Mathematical and Computational Applications

MDPI

Article

A Fast Factorisation of Semi-Primes Using Sum of Squares

Anthony Overmars and Sitalakshmi Venkatraman *

Department of Information Technology, Melbourne Polytechnic, Preston 3072, Australia; anthonyovermars@melbournepolytechnic.edu.au

* Correspondence: sitavenkat@melbournepolytechnic.edu.au; Tel.: +61-3-9269-1171

Received: 16 May 2019; Accepted: 2 June 2019; Published: 11 June 2019

Abstract: For several centuries, prime factorisation of large numbers has drawn much attention due its practical applications and the associated challenges. In computing applications, encryption algorithms such as the Rivest–Shamir–Adleman (RSA) cryptosystems are widely used for information security, where the keys (public and private) of the encryption code are represented using large prime factors. Since prime factorisation of large numbers is extremely hard, RSA cryptosystems take advantage of this property to ensure information security. A semi-prime being, a product of two prime numbers, has wide applications in RSA algorithms and pseudo number generators. In this paper, we consider a semi-prime number whose construction consists of primes, $N = p_1p_2$, being Pythagorean and having a representation on the Cartesian plane such that, $p = x^2 + y^2$. We prove that the product of two such primes can be represented as the sum of four squares, and further, that the sums of two squares can be derived. For such a semi-prime, if the original construction is unknown and the sum of four squares is known, by Euler's factorisation the original construction p_1p_2 can be found. By considering the parity of each of the squares, we propose a new method of factorisation of semi-primes. Our factorisation method provides a faster alternative to Euler's method by exploiting the relationship between the four squares. The correctness of the new factorisation method is established with mathematical proofs and its practical value is demonstrated by generating RSA-768 efficiently.

Keywords: Euler's factorisation; semi-prime factorisation; encryption key; RSA cryptosystem; Pythagorean prime; Gaussian prime

1. Introduction

Several mathematicians, since the work of Euclid, have been trying to uncover the mysteries behind prime numbers as they have a unique property of being divisible only by themselves and one [1,2]. The use of large prime numbers in providing information security in this digital age has triggered much research in this direction. With the advent of Rivest–Shamir–Adleman (RSA) encryption system in 1978, prime numbers are being combined innovatively to create cryptographic keys to allow secure transmission of private and sensitive information over computer networks [3,4]. Higher security can be enforced with larger prime numbers since prime factorisation is extremely hard and the RSA system takes advantage of this elegant property [5,6]. However, using very large prime numbers for RSA involves more computational time in encrypting and decrypting the information, which needs to be balanced for real-time applications. With this limitation, malicious attacks target on breaking the RSA system by finding efficient methods of prime factorisation [7,8].

Another advancement in this digital age is the evolution of the Internet of Things (IoT) that connects intelligent devices to work together in providing new personalised capabilities of products and services. However, the IoT has limited computing capabilities, storage and connectivity. In this

context, the greatest challenge is in securing IoT devices as well as the confidential communication of information over the IoT network [9]. In such an environment, the cryptographic algorithms are appropriately scaled down and the smaller prime numbers used in the encryption keys can provide more scope for hackers to perform their attacks [10]. Implementing information security capabilities involves several approaches to protect confidential data such as: (i) off-chip cryptographic memories to store sensitive information, (ii) cryptosystems such as symmetric and asymmetric cryptography, and (iii) hardware-level authentication of peripherals. In many situations, efficient and faster prime factorisation method facilitates in breaking the security algorithm in real-time, which serves as a test for establishing the security limits of the computing systems from any possible attack.

In recent years, several prime factorisation methods have been proposed, improving their efficiency to factor composite prime numbers (semi-primes) as large as 250 decimal digits utilising sufficiently large computing power [11,12]. However, semi-prime factorisation still remains a challenge that draws interest from the perspective of research in computational number theory as well as the practical difficulty of cracking RSA keys used in cryptosystems [13,14].

In this paper, we consider the application of prime factorisation for testing the security of RSA cryptography, which is based on a positive integer N, where the encryption and decryption of any message using a pair of public and private keys depends on N. In the RSA algorithm, N is a product of two prime numbers ($N = p_1 p_2$) and is a semi-prime [15]. In the secured transmission of a message, p_1 and p_2 are employed in RSA to generate the key pairs for encryption and decryption. If p_1 and p_2 are known, then the cracking of the RSA keys becomes possible [16]. Hence, the security of RSA depends on how difficult the factorisation of N is. This motivates research works to propose new factorisation methods. Euler's factorisation is the most popular method that is well suited for finding prime factors of semi-primes whose constructions are based on Pythagorean primes [17]. We identify the limitations of Euler's method as it is applicable to only semi-prime constructs that are Pythagorean primes. Our aim in this paper is to propose an improved method by considering the parity of the squares approach. We provide a proof theory that our proposed method requires much fewer steps for the factorisation process of RSA modulus N. Hence, our enhanced method could be applied to test for factorisation attacks that would provide insights into choosing the key size and the time period until which an RSA-based public key algorithm is safe from an attack.

2. Theory and Proposed Method

The definition of a generic Pythagorean triple is denoted as the triple:

$$(a,\ b,\ c)$$

where $a,\ b,\ c \in \mathrm{N}\backslash\{0\}$, with $c > \max\{a, b\}$, i.e., a and b denote the sides of a right triangle, and c denotes the length of the hypothenuse [18]. From the fundamental property of right-angled triangles, we have, $a^2 + b^2 = c^2$ which is among the Diophantine equations [19]. The set of all Pythagorean triples is denoted by P.

