Primes and Privacy!

Ritwik Prabin Kalita

Student, Mathematics and Computing (M.Sc.), Indian Institute of Technology Guwahati

E-mail: `ritwikkalita88@gmail.com`

"What is the application of Mathematics in real life?" In our life, at least once, we have been asked this question, or we have asked it to ourselves. In this short article we show one application of mathematics in real life.

Suppose you live in a different city than your parents and you want to send your newly acquired debit card number and its pin to your parents, how can you be so sure that the number doesn't get leaked on its way to your parents inbox? Well, the prime numbers are behind the process by which a text we send to someone through the internet is secure.

Cryptography is the method that is behind the protection of our information so that only those for whom the information is intended to can read and access it. One of the algorithms that cryptography works on, is the RSA algorithm. The three letters in the name represent the names of three cryptographers who invented it. In 1977, *Ronald Rivest, Adi Shamir and Leonard Adleman*, these three cryptographers came up with a public-key cryptography algorithm. Encryption is the process of converting your information to some specific code to prevent unwanted access. The RSA algorithm consists of one encryption key (N, e) and one decryption key (N, d). What are these e, N and d? They are nothing but some very large real numbers. We will explain the process briefly in this article.

Say you want to send a text T to your friend. Your friend (well, his device actually) will choose two prime numbers p and q. Now these two primes are known to your friend only. Why? Because he doesn't want your message to him get leaked on the way to him. He will multiply the primes to get N, where $N = p \times q$ and another number e, where e is less than and co-prime number to $(p-1) \times (q-1)$. These N and e together form a key, called, the public key (N, e). As its name says, it then gets published on the server, that is, anyone can access (N, e).

Now it's your turn. You will access (N, e) (remember that it is a public key, that is why you can access it). You will then first convert your text to number format, say, a number M, and will find $M^e \pmod N$. Let $K = M^e \pmod N$. Now your original text has been encrypted to a completely new number. Hence, you will send your encrypted message to your friend.

The last step is again done by your friend. Now it?s his turn to decrypt the encrypted message sent by you and thus access the original message that he was supposed to get. What he will do is find a number d, such that $e \times d = 1 \pmod{(p-1)(q-1)}$ and then calculate $K^d \pmod N$ to get the original text.

Let us work with proper numbers and see this process in action. Let us suppose, you want to send a text "Hi" to your friend.

Step 1: Making public and private key

(i) First, your friend will choose two primes, in this case, say, $p = 11, q = 13$.

(ii) $N = p * q = 11 * 13 = 143$.

(iii) e is any number less than and co-prime number to $(p-1) \times (q-1) = (11-1) \times (13-1) = 120$. Let your friend choose e as 7 which is co-prime to 120.

(iv) Then he will make these two numbers $(N, e) = (143, 7)$ public.

Step 2: Encrypting your text

(i) You will get the public key $(143, 7)$, as it is public and everyone has access to it.

(ii) You will then convert your text 'Hi' to a number. Say, in this case, we assume,

$$a = 1, b = 2, \ldots, z = 26.$$

Then $H = 8$, and $i = 9$. Then 'Hi' will be simply 89 (value of M in our case).

(iii) $M^e \pmod N = 89^7 \pmod{143} = 67$.

(iv) Your system will then send 67 to your friend.

Step 3: Decrypting your text

(i) Your friend will find out the value of d, such that $e \times d = 1 \pmod{(p-1)(q-1)}$, that is, $7d = 1 \pmod{120}$. A simple algorithm can be made to solve d, which in our case will be 103, as $7 \times 103 \pmod{120} = 721 \pmod{120} = 1$.

(ii) Taking $d = 103$, he calculates $K^d \pmod N = 67^{103} \pmod{143} = 89$.

(iii) Convert number to text using the same technique, assuming $a = 1, b = 2$ and so on. For 89, your friend then gets the original message "Hi".

Algorithm for finding d

We have, $e \times d = 1 \pmod{(p-1)(q-1)}$, that is, $(e \times d - 1) \pmod{(p-1)(q-1)} = 0$, where e, p and q are known.

```
for (i = 2 ; i > 0 ; i++)
{
if ((e*i - 1) mod (p-1) * (q-1) ==0)
{
d = i;
break;
}
}
```

Role of Primes

Now comes the question, what the actual role of primes in this algorithm is. Suppose, your friend publishes the public key (N, e). Now not only you, but everyone has access to it. When you encrypt your message and send it to your friend, say, a stranger wants to access your secret message. So basically, in the above-mentioned example, he wants to convert the encrypted message 67 to the original message 'Hi'. Can he do it? Remember, he has also access to the public key $(143, 7)$ and he knows how the RSA algorithm works. So, all he needs to do is to find the value of d.

He has $N = 143, e = 7$, encrypted code 67. But to find d, he must know the value of p and q. Because the only formula to get the value of d is $e \times d = 1 \pmod{(p-1)(q-1)}$. But p and q are nothing but two prime factors of 143. And the stranger knows the number 143. He will therefore try to factorize 143. If he succeeds in factorizing it to 11×13, he gets the value of d, consequently the secret text. The question is, then how is this algorithm secure. Well, for 143, it is not very difficult. But, what if, instead of 11 and 13 as primes, your friend chooses $p = 7907$ and $q = 7919$, hence $N = 7907 \times 7919 = 62615533$. Will the stranger be able to find the prime factors so easily as earlier? It is easier to multiply two large primes to get a larger N than to factorize a large N. And if someone can't find p and q, then they can't find the value of d, and so can't access the secret text. As p and q are known only to your friend, only he will be able to decrypt your secret text using the value of d in the specified formula. So, the key point of this algorithm is to choose two prime numbers as large as possible to get another larger number so that the process can't be reversed, that is, no one can break it into the original two prime numbers.

In the modern century, technology is developed so fast. There are powerful super computers which can factorize a large number to its prime factors. But everything has its limit. Even in the current decade, only 1024-bit primes can be prime factorized. So, now-a-days, many of the banks or websites like Twitter, etc. are using 2048-bit number which can't be factorized at the current time. But soon, as technology develops, someone will find some efficient algorithm to find prime factors of these large numbers too. The goal, therefore, for us is to find larger primes as they play a huge role in the process.

Finding large primes

The density of prime numbers decreases as we move to the right in the real line. So, it becomes difficult to find new prime numbers. There are several methods to find new primes or check if a number is prime or not. Mersenne Prime are one class of prime numbers which yields such a method. These primes are of the form $2^P - 1$, where P is also a prime. To check if for some prime P and $2^P - 1$ are primes or not, a test called Lucas-Lehmer Primality Test is used. It states that: *For any prime $p > 2, 2^P - 1$ is prime if and only if it divides a_{p-2}, where $a_o = 4$ and a_n is recursively defined by $a_n = (a_{n-1})^2 - 2$, for $n \geq 1$.*

If $2^P - 1$ is prime, then it is obviously much larger then the prime P. In the 17th century, when French mathematician Marin Mersenne discovered these primes, it might seem like they were of no use. But the whole privacy system in this modern world depends on these primes! Numbers are interesting indeed!

Ronald Rivest, Adi Shamir and Leonard Adleman