
**RSA CRYPTOGRAPHY IN ORDER TO BE FORMALIZED IN KURT
GÖDEL'S ALGEBRA, AND A STUDY OF PRIVATE AND PUBLIC
KEYS AND THE ASSUMPTION OF CALCULATING THE INVERSE
FOR PERFORMING THE DECODING IS DEFINED BY, $3(4K + 3) = D$.**

***Welken Charlois Goncalves**

Brazil.

Article Received: 13 August 2026

*Corresponding Author: Welken Charlois Goncalves

Article Revised: 03 September 2026

Brazil.

Published on: 23 September 2026

DOI: <https://doi-doi.org/101555/ijrpa.2562>

SUMMARY

The intention of the article is to ratify the RSA Cryptography article in [1] and [2], so that we have been working with recursive mathematical formulas and equations. A recursive equation defines each term of a mathematical sequence using the value of one or more previous terms, necessarily accompanied by an initial condition. There is no simple formula or pattern that predicts exactly what the next prime number will be, but mathematics finds fascinating patterns and behaviors in them. Almost all are odd: With the exception of the number 2, all primes are odd numbers, since any other even number would be divisible by 2. Except for 2 and 3, every prime number greater than 3 ends in 1, 3, 7, or 9. All prime numbers greater than 3 leave a remainder of 1, 5, 7, or 11 when divided by 12. In other words, they have the form $12n \pm 1$ or $12n \pm 5$. As the numbers get larger, primes become rarer. The Prime Number Theorem shows that the chance of a large number n being prime is approximately proportional to $1/\ln(n)$ (where $\ln$ is the natural logarithm). When you arrange integers in a spiral shape, prime numbers form surprising and visible diagonal lines, proving that there is a hidden order in their distribution. In 2016, mathematicians discovered that primes have a "rebellious" preference for not repeating the last digit in the next prime (for example, a prime ending in 1 is rarely followed by another ending in 1 immediately). Goldbach's Conjecture states that every even integer greater than 2 can be written as the sum of two prime numbers. If you choose any even number above 2 (such as 4, 6, 8, 10...), it results from the sum of two prime numbers (numbers divisible only by 1 and themselves). There is a variation called Goldbach's Weak Conjecture, which states that every odd number greater than 7 is the sum of

three primes. This weak version was proven true in 2013. According to Gödel's incompleteness theorems, any consistent formal mathematical system capable of expressing basic arithmetic has insurmountable limits. Published by the Austrian logician Kurt Gödel in 1931, these two theorems revolutionized mathematics and philosophy by demonstrating that truth is broader than probability.

KEYWORDS: RSA cryptography, Axioms, Algebra, Parameterization, Kurt Gödel's Incompleteness Theorem.

INTRODUCTION AND THEORETICAL FRAMEWORK

Recursive formula:
The recursive formula defines a term using previous terms.

General form:

$$a_n = f(a_{n-1}, a_{n-2}, \dots)$$

Example:

$$a_n = a_{n-1} + 3, \quad a_1 = 2$$

Then

$$a_2 = 2 + 3 = 5$$

$$a_3 = 5 + 3 = 8$$

$$a_4 = 8 + 3 = 11$$

So the sequence is:

$$2, 5, 8, 11, \dots$$

Characteristic equation:

Outside linear homogeneous recurrence, we can use a characteristic equation to find an explicit formula.

Example: $a_n = 3a_{n-1} - 2a_{n-2}$

Assume:

$$a_n = r^n$$

Substitute:

$$r^n = 3r^{n-1} - 2r^{n-2}$$

Divide by r^{n-2} :

$$r^2 - 3r + 2 = 0$$

This is the characteristic equation. Factor:

$$(r - 1)(r - 2) = 0$$

Therefore:

$$r_1 = 1, \quad r_2 = 2$$

The general solution is:

$$a_n = C_1 (1)^n + C_2 (2)^n$$

or simply:

$$a_n = C_1 + C_2 2^n$$

The constants C_1, C_2 are found using the **initial conditions**.

Quick distinction

Concept	Meaning
Recursive formula	Calculates the next term from previous terms
Characteristic equation	Helps solve a linear recurrence
Explicit formula	Gives a_n directly without calculating previous terms

Key pattern:

Recurrence → Characteristic equation → Explicit formula

Hilbert, Cantor, Peano and Gödel:

These four mathematicians were central to the development of modern mathematical logic and foundations of mathematics.

