers a^m and b^m. For each case, we obtain the bounds of $\sqrt{N} - (ab)^{m/2}$, which determine the search interval required to recover the unknown base values of a^m and b^m. We then show that when r_x and r_y are sufficiently small and correspond to only a few leaked LSBs, then the modulus N can be factored in polynomial time.

3.1. Case I. $N = pq = (a^m - r_x)(b^m - r_y)$

Case I examines RSA moduli in which both prime factors are formed by negative perturbations from their base-power values. Under the general assumptions stated at the beginning of this section, the modulus takes the form

$$N = pq = (a^m - r_x)(b^m - r_y).$$

In this configuration, both perturbation terms r_x and r_y reduce the base values a^m and b^m, respectively. Using Lemma 1, we obtain bounds on the deviation

$$(ab)^{m/2} - \sqrt{N},$$

which provide a narrow and efficiently searchable interval for recovering the unknown base values a^m and b^m. We then prove that when the leaked LSBs r_x and r_y are sufficiently small, the modulus N can be factored in polynomial time.

Lemma 3. *Suppose $a, b \in \mathbb{Z}^+$ and $m \geq 2$ is an even integer with*

$$a < b < 2a.$$

Assume further that

$$r_x < (2a)^{\frac{m}{2}} \quad \text{and} \quad r_y < (2b)^{\frac{m}{2}}.$$

If we set $N = pq$, then

$$\sqrt{r_x r_y} \; < \; (ab)^{m/2} - \sqrt{N} \; < \; \frac{r_y}{2} + 2^{\frac{m}{2}-1} r_x + 1.$$

Remark 1. *We emphasize that the derivation of this bound does not rely on estimating or approximating the value of ab in advance. Although one could obtain a simplified inequality by assuming an approximation of the form*

$$ab \approx \sqrt{N}$$

or by treating ab as known, such an assumption is incompatible with our attack model, since the purpose of the bound is to recover the hidden quantities a^m and b^m without any prior knowledge of ab. For this reason, the proof proceeds without using any heuristic approximation of ab, ensuring full generality and avoiding circular reasoning.

Proof. Expanding the modulus, we obtain

$$N = (a^m - r_x)(b^m - r_y) = (ab)^m - a^m r_y - b^m r_x + r_x r_y.$$

Define the deviation

$$X = (ab)^{m/2} - \sqrt{N}.$$

Since

$$N < (ab)^m,$$

we have $X > 0$. From

$$\sqrt{N} = (ab)^{m/2} - X,$$

it follows that

$$N = \left((ab)^{m/2} - X\right)^2 = (ab)^m - 2(ab)^{m/2}X + X^2.$$

Equating the two expressions for N and canceling $(ab)^m$, we get

$$2(ab)^{m/2}X - X^2 = a^m r_y + b^m r_x - r_x r_y. \tag{1}$$

Lower bound. From (1),

$$2(ab)^{m/2}X > a^m r_y + b^m r_x - r_x r_y.$$

By the arithmetic mean-geometric mean inequality,

$$a^m r_y + b^m r_x \geq 2\sqrt{(a^m r_y)(b^m r_x)} = 2(ab)^{m/2}\sqrt{r_x r_y}.$$

Hence,

$$2(ab)^{m/2}X > 2(ab)^{m/2}\sqrt{r_x r_y} - r_x r_y.$$

Dividing both sides by $2(ab)^{m/2}$ yields

$$X > \sqrt{r_x r_y} - \frac{r_x r_y}{2(ab)^{m/2}}.$$

Since

$$r_x r_y < 2(ab)^{m/2},$$

the correction term is less than 1, and thus

$$X > \sqrt{r_x r_y}.$$

Upper bound. By Lemma 1, there exist

$$0 < \varepsilon_1 < \frac{r_x}{2a^{m/2}}, \quad 0 < \varepsilon_2 < \frac{r_y}{2b^{m/2}},$$

such that

$$\sqrt{a^m - r_x} = a^{m/2} - \varepsilon_1$$

and

$$\sqrt{b^m - r_y} = b^{m/2} - \varepsilon_2.$$

Therefore,

$$\sqrt{N} = (a^{m/2} - \varepsilon_1)(b^{m/2} - \varepsilon_2) = (ab)^{m/2} - a^{m/2}\varepsilon_2 - b^{m/2}\varepsilon_1 + \varepsilon_1\varepsilon_2.$$

Hence,

$$X = a^{m/2}\varepsilon_2 + b^{m/2}\varepsilon_1 - \varepsilon_1\varepsilon_2 < a^{m/2}\varepsilon_2 + b^{m/2}\varepsilon_1.$$

Substituting the bounds on ε_1 and ε_2, we obtain

$$X < \frac{r_y}{2} + \frac{r_x}{2}\left(\frac{b}{a}\right)^{m/2}.$$

Since $a < b < 2a$, it follows that

$$\left(\frac{b}{a}\right)^{m/2} < 2^{m/2},$$

and thus

$$X < \frac{r_y}{2} + 2^{\frac{m}{2}-1} r_x.$$

Adding a unit slack to account for integrality gives

$$X < \frac{r_y}{2} + 2^{\frac{m}{2}-1} r_x + 1.$$

Combining both bounds completes the proof. □

Theorem 2. *Let*

$$N = pq = (a^m - r_x)(b^m - r_y),$$

where $m \geq 2$ is even,

$$a < b < 2a,$$

and

$$0 < r_x < 2a^{m/2}, \quad 0 < r_y < 2b^{m/2}.$$

Assume m, r_x, r_y are known and the k LSBs of p and q with

$$k \leq l_1 m, \quad k \leq l_2 m$$

are as in Definition 1. Then the prime factors p and q can be recovered to logN, assuming that the exponent m is a fixed constant.

Proof. From Lemma 3, we can see that

$$\sqrt{r_x r_y} < (ab)^{m/2} - \sqrt{N} < \frac{r_y}{2} + 2^{\frac{m}{2}-1} r_x + 1.$$

Let

$$X = (ab)^{m/2} - \sqrt{N} > 0.$$

Then by the bounds,

$$l_{\min} \leq [X] \leq l_{\max}.$$

Take $i = \lfloor X \rfloor$. Since

$$\lfloor \sqrt{N} \rfloor + i = (ab)^{m/2},$$

we get

$$\sigma = \left(\lfloor \sqrt{N} \rfloor + i\right)^2 = (ab)^m.$$

Congruence at the correct σ. Expanding N,

$$N = pq = (a^m - r_x)(b^m - r_y) = (ab)^m - a^m r_y - b^m r_x + r_x r_y.$$

Thus,

$$N - r_x r_y = (ab)^m - (a^m r_y + b^m r_x).$$

Reducing modulo $\sigma = (ab)^m$ gives

$$z \equiv (N - r_x r_y) \pmod{\sigma} = -(a^m r_y + b^m r_x).$$

Quadratic relation. We consider the quadratic equation

$$x^2 - zx + \sigma r_x r_y = 0.$$

Let x_1 and x_2 denote its two roots. Since

$$\sigma r_x r_y = (ab)^m r_x r_y = (a^m r_y)(b^m r_x),$$

the roots are

$$x_1 = a^m r_y \quad \text{and} \quad x_2 = b^m r_x.$$

Recovering a^m, b^m. Since r_x, r_y are known,

$$a^m = \frac{x_1}{r_y}, \qquad b^m = \frac{x_2}{r_x}.$$

Therefore, we can factor N by

$$p = \frac{N}{b^m - r_y}, \qquad q = \frac{N}{a^m - r_x}.$$

This completes the proof. □

Algorithm 1 enumerates candidate values within the obtained deviation interval and tests each possibility for divisibility. Since the exponent m is fixed as a constant, the enumeration interval of size $O\left(r_y + 2^{m/2} r_x\right)$ remains a polynomial in $\log N$. The following Algorithm 1 applies Theorem 2 to factorize $N = pq$ and can be described as follows:

Algorithm 1. Factoring $N = pq = (a^m - r_x)(b^m - r_y)$ using Theorem 2.

```
Input: $N$, $r_x$ (LSB), $r_y$ (LSB), $m$
Output: $p, q$
Begin
  $i \leftarrow \lfloor \sqrt{r_x r_y} \rfloor$
  while $i < \left\lceil \frac{r_y}{2} + 2^{m/2} + 1 \right\rceil$ do
    $\sigma \leftarrow (\lfloor \sqrt{N} \rfloor + i)^2$
    $z \equiv N - r_x r_y \pmod{\sigma}$
    Find roots $x_1, x_2$ of $x^2 - zx + \sigma r_x r_y = 0$
    if $(N/(x_1 - r_y))$ or $(N/(x_2 - r_x))$ is integer then
      Output $p, q$
    else
      $i \leftarrow i + 1$
    end if
  end while
End
```

To demonstrate the attack detailed in Theorem 2 and implemented via Algorithm 1, the following example was executed on a Windows 11 Home Single Language system, utilizing an HP Spectre x360 Convertible 13-aw2xxx laptop equipped with an 11th Gen Intel(R) Core(TM) i7-1165G7 CPU running at 2.80 GHz and 16.0 GB of RAM, using Python 3.10.8.

