MATH2088/2988
Number Theory and Cryptography
Lecture Notes

Semester 2, 2018

Contents

1 Introduction

1.1 Divisibility

Definition 1.1. Let $a, b \in \mathbb{Z}$. We say that a divides b if there exists an $\alpha \in \mathbb{Z}$ such that

$$b = \alpha a.$$

Properties of divisibility, given that $a, b, c \in \mathbb{Z}$:

- $a \,|\, 0$ $(a \neq 0)$
- $1 \,|\, a$
- If $a \,|\, b$ and $b \,|\, c$, then $a \,|\, c$
- If $a \,|\, b$ and $a \,|\, c$, then $a \,|\, (mb + nc)$, $\forall m, n \in \mathbb{Z}$

1.2 Division with a remainder

Proposition 1.2. Let $a \in \mathbb{Z}$, $b \in \mathbb{Z}^+$. Then there exists unique integers q and r such that

$$a = qb + r, \qquad 0 \leq r < b.$$

Proof. We will use the Least Integer Principle. The Least Integer Principle states that any nonempty subset of $\mathbb{Z}$ contains a minimal element. Consider

$$\mathbb{S}^+ = \{a - kb \,:\, k \in \mathbb{Z},\, a - kb > 0\} \subset \mathbb{N}.$$

If $a \geq 0$, then we can take $k = 0$. If $a < 0$, then we can take $k = a$. From here we can deduce that the set $\mathbb{S}^+$ is nonempty, and hence, it must follow the Least Integer Principle. This means that the set $\mathbb{S}^+$ must contain a minimal element, which we will call $r = a - qb$ ($r \geq 0$ by definition).

$$r - b = a - (q+1)b < 0 \Longrightarrow r < b.$$

Finally, we have

$$a = qb + r, \qquad 0 \leq r < b.$$

Now we can prove the uniqueness of the proposition. Assume for the sake of contradiction that we have (q_1, r_1) and (q_2, r_2) such that

$$a = q_1 b + r_1 = q_2 b + r_2, \qquad 0 \leq r_1 r_2 < b.$$

Subtracting both equations we get

$$(q_1 - q_2)b = r_2 - r_1.$$

If $q_1 > q_2$, then

$$b \leq (q_1 - q_2)b = r_2 - r_2 < b \Longrightarrow b < b,$$

which is a contradiction. If $q_1 < q_2$, we will derive a similar contradiction. Hence, the proposition is unique. □

Remark 1.3. For $a \in \mathbb{Z}^+$, $a \,|\, b$ iff the remainder after the division of b by a is 0. (Note that this is trivial.)

1.3 Greatest common divisor

Definition 1.4. Let $a, b \in \mathbb{Z}$. We say that $d \in \mathbb{Z}$ is the common divisor of a and b if

$$d \mid a,\ d \mid b.$$

- The greatest common divisor is the greatest d with this properly, formally written

$$\gcd(a,b) = \max\{d \in \mathbb{Z} : d \mid a,\ d \mid b\}.$$

- By convention, we have

$$\gcd(0,0) = 0.$$

Properties of gcd, given that $a, b \in \mathbb{Z}$:

- $\gcd(a,b) = \gcd(b,a)$
- $\gcd(a,0) = 0\ (a \geq 0)$
- $\gcd(a,b) = \gcd(-a,b)$

Lemma 1.5. Let $a, b, q \in \mathbb{Z}$. Then

$$\gcd(a,b) = \gcd(a, b-a) = \gcd(a, b-2a) = \cdots = \gcd(a, b-qa).$$

Proof. Consider a common divisor d of a and b. Then

$$d \mid a,\ d \mid b \Longrightarrow d \mid (b-a).$$

This implies that d is a common divisor of a and $b-a$. Now

$$d \mid a,\ d \mid (b-a) \Longrightarrow d \mid (a + (b-a)) \Longrightarrow d \mid b.$$

This implies that d is a common divisor of a and $b-a$. The set of common divisors of (a,b) and $(a, b-a)$ are the same, and hence, their maximums coincide.

$$\therefore \gcd(a,b) = \gcd(a, b-a).$$

By similar arguments,

$$\gcd(a,b) = \gcd(a, b-a) = \gcd(a, b-2a) = \cdots = \gcd(a, b-qa).$$

□

Theorem 1.6 (Euclidean Algorithm)**.** *Let $a \in \mathbb{Z}$ and $b \in \mathbb{Z}^+$. Then $\gcd(a,b)$ can be calculated by the following algorithm:*

$$\begin{aligned} a &= q_1 b + r_1 \\ b &= a_2 r_1 + r_2 \\ r_1 &= q_3 r_2 + r_3 \\ &\cdots \\ r_{n-1} &= q_{n+1} r_n + r_{n+1} \end{aligned}$$

This occurs until we have $r_{n+1} = 0$. Then $\gcd(a,b) = r_n$.

Proof. The algorithm will complete in a finite number of steps, in essence,

$$b > r_1 > r_2 > \cdots > r_n > r_{n+1} \geq 0.$$

Eventually,we will have $r_{n+1} = 0$ and the algorithm will finish. Now computing the gcd:

$$\begin{aligned}
\gcd(a,b) &= \gcd(b,a) \\
&= \gcd(b, a - q_1 b) \\
&= \gcd(b, r_1) \\
&= \gcd(r_1, b) \\
&= \gcd(r_1, b - q_2 r_1) \\
&= \gcd(r_1, r_2) \\
&= \cdots \\
&= \gcd(r_n, r_{n+1}) \\
&= \gcd(r_n, 0) \\
&= 0
\end{aligned}$$

□

1.4 Extended Euclidean Algorithm

Theorem 1.7. *The Euclidean Algorithm shows that* $\gcd(a,b)$ *can be written as*

$$\gcd(a,b) = s \cdot a + t \cdot b, \quad s, t \in \mathbb{Z}.$$

We can write $\gcd(a,b) = r_i$ *in the form*

$$r_i = (-1)^{i+1} k_i a + (-1)^i h_i b,$$

where k_i *and* h_i *are integers.*

Proof. We still start with:

$$r_{-1} = a = 1 \cdot a + 0 \cdot b$$

$$r_0 = b = 0 \cdot a + 1 \cdot b$$

Providing a formula for r_{i+1}:

$$\begin{aligned}
r_{i-1} &= q_{i+1} r_i + r_{i+1} \\
r_{i+1} &= r_{i-1} - q_{i+1} r_i \\
&= ((-1)^i k_{i-1} a + (-1)^{i-1} h_{i-1} b) - q_{i+1}((-1)^{i+1} k_i a + (-1)^i h_i b) \\
&= (-1)^{i+2} \underbrace{(k_{i-1} + q_{i+1} k_i)}_{k_{i+1}} a + (-1)^{i+1} \underbrace{(h_{i-1} + q_{i+1} h_i)}_{h_{i+1}} b
\end{aligned}$$

By induction, or by taking $i = 1, 2, \cdots, n-1$, we end up with

$$r_i = \gcd(a,b) = \underbrace{(-1)^{i+1} k_i}_{s} a + \underbrace{(-1)^i h_i}_{t} b.$$

□

2 Prime and composite numbers

2.1 Introduction to prime and composite numbers

Remark 2.1. The set of all primes is denoted by $\mathbb{P}$.

