International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

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 WC Goncalves May 20, 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 Goncalves

Brazil, graduated in Physics and Postgraduate in Information Security.

Summary:

This article studies twin prime numbers based on the analysis of Goldbach's Conjecture of 1742. The consideration of Goldbach's theory can be considered for a better understanding of the Theory of Modular Algebra. Only this point should be considered in Goldbach's Theory. A study is carried out on the congruences of any numbers and, specifically, the logic of the congruences of prime numbers and their modules based on the theory of modular algebra, with the aim of optimizing the May 2024 article on breaking RSA encryption. With a simple computer program in Fortran language in Linux Gfortran with a Core i7 processor and 16GB of RAM, without considering AMD RYZEN or Intel Core 9 14th generation processors, this processor note serves as a ratification of previous articles that used Windows with an 8GB Core i5 Processor in the Fortran Force Lepsch Program, there is a considerable difference, and the Fortran language on an online conversion site we can convert to JAVA, C and C++ or any other language that is viable for programming, or PYTHON in which we have today in this year of 2025 the help of ChatGTP artificial intelligence, and finally considering that the use of Linux for Information Security is much more viable than Windows and Mac. At the end, the results of the program are exposed, in a future article we will have the resolution of the first article [2] that generalizes to all types of RSA Encryption decoding break [1]. The articles also serve for a better understanding and analysis in the study of all types of decoding of countless types and equivalences of cryptography existing in the current world, as well as a broad understanding for the creation of new types of encryption and key decoding, and in addition to the analysis of passwords, encrypted messages, decoded encrypted passwords and credit card transactions, the latter uses this decoding method of the inverse of RSA Encryption [1], in the world of Ethical Hacking of Information Security, with its due references cited, as an example, either for breaking encryption in deciphering a decoded alphabet, or for an analysis of a brute force attack such as the Kali Linux program from Ethical Hacking, such as Hydra.

Introduction:

The Fortran language is adequate for a simple program, otherwise we could use Java or C++. Christian Goldbach's conjecture of 1742 fits into the formulation of Riemman's hypothesis and there is a brief explanation in Keith Devlin's books The Millennium Problems and Marcus du Sautoy's The Music of Prime Numbers [1] and [2], respectively. This article deals with the calculation of congruences in modular algebra, a brief demonstration of the theory of modular algebra is given and the computational calculus is demonstrated, emphasizing that this demonstration is a completion of the article Pre-analysis of breaking the decoding of RSA encryption, for small numbers, in order to generalize to larger numbers, up to very large numbers welken charlois gonçalves, as an introduction and a prerequisite to deepen the conclusion of the same article cited above and to use the inverse calculation method to perform the decoding defined by, 3(4k + 3) = d, [2], and the calculation of the decoding blocks through the private key. It is demonstrated with numbers smaller than 10000, but it can be extended to numbers above 100000 easily.

Theoretical framework:

The concepts necessary for understanding this article come from linear algebra, based on a national introductory mathematics book called Numbers, an Introduction to Mathematics by Cesar Polcino Milies and Sonia Pitta Coelho in Chapter 3 Page 97 in Sections 3.1 and 3.2 Linear Diophantine Equations and Congruences, or any other book on Number Theory. Based on the concepts of Diphanto of Alexandria in the ancient Greek period around 250 BC, what interests us are the congruences in chapter 3.2 in which the perfect numbers that led to the study of the form to $2^n - 1$look for their divisors. More generally, we can think of studying numbers of the form $a^n - 1$, with aez[3].

In a 1640 letter to Bernhard Frenicle de Bessy, Fermart announced a surprising result: if p is a prime and is an integer that is not divisible by p, then p divides. I$a^{p-1} - 1$n the same letter, he commented: "I would send you the proof if I didn't think it was too long." The first proof of this result, known as "Fermat's little theorem" (to distinguish it from Fermat's great theorem), was published in 1736, almost a century later, by Euler. Euler later gave other proofs of the same result. In one of them, he frequently uses the "results of divisions by p", which gave rise to the theory of congruences. This method of work was also used by Lagrange and Legendre, but it only became explicit in Gauss's disquitonnes, in which the precise definition and symbolism used today appear.

We will see in the examples how the introduction of this synthetic notation simplifies the study of many divisibility questions.

Definition 3.2.1

Be m≠0 is a fixed integer. Two integers a and b are said to be congruent modulo m divides the difference ab.

In this case, we write the ≡ b (mod m). To indicate that a and b are not congruent modulo m, we will write≢(mod m). (Gauss writes in the disquisitions that he was introduced to using the symbol ≡due to the great analogy with algebraic equality.)

With our definition, the≡b (mod m) if and only if m | (ab), or, equivalently, if there exists an integer q such that a=b+mq.

Since m|(ab) if and only if |m| |(ab), we will limit ourselves to considering the case where m>0.

For example, 5≡9 (mod 2) and also 5≡9 (mod 4). In fact, it is easy to verify that two numbers are congruent modulo 2 if and only if they are both even or both odd.

We can give another characterization of the notion of congruence.

3.2.2 Proposition

Let m be a fixed integer. Two integers a and b are congruent modulo m if and only if they have the same integer remainder when divided by m.

3.2.4 Proposition

Let m be a fixed integer and let a, b, and c be arbitrary integers. If gcd(c, m)=1, then ac≡bc (mod m), then ac≡ bc (mod m) implies the ≡ b (mod m).

Demonstration

If you ac≡ bc (mod m), we have that m | (ABC.

Since gcd(c,m) = 1, from Euclid's Theorem (2.3.7) it follows that m |(ab), whence≡ b (mod m).

We note in passing that if gcd (c,m) = d ≠1, there always exist integers a and b such that a ≢b (mod m), but ac≡bc (mod m); if d=m , that is , if c ≡ 0 (mod m), then, for arbitrary integers a and b we have that ac ≡ bc (mod m), regardless of whether a and b are congruent modulo m; if d < m, writing

m=kd

c=k'd,

It is necessary to

k≢0 (mod m), but ck≡c.0 (mod m), since ck = k`dk =k`m.

3.2.5 Example

Let's determine the remainder of the division of5^{60} by 26.

