AIMS Mathematics, 9(10): 28211–28231.
DOI: 10.3934/math.20241368
Received: 20 July 2024
Revised: 20 August 2024
Accepted: 27 August 2024
Published: 29 September 2024

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

Research article

Concurrent factorization of RSA moduli via weak key equations

Wan Nur Aqlili Ruzai[1], You Ying[2], Khairun Nisak Muhammad[3], Muhammad Asyraf Asbullah[3,4,*] and Muhammad Rezal Kamel Ariffin[2,4]

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

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

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

[4] Institute for Mathematical Research, Universiti Putra Malaysia, 43400 UPM Serdang, Selangor, Malaysia

* **Correspondence:** Email: ma_asyraf@upm.edu.my.

Abstract: The Rivest-Shamir-Adleman (RSA) algorithm is a widely utilized technique in asymmetric cryptography, primarily for verifying digital signatures and encrypting messages. Its security relies on the integer factorization problem's difficulty, which is computationally infeasible with large security parameters. However, this study revealed scenarios where an attacker can concurrently factorize multiple RSA moduli $N_i = p_i q_i$ under specific conditions. The attack is feasible when the attacker possesses a set of RSA key pairs with certain flaws, allowing each N_i to be factored in polynomial time. We identified vulnerabilities in RSA keys that satisfy particular equations by applying Diophantine approximation and Coppersmith's lattice-based technique. For instance, the study demonstrates that if RSA public exponents e_i and moduli N_i adhere to $e_i r - (N_i - p_i - q_i + u_i)s_i = t_i$, where r, s_i, u_i, and t_i are small integers, then all N_i can be factorized simultaneously. Additionally, another vulnerability arises when RSA parameters satisfy $e_i r_i - s(N_i - p_i - q_i + u_i) = t_i$, enabling concurrent factorization with small integers s, r_i, u_i, and t_i. This research expands the understanding of RSA security by identifying specific conditions under which RSA public-key pairs can be compromised. These findings are relevant to the broader field of cryptography and the ongoing efforts to secure communication systems against sophisticated adversaries.

Keywords: asymmetric cryptography; RSA; Diophantine approximations; lattice cryptanalysis; continued fractions
Mathematics Subject Classification: 94A60, 11T71, 68P25

1. Introduction

Information technology is the cornerstone of contemporary society, permeating nearly every facet of daily life. The rapid expansion of today's technological landscape generates an ever-increasing volume of data. As this digital universe expands, the imperative to safeguard and preserve data privacy intensifies, a concern shared by individuals and organizations alike. Security has long been a paramount issue within the realm of computing, particularly concerning the secure transmission of information and data across the Internet. Across various channels, whether via the Internet or through smart devices, reports of data thefts and breaches have shown a consistent upward trend [1]. In response to these challenges, researchers and cryptographers endeavour to innovate novel cryptographic models and enhance existing cryptographic algorithms. These advancements are geared toward practical implementation in real-world applications, aiming to enhance user privacy, fortify data security, strengthen authentication mechanisms, and address many related features [2].

In the domain of cryptographic algorithms, the RSA public key cryptographic algorithm stands out as one of the earliest and most widely adopted. Known as the RSA cryptosystem, it takes its name from its creators: Ron Rivest, Adi Shamir, and Leonard Adleman, who introduced RSA in their seminal 1977 paper [3]. Contemporary applications of the traditional RSA algorithm encompass key exchanges, digital signatures, and secure communication protocols employed in web browsers, chat applications, email services, VPNs, and other methods necessitating secure data transmission between entities. Initially, a fundamental aspect of the RSA cryptosystem involves multiplying two random prime numbers, denoted as p and q, resulting in the RSA modulus, represented by N. Selecting p and q must adhere to the constraint $q < p < 2q$ to prevent factorization using general factoring techniques. Subsequently, the Euler's totient function for N, denoted as $\phi(N)$, is computed as $(p-1)(q-1)$. Once $\phi(N)$ is determined, an integer e less than $\phi(N)$ is chosen. The private key component, d, is computed such that $ed \equiv 1 \pmod{\phi(N)}$. Next, the pair (N, e) is made publicly available for any encryptor, and the pair (d, N) is only available for the decryptor, with p, q, and $\phi(N)$ remaining as secret parameters.

RSA's strength is primarily based on the significant challenge of addressing the integer factorization problem (IFP) for a large integer N and the challenge of resolving the e^{th} root problem. While RSA is generally considered secure, several attacks have been designed to exploit the structure of its key equation. The quadratic sieve (QS) is an efficient algorithm for factoring N, functioning in sub-exponential time due to the use of large n-bit primes for RSA primes p and q [4]. Regarding algorithmic efficiency, the general number field sieve (GNFS) is also highly effective within sub-exponential time, typically with n set to 1024 bits [5]. Note that QS is favored for its simplicity over GNFS and is the fastest method for factoring integers below 100 decimal digits. However, for integers in the 110 to 120 digits range, GNFS outperforms QS.

In 2013, the government digital IDs agenda with RSA keys of Taiwanese citizens was signed by certificate authorities (CAs) and kept in the Citizen Digital Certificates (CDCs) database. As reported in [6], an attack successfully obtained the prime factors of 184 distinct RSA keys of 1024-bit size, revealing significant susceptibility in the RSA keys and emphasizing the need for robust cryptographic practices. Similarly, in 2017, [7] identified vulnerabilities in both Belgium's e-ID cards and Estonia's digital identity cards, compromising millions of RSA keys. This attack allowed the private key to be derived from the public key, contradicting the RSA principle that factoring the primes of the public key should be computationally infeasible. Due to shortcuts in key generation, it has become possible to

factorize 1024-bit keys in minutes and 2048-bit keys in weeks. Ongoing research into RSA public key aggregation is crucial to prevent such vulnerabilities. Note that both notorious cryptanalysis incidents used Coppersmith's partial-key-recovery technique.

In the context of related work, it is crucial to consider advancements in other areas of secure communication and computational frameworks. For instance, the study in [8] explores using advanced neural network architectures in the Internet of Things (IoT) for police applications. Using memristive systems enhances neural networks' dynamic analysis and performance, which can be crucial for real-time data processing and security in IoT environments. Although this study focuses on a different application area, the underlying principle of improving system robustness and security is highly relevant to our work on RSA vulnerabilities. The techniques developed in memristive neural networks could inspire new approaches to strengthening cryptographic systems. Similarly, [9] emphasizes the importance of privacy protection in the Internet of Medical Things (IoMT). By leveraging hyperchaotic behaviour in memristive Hopfield neural networks (HNN), this study addresses the need for secure data transmission and storage in medical applications. This study's emphasis on privacy and data protection parallels our objective of identifying and mitigating vulnerabilities in RSA encryption, ensuring the confidentiality and integrity of sensitive information. Additionally, [10] explores the design of complex dynamical systems with hidden and hyperchaotic behaviours. Developing fractional-order systems with multi-scroll attractors highlights the potential for creating intricate and secure communication protocols. The insights gained from analysing these systems can be applied to cryptographic research, particularly in developing new algorithms that resist attacks by leveraging the unpredictable nature of hyperchaotic systems. This aligns with our study's focus on addressing the inherent weaknesses in RSA by exploring new mathematical models and techniques.

Moreover, [11] introduces a Fibonacci-like prime sequence for prime numbers, the study showcases the utility of such sequences in predicting orbits within a fractal space. This concept relates to our work because both studies use advanced mathematical techniques to solve complex problems. In our cryptanalysis, the identification of weak RSA key equations and the concurrent factorization of multiple RSA moduli are achieved through sophisticated mathematical approaches such as Diophantine approximation and Coppersmith's lattice-based method. The Fibonacci-like prime sequence offers a novel perspective that could inspire further exploration of mathematical structures in cryptographic applications, potentially leading to new methods for identifying and mitigating vulnerabilities. Thus, integrating advanced mathematical techniques and dynamic system analysis, as seen in the referenced studies, provides valuable insights that can be applied to cryptographic research. Our findings on RSA vulnerabilities contribute to the ongoing efforts to enhance the security of encryption systems, ensuring the protection of sensitive information in various applications, from IoT and IoMT to broader communication frameworks. The practical implications of our study highlight the need for continuous evaluation and improvement of cryptographic protocols to stay ahead of potential adversaries.

