USE OF PRIME NUMBERS IN RSA ALGORITHM

SHILPA BHIMRAJ HUSALE

N. G. Acharya & D. K. Marathe college, Mumbai

ASSISTANT PROFESSOR

Abstract

E-commerce has brought a revolution in commercial world in last few decades .The transactions in E-commerce are based on public key and private key system. We need to exchange bank account numbers and other personal details in e-transactions. People would have accepted this concept unless there is very safe mechanism to protect these personal details. For maintaining secrecy we need code language which is easy to encode but very difficult to decode so that people find it easy to encode but the code breakers cannot find the information. RSA code has proved to be ideal in this regard. It is based on the basic and simple mathematical property that" It is very easy to multiply two big prime numbers but if the product is given, it is very difficult to factorize it into prime factors."

This paper intends to discuss the coding and decoding procedures used in RSA codes and also to discuss advantages and shortcomings of this scheme. Historical background of the RSA code is also described in the paper.

Keywords: *E-commerce, prime numbers, random nature of primes, big primes, RSA code ,public key, private key.*

What is a Prime number?

Let's begin with the definition of prime number p. The number p € N is said to be a prime number, if it has only two divisors 1 and p itself. 2 is the smallest prime number and every prime greater than 2 is an odd number.

Big prime numbers are the prime numbers having large number of digits.

Construction of big prime number :-Take a very big number(n) of pre-decided length and check its primality test. We can use trial division algorithm,if we found any number a between 1 and sqrt(n), which divides n then the number n is not a prime number. Even mersenne numbers helps to find big primes. Mersenne prime numbers are of the form $Mn=2^n-1$. But every Mersenne number is not prime here also we can apply different primality tests like Lucas-Lehmer primality test which says, for any p>2,$Mp=2^p-1$ is a prime if and only if Mp divides S_{p-2} where $S_0=4$ and $S_k=(S_{k-1})^2-2$ for k>0.

For example $M_3=2^3-1=7$ is a prime iff $7/S_1$ where $S_{1=}14$.

RSA algorithm

Introduction

RSA algorithm is based on simple mathematical property that "It is very easy to multiply two big prime numbers but if the product is given, it is very difficult to factorize it into prime factors." If suppose A wants to send a message to B then A encrypts his message and sends it to B then, B decrypts the message and reads it. To encrypt and decrypt the message Public and Private keys are formed. Encryption is nothing but coding a message into a secret code and decryption is decoding of the message. For that receiver should publish public key and sender has to use this public keys to encode the message and sends it to receiver. The main goal of public key encryption is to ensure that the communication is being sent and is kept confidential during transaction. Another application in public key encryption is the digital signature . Digital signature scheme can be used for sender's authentication and nonrepudiation. The sender computes the digital signature for the

message to be sent, then sends the digital signature with the message to the intended receiver. Digital signatures scheme have property that, it can be calculated onlywith the knowledge of correct private keys. RSA algorithm helps both encrypt and create digital signature.

History of RSA algorithm:

Around 1970 scientist Whitfield Diffie, Martin Hellman ,Clifford Cocks were working on one way function that even after knowing public key it is difficult to encode private key.Two computer scientists Ron Rivest, Adi Shamir and one Mathematician Leonard Adleman from MIT comes with this practical difficulty of factorising a very big number into large prime numbers. They made several attempts and tried many approaches but could not succeed. In april 1977 Ron Rivest spend the whole night in putting forward his thoughts, named as RSA coding made up of initial letters of names of these scientists.

Construction of Public key and Private key

RSA algorithm involves four steps key generation, key distribution, encryption and decryption. The basic step is key generations.

Suppose A wants to send a message (x) to B then first B has to disclose his public keys. Generation of public key has following steps

1) Choose two large prime numbers p and q.
2) n=p*q
3) Choose e≠1 so that e is relatively prime to (p-1)*(q-1)
4) Compute $d=e^{-1} \bmod (p-1)*(q-1)$
5) B will publish e and n which are his public keys.
6) And keeps d as a secret key(private key)

Now A will use these public keys e and n, to encrypt the message through following steps
1) A computes $y=x^e \bmod n$
2) Send y to B
3) B then computes $z=y^d \bmod n$ using his private key d
4) Read z

Now if $y^d \bmod n=x \bmod n$ then only we can say that B could read the message, and to prove this we can use the theorems in Number theory.

1) **Chinese Remainder theorem:** If p and q are relatively prime integers and a € Zp and b € Zq then the equations

s mod p=0
s mod q=0
have **one and only one solution** for integers s between 0 and pq-1
consider two equations
$(y^d - x) \bmod p=0$ and $(y^d - x) \bmod q=0$----------(*)
The statement $(y^d - x) \bmod p=0$ is equivalent to saying that $(y^d - x) = i*p$ for some integer i similarly $(y^d - x) = j*q$ for some integer j. If some number is multiple of the prime p and q them it is multiple of pq. Thus $(y^d - x) \bmod pq=0$ but x and $y^d \bmod pq$ are both integers between –(pq-1) and (pq-1). The only integer between these two values is 0 mod pq is 0 itself and
hence $(y^d - x) \bmod pq=0$
so $x \bmod pq=y^d \bmod pq$ So B receives perfect message.
We said equation (*) without any proof. These equation is true ,which applies Fermat's Little Theorem.

2) Fermat's Little Theorem: For every positive integer a and prime p, if a is not multiple of p then $\mathbf{a^{p-1} \bmod p=1}$

$y^d \bmod p= (x^e)^d \bmod p$
where e and d relatively prime to m, m=(p-1)*(q-1)
$ed \bmod (p-1)*(q-1)=1$
$ed=k*(p-1)*(q-1)+1$
$x^{ed} \bmod p =(x^{k(q-1)})^{p-1}.x \bmod p$
hence $y^d \bmod p= x \bmod p$

Examples:

Let's work out two example

- P=17 q=11

n=p*q=187
m=(p-1)*(q-1)=16*10=160
choose e=7 such that gcd(e,m)=1 now take $d=e^{-1}$
de=1 mod 160 and d<160 so d is 23
here Public keys are e =7 and n= 187
Private key=d=23

- p=127 q=131

n=p*q=16637
m=(p-1)(q-1)=126*130=16380
choose e=11, gcd(e,16380)=1
determine d such that de=1 mod 16380 and d<16380
so d= 1489
Public keys are e =11 and n= 16637
Private key=d=1489

In above two examples we describe how to generate public key and private key. where n is public key which is published by the receiver if n is such a small number, like above two examples then it is very easy to factorised it into two primes which can reveal the private key (d) by using the congruence relation ed≡1 mod (p-1)(q-1) which makes RSA coding insecure. Hence to generate Public key and Private key one should put the efforts to select very large primes(p,q)such that their product(n) again is extremely large value. Efforts made by several researchers concluded that it takes long period of time and no efficient algorithm exists to compute big prime factors of a very huge number. For example (RSA-768) a 232 digit number utilizes hundreds of machines and took more than two years to factorised into two primes.

Conclusion: In this paper we discussed prime numbers, big primes , construction of big primes and how these big primes can be efficiently used in RSA algorithm.

References:

- en.m.wikipedia.org
- math.bu.edu