Prime Numbers and Encryption

AMICS

October 1, 2024

Contents

1 Prime Numbers

1.1 Definition and basic theorems

Prime numbers have been closely studied by mathematicians ever since they were recognised. They are central to number theory due to their intriguing properties and their elusive unpredictability. First, let's ask ourselves: What are primes?

Definition 1 *An integer $p > 1$ is called a* ***prime*** *if its only divisors are 1 and p itself.*

By definition, the first few primes are 2, 3, 5, 7, 11, 13. They have no divisors other than 1 and themselves. Numbers that aren't prime are called **composite numbers**, and are based on primes. Notice how there are infinitely many integers, and how composite numbers are made up of primes. Then we have the two most fundamental properties of primes:

1. **Unique factorization** (Fundamental theorem of arithmetic):

   Any interger $n > 1$ has a unique representation as a product of primes.

2. **Infinitude** (Euclid's theorem):

   There are infinitely many primes.

For example, the number 75 can be written as $3 \times 5 \times 5$, or 3×5^2, where 3 and 5 are primes. The numbers 3 and 5 here are called the **prime factors** of 75. Notice that you can do this for all numbers, generating any combination of products of prime powers. We then have the **canonical factorization** of any integer n:

$$n = p_1^{\alpha_1} \cdots p_k^{\alpha_k}$$

2 Primality check with trial division

2.1 Trial division

The property of being prime is called **primality**. Suppose you are given a random integer n. How would you determine if it is prime? Well first, except for 2, every even number is automatically composite, as it is divisible by 2. Any number that ends with 5 other than 5 itself also automatically fails the primality test for the same reason. From this, we have established that primes can only end with digits 1, 3, 7, and 9. Even then, suppose you are given a random integer ending with one of those four digits, how would you determine if it is prime?

You might be tempted to check every integer up to n and see if it divides n, but that is a bit superfluous. You only need to check integers up to $\sqrt{n}$, as the product of 2 factors larger than that will produce a number bigger than n. This is a simple check, but it is impractical for large numbers. Luckily, we developed much better tests over time, some of them being important theorems in number theory.

3 Modular arithmetic and important theorems

3.1 Modular arithmetic

Modular arithmetic is important in expressing number theory ideas. For example, saying 18 divided by 4 gives a remainder of 2 is the same as saying 18 is congruent to 2 modulo 4, or in modular notation, $18 \equiv 2 \pmod 4$. It's like an analogue watch: you look at 2pm and you know that it's the 14th hour in the day, because $14 \equiv 2 \pmod{12}$. We consider modulo 12 for telling time, and other modulo for different purposes. More formally, we have:

Definition 2 *Given an integer m and integers a, b. We say a is congruent to b modulo m if $m|a-b$, in which case we write $a \equiv b \pmod m$. Conversely, we say a is not congruent to b modulo m and write $a \not\equiv b \pmod m$.*

From the definition, we see that $a \equiv b \pmod m$ if and only if there exists some integer k such that $a = b + km$. We also have some elementary properties:

1. (Reflexivity) $a \equiv a \pmod m$.
2. (Commutativity) If $a \equiv b \pmod m$ then $b \equiv a \pmod m$.
3. (Transitivity) If $a \equiv b \pmod m$ and $b \equiv c \pmod m$ then $a \equiv c \pmod m$.

Also, for any integer m and integers $a_1, a_2, \ldots, a_m; b_1, b_2, \ldots, b_m$ such that $a_i \equiv b_i$ for all $i = 1, 2, \ldots, m$, then the following holds:

1. $\displaystyle\sum_{i=1}^{m} a_i \equiv \sum_{i=1}^{m} b_i \pmod m$
2. $\displaystyle\prod_{i=1}^{m} a_i \equiv \prod_{i=1}^{m} b_i \pmod m$

Also, if $a \equiv b \pmod m$ then $a^n \equiv b^n \pmod n$.
We also have an important definition: Given a positive integer m and an integer a. An integer b is called the **modular inverse multiplicative** of a modulo n if $ab \equiv 1 \pmod n$

3.2 Fermat's primality test

First, we must know Fermat's little theorem, an important tool in finding primes.

Fermat's little theorem: If p is a prime, then for any integer a, we have $a^p - a \equiv 0 \pmod p$

However, the inverse is not true. There are composite numbers that satisfies $a^{n-1} \equiv 1 \pmod p$. These numbers are called **Carmichael numbers**, and are the sole reason why this test is not deterministic. Before going into further details about the accuracy of this test, we will first show you how it works:

Input: $n > 3$, a value to test for primality; k, a parameter that determines the number of times to test for primality
Output: composite if n is composite, other wise **probably prime**.
Repeat k times:

1. Pick a randomly in $[2, n-2]$
2. If $a^{n-1} \not\equiv 1$, return **composite**.

Otherwise, return **probably prime.**

We know that it holds trivially for $a \equiv 1 \pmod{p}$ and $a \equiv -1 \pmod{p}$, so naturally, we would pick all $1 < a < p-1$ for the least computation needed. If we have $a^{n-1} \equiv 1 \pmod{n}$, we cannot conclude that n is a prime number. Any a satisfying the relation above but with n being composite is called a **Fermat liar**, and n is called a **Fermat pseudoprime to base** a. If we do pick an integer a such that $a^{n-1} \not\equiv 1 \pmod{n}$, then that is the **Fermat's witness** for the compositeness of n.