1.1. Our contributions

This cryptanalysis study explores an adversary's ability to access k RSA moduli, $N_i = p_i q_i$, and their public exponents e_i. We present two attacks exploiting weak RSA key equations. The study demonstrates the simultaneous factorization of k RSA moduli (N_i, e_i) using a constant r that satisfies $e_i r - (N_i - p_i - q_i + u_i)s_i = t_i$. Our primary finding reveals that weaknesses in RSA public-key pairs can be exploited under certain conditions, allowing an adversary to simultaneously

factorize multiple RSA moduli N_i. Another vulnerability is identified when RSA parameters satisfy $e_i r_i - s(N_i - p_i - q_i + u_i) = t_i$, allowing concurrent factorization with small, unknown integers s, r_i, u_i, and t_i. Resolving the factorization of each RSA modulus, N_i, using lattice basis reduction enabled the discovery of this vulnerability. Our study demonstrates this new attack and includes a comparative evaluation with existing research, highlighting the novelty and significance of our findings, which extend the comprehension of RSA vulnerabilities and its security measures.

1.2. Paper organization

The remainder of this paper is organized in the following manner. The essential background information, including foundational theorems on continued fractions, lattice basis reduction, simultaneous Diophantine approximations, and other relevant theorems, is provided in Section 2. Our main work, including the proof of our primary attack and illustrative numerical examples, is presented in Section 3. This is followed by a comparison of our results with previously documented attacks on k RSA moduli instances in Section 4. Finally, the paper concludes with a summary of our findings and key takeaways in Section 5.

2. Foundational information

First, we introduce an essential tool for solving Diophantine equations. Continued fractions approximate rational and irrational numbers, forming the foundation for constructing the RSA cryptanalysis (and its variants). Continued fractions are defined as follows:

Definition 2.1. *Given any positive $\xi \in \mathbb{R}$, start with $\xi_0 = \xi$. For each $i = 1, 2, \ldots, n$, define ξ_i as $\lfloor x_i \rfloor$ and let $\xi_{i+1} = \frac{1}{\xi_i - x_i}$ until $\xi_n \in \mathbb{Z}$. As a result, ξ can be represented as continued fractions in the following form:*

$$\xi = x_0 + \cfrac{1}{x_1 + \cfrac{1}{x_2 + \cfrac{1}{x_3 + \cfrac{1}{\cdots + \cfrac{1}{x_n}}}}}. \tag{2.1}$$

For simplicity, a continued fraction is an expression created by iteratively representing a number as the sum of its integer part and the reciprocal of another number. This process is repeated by writing the new number as the sum of its integer part and another reciprocal, and so on. A finite continued fraction, or terminated continued fraction, stops this iteration after a certain number of steps, using an integer instead of another continued fraction. Conversely, an infinite continued fraction continues indefinitely. In both types, all integers in the sequence, except for the first, must be positive. The integers x_i are known as the coefficients or terms of the continued fraction.

The convergents $\frac{x}{y}$ of ξ are fractions denoted by $\frac{x}{y} = [x_0, x_1, \cdots, x_i]$ for $i \geq 0$. Importantly, suppose $\xi = \frac{x}{y}$ is a rational number with $\gcd(x, y) = 1$. In that case, the continued fraction expansion of ξ can be determined using the Euclidean algorithm in $O(\log(y))$ time.

According to Theorem 2.1 (Legendre's theorem), the unknown integers m and n are guaranteed to be among the list of convergents in the continued fraction expansion of a rational number χ, which satisfies the inequality specified in (2.2).

Theorem 2.1. *Consider a rational number χ and let m and n be positive integers such that $gcd(m, n) = 1$. If the inequality*

$$\left|\chi - \frac{m}{n}\right| < \frac{1}{2n^2}, \tag{2.2}$$

is satisfied, then $\frac{m}{n}$ is a convergent in the continued fraction expansion of χ.

Proof. Please see [12] for the proof.

The primes in the RSA cryptosystem's key generation should be of the same bit size to enhance its security. Allowing uneven factors is a potential security risk because the "small" factor could be easily found. Hence, the notation $q < p < 2q$ indicates that the primes p, q are balanced. Recalled that in Lemma 2.1, we have fixed the lower and upper bounds of the RSA primes of modulus N as follows:

Lemma 2.1. *For an RSA modulus $N = pq$ with p and q being primes of equal bit length such that $q < p < 2q$, the following inequality is satisfied:*

$$\frac{\sqrt{2N}}{2} < q < \sqrt{N} < p < \sqrt{2N}.$$

Proof. Please see Lemma 1 of [13] for the proof.

In the subsequent Lemma 2.2, we demonstrate that the term $p+q$ adheres to the inequalities given in (2.3):

Lemma 2.2. *For an RSA modulus $N = pq$ with p and q being primes of equal bit length with the condition $q < p < 2q$, the following statement is true:*

$$2\sqrt{N} < p + q < \frac{3\sqrt{2N}}{2}. \tag{2.3}$$

Proof. Please see Lemma 1 of [14] for the proof.

Given an accurate approximation of any multiple of a divisor of N, Coppersmith's general result directly offers an efficient factoring method, as shown in Theorems 2.2 and 2.3. These theorems demonstrate that the remaining bits can be determined if a significant portion of p bits are known.

Theorem 2.2. *Consider an RSA modulus $N = pq$ with $p > q$. Suppose there is an unknown integer b that is not divisible by q and an approximation $\tilde{p}$ of bp such that*

$$|bp - \tilde{p}| < N^{\frac{1}{4}}. \tag{2.4}$$

In this case, N can be factorized in polynomial time relative to $\log N$.

Proof. Please see [15] for the proof.

Theorem 2.3. *Suppose $N = pq$ is an RSA modulus with $p > q$. Let b be an unknown integer that does not divide by q. Given an approximation $\tilde{p}$ of bp satisfying*

$$|bp - \tilde{p}| < \sqrt{2}N^{\frac{1}{4}}, \tag{2.5}$$

then N can be factorized in polynomial time relative to $\log N$.

Proof. Please see [15] for the proof.

As a result, having an approximation of $p + q$ allows us to determine an integer p.

Lemma 2.3. *Suppose $N = pq$ is a valid RSA modulus satisfying $q < p < 2q$. Let S denote the approximation of $p + q$ where $S > 2N^{\frac{1}{2}}$ and fulfills:*

$$|p + q - S| < \frac{p - q}{3(p + q)} N^{\frac{1}{4}}.$$

Then an integer p can be estimated as:

$$\tilde{P} = \frac{\left(S + \sqrt{S^2 - 4N}\right)}{2}.$$

This approximation guarantees that:

$$|p - \tilde{P}| < N^{\frac{1}{4}}.$$

Proof. Suppose that $S > 2N^{\frac{1}{2}}$ and let $D = \sqrt{S^2 - 4N}$. We have

$$\begin{aligned} |(p - q)^2 - D^2| &= |(p - q)^2 - (S^2 - 4N)| \\ &= |p^2 - 2pq + q^2 - S^2 + 4pq| \\ &= |p^2 + 2pq + q^2 - S^2| \\ &= |(p + q)^2 - S^2|. \end{aligned} \tag{2.6}$$

