symmetry

MDPI

Article

On (Unknowingly) Using Near-Square RSA Primes

Wan Nur Aqlili Ruzai [1,†], Amir Hamzah Abd Ghafar [2,3,*,†], Nur Raidah Salim [2,†] and Muhammad Rezal Kamel Ariffin [2,3,†]

[1] School of Distance Education, Universiti Sains Malaysia, Penang 11800, Malaysia
[2] Institute for Mathematical Research, Universiti Putra Malaysia, Serdang 43400, Selangor, Malaysia
[3] Department of Mathematics and Statistics, Faculty of Science, Universiti Putra Malaysia, Serdang 43400, Selangor, Malaysia
* Correspondence: amir_hamzah@upm.edu.my
† These authors contributed equally to this work.

Abstract: The invention in 1978 of the first practical asymmetric cryptosystem known as RSA was a breakthrough within the long history of secret communications. Since its inception, the RSA cryptosystem has become embedded in millions of digital applications with the objectives of ensuring confidentiality, integrity, authenticity, and disallowing repudiation. However, the generation of the RSA modulus, $N = pq$ which requires p and q to be random primes, may accidentally entail the choice of a special type of prime called a near-square prime. This structure of N may be used unknowingly en masse in real-world applications since no current cryptographic implementation prevents its generation. In this study, we show that use of this type of prime will potentially lead to total destruction of RSA. We present three cases of near-square primes used as RSA primes, set in the form of (i) $N = pq = (a^m - r_a)(b^m - r_b)$; (ii) $N = pq = (a^m + r_a)(b^m - r_b)$; and (iii) $N = pq = (a^m - r_a)(b^m + r_b)$. Although (ii) and (iii) are quite similar, p and q must be within the same size range of n-bits, which results in different conditions for both cases. We formulate attacks using three different algorithms to better understand their feasibility. We also provide an efficient countermeasure that it is recommended is adopted by current cryptographic libraries with RSA implementation.

Keywords: public-key cryptography; RSA encryption scheme; RSA primes; cryptanalysis; near-square prime

Citation: Ruzai, W.N.A.; Abd Ghafar, A.H.; Salim, N.R.; Ariffin, M.R.K. On (Unknowingly) Using Near-Square RSA Primes. *Symmetry* **2022**, *14*, 1898. https://doi.org/10.3390/sym14091898

Academic Editors: Alexander Zaslavski and Kuo-Hui Yeh

Received: 8 July 2022
Accepted: 5 September 2022
Published: 11 September 2022

1. Introduction

The famous Rivest—Shamir—Adleman (RSA) public-key cryptosystem is the de facto standard utilized in global technologies as a powerful encryption and decryption mechanism. It was introduced in 1978 by Rivest, Shamir, and Adleman as the first working public-key encryption scheme [1]. It has been included in many cryptographic standards and libraries due to its practicality and simplicity. Its key generation algorithm computes two distinct n-bit primes, p and q, called RSA primes. These primes are the first two private keys of RSA which form $N = pq$, called the RSA modulus. The modulus N is the first RSA public key and each N has its corresponding $\phi(N)$ that is derived from Euler's phi function. Specifically, $\phi(N) = (p-1)(q-1)$; this value is also kept as the RSA private key. The second public key, e (also called the public exponent) is chosen such that $\gcd(\phi(N), e) = 1$. Each e has its corresponding private exponent, d where $ed \equiv 1 \pmod{\phi(N)}$. Thus, the RSA public keys are given by the pair (N, e), while the RSA private keys are represented by the tuple $(p, q, \phi(N), d)$.

Since its introduction, RSA has been successfully retained for forty years for its defence against various attacks [2]. The security of the RSA cryptosystem relies on the hardness of solving the following problems: Firstly the integer factorization problem (IFP), entrenched in the modulus $N = pq$. Secondly, the hardness of solving the key equation $ed - \phi(N)k = 1$

and, finally, the eth root problem in the encryption function. Constant cryptanalysis or 'attacks' on these three problems is crucial to maintain the security of RSA at the highest level [3]. This crucial need for information security has led to the rise of various cryptographic algorithms to implement security in different dimensions and for numerous purposes [4].

Before RSA was introduced, prior results had shown that $p-1$ and $q-1$ that have small factors cause $p \cdot q$ to be vulnerable when factored in polynomial time using the Pollard $p-1$ algorithm [5]. Pollard's $p-1$ algorithm is exceptionally efficient whenever all prime factors of $p-1$ and $q-1$ are small [6]. In addition, a technique known as an estimated prime factor (EPF) was improved by Tahir et. al [7] to solve N generated from balanced or unbalanced primes p and q. Furthermore, Pollard [8] showed that N with a small size is easily factored since the complexity of the factorization algorithm depends on the size of $\sqrt{N}$. Subsequently, research undertaken by many others [9–11] extended this complexity using the number field sieve method, which has dominated efforts to factor the RSA modulus ever since. In 2021, N with 829 bits was successfully factored using this method [12]. Later simulations demonstrated that the 2048-bit RSA modulus can only be factored by a quantum computer with 13 436 qubits within 177 days [13].

The development of quantum computers with effective factoring implementation is unlikely to be realized for many years. Thus, it can be assumed that RSA can still be used securely. However, in this investigation, we show that certain unexplored structures of p and q cause $N = pq$ to be factored without the aid of quantum computation in polynomial time. Specifically, if p and q are near-square primes, then N will be vulnerable. The general definition of a near-square prime is given in the following definition.

Definition 1. *Let a be any integer and m be a power of 2. If $p = a^m \pm r_a$ is a prime number where r_a is a countable integer (for example, $r_a < 100$), then we define p as a near-square prime.*

Prior to this work, factoring of near-square primes was only discussed using a theoretical sieve approach [14] and never in cryptographic settings. However, our previous investigations [15,16] showed that such primes can become vulnerable points in the RSA cryptosystem. Furthermore, the abundance of such primes due to the common size used for RSA primes in standard cryptographic libraries highlights the importance of defining near-square primes with a description that fits RSA in practice. We define below the particular notion of a near-square prime used in all of our attacks.

Definition 2. *Let a be any integer and m be a power of 2. If $p = a^m \pm r_a$ is a prime number where r_a is a 'sufficiently small' integer, then we define p as an r_a-near-square prime.*

In practical circumstances, the term 'sufficiently small' used in Definition 2 refers to the size of integers that are computationally feasible to be performed via an exhaustive search method in the future. For this, readers are advised to refer to the latest standard key size published by the National Institute of Standards and Technology (NIST) [17].

1.1. Motivation for this Paper

Since RSA is still a leading public-key cryptosystem used in digital applications, its security level must be maintained at the highest level. However, it has been found that there are weak primes that are unknowingly used as RSA primes. These weak primes are in the form of near-square primes as defined in Definition 1.

The authors of [15] showed that the number of near-square primes falling under Definition 2 is asymptotic to

$$\pi(\sqrt{s}) \sim \frac{\left\lfloor 2^{\frac{n}{2}}\left(1-2^{-\frac{1}{2}}\right)\right\rfloor}{2}\left(\frac{N^{\gamma}}{\log(s)}+\frac{N^{\gamma}}{\log\left(\sqrt{s}+\left\lfloor 2^{\frac{n}{2}}\left(1-2^{-\frac{1}{2}}\right)\right\rfloor\right)^{2}}\right). \quad (1)$$

Based on Equation (1), s is the smallest squared number with n-bit size, and N^{γ} is the higher value of two near-square primes, r_a and r_b. In terms of RSA-2048, then there are approximately 7.0265×10^{153} near-square primes with 1024-bits [15]. Based on this vast amount of near-square primes, this paper intends to emphasize the importance of *not* selecting near-square primes as RSA primes in the current implementation of the RSA key generation algorithm since there is the possibility they are being used unknowingly in digital applications using RSA today. This is because no current cryptographic standards have imposed any conditions to prevent appointing near-square primes as RSA primes. From the results provided in this paper, we hope that this practice may be amended in the near future to maintain the security of RSA.