Writing ,$5^{60} \equiv 26q + r$ the problem is equivalent to determining the integer r such that 0≤r ≤ 25 and such that $5^{60} \equiv r\ (mod\ 26)$.

We noticed that5^2=25, that is,5^2 ≡-1 (mod 26). Using part (vii) of proposition 1.3.3, we have that 5^4≡$(-1)^2$(mod 26), that is, 5^4≡1 (mod 26).

Finally,$5^{60} = (5)^{15}$, therefore,$5^{60} \equiv (-1)^{15}\ (mod\ 26)$, whence the remainder of the division of $5^{60}$by 26 is 1.

In this example 3.2.5 and the essence of the problem of the program of the articles and of the other two articles cited in this article for the final resolution article of the Problem of Decoding the Private Key of RSA Encryption as that of 5^{60} by 26.

It is written $5^{60} \equiv 26q + r$

which is equivalent to determining the integer r such that 0 ≤ r ≤ 25 and such that

$5^{60} \equiv r\ (mod\ 26)$.

Exactly on page 73 of the article [1]:

“It only remains to check if n = pq, where p and q are prime numbers, performing the pattern of primes 4K + 1 and 4K + 3, which are n, the final number, which are generated by multiplications of (4k + 1) x (4k + 1) generates a new 4k + 1, (4K + 3) x (4k + 3), generates 4k + 1, and (4k + 1) x (4k + 3) generates 4k + 3. To check numbers of the form 4K + 1 and 4K + 3, ONLY ONE PROGRAM”

And the article also says in the Conclusion of the article [1]:

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

$(2^4 + 2^3 - 1).(2^4 + 3) = 2^8 + 2^7 + 2.2^4 + 3.2^3 - 3 = 437$,

[5], 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 relate the form of the reference”.

Exactly the issue of ratification of the present article and the summary that explains the article[1]:

“The article is basically a pre-analysis for the assumption of calculating the inverse for performing the decryption defined by, 3(4k + 3) = d, [2], and the calculation of the decryption blocks through the private key.”

3.2.6 Example

Let's determine the units digit of 3^{100}.

Note that, in general,

$$a = a_n 10^n + a_{n-1} 10^{n-1} + a_1 10 + a_0$$

If , then the≡ a_0(mod 10). We must then determine a number x such that 0≤ x≤ 9 and 3^{100}≡x (mod 10).

Now ,3^2 ≡-1 (mod 10), therefore 3^4 ≡1 (mod 10) and therefore, 3^{100}≡ $(3^4)^{25}$ ≡1 (mod 10).

3.2.7 Example

In exercise 5 of 22, the reader determined criteria for a number to be divisible by 3, 9 and 11, based on its expression in base 10. We will show here how this discussion is simplified by introducing the language of congruences.

It is

$$a = a_n 10^n + a_{n-1} 10^{n-1} + a_1 10 + a_0$$

, with 0≤ a_i ≤9 for i=0,1,...,N and a_n≠0.

We initially notice that 10≡1 (mod 3), therefore, we have that 10^i ≡1 (mod 3), for every positive integer i, whence, using part (vi) of Proposition 3.2 we have that

$$a_i 10^i = a_i (mod\ 3), para 1 \leq i \leq n.$$

Adding all these congruences in order, we have

$a = a_n 10^n + a_{n-1} 10^{n-1} + a_1 10 + a_0 \equiv$ ≡

$a_n + a_{n-1} + \ldots + a_0 (mod\ 3)$,

That is, there exists an integer k such that

$a = a_n + a_{n-1} + \ldots + a_0 + 3k$. .

Consequently, a is a multiple of 3 if and only if the sum of its digits

$a = a_n + a_{n-1} + ... + a_0$ So be it.

As we will also have that 10≡ 1 (mod 9), reasoning identical to the previous one shows that a is a multiple of 9 if and only if the sum of its digits is.

Finally, as 10 ≡-1 (mod 11), we will have to

10^i≡-1 (mod 11) , if i is odd

10^i≡1 (mod 11), if i is even.

Multiplying by the corresponding digits and adding, as before, we have

$$a = a_n 10^n + a_{n-1} 10^{n-1} + ... + a_1 10 + a_0 =$$

$$(-1)^n a_n + (-1)^{n-1} a_{n-1} + ... + (-a_1) + a_0 \ (mod\ 3)$$

That is,

$$a \equiv (a_0 + a_2 + ...) - (a_1 + a_3 + ...) \ (mod\ 11)$$

Consequently, a is a multiple of 11 if and only if

$$11|(a_0 + a_2 + ...) - (a_1 + a_3 + ...) \ (mod\ 11)$$

Program:

```
program encryption
!!! Congruent Modular Calculation
implicit none
!!! real and integer numbers for loops and allocations in matrices
real a,b,c,restor,model,evax
!!! model= modular number
!!! restor is the remainder of the equation
integer :: d,k
integer :: i,j,m
integer :: mode,t
!!! mode=modular number
!!! remainder is the remainder of the equation
!!!Dimensions of Reals and Integers
dimension :: a(3000,3000), b(3000,30000), c(3000,30000),evax(3000,3000)
!!! evax(d,t) total complete number that will be divided
dimension :: m(3000,3000),k(3000,3000)
!!! Example
!!! Primes up to 3000 in linear congruence, you advise to start with
```