Observe that (2.6) can also be written as:

$$(p - q + D)|p - q - D| = (p + q + S)|p + q - S|. \tag{2.7}$$

Dividing (2.7) by $(p - q + D)$ will yield

$$|p - q - D| = \frac{|p + q - S|(p + q + S)}{(p - q + D)}.$$

Next, suppose $|p + q - S| < \frac{p - q}{3(p + q)} N^{\frac{1}{4}}$. Since $\frac{p - q}{3(p + q)} N^{\frac{1}{4}} < N^{\frac{1}{4}}$, then

$$p + q + S < 2(p + q) + N^{\frac{1}{4}} < 3(p + q).$$

Considering $p - q + D > p - q$, we infer that:

$$\begin{aligned} |p - q - D| &< \frac{3(p + q)|p + q - S|}{p - q} \\ &< \frac{3(p + q)}{p - q} \cdot \frac{p - q}{3(p + q)} N^{\frac{1}{4}} \\ &= N^{\frac{1}{4}}. \end{aligned}$$

Next, set $\tilde{p} = \dfrac{S+D}{2}$ which yields:

$$\begin{aligned}\left|p-\tilde{p}\right| &= \left|p-\frac{S+D}{2}\right| \\ &= \frac{|p+q-S+p-q-D|}{2} \\ &\leq \frac{|p+q-S|}{2}+\frac{|p-q-D|}{2} \\ &< \frac{1}{2}\cdot\frac{p-q}{3(p+q)}N^{\frac{1}{4}}+\frac{1}{2}N^{\frac{1}{4}} \\ &< N^{\frac{1}{4}},\end{aligned}$$

where we used $\dfrac{1}{2}\cdot\dfrac{p-q}{3(p+q)} < \dfrac{1}{2}$.

Using Lemma 2.2, we can straightforwardly deduce:

$$\phi(N) = N+1-(p+q) > N+1-\frac{3\sqrt{2N}}{2} > \frac{1}{2}N.$$

Suppose we satisfy specific conditions for approximating the term $p+q$ via S. We can apply the same concept when estimating an integer p via $\tilde{P}$, ensuring the error is less than $N^{1/4}$. A notable benefit of the LLL algorithm, as discussed in [16], is its proficiency in addressing simultaneous Diophantine approximations (see Theorem 2.4).

Theorem 2.4. *Let $\alpha_1, \alpha_2, \ldots, \alpha_n$ be a set of rational numbers, and let $0 < \varepsilon < 1$. Suppose there exists an algorithm capable of efficiently computing a sequence of integers p_i and an integer q with a computation time that is polynomial in $\log(p_i)$ such that $i = 1, 2, \ldots, k$. The theorem holds under the following conditions:*

$$\max_i |q\alpha_i - p_i| < \varepsilon \ \text{ and } \ q \leq 2^{k(k-3)/4}\cdot 3^k\cdot\varepsilon^{-k}.$$

Proof. Refer to [17] for the proof.

A method for factoring the prime numbers p_i and q_i by solving equations involving multiple RSA moduli is detailed in Theorem 2.5.

Theorem 2.5. *Suppose there are k RSA moduli $N_i = p_iq_i$, with N being the smallest among them, and k public exponents e_i for $i = 1, 2, \ldots, k$, where $k \geq 2$. Let δ as $\delta = \frac{k}{2(k+1)}$. If there exists an integer $x < N^{\delta}$ and k integers $y_i < N^{\delta}$ and $|z_i| < \frac{p_i-q_i}{3(p_i+q_i)}y_iN^{1/4}$, satisfying the Diophantine equation $e_ix - y_i\phi(N_i) = z_i$, then k RSA moduli, specifically, $N_1, \ldots, N_k$, can be factorized in polynomial time.*

Proof. Refer to [17] for the proof.

3. Successful cryptanalysis on the system of weak RSA key equations

This section describes conditions under which an attacker can factorize k RSA moduli N_i concurrently. This vulnerability occurs when the attacker accesses RSA public-key pairs with

specific vulnerabilities, allowing each N_i to be efficiently factored in polynomial time. Notably, this vulnerability was identified by employing the lattice basis reduction method to solve the factorization of each RSA modulus N_i. The RSA keys under consideration have parameters that meet the following equations:

- Case I: $e_i r - s_i(N_i - p_i - q_i + u_i) = t_i$.
- Case II: $e_i r_i - s(N_i - p_i - q_i + u_i) = t_i$.

3.1. Case I: Successful cryptanalysis on the system of weak RSA key equations

Based on the earlier motivation, we introduce another vulnerability in the RSA cryptosystem. This newly discovered weakness is revealed when an attacker acquires a collection of RSA key pairs with specific flaws, enabling each N_i to be factored within polynomial time. Assume the RSA public tuples (N_i, e_i) fulfil the following system of equations:

$$e_i r - (N_i - p_i - q_i + u_i)s_i = t_i,$$

where concurrent factorization of the system is possible if the integers r, u_i, s_i, and t_i are suitably small and unknown. This vulnerability was identified by factorizing each RSA modulus N_i using the lattice basis reduction method. The following theorem outlines the specific conditions for this cryptanalysis approach.

Theorem 3.1. *Examine k RSA moduli, each represented as $N_i = p_i q_i$, where this holds true for each i from 1 to k, with $k \geq 3$. Denote by N the smallest among these moduli and by e_i the corresponding public exponents. Assume there exists a fixed integer $r < N^{\delta}$, k integers $s_i < N^{\delta}$, and positive integers u_i such that $u_i + \frac{|t_i|}{s_i} < \frac{p_i - q_i}{p_i + q_i} N^{0.25}$ for each i, where $\delta = \frac{k}{2(k+1)}$. If the values meet the conditions of the system of equations*

$$e_i r - (N_i - p_i - q_i + u_i)s_i = t_i,$$

then it is possible to efficiently factor the primes of k RSA moduli N_i.

Proof. Assume $k \geq 3$ where k RSA moduli $N_i = p_i q_i$ are valid for $i = 1, \cdots, k$. The equation $e_i r - (N_i - p_i - q_i + u_i)s_i = t_i$ can be rewritten as:

$$\begin{aligned} e_i r - N_i s_i - (-p_i - q_i + u_i)s_i &= t_i \\ e_i r - N_i s_i &= t_i - (p_i + q_i - u_i)s_i. \end{aligned} \tag{3.1}$$

Dividing (3.1) with $N_i + u_i$ will yield

$$\left| \frac{e_i r}{N_i} - s_i \right| = \frac{|t_i - (p_i + q_i - u_i)s_i|}{N_i}. \tag{3.2}$$

From (3.1), let $N = \min N_i$ and assume that $s_i < N^{\delta}$, $u_i > 0$, and $u_i + \frac{|t_i|}{s_i} < \frac{p_i - q_i}{p_i + q_i} N^{0.25}$. Then we can have $|t_i| < \frac{p_i - q_i}{p_i + q_i} s_i N^{0.25} < s_i N^{0.25} < N^{\delta} N^{0.25} < N^{\delta + 0.25}$. Since from Lemma 2.2, $p_i + q_i < \frac{3\sqrt{2N}}{2}$, hence we obtain

$$\frac{|t_i - (p_i + q_i - u_i)s_i|}{N_i} \leq \frac{|t_i| + |(p_i + q_i - u_i)s_i|}{N}$$

$$
\begin{aligned}
&< \frac{|t_i| + |(p_i + q_i)s_i|}{N} \\
&< \frac{N^{\delta+0.25} + N^{\delta} \cdot \frac{3\sqrt{2N}}{2}}{N} \\
&= \frac{N^{\delta+0.25} + \frac{3\sqrt{2}}{2}N^{\delta+0.5}}{N} \\
&< \frac{\sqrt{5}N^{\delta+0.5}}{N} = \sqrt{5}N^{\delta-0.5}. \qquad (3.3)
\end{aligned}
$$

Plugging (3.3) into (3.2), we get

$$\left|\frac{e_i r}{N_i + u_i} - s_i\right| < \sqrt{5}N^{\delta-0.5}.$$

The existence of an integer r is demonstrated as follows. Let $\delta = \frac{k}{2(k+1)}$ and $\varepsilon = \sqrt{5}N^{\delta-\frac{1}{2}}$. Here, we get:

$$N^{\delta} \cdot \varepsilon^k = N^{\delta} \cdot N^{k\delta-\frac{k}{2}} \cdot (\sqrt{5})^k = N^{\delta(1+k)-\frac{k}{2}} \cdot (\sqrt{5})^k. \qquad (3.4)$$

