Journal of Cybersecurity and Privacy

MDPI

Article

New Semi-Prime Factorization and Application in Large RSA Key Attacks

Anthony Overmars and Sitalakshmi Venkatraman *

Department of Information Technology, Melbourne Polytechnic, 77 St. Georges Rd., Preston, VIC 3072, Australia; AnthonyOvermars@melbournepolytechnic.edu.au

* Correspondence: SitaVenkat@melbournepolytechnic.edu.au

Abstract: Semi-prime factorization is an increasingly important number theoretic problem, since it is computationally intractable. Further, this property has been applied in public-key cryptography, such as the Rivest–Shamir–Adleman (RSA) encryption systems for secure digital communications. Hence, alternate approaches to solve the semi-prime factorization problem are proposed. Recently, Pythagorean tuples to factor semi-primes have been explored to consider Fermat's Christmas theorem, with the two squares having opposite parity. This paper is motivated by the property that the integer separating these two squares being odd reduces the search for semi-prime factorization by half. In this paper, we prove that if a Pythagorean quadruple is known and one of its squares represents a Pythagorean triple, then the semi-prime is factorized. The problem of semi-prime factorization is reduced to the problem of finding only one such sum of three squares to factorize a semi-prime. We modify the Lebesgue identity as the sum of four squares to obtain four sums of three squares. These are then expressed as four Pythagorean quadruples. The Brahmagupta–Fibonacci identity reduces these four Pythagorean quadruples to two Pythagorean triples. The greatest common divisors of the sides contained therein are the factors of the semi-prime. We then prove that to factor a semi-prime, it is sufficient that only one of these Pythagorean quadruples be known. We provide the algorithm of our proposed semi-prime factorization method, highlighting its complexity and comparative advantage of the solution space with Fermat's method. Our algorithm has the advantage when the factors of a semi-prime are congruent to 1 modulus 4. Illustrations of our method for real-world applications, such as factorization of the 768-bit number RSA-768, are established. Further, the computational viabilities, despite the mathematical constraints and the unexplored properties, are suggested as opportunities for future research.

Keywords: Euler's factorization; Pythagorean quadvery minimal, with most computations operating onruples; Pythagorean triples; Lebesgue identity; Brahmagupta–Fibonacci identity; semi-primes; RSA cryptosystem

Citation: Overmars, A.; Venkatraman, S. New Semi-Prime Factorization and Application in Large RSA Key Attacks. *J. Cybersecur. Priv.* **2021**, *1*, 660–674. https://doi.org/10.3390/jcp1040033

Academic Editors: Nour Moustafa and Danda B. Rawat

Received: 10 August 2021
Accepted: 4 November 2021
Published: 12 November 2021

1. Introduction

Prime numbers have caught the attention of mathematicians since the work of Euclid due to their unfathomable structural properties. This paper leverages some elegant properties of prime numbers beyond their basic definition of being divisible by themselves and one only. They also possess the property of being randomly distributed, which is not exploited fully [1–3]. The current perception is that there appears to be only a limited understanding of their underlying structure, and several mathematicians are constantly trying to uncover the mysteries behind these prime numbers. There is still much to be carried out, and areas of further interest are channeled towards a better understanding of the structure of primes for arriving at faster prime number generating algorithms and faster solutions to the prime factorization problem [4–7]. There is a need for generating more robust primes that are less susceptible to known factorization methods. In this paper, we draw attention to creating new approaches for the prime factorization of large prime

factors or the semi-prime factorization problem. The focus of this research problem has an impact worldwide due to its practical application in digital communication, and in particular, the associated information security attacks and privacy challenges of today [8,9].

In information security, large semi-primes have applications in encryption algorithms. They are used for generating public keys and private keys, such as in the Rivest–Shamir–Adleman (RSA) cryptosystems [10]. The property that the prime factorization of large numbers is a challengingly difficult task is well utilized in RSA-based encryption algorithms. Due to the dominant application of the RSA public-key primitive in cryptography, the security of RSA has been extensively analyzed for various attack scenarios [11]. In this paper, we take a modest step further by proposing new methods of semi-prime factorization of the RSA primitive.

Previous research [1] proved that the semi-primes can be represented as the sum of four squares. A new factorization method, by exploiting the relationship among the four squares, was proposed as a faster alternative to Euler's method, as given in [12]. The purpose of this paper is to explore the topic of factorization further with a computationally simple approach for applications in RSA cryptography. In earlier work [1], we showed that a semi-prime N could be constructed from two primes, p_1 and p_2, in accordance with Fermat's Christmas theorem, as given in [13]. In other words, Fermat stated that an odd prime p can be represented as the sum of two squares of integers x_1 and x_2, if and only if $1 \ (mod\ 4) \equiv p = x_1^2 + x_2^2$. Hence, this determined which numbers can be represented as the sum of two squares and was later proved by Euler [12]. It is also mathematically represented that the semi-prime product is congruent to $1(mod\ 4) \equiv p_1 p_2$. This paper advances further by applying the Brahmagupta–Fibonacci identity uniquely to establish that the product of these two primes (which is a sum of four squares) could be mathematically reduced to two sums of two squares [14]. In other words, we prove mathematically how the Brahmagupta–Fibonacci identity reduces these four Pythagorean quadruples to two Pythagorean triples. Hence, our proposed factorization method based on this property would be computationally successful whenever the semi-prime, a product of two primes, is such that both factors are congruent to 1 modulus 4.

In this work, we leverage on the gaps found in the literature towards providing a novel proposal for semi-prime factorization. While there are several properties of Pythagorean triples, new patterns based on these properties are yet to be researched in the context of semi-prime factorization [15]. The main contributions of the paper are envisaged via the key features of our proposed factorization method, as listed below:

i. The novel semi-prime factorization method uses simple number theory uniquely for the first of its kind;
ii. The method applies new patterns of Pythagorean tuples and triples that are unexplored so far in literature;
iii. By employing simple arithmetic operations, the semi-prime factorization algorithm assures a low order of computing cost;
iv. The algorithm exhibits an enhanced solution space as compared to Fermat's method.

The paper is organized as follows: Section 2 provides related works, and Section 3 postulates the background theory of our proposed new method, which includes definitions, a theorem, and four lemmas. In Section 4, we show how to factorize semi-primes using Pythagorean quadruples and triples by proving the theorems and the four lemmas. Further, we provide the algorithm of our proposed semi-prime factorization method and summarize its complexity, comparison, and constraints with existing similar works qualitatively. In Section 5, case study examples illustrate the ease of computing the factors of a semi-prime integer with the proposed approach numerically. Moreover, we demonstrate the application of the method for RSA-768 successfully. Finally, we draw key conclusions and future research directions in Section 6.

2. Related Works

Several studies on the important problem of semi-primes factorization have challenged the intractability of RSA [16–18]. Since the first RSA cryptanalytic attack by Wiener in 1990 using the continued fractions method [19], several methods have been explored, such as the lattice reduction approach by Coppersmith [20] and by Blomer and May [21], as well as the Boneh and Durfee attack on short decryption exponents of RSA [22]. More recent works approach the problem differently, with varying purposes, such as to find weak RSA keys for transport layer security (TLS) attacks [23], LogJam [24], or to factorize RSA keys for smartcard cryptography attacks on several devices [25].