!!!50 or 100 on the first
!!!compilation and execution of the program.
!!! From Fermat's Little Theorem
!!! DATA SAVE FILES
open (22, file='prime.txt', status='replace')
open(11,file='data5.txt',status='replace')
open(369, file='result5.txt',status='replace')
a(1,1)=0
restor=0
!!! do=f,100 You can change the increment
!!! Allocation Ties
t=1
from d=1.3000
!!!from t=1.10
!!! here the predecessor term is called evax(d,t)=d
!!! here you can use infinitesimal increments for better analysis of calculations
evax(d,t)=d
restor=0
!!!conditional loop of size and largest calculated number
!!! It is advisable to start with 50 or 100 for the first compilation and execution of the program
from t=1.3000
a(d,t)=evax(d,t)+restor
!!! here are the modules of modular algebra division
do mode=1.50
!!!of the rest=1.7
!!!do t=1.75
!!! here is the definition of modular number and the rest.
model=mode
!!! You can change the increment to something other than 1, as a deeper example of more complex calculations.

```
!!!!advanced in Number Theory.
!!! setting evax(d,t)=a(d,t) for future calculations and allocations of the dimension loop
!!!evax(d,t)=a(d,t)
!!! equation evax(d,t) with remainder, therefore it is the previous term plus a remainder
!!! evax(d,t)+ remainder this is the congruence condition for the solution of the article
!!!a(d,t)=evax(d,t)+remainder
!!! It is always necessary to put a print in the program to check in the Fortran or Linux terminal
!!!where the calculations are.
!!!print*, evax(d,t)
!!!print*, a(d,t),d,t,rest
!!! Defining b(d,t) and c(d,t) mod and remainder between a(d,t) and model
!!! mod is the remainder between restor (restor x 1000 and model)
!!! model and restore were used because it is possible to use other increments without the rest of the loop
!!!It is very interesting to work with other increments and other numbers will be seen
!!!more in-depth calculations
b(d,t)=mod(a(d,t),model)
c(d,t)=mod(restor*1000,model)
!!! writing data to files 11 and 369
write (11,*)'*******************'
write(11,*) 'full number a(d,t)=',a(d,t),'allocation det=',d,t,'previous full number evax(d,t)=',evax(d,t),'remainder=', restor
write (11,*) 'remainder b(d,t)=',b(d,t),'complete number a(d,t)=',a(d,t),'module=', model
write (11,*) 'restor c(d,t)=',c(d,t),'resto=',restor,'modulo=',model
write(11,*) '********************'
write (11,*) '******'
!!! When the remainder c(d,t) is equal to b(d,t) it is equal in an if
if (c(d,t)==b(d,t)) then
write (369,*) 'remainder b(d,t)=',b(d,t),d,t,'full number a(d,t)=',a(d,t),'modulo model=',model
write (369,*) 'restor c(d,t)=',c(d,t),d,t,'resto=',restor,'modulo model=', model
write (369,*) '********************'
```

!!!end if
!!!START OF CALCULATIONS OF PRIME NUMBERS IE CALCULATE
!!!u=a(d,t)
k(d,t)=1
!!!HERE CALCULATES THE PRIME NUMBERS i AND CALCULATES
!!!BASE 2**IE PLACES IN VECTOR C(F) NOT YET
!!!REQUIRED IN THE ARTICLE
if (k(d,t) < 2) then
k(d,t) = 2
end if
if (a(d,t) < 3) then
end if
from i = 2.3000
from j = 2, i/j
if (mod(i,j) == 0) then
exit
end if
end of
!!!Prime calculation and printing of prime numbers
if (j > (i/j)) then
if (i==a(d,t)) then
write (369,*) '*****************************'
write (369,*) i,'IS PRIME IS PRIME',a(d,t),restor, model
!!! IT IS ALWAYS ADVISABLE TO PUT A PRINT ON THE
!!!PROGRAM TO CHECK
!!!IN THE TERMINAL OF THE FORTRAN OR LINUX PLATFORM
!!!WHERE CAN THE CALCULATIONS BE FOUND..
print*, i
!!! CALCULATING THE BASE RAISED TO THE PRIME FOR THE

```
!!!NEXT ARTICLE
m(d,t)=2**i
write (22,*) m(d,t),i
end if
end if
end of
end if
end of
!!! INCREMENT 1 OF THE REST
restor=restor+1
end of
end of
end program encryption
```

Result and discussions:

Consequently, from the Theoretical Framework, the article WC Gonçalves. A polynomial solution for factorial problems based on the complexity of solving the traveling salesman problem, combinatorial analysis. Mathematics and its applications: resources and strategies for effective teaching - aya editora and the article pre-analysis of breaking the decoding of rsa encryption, for small numbers, in order to generalize to larger numbers, up to very large numbers WC Gonçalves Doi 10.48021/978-65-270-2479-8-c4, the following results obtained in the program above and according to these two articles are applied to findthe calculation of the inverse to perform the decoding defined by, 3(4k + 3) = d, [2], and the calculation of the decoding blocks using the private key.From RSA Cryptography. The result of the program in the previous section serves to organize the conclusion of article [1] and article [2], being a support to organize the factorial of very large numbers, as an example starting from the primes 23 and 19 demonstrated with the citation of the bibliographic references of article [2] in the Theoretical Reference, reemphasizing:

Exactly the issue of ratification of the present article and the summary that explains the article[1]:

“The article is basically a pre-analysis for the assumption of calculating the inverse for performing the decryption defined by, 3(4k + 3) = d, [2], and the calculation of the decryption blocks through the private key.”