Since $\delta = \frac{k}{2(k+1)}$, (3.4) becomes

$$N^{\frac{k}{2(k+1)}(1+k)-\frac{k}{2}} \cdot (\sqrt{5})^k = N^0 \cdot (\sqrt{5})^k = (\sqrt{5})^k < 2^{\frac{k(k-3)}{4}} \cdot 3^k. \qquad (3.5)$$

Combining (3.4) and (3.5), we obtain

$$N^{\delta} < 2^{\frac{k(k-3)}{4}} \cdot 3^k \cdot \varepsilon^{-k}.$$

Hence, if $r < N^{\delta}$, it follows that $r < 2^{\frac{k(k-3)}{4}} \cdot 3^k \cdot \varepsilon^{-k}$. Consequently, it holds that:

$$\left|\frac{e_i r}{N_i} - s_i\right| < \varepsilon, \quad r < 2^{\frac{k(k-3)}{4}} \cdot 3^k \cdot \varepsilon^{-k},$$

which meets the conditions stated in Theorem 2.4, allowing for the successful determination of $r \in \mathbb{Z}$ and $s_i \in \mathbb{Z}$. Subsequently, via the equation $e_i r - (N_i - p_i - q_i + u_i)s_i = t_i$ will lead us to:

$$
\begin{aligned}
e_i r + s_i(p_i + q_i) - s_i(N_i + u_i) &= t_i \\
\frac{e_i r}{s_i} + p_i + q_i - N_i - u_i &= \frac{t_i}{s_i} \\
p_i + q_i - \left(N_i - \frac{e_i r}{s_i}\right) &= \frac{t_i}{s_i} + u_i \\
\left|p_i + q_i - \left(N_i - \frac{e_i r}{s_i}\right)\right| &= \left|\frac{t_i}{s_i} + u_i\right|.
\end{aligned}
$$

Given that $\frac{|t_i|}{s_i} + u_i < \frac{p_i - q_i}{p_i + q_i}N^{\frac{1}{4}}$ and $X_i = N_i - \frac{e_i r}{s_i}$ is an integer close to the sum of $p_i + q_i$ with an absolute difference smaller than $\frac{p_i - q_i}{p_i + q_i}N^{\frac{1}{4}}$, it is possible to approximate p_i via $\tilde{P}_i = \frac{1}{2}(X_i + \sqrt{X_i^2 - 4N_i})$, where $|p_i - \tilde{P}_i| < N_i^{\frac{1}{4}}$. Therefore, Theorem 2.5 states that it is possible to concurrently determine the prime factors of $N_1, N_2, \ldots, N_k$ within polynomial time.

Algorithm 1 is introduced next to demonstrate the process of factorizing $N_i = p_i q_i$ according to the approach described in Theorem 3.1.

Algorithm 1: Simultaneous factorization of k RSA moduli via theorem 3.1

Input: A set of $k \geq 3$ RSA public tuples (N_i, e_i) for each i from 1 to k.

Output: The corresponding prime numbers p_i and q_i or $\perp$.

Choose N to be the minimum of $N_1, N_2, \cdots, N_k$.
Calculate the value of δ via $\delta = \frac{k}{2(k+1)}$.
Calculate the value of ε via $\varepsilon = \sqrt{5}N^{\delta-\frac{1}{2}}$.
Determine C via $C = [2^{\frac{(k+1)(k-4)}{4}} \cdot 3^{k+1} \cdot \varepsilon^{-(k+1)}]$ and $c_i = [-\frac{C \cdot e_i}{N_i}]$.
Form a lattice $\mathcal{L}$ using the matrix $\mathcal{A}$ elements.
Execute the LLL algorithm on matrix $\mathcal{A}$ to generate a reduced basis matrix $\mathcal{B}$.
Determine matrix $\mathcal{D}$ by calculating $\mathcal{D} = \mathcal{B}\mathcal{A}^{-1}$.
Assign labels to each element in the first row of $\mathcal{D}$ from left to right, denoting them as $r, s_1, \cdots, s_k$.
For $i = 1, 2, \cdots, k$ **do**.
    Calculate $X_i = \left[N_i - \frac{e_i r}{s_i}\right]$.
    Calculate $\tilde{P}_i = \frac{1}{2}(X_i + \sqrt{X_i^2 - 4N_i})$.
    Utilize Coppersmith's method on $\tilde{P}_i$ to determine the value of p_i.
    Complete the factorization of N_i by computing $q_i = \frac{N_i}{p_i}$.
    If q_i is an integer, **then** return p_i, q_i.
    Else the algorithm has failed or return $\perp$.
end for

To demonstrate the attack detailed in Theorem 3.1 and implemented via Algorithm 1, we present the following example. This attack was performed on a Windows 10 system, utilizing a computer equipped with an Intel (R) Core (TM) i5-8265U CPU running at 1.60 GHz and 12.0 GB of RAM.

Example 3.1. *Assume an attacker has obtained three different RSA-256 moduli N_i, each paired with its respective public exponent e_i, given by:*

$$N_1 = 154986715500973178740003255217130728882098157712834240825677400304491465372 11,$$
$$e_1 = 812450633738051549076671110145787465340003558205912498882608335806148436791 9;$$
$$N_2 = 454669243180584597096861769244390225522987119505940900139874065870745352332 59,$$
$$e_2 = 161021982145417814444114899790364448274705045130081388169873215646515984105 13;$$
$$N_3 = 174373044438245690532791355983461301646587103652729253915503105399168527372 29,$$

$$e_3 = 560573431290151287579335391598796878121041328232673315472898329917875488 6809.$$

Start by selecting N as the smallest among N_1–N_3.

$$N = 154986715500973178740003255217130728882098157712834240825677400304491465372 11.$$

Setting $k = 3$, we compute δ using the formula $\delta = \frac{k}{2(k+1)} = 0.3750$. Subsequently, ε is determined by $\sqrt{5}N^{\delta-\frac{1}{2}} \approx 5.852 \times 10^{-10}$. The next step involves calculating:

$$C = [2^{\frac{(k+1)(k-4)}{4}} \cdot 3^{k+1} \cdot \varepsilon^{-(k+1)}] = 345332739100000000000000000000000000000000.$$

Additionally, for $i = 1, 2, 3$, we compute:

$$c_i = \left[-\frac{C \cdot e_i}{(N_i + 1)} \right],$$

which yielded the following results:

$$c_1 = -181025710381307170556997940432250134516,$$
$$c_2 = -122300250090817968499412983989936080793,$$
$$c_3 = -111017364591963408590726358270037864780.$$

Proceeding to step 5, a lattice $\mathcal{L}$ is generated by the entries of the matrix $\mathcal{A}$.

$$\mathcal{A} = \begin{bmatrix} 1 & c_1 & c_2 & c_3 \\ 0 & C & 0 & 0 \\ 0 & 0 & C & 0 \\ 0 & 0 & 0 & C \end{bmatrix}.$$

Subsequently, the LLL algorithm is executed on matrix $\mathcal{A}$ to generate a reduced basis matrix $\mathcal{B}$:

$$\mathcal{B} = \begin{bmatrix} B_{11} & B_{12} & B_{13} & B_{14} \\ B_{21} & B_{22} & B_{23} & B_{24} \\ B_{31} & B_{32} & B_{33} & B_{34} \\ B_{41} & B_{42} & B_{34} & B_{44} \end{bmatrix},$$

which has the following entries:

$$B_{11} = 134121503448813021468872928 71, \quad B_{31} = 453303170962888563948303376 26,$$
$$B_{12} = 402567436310423258191621645 64, \quad B_{32} = -146958282437106858361960990 16,$$
$$B_{13} = 153272384231922979886910732 97, \quad B_{33} = 477768347638618263405961825 82,$$
$$B_{14} = 203366435433316918006440166 20, \quad B_{34} = -628972246853791108915342122 80,$$
$$B_{21} = -194620699700859138250633076 24, \quad B_{41} = 843027732614069766454460617 69,$$
$$B_{22} = -459128568853066273101116500 16, \quad B_{42} = -391642028606864984726949188 04,$$
$$B_{23} = 323829378275885564118768658 32, \quad B_{43} = -547534069987036458804525028 17,$$
$$B_{24} = 578780666343314311092550827 20, \quad B_{44} = 460251418240110672392504041 80.$$