Example 1. *We use the RSA-2048 modulus in this example. Specifically, we are given*

N = 1828488729197497500922582588323566281420732854534550184036692103352968030011 7
5328902574247707696667032491116606141336536286588556431699865281091226099662916 11
7993174412021208219093816089813520727362344290741160743416411500354328670443155 53
6203228956259481863590550903133886162259573216190502334646098082192995730654188 36
4931860447028026870788992182707289758644335685761166788763860268785215680611146 03
5633991941742614535991699939884335972373013664483933449647042777463045360144644 90
4500948742527017139985561635903816792152807958286855191018907121014433496679535 06
43075858884739703917834237071158686280555626320117 51457,

where

$$r_x = (011011011011)_2 = 1755$$

and

$$r_y = (111110110011)_2 = 4019.$$

Then, we set

$$i = [\sqrt{r_x . r_y}] = 2656.$$

We calculate

$$\sigma = (\lfloor \sqrt{N} \rfloor + i)^2$$

and

$$z \equiv N - r_x r_y \pmod{\sigma},$$

and proceed to solve the equation

$$x^2 - zx + \sigma r_x r_y = 0.$$

It turns out that neither $\frac{x_1}{r_y} - r_x$ *nor* $\frac{x_2}{r_x} - r_y$ *yields an integer, indicating that* x_1 *and* x_2 *cannot be our final solutions. It also means* $\sigma \neq (ab)^m$ *at this point. To obtain the correct* σ*, we repeatedly compute*

$$\sigma = (\lfloor \sqrt{N} \rfloor + i)^2$$

and

$$z \equiv N - r_x r_y \pmod{\sigma},$$

and let $x_{1,2}$ *be the roots of*

$$x^2 - zx + \sigma r_x r_y = 0,$$

increasing the value of i each iteration. This iterative process remains polynomial in complexity since

$$i < \frac{r_y}{2} + 2^{\frac{m}{2}-1} r_x + 1 = 3766$$

as established in Lemma 3. Here, the valid σ *is obtained when* $i = 2816$. *Specifically, we then compute*

$\sigma = ([\sqrt{N}] + i)^2$
$=$ 18284887291974975009225825883235662814207328545345501840366921033529680300117532890257424770769666703249111660614133653628658855643169986528109122609966291611799317441202120821909381608981352072736234429074116074341641150035432867044315553620322895625948186359055090313388616225957321619050233464609808220060920494179915535505259827988526377044573148085124657241046843499191299130645647181033218397551507773356487211568032764574346626092474551767855333557941329487623003944756618425208315210169643620756032676741630254224879169286715157003815671357542169362712079060675668211062940958426516667819105286738856468545536

and

$z \equiv N - r_x r_y \pmod{\sigma}$
$=$ 761347428761079042319215125185839298145354877356148792807478267382512422744618768659465157282947944374162312950114433594580358192495237250401406940212976625209876699408742153934758220335916941906757476513151248575009598373458029637901924959256098819694758572629917079737092549175002809551950477231176224463847424.

By substituting the obtained values of σ *and z, the following equation is solved*

$$x^2 - zx + \sigma r_x r_y = 0.$$

The results of $x_1,_2$ *are then applied to compute*

$\frac{N}{\frac{x_2}{r_x} - r_y} = p$
$=$ 126135542922571013634187827202023685734131556998907161035438439566060214990194534431514094006346519907160700445410776049585399289275724821610603798032314881642883872462487706215894538773218180974902430626883169939934465957673023399699489417260542681894527532312092633846447225944360505068790918574581692234021

and

$\frac{N}{\frac{x_1}{r_y} - r_x} = q$
$=$ 14496221182636247209311353143071572944722515656839938040230836396952502489973044716764103331706055889873701302564588299219181677999777317577430356650201205633690265516811431711155594875414856164458465996414694351232467059584130253281867231501981641686981436049924093622918033108044410074347450415915899351117.

Hence, N has been successfully factored in polynomial time.

This second example is included to demonstrate the robustness of the proposed method under a different parameter selection within the same sign configuration.

Example 2. *We use the RSA-2048 modulus in this example. Specifically, we are given*

N = 13345170499750808953967616704995487386274044122780371735548747782493539641450383099181339385114994288478950172441587621621246674243799781725080967032199605908

535221459834959949991897399439289995399623095584668657807783734703879574923803053
991052061192309148170118155413351283020228224623673528266478680300260180109950078
882989363380859953255845869153382225691457919265465583781121890489873605493791344
535910452468480500135066785590385319322403434005163275595062694175548929640216423
287030743176212672876897129262213777801591367784459324636892315303977410183794360
8698046442687607281976867327120495960086628703802971597,

where

$$r_x = (11011101110101001)_2 = 113577$$

and

$$r_y = (00111111110000101)_2 = 32645.$$

Then, we set

$$i = [\sqrt{r_x.r_y}] = 60891.$$

We calculate

$$\sigma = (\lfloor \sqrt{N} \rfloor + i)^2$$

and

$$z \equiv N - r_x r_y \pmod{\sigma},$$

and proceed to solve the equation

$$x^2 - zx + \sigma r_x r_y = 0.$$

Here, the valid σ is obtained when i = 90409. *Specifically, we then compute*

$\sigma = ([\sqrt{N}] + i)^2$
= 1334517049975080895396761670499548738627404412278037173554874778249353964145038
309918133938511499428847895017244158762162124667424379978172508096703219960590853
522145983495994999189739943928999539962309558466865780778373470387957492380305399
105206119230914817011815541335128302022822462367352826647868030686564578913983612
408890264607303410186138501449937129206384864115012062650017720986093178152470857
427711046113347078985658788201289194836341666450218528189431211409231029108747088
972746689810667447670138111207135777481458296367399614168692111580103519277872899
20835260115590751930506649043083039296941195264000000

and

$z \equiv N - r_x r_y (mod \sigma)$
= 6605465679189757241099539265213080846015515861117145600605929375684536845378286
719987326287733364038366657992652970654789802291627572625959982659338909686831617
938543380650871047602696723721894001599804251849857579973223215179214671504794605
8118236250089843681222788817427983469953639321922587079210312495168749568.

By substituting the obtained values of σ and z, the following equation is solved

$$x^2 - zx + \sigma r_x r_y = 0.$$

The results of x are then applied to compute

$\frac{N}{\frac{x_2}{r_x} - r_y} = p$
$= 263873844835943384864162029336170322469653836080635100430971346126094811181591811956626151738693936318408258748157091417911930364879692793019008983397957421570588141522227028886188958148297909982553716384304772273644886357562891105726388518407108030494368093938361252747532481892627117817851727501725695886423$

and

$\frac{N}{\frac{x_1}{r_y} - r_x} = q$
$= 505740555986055280665141914335222168952633976353655433340291028663893247248648956201952879137712537155869078419892392813336220797469810686773915890770759469690232490326504550679155008747246657938099544548217238392690376722480357524574605711321114262523232672588684202976701141943957201376676769371584714735739.$

Hence, N has been successfully factored in polynomial time.

The numerical example confirms that the modulus

$$N = pq = (a^m - r_x)(b^m - r_y)$$

can be efficiently factored when the k LSBs of both primes are known. The recovered factors p and q match the constructed primes, validating the correctness of Theorem 2 and Algorithm 1. This shows that RSA moduli generated under the $(a^m - r_x)(b^m - r_y)$ structure are vulnerable to our LSB-based attack.