FORTRAN PROGRAM RESULTS AND OUTPUT DATA

(Output of result5.txt file):

```
Remainder b(d,t)= 0.00000000 1 6 complete number a(d,t)= 5.00000000 modulo model= 5.00000000
Remainder c(d,t)= 0.00000000 1 6 remainder= 5.00000000 modulo model= 5.00000000
**********************
********************************
        5 IS PRIME IS PRIME 5.00000000 5.00000000 5.00000000
Remainder b(d,t)= 5.00000000 1 6 complete number a(d,t)= 5.00000000 modulo model= 9.00000000
Remainder c(d,t)= 5.00000000 1 6 remainder= 5.00000000 modulo model= 9.00000000
********************
*****************************
        5 IS PRIME IS PRIME 5.00000000 5.00000000 9.00000000
Remainder b(d,t)= 5.00000000 1 6 complete number a(d,t)= 5.00000000 modulo model= 15.0000000
Remainder c(d,t)= 5.00000000 1 6 remainder= 5.00000000 modulo model= 15.0000000
********************
*****************************
        5 IS PRIME IS PRIME 5.00000000 5.00000000 15.0000000
Remainder b(d,t)= 5.00000000 1 6 complete number a(d,t)= 5.00000000 modulo model= 27.0000000
Remainder c(d,t)= 5.00000000 1 6 remainder= 5.00000000 modulo model= 27.0000000
********************
*****************************
        5 IS PRIME IS PRIME 5.00000000 5.00000000 27.0000000
Remainder b(d,t)= 5.00000000 1 6 complete number a(d,t)= 5.00000000 modulo model= 37.0000000
Remainder c(d,t)= 5.00000000 1 6 remainder= 5.00000000 modulo model= 37.0000000
********************
*****************************
        5 IS PRIME IS PRIME 5.00000000 5.00000000 37.0000000
Remainder b(d,t)= 5.00000000 1 6 complete number a(d,t)= 5.00000000 modulo model= 45.0000000
Remainder c(d,t)= 5.00000000 1 6 remainder= 5.00000000 modulo model= 45.0000000
********************
*****************************
        5 IS PRIME IS PRIME 5.00000000 5.00000000 45.0000000
Remainder b(d,t)= 0.00000000 1 7 complete number a(d,t)= 6.00000000 modulo model= 1.00000000
Remainder c(d,t)= 0.00000000 1 7 remainder= 6.00000000 modulo model= 1.00000000
********************
Remainder b(d,t)= 0.00000000 1 7 complete number a(d,t)= 6.00000000 modulo model= 2.00000000
Remainder c(d,t)= 0.00000000 1 7 remainder= 6.00000000 modulo model= 2.00000000
```

```
********************
Remainder b(d,t)= 0.00000000 1 7 complete number a(d,t)= 6.00000000 modulo model= 3.00000000
Remainder c(d,t)= 0.00000000 1 7 remainder= 6.00000000 modulo model= 3.00000000
********************
Remainder b(d,t)= 0.00000000 1 7 complete number a(d,t)= 6.00000000 modulo model= 6.00000000
Remainder c(d,t)= 0.00000000 1 7 remainder= 6.00000000 modulo model= 6.00000000
********************
Remainder b(d,t)= 6.00000000 1 7 complete number a(d,t)= 6.00000000 modulo model= 9.00000000
Remainder c(d,t)= 6.00000000 1 7 remainder= 6.00000000 modulo model= 9.00000000
********************
Remainder b(d,t)= 6.00000000 1 7 complete number a(d,t)= 6.00000000 modulus model= 18.0000000
https://www.Youtube.Com/watch?V=ljgjdsjh7gg&list=pltuonohycikx7p5xzfckjuwzvuudae1dw
Remainder c(d,t)= 6.00000000 1 7 remainder= 6.00000000 modulo model= 18.0000000
********************
Remainder b(d,t)= 6.00000000 1 7 complete number a(d,t)= 6.00000000 modulo model= 27.0000000
Remainder c(d,t)= 6.00000000 1 7 remainder= 6.00000000 modulo model= 27.0000000
********************
Remainder b(d,t)= 6.00000000 1 7 complete number a(d,t)= 6.00000000 modulo model= 37.0000000
Remainder c(d,t)= 6.00000000 1 7 remainder= 6.00000000 modulo model= 37.0000000
********************
Remainder b(d,t)= 0.00000000 1 8 complete number a(d,t)= 7.00000000 modulo model= 1.00000000
Remainder c(d,t)= 0.00000000 1 8 remainder= 7.00000000 modulo model= 1.00000000
********************
*****************************
      7 IS PRIME IS PRIME 7.00000000 7.00000000 1.00000000
Remainder b(d,t)= 1.00000000 1 8 complete number a(d,t)= 7.00000000 modulo model= 3.00000000
Remainder c(d,t)= 1.00000000 1 8 remainder= 7.00000000 modulo model= 3.00000000
********************
*****************************
      7 IS PRIME IS PRIME 7.00000000 7.00000000 3.00000000
```

```
Remainder b(d,t)= 0.00000000 1 8 complete number a(d,t)= 7.00000000 modulo model= 7.00000000
Remainder c(d,t)= 0.00000000 1 8 remainder= 7.00000000 modulo model= 7.00000000
********************
*****************************
        7 IS PRIME IS PRIME 7.00000000 7.00000000 7.00000000
Remainder b(d,t)= 7.00000000 1 8 complete number a(d,t)= 7.00000000 modulo model= 9.00000000
Remainder c(d,t)= 7.00000000 1 8 remainder= 7.00000000 modulo model= 9.00000000
********************
*****************************
        7 IS PRIME IS PRIME 7.00000000 7.00000000 9.00000000
Remainder b(d,t)= 7.00000000 1 8 complete number a(d,t)= 7.00000000 modulo model= 21.0000000
Remainder c(d,t)= 7.00000000 1 8 remainder= 7.00000000 modulo model= 21.0000000
********************
*****************************
        7 IS PRIME IS PRIME 7.00000000 7.00000000 21.0000000
Remainder b(d,t)= 7.00000000 1 8 complete number a(d,t)= 7.00000000 modulo model= 27.0000000
Remainder c(d,t)= 7.00000000 1 8 remainder= 7.00000000 modulo model= 27.0000000
********************
*****************************
        7 IS PRIME IS PRIME 7.00000000 7.00000000 27.0000000
Remainder b(d,t)= 7.00000000 1 8 complete number a(d,t)= 7.00000000 modulo model= 37.0000000
Remainder c(d,t)= 7.00000000 1 8 remainder= 7.00000000 modulo model= 37.0000000
********************
*****************************
        7 IS PRIME IS PRIME 7.00000000 7.00000000 37.0000000
```