In step 7, we compute the matrix $\mathcal{D}$ by performing $\mathcal{D} = \mathcal{B} \cdot \mathcal{A}^{-1}$.

$$\mathcal{D} = \begin{bmatrix} D_{11} & D_{12} & D_{13} & D_{14} \\ D_{21} & D_{22} & D_{23} & D_{24} \\ D_{31} & D_{32} & D_{33} & D_{34} \\ D_{41} & D_{42} & D_{34} & D_{44} \end{bmatrix},$$

where

$$\begin{aligned}
D_{11} &= 134121503448813021468872928 71, \\
D_{12} &= 703073809407904958744338731 7, \\
D_{13} &= 474993869886063669895662588 1, \\
D_{14} &= 431173015532924110154891990 6, \\
D_{21} &= -194620699700859138250633076 24, \\
D_{22} &= -102021460548670815111160666 81, \\
D_{23} &= -689252930616941160954736916 7, \\
D_{24} &= -625665473599265319154606289 7, \\
D_{31} &= 453303170962888563948303376 26, \\
D_{32} &= 237624526291709581682104023 67, \\
D_{33} &= 160538185056552853184750712 92, \\
D_{34} &= 145727635128470564415890217 34, \\
D_{41} &= 843027732614069766454460617 69, \\
D_{42} &= 441920724242170918123949376 82, \\
D_{43} &= 298559884014761669086990678 23, \\
D_{44} &= 271016056562337170303370271 81.
\end{aligned}$$

In step 8, assign the labels r, s_1–s_3 from left to right to the elements in the first row of $\mathcal{D}$ as follows:

$$\begin{aligned}
r &= D_{21} = 134121503448813021468872928 71, \\
s_1 &= D_{22} = 703073809407904958744338731 7, \\
s_2 &= D_{23} = 474993869886063669895662588 1, \\
s_3 &= D_{24} = 431173015532924110154891990 6.
\end{aligned}$$

Now, compute X_i via the formula $X_i = \left\lfloor N_i - \frac{e_i r}{s_i} \right\rfloor$ which outputs:

$$\begin{aligned}
X_1 &= 269402126838606667907066115287494725655, \\
X_2 &= 439723656699230572089169064734715107754, \\
X_3 &= 274132486228506946820442084641848440683.
\end{aligned}$$

Then, we compute $\tilde{P}_i = \frac{1}{2}(X_i + \sqrt{X_i^2 - 4N_i})$ resulting in the following output:

$$\tilde{P}_1 = 186137479859349591396353749057276843688,$$

$$\tilde{P}_2 = 27345566425615621900579824053952580354 8,$$
$$\tilde{P}_3 = 17380655554666303586678759277673954504 3.$$

Observe that the value $\tilde{P}_i$ is an approximation of the prime p_i and satisfies $|p_i - \tilde{P}_i| < N^{0.25}$. Therefore, we utilize Coppersmith's method on $\tilde{P}_i$ to determine p_i:

$$p_1 = 18613747985934959139635374905727684369 7,$$
$$p_2 = 27345566425615621900579824053952580355 3,$$
$$p_3 = 17380655554666303586678759277673954505 9.$$

Finally, we factor N_1–N_3 by identifying q_1–q_3 such that $q_i = \frac{N_i}{p_i}$. The obtained values are as follows:

$$q_1 = 83264646979257076510712366230217881963,$$
$$q_2 = 16626799244307435308337082419518930420 3,$$
$$q_3 = 10032593068184391095365449186510889563 1.$$

Remark 3.1. *It is crucial to observe that in Example 3.1, the value of r, which approximates $N^{0.369}$, exceeds the thresholds set by previous research. Notably, it surpasses the limit of $x < \frac{1}{3}N^{0.25}$ as reported in [18], as well as the bounds of $x \approx N^{0.344}$ found in [17] and $d \approx N^{0.345}$ in [19].*

3.2. Case II: Successful cryptanalysis on the system of weak RSA key equations

Given the earlier motivation discussed, we highlight an additional vulnerability in the RSA encryption system. That is when an adversary obtains a collection of RSA public key pairs exhibiting certain vulnerabilities, this newfound weakness permits the simultaneous factorization of each N_i in polynomial time. In particular, the RSA keys in this collection include parameters that satisfy the conditions of the following system of equations:

$$e_i r_i - s(N_i - p_i - q_i + u_i) = t_i,$$

allowing for concurrent factorization if suitably small, unknown integers s, u_i, r_i, and t_i are present. Notably, this vulnerability was discovered by successfully factoring each RSA modulus N_i via the lattice basis reduction method. The theorem below details this cryptanalytic technique.

Theorem 3.2. *Consider k RSA moduli of the form $N_i = p_i q_i$ for each i from 1 to k, with $k \geq 3$. Let N denote the largest modulus among these, and let e_i be k public exponents where $\min e_i = N^{\alpha}$. Suppose there exists an integer s with $s\frac{|t_i|}{s_i} + u_i < \frac{p_i - q_i}{p_i + q_i} N^{\frac{1}{4}}$, where $\delta = \frac{k(2\alpha-1)}{2(k+1)}$. These conditions must satisfy the equation*

$$e_i r_i - s(N_i - p_i - q_i + u_i) = t_i.$$

Under these conditions, the factorization of all k RSA moduli can be performed simultaneously in polynomial time.

Proof. Rewrite the equation $e_i r_i - s(N_i - p_i - q_i + u_i) = t_i$ as $e_i r_i - sN_i = t_i - s(p_i + q_i - u_i)$ and divide both sides of the equation by e_i. Subsequently, take the absolute value on both sides of the equation, resulting in:

$$\left| \frac{N_i s}{e_i} - r_i \right| = \frac{|t_i - s(p_i + q_i - u_i)|}{e_i}. \tag{3.6}$$

Choose $N = max\{N_i\}$ and assume that $s < N^{\delta}$, $u_i > 0$, and $\frac{|t_i|}{s} + u_i < \frac{p_i - q_i}{p_i + q_i} N^{\frac{1}{4}}$. Consequently, $|t_i| < \frac{p_i - q_i}{p_i + q_i} s N^{\frac{1}{4}} < s N^{\frac{1}{4}} < N^{\delta} N^{\frac{1}{4}} < N^{\delta + \frac{1}{4}}$. Utilizing Lemma 2.2, which asserts $p_i + q_i < \frac{3\sqrt{2}\sqrt{N}}{2}$, and considering $\min e_i = N^{\alpha}$, we derive:

$$\begin{aligned} \frac{|t_i - s(p_i + q_i - u_i)|}{e_i} &\leq \frac{|t_i| + s(p_i + q_i - u_i)}{N^{\alpha}} \\ &< \frac{|t_i| + s(p_i + q_i)}{N^{\alpha}} \\ &< \frac{N^{\delta + \frac{1}{4}} + N^{\delta} \cdot \frac{3\sqrt{2}}{2} N^{\frac{1}{2}}}{N^{\alpha}} \\ &< \frac{\sqrt{5} N^{\delta + \frac{1}{2}}}{N^{\alpha}} = \sqrt{5} N^{\delta + \frac{1}{2} - \alpha}. \end{aligned} \tag{3.7}$$

Substituting (3.7) into (3.6), we obtain

$$\left| \frac{N_i s}{e_i} - r_i \right| < \sqrt{5} N^{\delta + \frac{1}{2} - \alpha}.$$