For every $m,\ n \in \mathrm{N}\backslash\{0\}$, with the series of odd and even Pythagorean triples defined in terms of m and n, it has been proved in previous work by Overmars et al. [17] that:

$$\begin{aligned} a &= (2m + n - 1)^2 - n^2 \\ b &= 2\,(2m + n - 1)\,n \\ c &= (2m + n - 1)^2 + n^2. \end{aligned}$$

The above is referred to as the Overmars triangles, and we are interested in the properties of the hypotenuse of the triangle in this paper. As commonly represented, let us denote the hypotenuse in this paper by N, which could be represented as $N = (2m + n - 1)^2 + n^2$. From this equation, if n is odd, then n^2 is also odd and it follows that $2m + n - 1$ will be even and $(2m + n - 1)^2$ will also be even. Conversely, if n is even, n^2 will be even and both $2m + n - 1$ and $(2m + n - 1)^2$ will be odd.

Fermat's Christmas theorem [20] showed that for a Pythagorean prime $p \equiv 1 \; mod \; 4 = x^2 + y^2$. This was extended by Overmars [19] taking into consideration the parity of x and y such that $x^2 + y^2 = (2m + n - 1)^2 + n^2$, noting that for a particular Pythagorean triangle, the sides making up the hypotenuse where opposite in parity. For a semi-prime consisting of two such triangles whose hypotenuse are prime, it will be shown here that its two sums of two squares will have the following parity:

$$N = p_1 p_2 = odd_1^2 + even_1^2 = odd_2^2 + even_2^2.$$

If we consider the following differences:

$$\Delta o = odd_1 - odd_2 \; , \; \Delta e = even_1 - even_2, \text{ and } g = \gcd(\Delta o, \Delta e).$$

It can be shown that one of the primes p_2 can be represented as:

$$p_2 = \left(\frac{\Delta o}{g}\right)^2 + \left(\frac{\Delta e}{g}\right)^2, \; p_1 = \frac{N}{p_2}.$$

Euler's factorisation method is suited to semi-primes whose construction are prime factors that are said to be Pythagorean [21,22] and can further be improved upon by considering the parity of the squares. The limitations of this method pertain only to semi-prime constructs that are Pythagorean primes ($p = 1 \; mod \; 4$). It can also be shown (Section 3) that the combinations of Pythagorean primes with Gaussian primes cannot be represented as the sum of two squares. The implication here is that if the semi-prime construction selects Pythagorean and/or Gaussian primes randomly, only one quarter of the semi-prime constructions avail themselves to this factorisation method. The distribution of Pythagorean and Gaussian primes appear in the set of natural numbers with equal probability. A comprehensive description on the Pythagorean and Gaussian primes and their probabilistic distributions are provided by Oliver Knill [23].

Consider a semi-prime number whose construction consists of primes $p_1, \; p_2$ with $N = p_1 p_2$, being Pythagorean, and having a representation on the Cartesian plane such that, $p = x^2 + y^2$. It can easily be shown that the product of two such primes can be represented as the sum of four squares from which two sums of two squares can be derived. For such a semi-prime, if the original construction is unknown and the sum of four squares is known, the original construction p_1, p_2 can be found. This paper considers the sum of four squares from which two sums of squares is determined, and hence by Euler's factorisation the original construction $p_1 p_2$ can be found. By considering the parity of each of the squares, a new way of determining the semi-prime construction is described. Our proposed method provides an alternative to Euler and uses Overmars triangles. This exploits the relationship between the four squares, from which the two sums of two squares can be determined by considering each squares' parity, and thereby the factorisation is determined. We describe the Euler's factorisation forming the foundation of our proposed method and the related proofs in the next two sections of the paper.

3. Euler's Factorisation Method

We begin by considering Gaussian primes and Pythagorean primes. From the literature, Gaussian primes are of the form [24,25]:

$$4x - 1 \equiv 3 \; mod \; 4,$$

and Pythagorean primes of the form:

$$4x + 1 \equiv 3 \; mod \; 4.$$

According to Fermat's Christmas theorem on the sum of two squares, we have the following:

$$\text{an odd prime } p = x^2 + y^2 \text{ if } p \equiv 1 \; mod \; 4.$$

Gaussian primes are of the form $p \equiv 3 \ mod \ 4$ and are not representable as the sum of two squares.

Proposition: *A semi-prime whose prime factors are Pythagorean can be expressed as the sum of four squares, from which two sums of squares can be derived.*

Lemma: *A semi-prime* $N = p_1p_2$, $p_1 = a^2 + b^2$, $p_2 = c^2 + d^2$ *is expressed as the sum of four squares, such that:*

$$N = p_1p_2 = \left(a^2 + b^2\right)\left(c^2 + d^2\right) = (ac)^2 + (bc)^2 + (ad)^2 + (bd)^2.$$

Proof: Euler's factorisation

Let us consider the method of Euler's factorization, where a number (N) can be factored by writing it as a sum of two squares in two different ways as follows:

$$N = r^2 + s^2 = t^2 + u^2 \Rightarrow r^2 - t^2 = u^2 - s^2 \Rightarrow (r-t)(r+t) = (u-s)(u+s)$$
$$p_1 = \left(\tfrac{gcd(r-t,u-s)}{2}\right)^2 + \left(\tfrac{gcd(r+t,u+s)}{2}\right)^2, p_2 = \left(\tfrac{gcd(r+t,\ u-s)}{2}\right)^2 + \left(\tfrac{gcd(r-t,\ u+s)}{2}\right)^2$$