Historically, general-purpose factorization of integers can be dated back to the continued fraction factorization method (CFRAC) introduced by Lehmer and Powers in 1931 [26], which was later implemented as a computer algorithm by Morrison and Brillhart [27]. Subsequently, Pomerance and Wagstaff devised an improved algorithm [28], and such works including Pollard's ρ algorithm [29] and Pomerance's quadratic sieve method (1985) [30] lay the foundation for generating interests in general-purpose factorization of integers in various applications. In the context of RSA cryptography applications, recent works have considered such known factorization methods to focus on security parameters such as the length of the prime factors p_1 and p_2, of the RSA modulus $n = p_1 p_2$, or other structural properties of the primes. Modifications of existing methods have become popular recently, such as using the prime sum $p_1 + p_2$ with sublattice reduction techniques and Coppersmith's methods [31] or using a small prime difference $p_1 - p_2$ method with Wiener's original method [32]. The method by Lenstra's elliptic-curve method also serves as the state-of-the-art proposal for several future studies [33]. A recent work has considered Chengs's $4p - 1$ method [34] to provide simplified and asymptotically deterministic versions, as it is similar to the well-known Lenstra's methods [35]. The study analyzed existing methods as the means of a potential backdoor for the RSA primes generated.

A survey of the literature shows that the factorization problem of prime numbers is gaining research popularity, with a recent focus towards developing efficient mathematical techniques that are computationally faster and simpler [36–39]. Recent research interest in polynomials, which generate sums of squares, has featured applications to cryptography [40–45]. In this context, our previous works leverage the semi-prime representation as the sum of four squares [1,46] with an enhancement to Euler's method [47,48]. Such a factorization method focusing on the special form of primes allows for an efficient factorization of RSA moduli. Thus, an adversary is motivated to subvert the prime generation to produce such RSA keys and could serve as a backdoor. With this view, we propose a new semi-prime factorization based on unique properties of Pythagorean triples with new mathematical theories and underlying patterns that are unexplored so far, with the purpose of advancing our research further in this direction.

3. Background Theory

This section provides the background theory that forms the mathematical foundations of this research work. Mathematical proofs, along with a summary of the definitions and a theorem with the supporting lemmas that are used for our proposed semi-prime factorization approach, are given below.

Let us consider the Brahmagupta–Fibonacci identity [14], which expresses the product of two prime sums of two squares as two sums of two squares in two different ways with the following mathematical representation:

$$p_1 p_2 = \left(x_{11}^2 + x_{12}^2\right)\left(x_{21}^2 + x_{22}^2\right) = (x_{11}x_{21})^2 + (x_{11}x_{22})^2 + (x_{12}x_{21})^2 + (x_{12}x_{22})^2 \quad (1)$$

$$p_1 p_2 = (x_{11}x_{21} \mp x_{12}x_{22})^2 + (x_{11}x_{22} \pm x_{12}x_{21})^2 \quad (2)$$

Jacobi [2] provides various sums of four squares for a particular number. Among these, Equation (1) is a special case in the set of possible $r_4(n)$ solutions.

$$r_4(n) = 8\sum_{d|n}(n), \qquad n \text{ is odd}. \tag{3}$$

Let us consider Lagrange's four-square theorem, also known as Bachet's conjecture [49], which states that every natural number can be represented as the sum of four integer squares with the following mathematical representation:

$$n = \sum_{k=1}^{4} y_k^2 \ , \quad y_k \text{ are integers such that } \quad n = y_1^2 + y_2^2 + y_3^3 + y_4^2$$

According to Legendre's three-square theorem [50], a natural number n can be represented as the sum of three squares of integers as follows:

$$n = \sum_{l=1}^{3} z_l^2, \quad z_l \text{ are integers iff } n \neq 4^a(8b+7) \text{ where } \quad a,\ b \text{ are integers}$$

From Lebesgue's identity [51], the square of the sum of four squares can be given by the sum of three squares and can be represented as follows:

$$n^2 = \left(\sum_{k=1}^{4} y_k^2\right)^2 = \sum_{l=1}^{3} z_l^2, \ \ n^2 = z_1^2 + z_2^2 + z_3^2 \tag{4}$$

$$n = y_1^2 + y_2^2 + y_3^2 + y_4^2, \ \ z_1 = y_1^2 + y_2^2 - y_3^2 - y_4^2,$$
$$z_2 = 2(y_1y_3 + y_2y_4), \quad z_3 = 2(y_1y_4 - y_2y_3)$$
$$\left(y_1^2 + y_2^2 + y_3^2 + y_4^2\right)^2 = \left(y_1^2 + y_2^2 - y_3^2 - y_4^2\right)^2 + (2(y_1y_3 + y_2y_4))^2 + (2(y_1y_4 - y_2y_3))^2$$

Definitions 1. *The definitions of Pythagorean triple and Pythagorean quadruple are given below [52].*

A Pythagorean triple is an ordered triple of distinct positive integers (α, β, N) with a mathematical notation as follows [48]:

$$\text{Pythagorean triple}(\alpha, \beta, N)\alpha^2 + \beta^2 = N^2, \alpha \neq \beta \tag{5}$$

A Pythagorean quadruple is an ordered quadruple of positive integers (z_1, z_2, z_3, N) expressed mathematically as follows:

$$\text{Pythagorean quadruple } (z_1, z_2, z_3,\ N)z_1^2 + z_2^2 + z_3^2 \ = N^2, \quad z_1, z_2, z_3 \text{ not equal} \tag{6}$$

In this paper, we prove the following theorem with some lemmas:

Theorem 1. *If a Pythagorean quadruple has a square of a triple and N is semi-prime, then N can be factored.*

Lemma 1. *The square product of two primes (each sum of two squares) is four sums of three squares.*

Lemma 2. *The square of a semi-prime is four sums of two squares.*

Lemma 3. *The greatest common divisor of all the two squares factors the semi-prime.*

Lemma 4. *One of the squares of a Pythagorean quadruple represents a Pythagorean triple.*

Lemma 1 produces the four Pythagorean quadruples. Lemma 2 reduces the four Pythagorean quadruples to two Pythagorean triples. Lemma 3 factors the semi-prime. Lemma 4 reduces a Pythagorean quadruple to a triple. Overall, the theorem shows that it is sufficient to only find such a Pythagorean quadruple to factor the semi-prime.

4. Proposed Method for Factoring Semi-Primes

The inherent hardness of finding the prime factors of large semi-primes forms the fundamental premise for RSA robustness [53]. However, this holds good until there is an efficient method found to compute the unknown prime factors of RSA keys, which are essentially large semi-prime numbers [54]. While several methods for integer factorization exist in general [55–57], none of them focus on semi-prime factorization relating to RSA cryptosystems. Hence, in this paper we will take this challenge and focus on solving the semi-prime factorization problem with a proposed reduced problem approach. Through the rest of the paper, it will be shown that the problem can be reduced to finding only one suitable sum of three squares to factorize a semi-prime. We provide a mathematical proof for our proposed approach in this section.

Proposed method: "The sum of four squares factorization" generates $r4n$ solutions [48] (Equation (3)), of which only one solution is applicable to the Brahmagupta–Fibonacci identity, namely Equations (1) and (2). The remaining solutions do not lead to solutions for the semi-prime factorization. Some simple case examples are given below.