Mathematician	Main idea	Simple explanation
---------------	-----------	--------------------

David Hilbert	Axiomatic method & formalism	Mathematics should be built from clear axioms and rules of deduction. He wanted to prove that mathematics could be made consistent and complete.
Georg Cantor	Set theory	Developed the mathematical theory of sets and showed that there are different sizes of infinity. For example, the natural numbers are infinite, but the real numbers form a larger infinity.
Giuseppe Peano	Axioms of arithmetic	Formalized the natural numbers using a small set of axioms, now called the Peano axioms. They provide a foundation for arithmetic.
Kurt Gödel	Incompleteness theorems	Showed that sufficiently powerful consistent formal systems cannot prove every mathematical truth within the system. This limited Hilbert's original program.

The connection:

A useful way to remember them is:

Cantor → Sets

Peano → Natural numbers

Hilbert → Axioms and formal systems

..

Gödel → Limits of formal systems

In one sentence

Cantor built the language of sets, Peano formalized arithmetic, Hilbert tried to put mathematics on a rigorous axiomatic foundation, and Gödel demonstrated fundamental limits to what such formal systems can prove.

Gödel's Incompleteness Theorems

1. First Incompleteness Theorem

For any consistent formal mathematical system that is sufficiently powerful to express basic arithmetic, there are statements that are true but cannot be provided within that system.

In simple terms:

There are mathematical truths that the system cannot prove.

Gödel achieved this by constructing a statement that essentially says: “This statement cannot be proven in this system.”

If the system is consistent, that statement cannot be proven inside the system.

2. Second Incompleteness Theorem

A sufficiently powerful, consistent formal system cannot prove its own consistency using only the methods available within that system.

In symbols, roughly:

$$\text{Con}(S) \notin \text{Provable}(S)$$

where S is the formal system. In simple terms:

A consistent system cannot establish its own consistency.

The big idea

Before Gödel, Hilbert hoped that mathematics could be placed on a completely rigorous foundation where we could establish:

1. Consistency— no contradictions.
2. Completeness— every mathematical statement can be proven or disproved.
3. Decidability— there is a systematic way to determine mathematical truth. Gödel's results showed fundamental limitations on this program:

$$\text{Consistency} \neq \text{Completeness}$$

and

A system cannot generally prove its own consistency.

Important: Gödel's theorems do not say that mathematics is inconsistent or that nothing can be proven. They concern the limitations of sufficiently powerful formal systems.

Number Theory

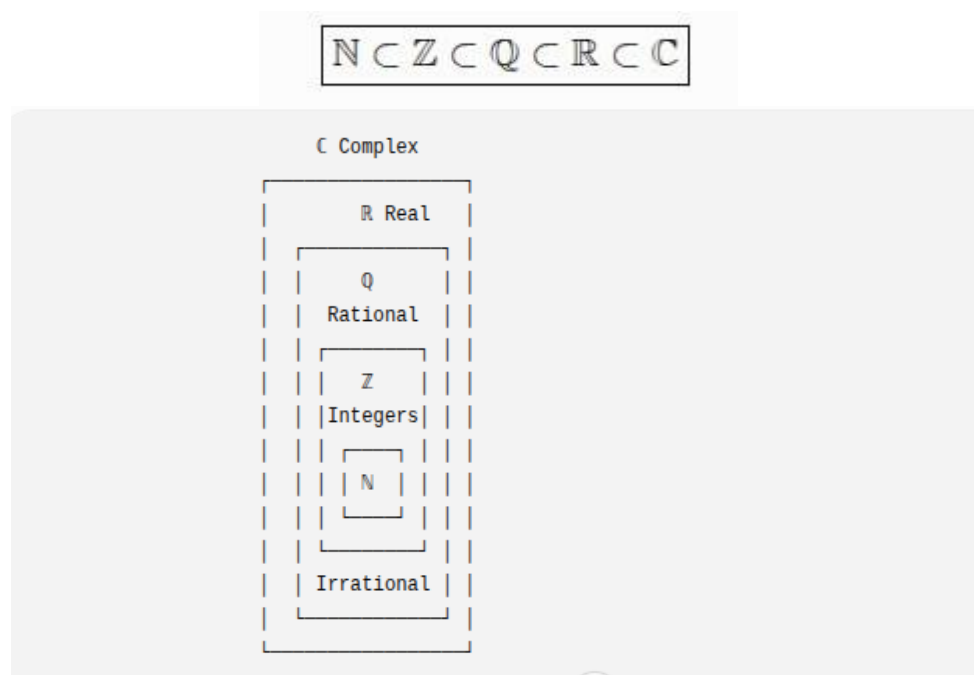
Number theory studies the properties and relationships of numbers. Natural numbers are the numbers used for counting, such as 0, 1, 2, 3, and so on. Integers include natural numbers, zero, and negative whole numbers. Prime numbers are integers greater than 1 that have exactly two

positive divisors: 1 and the number itself, such as 2, 3, 5, 7, and 11. Every integer greater than 1 can be uniquely decomposed into a product of prime numbers.