Let us consider the example N = 2137458620009 to find p_1 and p_2, using the sum of squares as follows:

$$N = 324403^2 + 1425560^2 = 643603^2 + 1312720^2,$$

combining even and odds we get:

$$1425560^2 - 1312720^2 = 643603^2 - 324403^2$$
$$a^2 - c^2 = d^2 - b^2 \Longrightarrow (a-c)(a+c) = (d-b)$$
$$(d+b) = (968006)\ (319200) = (2738280)\ (112840),$$

using the greatest common divisor (gcd):

$$\frac{gcd(a-c,\ d-b\ )}{2} = \frac{gcd(968006,\ 2738280)}{2} = 1201,$$
$$\frac{gcd(a+c,\ d+b)}{2} = \frac{gcd(319200, 112840)}{2} = 140,$$
$$p_1 = 1201^2 + 140^2 = 1462001$$
$$\frac{gcd(a+c, d-b)}{2} = \frac{gcd(319200,\ 2738280)}{2} = 1140,$$
$$\frac{gcd(a-c,\ d+b)}{2} = \frac{gcd(968006, 112840)}{2} = 403,$$
$$p_2 = 1140^2 + 403^2 = 1462009.$$

The above example illustrates how the semi-primes of N can be derived as the sum of two squares using Euler's factorisation method.

Now, express the sum of four squares as two sums of two squares.

Let $r = ad + bc,\ s = bd - ac,\ t = ac + bd,\ u = ad - bc$
$\Rightarrow r^2 = (ad)^2 + 2abcd + (bc)^2,\ s^2 = (ac)^2 - 2abcd + (bd)^2,$
$t^2 = (ac)^2 + 2abcd + (bd)^2,\ u^2 = (ad)^2 - 2abcd + (bc)^2$
$\Rightarrow r^2 + s^2 = (ac)^2 + (ad)^2 + (bc)^2 + (bd)^2 = t^2 + u^2$

$$N = p_1p_2 = \left(\mathrm{a}^2 + \mathrm{b}^2\right)\left(\mathrm{c}^2 + \mathrm{d}^2\right) \tag{1}$$

$$N = (\mathrm{ac})^2 + (bc)^2 + (ad)^2 + (bd)^2 \tag{2}$$

$$N = (ad + bc)^2 + (bd - ac)^2 = (ad - bc)^2 + (bd + ac)^2 \quad (3)$$

■

4. Proposed Semi-Prime Factorisation Using Sum of Squares

Overmars et al. [17] showed that all Pythagorean triples could be represented as $N = n^2 + (n + 2m - 1)^2$. If the semi-prime is constructed using two Pythagorean primes $(4x + 1)$ then two representations as the sum of two squares can be found and Euler's factorisation method can be applied. Finding these two representations is non-trivial and computationally intensive for large numbers even with computers with a high performance central processing unit (CPU). The equation $N(m, n) = n^2 + (n + 2m - 1)^2$ provides an elegant search using increments of n and fine convergence using m, and the CPU-intensive square root can be avoided. In this way n is incremented and m is decremented about N to find one of the two solutions along the diagonal of a field of $N(m, n) \approx N$. It can also be shown (as a future work) that once one sum of the squares is known, this can be used to find the other.

Consider the example of a large number, $N = 2137458620009$.

$$\begin{aligned} N(m_1, n_1) &= n_1^2 + (n_1 + 2m_1 - 1)^2 \\ &= 324403^2 + (324403 + 2(550579) - 1)^2 \\ &= 324403^2 + 1425560^2 \end{aligned}$$

$$\begin{aligned} N(m_2, n_2) &= n_2^2 + (n_2 + 2m_2 - 1)^2 \\ &= 643603^2 + (643603 + 2(334559) - 1)^2 \\ &= 643603^2 + 1312720^2 \end{aligned}$$

$$\begin{aligned} N_1(324403, 550579) &= N_2(643603, 334559) \\ &= 2137458620009. \end{aligned}$$

For completeness, N can be represented as two Pythagorean triangles as shown [2]:

$$\begin{aligned} &\Delta(m,\ n) = \Delta(a,\ b,\ c) \\ &a(m, n) = 2n(n + 2m - 1), \\ &b(m, n) = (2m - 1)(2n + 2m - 1), \\ &c(m.n) = n^2 + (n + 2m - 1)^2 \\ &\Delta(m_1, n_1) = \Delta(a_1, b_1, c_1) : \\ &\Delta(324403,\ 550579) = \Delta(28197495801360,\ 8357740887191,\ 29410042540009) \\ &\Delta(m_2, n_2) = \Delta(a_2, b_2, c_2) : \\ &\Delta(643603, 334559) = \Delta(1689741060320,\ 1309008976791,\ 29410042540009). \end{aligned}$$

Once the two sums of two squares have been found, Euler's factorisation method can be used. $N : N = p_1 p_2$.

5. Proposed Method Using Gaussian and Pythagorean Primes

According to Fermat's Christmas theorem, if Pythagorean primes $(4x + 1 \equiv 4x - 3)$ are used to construct a composite of the semi-prime number (N), a solution exists as two sums of two squares. However, if N is constructed using Gaussian primes $(4x - 1 \equiv 4x + 3)$, then Euler's sum of two squares method cannot be used [26,27]. There is a lack of research in this direction [28,29]. This motivates us to investigate in this paper, if there is a test case which we can use to see if a composite of the semi-prime number has been constructed using Pythagorean primes.

Consider the following composite constructions:

(i) $N = (4x + 1)(4y + 1)$ using Pythagorean primes;

(ii) $N = (4x-1)(4y-1)$ using Gaussian primes;
(iii) $N = (4x+1)(4y-1)$ or $(4x-1)(4y+1)$ using mixed Pythagorean and Gaussian primes.

(i) Pythagorean prime construction

$$N = (4x+1)(4y+1) = 16xy + 4(x+y) + 1.$$

We have verified that two sums of two squares representations exist and Euler's factorisation can be used.

$$1 \equiv N \ mod \ 4.$$

As an illustration, consider the following example for $N = 793$.

$$793 = 10^2 + 12^2 + 15^2 + 18^2 = 13 * 61 = 3^2 + 28^2 = 8^2 + 27^2.$$

Note the parity of the sum of four squares is (odd, even, even, even).