3.2. Case II. $N = pq = (a^m + r_x)(b^m - r_y)$

Case II considers a mixed-sign perturbation, where the smaller base value a^m is increased by r_x while the larger base value b^m is decreased by r_y. Under the general assumptions of this section, the modulus takes the form

$$N = pq = (a^m + r_x)(b^m - r_y).$$

This asymmetric perturbation produces a deviation pattern different from Case I. We use the same analytical framework to bound

$$\sqrt{N} - (ab)^{m/2}$$

showing that the resulting search interval remains polynomially small.

Lemma 4. *Let a, b, r_x, r_y be positive integers with $m \geq 2$ even and*

$$a < b < (2a^m + r_x)^{1/m}.$$

If

$$r_x \ll 2a^{m/2}$$

and

$$r_y \ll 2b^{m/2},$$

then

$$a^m r_y + 2(ab)^{\frac{m}{2}} (r_x r_y)^{\frac{1}{2}} > b^m r_x.$$

Proof. **Case 1.** $r_y > r_x$. Consider the ratio

$$\frac{a^m r_y + 2(ab)^{\frac{m}{2}}(r_x r_y)^{\frac{1}{2}}}{b^m r_x}.$$

Since $r_y > r_x$, replacing r_y by r_x gives a lower bound for the numerator:

$$\frac{a^m r_y + 2(ab)^{\frac{m}{2}}(r_x r_y)^{\frac{1}{2}}}{b^m r_x} > \frac{a^m r_x + 2(ab)^{\frac{m}{2}}(r_x^2)^{\frac{1}{2}}}{b^m r_x} = \frac{a^m r_x + 2(ab)^{\frac{m}{2}} r_x}{b^m r_x}.$$

Factor $a^m r_x$ in the numerator:

$$\frac{a^m r_x + 2(ab)^{\frac{m}{2}} r_x}{b^m r_x} = \frac{a^m r_x\left(1 + 2\,\frac{(ab)^{\frac{m}{2}}}{a^m}\right)}{b^m r_x} = \frac{a^m}{b^m}\left(1 + 2\left(\frac{b}{a}\right)^{\frac{m}{2}}\right).$$

Because

$$a < b < 2a,$$

we have

$$\frac{a^m}{b^m} < 1 \quad \text{and} \quad \left(\frac{b}{a}\right)^{\frac{m}{2}} > 1;$$

moreover, from

$$b < (2a^m + r_x)^{1/m},$$

we may use $b^m \lesssim 2a^m$, hence

$$\frac{a^m}{b^m}\left(1 + 2\left(\frac{b}{a}\right)^{\frac{m}{2}}\right) \gtrsim \frac{1}{2}\left(1 + 2\left(\frac{b}{a}\right)^{\frac{m}{2}}\right) > \frac{1}{2}(1+2) = \frac{3}{2} > 1.$$

Therefore,

$$a^m r_y + 2(ab)^{\frac{m}{2}}(r_x r_y)^{\frac{1}{2}} > b^m r_x,$$

when $r_y > r_x$.

Case 2. $r_x > r_y$. Write

$$r_x = k\,r_y$$

with $k > 1$. Then

$$\frac{a^m r_y + 2(ab)^{\frac{m}{2}}(r_x r_y)^{\frac{1}{2}}}{b^m r_x} = \frac{a^m r_y + 2(ab)^{\frac{m}{2}} r_y\sqrt{k}}{b^m k r_y} = \frac{a^m}{b^m k}\left(1 + 2\sqrt{k}\left(\frac{b}{a}\right)^{\frac{m}{2}}\right).$$

Once again using $b^m \lesssim 2a^m$ gives

$$\frac{a^m}{b^m k}\left(1 + 2\sqrt{k}\left(\frac{b}{a}\right)^{\frac{m}{2}}\right) \gtrsim \frac{1}{2k}\left(1 + 2\sqrt{k}\right) = \frac{1}{2}\left(\frac{1}{k} + \frac{2}{\sqrt{k}}\right) > 1 \quad \text{(for } k \text{ not too large).}$$

In particular, if

$$k \leq \tfrac{3}{2},$$

this lower bound is

$$\geq \tfrac{3}{2k} > 1,$$

hence the desired inequality holds (stronger bounds follow by tightening $b^m \lesssim 2a^m$ according to the actual b/a).

Combining both cases, we obtain

$$a^m r_y \ + \ 2(ab)^{\frac{m}{2}}(r_x r_y)^{\frac{1}{2}} \ > \ b^m r_x.$$

This completes the proof. □

Lemma 5. *Let*

$$N = (a^m + r_x)(b^m - r_y),$$

where a, b, r_x, r_y are positive integers and $m \geq 2$ is even, satisfying

$$a < b < (2a^m + r_x)^{1/m}.$$

If

$$r_x \ll 2a^{m/2}$$

and

$$r_y \ll 2b^{m/2},$$

then

$$\frac{r_x - r_y}{2} - 1 \ < \ N^{1/2} - (ab)^{m/2} \ < \ (r_x r_y)^{1/2}.$$

Proof. Expand N:

$$\begin{aligned} N &= (a^m + r_x)(b^m - r_y) \\ &= (ab)^m - a^m r_y + b^m r_x - r_x r_y. \end{aligned}$$

Hence

$$N^{1/2} = \sqrt{(ab)^m - a^m r_y + b^m r_x - r_x r_y}.$$

Upper bound. From Lemma 4, we have

$$a^m r_y - b^m r_x \ > \ -2(ab)^{\frac{m}{2}}(r_x r_y)^{\frac{1}{2}}.$$

Therefore

$$\begin{aligned} N^{1/2} &= \sqrt{(ab)^m - (a^m r_y - b^m r_x) - r_x r_y} \\ &< \sqrt{(ab)^m + 2(ab)^{\frac{m}{2}}(r_x r_y)^{\frac{1}{2}}} \\ &\leq (ab)^{\frac{m}{2}} + (r_x r_y)^{\frac{1}{2}}, \end{aligned}$$

which gives

$$N^{1/2} - (ab)^{\frac{m}{2}} < (r_x r_y)^{\frac{1}{2}}.$$

Lower bound. Using first-order square-root expansions (Lemma 1-type bounds),

$$\sqrt{a^m + r_x} = a^{\frac{m}{2}} + \varepsilon_1, \quad 0 < \varepsilon_1 < \frac{r_x}{2a^{\frac{m}{2}}},$$
$$\sqrt{b^m - r_y} = b^{\frac{m}{2}} - \varepsilon_2, \quad 0 < \varepsilon_2 < \frac{r_y}{2b^{\frac{m}{2}}}.$$

Thus,

$$\begin{aligned} N^{1/2} &= \sqrt{(a^m + r_x)(b^m - r_y)} = (a^{\frac{m}{2}} + \varepsilon_1)(b^{\frac{m}{2}} - \varepsilon_2) \\ &= (ab)^{\frac{m}{2}} + a^{\frac{m}{2}}\varepsilon_2 - b^{\frac{m}{2}}\varepsilon_1 - \varepsilon_1\varepsilon_2 \\ &> (ab)^{\frac{m}{2}} + \frac{r_y}{2} - \frac{r_x}{2} - \frac{r_x r_y}{4(ab)^{\frac{m}{2}}}. \end{aligned}$$

Since $\frac{r_x r_y}{4(ab)^{\frac{m}{2}}}$ is small (under $r_x \ll 2a^{m/2}$ and $r_y \ll 2b^{m/2}$), we may bound it by < 1, yielding

$$N^{1/2} > (ab)^{\frac{m}{2}} + \frac{r_x - r_y}{2} - 1, \quad \frac{r_x - r_y}{2} - 1 < N^{1/2} - (ab)^{\frac{m}{2}}.$$

Combining the two bounds completes the proof. □

Theorem 3. *Let*

$$N = (a^m + r_x)(b^m - r_y)$$

be an RSA modulus with $m \geq 2$ even and

$$a < b < (2a^m + r_x)^{1/m}.$$

Suppose

$$r_x \ll 2a^{m/2}, \quad r_y \ll 2b^{m/2},$$

and

$$\max\{r_x r_y\} = N^{\nu},$$

where the leakage exponent satisfies

$$\nu = O\left(\frac{\log\log N}{\log N}\right).$$

Under this condition, the enumeration range is polynomially bounded in $\log N$*, and N can be factored in polynomial time.*