Rational numbers are numbers that can be expressed as a fraction of two integers. Irrational numbers cannot be expressed as a fraction of two integers. Their decimal representations are infinite and non-repeating.

Real numbers consist of both rational and irrational numbers. They can be represented on the real number line. Complex numbers extend the real numbers by introducing the imaginary unit. A complex number has the form $a+bi$ where a and b are real numbers.

The main hierarchy of these systems is:



Thus, each larger number system extends the previous one, allowing mathematics to represent increasingly broad types of quantities.

Number system	Main idea	Examples
$\mathbb{N}$ Natural	Counting numbers	0, 1, 2, 3
$\mathbb{Z}$ Integers	Positive, negative and zero	-3, 0, 5
$\mathbb{Q}$ Rational	Fractions of integers	$1/2, -3/4, 5$
Irrational	Not expressible as a fraction of integers	$\sqrt{2}, \pi, e$
$\mathbb{R}$ Real	Rational + irrational	$2, \frac{1}{3}, \sqrt{2}, \pi$
$\mathbb{C}$ Complex	Real + imaginary numbers	$3 + 2i, -i$

Mathematical Induction

Mathematical induction is a proof method used to show that a statement is true for every natural number.

It has two main steps:

1. Base case: Prove that the statement is true for the first value, usually $n=1$.
2. Inductive step: Assume that the statement is true for $n=k$. This is called the inductive hypothesis. Then prove that it must also be true for $n=k+1$.

If both steps are correct, the statement is true for all natural numbers. Example:

$$1 + 2 + 3 + \dots + n = \frac{n(n+1)}{2}$$

The base case is checked for $n=1$. Then, assuming the formula is true for $n=k$, we prove it for $n=k+1$. Therefore, the formula holds for every natural number n .

Mathematical Deduction

Mathematical deduction is a method of reasoning in which a conclusion is logically derived from known facts, definitions, axioms, or previously proven results.

A deductive argument generally follows this structure:

- Start with accepted premises.
- Apply logical rules or mathematical properties.
- Reach a necessary conclusion. Example:
- All even numbers are divisible by 2.
- 18 is an even number.

- Therefore, 18 is divisible by 2.

Difference Between Induction and Deduction	
Induction	Deduction
Proves a statement for an infinite sequence of natural numbers.	Derives a specific conclusion from general facts.
Uses a base case and an inductive step.	Uses logic, definitions, axioms, and theorems.
Commonly proves formulas and patterns.	Commonly proves relationships and consequences.

Both induction and deduction are essential methods of mathematical reasoning and help establish the validity of mathematical statements.

DISCUSSION

You articles it works initially with relatively small numbers — such 391, 437 and 529 — and states that the analysis could be extended to much larger numbers, including numbers with 200 digits or more. The article consists of a preliminary analysis of RSA cryptography decoding, starting with small numbers and then generalizing to larger, even very large, numbers. It basically presents two programs, the first based on the reference [2] in which numbers in base 2 or any other integer raised to prime numbers are written and the reference [4] in prime numbers in the form $4k+1$ and $4k+3$ which multiplied generate forms $4k+1$. The article is basically a pre-analysis for the assumption of calculating the inverse for the decoding defined by, $3(4k+3) = d$, [2], and the calculation of the decoding blocks using the private key. [1]. Do give yourself generate examples by following the same mathematical idea as the article But with large numbers. For example:

Larger examples

1. Product of two prime factors Then:

$$p = 1000000007$$

$$q = 1000000009$$

And we can verify this:

$$n = pq = 10000000160000000063$$

Total large primes stored: 10000
Smallest stored prime: 99816439Largest: 99999989