(ii) Gaussian prime construction

$$N = (4x-1)(4y-1) = 16xy - 4(x+y) + 1 \equiv 4m - 3 \equiv 4n + 1.$$

Sums of three squares exist $1 \equiv N \ mod \ 4$.
As an illustration, consider the following example for $N = 649$.

$$649 = 11 \times 59 = 1^2 + 18^2 + 18^2 = 3^2 + 8^2 + 24^2 = 6^2 + 17^2 + 18^2 = 8^2 + 12^2 + 21^2$$
$$= 10^2 + 15^2 + 18^2 = 12^2 + 12^2 + 19^2.$$

(iii) Mixed Pythagorean-Gaussian prime construction

$$-1 \equiv N \ mod \ 4.$$

$$N = (4x+1)(4y-1) = 16xy - 4(x-y) - 1, N = (4x-1)(4y+1) = 16xy + 4(x-y) - 1.$$

Sums of four squares exist.
$3 \equiv N \ mod \ 4$.
$13 \times 59 = 767$
$= 1^2 + 1^2 + 6^2 + 27^2 = 1^2 + 1^2 + 18^2 + 21^2$
$= 1^2 + 3^2 + 9^2 + 26^2 = 1^2 + 6^2 + 17^2 + 21^2$
$= 1^2 + 9^2 + 18^2 + 19^2 = 1^2 + 10^2 + 15^2 + 21^2$
$= 2^2 + 3^2 + 5^2 + 27^2 = 2^2 + 3^2 + 15^2 + 23^2$
$= 3^2 + 6^2 + 19^2 + 19^2 = 3^2 + 7^2 + 15^2 + 22^2$
$= 3^2 + 11^2 + 14^2 + 21^2 = 5^2 + 6^2 + 9^2 + 25^2$
$= 6^2 + 9^2 + 11^2 + 23^2 = 6^2 + 9^2 + 17^2 + 19^2$
$= 6^2 + 11^2 + 13^2 + 21^2 = 7^2 + 9^2 + 14^2 + 21^2$
$= 7^2 + 13^2 + 15^2 + 18^2 = 9^2 + 9^2 + 11^2 + 22^2$
$= 9^2 + 10^2 + 15^2 + 19^2$
$= 11^2 + 14^2 + 15^2 + 15^2$.

Note the parity of the sum of four squares is (even, odd, odd, odd).

In summary, a semi-prime whose composite construction is based upon both Pythagorean and Gaussian primes can easily be identified when $N \ mod \ 4 \equiv 3$ is true and the sum of four squares parity is (even, odd, odd, odd) and Euler's factorisation cannot be used. Table 1 provides possible composite constructs of a semi-prime number using Pythagorean and Gaussian primes as the factors. When $P \ mod \ 4 \equiv 1$ is true, the composite could be constructed using Pythagorean primes or Gaussian primes. When the Pythagorean construct is confirmed, we can verify that: (i) the sum of four squares parity is

(odd, even, even, even), (ii) the two sums of two squares can be found, and (iii) Euler's factorisation can be employed.

Table 1. Possible composite constructs using Pythagorean and Gaussian primes.

Composite	4*x*−1	4*x*+1	3.15	11	13
$4y-1$	$16xy-4(x+y)+1$	$16xy-4(x-y)-1$	59	649	767
$4y+1$	$16xy-4(y-x)-1$	$16xy+4(x+y)+1$	61	671	793

Proof: Let N be a semi-prime and p_1 and p_2 are its two prime factors so that $N = p_1p_2$. Assume also that p_1 and p_2 are distinct. Suppose that the primes p_1 and p_2 are "Pythagorean" (2-square), that is, they can each be written as the sum of two squares of natural numbers: $p_1 = a^2 + b^2$, $p_2 = c^2 + d^2$, then:

$$\begin{gathered} N = p_1p_2 = \left(a^2+b^2\right)\left(c^2+d^2\right) \\ = a^2c^2 + a^2d^2 + b^2c^2 + b^2d^2 \\ N = a^2c^2 + 2abcd + b^2d^2 + a^2d^2 - 2abcd + b^2c^2 \\ = a^2d^2 + 2abcd + b^2c^2 + a^2c^2 - 2abcd + b^2d^2 \\ = (ac+bd)^2 + (ad-bc)^2 \\ = (ad+bc)^2 + (ac-bd)^2 \\ = t^2 + u^2 = r^2 + s^2. \end{gathered}$$

Therefore, N is also Pythagorean, and can be represented as the sum of two squares in two different ways:

$$\begin{gathered} r = ad + bc\ ,\ s = ac - bd\ , \\ t = ac + bd,\ u = ad - bc. \end{gathered}$$

The problem is rephrased as:
Given $N = r^2 + s^2 = t^2 + u^2$, is known. $r, s, t, u, N \in \mathbb{N}\backslash\{0\}$. Find p_1 and p_2.

$$\begin{gathered} s + t = 2ac,\ r + u = 2ad, \\ g = \gcd(s+t, r+u) = \gcd(2ac, 2ad) = 2a \\ \tfrac{s+t}{g} = \tfrac{2ac}{2a} = c,\ \tfrac{r+u}{g} = \tfrac{2ad}{2a} = d, \\ p_2 = c^2 + d^2 = \left(\tfrac{s+t}{g}\right)^2 + \left(\tfrac{r+u}{g}\right)^2 \\ = \tfrac{(s+t)^2+(r+u)^2}{g^2} \end{gathered}$$

then the factors of N are:

$$p_2 = c^2 + d^2 = \left(\frac{s+t}{g}\right)^2 + \left(\frac{r+u}{g}\right)^2, p_1 = \frac{N}{p_2}$$

