iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 3 **Issue:** VI **Month of publication:** June 2015

DOI:

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)

Prime Numbers and Modular Exponentiation in Encryption and Decryption

Babita Bist Ramola
Asstt. Professor, Deptt. of Mathematics, S. D. College(Lahore), Ambala Cantt.

Abstract: ***The Number theory is one of purest branches of mathematics, but it is also becoming a powerful tool when it comes to computer security. When we shop online, the Number theory helps to protect sensitive data such as credit card numbers which is the result of mathematics research and is now being applied worldwide. In this paper, we have discussed the some concept of number theory like prime numbers, integer factorization and Modular Exponentiation and Roots which provide way to worldwide used RAS cryptosystem.***
Keywords: Number theory, Rivest-Shamir-Adleman (RSA), Network Security, Cryptography, Encryption, Decryption

I. INTRODUCTION

Unlike the original use of cryptography in the past where it was implemented to secure both diplomatic and military secrets from the enemy, the cryptography of today has expanded its domain, and has been designed to secure and thus to protect large amounts of electronic data that is communicated and stored across the worldwide network. Sensitive data exchanged between Website and a user needs to be encrypted (coded) to prevent it from being modified by or disclosed to other parties. The encryption should be done so that the decryption (decoded) is only possible with complete knowledge of a secret decryption key. The decryption key should only be known by authorized parties. In traditional cryptography, the encryption and decryption operations are performed with the same key. This means that the party encrypting the data and the party decrypting it have to share the same decryption key but Today's cryptography is vastly more complex than its past. What if they don't already share a secret key, how do they establish the first one? Our computer doesn't initially share any secret keys with Web sites. How then does one encrypt data that are sending to the site? One might eventually set up a password, and the password could then be used to derive an encryption key [1]. In 1976, Whitfield Diffie and Martin Hellman suggested that the encryption and decryption is also possible with a pair of different keys rather than with the same key. But the decryption key should be kept secret and the encryption key could be made public without compromising the security of the decryption key. And this concept of making the encryption key public is called public-key cryptography. Diffie and Hellman answered the setup problems for protecting online data. To enable computers to encrypt data for a site, the site simply needs to publish its encryption key, for instance in a directory. Every computer can use that encryption key to protect data sent to the site. But only the site has the corresponding decryption key, so only it can decrypt the data [2].

How mathematics works in Cryptography

The concept of mathematics which provides the way to today's RAS public-key cryptosystem is prime generation. Prime numbers of any size are very common, and it's easy to test whether a number is a prime – even a large prime. To generate a random prime, we can simply generate random numbers of a given size and test them until a prime is found. It hasn't always been easy to test whether a number is a prime. In fact, it might seem that testing for primality would require one to determine all the factors of the number to see if there are others beside the number itself and 1. Faster methods for primality testing were discovered in the 1970s that test for certain properties held by prime numbers but not by composites, rather than finding the factors. Without these results, much of public-key cryptography today would not be practical because of the dependence on efficient methods of generating primes [3].

Diffie and Hellman demonstrated only that public-key encryption was possible in theory. But three MIT mathematicians-Ronald L. Rivest, Adi Shamir, and Leonard M. Adleman figured out a way to do it in the real world. In RSA, encryption scheme is the mathematical task of factoring. Factoring a number means identifying the prime numbers which, when multiplied together, produce that number. Thus 397,410 can be factored into 2 x 3 x 5 x 13 x 1,019, where 2, 3, 5, 13, and 1,019 are all prime. (A given number has only one set of prime factors) [4]. Despite the efforts of great mathematicians Fermat, Gauss, and Fibonacci, nobody has ever discovered a usable and consistent method for factoring large numbers. Mathematicians try potential factors by invoking complex rules of thumb, looking for numbers that divide evenly. For huge numbers the process is horribly time-consuming, even with fast computers. The largest number yet factored is 155 digits long. It took 292 computers, most of them fast workstations, more than seven months [5]. It is easy to multiply primes together. But there is no easy way to take the

©IJRASET 2015: All Rights are Reserved

product and reduce it back to its original primes. For example, it is very simple to multiply together 57191 and 11777, but the reverse problem is much harder. Suppose a number 1459160519 is given and even after telling that the number is multiplication of two integers it is very difficult to find those integers. A computer can factorize that number quickly, but it basically does it by trying many of the possible combinations [6].