Definition 2.2. Let $n \in \mathbb{Z}$, $n \geq 2$. n is called *prime* if all of its natural divisors are 1 and n. Otherwise, n is called composite.

Remark 2.3. The numbers 0 and 1 are neither prime nor composite.

Proposition 2.4. Let p be prime, $a, b \in \mathbb{Z}$. If $p \,|\, ab$, then either $p \,|\, a$ or $p \,|\, b$.

Proof. Consider $\gcd(p, a)$. If $\gcd(p, a) = p$, then $p \,|\, a$. Assume that $\gcd(p, a) = 1$. Now by the Extended Euclidean Algorithm, we have

$$1 = s \cdot p + t \cdot a \Longrightarrow b = s \cdot bp + t \cdot ab.$$

It is known that p divides sbp and tab, hence, this implies $p \,|\, b$. □

Theorem 2.5. *If p is prime, and $p \,|\, a_1 a_2 \cdots a_n$, then p divides one of a_1, a_2, $\cdots$, a_n.*

Proof. The proof is similar to the proof of the previous proposition, and hence will be left as an exercise to the reader. □

2.2 Fundamental Theorem of Arithmetic

Theorem 2.6. *Every positive integer can be written as a product of primes in a unique way.*

Proof. We will firstly prove the existence of this theorem. By the principles of mathematical induction,

- 1 is an empty product of primes,
- 2 is prime (a product of one number).

Assume that all numbers between 1 and n can be written as a product of primes. Now we must prove this is true for $n + 1$:
Case 1: $n + 1$ is prime. In this case, $n + 1 = (1)(n + 1)$.
Case 2: $n + 1$ is not prime. $n + 1 = d_1 d_2$, where $1 < d_1 d_2 < n + 1$. Both d_1 and d_2 can be written as a product of primes. Hence, $n + 1 = d_1 d_2$ can also be written as a product of primes.
Now we will prove the uniqueness of this theorem. Assume that

$$n = p_1 p_2 \cdots p_d = q_1 q_2 \cdots q_s,$$

where p_1, p_2, $\cdots$, p_d, q_1, q_2, $\cdots$ and p_s are prime, and $d \leq s$.
Now $p_1 | q_1 q_2 \cdots q_s \Longrightarrow p_1$ divides q_i $(i \in \{q_1, q_2, \cdots, q_s\})$. Without loss of generality, $p_1 | q_1 \Longrightarrow p_1 = q_1$. Now we can divide p_1 from both sides of $n = p_1 p_2 \cdots p_d = q_1 q_2 \cdots q_s$, giving us

$$p_2 \cdots p_d = q_2 \cdots q_s.$$

By repeating the same argument, we get

$$p_2 = q_2, p_3 = q_3, \cdots, p_d = q_d.$$

This equation is only possible if the product on the RHS is empty, in essence $s = d$.
Finally,

$$p_1 = q_1, p_2 = q_2, \cdots p_d = q_d, \quad s = d.$$

□

2.3 Factorisation

All previously learnt methods of factorising have been somewhat inefficient. A more efficient way of factorising is the Fermat Factorisation Method. This method of factorising can quickly factorise a positive integer, given that the positive integer has 2 neighbouring divisors.

Definition 2.7 (Fermat Factorisation Method)**.** If we want to factorise n, first, we compute $\sqrt{n}$. Then we will pick $m \geq n$, where m is as small as possible. We will increase m until $m^2 - n$ is a perfect square.

Theorem 2.8 (Euclid)**.** *There are infinitely many primes.*

Proof. Assume that there are finitely many primes, we will call these primes p_1, p_2, $\cdots$, p_d. Consider

$$N = p_1 p_2 \cdots p_d + 1.$$

By the Fundamental Theorem of Arithmetic, there exists a prime $q|N$. $\gcd(N, p_i) = 1$ for all p_i, $i \in \mathbb{N}$, $1 \leq i \leq d$. This implies that $\gcd(q, p_i) = 1$, which implies that q is not on the list. This is a contradiction. Thus, there are not a finite number of primes. Therefore, there are an infinite number of primes. □

Remark 2.9. $p_1 p_2 \cdots p_d + 1$ is not always prime. There exists a counterexample

$$2 \times 3 \times 5 \times 7 \times 11 \times 13 + 1 = 59509$$

which is not prime.

3 Congruences

3.1 Introduction to congruences

Definition 3.1. Let $m \in \mathbb{Z}^+$ (the modulus), $a, b \in \mathbb{Z}$. We say that a is congruent to b modulo m if $m \,|\, b - a$ or $b = a + mk$ ($k \in \mathbb{Z}$), or if a and b have the same residues modulo m.

The notation used for congruence is

$$a \equiv b \pmod{m}.$$

Basic properties of congruences (given $a, b, c \in \mathbb{Z}$, $m \in \mathbb{Z}^+$):

- $a \equiv a \pmod{m}$
- $a \equiv b \pmod{m} \iff b \equiv a \pmod{m}$
- $a \equiv b \pmod{m} \,\&\, b \equiv c \pmod{m} \implies a \equiv c \pmod{m}$

Definition 3.2. Let $m \in \mathbb{Z}^+$, $a \in \mathbb{Z}$. The congruence class of a modulo m is the set of integers which are congruent to a modulo m.