FOR LARGER NUMBERS

```
Remainder b(d,t)= 16.0000000 56 41 complete number a(d,t)= 40.0000000 modulo model= 24.0000000
Remainder c(d,t)= 16.0000000 56 41 remainder= 40.0000000 modulo model= 24.0000000
********************
```

```
Remainder b(d,t)= 13.0000000 56 41 complete number a(d,t)= 40.0000000 modulo model= 27.0000000
Remainder c(d,t)= 13.0000000 56 41 remainder= 40.0000000 modulo model= 27.0000000
********************
Remainder b(d,t)= 10.0000000 56 41 complete number a(d,t)= 40.0000000 modulo model= 30.0000000
Remainder c(d,t)= 10.0000000 56 41 remainder= 40.0000000 modulo model= 30.0000000
********************
Remainder b(d,t)= 4.00000000 56 41 complete number a(d,t)= 40.0000000 modulo model= 36.0000000
Remainder c(d,t)= 4.00000000 56 41 remainder= 40.0000000 modulo model= 36.0000000
********************
Remainder b(d,t)= 3.00000000 56 41 complete number a(d,t)= 40.0000000 modulo model= 37.0000000
Remainder c(d,t)= 3.00000000 56 41 remainder= 40.0000000 modulo model= 37.0000000
********************
Remainder b(d,t)= 0.00000000 56 41 complete number a(d,t)= 40.0000000 modulo model= 40.0000000
Remainder c(d,t)= 0.00000000 56 41 remainder= 40.0000000 modulo model= 40.0000000
********************
Remainder b(d,t)= 40.0000000 56 41 complete number a(d,t)= 40.0000000 modulo model= 45.0000000
Remainder c(d,t)= 40.0000000 56 41 remainder= 40.0000000 modulo model= 45.0000000
********************
Remainder b(d,t)= 0.00000000 56 42 complete number a(d,t)= 41.0000000 modulo model= 1.00000000
Remainder c(d,t)= 0.00000000 56 42 remainder= 41.0000000 modulo model= 1.00000000
********************
*****************************
          41 IS PRIME IS PRIME 41.0000000 41.0000000 1.00000000
Remainder b(d,t)= 2.00000000 56 42 complete number a(d,t)= 41.0000000 modulo model= 3.00000000
Remainder c(d,t)= 2.00000000 56 42 remainder= 41.0000000 modulo model= 3.00000000
********************
*****************************
          41 IS PRIME IS PRIME 41.0000000 41.0000000 3.00000000
Remainder b(d,t)= 5.00000000 56 42 complete number a(d,t)= 41.0000000 modulo model= 9.00000000
Remainder c(d,t)= 5.00000000 56 42 remainder= 41.0000000 modulo model= 9.00000000
********************
*****************************
        41 IS PRIME IS PRIME 41.0000000 41.0000000 9.00000000
Remainder b(d,t)= 14.0000000 56 42 complete number a(d,t)= 41.0000000 modulo model= 27.0000000
Remainder c(d,t)= 14.0000000 56 42 remainder= 41.0000000 modulo model= 27.0000000
********************
*****************************
        41 IS PRIME IS PRIME 41.0000000 41.0000000 27.0000000
Remainder b(d,t)= 4.00000000 56 42 complete number a(d,t)= 41.0000000 modulo model= 37.0000000
Remainder c(d,t)= 4.00000000 56 42 remainder= 41.0000000 modulo model= 37.0000000
********************
*****************************
        41 IS PRIME IS PRIME 41.0000000 41.0000000 37.0000000
Remainder b(d,t)= 0.00000000 56 42 complete number a(d,t)= 41.0000000 modulo model= 41.0000000
Remainder c(d,t)= 0.00000000 56 42 remainder= 41.0000000 modulo model= 41.0000000
********************
*****************************
        41 IS PRIME IS PRIME 41.0000000 41.0000000 41.0000000
Remainder b(d,t)= 0.00000000 56 43 complete number a(d,t)= 42.0000000 modulo model= 1.00000000
Remainder c(d,t)= 0.00000000 56 43 remainder= 42.0000000 modulo model= 1.00000000
********************
Remainder b(d,t)= 0.00000000 56 43 complete number a(d,t)= 42.0000000 modulo model= 2.00000000
Remainder c(d,t)= 0.00000000 56 43 remainder= 42.0000000 modulo model= 2.00000000
********************
Remainder b(d,t)= 0.00000000 56 43 complete number a(d,t)= 42.0000000 modulo model= 3.00000000
Remainder c(d,t)= 0.00000000 56 43 remainder= 42.0000000 modulo model= 3.00000000
********************
Remainder b(d,t)= 0.00000000 56 43 complete number a(d,t)= 42.0000000 modulo model= 6.00000000
Remainder c(d,t)= 0.00000000 56 43 remainder= 42.0000000 modulo model= 6.00000000
********************
Remainder b(d,t)= 0.00000000 56 43 complete number a(d,t)= 42.0000000 modulo model= 7.00000000
Remainder c(d,t)= 0.00000000 56 43 remainder= 42.0000000 modulo model= 7.00000000
********************
Remainder b(d,t)= 6.00000000 56 43 complete number a(d,t)= 42.0000000 modulo model= 9.00000000
Remainder c(d,t)= 6.00000000 56 43 remainder= 42.0000000 modulo model= 9.00000000
********************
Remainder b(d,t)= 0.00000000 56 43 complete number a(d,t)= 42.0000000 modulo model= 14.0000000
Remainder c(d,t)= 0.00000000 56 43 remainder= 42.0000000 modulo model= 14.0000000
********************
Remainder b(d,t)= 6.00000000 56 43 complete number a(d,t)= 42.0000000 modulo model= 18.0000000
Remainder c(d,t)= 6.00000000 56 43 remainder= 42.0000000 modulo model= 18.0000000
********************
Remainder b(d,t)= 0.00000000 56 43 complete number a(d,t)= 42.0000000 modulo model= 21.0000000
Remainder c(d,t)= 0.00000000 56 43 remainder= 42.0000000 modulo model= 21.0000000
********************
Remainder b(d,t)= 15.0000000 56 43 complete number a(d,t)= 42.0000000 modulo model= 27.0000000
Remainder c(d,t)= 15.0000000 56 43 remainder= 42.0000000 modulo model= 27.0000000
********************
Remainder b(d,t)= 5.00000000 56 43 complete number a(d,t)= 42.0000000 modulo model= 37.0000000
Remainder c(d,t)= 5.00000000 56 43 remainder= 42.0000000 modulo model= 37.0000000
********************
Remainder b(d,t)= 0.00000000 56 43 complete number a(d,t)= 42.0000000 modulo model= 42.0000000
Remainder c(d,t)= 0.00000000 56 43 remainder= 42.0000000 modulo model= 42.0000000
********************
Remainder b(d,t)= 0.00000000 56 44 complete number a(d,t)= 43.0000000 modulo model= 1.00000000
Remainder c(d,t)= 0.00000000 56 44 remainder= 43.0000000 modulo model= 1.00000000
********************
```

****
43 IS A PRIME IS A PRIME 43.0000000 43.0000000

FOR SLIGHTLY LARGER NUMBERS.