1.2. Contribution of This Paper

The results presented in this paper represent a continuation of previous research in [15,16] which exposed the vulnerabilities of using $N = p = (a^m + r_a)(b^m + r_b)$ as the RSA modulus. The main aim of this paper is to cryptanalyze (or attack) three other distinct forms of the RSA modulus with near-square prime factors. Specifically, in the first attack, the RSA primes are set to be in the form of $N = pq = (a^m - r_a)(b^m - r_b)$. In the second attack, the RSA primes have the form of $N = pq = (a^m + r_a)(b^m - r_b)$, while in the third attack, the prime factors are considered to be in the form $N = pq = (a^m - r_a)(b^m + r_b)$. As a result, we show that near-square primes should not be used as RSA primes since they enable N to be factored using the quadratic root method which can feasibly computed by any adversary.

A summary of the structures of near-square primes computed to be N covered in our previous work [16] and in this section is shown in Figure 1.

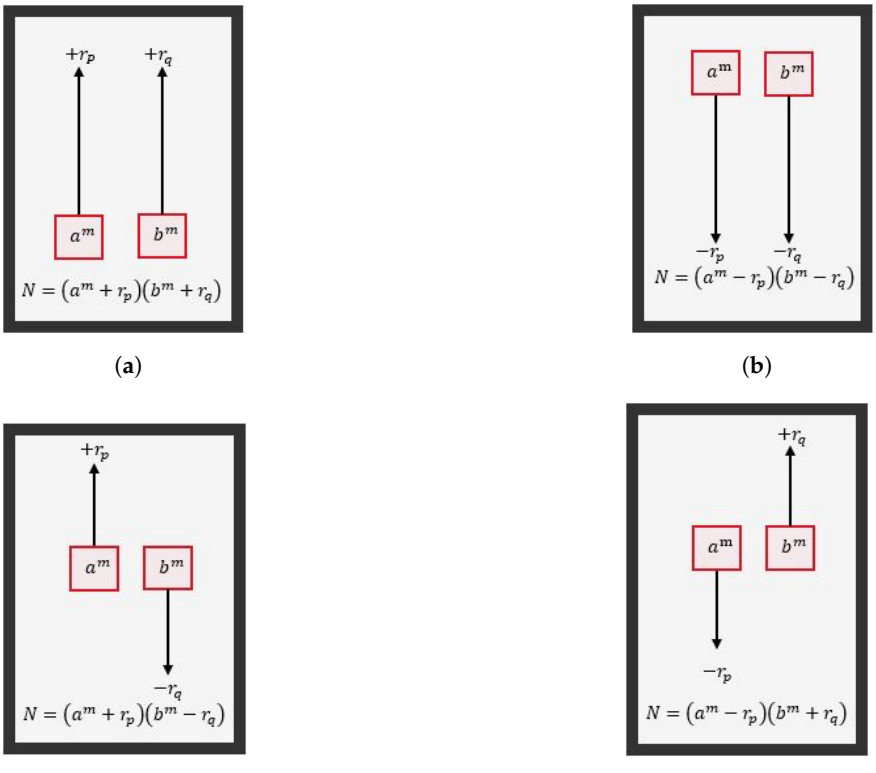

Figure 1. Distinct structures of near-square prime factors are covered in [16] and Section 3 of this paper. This means that we have enclosed all the remaining cases left for using near-square primes as RSA primes. (**a**) The case when $N = pq = (a^m + r_a)(b^m + r_b)$ is presented in [16]. (**b**) The case when $N = pq = (a^m - r_a)(b^m - r_b)$ is presented in Section 3.1. (**c**) The case when $N = pq = (a^m + r_a)(b^m - r_b)$ is presented in Section 3.2. (**d**) The case when $N = pq = (a^m - r_a)(b^m + r_b)$ is presented in Section 3.3.

1.3. Organization of the Paper

The paper is organized as follows: In Section 2, we discuss some previous related studies that how how the structures and conditions imposed on the RSA primes can lead to a total break. Section 3 highlights and compiles three new attacks to factor the RSA modulus N. We show that there are some types of RSA primes that can feasibly lead to a total break of RSA. Additionally, we provide three algorithms to perform the newly proposed attacks. In Section 4, we propose a countermeasure against all the proposed attacks. Our proposed countermeasure is straightforward and can be easily implemented in RSA key generation standard practices. In Section 5, we provide a comparative analysis of attacks that focus on the structure of RSA primes in order to factor N. Finally, we conclude the paper and provide suggestions for future work in Section 6.

2. Related Work

In this section, we review some of the past attacks against RSA that exploit the structures of the primes as the source of the vulnerabilities so that N can be factored in polynomial time.

One of the earliest such papers was presented even before RSA was established.The authors of [5] showed that a composite number can be factored easily if the value preceding one of its prime factors comprises negligibly small primes e.g., $2, 3, 11, 17, \ldots$. This work showed that there exists a condition on a prime that causes the composite number it formed to be easily factored. The algorithm from this condition is called a specific factorization algorithm, i.e., an algorithm that can factor a composite number with the specific condition.

Apart from the specific-purpose factorization algorithm, there are also algorithms designed for any composite number without specific structures. This kind of algorithm is called a general-purpose factorization algorithm. It is of note that the running time of a general-purpose factorization algorithm depends solely on the size of a composite number N. Among the popular factoring algorithms belonging to this category are the quadratic sieve (QS) and general number field sieve (GNFS). In practice, the QS algorithm has proven to be simpler than the GNFS algorithm and is fastest for integers below 100 decimal digits, but no better than the GNFS algorithm for integers with 110–120 digits [4]. It was first introduced by [9] and called the quadratic sieve algorithm. It is regarded as the fastest factoring algorithm for 50–100 bit integers. The authors of Lenstra et al. [11] then introduced a more general approach called the number field sieve factorization algorithm; this algorithm has since been able to factor RSA numbers up to 829 bits [12]. However, since its complexity is sub-exponential, the size of integers remains a significant hurdle for it to break the RSA modulus with 2048 bits efficiently.

In De Weger's result [18], it was then shown that the prime difference of p and q in RSA can influence the result shown previously by [19]. Specifically, if $|p - q| < N^{1/4}$ then the adversary only requires $d < N$ to factor N using a lattice-based attack. This work also showed the relation between the prime difference and the early work on small decryption exponents introduced by [20].

A further assumption commonly applied when attacking RSA is that the adversary is able to know certain bits of RSA private keys beforehand. For example, Ernst et al. [21] showed that, by knowing certain bits of the RSA private key exponent, d, the adversary can factor N in polynomial time. Later, Sarkar and Maitra [22] extended this attack by combining this assumption with an additional assumption that, if certain bits of p and/or q are also known, then the result by [21] can be extended to a more generalized form. However, the attack depends solely on the capabilities of the adversary to collect the secret bits, either from the side-channel method or through faulty coding from implementation.

However, in 2019, Abd Ghafar et al. [16] studied the impact of using near-square RSA primes which yield factorization of the RSA modulus N. Note that, the objective of the work in [16] was to factor an RSA modulus with near-square prime factors, i.e., $N = pq = (a^m + r_a)(b^m + r_b)$, such that $p = a^m + r_a$ and $q = b^m + r_b$. This shows the importance of work exploring extended conditions of the near-square primes. Application

of this research shows that an adversary can also conduct a partial key exposure attack—similar to assumptions made by [21,22]—on the LSBs of primes that satisfy both given conditions [23].

Useful Lemmas

Next, we present some previous findings from [16] that are used as in our result. In Lemma 1, the aim is to show the integer and decimal forms of the equality of $\sqrt{a^m + r}$.