3.2 Modular arithmetic

Proposition 3.3. Let $m \in \mathbb{Z}^+$ and $a, a', b, b' \in \mathbb{Z}$ such that $a \equiv a' \pmod{m}$ and $b \equiv b' \pmod{m}$. Then

- $a + b \equiv a' + b' \pmod{m}$
- $a \cdot b \equiv a' \cdot b \pmod{m}$

Proof. Let $k, l \in \mathbb{Z}$. Then

$$a \equiv a' \pmod{m} \implies a = a' + km,$$

$$b \equiv b' \pmod{m} \implies b = b' + lm.$$

$a + b \equiv a' + b' \pmod{m}$:

$$\begin{aligned} a + b &= a' + km + b' + lm \\ &= (a' + b') + (l + k)m \\ &\equiv a' + b' \pmod{m} \end{aligned}$$

$a \cdot b \equiv a' \cdot b \pmod{m}$:

$$\begin{aligned} a \cdot b &= (a' + km) \cdot (b' + lm) \\ &= a'b' + a'lm + b'km + klm^2 \\ &= a'b' + (a'l + b'k + klm)m \\ &\equiv a' \cdot b' \pmod{m} \end{aligned}$$

□

Proposition 3.4. Let $m \in \mathbb{Z}^+$, $a, b, c \in \mathbb{Z}$, and $\gcd(c, m) = 1$. Then

$$ac \equiv bc \pmod{m} \implies a \equiv b \pmod{m}.$$

Proof. From the Extended Euclidean Algorithm, for $s, t \in \mathbb{Z}$,

$$1 = sc + tm \implies 1 \equiv sc \text{ (mod } m).$$

Multiplying both sides of $ac = bc$ (mod m) by s, we get

$$asc \equiv bsc \text{ (mod } m).$$

As $1 \equiv sc$ (mod m), then the above expression simplifies to

$$a \equiv b \text{ (mod } m),$$

which completes the proof. □

Remark 3.5. The number s from the proof above is called the inverse of c modulo m. The notation for s is

$$s \equiv c^{-1} \pmod{m}.$$

3.3 Application of congruences

Proposition 3.6. An integer a is divisible by 9 iff the sum of its digits are divisible by 9.

Proof.

$$a = 10^0 d_0 + 10^1 d_1 + 10^2 d_2 + \cdots + 10^n d_n.$$

Consider

$$10 \equiv 1 \pmod 9 \implies 10^j \equiv 1^j \equiv 1 \pmod 9,$$

for $j \in \{0, 1, 2, \cdots, n\}$. Therefore, we can deduce that

$$10^0 \equiv 1^0 \pmod 9,\ 10^1 \equiv 1^1 \pmod 9,\ 10^2 \equiv 1^2 \pmod 9,\ \cdots, 10^n \equiv 1^n \pmod 9.$$

$$\therefore a \equiv d_0 + d_1 + d_2 + \cdots d_n \pmod 9.$$

□

Remark 3.7. There is a similar proof for the fact that an integer a is divisible by 11 iff the alternating sum of its digits are divisible by 11.

3.4 Complete and redcued systems of residues moduluo m

Definition 3.8. A complete system of residues modulo m is a set of integers containing exactly one representative from each congruence class modulo m.

Definition 3.9. A reduced system of residues modulo m is a set of integers containing exactly one representative of each invertible congruence class.

Definition 3.10. The standard reduced system of residues modulo m is given by

$$\{a \in \mathbb{Z} \,:\, 0 \leq a \leq, m-1,\ \gcd(a, m) = 1\}.$$

Definition 3.11. The cardinality of a reduced system of residues modulo m is called Euler's totient function. The notation for Euler's totient function is

$$\varphi(m) := \#\{a \in \mathbb{Z} \,:\, 0 \leq a \leq m-1,\ \gcd(a, m) = 1\}.$$

3.5 Powers in modular arithmetic

Proposition 3.12. Let $m \in \mathbb{Z}^+$, $a \in \mathbb{Z}$, and $\gcd(a, m) = 1$. Then there exists $a, d > 0$ such that

$$a^d \equiv 1 \pmod m.$$

Proof. Compute

$$\underbrace{a^0, a^1, a^2, \cdots, a^m}_{m+1 \text{ values}} \pmod m.$$

Therefore we must have i, j such that $0 \leq i < j \leq m$ and $a^i \equiv a^j \pmod m \implies 1 \equiv a^{j-i} \pmod m$. □

Definition 3.13. Let $m \in \mathbb{Z}^+$, $a \in \mathbb{Z}$, and $\gcd(a, m) = 1$. The order of a modulo m is the smallest $d \in \mathbb{Z}^+$, such that $a^d \equiv 1 \pmod m$.

The notation for order is given by

$$\mathrm{ord}_m(a) = d.$$

4 Cryptography

4.1 Codes vs Ciphers

From a mathematical standpoint, encryption is just a function from one of set of messages (the plaintext) to another set of messages (the ciphertext). The function should be invertible, and decryption is the inverse function.

In everyday English, encoding and decoding are used for such an inverse pair of functions, rather than *encryption* and *decryption*. Note that encoding and enciphering should be kept distinct. The world code should be reserved for situations where there the encoding and decoding processese are not kept secret. An example of this is Morse code.

4.2 Ciphers

4.2.1 Caesar's ciphers

Caesar's cipher was a simple cipher used by Julius Caesar. Represent the letters A - Z by the numbers 0 - 25. To use the cipher, replace each letter i with $i + 3 \pmod{26}$.

4.2.2 Translation ciphers

A translation cipher, or alphabetic shift, is a cipher which encrypts a message by replacing each letter i with $i+k$ (reduced modulo 12) for some fixed k. If we were to decrypt the message, each letter j is to be replaced with $j - k$.

In this case, the key is just the number k (or rather, its residue modulo 26). With this key, anyone will be able to encrypt and decrypt messages.

4.2.3 Simple substitution ciphers

Translation ciphers are an example of simple substitution ciphers, where a message is encrypted by applying the same pre-determined substitution rule to each letter. The encryption key is just the substitution rule, which could be any invertible function f from the source alphabet to the target alphabet. The decryption key is f^{-1}.

Thus, in a simple substitution cipher, if the plaintext is

$$x_1 x_2 \cdots x_N,$$

the ciphertext is

$$f(x_1) f(x_2) \cdots f(x_N).$$

For an arbitrarily simple substitution key on the alphabet A - Z, the key is a permutation of the 26 letters. The number of possible keys is

$$26! = 403291461126605635584000000,$$

which makes an exhaustive key search impossible.