Building RSA module and recovering private key d ---

p = 99816439
q = 99816439
n = p*q = 9963321494640721
phi(n) = 9963321295007844
public e = 5
*** GENERAL CASE – modular inverse via extended Euclid ***
d = modinv(e,phi) = 1992664259001569
e*d mod phi = 1
*** SUCCESS: d is the correct private key ***
99816439 ≡ 3 (mod 4) → form 4k+3
99816439 ≡ 3 (mod 4) → form 4k+3
9963321494640721 ≡ 1 (mod 4) → form 4k+1

p = 99816439
q = 99816449
n = p*q = 9963322492805111
phi(n) = 9963322293172224
public e = 5
*** GENERAL CASE – modular inverse via extended Euclid ***
d = modinv(e,phi) = 1992664458634445
e*d mod phi = 1
*** SUCCESS: d is the correct private key ***
99816439 ≡ 3 (mod 4) → form 4k+3
99816449 ≡ 1 (mod 4) → form 4k+1
9963322492805111 ≡ 3 (mod 4) → form 4k+3

p = 99816439
q = 99816461
n = p*q = 9963323690602379
phi(n) = 9963323490969480
public e = 7
*** GENERAL CASE – modular inverse via extended Euclid ***
d = modinv(e,phi) = 4269995781844063
e*d mod phi = 1
*** SUCCESS: d is the correct private key ***
99816439 ≡ 3 (mod 4) → form 4k+3
99816461 ≡ 1 (mod 4) → form 4k+1
9963323690602379 ≡ 3 (mod 4) → form 4k+3

p = 99816439
q = 99816499
n = p*q = 9963327483627061
phi(n) = 9963327283994124
public e = 5
*** GENERAL CASE – modular inverse via extended Euclid ***
d = modinv(e,phi) = 1992665456798825
e*d mod phi = 1
*** SUCCESS: d is the correct private key ***
99816439 ≡ 3 (mod 4) → form 4k+3
99816499 ≡ 3 (mod 4) → form 4k+3
9963327483627061 ≡ 1 (mod 4) → form 4k+1

p = 99816439
q = 99816553
n = p*q = 9963332873714767
phi(n) = 9963332674081776
public e = 5
*** GENERAL CASE – modular inverse via extended Euclid ***
d = modinv(e,phi) = 7970666139265421
e*d mod phi = 1
*** SUCCESS: d is the correct private key ***
99816439 ≡ 3 (mod 4) → form 4k+3
99816553 ≡ 1 (mod 4) → form 4k+1
9963332873714767 ≡ 3 (mod 4) → form 4k+3

p = 99816439
q = 99816581
n = p*q = 9963335668575059
phi(n) = 9963335468942040
public e = 7
*** GENERAL CASE – modular inverse via extended Euclid ***
d = modinv(e,phi) = 2846667276840583
e*d mod phi = 1
*** SUCCESS: d is the correct private key ***
99816439 ≡ 3 (mod 4) → form 4k+3
99816581 ≡ 1 (mod 4) → form 4k+1
9963335668575059 ≡ 3 (mod 4) → form 4k+3

“Finish with the conclusion with the first program of numbers 23 and 19 written as,

$$\begin{aligned} & (2^4 + 2^3 - 1)(2^4 + 3) \\ &= 2^8 + 3 \cdot 2^4 + 2^7 + 3 \cdot 2^3 - 2^4 - 3 \\ &= 256 + 48 + 128 + 24 - 16 - 3 \\ &= 437 \end{aligned}$$

[1], so the program becomes easier to generalize to larger numbers of prime factors for multiplication. Finally, to continue the analysis, the big question of decoding and determining the parameters of the inverse in which decoding becomes possible $3(4k+3)=d$, [2], that is, the last step is the analysis of the items mentioned in the conclusion in order to report the form of the reference". by, $3(4k + 3) = d$, [2], and the calculation of the decryption blocks through the "Private key." [3]

Why Python Can Be Better Than Fortran for Very Large-Number Calculations

For calculations involving very large or arbitrary-precision integers, Python can be more convenient than traditional Fortran because Python's built-in `int` type supports arbitrarily large integers. The number of digits is mainly limited by available computer memory.

For example, Python can directly calculate:

```
n = 10**1000
print(n)
```

This produces an integer with 1,001 digits without requiring a special integer declaration.

Traditional Fortran integer types, such as `INTEGER(KIND=8)` or `INTEGER(KIND=16)` have a fixed number of bits. Once the maximum representable value is exceeded, the calculation can overflow. `INTEGER(KIND=16)` can provide a very large fixed range on compilers that support it, but it is still not arbitrary precision. For huge numbers, Python can also use libraries such as `gmpy2`, which provide highly optimized multiple-precision arithmetic. However, Fortran is often faster for large quantities of ordinary fixed-size numerical calculations, especially scientific and numerical simulations. Therefore, the distinction is important:

- Python: excellent for arbitrary-precision integers and very large exact integers.
- Fortran: excellent for high-performance numerical computation with fixed-size numbers.
- Python + GMP/gmpy2: particularly useful when numbers have thousands, millions, or

more digits and exact integer arithmetic is required.