1.00000000
Remainder b(d,t)= 1.00000000 56 44 complete number a(d,t)= 43.0000000 modulo model= 3.00000000
Remainder c(d,t)= 1.00000000 56 44 remainder= 43.0000000 modulo model= 3.00000000
****
****
43 IS PRIME IS PRIME 43.0000000 43.0000000 3.00000000
Remainder b(d,t)= 7.00000000 56 44 complete number a(d,t)= 43.0000000 modulo model= 9.00000000
Remainder c(d,t)= 7.00000000 56 44 remainder= 43.0000000 modulo model= 9.00000000
****
****
43 IS A PRIME IS A PRIME 43.0000000 43.0000000 9.00000000
Remainder b(d,t)= 16.0000000 56 44 complete number a(d,t)= 43.0000000 modulo model= 27.0000000
Remainder c(d,t)= 16.0000000 56 44 remainder= 43.0000000 modulo model= 27.0000000
****
****
43 IS A PRIME IS A PRIME 43.0000000 43.0000000 27.0000000
Remainder b(d,t)= 6.00000000 56 44 complete number a(d,t)= 43.0000000 modulo model= 37.0000000
Remainder c(d,t)= 6.00000000 56 44 remainder= 43.0000000 modulo model= 37.0000000
****
****
43 IS A PRIME IS A PRIME 43.0000000 43.0000000 37.0000000
Remainder b(d,t)= 0.00000000 56 44 complete number a(d,t)= 43.0000000 modulo model= 43.0000000
Remainder c(d,t)= 0.00000000 56 44 remainder= 43.0000000 modulo model= 43.0000000
****
****
43 IS A PRIME IS A PRIME 43.0000000 43.0000000 43.0000000

Remainder b(d,t)= 0.00000000 56 45 complete number a(d,t)= 44.0000000 modulo model= 1.00000000
Remainder c(d,t)= 0.00000000 56 45 remainder= 44.0000000 modulo model= 1.00000000
****
Remainder b(d,t)= 0.00000000 56 45 complete number a(d,t)= 44.0000000 modulo model= 2.00000000
Remainder c(d,t)= 0.00000000 56 45 remainder= 44.0000000 modulo model= 2.00000000
****
****

FOR NUMBERS A LITTLE BIGGER THAT GENERALIZE UP TO 100000 FOR NUMBERS A LITTLE BIGGER THAT GENERALIZE UP TO INFINITY, WITH COMMON COMPUTERS IT IS EASILY CALCULATED UP TO 100000 AND 1000000.

Remainder b(d,t)= 0.00000000 97 2026 complete number a(d,t)= 2025.00000 modulo model= 45.0000000
Remainder c(d,t)= 0.00000000 97 2026 remainder= 2025.00000 modulo model= 45.0000000
****
Remainder b(d,t)= 0.00000000 97 2027 complete number a(d,t)= 2026.00000 modulo model= 1.00000000
Remainder c(d,t)= 0.00000000 97 2027 remainder= 2026.00000 modulo model= 1.00000000
****
Remainder b(d,t)= 0.00000000 97 2027 complete number a(d,t)= 2026.00000 modulo model= 2.00000000
Remainder c(d,t)= 0.00000000 97 2027 remainder= 2026.00000 modulo model= 2.00000000
****
Remainder b(d,t)= 1.00000000 97 2027 complete number a(d,t)= 2026.00000 modulo model= 3.00000000
Remainder c(d,t)= 1.00000000 97 2027 remainder= 2026.00000 modulo model= 3.00000000
****
Remainder b(d,t)= 4.00000000 97 2027 complete number a(d,t)= 2026.00000 modulo model= 6.00000000
Remainder c(d,t)= 4.00000000 97 2027 remainder= 2026.00000 modulo model= 6.00000000
****
Remainder b(d,t)= 1.00000000 97 2027 complete number a(d,t)= 2026.00000 modulo model= 9.00000000
Remainder c(d,t)= 1.00000000 97 2027 remainder= 2026.00000 modulo model= 9.00000000
****
Remainder b(d,t)= 10.0000000 97 2027 complete number a(d,t)= 2026.00000 modulo model= 18.0000000

```
Remainder c(d,t)= 10.0000000 97 2027 remainder= 2026.00000 modulo model= 18.0000000
********************
Remainder b(d,t)= 1.00000000 97 2027 complete number a(d,t)= 2026.00000 modulo model= 27.0000000
Remainder c(d,t)= 1.00000000 97 2027 remainder= 2026.00000 modulo model= 27.0000000
********************
Remainder b(d,t)= 28.0000000 97 2027 complete number a(d,t)= 2026.00000 modulo model= 37.0000000
Remainder c(d,t)= 28.0000000 97 2027 remainder= 2026.00000 modulo model= 37.0000000
********************
Remainder b(d,t)= 0.00000000 97 2028 complete number a(d,t)= 2027.00000 modulo model= 1.00000000
Remainder c(d,t)= 0.00000000 97 2028 remainder= 2027.00000 modulo model= 1.00000000
********************
*****************************
      2027 IS PRIME IS PRIME 2027.00000 2027.00000 1.00000000
Remainder b(d,t)= 2.00000000 97 2028 complete number a(d,t)= 2027.00000 modulo model= 3.00000000
Remainder c(d,t)= 2.00000000 97 2028 remainder= 2027.00000 modulo model= 3.00000000
********************
*****************************
      2027 IS PRIME IS PRIME 2027.00000 2027.00000 3.00000000
Remainder b(d,t)= 2.00000000 97 2028 complete number a(d,t)= 2027.00000 modulo model= 9.00000000
Remainder c(d,t)= 2.00000000 97 2028 remainder= 2027.00000 modulo model= 9.00000000
********************
*****************************
      2027 IS PRIME IS PRIME 2027.00000 2027.00000 9.00000000
Remainder b(d,t)= 2.00000000 97 2028 complete number a(d,t)= 2027.00000 modulo model= 27.0000000
Remainder c(d,t)= 2.00000000 97 2028 remainder= 2027.00000 modulo model= 27.0000000
********************
*****************************
      2027 IS PRIME IS PRIME 2027.00000 2027.00000 27.0000000
Remainder b(d,t)= 29.0000000 97 2028 complete number a(d,t)= 2027.00000 modulo model= 37.0000000
```