■

6. Verification with Ordering Ambiguity

In this section, we verify that the ordering of the odd and even pairs does not affect the results. Consider the following example 1 of ordering of odd and even pairs as follows:

$$1000009 = 1000^2 + 3^2 = 972^2 + 235^2,$$

$$a = 1000,\ b = 3,\ c = 972,\ d = 235$$

$$g = gcd(1000 + 972,\ 235 + 3) = 34,$$

$$x_1 = 1972/34 = 58,\ y_1 = 238/34 = 7$$

$$p_2 = 58^2 + 7^2 = 3413.$$

Consider the following example 2 of ordering of odd and even pairs as follows:

$$1000009\ = 1000^2 + 3^2 = 235^2 + 972^2,$$

$$a = 1000,\ b = 3,\ c = 235,\ d = 972,$$

$$g = gcd(1000 + 235,\ 972 + 3) = 65,$$

$$x_1 = 1235/65 = 19,\ y_1 = 975/65 = 15,$$

$$p_1 = 19^2 + 15^2\ = 586\ =\ 2 * 293.$$

Furthermore, we have additional information which can assist in removing this ambiguity. If we consider odd and even pairs when ordering the sums, we can use Overmars triangles to conserve parity and remove this ambiguity.

Consider the following form:

$$p = a^2 + b^2 = (2m + n - 1)^2 + n^2, a = 2m + n - 1,\ b = n.$$

When $n = odd,\ 2m + n - 1 = even \Rightarrow a\ is\ even,\ b\ is\ odd$.

Conversely, $n = even,\ 2m + n - 1 = odd \Rightarrow a\ is\ odd,\ b\ is\ even$.

Odd/even or even/odd parity is thus assured and preserved for each of the sums of squares and this additional information can be used to remove the ordering ambiguity.

Consider the difference between odd and even parts of the two sums of two squares, this removes the ordering ambiguity:

$$\Delta o = odd_1 - odd_2, \Delta e = even_1 - even_2, g = \gcd(\Delta o, \Delta e),$$

one of the primes p_2 can be given as:

$$\begin{gathered} p_2 = \left(\tfrac{\Delta o}{g}\right)^2 + \left(\tfrac{\Delta e}{g}\right)^2 \Delta o = |3 - 235| = 232, \\ \Delta e = |1000 - 972| = 28, \\ g = \gcd(232, 28) = 4, \\ \Rightarrow p_2 = \left(\tfrac{232}{4}\right)^2 + \left(\tfrac{28}{4}\right)^2 = 58^2 + 7^2 = 3413. \end{gathered}$$

Proof: Express $p_1,\ p_2$ as Overmars triangles.

$$\begin{gathered} p_1 = a^2 + b^2 = (2m_1 + n_1 - 1)^2 + n_1^2, \\ p_2 = c^2 + d^2 = (2m_2 + n_2 - 1)^2 + n_2^2 \end{gathered}$$

$$\begin{gathered} N = p_1 p_2 = \left(a^2 + b^2\right)\left(c^2 + d^2\right) \\ = \left((2m_1 + n_1 - 1)^2 + {n_1}^2\right)\left((2m_2 + n_2 - 1)^2 + {n_2}^2\right) \\ = (ac)^2 + (bd)^2 + (ad)^2 + (bc)^2 \\ = ((2m_1 + n_1 - 1)(2m_2 + n_2 - 1))^2 + (n_1 n_2)^2 \\ +((2m_1 + n_1 - 1)n_2)^2 + (n_1(2m_2 + n_2 - 1))^2 \end{gathered}$$

$$\begin{aligned} N &= (ac+bd)^2+(ad-bc)^2 = s^2+t^2 \\ &= ((2m_1+n_1-1)(2m_2+n_2-1)+n_1n_2)^2 \\ &+((2m_1+n_1-1)n_2-n_1(2m_2+n_2-1))^2 \\ &= (ad+bc)^2+(ac-bd)^2 = u^2+r^2 \\ &= ((2m_1+n_1-1)n_2+n_1(2m_2+n_2-1))^2+((2m_1+n_1-1)(2m_2+n_2-1)-n_1n_2)^2. \end{aligned}$$

Let us express two semi-primes $N_1,\ N_2$ as Overmars triangles as follows:

$$\begin{aligned} N_1 &= s^2+t^2 = (2v+t-1)^2+t^2, \\ N_2 &= u^2+r^2 = (2w+r-1)^2+r^2, \end{aligned}$$

where $v=\frac{s-t+1}{2},\ w=\frac{u-r+1}{2}$.

Substitute for $v,\ t,\ w, r,$

$N_1 = s^2+t^2 = (2v+t-1)^2+t^2,$
$v = 2m_1m_2-m_1+2n_1m_2+n_1n_2-n_1-m_2+1,$
$t = 2m_1m_2-2n_1m_2+n_1-n_2,$
$N_2 = u^2+r^2 = (2w+r-1)^2+r^2,$
$r = 4m_1m_2+2m_1n_2-2m_1+2n_1m_2-n_1-2m_2-n_2+1,$
$w = -2m_1m_2+m_1+m_2+n_1n_2,$