Remark 2. *We note that this condition is compatible with the near-square hypothesis assumed throughout the manuscript. Since m is treated as a fixed even integer and $a < b < 2a$, the quantity $|a^m - b^m|$ remains sufficiently small relative to $N^{1/2}$. Hence, the bound imposed above does not contradict the near-square assumption and is consistently used in Lemma 5, Theorem 3, Lemma 6, and subsequent results.*

Proof. From Lemma 5, we know that

$$\frac{r_x - r_y}{2} - 1 \; < \; N^{1/2} - (ab)^{m/2} \; < \; (r_x r_y)^{1/2}.$$

This can be rewritten as

$$N^{1/2} - (r_x r_y)^{1/2} \; < \; (ab)^{m/2} \; < \; N^{1/2} + \frac{r_y - r_x}{2} + 1.$$

Hence the search width for $(ab)^{m/2}$ is

$$\left(N^{1/2} + \frac{r_y - r_x}{2} + 1\right) - \left(N^{1/2} - (r_x r_y)^{1/2}\right) = \frac{r_y - r_x}{2} + (r_x r_y)^{1/2}.$$

Since

$$\max\{r_x r_y\} = N^{\nu},$$

this width is bounded by

$$\frac{r_y - r_x}{2} + (r_x r_y)^{1/2} \; < \; \frac{N^{\nu}}{2} + N^{\nu} = \frac{3}{2} N^{\nu}.$$

If $\frac{3}{2}N^{\nu}$ is small, the interval is short and $(ab)^{m/2}$ can be found in polynomial time by enumeration over integers in that interval. Once $(ab)^{m/2}$ is found,

$$(ab)^m = ((ab)^{m/2})^2.$$

Next compute

$$N + r_x r_y \equiv b^m r_x - a^m r_y \pmod{(ab)^m}.$$

Because

$$r_x \ll 2a^{m/2}$$

and

$$r_y \ll 2b^{m/2},$$

the quantity $-a^m r_y + b^m r_x$ is smaller than $(ab)^m$, so we can work without modular reduction and recover it exactly. Now consider the quadratic

$$X^2 + (b^m r_x - a^m r_y)X - (ab)^m r_x r_y = 0.$$

Its roots are

$$X_1 = a^m r_y, \qquad X_2 = -\, b^m r_x.$$

Thus

$$a^m = \frac{X_1}{r_y}, \qquad b^m = -\,\frac{X_2}{r_x}.$$

Finally, since

$$N = (a^m + r_x)(b^m - r_y),$$

we obtain the factors as

$$p = a^m + r_x, \quad q = b^m - r_y,$$

so $N = pq$ is factored. This completes the proof. □

The following Algorithm 2 applies Theorem 3 to factorize $N = pq$ is as follows:

Algorithm 2. Factoring $N = pq = (a^m + r_x)(b^m - r_y)$ using Theorem 3.

```
Input: $N$, $r_x$ (LSB), $r_y$ (LSB), $m$
Output: $p, q$
Begin
  $i \leftarrow \lceil \sqrt{r_x r_y} \rfloor$
  while $i < \left\lfloor \sqrt{N} + \frac{r_y - r_x}{2} \right\rfloor$ do
    $\sigma \leftarrow (\lfloor \sqrt{N} \rfloor - i)^2$
    $z \equiv (N + r_x r_y) \pmod{\sigma}$
    Find roots $x_1, x_2$ of $x^2 - zx - \sigma r_x r_y = 0$
    if $\frac{x_1}{r_x}$ and $\frac{-x_2}{r_y}$ are not integers then
      $i \leftarrow i + 1$
    else
      $a^m \leftarrow \frac{x_1}{r_y}$ and $b^m \leftarrow -\frac{x_2}{r_x}$
      break
    end if
  end while
Output $p$, $q$
End
```

The enumeration range in Algorithm 2 follows directly from the deviation bound obtained in Theorem 3, ensuring that the search remains polynomial under the stated parameter restrictions. To demonstrate the attack detailed in Theorem 3 and implemented via Algorithm 2, the following example was executed on a Windows 11 Home Single Language system, utilizing an HP Spectre x360 Convertible 13-aw2xxx laptop equipped with an 11th Gen Intel(R) Core(TM) i7-1165G7 CPU running at 2.80 GHz and 16.0 GB of RAM, using Python 3.10.8.

Example 3. *We use the RSA-2048 modulus in this example. Specifically, we are given:*

N = 6391919485201160052365761581698232022860877075563899411234210524425280234414 4
63703730191029927515719041718275174361571859508318392653288163060660157129728951 8
10735090083624066077882376221801888569526333146384673815237844117151603257214988 3
00409329567338940990914855695830524312211586875923809457736185482608782705294389 7
50243718218608315893874911878694194990553792981220153530346566615072350125551238 5
58318403960018045515357062098317087719073684411368868954534247094573237426996026 5
90986249070731951251089251212736502491388668461777984144433084326839200637723266
08446501239886601641568718155534884847401278687241989,

where

$$r_x = (101000111011)_2 = 2619$$

and

$$r_y = (000001000001)_2 = 65.$$

Then, we set

$$i = [\sqrt{r_x.r_y}] = 413.$$

We calculate

$$\sigma = (\lfloor \sqrt{N} \rfloor - i)^2$$

and

$$z \equiv N - r_x r_y \pmod{\sigma},$$

and solve the equation

$$x^2 - zx - \sigma r_x r_y = 0.$$

In this case, we find the correct σ when $i = 1476$. That is, we compute

$\sigma = ([\sqrt{N}] - i)^2$
$= 63919194852011600523657615816982320228608770755638994112342105244252802344144\!63$
703730191029927515719041718275174361571859508318392653288163060660157129728951810
735090083624066077882376221801888569526333146384673815237844117151603257214988300
409329567338940990914855695830524312211586875923809457736185482372733744041576892
808682247104843260214067108242853409673347010414664455292874279348325592651257251
192623120142762800586027026767662179886104044085872184954445412780328817427106660
019688078642923733310233826260818965785791305805344543889645918328918674493925764
510926556482784487549995490836243059136528384000000

and

$z \equiv N - r_x r_y \pmod{\sigma}$
$= 236048961252812857435035971503472633660844770451341580880445970805489075053692\!3$
357240245328999813071257808398752827147710350715494255391875803672829967695798016
817929086095689199309665609920890275177790173864756835256028771559726396005436625
14355001389278400843935574683403817154018722664698641788264750303412224.

Using values of σ and z, we solve the equation

$$x^2 - zx + \sigma r_x r_y = 0.$$

The solutions of $x_{1,2}$ are used to compute

$\frac{N}{\frac{x_2}{r_x}+r_y} = p$
$= 695859607224506842930793797847595311125564754454609527738780110060180901513433\!7$
055601198060843610365250349030394446171679684535686388970730926416996991122811081
846562102872068474245801323647772688630181052044835990740467263096603815263358885
1598257200284935045849130872779647966196950857312260628168704002619

and

$\frac{N}{\frac{x_1}{r_y}+r_x} = q$
$= 918564523481375150492883280601305854078506839844583974076930284539443493369719\!9$

49637897333484267483250070892497285623454092655584657090563403854349159273805379476873516519418726381162015473310728390181882666303271934265677269894738757024993469678839661309360908422177129188599739677451204367845471919496 23231.

Hence, N has been successfully factored in polynomial time.

We include this additional example to illustrate that the attack remains effective for distinct parameter choices in Case II and is not limited to a single numerical instance.

Example 4. *We use the RSA-512 modulus in this example. Specifically, we are given*

N = 6081088001984419610810210226324080001844343547965683846688788044166898702485086017881730186855070210304879460004004870307560395624287837214771347620805051070537509884559682013570450696646938416609428689960355898651649855870028910998449569089971638035822827294491252752206590565045605787374775195872584269654409450947 9,

where

$$r_x = (101101001101111)_2 = 23151$$

and

$$r_y = (000111110110111)_2 = 4023.$$

Then, we set

$$i = [\sqrt{r_x.r_y}] = 9651.$$

We calculate

$$\sigma = ([\sqrt{N}] - i)^2$$

and

$$z \equiv N - r_x r_y \pmod{\sigma},$$

and solve the equation

$$x^2 - zx - \sigma r_x r_y = 0.$$

In this case, we find the correct σ when $i = 10909$. That is, we compute

$\sigma = ([\sqrt{N}] - i)^2$
= 608108800198441961081021022632408000184434354796568384668878804416689870248508601788173018685507021030487946000400487030756039562428783721477134762080505053247370259523421840492451267862882488725097709692619165041935956976508787761142803151033666616664561226199334263893187377625615423680451462557202265769875865 60000

and

$z \equiv N - r_x r_y \pmod{\sigma}$
= 538063807289325463608645938018018113529358451593034164248232290286104941033387021537579634971869177215032497910113274716788789451550570260570300561611955660108595 2.