Lemma 1 ([16]). *Suppose a, r are positive integers and $m \geq 2$ is a power of 2. If $\sqrt{a^m + r} = a^{\frac{m}{2}} + \epsilon$, then $\epsilon < \frac{r}{2a^{\frac{m}{2}}}$.*

Proof. Refer to Lemma 3.1 of [16]. □

Subsequently, the lower bound and upper bound of $N^{1/2} - (ab)^{m/2}$ can be determined as shown in Lemma 2.

Lemma 2 ([16]). *Suppose a, b, r_a, r_b are positive integers and $m \geq 2$ is an even integer satisfying $a < b < (2a^m + r_a)^{\frac{1}{m}}$. Let $N = (a^m + r_a)(b^m + r_b)$ where $r_a \leq r_b < N^\gamma$. If $r_a < 2a^{m/2}$ and $r_b < 2b^{m/2}$, then*

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

Proof. Refer to Lemma 3.2 of [16]. □

Then, the following Theorem 1 to find the factorization of the RSA modulus $N = pq$ is proposed upon determining the lower bound and upper bound of $N^{1/2} - (ab)^{m/2}$.

Theorem 1 ([16]). *Suppose a, b, r_a, r_b are positive integers and $m \geq 2$ is an even integer with $a < b < (2a^m + r_a)^{\frac{1}{m}}$. Suppose $N = (a^m + r_a)(b^m + r_b)$ is a valid RSA modulus. Let $r_a < 2a^{m/2}$ and $r_b < 2b^{m/2}$ where $\max\{r_a, r_b\} = N^\gamma$. If N^γ is sufficiently small, then the factorization of N can be performed in polynomial time.*

Proof. Refer to Theorem 3.1 of [16]. □

3. Attacks on Near-Square RSA Primes

This section presents our newly proposed attacks to factor the RSA modulus N. Following the direction of our previous investigations, we propose new results regarding the near-square RSA primes which yield the factorization of N in polynomial time. In the following subsections, we describe three new attacks to factor the RSA modulus N with distinct structures of near-square prime factors. Specifically, the attacks are structured as follows:

- Attack I: When the prime factors have the form $N = pq = (a^m - r_a)(b^m - r_b)$
- Attack II: When the prime factors have the form $N = pq = (a^m + r_a)(b^m - r_b)$
- Attack III: When the prime factors have the form $N = pq = (a^m - r_a)(b^m + r_b)$

3.1. Attack I: $N = pq = (a^m - r_a)(b^m - r_b)$

The objective of Attack I is to factor an RSA modulus with near-square prime factors, i.e., $N = pq = (a^m - r_a)(b^m - r_b)$ where $p = a^m - r_a$ and $q = b^m - r_b$. First, we need to show the equality of $\sqrt{a^m - r}$ to its integer and decimal forms as follows.

Lemma 3. *Suppose a, r are positive integers where $m \geq 2$ is a power of 2. If $\sqrt{a^m - r} = a^{m/2} - \epsilon$, then $\epsilon < \frac{r}{2a^{m/2}}$.*

Proof. Suppose $\sqrt{a^m - r}$ is an integer where a is a positive integer. Then

$$\sqrt{a^m - r} < \sqrt{a^m + \frac{r^2}{4a^m} - r} = \sqrt{\left(a^{m/2} - \frac{r}{2}a^{-m/2}\right)^2} = a^{m/2} - \frac{r}{2}a^{-m/2}.$$

Since $\sqrt{a^m - r} = a^{m/2} - \epsilon$, then $\epsilon < \frac{r}{2}a^{-m/2}$. □

Based on the result obtained in Lemma 3, we proceed to determine the upper and lower bounds of $(ab)^{m/2} - N^{1/2}$ in the next Lemma 4.

Lemma 4. *Suppose a, b, r_a, r_b are positive integers and $m \geq 2$ is a power of 2 such that $a < b < (2a^m + r_a)^{\frac{1}{m}}$. Let $N = (a^m - r_a)(b^m - r_b)$ where $\max\{r_a, r_b\} = N^\gamma$. If $r_a < 2a^{m/2}$ and $r_b < 2b^{m/2}$, then*

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

Proof. We need to satisfy the following statement to prove the lower bound:

$$a^m r_b + b^m r_a > 2(ab)^{m/2}(r_a r_b)^{1/2} \implies -(a^m r_b + b^m r_a) < -2(ab)^{m/2}(r_a r_b)^{1/2}.$$

Observe that

$$\left(a^{m/2}r_b^{1/2} - b^{m/2}r_a^{1/2}\right)^2 = a^m r_b + b^m r_a - 2(ab)^{m/2}(r_a r_b)^{1/2}.$$

Since $\left(a^{m/2}r_b^{1/2} - b^{m/2}r_a^{1/2}\right)^2$ will always be a positive integer, this implies

$$a^m r_b + b^m r_a > 2(ab)^{m/2}(r_a r_b)^{1/2}.$$

Then

$$\begin{aligned}
\sqrt{(a^m - r_a)(b^m - r_b)} &= \sqrt{(ab)^m - a^m r_b - b^m r_a + r_a r_b} \\
&= \sqrt{(ab)^m - (a^m r_b + b^m r_a) + r_a r_b} \\
&< \sqrt{(ab)^m - 2(ab)^{m/2}(r_a r_b)^{1/2} + r_a r_b} \\
&= \sqrt{[(ab)^{m/2} - (r_a r_b)^{1/2}]^2} \\
&= (ab)^{m/2} - (r_a r_b)^{1/2}.
\end{aligned}$$

Thus, $\sqrt{(a^m - r_a)(b^m - r_b)} - (ab)^{m/2} = N^{1/2} - (ab)^{m/2} < -(r_a r_b)^{1/2}$ or can be written as $(ab)^{m/2} - N^{1/2} > (r_a r_b)^{1/2}$.
Now, the task is to prove the upper bound. Observe that $\sqrt{a^m - r_a} = a^{m/2} - \epsilon_1$ and $\sqrt{b^m - r_b} = b^{m/2} - \epsilon_2$. Then, based on Lemma 3,

$$\begin{aligned}
N^{1/2} &= \sqrt{(a^m - r_a)(b^m - r_b)} \\
&= \sqrt{a^m - r_a}\sqrt{b^m - r_b} \\
&= (a^{m/2} - \epsilon_1)(b^{m/2} - \epsilon_2) \\
&= (ab)^{m/2} - a^{m/2}\epsilon_2 - b^{m/2}\epsilon_1 + \epsilon_1\epsilon_2 \\
&> (ab)^{m/2} - \left(a^{m/2}\frac{r_b}{2b^{m/2}} + b^{m/2}\frac{r_a}{2a^{m/2}}\right) + \frac{r_a}{2a^{m/2}}\frac{r_b}{2b^{m/2}}.
\end{aligned} \tag{2}$$

If $r_a < 2a^{m/2}$ and $r_b < 2b^{m/2}$, then

$$\frac{r_a}{2a^{m/2}} \cdot \frac{r_b}{2b^{m/2}} = \frac{r_a r_b}{4(ab)^{m/2}} < \frac{4(ab)^{m/2}}{4(ab)^{m/2}} = 1. \tag{3}$$

If $a < b < (2a^m + r_a)^{1/m}$, then (2) becomes

$$\begin{aligned} N^{1/2} - (ab)^{m/2} &> -\left(a^{m/2}\frac{r_b}{2b^{m/2}} + b^{m/2}\frac{r_a}{2a^{m/2}}\right) + 1 \\ &= -\left(\left(\frac{a}{b}\right)^{m/2}\frac{r_b}{2} + \left(\frac{b}{a}\right)^{m/2}\frac{r_a}{2}\right) + 1 \\ &> -\left((1)^{m/2}\frac{r_b}{2} + (2)^{m/2}\frac{r_a}{2}\right) + 1 \\ &= -\frac{r_b}{2} - 2^{\frac{m}{2}-1}r_a + 1, \end{aligned}$$

or can be written as $(ab)^{m/2} - N^{1/2} < \frac{r_b}{2} + 2^{\frac{m}{2}-1}r_a - 1$.
Thus, the bounds are written as $(r_a r_b)^{1/2} < (ab)^{m/2} - N^{1/2} < \frac{r_b}{2} + 2^{\frac{m}{2}-1}r_a - 1$. □

Next, we propose the following theorem to show that the modulus $N = pq$ can be factored in polynomial time upon obtaining the upper and lower bounds of $(ab)^{m/2} - N^{1/2}$ in Lemma 4.