So, it is not correct to say that Python is always faster than Fortran. Python is generally more convenient for calculations involving integers larger than standard machine integer limits, while Fortran can be substantially faster for conventional numerical calculations within fixed precision. Example of a Modular Division program for Larger and Gigantic Numbers in Python:

[do a = 100,026](iteration 27)
=====
===
100,026 is not prime.
Congruences:
$100,026 \equiv 4 \pmod{26}$
$100,026 \equiv 15 \pmod{37}$
$100,026 \equiv 27 \pmod{41}$
$100,026 \equiv 19 \pmod{97}$
$100,026 \equiv 36 \pmod{101}$
$100,026 \equiv 53 \pmod{257}$
$100,026 \equiv 135 \pmod{1,009}$
$100,026 \equiv 9,963 \pmod{10,007}$
$100,026 \equiv 23 \pmod{100,003}$
$100,026 \equiv 100,026 \pmod{1,000,003}$
Modular pot example: $3^{(100,026 \bmod 100000)} \equiv 865,810,542 \pmod{1000000007}$
=====
===
[do a = 100,027](iteration 28)
=====
===
100,027 is not prime.
Congruences:
$100,027 \equiv 5 \pmod{26}$
$100,027 \equiv 16 \pmod{37}$
$100,027 \equiv 28 \pmod{41}$
$100,027 \equiv 20 \pmod{97}$
$100,027 \equiv 37 \pmod{101}$
$100,027 \equiv 54 \pmod{257}$
$100,027 \equiv 136 \pmod{1,009}$
$100,027 \equiv 9,964 \pmod{10,007}$
$100,027 \equiv 24 \pmod{100,003}$
$100,027 \equiv 100,027 \pmod{1,000,003}$

Modular pot example: $3^{(100,027 \bmod 100000)} \equiv 597,431,612 \pmod{1000000007}$
=====
====
[do a = 99,878,941](iteration 99778942)
=====
====
*** 99,878,941 IS A COUSIN ***
Congruences: $99,878,941 \equiv 19 \pmod{26}$
$99,878,941 \equiv 31 \pmod{37}$
$99,878,941 \equiv 30 \pmod{41}$
$99,878,941 \equiv 78 \pmod{97}$
$99,878,941 \equiv 41 \pmod{101}$
$99,878,941 \equiv 3 \pmod{257}$
$99,878,941 \equiv 49 \pmod{1,009}$
$99,878,941 \equiv 9,081 \pmod{10,007}$
$99,878,941 \equiv 75,947 \pmod{100,003}$
$99,878,941 \equiv 878,644 \pmod{1,000,003}$
Modular pot example: $3^{(99,878,941 \bmod 100000)} \equiv 934,405,175 \pmod{1000000007}$
=====
====
[do a = 99,878,942](iteration 99778943)
=====
====
99,878,942 is not a prime number.
Congruences: $99,878,942 \equiv 20 \pmod{26}$
$99,878,942 \equiv 32 \pmod{37}$
$99,878,942 \equiv 31 \pmod{41}$
$99,878,942 \equiv 79 \pmod{97}$
$99,878,942 \equiv 42 \pmod{101}$
$99,878,942 \equiv 4 \pmod{257}$
$99,878,942 \equiv 50 \pmod{1,009}$
$99,878,942 \equiv 9,082 \pmod{10,007}$
$99,878,942 \equiv 75,948 \pmod{100,003}$
$99,878,942 \equiv 878,645 \pmod{1,000,003}$
Modular pot example: $3^{(99,878,942 \bmod 100000)} \equiv 803,215,511 \pmod{1000000007}$
=====
====
[do a = 99,878,943](iteration 99778944)
=====
====
99,878,943 is not a prime number.