Using values of σ and z, we solve the equation

$$x^2 - zx + \sigma r_x r_y = 0.$$

The solutions of $x_{1,2}$ are used to compute

$\frac{N}{\frac{x_2}{r_x}+r_y} = p$
$=$ 2240990774411054300235608723241587521241444537244864525977557797174539922300153
16404190953815117704005435662983216172894110992518305466414628284749608926665327

and

$\frac{N}{\frac{x_1}{r_y}+r_x} = q$
$=$ 2713571189770991223304066679426492071655227702157120919736985186240061691104151
8004707600349517991859376237101130022075594559273537422115639724397061881855597 7.

Hence, N has been successfully factored in polynomial time.

The experiment shows that for

$$N = pq = (a^m + r_x)(b^m - r_y),$$

the attack succeeds in polynomial time once the LSBs are revealed. The algorithm correctly reconstructed both primes, consistent with the bounds we get in Theorem 3. This confirms that the $(a^m + r_x)(b^m - r_y)$ case, which mixes positive and negative offsets, is equally susceptible to LSB leakage and can be broken under the proposed method.

3.3. Case III. $N = pq = (a^m - r_x)(b^m + r_y)$

Case III mirrors Case II but with the positive perturbation applied to the larger base value b^m and the negative perturbation applied to a^m. Although the sign pattern appears symmetrical to Case II, the resulting deviation expression is not identical, and the analysis must be carried out separately. Under the general assumptions above, the modulus is

$$N = pq = (a^m - r_x)(b^m + r_y).$$

We obtain the corresponding deviation bounds for

$$\sqrt{N} - (ab)^{m/2}$$

demonstrating that the interval remains efficiently searchable whenever only a few LSBs of the primes are known.

Lemma 6. *Let*

$$N = (a^m - r_x)(b^m + r_y),$$

where a, b, r_x, r_y are positive integers and $m \geq 2$ is even, with

$$a < b < (2a^m - r_x)^{1/m}.$$

If

$$r_x \ll 2a^{m/2}$$

and

$$r_y \ll 2b^{m/2},$$

then

$$-(1 + \sqrt{2})\,(r_x r_y)^{1/2} \; < \; N^{1/2} - (ab)^{m/2} \; < \; \frac{r_y - r_x}{2}.$$

Proof. Expand the modulus:

$$N = (a^m - r_x)(b^m + r_y) = (ab)^m + a^m r_y - b^m r_x - r_x r_y.$$

Lower bound, using binomial approximation. Use the first-order expansion for square roots,

$$\sqrt{x+y} \;\approx\; \sqrt{x} + \frac{y}{2\sqrt{x}} \quad (\text{for } |y| \ll x).$$

Let $x = (ab)^m$ and

$$y = a^m r_y - b^m r_x - r_x r_y.$$

Then

$$N^{1/2} \approx (ab)^{m/2} + \frac{a^m r_y - b^m r_x - r_x r_y}{2(ab)^{m/2}}.$$

Hence

$$N^{1/2} - (ab)^{m/2} \;\gtrsim\; \frac{a^m r_y - b^m r_x}{2(ab)^{m/2}} - \frac{r_x r_y}{2(ab)^{m/2}}.$$

Using

$$a < b < 2a$$

and the arithmetic mean-geometric mean inequality, which states that for any non-negative real numbers x and y,

$$\frac{x+y}{2} \geq \sqrt{xy},$$

we obtain

$$\frac{a^m r_y - b^m r_x}{2(ab)^{m/2}} \geq -\sqrt{r_x r_y}$$

and

$$\frac{r_x r_y}{2(ab)^{m/2}} \leq \sqrt{2}\,\sqrt{r_x r_y}$$

(for sufficiently small r_x, r_y relative to $a^{m/2}, b^{m/2}$), which yields

$$N^{1/2} - (ab)^{m/2} \;>\; -(1+\sqrt{2})\,\sqrt{r_x r_y}.$$

Upper bound using first-order root expansions. By Lemma 1-type root bounds,

$$\sqrt{a^m - r_x} = a^{m/2} - \varepsilon_1, \qquad 0 < \varepsilon_1 < \frac{r_x}{2a^{m/2}},$$
$$\sqrt{b^m + r_y} = b^{m/2} + \varepsilon_2, \qquad 0 < \varepsilon_2 < \frac{r_y}{2b^{m/2}}.$$

Then

$$\begin{aligned}\sqrt{N} &= \sqrt{(a^m - r_x)(b^m + r_y)} = (a^{m/2} - \varepsilon_1)(b^{m/2} + \varepsilon_2) \\ &= (ab)^{m/2} + a^{m/2}\varepsilon_2 - b^{m/2}\varepsilon_1 - \varepsilon_1\varepsilon_2.\end{aligned}$$

Thus,

$$N^{1/2} - (ab)^{m/2} = a^{m/2}\varepsilon_2 - b^{m/2}\varepsilon_1 - \varepsilon_1\varepsilon_2 \;<\; a^{m/2}\varepsilon_2 - b^{m/2}\varepsilon_1,$$

and with the bounds on $\varepsilon_1, \varepsilon_2$,

$$N^{1/2} - (ab)^{m/2} < \frac{r_y}{2} - \frac{r_x}{2} - \frac{r_x r_y}{4(ab)^{m/2}} \le \frac{r_y - r_x}{2}.$$

Combining the two sides gives

$$-(1+\sqrt{2})\,(r_x r_y)^{1/2} < N^{1/2} - (ab)^{m/2} < \frac{r_y - r_x}{2}$$

as claimed. $\square$

Theorem 4. *Let a, b, r_x, r_y be positive integers and let $m \ge 2$ be a power of 2. Suppose*

$$a < b < (2a^m - r_x)^{1/m},$$

and define

$$N = (a^m - r_x)(b^m + r_y)$$

as an RSA modulus. Further assume that

$$r_x \ll 2a^{m/2}, \quad r_y \ll 2b^{m/2},$$

and define

$$\max\{r_x r_y\} = N^{\gamma},$$

where

$$\gamma = O\left(\frac{\log\log N}{\log N}\right).$$

Then the search interval remains polynomial in $\log N$*, and N can be factored in polynomial time.*

Proof. From Lemma 6, we have

$$-(1+\sqrt{2})\,(r_x r_y)^{1/2} < N^{1/2} - (ab)^{m/2} < \frac{r_y - r_x}{2}.$$

Since $(1+\sqrt{2})\,N^{\gamma}$ is sufficiently small, we can locate $(ab)^{m/2}$ within a short range:

$$N^{1/2} - (1+\sqrt{2})\,(r_x r_y)^{1/2} < (ab)^{m/2} < N^{1/2} + \frac{r_y - r_x}{2}.$$

Hence the search range for $(ab)^{m/2}$ is from

$$N^{1/2} - (1+\sqrt{2})\,(r_x r_y)^{1/2}$$

to

$$N^{1/2} + \frac{r_y - r_x}{2}.$$

Because $(1+\sqrt{2})\,N^{\gamma}$ is small, $(ab)^{m/2}$ can be found in polynomial time. Once $(ab)^{m/2}$ is known, compute $(ab)^m$ by squaring $(ab)^{m/2}$. Then p and q can be recovered from the solutions of

$$X^2 + (a^m r_y - b^m r_x)X - (ab)^m r_x r_y = 0.$$

The roots are

$$x_1 = a^m r_y, \qquad x_2 = -b^m r_x.$$

Since r_x, r_y are known,

$$a^m = \frac{x_1}{r_y}, \qquad b^m = -\frac{x_2}{r_x}.$$

Thus, the modulus N factors as

$$p = \frac{N}{b^m + r_y}, \qquad q = \frac{N}{a^m - r_x}.$$

□

The following Algorithm 3 applies Theorem 4 to factorize $N = pq$ as follows:

Algorithm 3. Factoring $N = pq = (a^m - r_x)(b^m + r_y)$ using Theorem 4.

```
Input: $N$, $r_x$ (LSB), $r_y$ (LSB), $m$
Output: $p, q$
Begin
  $i \leftarrow \lceil \sqrt{r_x r_y} \rfloor$
  while $i < \left\lfloor \sqrt{N} + \frac{r_y - r_x}{2} \right\rfloor$ do
    $\sigma \leftarrow (\lfloor \sqrt{N} \rfloor + i)^2$
    $z \equiv (N + r_x r_y) \pmod{\sigma}$
    Find roots $x_1, x_2$ of $x^2 - zx + \sigma r_x r_y = 0$
    if $\frac{x_1}{r_x}$ and $\frac{-x_2}{r_y}$ are not integers then
      $i \leftarrow i + 1$
    else
      $a^m \leftarrow \frac{x_1}{r_y}$ and $b^m \leftarrow -\frac{x_2}{r_x}$
      break
    end if
  end while
Output $p, q$
End
```

To demonstrate the attack detailed in Theorem 4 and implemented via Algorithm 3, the following example was executed on a Windows 11 Home Single Language system, utilizing an HP Spectre x360 Convertible 13-aw2xxx laptop equipped with an 11th Gen Intel(R) Core(TM) i7-1165G7 CPU running at 2.80 GHz and 16.0 GB of RAM, using Python 3.10.8.

Example 5. *We use the RSA-2048 modulus in this example. Specifically, we are given*

N = 157070468563539046335241567259777592954572856919138150831741192649026056844742511102869700193260596766855402301671028022078556341807178225775059188601894357 96

064544515227128189698876073075646333334738263848584110585936433432366935548826520
051413180054841154228663906415807572896910765221596311393624918446867621998126242
318889049330248714547005020880871962735413214556580752445379043314206975623250107
225928765744647140160102611398631995183588037890262919799361723489737885996390496
913402581576881886332319721759694033341077649935126033389791992036348371569204706
477153772655498778895479988848723771370363820067434181,

where

$$r_x = (110001101011)_2 = 3179$$

and

$$r_y = (001001110001)_2 = 625.$$

Then, we set

$$i = [\sqrt{r_x.r_y}] = 1410.$$

Then we calculate

$$\sigma = ([\sqrt{N}] + i)^2 \quad \textit{and} \quad z \equiv N + r_x r_y \pmod{\sigma},$$

and solve the equation

$$x^2 - zx + \sigma r_x r_y = 0.$$

In this case, we find the correct σ *when* $i = 731$. *That is, we compute*

$\sigma = ([\sqrt{N}] + i)^2$
$= 1570704685635390463352415672597775929545728569191381508317411926490260568447425$
$111028697001932605967668554023016710280220785563418071782257750591886018943579606$
$454451522712818969887607307564633333473826384858411058593643343236693554882652005$
$141318005484115422866390641580757289691076522159631139362491844730132709188042231$
$101515833812921800668318743655246780046525884762651216006476473557964287156789757$
$228069063394829394153583277729719999749094005651435867043004877356284366521661325$
$363378855664811287094138228174320448571397005314959642417631361534042463281365954$
$1310017969419128534337418329613451558841787351040000$

and

$z \equiv N + r_x r_y (mod \sigma)$
$= 4337050937541799921261090078805034596781665556805050650520442910457597146857214$
$213726672483177903463519248893011537814332213786652048139029021662514388710683252$
$838249576688261163402312069797662265386216605220491711446363201180235630343843215$
$78992053063608953064156245313920349638857429480889680188477967281618944.$

Using values of σ *and* z, *we solve the equation*

$$x^2 - zx + \sigma r_x r_y = 0.$$

The solutions of $x_1,_2$ *are used to compute*

$\frac{N}{\frac{x2}{rx} - r_y} = p$
$= 1005584299872817200722658201027323827015858364695782086949001845213453781365725$

424272576984829659986280532468989717814038897516466855543112718002657133988135698
376769122767483302110794043211205231851179438684289734991235371861191252288087064
47339813384421672268589868331462232600996249649659355091062128636821

and

$\frac{N}{\frac{x_1}{r_y}-r_x} = q$
$=$ 156198210914196623834940278529320918171329884703992276025906544942309083366177810754861060040781407830908514820072709912127232917390149527350437690314484217303061457385291536276414644804869602564809474795671537363819310641981550178138530910068757293623426391391011296955358365842868853574056866033467064386161.

Hence, N has been successfully factored in polynomial time.

This example complements the previous one by confirming that the proposed approach consistently succeeds for alternative parameter values in Case III.

Example 6. *We use the RSA-2048 modulus in this example. Specifically, we are given*

N = 29980928041003273423498491368331737853338541634961484376859949428914221541181940020898510002681204614120311203507519124924409977236014498543291942147703963823107447801593358800473031721521071677236175883757135687112653851938318614399688776193357639158309019718841033032659896464879195721839753251112509052537409060110359275949802076007980092858639334477424306283413693322733237303818895743433159740659450298181096122238578849920613071370000823844246529513932400352877067979905499243812886434862749519273170347576673565143518708025543568295288851878625822848975515439132451454952202622411892924162367342382281908612909,

where

$$r_x = (00001111000011)_2 = 963$$

and

$$r_y = (11110110110001)_2 = 15793.$$

Then, we set

$$i = [\sqrt{r_x.r_y}] = 7260.$$

Then we calculate

$$\sigma = (\lfloor \sqrt{N} \rfloor + i)^2$$

and

$$z \equiv N + r_x r_y \pmod{\sigma}.$$

Next, we solve the equation

$$x^2 - zx + \sigma r_x r_y = 0.$$

In this case, we find the correct σ when i = 731. *That is, we compute*

$\sigma = ([\sqrt{N}] + i)^2$
$=$ 2998092804100327342349849136833173785333854163496148437685994942891422154118194002089851000268120461412031120350751912492440997723601449854329194214770396382310

7447801593358800473031721521071677236175883757135687112653851938318614399688776193357639158309019718841033032659896464879195721839753251112509050023010902547103863332936451698922283838883170131780254891649021420575893887627992153358416552512600676611417724888616814366101924701114755380755148629960187901865568664881026008399574537906852594143594181225916142710890328489792803954797018910521959517248357067777369011474849064761934309562314709452541734682624

and

$z \equiv N + r_x r_y \pmod{\sigma}$
$= 2514398157563255412616865624309057809019756164345644051391764671902157343416190903590074743188146849621569678397349962035554511146668886068463491380883972212451011499315024473235413311896955896925129576166350757422432628379535750764340491832968103863331727158371355082443477353557649958614600052632929740189138944.$

Using values of σ and z, we solve the equation

$$x^2 - zx + \sigma r_x r_y = 0.$$

The solutions of $x_{1,2}$ are used to compute

$\frac{N}{\frac{x_2}{r_x} - r_y} = p$
$= 169965537435254768209783231685184942827815280787657902767614594230916446778784074782190927930064047724031639661887445378696269897647756286419105083637854320544264215201309955859904190105909172588401152286383978128477260091182312484053979119714724075662498340207726483628206031158894795842757814744854781230141$

and

$\frac{N}{\frac{x_1}{r_y} - r_x} = q$
$= 176394159036057260353313555238907573270955934718418698875568655022540083656434049267981912369007950243055043388201934403110778137987672858726308624100374944864541278670055721350003699320527171093966585558163457072283219356704579746961578821065974520878617994319075050360104357772352598265912892661019542896049.$

Hence, N has been successfully factored in polynomial time.

The example illustrates that the modulus

$$N = pq = (a^m - r_x)(b^m + r_y)$$

is factored successfully using only a small number of LSBs. The primes are obtained with the constructed values, supporting the theoretical analysis of Theorem 4 and Algorithm 3. This establishes that the modulus

$$N = pq = (a^m - r_x)(b^m + r_y)$$

case also falls within the vulnerable near-square RSA primes and further expand the scope of LSB-based attacks.

4. Numbers of primes with vulnerable special structures

The effectiveness of attacks with LSBs depends not only on the proximity of $(ab)^{\frac{m}{2}}$ to N, but also on how many primes actually satisfy these structural conditions. In the random reconstruction algorithm (RRA), the unknown higher-order bits of the primes must correspond to the perfect squares once the lower k bits are fixed. Specifically, if p and q are n-bit primes and only the lower k LSB is known, then the remaining bits $(n-k)$ must be within the interval $\left[2^{(n-k-1)},\ 2^{(n-k)}-1\right]$.