$$\begin{aligned} N_1 &= 16\,{m_1}^2\,m_2^2+16\,m_1^2\,m_2\,n_2-16\,m_1^2\,m_2+8\,m_1^2\,n_2^2-8\,m_1^2\,n_2+4\,m_1^2+16\,m_1\,n_1\,m_2^2 \\ &+16\,m_1\,n_1\,m_2\,n_2-16\,m_1\,n_1\,m_2+8\,m_1\,n_1\,n_2^2-8\,m_1\,n_1n_2+4\,m_1\,n_1 \\ &-16\,m_1\,m_2^2-16\,m_1\,m_2\,n_2+16\,m_1\,m_2-8\,m_1\,n_2^2+8\,m_1\,n_2-4\,m_1+8\,n_1^2\,m_2^2 \\ &+8\,n_1^2\,m_2\,n_2-8\,n_1^2\,m_2+4\,n_1^2\,n_2^2-4\,n_1^2\,n_2+2\,n_1^2-8\,n_1\,m_2^2-8\,n_1\,m_2\,n_2 \\ &+8\,n_1\,m_2-4\,n_1\,n_2^2+4\,n_1\,n_2-2\,n_1+4\,m_2^2+4\,m_2\,n_2-4\,m_2+2\,n_2^2-2\,n_2 \\ &+1, \end{aligned}$$

$$\begin{aligned} N_2 &= 16\,{m_1}^2\,m_2^2+16\,m_1^2\,m_2\,n_2-16\,m_1^2\,m_2+8\,m_1^2\,n_2^2-8\,m_1^2\,n_2+4\,m_1^2+16\,m_1\,n_1\,m_2^2 \\ &+16\,m_1\,n_1\,m_2\,n_2-16\,m_1\,n_1\,m_2+8\,m_1\,n_1\,n_2^2-8\,m_1\,n_1n_2+4\,m_1\,n_1 \\ &-16\,m_1\,m_2^2-16\,m_1\,m_2\,n_2+16\,m_1\,m_2-8\,m_1\,n_2^2+8\,m_1\,n_2-4\,m_1+8\,n_1^2\,m_2^2 \\ &+8\,n_1^2\,m_2\,n_2-8\,n_1^2\,m_2+4\,n_1^2\,n_2^2-4\,n_1^2\,n_2+2\,n_1^2-8\,n_1\,m_2^2-8\,n_1\,m_2\,n_2 \\ &+8\,n_1\,m_2-4\,n_1\,n_2^2+4\,n_1\,n_2-2\,n_1+4\,m_2^2+4\,m_2\,n_2-4\,m_2+2\,n_2^2-2\,n_2 \\ &+1, \end{aligned}$$

$$\begin{aligned} N_1 = N_2 &= \left(4\,m_1^2+4\,m_1\,n_1-4\,m_1+2\,n_1^2-2\,n_1+1\right)\left(4\,m_2^2+4\,m_2\,n_2-4\,m_2+2\,n_2^2-2\,n_2+1\right) \\ &= \left[(2\,m_1+n_1-1)^2+n_1^2\right]\left[(2m_2+n_2-1)^2+n_2^2\right], \end{aligned}$$

$N_1 = N_2 = p_1p_2$
$\Rightarrow p_1 = (2m_1+n_1-1)^2+n_1^2\,,$
$p_2 = (2m_2+n_2-1)^2+n_2^2.$ ■

Recall from Section 2:

$$N = p_1p_2 = \left(\mathrm{a}^2+\mathrm{b}^2\right)\left(\mathrm{c}^2+\mathrm{d}^2\right) \tag{4}$$

$$N = (\mathrm{ac})^2+(bc)^2+(ad)^2+(bd)^2 \tag{5}$$

$$N = (ad+bc)^2+(bd-ac)^2 = (ad-bc)^2+(bd+ac)^2 \tag{6}$$

Let $a = 2m_1+n_1-1,\ b=n_1,\ c=2m_2+n_2-1\,,\ d=n_2$ and consider the parities given in Table 2.

Table 2. Parity considerations for factorisation.

b	d	a	c	$ad+bc$	$ac-bd$	$ad-bc$	$ac+bd$	Δo	Δe	g	$\frac{\Delta o}{g}$	$\frac{\Delta e}{g}$
o	o	e	e	e	o	e	o	$2bd$	$2bc$	$2b$	d	c
e	e	o	o	e	o	e	o	$2bd$	$2bc$	$2b$	d	c
o	e	e	o	o	e		e	$2bc$	$2bd$	$2b$	c	d
e	o	o	e	o	e	o	e	$2bc$	$2bd$	$2b$	c	d

($o = odd, e = even$)

From Equation (1) $p_2 = c + d$, observe from the table the parities, recalling ($b = n_1,\ d = n_2$), $parity(b) = parity(d)$,

$$c = \frac{\Delta e}{g},\ d = \frac{\Delta o}{g},\ p_2 = \left(\frac{\Delta e}{g}\right)^2 + \left(\frac{\Delta o}{g}\right)^2$$

$parity(b) \neq parity(d)$,

$$c = \frac{\Delta o}{g},\ d = \frac{\Delta e}{g},\ p_2 = \left(\frac{\Delta o}{g}\right)^2 + \left(\frac{\Delta e}{g}\right)^2$$

$$\therefore p_2 = \left(\frac{\Delta o}{g}\right)^2 + \left(\frac{\Delta e}{g}\right)^2$$

■

$$p_2 = \left(\frac{\Delta o}{g}\right)^2 + \left(\frac{\Delta e}{g}\right)^2,$$

$$p_1 = \frac{N}{p_2},$$

$$N = p_1 p_2$$

$$= o_1^2 + e_1^2 = o_2^2 + e_2^2,$$

$$\Delta o = o_1 - o_2,$$

$$\Delta e = e_1 - e_2,$$

$$g = \gcd(\Delta o, \Delta e).$$

7. Application of Proposed Method for RSA Factorisation

Historically, an RSA algorithm was experimentally tested using brute force attacks by trying all possible secret keys (public and private keys). When RSA employed shorter keys for encryption and decryption, they became easier to identify using brute force attacks [15]. Larger keys could escape from brute force attacks since they are exponentially more difficult to crack and hence, the key length is an indicator of how a brute force attack is practically feasible. Therefore, the strength of an RSA cryptosystem is measured theoretically by determining how many steps it would take for a brute force attack to crack the keys. However, with greater computations involved with larger keys, and as the encryption and decryption algorithm takes much larger time, there is a limitation on the key length used for practical applications. Hence, different factoring algorithms and faster cryptosystems have been researched [29–31]. Different implementations of the modular exponentiation have resulted in the timing variations of the attacks used for performing an RSA attack. In other words, for an encrypted message or cipher text C, it is the ability to find d by determining the time taken to compute $C^d (mod\ N)$.