4.2.4 The Vigenère cipher

- A Vigenère cipher is a polyalphabetic translation cipher. That is, m alphabetic shifts are used, for some m known as the period.
- The key for a Vigenère cipher is the m-letter word giving the images of the letter A under the m translations.
- The plaintext M can be thought of as a sequence of residues modulo 26. So $M = c_1c_2c_3\cdots c_N$, where each c_i is a natural number less than 26, and N is the length of the message.
- The keyword $K = k_1k_2\cdots k_m$ is also a sequence of residues modulo 26. Here m is the period.
- Define $k_{m+1} = k_1$, $k_{m+2} = k_2$, etc. More precisely, for each $i \in \mathbb{Z}$ let $k_i = k_r$, where r is the residue of i mod m. The ciphertext is $M' = c'_1c'_2c'_3\cdots c'_N$, where the i-th term c'_i is the mod 26 residue of $c_i + k_i$.
- In particular the sequence $c'_1c'_{m+1}c'_{2m+1}\cdots$ is simply an "alphabetic shift" of the sequence $c_1c_{m+1}c_{2m+1}\cdots$. To get c'_{am+1} you just add k_1 to c_{am+1} (mod 26).
- We define the decimation of M with period m and index r to be the sequence $\mathrm{Dec}(M, m, r) = c_rc_{m+r}c_{2m+r}c_{3m+r}\cdots$.

4.3 Finding the period using the coincidence index

- The coincidence index of a piece of text is the probability that two randomly chosen letters are the same. That is, if the relative frequencies of the 26 letters are $p_0, p_1, \cdots, p_{25}$ then the coincidence index is
$$\sum_{i=0}^{25} p_i^2.$$
- By the Cauchy–Schwarz inequality, the coincidence index is always at least $\frac{1}{26} = 0.0385\cdots$. The more skewed the distribution of frequencies is, the higher the coincidence index will be. For English text the coincidence index is usually about 0.065.
- An alphabetic shift (or any simple substitution cipher) does not change the coincidence index. In particular, if a Vigenère cipher has period m, the decimations $\mathrm{Dec}(M', m, i)$ will have coincidence index about 0.065.
- This provides a convenient way to find m: compute the coincidence index of $\mathrm{Dec}(M', m, i)$ for $m = 1, 2, 3, \cdots$ until we find an m that gives a value greater than about 0.06.

4.4 Finding the key using the period

- Suppose that the period is m. Then
$$\mathrm{Dec}(M', m, 1) = c'_1c'_{m+1}c'_{2m+1}c'_{3m+1}\cdots$$
is the same as
$$\mathrm{Dec}(M, m, 1) = c_1c_{m+1}c_{2m+1}c_{3m+1}\cdots$$
alphabetically shifted by k_1, where k_1 is the first term of the key.

- We can find k_1 by examining letter frequencies in $\mathrm{Dec}(M', m, 1)$: whatever is the most frequent letter in $\mathrm{Dec}(M', m, 1)$ is probably the shift of E by k_1, and we can check that guess easily using the next most frequent letters.
- And we can similarly find $k_2, k_3, \ldots, k_m$ by examining letter frequencies in $\mathrm{Dec}(M', m, 2)$, $\mathrm{Dec}(M', m, 3)$, …, $\mathrm{Dec}(M', m, m)$.
- For the above methods to work one needs a piece of ciphertext of length many times the length of the period m. Otherwise the decimations $\mathrm{Dec}(M', m, i)$ will not be long enough to give meaningful frequency distributions. In the extreme case that m is greater than the length of the ciphertext, it is impossible to decrypt the message without knowing the key.

4.5 Transposition ciphers

pass

5 Euler-Fermat Theorem

5.1 Reduced systems of residues – extension

Proposition 5.1. Let $a, b \in \mathbb{Z}$, $m \in \mathbb{Z}^+$, $\gcd(a, m) = 1$. If $m \mid ab$, then $m \mid b$.

Proof. By the Extended Euclidean Algorithm,

$$1 = sa + tm,$$

for some $s, t \in \mathbb{R}$.

$$\implies b = sab + tmb$$

As m divides both sab and tmb, then $m \mid b$. □

Proposition 5.2. Let $a, b \in \mathbb{Z}$. If $\gcd(a, m) = \gcd(b, m) = 1$, then $\gcd(ab, m) = 1$.

Proof. Assume that $\gcd(ab, m) = d > 1$.

$$\implies d \mid m, \gcd(a, m) = 1 \implies \gcd(a, d) = 1$$

But $d \mid ab$, and by Proposition 8.1, $d \mid b$.

$$\implies \gcd(b, m) \geq d$$

But this contradicts with $\gcd(b, m) = 1$.

$$\therefore \gcd(ab, m) = 1$$

□

5.2 Euler-Fermat Theorem

Proposition 5.3. Let R be a reduced system of residues modulo m. Let $a \in \mathbb{Z}$, $\gcd(a, m) = 1$. Then $aR := \{ar : r \in R\}$ is also a reduced system of residues modulo m.

Proof. We must firstly show that all elements in aR are distinct modulo m.
Assume

$$ar_1 \equiv ar_2 \pmod{m}.$$

Then

$$r_1 \equiv r_2 \pmod{m} \Longrightarrow r_1 = r_2.$$

ar is comprime with $m \Longrightarrow \gcd(ar, m) = 1$.
Hence, all values ar from aR are distinct representatives from each invertible congruence class $\Longrightarrow aR$ is a reduced system. □

Remark 5.4. The same proposition is true for complete system of residues.

Theorem 5.5 (Euler-Fermat Theorem). *Let $m \in \mathbb{Z}^+$, $a \in \mathbb{Z}$, $\gcd(a, m) = 1$. Then*

$$a^{\varphi(m)} \equiv 1 \pmod{m}.$$

Proof. Consider R as a reduced system of residues mod m. In essence,

$$R = \{r_1,\, r_2,\, \cdots,\, r_{\varphi(m)}\}.$$

aR is also a reduced system of residues mod m.

$$\Longrightarrow r_1,\, ar_2,\, \cdots,\, ar_{\varphi(m)} \text{ are congruent to } r_1,\, r_2,\, \cdots,\, r_{\varphi(m)} \text{ in a different order.}$$

$$\begin{aligned} \Longrightarrow r_1 r_2 \cdots r_{\varphi(m)} &\equiv ar_1 \cdot ar_2 \cdot \cdots \cdot ar_{\varphi(m)} \\ &\equiv a^{\varphi(m)} r_1 r_2 \cdots r_{\varphi(m)} \pmod{m} \end{aligned}$$

$$\Longrightarrow 1 \equiv a^{\varphi(m)} \pmod{m} \qquad \text{(we can cancel } r_1,\, r_2,\, \cdots,\, r_{\varphi(m)} \text{ from both sides.)}$$