Theorem 2. *Suppose a, b, r_a, r_b are positive integers and $m \geq 2$ is a power of 2 with $a < b < (2a^m + r_a)^{\frac{1}{m}}$. Let $N = (a^m - r_a)(b^m - r_b)$ be a valid RSA modulus. Let $r_a < 2a^{m/2}$ and $r_b < 2b^{m/2}$ where $\max\{r_a, r_b\} = N^\gamma$. If N^γ is sufficiently small, then N can be factored in polynomial time.*

Proof. Observe that from Lemma 4, we have

$$(r_a r_b)^{1/2} < (ab)^{m/2} - N^{1/2} < \frac{r_b}{2} + 2^{\frac{m}{2}-1}r_a - 1. \tag{4}$$

Thus, (4) can also be rewritten as

$$N^{1/2} + (r_a r_b)^{1/2} < (ab)^{m/2} < N^{1/2} + \frac{r_b}{2} + 2^{\frac{m}{2}-1}r_a - 1. \tag{5}$$

Assume that r_a and r_b are known since $\max\{r_a, r_b\} = N^\gamma$ is sufficiently small. Then, the difference between the lower and upper bounds of (5) is given by

$$\begin{aligned} &N^{1/2} + \frac{r_b}{2} + 2^{\frac{m}{2}-1}r_a - 1 - N^{1/2} - (r_a r_b)^{1/2} \\ &< N^\gamma\left(2^{\frac{m}{2}-1} + \frac{1}{2}\right) - [(\min\{r_a, r_b\})^2]^{1/2} - 1 \\ &= N^\gamma\left(\frac{2^{m/2}+1}{2}\right) - \min\{r_a, r_b\} - 1; \end{aligned} \tag{6}$$

which shows the maximum number of integers to find $(ab)^{m/2}$. If N^γ is sufficiently small, then we can find $(ab)^{m/2}$ in polynomial time.

Note that $(ab)^m$ can be found by computing $\left((ab)^{m/2}\right)^2$. Then, we can observe that

$$\begin{aligned} r_a r_b - N &\equiv r_a r_b - [(a^m - r_a)(b^m - r_b)] \\ &\equiv r_a r_b - (ab)^m + a^m r_b + b^m r_a - r_a r_b \\ &\equiv -(ab)^m + a^m r_b + b^m r_a \quad \text{since } -(ab)^m \ (\text{mod}(ab)^m) \equiv 0 \\ &\equiv a^m r_b + b^m r_a \ (\text{mod}(ab)^m). \end{aligned}$$

From $r_a < 2a^{m/2}$ and $r_b < 2b^{m/2}$, then

$$a^m r_b + b^m r_a < (ab)^m.$$

Here, the value of $(a^m r_b + b^m r_a)$ can be computed without modular reduction. Considering the values of $a^m r_b + b^m r_a, r_a, r_b$, and $(ab)^m$ are already known, p and q can be obtained by finding the solutions of the following quadratic equation:

$$X^2 - (a^m r_b + b^m r_a)X + ((ab)^m r_a r_b).$$

We have determined that $X_1 = a^m r_b$ and $X_2 = b^m r_a$. Since r_a and r_b are known, we can obtain

$$a^m = \frac{X_1}{r_b} \quad \text{and} \quad b^m = \frac{X_2}{r_a}.$$

Thus, the modulus N can be factored by computing

$$\frac{N}{b^m - r_b} = a^m - r_a.$$

This terminates the proof. □

The following Algorithm 1 demonstrates the factorization of $N = pq$ via Theorem 2. The algorithm is as follows:

Algorithm 1 Factoring $N = pq = (a^m - r_a)(b^m - r_b)$ via Theorem 2.

Require: N, r_a, r_b, m
Ensure: p, q
1: Set $i = N^{1/2} + (r_a r_b)^{1/2}$.
2: **while** $i < N^{1/2} + \frac{r_b}{2} + 2^{\frac{m}{2}-1} r_a - 1$ **do**
3: Set $\sigma = \left(i - \left[\sqrt{N}\right]\right)^2$
4: Calculate $z \equiv r_a r_b - N \pmod{\sigma}$
5: Solve $x^2 - zx + \sigma r_a r_b = 0$
6: **if** $\frac{N}{\frac{x_1}{r_b} + r_a}$ or $\frac{N}{\frac{x_2}{r_a} + r_b} \neq$ integer **then**
7: $i++$
8: **else** Compute $p = \frac{x_1}{r_b} + r_a$ and $q = \frac{x_2}{r_a} + r_b$.
9: **end if**
10: **end while**
11: **Output** p and q

3.1.1. The Complexity of Attack I

Observe that the most expensive operation in Algorithm 1 is the modular reduction of calculating $z \equiv r_a r_b - N \pmod{\sigma}$. From [24], we know that the classical modular reduction of modulo σ works at $O(2 \log_2 \sigma)$. Since σ is the potential value of $(ab)^{m/2}$, the maximum integers to find it are less than $N^{\gamma\left(\frac{2^{m/2}+1}{2}\right)}$, as shown in Equation (6). Based on this computation, we have the complexity of Attack I presented in Algorithm 2 to be $O(2 \log_2 N^{\gamma\left(\frac{2^{m/2}+1}{2}\right)})$. As we assume N^γ to be sufficiently small, the attack can also feasibly be computed.

Algorithm 2 Factoring $N = pq = (a^m + r_a)(b^m - r_b)$ via Theorem 3.

Require: N, r_a, r_b, m
Ensure: p, q
Set $i = N^{1/2} - (r_a r_b)^{1/2}$.
while $i < N^{1/2} + \frac{r_b - r_a}{2} + 1$ **do**
  if $i < N^{1/2}$ **then**
    Set $\sigma = \left(\left[\sqrt{N}\right] - i\right)^2$
  else Set $\sigma = \left(i - \left[\sqrt{N}\right]\right)^2$
  end if
  Calculate $z \equiv N + r_a r_b \pmod{\sigma}$
  Solve $x^2 - zx + \sigma r_a r_b = 0$
  if $\frac{N}{\frac{x_1}{r_b} + r_a}$ or $\frac{N}{\frac{x_2}{r_a} + r_b} \neq$ integer **then**
    $i++$
  else Compute $p = \frac{x_1}{r_b} + r_a$ and $q = \frac{x_2}{r_a} + r_b$.
  end if
end while
Output p and q

3.2. Attack II: $N = pq = (a^m + r_a)(b^m - r_b)$

The objective of Attack II is to factor an RSA modulus with near-square prime factors, i.e., $N = pq = (a^m + r_a)(b^m - r_b)$. First, we introduce Lemma 5 that will aid our attack later. It will be used not only in the second attack, but also in the following Attack III.