Ghafar et al. [7] proved that there is exponentially more value for such values at

$$\left\lfloor 2^{\frac{n-k}{2}}\left(1-2^{-\frac{1}{2}}\right)\right\rfloor.$$

Squares exist between $2^{(n-k-1)}$ and $2^{(n-k)}-1$. As mentioned in Ghafar et al. [6], "the number of primes satisfying our conditions are exponentially many" means that the number of primes will remain compatible when the attack conditions are extremely large, even if only a few LSBs are revealed.

It also applies to all the newly proposed cases, which are:

- **Case I.** $N = pq = (a^m - r_x)(b^m - r_y)$;
- **Case II.** $N = pq = (a^m + r_x)(b^m - r_y)$;
- **Case III.** $N = pq = (a^m - r_x)(b^m + r_y)$.

In each of these three cases, the small perturbations r_x and r_y represent the known LSBs, which determine how close the structure lies within the bounded interval around $\sqrt{N}$. Since the primes are generated without imposing any density assumptions on these structures, our results do not require the number of such primes to be large. Rather, the analysis shows that whenever primes do satisfy these perturbation conditions, they become vulnerable to LSB-based factorization. A full density analysis of these structured primes is outside the scope of this work and remains an interesting direction for future research.

5. Comparative analysis

The study of partial-information attacks on RSA has grown over the last four decades. Rivest and Shamir [9] first observed that RSA moduli can be factored if some of the bits are known. Coppersmith [10] later improved this by showing that lattice reduction methods allow factorization with only about half of the bits revealed. Herrmann and May [11] proposed a heuristic lattice-based algorithm for finding small solutions, which solves the factoring-with-known-bits problem when approximately 70% of the bits of p are known at arbitrary positions. In contrast, Heninger and Shacham [12] introduced the RRA, which succeeds when the combined fraction of known bits across both primes exceeds roughly 57%. Although these results advanced the theory of partial key exposure, they still require a significant proportion of prime bits to be known, which limits their practicality in real settings.

Other researchers studied partial key exposure from different angles. Blömer and May [13] showed that exposing about half of the LSBs of the private exponent d allows recovery of the factorization using lattice techniques. Ernst et al. [14] extended these attacks to cover large exponents, demonstrating that even d is vulnerable if a sufficiently large number of MSBs or LSBs are known. Patsakis [15] examined implementations where the primes share their least significant bits (LSBS-RSA), and showed that Boolean satisfiability solvers can exploit this overlap to recover the secret key efficiently.

Ghafar et al. [6] first examined RSA moduli-generated primes generated from expressions of the form

$$p = (a^m + r_x), \quad q = (b^m + r_y),$$

but without assuming any LSBs. This structural study was later extended into an explicit LSB-based attack in Ghafar et al. [7] showing that if only a small number of LSBs of both primes are known, then the modulus can be factored efficiently, and further proved that the number of vulnerable primes is exponentially large. Ruzai et al. [8] highlighted the practical risks of unknowingly generating near-square primes, reinforcing the importance of careful prime selection. A key advantage of the three new sign configurations analyzed in this work is that each structure produces a deviation term that remains tightly bounded around $(ab)^{\frac{m}{2}}$. This results in a significantly smaller search interval for recovering the unknown base values compared with previous LSB-based attacks proposed by Ghafar et al. [6, 7] and Ruzai et al. [8]. Consequently, our method requires fewer known LSBs of the primes to guarantee successful factorization: even when the perturbation terms r_x and r_y are very small, the bounds remain polynomially narrow. This substantially broadens the range of RSA moduli that become vulnerable under partial LSB leakage and enlarges the family of special-structured primes previously identified in the literature.

The polynomial-time complexity of the proposed attacks holds under the assumption that the exponent m is treated as a fixed constant independent of the RSA modulus size N. Under this assumption, the enumeration factor $2^{m/2}$ remains constant, and the overall search complexity is polynomial in $\log N$. If m were allowed to grow as a function of $\log N$, the term $2^{m/2}$ would result in exponential-time behavior. Such cases are therefore outside the scope of this work.

Therefore, our contribution builds directly on these LSB approaches. We extend the framework of Ghafar et al. [7] by analyzing three new near-square structures:

(i) $p = (a^m - r_x), q = (b^m - r_y)$;
(ii) $p = (a^m + r_x), q = (b^m - r_y)$;
(iii) $p = (a^m - r_x), q = (b^m + r_y)$.

Each case is formalized by Theorem 2–4 that bound the gap between $\sqrt{N}$ and $(ab)^{\frac{m}{2}}$. This significantly broadens the class of RSA moduli proven vulnerable to LSB-based attacks. Table 1 illustrates the comparison of our results with related existing attacks.

From Table 1, we can see that classical partial-bit attacks, such as those proposed by Rivest and Shamir [9], Coppersmith [10], Hermann and May [11], and Heninger and Shacham [12], need a large number of known bits and are mostly theoretical. Research on key exposure related to the private exponent, including the works of Blömer and May [13] and Ernst et al. [14], and studies on structural overlaps such as Patsakis [15] have expanded the field, but still depend on significant leakage or some unusual assumptions. On the other hand, LSB-based attacks on structured primes proposed by Ghafar et al. [6, 7] show that only LSBs can effectively factor RSA moduli. Our new theorems take this vulnerability further by applying it to multiple sign configurations, revealing that the threat of LSB-based factorization is even more widespread than we previously thought.

Table 1. Comparison of our method against other attacks based on partial information and LSBs.

Attacks	Number of Known Bits	Attack Complexity	Vulnerable Structures	Remarks
Rivest & Shamir (1985) [9]	≈ 60% of bits of one prime (consecutive MSBs)	Exponential (Integer programming)	General RSA primes	First partial key exposure attack
Coppersmith (1996) [5]	≥ 50% MSBs of one prime	Polynomial (LLL lattice reduction)	General RSA primes	Bivariate small-root method; factoring with high bits known
Coppersmith (1997) [10]	Unknown portion bounded by small-root condition	Polynomial (General lattice method)	General RSA primes	General small root problem
Blömer & May (2003) [13]	≥ 50% LSBs of private exponent d	Polynomial (Lattice-based)	RSA with small e	Partial key exposure on d
Ernst et al. (2005) [14]	Large fraction of MSBs/LSBs of d	Polynomial (Lattice-based)	RSA with small e	Extends exponent exposure attacks
Herrmann & May (2008) [11]	70% of bits of a prime of any positions	Polynomial in $\log N$	General RSA primes	Heuristic lattice-based; works for arbitrary bit locations
Heninger & Shacham (2009) [12]	Combined known fraction across p, q > 0.57	Polynomial (probabilistic reconstruction)	General RSA primes	RRA
Patsakis (2013) [15]	Shared LSBs of primes	Polynomial	LSBs-RSA (tampered implementations)	Exploits overlapping prime LSBs
Ghafar et al. (2019) [6]	No bit k leak required	Polynomial	Structured primes $q = (b^m + r_y)$ $p = (a^m + r_x)$,	Structural attack, no LSBs
Ghafar et al. (2020) [7]	Small k LSBs of both primes	Polynomial	$p = (a^m + r_x)$, $q = (b^m + r_y)$	LSB attack, exponentially many vulnerable primes
Ruzai et al. (2022) [8]	No bit leak	Polynomial	Near-square RSA primes	Structural weakness analysis
Our method: Theorem 2	Small k LSBs of both primes	Polynomial	$p = (a^m - r_x)$, $q = (b^m - r_y)$	Negative sign LSB-based attack
Our method: Theorem 3	Small k LSBs of both primes	Polynomial	$p = (a^m + r_x)$, $q = (b^m - r_y)$	Mixed sign LSB-based attack
Our method: Theorem 4	Small k LSBs of both primes	Polynomial	$p = (a^m - r_x)$, $q = (b^m + r_y)$	Mixed sign LSB-based attack

6. Countermeasure of the attack

The LSB-based attacks presented in Sections 3–5 exploit the fact that, for specially structured primes, the value of $(ab)^{\frac{m}{2}}$ lies in a very narrow interval around $\sqrt{N}$. Once a small number of least significant bits of both primes are known, this closeness condition allows efficient reconstruction of p and q in polynomial time. Therefore, any countermeasure must be done to prevent RSA moduli from satisfying these gap conditions at the time of key generation.