□

Corollary 5.6 (Fermat's Little Theorem). Let p be prime, $a \in \mathbb{Z}$, $a^{p-1} \not\equiv 0 \pmod{p}$. Then

$$a^{p-1} \equiv 1 \pmod{p}.$$

Proof. We know that $\varphi(p) = p - 1$, where p is prime. Now letting $m = p$ in the Euler-Fermat Theorem, we get the desired result. □

Proposition 5.7. If $m \in \mathbb{Z}^+$, $a \in \mathbb{Z}$ and $\gcd(a, m) = 1$. Then

$$\operatorname{ord}_m(a) \mid \varphi(m).$$

$$d = \operatorname{ord}_m(a)$$

Divide $\varphi(m)$ by d with the remainder

$$\varphi(m) = qd + r, \quad 0 \le r < d.$$

$$a^{\varphi(m)} = a^{qd+r} = (a^d)^q \cdot a^r \equiv a^r \pmod{m}$$

as $a^d \equiv 1 \pmod{m}$. But by the Euler–Fermat Theorem,

$$a^{\varphi(m)} \equiv 1 \pmod{m} \Longrightarrow a^r \equiv 1 \pmod{m}.$$

Since d is the smallest positive integer with $a^d \equiv 1 \pmod{m}$, r has to be 0. Hence

$$d \mid \varphi(m).$$

5.3 Fermat's Little Theorem when $\gcd(a,p) \neq 1$

Theorem 5.8. *Let p be prime for some arbitrary $a \in \mathbb{Z}$. Then*

$$a^p \equiv a \pmod{p}.$$

Proof. If $\gcd(a,p) = 1$, then

$$a^{p-1} \equiv 1 \pmod{p} \Longrightarrow a^p \equiv a \pmod{p}.$$

If $\gcd(a,p) > 1$, then

$$a \equiv 0 \pmod{p} \Longrightarrow 0 \equiv a^p \equiv a \pmod{p}.$$

□

Proposition 5.9. If the input of the Euler's totient function is a product of two primes, say pq, where p and q are prime, then

$$\varphi(pq) = (p-1)(q-1).$$

Proof. $\gcd(a,pq)$ can equal to 1, p, q and pq.

$$\gcd(a,pq) = pq$$

for $a = 0$ (considering $0 \leq a < pq$).

$$\gcd(a,pq) = p$$

for $a = p, 2p, 3p, \cdots, (q-1)p$.

$$\gcd(a,pq) = q$$

for $a = q, 2q, 3q, \cdots, (p-1)q$.

$$\Longrightarrow \varphi(pq) = pq - 1 - (q-1) - (p-1) = pq - p - q + 1 = (p-1)(q-1)$$

□

Proposition 5.10. Let $m = pq$, where p and q are distinct primes. Let $a \in \mathbb{Z}$. Then

$$a^{\varphi(m)+1} \equiv a \pmod{m}.$$

Proof.

1. If $\gcd(a,m) = 1$, then

$$a^{\varphi(m)} \equiv 1 \pmod{m} \Longrightarrow a^{\varphi(m)+1} \equiv a \pmod{m}.$$

2. If $\gcd(a,m) = pq = m$, then

$$a \equiv 0 \pmod{m} \Longrightarrow a^{\varphi(m)+1} \equiv a \equiv 0 \pmod{m}.$$

3. If $\gcd(a,m) = \gcd(a,pq) = p$, then by Euler's Theorem ($\gcd(a,q) = 1$),

$$a^{q-1} \equiv 1 \pmod{q} \Longrightarrow a^{(q-1)(p-1)} \equiv 1 \pmod{q} \Longrightarrow a^{\varphi(m)+1} \equiv a \pmod{q}.$$

$$a \equiv 0 \pmod{p} \Longrightarrow a^{\varphi(m)+1} \equiv a \equiv 0 \pmod{p}$$

We have

$$p \mid a^{\varphi(m)+1} - a, q \mid a^{\varphi(m)+1} - a \Longrightarrow pq \mid a^{\varphi(m)+1} - a$$
$$\Longrightarrow a^{\varphi(m)+1} \equiv a \pmod{pq}.$$

□

Theorem 5.11 (RSA Theorem)**.** *Let $m = pq$, where p and q are distinct primes. Let $a \in \mathbb{Z}$, $k \in \mathbb{Z}^+$. Then*

$$a^{k\varphi(m)+1} \equiv a \pmod{m}.$$

Proof. We will conduct this proof by induction. $k = 0$ is obvious. $k = 1$ is true by a previous proposition. Assume that the congruence is true for k. We will now prove that the congruence is true for $k + 1$.

$$a^{(k+1)\varphi(m)+1} \equiv \underbrace{a^{k\varphi(m)+1}}_{\equiv a \text{ (by assumption)}} a^{\varphi(m)} \equiv a^{\varphi(m)+1} \equiv a \pmod{m}$$

□

Corollary 5.12. Let p and q be distinct primes and consider $d \in \mathbb{Z}$ such that $\gcd(d, \varphi(pq)) = 1$, $e \equiv d^{-1} \pmod{\varphi(pq)}$. Then the following two functions are inverses of each other:

$$m = pq$$

$$\begin{tikzcd}
\{0,1,2,\cdots,m-1\} \arrow[r, bend left, "a \to a^d \pmod{m}"] & \{0,1,2,\cdots,m-1\} \arrow[l, bend left, "b \to b^d \pmod{m}"]
\end{tikzcd}$$

Proof. $(a^d)^e \equiv a \pmod{m}$
We have $ed \equiv 1 \pmod{\varphi(m)}$ or $ed = 1 + k\varphi(m)$ for some $k \in \mathbb{Z}$

$$\therefore (a^d)^e \equiv a^{ed} \equiv a^{k\varphi(m)+1} \equiv a \pmod{m}.$$

□

6 Relating congruences with different moduli

6.1 Principles

Proposition 6.1 (Principle 1)**.** Let $m_1, m_2 \in \mathbb{Z}^+$ and $m_1 \mid m_2$. Then

$$a \equiv b \pmod{m_2} \Longrightarrow a \equiv b \pmod{m_1}.$$

Proof. $m_2 = dm_1$ for some $d \in \mathbb{Z}$. Hence

$$a \equiv b \pmod{m_2} \Longrightarrow b = a + km_2 \Longrightarrow b = a + kdm_1 \Longrightarrow a \equiv b \pmod{m_1}.$$

□

Remark 6.2. The converse of the above proposition is not true, however, we can state a weaker statement:

$$a \equiv b \pmod{m_1} \Longrightarrow a \equiv b + m_2 - m_1 \pmod{m_2}$$

Proposition 6.3 (Principle 2)**.** If $a \equiv b \pmod{m}$, then $ac \equiv bc \pmod{mc}$.