Case Example 1. Let us consider N = 169.

$$169 = 13^2 = \left(2^2 + 3^2\right)\left(2^2 + 3^2\right) = 4^2 + 6^2 + 6^2 + 9^2 = 4^2 + 4^2 + 4^2 + 11^2$$

Applying the Brahmagupta–Fibonacci identity Equation (2), we obtain:

$$4^2 + 6^2 + 6^2 + 9^2 = (9-4)^2 + (6+6)^2 = 5^2 + 12^2 = (9+4)^2 + (6-6)^2 = 13^2 + 0^2$$

$$4^2 + 4^2 + 4^2 + 11^2 \neq (11-4)^2 + (4+4)^2 = 7^2 + 8^2 = 113 \neq 169$$

$$4^2 + 4^2 + 4^2 + 11^2 \neq (11+4)^2 + (4-4)^2 = 15 + 0^2 = 225 \neq 169$$

Case Example 2. Let us consider N = 377.

$$377 = (13)(29) = \left(2^2 + 3^2\right)\left(2^2 + 5^2\right) = 4^2 + 6^2 + 10^2 + 15^2$$

Applying the Brahmagupta–Fibonacci identity provides the two sums of squares.

$$377 = 4^2 + 6^2 + 10^2 + 15^2 = (15-4)^2 + (10+6)^2 = 11^2 + 16^2 = (15+4)^2 + (10-6)^2 = 19^2 + 4^2$$

$$377 = 4^2 + 19^2 = 11^2 + 16^2$$

Once the two sums of two squares are known, the prime factors can be found using a modified Euler factorization [1].

$$\Delta e = 16 - 4 = 12, \quad \Delta o = 19 - 11 = 8, \quad g = \gcd(8, 12) = 4, \quad p_1 = \left(\frac{8}{4}\right)^2 + \left(\frac{12}{4}\right)^2 = 13$$

However, in this case, there are nine sums of four squares, as follows:

$$(1,4,6,18),(1,6,12,14),(2,2,12,15),(2,6,9,16),(4,6,6,17)$$

$$(\mathbf{4},\mathbf{6},\mathbf{10},\mathbf{15}),(5,8,12,12),(6,6,7,16),(6,8,9,14)$$

Only one of these is applicable to the Brahmagupta–Fibonacci identity, providing the two sums of two squares. A faster method, using a modified binary greatest common divisor [14], quickly validates a sum of four squares given below:

$$(4,6,10,15)=15,\ 2(5,2,3)\Rightarrow(5,2),(2,3)\Rightarrow\left(2^2+5^2\right)\left(2^2+3^2\right)=(29)(13)=377 \quad (7)$$

This requires the sums of four squares to be found until the correct sum of four squares is factored. This is not viable for the factorization of large semi-primes. In this paper, it will be shown that it is sufficient to find only one suitable sum of three squares to factorize a semi-prime, and a theorem proof and four lemmas follow.

Restating Theorem 1. *If a Pythagorean quadruple has a square of a triple and N is semi-prime, then N can be factored. Some lemmas required are provided below.*

Lemma 1. *The square product of two primes (each sum of two squares) is four sums of three squares.*

Proof. Consider the Lebesgue identity. The square of the sum of four squares can be given by the sum of three squares. From Equation (4), we have:

$$n^2=\left(\sum_{k=1}^{4}y_k^2\right)^2=\sum_{l=1}^{3}z_l^2,\ \ n^2=z_1^2+z_2^2+z_3^2$$

and

$$c_1^2=a_1^2+b_1^2,\quad c_2^2=a_2^2+b_2^2,\quad n=c_1c_2,\quad n^2=(c_1c_2)^2=c_1^2c_2^2$$

$$c_1^2c_2^2=c_1^2\left(a_2^2+b_2^2\right)=(a_2c_1)^2+(b_2c_1)^2 \quad (8)$$

$$c_1^2c_2^2=c_2^2\left(a_1^2+b_1^2\right)=(a_1c_2)^2+(b_1c_2)^2 \quad (9)$$

The four sums of three squares are thus:

$$(a_2c_1)^2+(b_2c_1)^2=(a_2c_1)^2+(b_2a_1)^2+(b_2b_1)^2 \quad (10)$$

$$(a_2c_1)^2+(b_2c_1)^2=(a_2a_1)^2+(a_2b_1)^2+(b_2c_1)^2 \quad (11)$$

$$(a_1c_2)^2+(b_1c_2)^2=(a_1c_2)^2+(b_1a_2)^2+(b_1b_2)^2 \quad (12)$$

$$(a_1c_2)^2+(b_1c_2)^2=(a_1a_2)^2+(a_1b_2)^2+(b_1c_2)^2 \quad (13)$$

The four Pythagorean quadruples from Equations (9)–(12) are given as:

$$(a_1b_2,\ b_1b_2,c_1a_2,\ c_1c_2),\ \ (a_1a_2,\ b_1a_2,\ c_1b_2,c_1c_2),\ \ (a_1c_2,b_1a_2,b_1b_2,c_1c_2)\ \ (a_1a_2,\ a_1b_2,b_1c_2,\ c_1c_2) \quad (14)$$

□

Lemma 2. *The square of the semi-prime is four sums of two squares.*

Proof: We have the following product of two sums of squares:

$$(c_1c_2)^2=c_1^2c_2^2=\left(a_1^2+b_1^2\right)^2\left(a_2^2+b_2^2\right)^2$$

Using the Brahmagupta–Fibonacci identity, we obtain:

$$c_1^2 = \left(a_1^2\right)^2 + (a_1b_1)^2 + (a_1b_1)^2 + \left(b_1^2\right)^2 = \left(a_1^2 - b_1^2\right)^2 + (2a_1b_1)^2 = \left(a_1^2 + b_1^2\right)^2$$

$$c_2^2 = \left(a_2^2\right)^2 + (a_2b_2)^2 + (a_2b_2)^2 + \left(b_2^2\right)^2 = \left(a_2^2 - b_2^2\right)^2 + (2a_2b_2)^2 = \left(a_2^2 + b_2^2\right)^2$$

$$c_1^2 = \left(a_1^2 - b_1^2\right)^2 + (2a_1b_1)^2 = \left(a_1^2 + b_1^2\right)^2,\ c_2^2 = \left(a_2^2 - b_2^2\right)^2 + (2a_2b_2)^2 = \left(a_2^2 + b_2^2\right)^2 (c_1c_2)^2 = \left[\left(a_1^2 + b_1^2\right)\left(a_2^2 + b_2^2\right)\right]^2$$

$$c_1c_2 = (a_1a_2)^2 + (b_1a_2)^2 + (a_1b_2)^2 + (b_1b_2)^2$$

$$= (a_1a_2 - b_1b_2)^2 + (a_1b_2 + b_1a_2)^2 = (a_1a_2 + b_1b_2)^2 + (a_1b_2 - b_1a_2)^2$$

$$(c_1c_2)^2 = \left[(a_1a_2 - b_1b_2)^2 + (a_1b_2 + b_1a_2)^2\right]\left[(a_1a_2 + b_1b_2)^2 + (a_1b_2 - b_1a_2)^2\right]$$

$$\begin{aligned}(c_1c_2)^2 &= ((a_1a_2 - b_1b_2)(a_1a_2 + b_1b_2))^2 + ((a_1a_2 - b_1b_2)(a_1b_2 - b_1a_2))^2 \\ &+((a_1b_2 + b_1a_2)(a_1a_2 + b_1b_2))^2 + ((a_1b_2 + b_1a_2)(a_1b_2 - b_1a_2))^2\end{aligned}$$

$$\begin{aligned}(c_1c_2)^2 &= ((a_1a_2 - b_1b_2)(a_1a_2 + b_1b_2) + (a_1b_2 + b_1a_2)(a_1b_2 - b_1a_2))^2 \\ &+((a_1a_2 - b_1b_2)(a_1b_2 - b_1a_2) - (a_1b_2 + b_1a_2)(a_1a_2 + b_1b_2))^2\end{aligned}$$

$$\begin{aligned}(c_1c_2)^2 &= ((a_1a_2 - b_1b_2)(a_1a_2 + b_1b_2)) - ((a_1b_2 + b_1a_2)(a_1b_2 - b_1a_2))^2 \\ &+((a_1a_2 - b_1b_2)(a_1b_2 - b_1a_2)) + ((a_1b_2 + b_1a_2)(a_1a_2 + b_1b_2))^2 \\ &+((a_1a_2 - b_1b_2)(a_1b_2 - b_1a_2)) + ((a_1b_2 + b_1a_2)(a_1a_2 + b_1b_2))^2\end{aligned} \tag{15}$$