Another method is to perform factorisation attacks mathematically by factoring the modulus N, which forms the underlying structure of an RSA function [15]. While there are several factorisation approaches, the common goal is to factor the semi-prime, $N = p_1 p_2$. The encryption algorithm selects the secret keys p_1 and p_2 to calculate the public key $N = p_1 p_2$. The decryption algorithm then factors N to obtain the keys p_1 and p_2. Hence, new factorisation algorithms have been mathematically derived in

generating public and private keys of an RSA algorithm, however, they take a long period of time for the factorisation of N when the keys p_1 and p_2 are very large. The first RSA number successfully generated in 1991 was RSA-100 and subsequently up to RSA-500. They were labeled according to their key size, namely number of decimal digits occupied when implemented in the computer. While factorisation of RSA-617 was successful before RSA-576, many of the bigger numbers have still not been factored. Hence, the factoring challenge was introduced to give an insight into which key length is safe and for how long so that applications could choose the key length for their RSA encryption algorithm ensuring security until it is proven to be safe. This forms the motivation for researchers to mathematically prove RSA factorisation time limits that help in the understanding of the cryptanalytic strength of commonly adopted cryptosystems in practice. Hence, in this paper, we focus on proposing a new factorisation method that is efficient in terms of the speed of factorisation. We demonstrate the application of our proposed factorisation method for RSA-768 as shown in Table 3.

Table 3. Rivest–Shamir–Adleman (RSA) factorisation (e.g. RSA-768) using the proposed method.

Factorisation Steps	Outputs
RSA-768	123018668453011775513049495838496272077285356959533479219732245215172640050726365751874520219978646938995647494277406384592519255732630345373154826850791702612214291346167042921431160222124047927473779408066535141959745985690214341324735225095563949167293855908848935409374528369229805467041
$even_1$	006072973454407040141205804085247474318845727809745397174221955433318197941448284339595819001813420702779026856093076047024562771910171296992175792528202395940970592960030457
odd_1	247901706805007760807736122453772349433151532756821778429032745141728462002443175627130760256835410680433963545331772481197537806608101061289846883397001761964821768695812460015562606173107714323756032046030289409406476420809343834223428347373623028346324892726495582331298944504966553217498272274896100742900730205705372834974811451270728033945027202540275577573410662325660619655384076125635474611096546408504469117233298969036454836249445802247809636016519588320627928153105939350555378446039188539852332233574215955680451442947458445760405926068508893760679171519836489078225440454876326349940386324642607085134081244951155373698570493674604366679959042824463379962795263227915816434308764267603228381573966651127923337341714339681027009279873630891733478071698956898786044169848212690817704794983713768568912431388982883793878002287614711652531743087737814467999489
$\Delta o = odd_1 - odd_2$	
$\Delta e = even_1 - even_2$	
$g = \gcd(\Delta o, \Delta e)$	
$\frac{\Delta o}{g}$	
$\frac{\Delta e}{g}$	
$p_2 = \left(\frac{\Delta o}{g}\right)^2 + \left(\frac{\Delta e}{g}\right)^2$	
$p_1 = \frac{N}{p_2}$	

In general, the computational complexity of a factorisation algorithm can be measured by the number of operations it performs [32]. Factorisation algorithms are also interesting from the computational complexity viewpoint. Since the existing algorithms are not able to solve the factoring problem in polynomial time, encryption keys are developed as they cannot be factored in a reasonable amount of computer time [33]. Overall, our proposed factorisation method provides a faster alternative to the commonly used modulus exponentiation methods and Euler's method, by exploiting the relationship between the four squares. The implementation of our method requires a constant amount of clock cycles for completing such simple arithmetic operations and hence requires a reduced computational complexity. The mathematical proof of our new factorisation method and its demonstration for generating RSA-768 have been established. Many of the bigger prime numbers are yet to be factored and are expected to remain unfactored for quite some time. However, such beliefs can be challenged when new factorisation techniques and new technologies are introduced. Since encryption schemes are being used today to protect financial and other confidential data, ways and means of developing a single quantum computer to factor very large primes quickly and in parallel are

under consideration. Shor's quantum algorithm developed in 1994 depends on a computer with a large number of quantum bits to calculate the prime factors of a large number. A large prime number, with even 232 digits, could take more than two years to factor using hundreds of computers working in parallel. Hence, a major breakthrough in technologies such as quantum computers along with the innovation of Shor's algorithm and other work, including ours, could make this problem domain an interesting space for academic researchers and industry practitioners to explore further.

8. Conclusions and Future Work

In this paper, we proposed a new method for semi-prime factorisation which forms a cornerstone for security in RSA cryptosystems. By exploiting the relationship between a set of four squares, we provide a relatively simple, fast and scalable factorisation method that is computationally more efficient than the existing, commonly used modulus exponentiation methods and Euler's method. The mathematical proofs behind the development of our simple and reliable algorithm for semi-prime factorisation were presented. In addition, the application of our method to factorise large semi-primes for generating RSA-768 was established.