Proof. $m \,|\, ab \Longrightarrow mc \,|\, (a-b)c = ac - bc \Longrightarrow ac \equiv bc \pmod{mc}$. □

Remark 6.4. The converse of the above proposition is true.

Proposition 6.5. Let $m \in \mathbb{Z}^+$, $a, b \in \mathbb{Z}$ and

$$a \equiv b \pmod{m}.$$

If $d \,|\, a$ and $d \,|\, m$, then d also divides b and

$$\frac{a}{d} \equiv \frac{b}{d} \pmod{\frac{m}{d}}.$$

Proof. $b = a + km \Longrightarrow d \,|\, b$. Then we can write $a = da_1$, $b = db_1$, $m = dm_1$ and

$$db_1 = da_1 + kdm_1 \Longrightarrow b_1 = a_1 + km_1 \Longrightarrow b_1 \equiv a \pmod{m_1}$$

□

Corollary 6.6. Every linear congruence

$$ax \equiv b \pmod{m}$$

either does not have integer solutions, or have integer solutions of the form

$$x \equiv c \pmod{m_1}$$

for some values c and m_1.

When solving linear congruences, there are three steps:

1. Compute $\gcd(a, m)$
2. If $\gcd(a, m) = 1$, then there exists $a^{-1} \pmod{m}$ and
$$x \equiv a^{-1}b \pmod{m}.$$
3. Assume $\gcd(a, m) = d > 1$. If $d \nmid d$, there are no solutions. If $d \,|\, b$, we can rewrite the congruence
$$\frac{a}{d} \equiv \frac{b}{d} \pmod{\frac{m}{d}}.$$
Now that $\gcd(\frac{a}{d}, \frac{m}{d}) = 1$, repeat step 2.

Proposition 6.7. If

$$a \equiv b \pmod{m_1}$$
$$a \equiv b \pmod{m_2}$$

and $\gcd(m_1, m_2) = 1$, then

$$a \equiv b \pmod{m_1 m_2}.$$

Proof.

$$a \equiv b \pmod{m_1} \Longrightarrow m_1 \,|\, a - b$$
$$a \equiv b \pmod{m_2} \Longrightarrow m_2 \,|\, a - b$$

Since $\gcd(m_1, m_2) = 1$, it implies that $m_1 m_2 \,|\, a - b \Longrightarrow a \equiv b \pmod{m_1 m_2}$. □

Theorem 6.8 (Chinese Remainder Theorem)**.** *Let $m_1, m_2 \in \mathbb{Z}^+$, $\gcd(m_1, m_2) = 1$. Then for any pair $a, b \in \mathbb{Z}$, the system of congruences*

$$x \equiv a \pmod{m_1}$$

$$x \equiv b \pmod{m_2}$$

is equivalent to $x \equiv c \pmod{m_1 m_2}$ for some integer c.

Proof. We will firstly prove the existence of this theorem. By the Extended Euclidean Algorithm, we can write

$$1 = sm_1 + tm_2$$

for some $s, t \in \mathbb{Z}$.

$$b_1 = b_1 sm_1 + b_1 tm_2$$

Consider $b_1 tm_2$ modulo m_1 and m_2:

$$b_1 tm_2 \equiv b_1 \pmod{m_1}$$

$$b_1 tm_2 \equiv 0 \pmod{m_2}.$$

Now multiply both sides of $1 = sm_1 + tm_2$ by b_2 and consider $b_2 sm_1$ modulo m_1 and m_2:

$$b_2 tm_1 \equiv 0 \pmod{m_1}$$

$$b_2 tm_1 \equiv b_2 \pmod{m_2}.$$

Now consider $b_1 tm_2 + b_2 sm_1$:

$$b_1 tm_2 + b_2 sm_1 \equiv b_1 \pmod{m_1}$$

$$b_1 tm_2 + b_2 sm_1 \equiv b_2 \pmod{m_2}.$$

Hence, we have the solution

$$x = b_1 tm_2 + b_2 sm_1.$$

Now we will prove the uniqueness of this theorem. Consider two solutions, c and c' of the system

$$c \equiv b_1 \equiv c' \pmod{m_1}$$

$$c \equiv b_2 \equiv c' \pmod{m_2}.$$

$$\implies c - c' \equiv 0 \pmod{m_1},\ c - c' \equiv 0 \pmod{m_2} \implies c - c' \equiv 0 \pmod{m_1 m_2}$$

$$\implies c \equiv c' \pmod{m_1 m_2}$$

Now we can check that any $x \equiv c \pmod{m_1 m_2}$ is the solution of the system (this is left as an exercise to the reader). □

Theorem 6.9 (Chinese Remainder Theorem (Full version))**.** *Let $m_1, m_2, \cdots, m_d \in \mathbb{Z}^+$ be pairwise coprime, meaning that $\gcd(m_i, m_j) = 1$ for any $i \neq j$. Then for any $b_1, b_2, \cdots, b_d \in \mathbb{Z}$, the system*

$$x \equiv b_1 \pmod{m_1}$$

$$x \equiv b_2 \pmod{m_2}$$

$$\cdots$$

$$x \equiv b_d \pmod{m_d}$$

has a unique solution modulo $m_1 m_2 \cdots m_d$.

Proof. This proof is based off the two congruences version of the Chinese Remainder Theorem. Note that the system of the first two congruences is equivalent to

$$x \equiv c_1 \pmod{m_1 m_2}.$$

Replace the first two congruences with the congruence above. Now notice that $\gcd(m_1 m_2, m_i) = 1$ for $3 \leq i \leq d$. From here, we will decrease the number of congruences in the system until we have one:

$$x \equiv c \pmod{m_1 m_2 \cdots m_d}.$$

□

7 Computing powers in modulo arithmetic

7.1 Several approaches

Let's compute

$$2^{2016} \pmod{1739} = 2^{2016} \pmod{37 \times 47}.$$

Approach 1 (naive). Compute

$$2^1, 2^2, \cdots, 2^{2016}.$$

This is inefficient as it requires a total of 2015 multiplications.

Approach 2. Use the Euler-Fermat Theorem.

$$2^{\varphi(1739)} = \equiv 1 \pmod{1739}$$

$$\implies 2^{2016} \equiv 2^{360} \pmod{1739},$$

which is quicker to compute, but still takes long (359 multiplications).

Approach 3 (successive squaring).