output of result5.txt file

```
Remainder c(d,t)= 29.0000000 97 2028 remainder= 2027.00000 modulo model= 37.0000000
********************
*****************************
    2027 IS PRIME IS PRIME 2027.00000 2027.00000 37.0000000
Remainder b(d,t)= 0.00000000 97 2029 complete number a(d,t)= 2028.00000 modulo model= 1.00000000
Remainder c(d,t)= 0.00000000 97 2029 remainder= 2028.00000 modulo model= 1.00000000
********************
Remainder b(d,t)= 0.00000000 97 2029 complete number a(d,t)= 2028.00000 modulo model= 2.00000000
Remainder c(d,t)= 0.00000000 97 2029 remainder= 2028.00000 modulo model= 2.00000000
********************
Remainder b(d,t)= 0.00000000 97 2029 complete number a(d,t)= 2028.00000 modulo model= 3.00000000
Remainder c(d,t)= 0.00000000 97 2029 remainder= 2028.00000 modulo model= 3.00000000
********************
Remainder b(d,t)= 0.00000000 97 2029 complete number a(d,t)= 2028.00000 modulo model= 4.00000000
Remainder c(d,t)= 0.00000000 97 2029 remainder= 2028.00000 modulo model= 4.00000000
********************
Remainder b(d,t)= 0.00000000 97 2029 complete number a(d,t)= 2028.00000 modulo model= 6.00000000
Remainder c(d,t)= 0.00000000 97 2029 remainder= 2028.00000 modulo model= 6.00000000
********************
Remainder b(d,t)= 3.00000000 97 2029 complete number a(d,t)= 2028.00000 modulo model= 9.00000000
Remainder c(d,t)= 3.00000000 97 2029 remainder= 2028.00000 modulo model= 9.00000000
********************
Remainder b(d,t)= 0.00000000 97 2029 complete number a(d,t)= 2028.00000 modulo model= 12.0000000
Remainder c(d,t)= 0.00000000 97 2029 remainder= 2028.00000 modulo model= 12.0000000
********************

********************
Remainder b(d,t)= 32.0000000 102 1439 complete number a(d,t)= 1438.00000 modulo model= 37.0000000
Remainder c(d,t)= 32.0000000 102 1439 remainder= 1438.00000 modulo model= 37.0000000
```

****
Remainder b(d,t)= 0.00000000 102 1440 complete number a(d,t)= 1439.00000 modulo model= 1.00000000
Remainder c(d,t)= 0.00000000 102 1440 remainder= 1439.00000 modulo model= 1.00000000
****
****
1439 IS PRIME IS PRIME 1439.00000 1439.00000 1.00000000
Remainder b(d,t)= 2.00000000 102 1440 complete number a(d,t)= 1439.00000 modulo model= 3.00000000
Remainder c(d,t)= 2.00000000 102 1440 remainder= 1439.00000 modulo model= 3.00000000
****
****
1439 IS PRIME IS PRIME 1439.00000 1439.00000 3.00000000
Remainder b(d,t)= 8.00000000 102 1440 complete number a(d,t)= 1439.00000 modulo model= 9.00000000
Remainder c(d,t)= 8.00000000 102 1440 remainder= 1439.00000 modulo model= 9.00000000
****
****
1439 IS A PRIME IS A PRIME 1439.00000 1439.00000 9.00000000
Remainder b(d,t)= 8.00000000 102 1440 complete number a(d,t)= 1439.00000 modulo model= 27.0000000
Remainder c(d,t)= 8.00000000 102 1440 remainder= 1439.00000 modulo model= 27.0000000
****
****
1439 IS A PRIME IS A PRIME 1439.00000 1439.00000 27.0000000
Remainder b(d,t)= 33.0000000 102 1440 complete number a(d,t)= 1439.00000 modulo model= 37.0000000
Remainder c(d,t)= 33.0000000 102 1440 remainder= 1439.00000 modulo model= 37.0000000

FOR NUMBERS A LITTLE BIGGER THAT GENERALIZE TO INFINITY, WITH COMMON COMPUTERS IT IS EASILY CALCULATED UP TO 100000 AND 1000000.

****
****
1439 IS A PRIME IS A PRIME 1439.00000 1439.00000 37.0000000
Remainder b(d,t)= 0.00000000 102 1441 complete number a(d,t)= 1440.00000 modulo model= 1.00000000

Remainder c(d,t)= 0.00000000 102 1441 remainder= 1440.00000 modulo model= 1.00000000
****
Remainder b(d,t)= 0.00000000 102 1441 complete number a(d,t)= 1440.00000 modulo model= 2.00000000
Remainder c(d,t)= 0.00000000 102 1441 remainder= 1440.00000 modulo model= 2.00000000
****
Remainder b(d,t)= 0.00000000 102 1441 complete number a(d,t)= 1440.00000 modulo model= 3.00000000
Remainder c(d,t)= 0.00000000 102 1441 remainder= 1440.00000 modulo model= 3.00000000
****
Remainder b(d,t)= 0.00000000 102 1441 complete number a(d,t)= 1440.00000 modulo model= 4.00000000
Remainder c(d,t)= 0.00000000 102 1441 remainder= 1440.00000 modulo model= 4.00000000
****
Remainder b(d,t)= 0.00000000 102 1441 complete number a(d,t)= 1440.00000 modulo model= 5.00000000
Remainder c(d,t)= 0.00000000 102 1441 remainder= 1440.00000 modulo model= 5.00000000
****
Remainder b(d,t)= 0.00000000 102 1441 complete number a(d,t)= 1440.00000 modulo model= 6.00000000
Remainder c(d,t)= 0.00000000 102 1441 remainder= 1440.00000 modulo model= 6.00000000
****
Remainder b(d,t)= 0.00000000 102 1441 complete number a(d,t)= 1440.00000 modulo model= 8.00000000
Remainder c(d,t)= 0.00000000 102 1441 remainder= 1440.00000 modulo model= 8.00000000
****
Remainder b(d,t)= 0.00000000 102 1441 complete number a(d,t)= 1440.00000 modulo model= 9.00000000
Remainder c(d,t)= 0.00000000 102 1441 remainder= 1440.00000 modulo model= 9.00000000
****

Conclusion:

Therefore, the resolution of calculations of congruences of very large numbers is defined here, as $5^{60} \equiv 26q + r$, which takes us to the organization of the problems addressed in a more comprehensive way and more emphasized with the final resolution of the problem, which will be concluded from our emphasis below:

3.2.5 Example

Let's determine the remainder of the division of5^{60} by 26.