Lemma 5. *Suppose a, b, r_a, r_b are positive integers and $m \geq 2$ is a power of 2 such that $a < b < (2a^m + r_a)^{\frac{1}{m}}$. If $r_a << 2a^{m/2}$ and $r_b << 2b^{m/2}$, then*

$$a^m r_b + 2(ab)^{m/2}(r_a r_b)^{1/2} > b^m r_a.$$

Proof. If $r_b > r_a$ then

$$\begin{aligned} \frac{a^m r_b + 2(ab)^{m/2}(r_a r_b)^{1/2}}{b^m r_a} &> \frac{a^m r_a + 2a^m r_a}{b^m r_a} \\ &= \frac{3a^m r_a}{b^m r_a} = \frac{3a^m}{b^m} \\ &> 3\frac{a^m}{2a^m + r_a} \approx 3\left(\frac{1}{2}\right) > 1. \end{aligned} \tag{7}$$

Since r_a is negligible in (7) because $r_a << 2a^{m/2}$. This shows that

$$a^m r_b + 2(ab)^{m/2}(r_a r_b)^{1/2} > b^m r_a$$

when $r_b > r_a$. Now, if $r_a > r_b$, then

$$\begin{aligned} \frac{a^m r_b + 2(ab)^{m/2}(r_a r_b)^{1/2}}{b^m r_a} &> \frac{a^m r_b + 2a^m r_b}{b^m r_a} \\ &= \frac{3a^m r_b}{b^m r_a}. \end{aligned} \tag{8}$$

Since r_a and r_b are negligible in (8) because $r_a << 2a^{m/2}$ and $r_b << 2b^{m/2}$, then

$$\begin{aligned}\frac{a^m r_b + 2(ab)^{m/2}(r_a r_b)^{1/2}}{b^m r_a} &> \frac{3a^m}{b^m} \\ &> 3\frac{a^m}{2a^m + r_a} \approx 3\left(\frac{1}{2}\right) > 1.\end{aligned}$$

This shows that

$$a^m r_b + 2(ab)^{m/2}(r_a r_b)^{1/2} > b^m r_a$$

when $r_a > r_b$. This completes the proof. □

Based on the result obtained in Lemma 5, we continue to determine the upper bound and lower bound of $N^{1/2} - (ab)^{m/2}$ as shown in Lemma 6.

Lemma 6. *Suppose a, b, r_a, r_b are positive integers and $m \geq 2$ is a power of 2 such that $a < b < (2a^m + r_a)^{\frac{1}{m}}$. Let $N = (a^m + r_a)(b^m - r_b)$. If $r_a << 2a^{m/2}$ and $r_b << 2b^{m/2}$, then*

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

Proof. By using the result in Lemma 5, we have

$$\begin{aligned}\left(a^{m/2} r_b^{1/2} + b^{m/2} r_a^{1/2}\right)^2 &= a^m r_b + b^m r_a + 2(ab)^{m/2}(r_a r_b)^{1/2} \\ &> b^m r_a + b^m r_a = 2b^m r_a.\end{aligned}$$

Since $\left(a^{m/2} r_b^{1/2} + b^{m/2} r_a^{1/2}\right)^2$ is always a positive number, it follows that

$$\begin{aligned}a^m r_b + b^m r_a + 2(ab)^{m/2}(r_a r_b)^{1/2} - 2b^m r_a &> 0 \\ a^m r_b - b^m r_a &> -2(ab)^{m/2}(r_a r_b)^{1/2} \qquad (9)\end{aligned}$$

or can be written as

$$-(a^m r_b - b^m r_a) < 2(ab)^{m/2}(r_a r_b)^{1/2}.$$

Then,

$$\begin{aligned}\sqrt{(a^m + r_a)(b^m - r_b)} &= \sqrt{(ab)^m - a^m r_b + b^m r_a - r_a r_b} \\ &= \sqrt{(ab)^m - (a^m r_b - b^m r_a) - r_a r_b} \\ &< \sqrt{(ab)^m + 2(ab)^{m/2}(r_a r_b)^{1/2} - r_a r_b} \\ &= \sqrt{\left((ab)^{m/2} + (r_a r_b)^{1/2}\right)^2 - 2r_a r_b} \\ &< \sqrt{\left((ab)^{m/2} + (r_a r_b)^{1/2}\right)^2} \\ &= (ab)^{m/2} + (r_a r_b)^{1/2}.\end{aligned}$$

Thus, the upper bound can be rewritten as

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

Now, we want to prove the lower bound. Based on Lemmas 1 and 3, observe that $\sqrt{a^m + r_a} = a^{m/2} + \epsilon_1$ and $\sqrt{b^m - r_b} = b^{m/2} - \epsilon_2$. Then,

$$\begin{aligned}
N^{1/2} &= \sqrt{(a^m + r_a)(b^m - r_b)} \\
&= \sqrt{a^m + r_a}\sqrt{b^m - r_b} \\
&= (a^{m/2} + \epsilon_1)(b^{m/2} - \epsilon_2) \\
&= (ab)^{m/2} - a^{m/2}\epsilon_2 + b^{m/2}\epsilon_1 - \epsilon_1\epsilon_2 \\
&> (ab)^{m/2} - a^{m/2}\frac{r_b}{2b^{m/2}} + a^{m/2}\frac{r_a}{2a^{m/2}} - \frac{r_a}{2a^{m/2}}\frac{r_b}{2b^{m/2}} \\
&= (ab)^{m/2} - \left(a^{m/2}\frac{r_b}{2b^{m/2}} - a^{m/2}\frac{r_a}{2a^{m/2}}\right) - \frac{r_a}{2a^{m/2}}\frac{r_b}{2b^{m/2}}. \\
&= (ab)^{m/2} - \left(a^{m/2}\frac{r_b}{2b^{m/2}} - \frac{r_a}{2}\right) - \frac{r_a}{2a^{m/2}}\frac{r_b}{2b^{m/2}}. \qquad (10)
\end{aligned}$$

From (3), we know that

$$\frac{r_a}{2a^{m/2}} \cdot \frac{r_b}{2b^{m/2}} < 1.$$

If $a < b < (2a^m + r_a)^{1/m}$, then (10) will become

$$\begin{aligned}
N^{1/2} - (ab)^{m/2} &> -\left(a^{m/2}\frac{r_b}{2b^{m/2}} - \frac{r_a}{2}\right) - 1 \\
&= -\left(\left(\frac{a}{b}\right)^{m/2}\frac{r_b}{2} - \frac{r_a}{2}\right) - 1 \\
&> -\left((1)^{m/2}\frac{r_b}{2} - \frac{r_a}{2}\right) - 1 \\
&= \frac{r_a - r_b}{2} - 1.
\end{aligned}$$

Therefore, the bounds are written as

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

This completes the proof. □

Now, we propose the following theorem to show that the modulus $N = pq$ can be factored in polynomial time upon obtaining the bounds of $N^{1/2} - (ab)^{m/2}$ in Lemma 6.

Theorem 3. *Suppose a, b, r_a, r_b are positive integers and $m \geq 2$ is a power of 2 satisfying $a < b < (2a^m + r_a)^{\frac{1}{m}}$. Let $N = (a^m + r_a)(b^m - r_b)$ be a valid RSA modulus. Let $r_a << 2a^{m/2}$ and $r_b << 2b^{m/2}$ where $\min\{r_a, r_b\}^{1.5} = N^\gamma$. If $\frac{3}{2}N^\gamma$ is sufficiently small, then N can be factored in polynomial time.*

Proof. As observed from Lemma 6, we have

$$\frac{r_a - r_b}{2} - 1 < N^{1/2} - (ab)^{m/2} < (r_a r_b)^{1/2}. \qquad (11)$$

Thus,

$$N^{1/2} - (r_a r_b)^{1/2} < (ab)^{m/2} < N^{1/2} + \frac{r_b - r_a}{2} + 1. \qquad (12)$$