Congruences: $99,878,943 \equiv 21 \pmod{26}$
$99,878,943 \equiv 33 \pmod{37}$
$99,878,943 \equiv 32 \pmod{41}$
$99,878,943 \equiv 80 \pmod{97}$
$99,878,943 \equiv 43 \pmod{101}$
$99,878,943 \equiv 5 \pmod{257}$
$99,878,943 \equiv 51 \pmod{1,009}$
$99,878,943 \equiv 9,083 \pmod{10,007}$
$99,878,943 \equiv 75,949 \pmod{100,003}$
$99,878,943 \equiv 878,646 \pmod{1,000,003}$
Modular pot example: $3^{(99,878,943 \bmod 100000)} \equiv 409,646,519 \pmod{1000000007}$

The intentionIt was to demonstrate the results of the programs for numbers.biggerIn previous articles [1] and [2], this makes it possible to formalize the RSA Cryptography Theory, Public and Private Keys, and its Inverse Key Decoder. With the demonstration of the program results, the resolution of the most complex problem in Mathematics becomes understandable, encompassing the abstraction for all the most complex problems in Mathematics, such as the examples in the Introduction and Theoretical Framework, of recursive formulas, Gödel's Incompleteness Theorem, the Theories of Cantor, Peano and Hilbert, and the Theory of Induction andMathematical deduction, to be applied in the construction of the formalization of the solution to the RSA cryptography problem and its inverse decoder, as well as problems in abstract algebra, serve as support for the analysis of solutions to complex problems in mathematics, considering the qualitative and quantitative properties of numbers in the number theory of mathematics. And also analyze that numbers are written in base 2 or any other integer raised to prime numbers and the reference [4] in prime numbers in the form $4k+1$ and $4k+3$ which, when multiplied, generate forms $4k+1$, in order to abstract and arrive at an algebraic formalization of RSA cryptography, studying and analyzing various and all possibilities of prime numbers and number theory.Using the theory of Modular Algebra simplifies the problem considerably for larger and large numbers, where one can proceed to the base-2 solution of mathematical problems, as explained in [1] and [2].

CONCLUSION

The intention of this article is to provide a didactic and more accessible approach to mathematical language so that the outcome of solving exercises leads to a conclusion.Consistency

— no contradictions, ccompleteness— every mathematical statement can be proven or disprovededecidability— there is a systematic way to determine mathematical

truth. Showing that sufficiently powerful consistent formal systems cannot prove every mathematical truth within the system. This limited Hilbert's original program. Showed that sufficiently powerful consistent formal systems cannot prove every mathematical truth within the system. This limited Hilbert's original program. Formalizing a mathematical theory using logic and algebra, with the aim of solving unsolvable problems, as through the logic of Kurt Gödel. and Abstract Algebra, In other words, formally writing a mathematical theory. For specific or various problems in number theory in mathematics and even physics. Through this algebraic formulation with number theory, more difficult problems are solved in a simpler way and with a broader and more understandable abstraction for the mathematical intellect, without going through the difficulties of deduction and induction hypotheses in problem solving, abstracting all concepts in the best and most useful way that mathematics offers with its tools. This also creates new paths for problem solving, such as conjectures and theorems.

BIBLIOGRAPHIC REFERENCES

1. Preliminary Analysis of the RSA Encryption Decoding Break, for Small Numbers, in Order to Generalize to Larger Numbers, Up to Very Large Numbers.; Welken Charlois Gonçalves; <https://loja.editoradialetica.com/ciencias-exatas-e-tecnologias/ciencias-em-sintoniaexplorando-conexoes-entre-exatas-e-naturais-volume-3?srsId=AfmBOoo04UDE1ozyiHwrCKgYHMY-22f79qzp46CeCxN6H-Ug6Zzu3N2H>
2. Welken Charlois Gonçalves. A polynomial solution for factorial problems starting from the complexity of solving the traveling salesman problem, combinatorial analysis. Mathematics and its applications: resources and strategies for effective teaching - AYA Editora. <https://ayaeditora.com.br/livros/L384.pdf>
3. A Study in the Linear Algebra of Congruence and RSA Encryption, a Ratification of the article Pre-Analysis of the Breaking of RSA Encryption Decoding, for Small Numbers, in Order to Generalize to Larger Numbers, Even Very Large Numbers WCGoncalvesMay20,2024, A Concept Complement in the Article for Solving the Congruence Calculation for the Calculation of the Inverse for Performing the Decoding Defined by, $3(4k + 3) = d$, [2], and the Calculation of the Decoding Blocks using the private key, a Complement. Welken Charlois Gonçalves; <https://doi.org/10.55248/gengpi.6.0325.1219>
<https://ijrpr.com/uploads/V6ISSUE3/IJRPR40215.pdf>
4. Ian Stewart; The Greatest Mathematical Problems; Zahar Publishing House; Chapter 7; ISBN 978-85-378-1310-2.
5. <http://lattes.cnpq.br/7210808817928708>