We make use of Equations (7) and (8), and we have the four sums of two squares for $(c_1c_2)^2$, as follows:

$$\begin{aligned}&((a_1a_2 - b_1b_2)(a_1a_2 + b_1b_2) + (a_1b_2 + b_1a_2)(a_1b_2 - b_1a_2))^2 \\ &+ ((a_1a_2 - b_1b_2)(a_1b_2 - b_1a_2) - (a_1b_2 + b_1a_2)\ (a_1a_2 + b_1b_2))^2\end{aligned} \tag{16}$$

$$\begin{aligned}&((a_1a_2 - b_1b_2)(a_1a_2 + b_1b_2) - (a_1b_2 + b_1a_2)(a_1b_2 - b_1a_2))^2 \\ &+ ((a_1a_2 - b_1b_2)(a_1b_2 - b_1a_2) + (a_1b_2 + b_1a_2)(a_1a_2 + b_1b_2))^2.\end{aligned} \tag{17}$$

Two of the four Pythagorean triples from Equations (18) and (19) have no common factors. The other two of the four Pythagorean triples, which are factorable from Equations (7) and (8), are given by:

$$(c_1a_2,\ c_1b_2, c_1c_2),\ \ (c_2a_1,\ c_2b_1, c_1c_2) \tag{18}$$

□

Lemma 3. *The greatest common divisors of the sums of two squares factor the semi-prime.*

Proof. Consider the semi-prime $n = c_1c_2$ and assume that $c_1, c_2 = 1 \pmod 4$, i.e., c_1 and c_2 are the sum of two squares, then we have:

$$1(mod\ 4) \equiv c = x_1^2 + x_2^2$$

From Equation (14), we have:

$$(c_1a_2,\ c_1b_2,\ c_1c_2),\ (c_2a_1,\ c_2b_1, c_1c_2)$$

$$(c_1a_2,\ c_1b_2,\ c_1c_2) \Rightarrow \gcd(c_1a_2,\ c_1b_2) = c_1,\ (c_2a_1,\ c_2b_1,\ c_1c_2) \Rightarrow \gcd(c_2a_1,\ c_2b_1) = c_2$$

The greatest common divisors of the sums of two squares are the factors of the semi-prime.

$$(\alpha, \beta, N),\ \ c_1 = \gcd(\alpha, \beta),\ \ c_2 = \frac{N}{c_1} \tag{19}$$

□

Lemma 4. *There exists a Pythagorean quadruple, a square of which represents a triple.*

Proof. From Pythagorean quadruple Equation (6) given by $N^2 = z_1^2 + z_2^2 + z_3^2$ and Pythagorean triple Equation (5) given by $N^2 = \alpha^2 + \beta^2,\ \alpha \neq \beta$ we have the following:

$$\text{If } \beta \in \{z_1, z_2, z_3\}\ \exists\ (\alpha, \beta, N)\ : \alpha^2 = N^2 - \beta^2,\ \alpha \neq \beta \tag{20}$$

From Equation (13), it can be seen that:

$$(a_1b_2,\ b_1b_2, c_1a_2,\ c_1c_2) \Rightarrow \beta_1 = c_1a_2 \Rightarrow$$

$$\alpha_1, \beta_1,\ c_1c_2) = (\alpha_1, c_1a_2,\ c_1c_2) \Rightarrow \alpha_1 = c_1b_2$$

$$(a_1a_2,\ b_1a_2,\ c_1b_2, c_1c_2) \Rightarrow \beta_2 = c_1b_2 \Rightarrow$$

$$(\alpha_2, \beta_2,\ c_1c_2) = (\alpha_2, c_1b_2,\ c_1c_2) \Rightarrow \alpha_2 = c_1a_2$$

$$(a_1c_2, b_1a_2, b_1b_2, c_1c_2) \Rightarrow \beta_3 = c_2a_1 \Rightarrow\ (\alpha_3, \beta_3,\ c_1c_2) = (\alpha_3, c_2a_1,\ c_1c_2) \Rightarrow \alpha_3 = c_2b_1$$

$$(a_1a_2,\ a_1b_2, b_1c_2,\ c_1c_2) \Rightarrow \beta_4 = c_2b_1 \Rightarrow\ (\alpha_4, \beta_4,\ c_1c_2) = (\alpha_4, c_2b_1,\ c_1c_2) \Rightarrow \alpha_4 = c_2a_1$$

$$(\beta_1, \alpha_1,\ c_1c_2) = (\alpha_2, \beta_2,\ c_1c_2) = (c_1a_2,\ c_1b_2,\ c_1c_2),\ (\beta_3, \alpha_3,\ c_1c_2) =$$

$$(\alpha_4, \beta_4,\ c_1c_2) = (c_2a_1,\ c_2b_1,\ c_1c_2)$$

$$\beta \in \{c_1b_2, c_1a_2, c_2b_1, c_2a_1\} \exists\ (\alpha, \beta, N)\ : \alpha^2 = N^2 - \beta^2,\ \alpha \neq \beta$$

□

Theorem 2. *If a Pythagorean quadruple has a square of a triple, the semi-prime can be factored.*

Proof. From Lemma 1, we have:

The four Pythagorean quadruples from Equation (13) are given as:

$$(a_1b_2,\ b_1b_2, c_1a_2,\ c_1c_2),\ (a_1a_2,\ b_1a_2,\ c_1b_2, c_1c_2),$$

$$(a_1c_2, b_1a_2, b_1b_2, c_1c_2), (a_1a_2,\ a_1b_2, b_1c_2,\ c_1c_2)$$

From Lemma 4, there exists a Pythagorean quadruple, a square of which represents a triple.

$$\beta \in \{c_1b_2, c_1a_2, c_2b_1, c_2a_1\} \exists\ (\alpha, \beta, N)\ : \alpha^2 = N^2 - \beta^2,\ \alpha \neq \beta \tag{21}$$

From Lemma 2, the Pythagorean triples with prime factors from Equation (14) are given as follows:

$$(c_1a_2,\ c_1b_2,\ c_1c_2),\ (c_2a_1,\ c_2b_1, c_1c_2)$$

$$N = c_1c_2,\ \exists\ (\alpha, \beta, N) : (\alpha, \beta) \in \{(a_2c_1,\ b_2c_1),\ (a_1c_2,\ b_1c_2)\} \tag{22}$$

From Lemma 3, we have:

$$(\alpha, \beta, N),\ c_1 = \gcd(\alpha, \beta),\ c_2 = \frac{N}{c_1}$$

□

4.1. Algorithm of Our Proposed Semi-Prime Factorization

Cryptographic algorithms make use of standard integer factorization algorithms found in the literature, such as Pollard's factoring algorithm, Lenstra's elliptic curve factorization algorithm, and others with different kinds of number sieves [58,59]. For our algorithm to factor semi-primes, we capitalize on the properties of semi-prime representations as the sum of three squares, which is supported by well-proven theorems [60]. Based on the theoretical background postulated in this work, our algorithm consists of four key steps, as given below:

Step 1. Square the semi-prime to be factored: N^2.