Assume that $\max\{r_a, r_b\} = N^\gamma$. Then, the difference between the upper bound and lower bound of (12) is given by

$$\begin{aligned} & N^{1/2} + \frac{r_b - r_a}{2} + 1 - N^{1/2} + (r_a r_b)^{1/2} \\ & < \frac{r_b - r_a}{2} + (r_a r_b)^{1/2} \\ & < \frac{|r_b - r_a|}{2} + N^\gamma \\ & < \frac{N^\gamma}{2} + N^\gamma = \frac{3}{2} N^\gamma; \end{aligned} \tag{13}$$

which represents the maximum number of integers to find $(ab)^{m/2}$.
Since N^γ is sufficiently small then r_a and r_b can be found in polynomial time. Subsequently, as $\frac{3}{2}N^\gamma$ is sufficiently small, then $(ab)^{m/2}$ can be obtained in polynomial time.
Note that $(ab)^m$ can be found by computing $\left((ab)^{m/2}\right)^2$. Then, we can see that

$$\begin{aligned} N + r_a r_b &\equiv (a^m + r_a)(b^m - r_b) + r_a r_b \\ &\equiv (ab)^m - a^m r_b + b^m r_a - r_a r_b + r_a r_b \\ &\equiv (ab)^m - a^m r_b + b^m r_a \quad \text{since } (ab)^m \ (\text{mod}(ab)^m) \equiv 0 \\ &\equiv (b^m r_a - a^m r_b) \ (\text{mod}(ab)^m). \end{aligned}$$

Notice that $r_a << 2a^{m/2}$ and $r_b << 2b^{m/2}$, hence, it yields

$$b^m r_a - a^m r_b < (ab)^m.$$

Accordingly, we can compute $b^m r_a - a^m r_b$ without modular reduction. Considering the values of $r_a, r_b, (ab)^m$ and $b^m r_a - a^m r_b$ are already known, p and q can be obtained by finding the solutions of the following quadratic equation:

$$X^2 + (b^m r_a - a^m r_b)X - ((ab)^m r_a r_b).$$

We find that $X_1 = a^m r_b$ and $X_2 = -b^m r_a$. Since r_a and r_b are known, we can obtain

$$a^m = \frac{X_1}{r_b} \quad \text{and} \quad b^m = -\frac{X_2}{r_a}.$$

Thus, the modulus N can be factored by calculating

$$\frac{N}{b^m - r_b} = a^m + r_a.$$

This completes the proof. □

As shown in Algorithm 2 is the factorization of $N = pq$ via Theorem 3.

3.2.1. The Complexity of Attack II

Observe that the most expensive operation in Algorithm 2 is the modular reduction of calculating $z \equiv N + r_a r_b \ (\text{mod } \sigma)$. Using the similar reference from Attack I, we know that the classical modular reduction of modulo σ works at $O(2\log_2 \sigma)$. Since σ is the potential value of $(ab)^{m/2}$, the maximum integer to find it is $\frac{3}{2}N^\gamma$, as shown in Equation (13). Based on this computation, we have the complexity of Attack II presented in Algorithm 2 to be $O(2\log_2 \frac{3}{2}N^\gamma)$. As we assume N^γ to be sufficiently small, the attack can also feasibly be computed.

3.3. Attack III: $N = pq = (a^m - r_a)(b^m + r_b)$

The aim of Attack III presented in this section is to factor an RSA modulus with near-square prime factors, i.e., $N = pq = (a^m - r_a)(b^m + r_b)$.

According to the result obtained in Lemma 5, we proceed to determine the lower and upper bounds of $N^{1/2} - (ab)^{m/2}$ for the case when the prime factors are in the forms $p = a^m - r_a$ and $q = b^m + r_b$, respectively.

Lemma 7. *Suppose a, b, r_a, r_b are positive integers and $m \geq 2$ is a power of 2 such that $a < b < (2a^m - r_a)^{\frac{1}{m}}$. Let $N = (a^m - r_a)(b^m + r_b)$. If $r_a << 2a^{m/2}$ and $r_b << 2b^{m/2}$, then*

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

Proof. We refer to (9) to prove the lower bound. It states that

$$a^m r_b - b^m r_a > -2(ab)^{m/2}(r_a r_b)^{1/2}.$$

This inequality is true regardless of the structure of N since it discusses results obtained from Lemma 5 and $\left(a^{m/2} r_b^{1/2} + b^{m/2} r_a^{1/2}\right)^2$ in general.

Then, observe

$$\begin{aligned}
\sqrt{(a^m - r_a)(b^m + r_b)} &= \sqrt{(ab)^m + a^m r_b - b^m r_a - r_a r_b} \\
&> \sqrt{(ab)^m - 2(ab)^{m/2}(r_a r_b)^{1/2} - r_a r_b} \\
&= \sqrt{\left((ab)^{m/2} - (r_a r_b)^{1/2}\right)^2 - 2r_a r_b} \\
&\geq \sqrt{\left((ab)^{m/2} - (r_a r_b)^{1/2}\right)^2} - \sqrt{2r_a r_b} \\
&= (ab)^{m/2} - (r_a r_b)^{1/2} - (2r_a r_b)^{1/2} \\
&= (ab)^{m/2} - (1+\sqrt{2})(r_a r_b)^{1/2}.
\end{aligned}$$

Thus, the lower bound is written as

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

Now, we want to prove the upper bound. Based on Lemmas 1 and 3, observe that $\sqrt{a^m - r_a} = a^{m/2} - \epsilon_1$ and $\sqrt{b^m + r_b} = b^{m/2} + \epsilon_2$. Then,

$$\begin{aligned}
N^{1/2} &= \sqrt{(a^m - r_a)(b^m + r_b)} \\
&= \sqrt{a^m - r_a}\sqrt{b^m + r_b} \\
&= (a^{m/2} - \epsilon_1)(b^{m/2} + \epsilon_2) \\
&= (ab)^{m/2} + a^{m/2}\epsilon_2 - b^{m/2}\epsilon_1 - \epsilon_1\epsilon_2 \\
&< (ab)^{m/2} + b^{m/2}\frac{r_b}{2b^{m/2}} - b^{m/2}\frac{r_a}{2a^{m/2}} - \frac{r_a}{2a^{m/2}}\frac{r_b}{2b^{m/2}} \\
&= (ab)^{m/2} + \frac{r_b}{2} - b^{m/2}\frac{r_a}{2a^{m/2}} - \frac{r_a}{2a^{m/2}}\frac{r_b}{2b^{m/2}}.
\end{aligned} \tag{14}$$

Then (14) will become

$$\begin{aligned} N^{1/2} - (ab)^{m/2} &< \frac{r_b}{2} - b^{m/2}\frac{r_a}{2a^{m/2}} \\ &= \frac{r_b}{2} - \left(\frac{b}{a}\right)^{m/2}\frac{r_a}{2} \\ &< \frac{r_b}{2} - (1)^{m/2}\frac{r_a}{2} \\ &= \frac{r_b - r_a}{2}; \end{aligned}$$

since $a < b < (2a^m - r_a)^{1/m}$. Therefore, the bounds are written as

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

This terminates the proof. □

Subsequently, we propose Theorem 4 to show that the modulus $N = pq$ can be factored in polynomial time upon obtaining the upper and lower bounds of $N^{1/2} - (ab)^{m/2}$ in Lemma 7.

Theorem 4. *Suppose a, b, r_a, r_b are positive integers and $m \geq 2$ is a power of 2 satisfying $a < b < (2a^m - r_a)^{\frac{1}{m}}$. Let $N = (a^m - r_a)(b^m + r_b)$ be a valid RSA modulus. Let $r_a << 2a^{m/2}$ and $r_b << 2b^{m/2}$ where $\max\{r_a, r_b\} = N^\gamma$. If $(1+\sqrt{2})N^\gamma$ is sufficiently small, then N can be factored in polynomial time.*

Proof. As observed from Lemma 7, we have

$$-(1+\sqrt{2})(r_a r_b)^{1/2} < N^{1/2} - (ab)^{m/2} < \frac{r_b - r_a}{2}. \tag{15}$$

Thus,

$$N^{1/2} + \frac{r_a - r_b}{2} < (ab)^{m/2} < N^{1/2} + (1+\sqrt{2})(r_a r_b)^{1/2}. \tag{16}$$

Assume that $\max\{r_a, r_b\} = N^\gamma$. Then, the difference between the lower and upper bounds of (16) is given by

$$\begin{aligned} &\left(N^{1/2} + (1+\sqrt{2})(r_a r_b)^{1/2}\right) - \left(N^{1/2} + \frac{r_a - r_b}{2}\right) \\ &= (1+\sqrt{2})(r_a r_b)^{1/2} - \frac{r_a - r_b}{2} \\ &< (1+\sqrt{2})N^\gamma; \end{aligned} \tag{17}$$

which represents the maximum number of integers required to find $(ab)^{m/2}$.