Our goal now is to verify the existence of an integer s. Define $\delta = \frac{k(2\alpha - 1)}{2(k+1)}$ and $\epsilon = \sqrt{5} N^{\delta + \frac{1}{2} - \alpha}$. The subsequent calculations unfold as follows:

$$\begin{aligned} N^{\delta} \cdot \epsilon^k &= N^{\delta} \cdot N^{k\delta + \frac{k}{2} - k\alpha} \cdot (\sqrt{5})^k \\ &= N^{\delta(1+k) + \frac{k}{2} - k\alpha} \cdot (\sqrt{5})^k \\ &= N^{\frac{k(2\alpha-1)}{2(k+1)}(1+k) + \frac{k}{2} - k\alpha} \cdot (\sqrt{5})^k \\ &= N^0 \cdot (\sqrt{5})^k \\ &= (\sqrt{5})^k \\ &< 2^{\frac{k(k-3)}{4}} \cdot 3^k. \end{aligned}$$

Consequently, we derive

$$N^{\delta} < 2^{\frac{k(k-3)}{4}} \cdot 3^k \cdot \epsilon^{-k}.$$

If $s < N^{\delta}$, then $s < 2^{\frac{k(k-3)}{4}} \cdot 3^k \cdot \epsilon^{-k}$. Now, we summarize for:

$$\left| \frac{N_i s}{e_i} - r_i \right| < \epsilon, \quad s < 2^{\frac{k(k-3)}{4}} \cdot 3^k \cdot \epsilon^{-k}.$$

Thus, the conditions of Theorem 2.4 are satisfied, which enables us to determine s and r_i. Subsequently, using the equation $e_i r_i - s(N_i - p_i - q_i + u_i) = t_i$, we obtain:

$$\begin{aligned} e_i r_i + s(p_i + q_i) - s(N_i + u_i) &= t_i \\ \frac{e_i r_i}{s} + p_i + q_i - N_i - u_i &= \frac{t_i}{s} \\ p_i + q_i - (N_i - \frac{e_i r_i}{s}) &= \frac{t_i}{s} + u_i \\ |p_i + q_i - (N_i - \frac{e_i r_i}{s})| &= |\frac{t_i}{s} + u_i|. \end{aligned}$$

Since $\frac{|t_i|}{s} + u_i < \frac{p_i - q_i}{p_i + q_i} N^{\frac{1}{4}}$, it follows that $|\frac{t_i}{s} + u_i| < \frac{|t_i|}{s} + u_i < \frac{p_i - q_i}{p_i + q_i} N^{\frac{1}{4}}$ and $A_i = N_i - \frac{e_i r}{s}$ approximates $p_i + q_i$ with an error less than $\frac{p_i - q_i}{p_i + q_i} N^{\frac{1}{4}}$. Therefore, according to Lemma 2.3, let $B_i = \sqrt{|A_i^2 - 4N_i|}$, and we can determine an approximation $\widetilde{P_i} = \frac{1}{2}(A_i + B_i)$ of p_i satisfying $|p_i - \widetilde{P_i}| < N^{\frac{1}{4}}$.

As a result, for each $i = 1, 2, \cdots, k$, we can determine p_i using Theorem 2.5, thereby achieving the factorization of $N_1, N_2, \cdots, N_k$.

Next, Algorithm 2 is provided to explain the steps involved in factorizing $N_i = p_i q_i$ using the methodology described in Theorem 3.2.

Algorithm 2: Simultaneous factorization of k RSA moduli using theorem 3.2

Input: $k \geq 3$ set of RSA public key sets (N_i, e_i) such that $i = 1, 2, \cdots, k$.

Output: The corresponding prime numbers p_i and q_i or $\perp$.

1: Select N as the maximum among $N_1, N_2, \cdots, N_k$.
2: Determine α such that $N^\alpha = \min(e_1, e_2, \cdots, e_k)$.
3: Compute δ using $\delta = \frac{k(2\alpha - 1)}{2(k+1)}$.
4: Evaluate $\epsilon = \sqrt{5} N^{\delta + \frac{1}{2} - \alpha}$.
5: Calculate $C = [3^{k+1} \cdot 2^{\frac{(k+1)(k-4)}{4}} \cdot \epsilon^{-k-1}]$ and $c_i = [-\frac{C \cdot N_i}{e_i}]$.
6: Form the lattice $\mathscr{L}$ using the elements of matrix $\mathscr{M}$.
7: Apply the LLL algorithm to matrix $\mathscr{L}$ to obtain matrix $\mathscr{K}$.
8: Evaluate a matrix $\mathscr{H}$ via $\mathscr{H} = \mathscr{K} \cdot \mathscr{M}^{-1}$.
9: Extract each element from the first row of matrix $\mathscr{H}$ and denote them as $s, r_1, r_2, \cdots, r_k$.
10: **For** $i = 1, 2, \cdots, k$ **do**.
11: Compute $A_i = N_i - \frac{e_i r_i}{s}$ and $B_i = \sqrt{|A_i^2 - 4N_i|}$.
12: Compute $\widetilde{p_i} = \frac{1}{2}(A_i + B_i)$.
13: Utilize Coppersmith's method on $\tilde{P}_i$ to determine the value of p_i.
14: Calculate $q_i = \frac{N_i}{p_i}$.
15: **If** q_i ia an integer, **then** output p_i, q_i.
16: **Else** algorithm has failed or output $\perp$.
17: **end for**

In this section, we illustrate to clarify the methodology outlined in Theorem 3.2, employing Algorithm 2. The attack detailed in Theorem 3.2 was executed on a Windows 10 system, using a computer equipped with an Intel(R) Core(TM) i5-8265U CPU operating at 1.60 GHz and 12.0 GB of RAM.

Example 3.2. *Suppose an attacker has acquired three distinct pairs of RSA-140 moduli N_i, each associated with its corresponding public exponent e_i as follows:*

$$\begin{aligned} N_1 &= 20772137973658819116693425088379962399406 3, \\ e_1 &= 14947091956444666045808009203078043358579, \\ N_2 &= 80073352553180200663292316529550278485457 1, \\ e_2 &= 52536194339797485372565490843212651987313, \end{aligned}$$

$$N_3 = 679910618939422296290429319442096304194471,$$
$$e_3 = 778597060257389089895107564438805653812 6.$$

Start by selecting N as the maximum among N_1–N_3 and $e = \min(e_1, e_2, e_3)$.

$$N = \max(N_1, N_2, N_3) = 800733525531802006632923165295502784854571,$$
$$e = \min(e_1, e_2, e_3) = 778597060257389089895107564438805653812 6.$$

Since $N^{\alpha} = \min(e_1, e_2, e_3)$, we can get $\alpha = 0.951981$. Let $k = 3$. Following this, we obtain $\delta = \frac{k(2\alpha-1)}{2(k+1)} = 0.338985$ and $\epsilon = \sqrt{5}N^{\delta+\frac{1}{2}-\alpha} \approx 0.0000411678$.