But what if the number to be factored is not of ten digits, but rather 400 or more digits? It is, however, not too hard to check if a number is prime. In other words, to check that it cannot be factored. If it is not prime, it is difficult to factor, but if it is prime, it is not hard to show it is prime. So RSA encryption works on this concept [6]. Let us consider two prime numbers, p and q that have 200 or maybe 300 digits each. We will keep these numbers secret (these numbers are private key), and multiply them to make a number N = pq. The number N is basically public key. It is relatively easy to obtain value of N just by multiplying the two numbers. But if we know N, it seems to be basically impossible to obtain the values of p and q. Here is complete example of exactly how is N used to encode a message, and how is p and q used to decode it? But we are using small prime numbers so that it would be easy to follow the arithmetic (Keep in mind that in a real RSA encryption system the prime numbers are huge). In the following example, suppose that X wants to make a public key, and Y wants to use that key to send a message to A. We will suppose that the message X sends to Y is just a number. We assume that X and Y have agreed on a method to encode text as numbers. Steps are following:

Step 1: Let X selects two prime numbers - p = 23 and q = 41 (keep in mind that the real numbers should be much larger)

Step 2: X multiplies p and q together to get pq = (23) (41) = 943. 943 is the “public key” which he tells to Y (may be to anyone, if he wishes).

Step 3: X also chooses another number e which should be relatively prime to (p-1) (q-1). In this case, (p-1) (q-1) = (22)(40) = 880, so we could choose the number e = 7 which is co-prime to 880. The number e is also part of the public key, so the value of e is also told to Y.

Step 4: Now Y knows sufficient to encode a message to X. Suppose that the message is the number M = 35.

Step 5: Y calculates the value of $C = M^e \pmod N = 35^7 \pmod{943}$.

Step 6: $35^7 = 64339296875$ and $64339296875 \pmod{943} = 545$. The number 545 is the encoding that Y sends to X.

Step 7: Now it’s X’s turn to decode 545. For this, he has to find a number d such that $ed = 1 \pmod{(p-1)(q-1)}$, i.e. $7d = 1 \pmod{880}$ which gives d = 503, since 7 x 503 = 3521 = 4(880) + 1 = 1(mod 880).

Step 8: To get the decoding, A have to calculate $C^d \pmod N = 545^{503} \pmod{943}$. Which seems to be a very horrible calculation, but notice that 503 = 256+128+64+32+16+4+2+1 (this is just the binary expansion of 503). So this means that

$545^{503} = 545^{256+128+64+32+16+4+2+1} = 545^{256}\ 545^{128}$ …….. 545^1

The line above just uses basic rules about how exponents work. Now since we only care about the result (mod 943), we can calculate all the parts of the product(mod 943). By repeated squaring of 545, we can get all the exponents that are powers of 2. For example, $545^2 \pmod{943} = 545 \cdot 545 = 297025 \pmod{943} = 923$. Then square again: $545^4 \pmod{943} = (545^2)^2 \pmod{943} = 923 \cdot 923 = 851929 \pmod{943} = 400$, and so on. We obtain the following table:

$545^1 \pmod{943}$	=	545
$545^2 \pmod{943}$	=	923
$545^4 \pmod{943}$	=	400
$545^8 \pmod{943}$	=	633
$545^{16} \pmod{943}$	=	857
$545^{32} \pmod{943}$	=	795
$545^{64} \pmod{943}$	=	215
$5451^{28} \pmod{943}$	=	18
$545^{256} \pmod{943}$	=	324

So the result we want is:

$545^{503} \pmod{943} = 324 \cdot 18 \cdot 215 \cdot 795 \cdot 857 \cdot 400 \cdot 923 \cdot 545 \pmod{943} = 35$.

So X can decode Y's message and obtain the original message N = 35 using above tedious calculation [7].