Step 2. Find sums of three squares such that:

$$\left(N^2 - x^2\right) = prime \equiv 1 \; mod \; 4 = \beta^2 + a^2$$

//Increment x until a prime is found congruent to 1 mod 4 [61]
{x+
$+x^2$
$N^2 - x^2$ }
Do primality test, 1 mod 4 test, sum of two squares}

Step 3. Save $\{x,\ a,\ b\}$ such that $(N, x, a, b), \beta:\ \beta \in \{x, a, b\}$
Test β to find α such that $\alpha^2 = N^2 - \beta^2$
Step 3.1 Test $\{x,\ a,\ b\}$ to find $\{\alpha, \beta\}$, $x^2, a^2,\ b^2; N^2 - x^2,\ N^2 - a^2,\ N^2 - b^2$
Step 3.2 Test for perfect square // use a square root function)

Step 4. Calculate and output the semi-prime factors c_1 and c_2 such that $N = c_1 c_2$
Step 4.1 $c_1 = \gcd(\alpha, \beta)$ // use gcd function
Step 4.2 $c_2 = \frac{N}{c_1}$

4.2. Complexity, Comparison and Constraints of Our Algorithm

We summarize the complexity of our proposed semi-prime factorization algorithm in terms of memory and computational time. These complexity measures are followed in similar lines to existing methods reported in the literature [61]. The memory requirement of our algorithm is very minimal, with most computations operating on the memory variables (N, x, a, b, α, β), which use BigInteger arithmetic. In terms of time complexity, our method compares favorably against Fermat's factorization method, in that only one solution exists in Fermat's method. As was shown in Case Example 2, our method has identified 45 solutions, of which 11 could lead directly to a factorization and the others may as well, if their sums of squares are part of a tree that reveals other sums of squares.

In our algorithm, the initial step of squaring the number to be factorized could be a constraint for large semi-prime numbers. Further, the method is also probabilistic, while in many ways it is comparable to the stochastic nature of finding a sum of two squares. However, the advantage of our algorithm is that there are many possible solutions in the set of the sum of three squares, making it more likely that such a solution also leads towards finding a sum of two squares. Moreover, once a sum of three squares is known, the squares themselves form trees, which will be explored in future work. For instance, in Case Example 2 discussed earlier, 45 possible sums of squares were obtained. Among these, 11 created direct solutions to the factorization of the semi-prime and our method provides more possibilities of the solution space. Otherwise, using common approaches such as Fermat's, only two sums of squares exist for the semi-prime. Therefore, using our method, the probability of finding a solution is greatly enhanced and, in this case, it is more likely by an order of six times than the common methods. Further, traversing the tree of squares, if any of the 45 possible solutions are discovered, then any of these may lead to a factorization solution, which is 22 times more likely to lead to a solution. These trees provide additional sums of three squares, some of which will lead to a sum of two squares. Once a sum of two squares is known, the search for the second sum of squares is

contained in a subset, thereby reducing the search space via the efficient use of polynomials to factorize semi-primes quickly [1].

5. Case Study Examples Applied to RSA Key Factorization

In this section, we illustrate the application of our proposed semi-prime factorization method using case study examples. An application area of particular interest in considering specific variants of RSA is how both small and large encryption keys perform within our proposed factorization approach. With the evolution of the Internet of things (IoT), the emergence of lightweight cryptography is on the rise. However, due to the relatively low computational power of personal devices, malicious attacks are recently targeting IoT networks [62–64]. Since the security of cryptographic operations in both small keys as well as large keys depends upon whether the semi-prime factorization can be solved efficiently, we consider one application example for each of the two scenarios.

Application Example 1. Let us consider Case Example 2 with N = 377.

The 45 Pythagorean quadruples are:
(12,81,368,377), (12,108,361,377), (12,156,343,377), (12,224,303,377), (15,252,280,377), (17,144,**348**,377), (17,192,324,377), (24,143,**348**,377), (24,177,332,377), (28,72,369,377), (28,252, 279,377), (39,72,368,377), (39,208,312,377), (44,207,312,377), (44,228,297,377), (64,252,273,377), (72,199,312,377), (72,252,271,377), (81,108,**352**,377), (84,132,343,377), (84,208,303,377), (87,116, **348**,377), (87,156,332,377), (87,172,324,377), (89,192,312,377), (100,105,**348**,377), (100,240,**273**, 377), (105,252,**260**,377), (108,207,296,377), (108,233,276,377), (116,177,312,377), (116,192,303, 377), (132,224,**273**,377), (143,156,312,377), (143,228,264,377), (144,172,303,377), (**145**,240,252, 377), (**152**,207,276,377), (156,172,297,377), (156,208,**273**,377), (156,233,252,377), (172,207,264, 377), (177,180,280,377), (180,215,252,377), (192,208,249,377)
$\beta \in$ {12, 15, 17, 24, 28, 39, 44, 64, 72, 81, 84, 87, 89, 100, 105, 108, 116, 132, 143, 144, **145**, **152**, 156, 172, 177, 180, 192, 207, 208, 215, 224, 228, 233, 240, 249, 252, **260**, 264, 271, **273**, 276, 279, 280, 296, 297, 303, 312, 324, 332, 343, **348**, **352**, 361, 368, 369}
From Lemma 4 and Equation (20) we have:

$$\beta \in (z_1, z_2, z_3)\ \exists\ (\alpha, \beta, N)\ : \alpha^2 = N^2 - \beta^2,\ \alpha \neq \beta$$

The four Pythagorean triples are:
(***135***,352,377), (145,348,377), (152,***345***,377), (260,273,377)
Note that:

$$\exists \alpha : \{\alpha\} \nsubseteq \{\beta\}\ : \{135,\ 345\}$$

$\nexists\{\beta\}$ with $\gcd(\alpha, \beta) = 1$ are not solutions to the semi-prime. However, solutions can be found as follows [1,14]:

$$\{\alpha\} \nsubseteq \{\beta\} \Rightarrow \gcd(\alpha, \beta) = 1,$$

$$(135, 352, 377) \Rightarrow \gcd(135, 352) = 1,$$

$$x_1 = \sqrt{\left(\frac{377 - 135}{2}\right)} = 11,\ x_2 = \sqrt{377 - 11^2} = 16$$

$$377 = 11^2 + 16^2$$

$$(152, 345, 377) \Rightarrow \gcd(152, 345) = 1$$

$$x_3 = \sqrt{\left(\frac{377 - 345}{2}\right)} = 4,\ x_4 = \sqrt{377 - 4^2} = 19$$

$$377 = 4^2 + 19^2$$

$$o = 19 - 11 = 8,\ e = 16 - 4 = 12,\ g = \gcd(8, 12) = 4$$

$$p_1 = \left(\frac{o}{g}\right)^2 + \left(\frac{e}{g}\right)^2 = \left(\frac{8}{4}\right)^2 + \left(\frac{12}{4}\right)^2 = 2^2 + 3^2 = 13$$

$$p_2 = \frac{N}{p_1} = \frac{377}{13} = 29$$

From Lemma 3 and Equation (15) we have:

$$(\alpha, \beta, N),\ c_1 = \gcd(\alpha, \beta),\ c_2 = \frac{N}{c_1}$$

$$\{\alpha\} \cap \{\beta\} \Rightarrow \gcd(\alpha, \beta) = c_1$$

$$(145, 348, 377) \Rightarrow \gcd(145, 348) = 29,\ \frac{377}{29} = 13$$

$$(260, 273, 377) \Rightarrow \gcd(260, 273) = 13,\ \frac{377}{13} = 29$$

$$N = c_1 c_2 = 377 = (13)(29)$$

From Lemma 2 we have:
The Pythagorean triples with prime factors from Equation (14) are given as follows:

$$(c_1 a_2,\ c_1 b_2,\ c_1 c_2),\ (c_2 a_1,\ c_2 b_1, c_1 c_2)$$

$$(a_1,\ b_1, c_1) = (5, 12, 13),\ (a_2, b_2,\ c_2) = (20, 21, 29)$$

$$(c_1 a_2,\ c_1 b_2,\ c_1 c_2) = (13 * 20,\ 13 * 21,\ 377) = (260,\ 273,\ 377) \Rightarrow \gcd(260,\ 273) = 13$$

$$(c_2 a_1,\ c_2 b_1, c_1 c_2) = (29 * 5,\ 29 * 12,\ 377) = (145,\ 348,\ 377) \Rightarrow \gcd(145,\ 348) = 29$$

Figure 1 provides a pictorial representation of Pythagorean triples with prime factors for the case example with $N = 377$.