Although RSA libraries aim to generate primes uniformly at random, special-structured primes of the form analysed in this work may still arise in practical settings. Such structures are more likely to occur in constrained environments, including embedded devices, Internet of Things hardware, smart cards, or systems employing deterministic or biased random-number generators. Previous studies on near-square RSA primes proposed by Ghafar et al. [6, 7] and Ruzai et al. [8] have shown that these situations can unintentionally restrict the search space of candidate primes, leading to values that satisfy the small-gap condition exploited by our attack. This highlights the practical relevance of detecting and rejecting such primes during RSA key generation.

The bounds differ slightly for each case, but the underlying weakness is identical if the gap between $\sqrt{N}$ and $(ab)^{\frac{m}{2}}$ is sufficiently small, and then the modulus is susceptible to our LSB attacks. Following Ghafar et al. [7], the most effective countermeasure is to enforce a rejection test during RSA key generation. After generating primes p and q, compute

$$\Delta = \left| \sqrt{N} - (ab)^{\frac{m}{2}} \right|.$$

If Δ falls below the smallness threshold indicated by the bounds above, then the modulus N should be discarded and new primes must be generated. This way, we can make sure that no RSA moduli are created that could allow for polynomial-time factorization if there is any LSB leakage.

7. Conclusions

In this paper, we presented new least significant bit (LSB) attacks on RSA moduli from special-structured near-square primes. Building on the framework of Ghafar et al. [7], who considered

$$N = pq = (a^m + r_x)(b^m + r_y),$$

we extended the analysis to three additional configurations:

(i) $N = pq = (a^m - r_x)(b^m - r_y)$;
(ii) $N = pq = (a^m + r_x)(b^m - r_y)$;
(iii) $N = pq = (a^m - r_x)(b^m + r_y)$.

Each case was formalized with lemmas and theorems that established bounds on the difference between $\sqrt{N}$ and $(ab)^{\frac{m}{2}}$. These bounds demonstrate that when the gap is sufficiently small, the modulus can be factored efficiently in polynomial time when given only a few LSBs of both primes.

A comparative analysis with classical partial information attacks highlights the significance of these findings. Although earlier methods required a large number of prime bits to be known, as shown by Rivest and Shamir [9], Coppersmith [10], Hermann and May [11], and Heninger and Shacham [12], our results show that only a small number of LSBs are sufficient when primes are known from near-square structures. Moreover, Ghafar et al. [7] showed that the number of such vulnerable primes is exponentially large, further confirming the practical relevance of the attack.

We also discussed countermeasures for these vulnerabilities. An effective approach is to calculate the gap between $|\sqrt{N}|$ and $|(ab)^{\frac{m}{2}}|$, and to remove the moduli where it is too small, providing a simple and effective safeguard. This countermeasure is cost-effective in terms of computation and works consistently across all near-square structures.

In general, our findings broaden the understanding of LSB-based vulnerabilities in RSA. They demonstrate that the risk is not confined to a single structural case but extends across multiple sign configurations of near-square primes. Future work may focus on extending these results to asymmetric prime constructions, exploring side-channel attacks where LSB leakage naturally occurs, and investigating further refinements of the rejection criteria to ensure robust key generation against advanced attacks.

Overall, this work extends the existing literature on partial-information RSA attacks by showing that three additional near-square prime constructions previously unanalyzed remain vulnerable under minimal LSB leakage. Our deviation bounds refine and generalize those of earlier works and lead to recovery with fewer leaked bits and without requiring MSB knowledge or lattice-based constructions. These results demonstrate that the vulnerabilities of structured RSA primes are broader than previously known, providing a clear advancement over existing LSB-only attack frameworks.

Author contributions

Priscilla Kyle Payne: writing—original draft, formal analysis, software; Wan Nur Aqlili Ruzai: methodology, formal analysis, supervision, writing—review and editing; Amir Hamzah Abd Ghafar: conceptualization, validation; Muhammad Asyraf Asbullah: methodology, validation; Muhammad Rezal Kamel Ariffin: funding acquisition, validation. All authors have read and approved the final version of the manuscript for publication.

Use of Generative-AI tools declaration

The authors declare they have not used Artificial Intelligence (AI) tools in the creation of this article.

Acknowledgments

The authors sincerely thank the anonymous reviewers for their insightful comments and constructive suggestions. This research was supported by the Malaysia Cryptology Technology and Management Centre (PTPKM).

Conflict of interest

The authors declare no conflicts of interest.

References

1. R. L. Rivest, A. Shamir, L. Adleman, A method for obtaining digital signatures and public-key cryptosystems, *Commun. ACM*, **21** (1978), 120–126. https://doi.org/10.1145/359340.359342
2. R. Crandall, R. Pomerance, *Prime numbers: a computational perspective*, 2 Eds., Springer, 2006. https://doi.org/10.1007/0-387-28979-8
3. J. M. Pollard, Theorems on factorization and primality testing, *Proc. Cambridge Philos. Soc.*, **76** (1974), 521–528. https://doi.org/10.1017/S0305004100049252

4. H. W. Lenstra, Factoring integers with elliptic curves, *Ann. Math.*, **126** (1987), 649–673. https://doi.org/10.2307/1971363
5. D. Coppersmith, Finding a small root of a bivariate integer equation; factoring with high bits known, In: U. Maurer, *Advances in cryptology—EUROCRYPT '96*, Springer, 1996, 178–189. https://doi.org/10.1007/3-540-68339-9_16
6. A. H. A. Ghafar, M. R. K. Ariffin, M. A. Asbullah, A new attack on special-structured RSA primes, *Malays. J. Math. Sci.*, **13** (2019), 111–125.
7. A. H. A. Ghafar, M. R. K. Ariffin, M. A. Asbullah, A new LSB attack on special-structured RSA primes, *Symmetry*, **12** (2020), 838. https://doi.org/10.3390/sym12050838
8. W. N. A. Ruzai, A. H. A. Ghafar, N. R. Salim, M. R. K. Ariffin, On (unknowingly) using near-square RSA primes, *Symmetry*, **14** (2022), 1898. https://doi.org/10.3390/sym14091898
9. R. L. Rivest, A. Shamir, Efficient factoring based on partial information, In: H. C. Williams, *Advances in cryptology—CRYPTO'85*, Springer, 1985, 31–34. https://doi.org/10.1007/3-540-39805-8_3
10. D. Coppersmith, Small solutions to polynomial equations, and low exponent RSA vulnerabilities, *J. Cryptology*, **10** (1997), 233–260. https://doi.org/10.1007/s001459900030
11. M. Herrmann, A. May, Solving linear equations modulo divisors: on factoring given any bits, In: J. Pieprzyk, *Advances in cryptology-ASIACRYPT 2008*, Springer, 2008, 406–424. https://doi.org/10.1007/978-3-540-89255-7_25
12. N. Heninger, H. Shacham, Reconstructing RSA private keys from random key bits, In: S. Halevi, *Advances in cryptology-CRYPTO 2009*, Springer, 2009, 1–17. https://doi.org/10.1007/978-3-642-03356-8_1
13. J. Blömer, A. May, New partial key exposure attacks on RSA, In: D. Boneh, *Advances in cryptology-CRYPTO 2003*, Springer, 2003, 27–43. https://doi.org/10.1007/978-3-540-45146-4_2
14. M. Ernst, E. Jochemsz, A. May, B. de Weger, Partial key exposure attacks on RSA up to full size exponents, In: R. Cramer, *Advances in cryptology—EUROCRYPT 2005*, Springer, 2005, 371–386. https://doi.org/10.1007/11426639_22
15. C. Patsakis, Recovering RSA private keys on implementations with tampered LSBs, *Proceedings of the 10th International Conference on Security and Cryptography (SECRYPT 2013)*, 2013, 453–460. https://doi.org/10.5220/0004534904530460
16. A. Takayasu, N. Kunihiro, Partial key exposure attacks on RSA: achieving the Boneh–Durfee bound, *Theor. Comput. Sci.*, **761** (2018), 51–77. https://doi.org/10.1016/j.tcs.2018.08.021
17. 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
18. 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