II. CONCLUSION

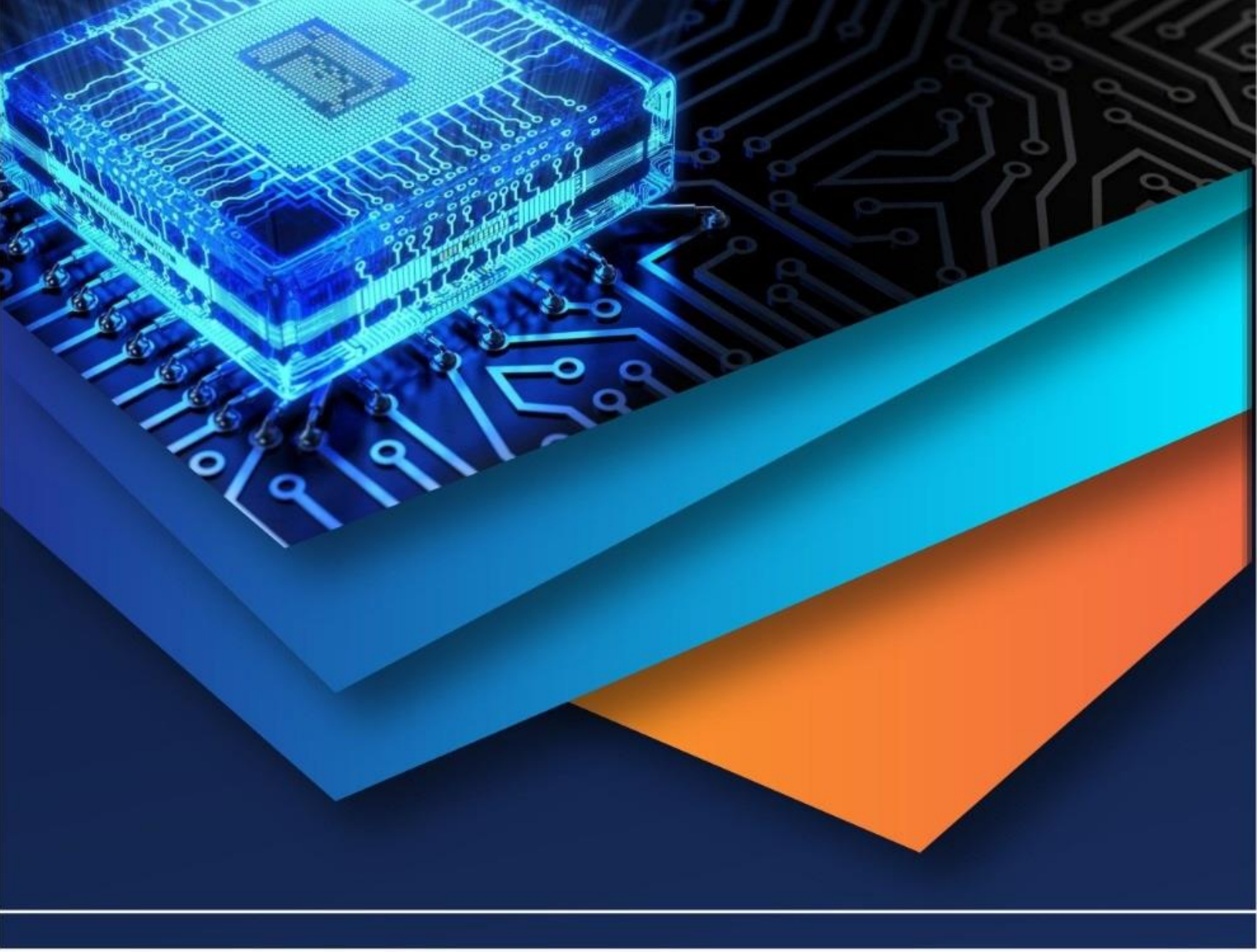
The main areas of mathematics that have been used in cryptography have been discussed in the paper. The paper presents an overview of importance of mathematical concepts- Prime generation and modular exponentiation. Many older encryption algorithms and public key systems are based on Modular arithmetic and prime generation.

©IJRASET 2015: All Rights are Reserved

REFERENCES

[1] D. L. Calloway, Literature review on Cryptography and Network Security, Capella University, OM 8302, 2008.
[2] W. Diffie and M.E. Hellman, New directions in cryptography, IEEE transactions on Information theory, vol. 22, issue. 6, pp: 644-654, 1976.
[3] B. Kaliski, The Mathematics of the RSA Public - Key Cryptosystem, RSA Laboratories, 2006.
[4] R. Rivest, A. Shamir and L. Adleman, A Method for Obtaining Digital Signatures and Public-Key Cryptosystems, Communications of the ACM 21 (2) pp: 120–126, 1978.
[5] P. E. Agre, Internet security and privacy, 2010 Downloaded from http://www.scribd.com/doc/21407432/Internet-Security-and-Privacy-Inside-p-94#scribd on date 15/04/2015.
[6] C. Mann, Published in The Atlantic magazine, 2002.
[7] http://www.macalester.edu/~hutchinson/book/chapter4.pdf, Downloaded on date 02/02/2015.

©IJRASET 2015: All Rights are Reserved



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)