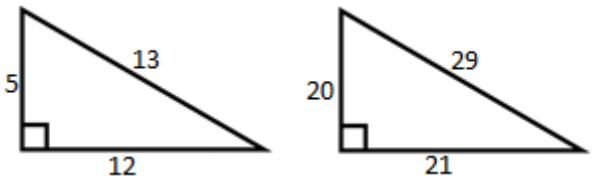

Figure 1. An illustration of Pythagorean triples.

Application Example 2. In computer security applications, the aim of a cryptosystem is to encrypt a message before communication, and only the authenticated user that has the right key can be able to decrypt the message, thereby enforcing message integrity, confidentiality, authentication, and privacy. Among several ciphers developed for this purpose, RSA is widely used in key exchange, digital signatures, and small blocks of data encryption. For the second application example, we consider RSA where the key is derived from a very large number (a semi-prime). Since determining the two prime factors of the large number is computationally difficult, research studies on evaluating the RSA and the factorization schemes are gaining attention [65–67]. Hence, we describe how our proposed semi-prime factorization method can be applied to factorize RSA quickly.

The steps involved in RSA encryption are given below:

1. Pick two large prime numbers c_1 and c_2;
2. Consider the semi-prime $N = c_1 c_2$;
3. Consider $\varphi(N) = (c_1 - 1) * (c_2 - 1)$;
4. Choose an integer e such that $1 < e < \varphi(N)$ and $\gcd(e,\ \varphi(N)) = 1$;
5. Compute d such that $de \equiv 1 \pmod{\varphi(N)}$.

While the public key (N, e) is distributed for encrypting a message, the private key (N, d) is kept a secret for decrypting the message only by an authorized entity. Hence,

based on this RSA scheme, it is evident that with the factorization of N, we can compute $de \equiv 1 \pmod{\varphi(N)}$ [68], which gives the private key (N, d). When the factorization is efficient, RSA can be broken. The factoring challenge was introduced to identify the safety limits of the key length to be used for the RSA encryption algorithm that can ensure information security. Hence, researchers focus on mathematically proving the cryptanalytic strength using efficient RSA factorization methods. Table 1 demonstrates the application of our proposed semi-prime factorization method for a key length of 768 decimal digits, denoted as RSA-768. The sums of squares and polynomials have been explored for semi-prime factorization in previous research works [1,46,69]. However, there are more than 50 properties of Pythagorean triples that have been reported and new patterns yet to be explored [15,70]. In this research work, the proposed method has applied such unique patterns unexplored so far in the literature and maintains the algorithm's order of computational complexity, similar to the existing approaches that have been evaluated and reported recently [71,72]. A recent experimental study employed the representation of primes in the form $p = 6 \cdot x \pm 1$ and applied the theory to the RSA factorization problem [73]. Another work shows the decomposition of the two prime numbers with the Pisano period factorization method, which has been proven to be a subexponential complexity method [74]. Several integer factorization methods have also suggested direct application to cryptanalysis of RSA by applying different genetic algorithms [75]. While genetic algorithms could be a promising avenue of research for integer factorization, they are computationally complex. This paper's focus to involve new simple arithmetic operations while exploring unique structures of prime numbers has resulted in an efficient factorization.

Table 1. Application of the proposed semi-prime factorization method for RSA-768 attack.

c_1c_2 RSA768	123018668453011775513049495838496272077285356959533479219732245215 172640050726365751874520219978646938995647494277406384592519255732 630345373154826850791702612214291346167042921431160222124047927473 77940806653514195974598569021434 13
$(c_1c_2)^2$	151335927879520346290803999457322598363508796074958341058717144380 245872835312747521375274637572332720319718269519140130366434717995 557448975805325285592901195789698493475842912499609903738365207236 511522080338199512854710820317535056362120218189196094883472059867 670273711218840191206761008283107936158558105422755888818703810974 813042919826949446811902964522404849739464632596646341875606194756 29854679990130064794624846965113725044885716357780585197936192885 69
β	122282464059209245992171436897658189706307595059665852419459721623 575268394641485120055721728993778585098737802795637062458727992772 215791535778942916633321748296084895133564298582155097077049551048 78914615041542203627809725953681 88
$\sqrt[\alpha]{(c_1c_2)^2 - \beta^2}$	134384437923531027642685110944406395128150457221349204815473265992 860466317855675453545164265884545802321235838004890967990322730328 639778020385449818260083476767779296347341551505950443482805814798 59356430564624241138955984428923 5
$c_1\gcd(\alpha, \beta)$	33478071698956898786044169848212690817704794983713768568912431388 9 828837938780022876147116525317430877378144679994 89
$c_2 \frac{\text{RSA768}}{c_1}$	367460436667995904282446337996279526322791581643430876426760322838 157396665112792333734171433968102700927987363089 17

In our computationally simple method, the need to square the number that is to be factorized could be considered a constraint for large semi-primes to be attacked quickly. An example for RSA-768 is given below, illustrating our method of squaring the number to achieve semi-prime factorization. Future work will be devoted towards how the inherent mathematical constraint can be overcome by reducing the solution search space. One method is to use low order prime multipliers congruent to 1 mod 4 (for example

5,13,50) to increase the likelihood of finding sums of three squares (and two squares) without the complexity of first squaring the number to be factored. An illustration of the complexity is given below. The likelihood of finding a solution, β, is sparse and becomes less likely for larger semi-primes, as shown below. Future work should explore reducing the search space implied by squaring of the semi-prime to be factored.
$c_1c_2\text{RSA768}(c_1c_2)^2\beta\alpha\sqrt{(c_1c_2)^2-\beta^2}c_1\gcd(\alpha,\beta)c_2\frac{\text{RSA768}}{c_1}$

6. Conclusions and Future Research

In this paper, we proposed a new method for semi-prime factorization and emphasized its key contribution in the context of information security underpinned by the RSA cryptosystem of the current digital world. Some new ideas have resulted in a breakthrough of factoring the RSA-129 challenge number, but these were possible only after several years. Our novel method follows with the proof that the sum of fours squares of a semi-prime c_1c_2 has many solutions, but only one solution leads to factorization. The validation is fast, and the method uses a binary greatest common division approach with simple arithmetic operations to find the sum of two squares of one (or both) of the prime factors.

The sum of two squares has only two solutions and both are valid, though hard to find. Once these are known, previous work has proved that a modified Euler factorization can easily determine the prime factorization [1]. This paper was enhanced further by considering the sum of three squares, which has many solutions. However, the semi-prime must first be squared, resulting in larger numbers required to be processed. This is offset by the abundance of suitable solutions, leading to factorization successfully without affecting the order of computational complexity. The algorithm and the case examples have demonstrated the simplicity of our proposed method and its enhanced solution space, as compared to Fermat's method. The complexity of our proposed approach was demonstrated using numerical illustrations, including the real-time factorization of the 768-bit number RSA-768.

It is noted that for extremely large semi-primes, the search space may be constrained with the need to square the semi-prime. One approach to address this is highlighted and forms the key motivation for future research. In this context, one of the properties of semi-primes that forms a motivation for future research is given as follows: once the sum of three squares is known, the squares themselves form trees. Hence, reducing the solution search space of these trees for such cases, using our earlier associated research work, will be quite promising to explore.

Author Contributions: Conceptualization, A.O.; resources, A.O. and S.V.; writing—original draft preparation, A.O. and S.V.; writing—review and editing, A.O. and S.V.; supervision, S.V. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not Applicable.