Since N^γ is sufficiently small then r_a and r_b can be found in polynomial time. Subsequently, as $(1+\sqrt{2})N^\gamma$ is sufficiently small, then we can find $(ab)^{m/2}$ in polynomial time.

As previously mentioned, $(ab)^m$ can be found by calculating $\left((ab)^{m/2}\right)^2$. Then, we can see that

$$\begin{aligned} N + r_a r_b &\equiv (a^m - r_a)(b^m + r_b) + r_a r_b \\ &\equiv (ab)^m + a^m r_b - b^m r_a - r_a r_b + r_a r_b \\ &\equiv (ab)^m + a^m r_b - b^m r_a \quad \text{since } (ab)^m \ (\text{mod}(ab)^m) \equiv 0 \\ &\equiv (a^m r_b - b^m r_a) \ (\text{mod}(ab)^m). \end{aligned}$$

Observe that from $r_a << 2a^{m/2}$ and $r_b << 2b^{m/2}$, then we can have

$$a^m r_b - b^m r_a < (ab)^m.$$

Thus, the value of $a^m r_b - b^m r_a$ can be computed without modular reduction. Considering the values of $r_a, r_b, (ab)^m$ and $b^m r_a - a^m r_b$ are already known, p and q can be obtained by finding the solutions of the following quadratic equation:

$$X^2 + (a^m r_b - b^m r_a)X - ((ab)^m r_a r_b).$$

We find that $X_1 = a^m r_b$ and $X_2 = -b^m r_a$. Since r_a and r_b are known, we can obtain

$$a^m = \frac{X_1}{r_b} \quad \text{and} \quad b^m = -\frac{X_2}{r_a}.$$

Thus, the modulus N can be factored by calculating

$$\frac{N}{b^m + r_b} = a^m - r_a.$$

This completes the proof. □

The Algorithm 3 to factor $N = pq$ via Theorem 4 is as follows:

Algorithm 3 Factoring $N = pq = (a^m + r_a)(b^m - r_b)$ via Theorem 4.

Require: N, r_a, r_b, m
Ensure: p, q
1: Set $i = N^{1/2} + \frac{r_a - r_b}{2}$.
2: **while** $i < N^{1/2} + (1+\sqrt{2})(r_a r_b)^{1/2}$ **do**
3: **if** $i < N^{1/2}$ **then**
4: Set $\sigma = \left(\left\lceil \sqrt{N} \right\rceil - i\right)^2$
5: **else** Set $\sigma = \left(i - \left\lceil \sqrt{N} \right\rceil\right)^2$
6: **end if**
7: Calculate $z \equiv N + r_a r_b \pmod{\sigma}$
8: Solve $x^2 - zx + \sigma r_a r_b = 0$
9: **if** $\frac{N}{\frac{x_1}{r_b} + r_a}$ or $\frac{N}{\frac{x_2}{r_a} + r_b} \neq$ integer **then**
10: $i++$
11: **else** Compute $p = \frac{x_1}{r_b} + r_a$ and $q = \frac{x_2}{r_a} + r_b$.
12: **end if**
13: **end while**
14: **Output** p and q

3.3.1. The Complexity of Attack III

Observe that the most expensive operation in Algorithm 3 is the modular reduction of calculating $z \equiv N + r_a r_b \pmod{\sigma}$. Using the similar reference from Attack I, we know that the classical modular reduction of modulo σ works at $O(2\log_2 \sigma)$. Since σ is the potential value of $(ab)^{m/2}$, the maximum integer to find it is $(1+\sqrt{2})N^\gamma$, as shown in Equation (17). Based on this computation, we have the complexity of Attack III presented in Algorithm 3 to be $O(2\log_2(1+\sqrt{2})N^\gamma)$. As we assume N^γ to be sufficiently small, hence the attack is also feasible to be computed.

4. Countermeasure of the Attacks

From Equations (4), (11), and (15), we observe that all attacks discussed previously have a sufficiently small set of integers to find the actual value of $(ab)^{m/2}$. Since p and q

discussed in the attacks are near-square primes, we can find the nearest squared integer of both by computing

$$\left\lceil p^{1/2} \right\rfloor = \left\lceil (a^m \pm r_a)^{1/2} \right\rfloor = a^{m/2}$$

and

$$\left\lceil q^{1/2} \right\rfloor = \left\lceil (b^m \pm r_b)^{1/2} \right\rfloor = b^{m/2}$$

where r_a and r_b are fundamentally sufficiently small integers, and depend on the types of attacks presented previously. This implies that the owner of the private keys can check the distance between N and $(ab)^{m/2}$ by computing

$$D = \left| N - (ab)^{m/2} \right| = \left| N - \left\lceil p^{1/2} \right\rfloor \cdot \left\lceil q^{1/2} \right\rfloor \right|.$$

If D is sufficiently small, we know that an adversary can find the values of $(ab)^{m/2}$ in polynomial time, as shown in the previous sections. Thus, p and q must not be used as the private keys and another set of RSA primes must be generated. This countermeasure is efficient since it only requires minimal computations; hence, it can easily be adopted in future implementations of RSA.

5. Comparative Analysis

This section provides a comparative analysis between attacks that focus on the structure of RSA primes in order to factor N. For this comparison, we choose five types of attack, as discussed in Section 2: (a) specific-purpose factorization algorithm; (b) general-purpose factorization algorithm; (c) small prime difference; (d) partial key exposure; and (e) near-square primes, as shown in this research.

In most discussions of implementing RSA correctly (e.g., [25]), there are preventive measures to avoid all of these attacks (except for (e)). Hence, we believe the analyses of these attacks (except for (e)) have contributed to maintaining the security of RSA at its highest level, which is our aim in this research. We compare the advantages and disadvantages of these attacks with results presented in this research, as shown in Table 1.

Table 1. Comparison of type of attacks that focus on RSA primes to factor N.

Type of Attacks	Advantage	Disadvantage
Specific-purpose factorization algorithm [5]	Most algorithms can be computed efficiently even for large composite number	The prime factors must satisfy specific conditions or structures
General-purpose factorization algorithm [9,11]	Able to factor a composite number of any size	The complexity is sub-exponential (inefficient) even in the best case
Small prime difference [18]	Able to factor N with full-sized decryption exponent, d	RSA primes must satisfy the condition $\lvert p - q \rvert < N^{1/4}$ to conduct the attack

Table 1. *Cont.*

Type of Attacks	Advantage	Disadvantage
Partial key exposure [21,22]	Flexible since it can be combined with other exposed bits of decryption exponent, d	Requires information of certain bits in advance to accomplish the attack
Near-square primes [16], [Theorems 2–4]	Efficient even for large p and q	Requires both p and q to satisfy conditions that $N = pq = (a^m \pm r_a)(b^m \pm r_b)$ and $N = pq = (a^m \pm r_a)(b^m \mp r_b)$ for sufficiently small r_a, r_b

From the comparison in Table 1, we can see that our attack is efficient since its complexity shown in Sections 3.1.1, 3.2.1 and 3.3.1 are all in polynomial time. However, the structure of p and q must be in specific forms to conduct the attack although the number of primes in these forms is large, as shown in Equation (1).