Our work in this paper forms the backbone in creating new research opportunities. With new technologies such as IoT, blockchain and quantum computers evolving, future work would involve exploring our factorisation method in various cryptographic protocols within such new computing paradigms.

Author Contributions: Conceptualization, A.O. and S.V.; methodology, A.O.; validation, A.O.; resources, S.V.; data curation, A.O.; writing—original draft preparation, A.O.; writing—review and editing, S.V.

Funding: This research received no external funding.

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

References

1. Goldston, D.A.; Graham, S.; Pintz, J.; Yildirim, C.Y. Small gaps between primes or almost primes. *Trans. Am. Math. Soc.* **2009**, *361*, 5285–5330. [CrossRef]
2. Kaddoura, I.; Abdul-Nabi, S. On formula to compute primes and the nth prime. *Appl. Math. Sci.* **2012**, *6*, 3751–3757.
3. Rivest, R.L.; Shamir, A.; Adleman, L. A method for obtaining digital signatures and public-key cryptosystems. *Commun. Acm* **1978**, *21*, 120–126. [CrossRef]
4. Rescorla, E. *SSL and TLS: Designing and Building Secure Systems*; Addison-Wesley: Reading, UK, 2001.
5. Sun, H.M.; Wu, M.E.; Ting, W.C.; Hinek, M.J. Dual RSA and its Security Analysis. *IEEE Trans. Inf. Theory* **2007**, *53*, 2922–2933.
6. Schneier, B. *Applied Cryptography*, 2nd ed.; John Wiley & Sons, Inc.: New York, NY, USA, 1996.
7. Clark, J.; van Oorschot, P.C. SoK: SSL and HTTPS: Revisiting past challenges and evaluating certificate trust model enhancements. In Proceedings of the 2013 IEEE Symposium on Security and Privacy (SP), Berkeley, CA, USA, 19–22 May 2013; pp. 511–525.
8. Aboud, S.J. An efficient method for attack RSA scheme. In Proceedings of the ICADIWT 2nd International Conference, London, UK, 4–6 August 2009; pp. 587–591.
9. 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] [PubMed]
10. Sen, S.; Koo, J.; Bagchi, S. TRIFECTA: Security, Energy Efficiency, and Communication Capacity Comparison for Wireless IoT Devices. *IEEE Internet Comput.* **2018**, *22*, 74–81. [CrossRef]
11. 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.
12. Yamagishi, S. Diophantine equations in semiprimes. *arXiv* **2017**, arXiv:1709.03605.
13. Weisstein, E.W. *Semiprime*; Wolfram Research, Inc.: Champaign, IL, USA, 2003.

14. 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.
15. Ambedkar, B.R.; Bedi, S.S. A New Factorization Method to Factorize RSA Public Key Encryption. *Int. J. Comput. Sci. Issues (IJCSI)* **2011**, *8*, 242–247.
16. Yan, S.Y. Factoring Based Cryptography. In *Cyber cryptography: Applicable Cryptography for Cyberspace Security*; Springer: Berlin/Heidelberg, Germany, 2018; pp. 217–286. [CrossRef]
17. Overmars, A.; Ntogramatzidis, L.; Venkatraman, S. A new approach to generate all Pythagorean triples. *AIMS Math.* **2019**, *4*, 242–253. [CrossRef]
18. Overmars, A.; Venkatraman, S. Pythagorean-Platonic lattice method for finding all co-prime right angle triangles. *Int. J. Comput. Inf. Eng.* **2017**, *11*, 1192–1195.
19. Overmars, A.; Ntogramatzidis, L. A new parameterisation of Pythagorean triples in terms of odd and even series. *arXiv* **2015**, arXiv:1504.03163.
20. Bell, E.T. *The Prince of Amateurs: Fermat*; Simon and Schuster: New York, NY, USA, 1986; pp. 56–72.
21. Hiary, G.A. A Deterministic Algorithm for Integer Factorization. *Math. Comput.* **2016**, *85*, 2065–2069. [CrossRef]
22. Malapert, A.; Provillard, J. Puzzle—Solving the n-Fractions Puzzle as a Constraint Programming Problem. *INFORMS Trans. Educ.* **2018**, *19*, 48–55. [CrossRef]
23. Knill, O. Some experiments in number theory. *arXiv* **2016**, arXiv:1606.05971.
24. Pollard, J. Monte Carlo methods for index computation (mod p). *Math. Comput.* **1978**, *32*, 918–924.
25. Kostopoulos, G. An Original Numerical Factorization Algorithm. *J. Inf. Assur. Cyber Secur.* **2016**, *2016*, 775081. [CrossRef]
26. Pollard, J. Theorems on factorization and primality testing. *Proc. Camb. Philos. Soc.* **1974**, *76*, 521–528. [CrossRef]
27. McKee, J. Turning Euler's factoring method into a factoring algorithm. *Bull. Lond. Math. Soc.* **1996**, *28*, 351–355. [CrossRef]
28. Scripcariu, L.; Frunza, M.D. A New Character Encryption Algorithm. In Proceedings of the ICMCS 2005, Montreal, QC, Canada, 14 August 2005; pp. 83–86.
29. Wiener, M. Cryptanalysis of short RSA secret exponents. *IEEE Trans. Inf. Theory* **1990**, *160*, 553–558. [CrossRef]
30. McKee, J.; Pinch, R. Old and new deterministic factoring algorithms. *Algorithmic Number Theory* **2005**, 217–224. [CrossRef]
31. Overmars, A.; Venkatraman, S. A new method of golden ratio computation for faster cryptosystems. In Proceedings of the IEEE Cybersecurity and Cyber forensics Conference, London, UK, 21–23 November 2017.
32. Karatsuba, A. The complexity of computations. *Proc. Steklov Inst. Math.* **1995**, *211*, 169–183.
33. 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]