Writing ,$5^{60} \equiv 26q + r$ the problem is equivalent to determining the integer r such that 0≤r ≤ 25 and such that $5^{60} \equiv r \ (mod\ 26)$.

We noticed that 5^2=25, that is,5^2 ≡-1 (mod 26). Using part (vii) of proposition 1.3.3, we have that 5^4≡(-1)^2(mod 26), that is, 5^4≡1 (mod 26).

Finally,$5^{60} = (5)^{15}$**, therefore,**$5^{60} \equiv (-1)^{15}\ (mod\ 26)$**, whence the remainder of the division of** 5^{60}**by 26 is 1.**

In this example 3.2.5 and the essence of the problem of the program of the articles and of the other two articles cited in this article for the final resolution article of the Problem of Decoding the Private Key of RSA Encryption as that of5^{60} **by 26.**

It is written $5^{60} \equiv 26q + r$

which is equivalent to determining the integer r such that 0 ≤ r ≤ 25

and such that

$5^{60} \equiv r\ (mod\ 26)$.

From reference [2]:

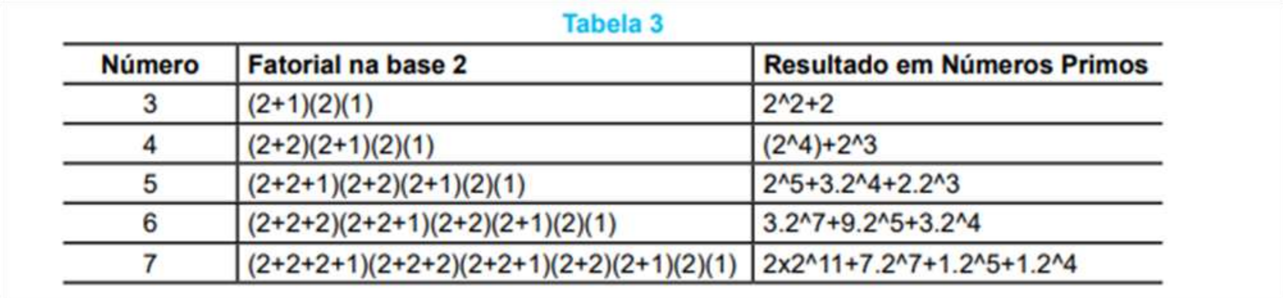

Tabela 3

Número	Fatorial na base 2	Resultado em Números Primos
3	(2+1)(2)(1)	2^2+2
4	(2+2)(2+1)(2)(1)	(2^4)+2^3
5	(2+2+1)(2+2)(2+1)(2)(1)	2^5+3.2^4+2.2^3
6	(2+2+2)(2+2+1)(2+2)(2+1)(2)(1)	3.2^7+9.2^5+3.2^4
7	(2+2+2+1)(2+2+2)(2+2+1)(2+2)(2+1)(2)(1)	2x2^11+7.2^7+1.2^5+1.2^4

“To solve the problem we referenced the data in Table 3, it was necessary to write the factorial numbers n! = n.(n-1)(n-2)...1 in the base of the sum 2 and 1, for example 6, it is written, 6.5.4.3.2.1 = (2+2+2)(2+2+1)(2+2)(2+1)(2)1, doing this for several factorial numbers and studying the sequence it was possible to deduce that six can be written as, 6! = 3.2^7+9.2^5+3.2^4, where 6 is the fourth factorial number starting from 3, which corresponds exactly to the fourth prime number in the organization of the first term. By doing the same for several factorial numbers, the pattern is maintained and the above Fortran code makes it possible to find the first term in a prime number and, when necessary, to find the corresponding term by multiplication. Consequently, it is simple to find the factorial number written in a base 2 sum with prime exponents. An adaptation of the code results in finding only the nth term of the factorial and the first term in the corresponding prime number. So, it is enough to rearrange the terms in base 2 in other prime numbers and, when necessary, multiply the base raised to a prime exponent. Interestingly, this multiplicative factor is usually a prime number. Considering that we consider base 2 and a prime exponent, it is very possible to write these factors in even numbers such as 4 and 8, or 6, and so on. However, 6 is equal to 2x3, which requires another code. However, it is possible to use the number 3 or any other number in a similar way. To arrive at the deduction of the necessary code and possible adaptations, it is enough to start from the organization of the terms of the factorial numbers, for example, 6! = 6.5.4.3.2.1 = (3+3)(3+2)(3+1).2, thus we generate other parameters for the problems and generate a new solution, with a different basis”[2].

Bibliographic References:

[1]Pre-Analysis of Breaking the Decoding of RSA Encryption, for Small Numbers, in Order to Generalize to Larger Numbers, Up to Very Large Numbers Welken Charlois Gonçalves

doi 10.48021/978-65-270-2479-8-C4

[2]Welken Charlois Gonçalves. A polynomial solution for factorial problems based on the complexity of solving the traveling salesman problem, combinatorial analysis. Mathematics and its applications: resources and strategies for effective teaching - AYA Editora. DOI: 10.47573/aya.5379.2.196.1

[3]Numbers: An Introduction to Mathematics Portuguese Edition by Cesar Polcino Milies and Sonia Pitta Coelho. Available at:https://www.ime.usp.br/~iusenko/ensino_2024_1/MAT0120/books/polcino.pdf

[4]The music of prime numbers Marcus Du Sautoy Publisher: Zahar Year: 2007

[5]Prime Numbers - Old mysteries, new records Author(s):Paulo Ribenboim Pages: 317 Publication: IMPA, 2020