6. Conclusions and Future Work

We have successfully shown that the RSA modulus with near-square prime factors would render the factorization of N in polynomial time. Specifically, we showed that such primes can become vulnerable points in the RSA cryptosystem. This poses a danger to the existing RSA implementation since there are potentially significant numbers of the RSA modulus that unknowingly employ the structure used in digital applications today. An RSA modulus with near-square prime factors, i.e., $N = pq = (a^m \pm r_a)(b^m \pm r_b)$ and $N = pq = (a^m \pm r_a)(b^m \mp r_b)$ can be factored using the quadratic root method to solve for the prime factors of N. In all of our attacks, it is necessary to examine the distance between N and $(ab)^{m/2}$ which is sufficiently small, i.e., N^γ, in order to find the factorization of the RSA modulus N to be feasible in polynomial time. This poses a danger to the digital applications using RSA today since many implementations ignore this value, and, unknowingly, in some RSA key generation processes, the values are sufficiently small. To avoid this catastrophe for many digital users, we have proposed a countermeasure that can avoid the attacks which fits RSA in practice.

For future work, we suggest that further analysis should be carried out to find the conditions that allow factorization of the RSA modulus when only one of the RSA primes is a near-square prime. If such conditions exist, then we believe many current RSA keys are weak since there is a high possibility to generate such an RSA modulus. This belief is based on the current implementation of cryptographic libraries that is lenient on near-square primes chosen as RSA primes. Thus, a mitigation plan is required to prevent the keys from being exploited by real-world adversaries.

Author Contributions: Conceptualization: A.H.A.G.; methodology, formal analysis, investigation, writing—original draft preparation: W.N.A.R. and A.H.A.G.; writing—review and editing: W.N.A.R., A.H.A.G. and N.R.S.; supervision and project administration: M.R.K.A.; funding acquisition: A.H.A.G. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by Universiti Putra Malaysia (UPM) under its grant Inisiatif Putra Muda scheme (GP-IPM/2021/9704300); and the Institute for Mathematical Research (INSPEM), Universiti Putra Malaysia (UPM) through the Interim Researcher Funding Initiative (INSPEM/IRFI/1/2021/6233205).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest. None of the authors has a financial or personal relationship with other people or organizations that could inappropriately influence or bias the content of this paper.

References

1. Rivest, R.L.; Shamir, A.; Adleman, L. A method for obtaining digital signatures and public-key cryptosystems. *Commun. ACM* **1978**, *21*, 120–126. [CrossRef]
2. Mumtaz, M.; Ping, L. Forty years of attacks on the RSA cryptosystem: A brief survey. *J. Discret. Math. Sci. Cryptogr.* **2019**, *22*, 9–29. [CrossRef]
3. Nitaj, A.; Ariffin, M.R.B.K.; Adenan, N.N.H.; Lau, T.S.C.; Chen, J. Security issues of novel RSA variant. *IEEE Access* **2022**, *10*, 53788–53796. [CrossRef]
4. Ruzai, W.N.A.W.M.; Nitaj, A.; Ariffin, M.R.K.; Mahad, Z.; Asbullah, M.A. Increment of insecure RSA private exponent bound through perfect square RSA diophantine parameters cryptanalysis. *Comput. Stand. Interfaces* **2022**, *80*, 103584. [CrossRef]
5. Pollard, J.M. Theorems on factorization and primality testing. In *Mathematical of the Cambridge Philosophical Society*; Cambridge University Press: Cambridge, UK, 1974; Volume 76, pp. 521–528. [CrossRef]
6. Somsuk, K. An efficient variant of Pollard's $p-1$ for the case that all prime factors of the $p-1$ in B-Smooth. *Symmetry* **2022**, *14*, 312. [CrossRef]
7. Tahir, R.R.M.; Asbullah, M.A.; Ariffin, M.R.K.; Mahad, Z. Determination of a good indicator for estimated prime factor and its modification in Fermat's Factoring Algorithm. *Symmetry* **2021**, *13*, 735. [CrossRef]
8. Pollard, J.M. Monte Carlo methods for index computation ($\pmod p$). *Math. Comput.* **1978**, *32*, 918–924. [CrossRef]
9. Pomerance, C. Analysis and comparison of some integer factoring algorithms. *Comput. Methods Number Theory* **1982**, *154*, 89–139.
10. Montgomery, P.L. Speeding the Pollard and elliptic curve methods of factorization. *Math. Comput.* **1987**, *48*, 243–264. [CrossRef]
11. Lenstra, A.K.; Lenstra, H.W.; Manasse, M.S.; Pollard, J.M. The number field sieve. In *The Development of the Number Field Sieve*; Association for Computing Machinery: New York, NY, USA, 1993; pp. 11–42. [CrossRef]
12. Boudot, F.; Gaudry, P.; Guillevic, A.; Heninger, N.; Thomé, E.; Zimmermann, P. Comparing the difficulty of factorization and discrete logarithm: A 240-digit experiment. In Proceedings of the Annual International Cryptology Conference, Santa Barbara, CA, USA, 17–21 August 2020; Springer: Cham, Denmark, 2020; pp. 62–91. [CrossRef]
13. Gouzien, E.; Sangouard, N. Factoring 2048-bit RSA integers in 177 days with 13 436 qubits and a multimode memory. *Phys. Rev. Lett.* **2021**, *127*, 140503. [CrossRef] [PubMed]
14. Shanks, D. A sieve method for factoring numbers of the form n^2+1. *Math. Comput.* **1959**, *13*, 78–86. [CrossRef]
15. Ghafar, A.H.A.; Ariffin, M.R.K.; Asbullah, M.A. Extending Pollard class of factorable RSA modulus. In Proceedings of the Cryptology and Information Security Conference, Port Dickson, Malaysia, 9–11 July 2018; p. 103.
16. Ghafar, A.; Ariffin, M.; Asbullah, M. A new attack on special-structured RSA primes. *Malays. J. Math. Sci.* **2019**, *13*, 111–125. [CrossRef]
17. Barker, E.; Roginsky, A. Transitioning the Use of Cryptographic Algorithms and Key Lengths: NIST Special Publication 800-131A Revision 2. Technical Report, National Institute of Standards and Technology, 2019. Availale online: https://csrc.nist.gov/publications/detail/sp/800-131a/rev-2/final (accessed on 18 July 2022).
18. De Weger, B. Cryptanalysis of RSA with small prime difference. *Appl. Algebra Eng. Commun. Comput.* **2002**, *13*, 17–28. [CrossRef]
19. Boneh, D.; Durfee, G. Cryptanalysis of RSA with private key d less than $N^{0.292}$. *IEEE Trans. Inf. Theory* **2000**, *46*, 1339–1349. [CrossRef]
20. Wiener, M.J. Cryptanalysis of short RSA secret exponents. *IEEE Trans. Inf. Theory* **1990**, *36*, 553–558. [CrossRef]
21. Ernst, M.; Jochemsz, E.; May, A.; De Weger, B. Partial key exposure attacks on RSA up to full size exponents. In Proceedings of the Annual International Conference on the Theory and Applications of Cryptographic Techniques, Aarhus, Denmark, 22–26 May 2005; Springer: Berlin/Heidelberg, Germany, 2005; pp. 371–386. [CrossRef]
22. Sarkar, S.; Maitra, S. Partial key exposure attacks on RSA and its variant by guessing a few bits of one of the prime factors. *Bull. Korean Math. Soc.* **2009**, *46*, 721–741. [CrossRef]
23. Abd Ghafar, A.H.; Kamel Ariffin, M.R.; Asbullah, M.A. A new LSB attack on special-structured RSA primes. *Symmetry* **2020**, *12*, 838. [CrossRef]
24. Bosselaers, A.; Govaerts, R.; Vandewalle, J. Comparison of three modular reduction functions. In Proceedings of the Annual International Cryptology Conference, Santa Barbara, CA, USA, 22–26 August 1993; Springer: Berlin/Heidelberg, Germany, 1993; pp. 175–186. [CrossRef]
25. Bernstein, D.J.; Heninger, N.; Lange, T. FactHacks: RSA factorization in the real world. *Chaos Comput. Club 29c3* **2012**, 26. Availale online: https://www.hyperelliptic.org/tanja/vortraege/facthacks-29C3.pdf (accessed on 18 July 2022).