Informed Consent Statement: Not Applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Overmars, A.; Venkatraman, S. A Fast Factorisation of Semi-Primes Using Sum of Squares. *Math. Comput. Appl.* **2019**, *24*, 62. [CrossRef]
2. Moreno, C.J.; Wagstaff, S.S. *Sums of Squares of Integers*, 1st ed.; Chapman and Hall/CRC Press: New York, NY, USA, 2005.
3. Dunn, A.; Zaharescu, A. Sums of Kloosterman Sums Over Primes in an Arithmetic Progression. *Q. J. Math.* **2019**, *70*, 319–342. [CrossRef]
4. Erdós, P. On the Normal Number of Prime Factors of P-1 and Some Related Problems Concerning Euler's Ø-Function. *Q. J. Math.* **1935**, *os-6*, 205–213. [CrossRef]
5. Pollard, J. Theorems on factorization and primality testing. *Proc. Camb. Philos. Soc.* **1974**, *76*, 521–528. [CrossRef]

6. Traversa, F.L.; di Ventra, M. Polynomial-time solution of prime factorization and NP-complete problems with digital memcomputing machines. *Chaos Interdiscip. J. Nonlinear Sci.* **2017**, *27*, 023107. [CrossRef] [PubMed]
7. Malapert, A.; Provillard, J. Puzzle—Solving the n-Fractions Puzzle as a Constraint Programming Problem. *INFORMS Trans. Educ.* **2018**, *19*, 48–55. [CrossRef]
8. Rescorla, E. *SSL and TLS: Designing and Building Secure Systems*; Addison-Wesley Reading: London, UK, 2001.
9. Schneier, B. *Applied Cryptography*, 2nd ed.; John Wiley & Sons, Inc.: New York, NY, USA, 1996.
10. Rivest, R.L.; Shamir, A.; Adleman, L. A method for obtaining digital signatures and public-key cryptosystems. *Commun. ACM* **1978**, *21*, 120–126. [CrossRef]
11. Sun, H.-M.; Wu, M.-E.; Ting, W.-C.; Hinek, M. Dual RSA and Its Security Analysis. *IEEE Trans. Inf. Theory* **2007**, *53*, 2922–2933.
12. McKee, J.F. Turning Euler's Factoring Method into a Factoring Algorithm. *Bull. Lond. Math. Soc.* **1996**, *28*, 351–355. [CrossRef]
13. Zagier, D. A One-Sentence Proof That Every Prime p $\equiv$ 1 (mod 4) Is a Sum of Two Squares. *Am. Math. Mon.* **1990**, *97*, 144.
14. Li, S. *The Sum of Two Squares*; Cornell University Press: New York, NY, USA, 2013.
15. Agarwal, R.P. Pythagorean Triples before and after Pythagoras. *Computation* **2020**, *8*, 62. [CrossRef]
16. Boneh, D. Twenty years of attacks on the RSA cryptosystem. *Not. Am. Math. Soc. (AMS)* **1999**, *46*, 203–213.
17. Valenta, L.; Cohney, S.; Liao, A.; Fried, J.; Bodduluri, S.; Heninger, N. Factoring as a Service. In *Financial Cryptography and Data Security. FC 2016. Lecture Notes in Computer Science*; Grossklags, J., Preneel, B., Eds.; Springer: Berlin, Germany, 2017; Volume 9603.
18. Durumeric, Z.; Kasten, J.; Bailey, M.; Halderman, J.A. Analysis of the HTTPS certificate ecosystem. In Proceedings of the 13th Internet Measurement Conference, Barcelona, Spain, 23–25 October 2013.
19. Wiener, M. Cryptanalysis of short RSA secret exponents. *IEEE Trans. Inf. Theory* **1990**, *160*, 553–558. [CrossRef]
20. Coppersmith, D. Solving homogeneous linear equations over GF(2) via block Wiedemann algorithm. *Math. Comput.* **1994**, *62*, 333–350. [CrossRef]
21. Blömer, J.; May, A. New Partial Key Exposure Attacks on RSA. In *Crypto 2003, LNCS*; Springer: Berlin, Germany, 2003; pp. 27–43.
22. Boneh, D.; Durfee, G. Cryptanalysis of RSA with Private Key D Less than N^0.292. In Proceedings of the International Conference on the Theory and Application of Cryptographic Techniques, Prague, Czech Republic, 2–6 May 1999; Springer: Berlin, Germany, 1999; Volume 1592, p. 111.
23. Heninger, N.; Durumeric, Z.; Wustrow, E.; Halderman, J.A. Mining your Ps and Qs: Detection of widespread weak keys in network devices. In Proceedings of the 21st USENIX Security Symposium, Bellevue, WA, USA, 8–10 August 2012.
24. Adrian, D.; Bhargavan, K.; Durumeric, Z.; Gaudry, P.; Green, M.; Halderman, J.A.; Heninger, N.; Springall, D.; Thomé, E.; Valenta, L.; et al. Imperfect forward secrecy: How Diffie-Hellman Fails in Practice. In Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, Denver, CO, USA, 12–16 October 2015; pp. 5–17.
25. Nemec, M.; Sys, M.; Svenda, P.; Klinec, D.; Matyas, V. The Return of Coppersmith's Attack: Practical Factorization of Widely Used RSA Moduli. In Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, CCS 2017, Dallas, TX, USA, 30 October–3 November 2017; pp. 1631–1648.
26. Lehmer, D.H.; Powers, R.E. On Factoring Large Numbers. *Bull. Am. Math. Soc.* **1931**, *37*, 770–776. [CrossRef]
27. Morrison, M.A.; Brillhart, J. A Method of Factoring and the Factorization of F7. *Math. Comput. Am. Math. Soc.* **1975**, *29*, 183–205.
28. Pomerance, C.; Wagstaff, S.S. Implementation of the Continued Fraction Integer Factoring Algorithm. *Congr. Numer.* **1983**, *37*, 99–118.
29. Pollard, J.M. A Monte Carlo method for factorization. In *BIT Numerical Mathematics*; Springer: Berlin, Germany, 1975; Volume 15, pp. 331–334.
30. Pomerance, C. The Quadratic Sieve Factoring Algorithm. In *Advances in Cryptology: EUROCRYPT'84*; Springer: Berlin, Germany, 1985; pp. 169–182.
31. Kameswari, P.A.; Jyotsna, L. An Attack Bound for Small Multiplicative Inverse of φ(N) mod e with a Composed Prime Sum p + q Using Sublattice Based Techniques. *Cryptography* **2018**, *2*, 36. [CrossRef]
32. Kamel Ariffin, M.R.; Abubakar, S.I.; Yunos, F.; Asbullah, M.A. New Cryptanalytic Attack on RSA Modulus N = pq Using Small Prime Difference Method. *Cryptography* **2019**, *3*, 2. [CrossRef]
33. Lenstra, A.K.; Lenstra Jr, H.W.; Manasse, M.S.; Pollard, J.M. *The Number Field Sieve*; Springer: Berlin, Germany, 1993.
34. Cheng, Q. A New Special-Purpose Factorization Algorithm. Citeseer. 2002. Available online: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.8.9071&rep=rep1&type=pdf (accessed on 18 January 2021).
35. Sedlacek, V.; Klinec, D.; Sys, M.; Svenda, P.; Matyas, V. I Want to Break Square-free: The 4p 1 Factorization Method and Its RSA Backdoor Viability. In Proceedings of the 16th International Joint Conference on e-Business and Telecommunications (ICETE 2019), Prague, Czech Republic, 26–28 July 2019; pp. 25–36.
36. Grosswald, E. *Representations of Integers as Sums of Squares*; Springer: Berlin, Germany, 1985.
37. Northshield, S. A Short Proof of Fermat's Two-square Theorem. *Am. Math. Mon.* **2020**, *127*, 638.
38. Jackson, T. *From Polynomials to Sums of Squares*; CRC Press: New York, NY, USA, 1995.
39. Dickson, L.E. *History of the Theory of Numbers: Diophantine Analysis*, 2nd ed.; Dover Publications: New York, NY, USA, 2005.
40. Roy, T.; Soni, F.J. A direct method to generate Pythagorean triples and its generalization to Pythagorean quadruples and n-tuples. *arXiv* **2012**, arXiv:1201.2145.
41. Christopher, A.D. A partition-theoretic proof of Fermat's Two Squares Theorem. *Discret. Math.* **2016**, *339*, 1410–1411. [CrossRef]
42. Knill, O. Some experiments in number theory. *arXiv* **2016**, arXiv:1606.05971.