For example, let's check the number 221. Picking $a = 38$ from $1 < a < 220$, we see that:

$$a^{n-1} = 38^{220} \equiv 1 \pmod{221}$$

Then either 221 is prime, or 38 is a Fermat liar. Taking another number $a = 24$, we have:

$$a^{n-1} = 24^{220} \equiv 81 \not\equiv 1 \pmod{221}$$

Then 221 is not a prime, and 24 is the Fermat witness of the compositeness of 221. Given the flaws, the algorithm for Fermat's primality test can still tell us if a number is probably a prime or not

It is better than Wilson's theorem to test primality, but we can do better than this probabilistic method. Throughout the years, we have developed very reliable prime checking methods, namely the Miller-Rabin primality test, the AKS primality test, Lucas-Lehmer test (for Mersenne primes, which are primes of the form $2^n - 1$), and elliptic curve primality proving. In the list above, all but the Miller-Rabin test is deterministic, but they are all too slow for practical usage. Therefore, the Miller-Rabin test, though probabilistic, remains the most efficient prime checking method, as it strikes a balance between speed and accuracy, making it the go-to choice for generating large primes where absolute certainty is not necessary (since multiple rounds of the test drastically reduce the probability of error).

4 Prime Numbers in RSA Encryption

4.1 A little history

One of the most well-known applications of prime in the real world is the RSA - When mathematics protects your money. The RSA (Rivest - Shamir - Adleman) is an old and widely used cryptosystem invented in the 1970s. Ron Rivest, Adi Shamir and Leonard Adleman are widely regarded as the designers of the system. Their surnames are used for the name of the scheme. While a similar system was created around the same time by Clifford Cocks, it didn't gain widespread attraction and usage until Rivest, Shamir and Adleman publicly described it in 1977.

To put it simply, a cryptosystem is a set of algorithms that encrypts plain texts into forms that cannot be deciphered without a key - strings of words or numbers that are used to scramble and encrypt the texts in the first place. We will be diving deeper into the way keys are made and used in RSA.

4.2 How does it work?

4.2.1 Key generation and distribution

The RSA is an asymmetric cryptosystem - a cryptosystem that has 2 keys, one being public and the other being private. Fermat's Little Theorem guarantees that modular exponentiation works efficiently, even with large numbers, which is fundamental for RSA encryption and decryption. The public and private keys of the RSA are generated in the following process:

1. Select two prime numbers, in this article will be notated as p and q, which will be kept secret.

2. Compute $n = q.p$, which will be used as the modulus for both the private and public key and will be released as a part of the public key.

3. Compute $\lambda = (p-1)(q-1)$. Here, λ denotes the **Carmichael totient function.**

4. Choose a coprime of λ in the range between 1 and λ, this number will be notated as e and will be released as a part of the public key.

5. Compute d as $d \equiv e^{-1} \pmod{\lambda(n)}$ - this is known as the modular multiplicative inverse of e modular $\lambda(n)$.

After finished generating the public and private key, we have 2 set of numbers, (n, e) is the public key and (d) is the private key. To help you understand the key distribution, suppose we have 2 friends, Alice and Bob, they will exchange their public keys and keep their private key to themselves.

4.2.2 Encryption and decryption

Having obtained Alice's public key, from now on will be notated as (n_a, e_a), Bob can send Alice a number K by first turning it a into k via a process of encryption, represented by this equation:

$$k \equiv K^{e_a} \pmod{n_a}$$

After k is sent to Alice, she can decrypt the message with his private key (d_a) by computing:

$$K^{d_a} \equiv (k^{e_a})^{d_a} \pmod{n_a}$$

Let's look at an example:

1. Alice chose 2 primes $p = 5$ and $q = 11$.

2. He then calculated $n = 5 \cdot 11 = 55$.

3. Next, she computed $\lambda = (5-1)(11-1) = 40$.

4. After that, she chose a coprime of λ, which is $e = 3$.

5. Next, she calculated the modular multiplicative inverse of e modulo λ , which is $d = 27$.

6. Now, Alice has had 2 keys, her public key, which will be sent to Bob, is (55,3) and his private key is (27).

7. Now if Bob wants to send him the number 13, he can encrypt the number using Alice's public key : $52 \equiv 133 \pmod{55}$. After the number 52 has been sent to Alice, he can then decrypt it using his private key: $52^{27} \equiv 13 \pmod{55}$.

$\longrightarrow$ The information has been safely transferred to Alice.

4.2.3 Application

The RSA, despite it being one of the oldest cryptosystems, are still in use today, working as a layer of protection in many digital schemes:

1. Digital signature

2. Communication protocols such as HTTP or SSH

3. Encrypting email messages

4. VPN - Virtual Private Network

5. Protect customer data and transaction record in banks

6. etc