Next, the following calculation is performed:

$$C = [3^{k+1} \cdot 2^{\frac{(k+1)(k-4)}{4}} \cdot \epsilon^{-k-1}] = 14100171250000000000.$$

Then, we calculate the following for $i = 1, 2, 3$:

$$c_i = [-\frac{C \cdot N_i}{e_i}],$$

which yields the following results:

$$c_1 = -195951629595035077864,$$
$$c_2 = -214908597348891597486,$$
$$c_3 = -1231298787407458103035.$$

Proceeding to step 6, we form a lattice using the entries of the matrix $\mathscr{M}$ as its generating elements.

$$\mathscr{M} = \begin{bmatrix} 1 & c_1 & c_2 & c_3 \\ 0 & C & 0 & 0 \\ 0 & 0 & C & 0 \\ 0 & 0 & 0 & C \end{bmatrix}$$
$$= \begin{bmatrix} 1 & -195951629595035077864 & -214908597348891597486 & -1231298787407458103035 \\ 0 & 14100171250000000000 & 0 & 0 \\ 0 & 0 & 14100171250000000000 & 0 \\ 0 & 0 & 0 & 14100171250000000000 \end{bmatrix}.$$

Subsequently, the LLL algorithm is executed on matrix $\mathscr{M}$ to generate a reduced basis matrix $\mathscr{K}$:

$$\mathscr{K} = \begin{bmatrix} 32672235172336 & -34621218770304 & -333554347296 & -86682869639760 \\ 27174390840652 & 51618763472672 & 136997250199128 & 21007117421180 \\ -243277378527129 & -252743848627544 & 84350599197694 & 40874624736515 \\ 346203828870367 & -294259007256088 & 10140562902638 & 286245255736155 \end{bmatrix}.$$

Next, compute matrix $\mathscr{H} = \mathscr{K} \cdot \mathscr{M}^{-1}$:

$$\mathscr{H} = \begin{bmatrix} 32672235172336 & 454049643158161 & 497975812395885 & 2853106025190199 \\ 27174390840652 & 377645496218931 & 414180091562724 & 2373006249170991 \\ -243277378527129 & -3380852467731491 & -3707926610180756 & -21244220078824867 \\ 346203828870367 & 4811232660680562 & 5276686214668937 & 30232282085503183 \end{bmatrix}.$$

Observe the first row of the matrix $\mathscr{H}$ as follows:

$$\begin{bmatrix}32672235172336 & 454049643158161 & 497975812395885 & 2853106025190199\end{bmatrix}.$$

Now, deduce the elements in the first row of $\mathscr{H}$ as s, r_1–r_3 from left to right as follows:

$$s = 32672235172336, r_1 = 454049643158161, r_2 = 497975812395885, s_3 = 2853106025190199.$$

Then, define $A_i = N_i - \frac{e_i r_i}{s}$ and $B_i = \sqrt{|A_i^2 - 4N_i|}$ which yield the following outputs:

$$\begin{aligned}
A_1 &= 1195802353093372472503,\\
A_2 &= 1802897344297980320018,\\
A_3 &= 1710450864196376374599,\\
B_1 &= 773988209675892859456,\\
B_2 &= 217955802743359334967,\\
B_3 &= 453871879578853827542.
\end{aligned}$$

Then, compute $\widetilde{p_i} = \frac{1}{2}(A_i + B_i)$, which returns:

$$\begin{aligned}
\widetilde{p_1} &= 984895281384632665980,\\
\widetilde{p_2} &= 1010426573520669827493,\\
\widetilde{p_3} &= 1082161371887615101071.
\end{aligned}$$

Observe that the value $\tilde{P}_i$ is an approximation of the prime p_i and satisfies $|p_i - \tilde{P}_i| < N^{0.25}$. Therefore, we utilize Coppersmith's method on $\widetilde{p_i}$ to determine p_i. Finally, we complete the factorization of N_1–N_3 by identifying q_1–q_3 such that $q_i = \frac{N_i}{p_i}$. The obtained values are as follows:

$$\begin{aligned}
p_1 &= 984895281384632665981, \quad q_1 = 210907071708739806523,\\
p_2 &= 792470770777310492489, \quad q_2 = 1010426573520669827539,\\
p_3 &= 1082161371887615101073, \quad q_3 = 628289492308761273527.
\end{aligned}$$

Remark 3.2. *In Example 3.2, we observe that* $\min(r_1, r_2, r_3) \approx N^{0.350}$ *surpasses the limits set by previous research. Specifically, it exceeds the bounds established by Blömer-May ($x < \frac{1}{3}N^{0.25}$) as documented in [18], by Ariffin et al. ($d \approx N^{0.336}$) as reported in [19], and by Nitaj et al. ($x \approx N^{0.337}$) as mentioned in [17].*

4. Comparing results

As shown in Table 1, our findings are compared with established cryptanalysis on multiple instances of RSA moduli, each represented as $N_i = p_i q_i$. We classify these attacks based on the specific modifications made to the key equation's structure and the corresponding requirements to execute them. The table highlights the specific conditions and parameters under which these attacks succeed.

Our comparison criteria include the type of RSA moduli, weaknesses exploited in the key generation process, and mathematical techniques, such as lattice basis reduction or Coppersmith's

method. Our findings demonstrate broader applicability and robustness compared to previously documented attacks requiring specific conditions. This analysis underscores the significant impact of our contributions to cryptanalysis and the need to revisit RSA-based systems' security assumptions.

In summary, Table 1 illustrates the advancements made by our study about existing cryptanalysis, reinforcing the importance of ongoing research to enhance RSA encryption security.

Table 1. Comparison of our attacks with established cryptanalysis of RSA moduli.

Cryptanalysis	Manipulated Key Equation	Associated Requirements
Hinek (2009, [20])	$e_i d - k_i\phi(N_i) = 1$	The exponent d where $d < N_k^{\delta}$, $\delta < \frac{1}{2} - \frac{k}{2(k+1)} - \varepsilon$ where $\varepsilon = \log_{N_k}(6)$
Nitaj et al. (2014, [17])	$e_i x_i - y\phi(N_i) = z_i$	Fixed unknown y where $y < N^{\delta}$, $x_i < N^{\delta}$, $\lvert z_i\rvert < \frac{p_i - q_i}{3(p_i+q_i)} y N^{0.25}$, where $\delta = \frac{k(2\alpha-1)}{2(k+1)}$, $x_i, y \in \mathbb{Z}$
Ariffin et al. (2019, [19])	$e_i d_s - k\phi(N_s) = z_s$	Fixed unknown k where $k < N^{\gamma}$, $d_s < N^{\gamma}$, $z_s < N^{\gamma}$, where $\gamma = \frac{t(1+2\alpha)}{2(4t+1)}$
Ruzai et al. (2022, [2])	$e_i x^2 - y_i^2\phi(N_i) = z_i$	Fixed unknown x where $x^2 < N^{\delta}$, $y_i^2 < N^{\delta}$, $\lvert z_i\rvert < \frac{p_i - q_i}{3(p_i+q_i)} y_i^2 N^{\frac{1}{4}}$, where $\delta = \frac{k}{2(k+1)}$, $x^2, y_i^2 \in \mathbb{Z}$
Ruzai et al. (2024, [21])	$e_i x_i^2 - y^2\phi(N_i) = z_i$	Fixed unknown y where $y^2 < N^{\delta}$, $x_i^2 < N^{\delta}$, $\lvert z_i\rvert < \frac{p_i - q_i}{3(p_i+q_i)} y^2 N^{\frac{1}{4}}$, where $\delta = \frac{k(2\alpha-1)}{2(k+1)}$, $x^2, y_i^2 \in \mathbb{Z}$
Our result: Case I (Theorem 3.1)	$e_i r - s_i(N_i - p_i - q_i + u_i) = t_i$	Fixed unknown r where $r < N^{\delta}$, $s_i < N^{\delta}$, $u_i > 0$, $\frac{\lvert t_i\rvert}{s_i} + u_i < \frac{p_i - q_i}{(p_i+q_i)} N^{\frac{1}{4}}$, where $\delta = \frac{k}{2(k+1)}$
Our result: Case II (Theorem 3.2)	$e_i r_i - s(N_i - p_i - q_i + u_i) = t_i$	Fixed unknown s where $s < N^{\delta}$, $r_i < N^{\delta}$, $u_i > 0$, $\frac{\lvert t_i\rvert}{s_i} + u_i < \frac{p_i - q_i}{(p_i+q_i)} N^{\frac{1}{4}}$, where $\min e_i = N^{\alpha}$, $\delta = \frac{k(2\alpha-1)}{2(k+1)}$

5. Conclusions

In conclusion, our research has effectively shown the simultaneous factorization of (N_i, e_i) for k RSA moduli instances. This was achieved by utilizing a constant value r that meets the weak RSA key equation $e_i r - (N_i - p_i - q_i + u_i)s_i = t_i$. Our study's key finding is that, given certain detailed conditions, an attacker can concurrently factorize multiple RSA moduli N_i. The defect occurs once the attacker obtains a collection of RSA key pairs with specific flaws, enabling each N_i to be factored concurrently in polynomial time, given the existence of suitably small, unknown integers r, s_i, u_i, and t_i.