1. Write number as a sum of powers of 2. In this case,

$$2^{360} = 2^8 + 2^6 + 2^5 + 2^3.$$

2. Compute the sequence $a_n \equiv 2^{2^n} \pmod{()}m$. In this case, $m = 1739$. Notice that

$$a_{n+1} = a_n^2 \pmod{1739}.$$

n	0	1	2	3	4	5	6	7	8
$2^{2^n} \pmod{1739}$	2	4	16	256	1193	747	1529	625	1089

3. Here, the computation of your power of 2 is trivial.

Approach 4. Use Fermat's little theorem, then use the Chinese Remainder Theorem.

8 Computing kth roots in modular arithmetic

We want to solve

$$x^k \equiv a \pmod{m}.$$

In other words, for given $m \in \mathbb{Z}^+$, $a \in \mathbb{Z}$, $k \in \mathbb{Z}^+$, we want to find an integer x which satisfies the above congruence. There are several restrictions:

- $\gcd(k, \varphi(m)) = 1$
- $\gcd(a, m) = 1$ (In some cases, we can drop this restriction. For example, for $m = p$ or $m = pq$, where $p \neq q$ are primes (see RSA Theorem))

A quick way of computing kth roots would be:

1. Compute $\varphi(m)$
2. Compute s and t (where $s, t \in \mathbb{Z}$) such that
$$1 = sk + t\varphi(m) \qquad \text{(EEA)}$$
3. Compute $x \equiv a^s \pmod{m}$

Remark 8.1. This method does not guarantee that the solution x will be unique. In fact, it is unique, but we do not prove it here.

9 Multiplicative functions

Definition 9.1. A function $f : \mathbb{Z}^+ \to \mathbb{Z}$ is called multiplicative if for any $m, n \in \mathbb{Z}$, with $\gcd(m, n) = 1$, we have

$$f(mn) = f(m)f(n).$$

f is called completely multiplicative if the above equation is true for all pairs $m, n \in \mathbb{Z}$.

9.1 Euler's phi function

Recall that

$$\varphi(n) := \#\{a \in \mathbb{Z} \, : \, 0 \leq a < n, \, \gcd(a, n) = 1\},$$

where

- $\varphi(p) = p - 1$
- $\varphi(p^n) = p^n - p^{n-1}$
- $\varphi(pq) = (p - 1)(q - 1)$ for distinct primes p and q.

Theorem 9.2. *Euler's phi function is multiplicative*

Proof. We will show that $\varphi(mn) = \varphi(m)\varphi(n)$ for coprime m and n. Firstly, we will construct one-to-one correspondence between the following sets:

$$\{x \in \mathbb{Z} \, : \, 0 \leq x < mn, \, \gcd(x, mn) = 1\}$$

$$\Longleftrightarrow$$

$$\{y \in \mathbb{Z} \,:\, 0 \leq y < m,\, \gcd(y,m) = 1\} \times \{z \in \mathbb{Z} \,:\, 0 \leq z < n,\, \gcd(z,n) = 1\}$$

Now we consider the set of pairs (y, z) such that $0 \leq y < m$, $0 \leq z < n$, $\gcd(y,m) = \gcd(z,n) = 1$. We map x to a pair (y, z) so that

$$y \equiv x \pmod{m} \qquad \text{and} \qquad z \equiv x \pmod{n},$$

where $f(x) := (y, z)$. We will now check that the function is injective, that is, $f(x) = f(x')$ implies that $x = x'$. Assume that $f(x) = f(x')$

$$\implies x \equiv x' \equiv y \pmod{m} \qquad \text{and} \qquad x \equiv x' \equiv z \pmod{n}.$$

By Principle 3, we have that

$$x \equiv x' \pmod{mn}$$

and hence

$$x = x'.$$

Now we will check that the function is surjective, that is, every element (y, z) has at least one preimage. We are given (y, z), whereby $0 \leq y < m$, $\gcd(y, m) = 1$ and $0 \leq z < n$, $\gcd(z, n) = 1$. Now we need to find x such that

$$x \equiv y \pmod{m} \qquad \text{and} \qquad x \equiv z \pmod{n}.$$

By the Chinese Remainder Theorem, such x exists. We can make x such that $0 \leq x < mn$ by taking the remainder after division of x by mn. Hence

$$\gcd(y, m) = 1 \implies \gcd(x, m) = 1.$$

Similarly, we have

$$\gcd(x, n) = 1.$$

Hence,

$$\gcd(x, mn) = 1,$$

which implies that the map f is a bijection, and thus, the cardinalities of the two sets coincide. Therefore, we can conclude that

$$\varphi(mn) = \varphi(m)\varphi(n)$$

for distinct primes m, n. □

Proposition 9.3. Let

$$n = p_1^{\alpha_1} \cdot p_2^{\alpha_2} \cdot \cdots \cdot p_d^{\alpha_d}$$

be the factorisation of n as a product of primes. Then

$$\begin{aligned}\varphi(n) &= (p_1^{\alpha_1} - p_1^{\alpha_1 - 1}) \cdot (p_2^{\alpha_2} - p_2^{\alpha_2 - 1}) \cdot \cdots \cdot (p_d^{\alpha_d} - p_d^{\alpha_d - 1}) \\ &= n\left(1 - \frac{1}{p_1}\right) \cdot \left(1 - \frac{1}{p_2}\right) \cdot \cdots \cdot \left(1 - \frac{1}{p_d}\right).\end{aligned}$$

Proof.

$$\begin{aligned}\varphi(p_1^{\alpha_1} \cdot p_2^{\alpha_2} \cdot \cdots \cdot p_d^{\alpha_d}) &= \varphi(p_1^{\alpha_1}) \cdot \varphi(p_2^{\alpha_2}) \cdot \cdots \cdot p_d^{\alpha_d} \\ &= (p_1^{\alpha_1} - p_1^{\alpha_1 - 1}) \cdot (p_2^{\alpha_2} - p_2^{\alpha_2 - 1}) \cdot \cdots \cdot (p_d^{\alpha_d} - p_d^{\alpha_d - 1}) \\ &= p_1^{\alpha_1}\left(1 - \frac{1}{p_1}\right) \cdot p_2^{\alpha_2}\left(1 - \frac{1}{p_2}\right) \cdot \cdots \cdot p_d^{\alpha_d}\left(1 - \frac{1}{p_d}\right) \cdot \\ &= n\left(1 - \frac{1}{p_1}\right) \cdot \left(1 - \frac{1}{p_2}\right) \cdot \cdots \cdot \left(1 - \frac{1}{p_d}\right)\end{aligned}$$