43. Kostopoulos, G.L. An Original Numerical Factorization Algorithm. *J. Inf. Assur. Cyber Secur.* **2016**, *2016*, 775081. [CrossRef]
44. Kaddoura, I.; Abdul-Nabi, S.; Al-Akhrass, K. New Formulas for Semi-Primes. Testing, Counting and Identification of the nth and next Semi-Primes. *arXiv* **2016**, arXiv:1608.05405.
45. Hiary, G.A. A Deterministic Algorithm for Integer Factorization. *Math. Comput.* **2016**, *85*, 2065–2069. [CrossRef]
46. Overmars, A.; Venkatraman, S. Mathematical Attack of RSA by Extending the Sum of Squares of Primes to Factorize a Semi-Prime. *Math. Comput. Appl.* **2020**, *25*, 63. [CrossRef]
47. McKee, J.F. Speeding Fermat's factoring method. *Math. Comput.* **1999**, *68*, 1729–1737. [CrossRef]
48. Overmars, A.; Ntogramatzidis, L.; Venkatraman, S. A New approach to generate all Pythagorean triples. *AIMS Math.* **2019**, *4*, 242–253. [CrossRef]
49. Boucard, J. Lagrange and the four-square theorem. *Lett. Mat.* **2014**, *2*, 59–66. [CrossRef]
50. Dickson, L.E. *History of the Theory of Numbers*; Carnegie Institute of Washington 1919; AMS Chelsea Publishing: Providence, RI, USA, 1992; Volume II, p. 15.
51. Fenster, D.D. Leonard Dickson's History of the theory of numbers: An historical study with mathematical implications. *J. Hist. Math.* **1999**, *5*, 159–179.
52. Mitchell, D.W. An alternative characterisation of all Primitive Pythagorean Triples. *Math. Gaz.* **2001**, *85*, 273–275. [CrossRef]
53. Venkatraman, S.; Overmars, A. New method of prime factorisation based attacks on RSA Authentication in IoT. *Cryptography* **2019**, *3*, 20. [CrossRef]
54. Da Silva, J.C.L. Factoring Semi primes and Possible Implications. In Proceedings of the 26th IEEE Convention in Israel, Eliat, Israel, 17–20 November 2010; pp. 182–183.
55. Bahig, H.M.; Mahdi, M.A.; Alutaibi, K.A.; AlGhadhban, A.; Bahig, H.M. Performance Analysis of Fermat Factorization Algorithms. *Int. J. Adv. Comput. Sci. Appl. (IJACSA)* **2020**, *11*, 340–352. [CrossRef]
56. Baier, S.; Mazumder, D. Diophantine approximation with prime restriction in real quadratic number fields. *Math. Z.* **2021**, *299*, 699–750. [CrossRef]
57. Pomerance, C. *Analysis and Comparison of Some Integer Factoring Algorithms, in Computational Methods in Number Theory, Part 1*; Lenstra, H.W., Jr., Tijdeman, R., Jr., Eds.; Math. Centre Tract 154: Amsterdam, The Netherlands, 1982; pp. 89–139.
58. Hoffstein, J.; Pipher, J.; Silverman, J. *An Introduction to Mathematical Cryptography*, 1st ed.; Springer Publishing Company: Oakland, CA, USA, 2008; Incorporated.
59. Stanoyevitch, A. *Introduction to Cryptography with Mathematical Foundations and Computer Implementations*, 1st ed.; Chapman & Hall/CRC: New York, NY, USA, 2010.
60. Moreno, C.J.; Wagstaff, S.S. *Sums of Squares of Integers. Discrete Mathematics and Its Applications*; Chapman & Hall, CRC: Boca Raton, FL, USA, 2006; pp. 325–326. ISBN 978-1-58488-456-9.
61. Kloster, K. Factoring a Semiprime n by Estimating φ(n). 2010. Available online: http://www.gregorybard.com/papers/phi_version_may_7.pdf (accessed on 30 September 2020).
62. Cekerevac, Z.; Dvorak, Z.; Prigoda, L.; Cekerevac, P. Man in the Middle Attacks and the Internet of Things—Security and economic risks. *FBIM Trans.* **2017**, *5*, 25–35. [CrossRef]
63. El-hajj, M.; Fadlallah, A.; Chamoun, M.; Serhrouchni, A. A Survey of Internet of Things (IoT) Authentication Schemes. *Sensors* **2019**, *19*, 1141. [CrossRef]
64. Zhou, L.; Li, X.; Yeh, K.H.; Su, C.; Chiu, W. Lightweight IoT-based authentication scheme in cloud computing circumstance. *Future Gen. Comput. Syst.* **2019**, *91*, 244–251. [CrossRef]
65. Yan, S.Y. *Factoring Based Cryptography. In Cybercryptography: Applicable Cryptography for Cyberspace Security*; Springer: Berlin/Heidelberg, Germany, 2018; pp. 217–286.
66. Suárez-Albela, M.; Fraga-Lamas, P.; Fernández-Caramés, T.M. A Practical Evaluation on RSA and ECC-Based Cipher Suites for IoT High-Security Energy-Efficient Fog and Mist Computing Devices. *Sensors* **2018**, *18*, 3868. [CrossRef]
67. Buhler, J.P.; Lenstra, H.W., Jr.; Pomerance, C. *Factoring Integers with the Number Field Sieve*; Lecture Notes in Mathematics; Springer: New York, NY, USA, 1993; Volume 1554, pp. 50–94.
68. Pollard, J. Monte Carlo methods for index computation (mod p). *Math. Comput.* **1978**, *32*, 918–924. [CrossRef]
69. Overmars, A.; Venkatraman, S. A New Method for Factorizing Semi-primes Using Simple Polynomials. In Proceedings of the 3rd International Conference on Research in Applied Science, Munich, Germany, 6–8 November 2020.
70. Stillwell, J. *Mathematics and Its History*, 2nd ed.; Springer: New York, NY, USA, 2010.
71. Vogel, D.; Onayemi, Y.; Murad, V. Integer Factorization Algorithms. 2016. Available online: http://maths.dk/teaching/courses/math357-spring2016/projects/factorization.pdf (accessed on 6 March 2021).
72. Benedetto, R.; Ingram, P.; Jones, R.; Manes, M.; Silverman, J.H.; Tucker, T.J. Current Trends and Open Problems in Arithmetic Dynamics. *Am. Math. Soc.* **2019**, *56*, 611–685. [CrossRef]
73. Wiśniewski, R.; Wisniewski, R. Representation of primes in the form $p = 6{\cdot}x \pm 1$ and its application to the RSA prime factorization. In *AIP Conference Proceedings*; AIP Publishing Center: New York, NY, USA, 2018; Volume 2040, p. 080006. [CrossRef]
74. Wu, L.; Cai, H.J.; Gong, Z. The Integer Factorization Algorithm with Pisano Period. *IEEE Access* **2019**, *7*, 167250–167259. [CrossRef]
75. Rutkowski, E.; Houghten, S. Cryptanalysis of RSA: Integer Prime Factorization Using Genetic Algorithms. In Proceedings of the 2020 IEEE Congress on Evolutionary Computation (CEC), Glasgow, UK, 19–24 July 2020; pp. 1–8.