Additionally, we discovered another weakness where an adversary can exploit RSA parameters that satisfy the system of equations $e_i r_i - (N_i - p_i - q_i + u_i)s = t_i$. These parameters can be factored simultaneously if the necessary small, unknown integers s, r_i, u_i, and t_i are present. It is essential

to note that this vulnerability was identified through the factorization of each RSA modulus N_i using Diophantine approximation and Coppersmith's lattice-based technique. Furthermore, our cryptanalysis demonstrates this new attack and provides a comparative analysis with existing research.

Significantly, the results of this work broaden the scope of insecure RSA decryption exponents. For example, consider the case of rogue certificate authorities (RCAs). RCA can issue seemingly legitimate but compromised certificates, introducing hidden vulnerabilities into the public key infrastructure. These certificates can be exploited between issuance and discovery to compromise private keys. The existence of RCAs underscores the importance of identifying weak public keys. Since the weak keys often meet standard key generation criteria, the cryptosystem may continue operating undetected. If an adversary uncovers these certificates, they can exploit them to derive private keys, even without direct access to private information. Certificate authorities and organizations must adopt proactive strategies to mitigate these risks. Strengthening key generation practices, avoiding predictable patterns, and addressing weak keys as vulnerabilities can help reduce the risks. This paper explores a potential RCA methodology for the RSA cryptosystem, offering practical solutions to enhance cryptosystem resilience.

Author contributions

Wan Nur Aqlili Ruzai: Writing-original draft, Writing-review & editing, Methodology, Conceptualization, Formal analysis; You Ying: Writing-original draft, Formal analysis, Software; Khairun Nisak Muhammad: Writing-review & editing, Validation; Muhammad Asyraf Asbullah: Conceptualization, Methodology, Funding acquisition, Validation, Writing-review & editing; Muhammad Rezal Kamel Ariffin: Supervision, Validation. All authors have read and approved the final version of the manuscript for publication.

Acknowledgments

The authors sincerely thank the anonymous reviewers for their insightful comments and constructive suggestions.

This research was made possible through financial support from Universiti Putra Malaysia.

Conflict of interest

The authors declare no conflict of interest.

References

1. A. Nitaj, M. R. K. Ariffin, N. N. H. Adenan, T. S. C. Lau, J. Chen, Security issues of novel RSA variant, *IEEE Acce.*, **10** (2022), 53788–53796. https://doi.org/10.1109/ACCESS.2022.3175519

2. W. N. A. Ruzai, A. Nitaj, M. R. K. Ariffin, Z. Mahad, M. A. Asbullah, Increment of insecure RSA private exponent bound through perfect square RSA diophantine parameters cryptanalysis, *Comput. Stand. Inter.*, **80** (2022), 103584. https://doi.org/10.1016/j.csi.2021.103584

3. R. Rivest, A. Shamir, L. Adleman, A method for obtaining digital signatures and public-key cryptosystems, *Commun. ACM.*, **21** (1978), 17–28.
4. T. R. Herman, L. Walter, D. Winter, Factoring with the quadratic sieve on large vector computers, *J. Comput. Appl. Math.*, **27** (1989), 267–278. https://doi.org/10.1016/0377-0427(89)90370-1
5. A. H. A. Ghafar, M. R. K. Ariffin, S. M. Yasin, S. H. Sapar, Partial key attack given MSBs of CRT-RSA private keys, *Mathematics*, **8** (2020), 2188. https://doi.org/10.3390/math8122188
6. D. J. Bernstein, Y. A. Chang, C. M. Cheng, L. P. Chou, N. Heninger, T. Lange, et al., Factoring RSA keys from certified smart cards: Coppersmith in the wild, *Adv. Crypt.-ASIACR.*, 2013, 341–360. https://doi.org/10.1007/978-3-642-42045-0_18
7. M. Nemec, M. Sys, P. Svenda, D. Klinec, V. Matyas, The return of Coppersmith's attack: Practical factorization of widely used RSA moduli, *Proc. 2017 ACM SIGSAC Conf. Comput. Commun. Secur.*, 2017, 1631–1648. https://doi.org/10.1145/3133956.3133969
8. H. Lin, X. Deng, F. Yu, Y. Sun, Grid multi-butterfly memristive neural network with three memristive systems: Modeling, dynamic analysis, and application in police IoT, *IEEE Int. Things J.*, 2024. https://doi.org/10.1109/JIOT.2024.3409373
9. H. Lin, X. Deng, F. Yu, Y. Sun, Diversified butterfly attractors of memristive HNN with two memristive systems and application in IoMT for privacy protection, *IEEE Trans. Comput.-Aided Design Int. Circu. Syst.*, 2024. https://doi.org/10.1109/TCAD.2024.3429410
10. F. Yu, S. Xu, Y. Lin, T. He, C. Wu, H. Lin, Design and analysis of a novel fractional-order system with hidden dynamics, hyperchaotic behavior and multi-scroll attractors, *Mathematics*, **12** (2024), 2227. https://doi.org/10.3390/math12142227
11. J. H. He, Q. Yang, C. H. He, A. A. Alsolami, Unlocking the plants' distribution in a fractal space, *Fractals*, **31** (2023), 2350102. https://doi.org/10.1142/S0218348X23501025
12. M. J. Wiener, Cryptanalysis of short RSA secret exponents, *IEEE Trans. Inform. Theory*, **36** (1990), 553–558. https://doi.org/10.1109/18.54902
13. A. Nitaj, Another generalization of Wiener's attack on RSA, *Prog. Crypt.-AFRICACRYPT*, 2008, 174–190. https://doi.org/10.1007/978-3-540-68164-9_12
14. A. Nitaj, Diophantine and lattice cryptanalysis of the RSA cryptosystem, *Artif. Intell. Evolut. Comput. Metah.*, 2013, 139–168. https://doi.org/10.1007/978-3-642-29694-9_7
15. D. Coppersmith, Small solutions to polynomial equations, and low exponent RSA vulnerabilities, *J. Cryptology*, **10** (1997), 233–260. https://doi.org/10.1007/s001459900030
16. A. K. Lenstra, H. W. Lenstra, L. Lovász, Factoring polynomials with rational coefficients, *Mathemat. Annalen*, **261** (1982), 515–534. https://doi.org/10.1007/BF01457454
17. A. Nitaj, M. R. K. Ariffin, D. I. Nassr, H. M. Bahig, New attacks on the RSA cryptosystem, in *Prog. Crypt.-AFRICACRYPT*, 2014, 178–198. https://doi.org/10.1007/978-3-319-06734-6_12
18. J. Blömer, A. May, A generalized Wiener attack on RSA, *Publ. Key Crypt.-PKC*, 2004, 1–13. https://doi.org/10.1007/978-3-540-24632-9_1

19. M. R. K. Ariffin, S. I. Abubakar, F. Yunos, M. A. Asbullah, New cryptanalytic attack on RSA modulus $N = pq$ using small prime difference method, *Cryptography*, **3** (2019), 2. https://doi.org/10.3390/cryptography3010002

20. M. J. Hinek,*Cryptanalysis of RSA and its Variants*, New York: Chapman and Hall/CRC, 2009. https://doi.org/10.1201/9781420075199

21. W. N. A. Ruzai, M. R. K. Ariffin, M. A. Asbullah, A. H. Abd Ghafar, New simultaneous Diophantine attacks on generalized RSA key equations, *J. King Saud Univ.-Computer Inf. Sci.*, **36** (2024), 102074. https://doi.org/10.1016/j.jksuci.2024.102074

AIMS Press