□

9.2 Liouville and Mobius functions

Definition 9.4. Let

$$n = p_1^{\alpha_1} \cdot p_2^{\alpha_2} \cdot \dots \cdot p_d^{\alpha_d}.$$

Then the Liouville functon is defined as

$$\lambda(n) = (-1)^{\alpha_1+\alpha_2+\dots+\alpha_d}.$$

Theorem 9.5. *$\lambda(n)$ is completely multiplicative.*

Proof. Let

$$n = p_1^{\alpha_1} \cdot p_2^{\alpha_2} \cdot \dots \cdot p_d^{\alpha_d}$$

and

$$m = p_1^{\beta_1} \cdot p_2^{\beta_2} \cdot \dots \cdot p_d^{\beta_d}.$$

Then

$$\begin{aligned}\varphi(mn) &= \lambda\left(p_1^{\alpha_1+\beta_1} \cdot p_2^{\alpha_2+\beta_2} \cdot \dots \cdot p_d^{\alpha_d+\beta_d}\right)\\ &= (-1)^{\alpha_1+\beta_1+\alpha_2+\beta_2+\dots+\alpha_d+\beta_d}\\ &= (-1)^{\alpha_1+\alpha_2+\dots+\alpha_d} \cdot (-1)^{\beta_1+\beta_2+\dots+\beta_d}\\ &= \varphi(n)\varphi(m).\end{aligned}$$

□

Definition 9.6. The Mobius function is defined as

$$\mu(n) := \begin{cases} \lambda(n) & \text{if } n \text{ is squarefree} \\ 0 & \text{otherwise.} \end{cases}$$

Remark 9.7. n is called squarefree if it is not divisible by any square of an integer except 1.

Remark 9.8. $\mu(n)$ is multiplicative but not completely multiplicative.

9.3 $\lambda(n)$, $\mu(n)$ and the distribution of primes

$\lambda(n)$ and $\mu(n)$ are closely related with the distribution of primes. It is known that

$$\frac{1}{N}\sum_{n=1}^{N} \lambda(n) \to 0$$

as $n \to \infty$. This is equivalent to the *Prime Number Theorem.*

Theorem 9.9 (Prime Number Theorem)**.** *The number of primes between 1 and N is approximately*

$$\frac{N}{\ln N}.$$

The Riemann hypothesis is equivalent to: $\forall \varepsilon > 0$, we have

$$\frac{1}{N^{\frac{1}{2}+\varepsilon}}\sum_{n=1}^{N} \lambda(n) \to 0 \qquad \text{and} \qquad \frac{1}{N^{\frac{1}{2}+\varepsilon}}\sum_{n=1}^{N} \mu(n),$$

as $n \to \infty$.

9.4 The number and sum of divisors

Definition 9.10. $\tau(n)$ denotes the number of positive integer divisors of n.

Some properties of $\tau(n)$ include

- $\tau(p) = 2$ for primes p
- $\tau(p^k) = k + 1$.

We can also represent $\tau(n)$ by

$$\tau(n) = \sum_{d \,|\, n} 1.$$

Definition 9.11. $\sigma(n)$ denote the sum of positive integer divisors of n.

Some properties of $\sigma(n)$ include

- $\sigma(p) = p + 1$ for primes p
- $\sigma(p^k) = 1 + p + p^2 + \cdots + p^k = \frac{p^{k+1}-1}{p-1}$.

Before we go on with the next proposition, we will firstly need a lemma.

Lemma 9.12. For $n, m \in \mathbb{Z}^+$ with $\gcd(n, m) = 1$, the map given by

$$(d_1, d_2) \longrightarrow d_1 d_2$$

between the sets

$$\{d_1 \in \mathbb{Z}^+ \,:\, d_1 \,|\, n\} \times \{d_2 \in \mathbb{Z}^+ \,:\, d_2 \,|\, n\} \qquad \text{and} \qquad \{d \in \mathbb{Z}^+ \,:\, d \,|\, mn\}$$

is a bijection.

Proof. We will first prove that the map is surjective, that is, any $d \,|\, mn$ has a preimage (d_1, d_2). Consider

$$m = p_1^{\alpha_1} \cdot p_2^{\alpha_2} \cdot \cdots \cdot p_d^{\alpha_d} \qquad \text{and} \qquad n = q_1^{\beta_1} \cdot q_2^{\beta_2} \cdot \cdots \cdot q_d^{\beta_d},$$

whereby m and n are prime factorised and all p_i's and q_i's are distinct. Now consider $d \,|\, mn$. For d_2, we will take all powers of p_i from the factorisation of d. For d_1, we will take all the powers of q_i from the factorisation of d. Hence

$$d_1 \,|\, n \quad \text{and} \quad d_2 \,|\, m \Longrightarrow d_1 d_2 = d.$$

Now we will prove that the map is injective, that is, if $d_1 d_2 \,|\, nm$ and $d_1' d_2' \,|\, nm$ with $d_1 d_2 = d_1' d_2'$, then $d_1 = d_1'$ and $d_2 = d_2'$.

$$d_1' \,|\, d_1' d_2' = d_1 d_2,$$

but as $\gcd(d_1', d_2) = 1$, then $d_1' \,|\, d_1$. Similarly, $d_1 \,|\, d_1' \implies d_1 = d_1'$. From here, we have that $d_2 = d_2'$, which completes the proof. □

Proposition 9.13. Let $f : \mathbb{Z}^+ \to \mathbb{Z}$ be a multiplicative function. Then the function $F : \mathbb{Z}^+ \to \mathbb{Z}$ defined as

$$F(n) := \sum_{d \,|\, n} f(d)$$

is also multiplicative.

Proof.

$$\begin{aligned}
F(nm) &= \sum_{d\,|\,nm} f(d) \\
&= \sum_{d_1\,|\,n,\, d_2\,|\,m} f(d_1 d_2) \qquad \text{(by the previously proven lemma)} \\
&= \sum_{d_1\,|\,n,\, d_2\,|\,m} f(d_1) f(d_2) \\
&= \sum_{d_1\,|\,n} \sum_{d_2\,|\,m} f(d_1) f(d_2) \\
&= \left(\sum_{d_1\,|\,n} f(d_1) \right) \left(\sum_{d_2\,|\,m} f(d_2) \right) \\
&= F(n)F(m)
\end{aligned}$$

□

Corollary 9.14. $\tau(n)$ and $\sigma(n)$ are multiplicative.

Proof. Easy. □

9.5 Applications of the multiplicative nature of $\sigma(n)$: classification of perfect numbers

Definition 9.15. $n \in \mathbb{Z}^+$ is called perfect if it is equal to the sum of all of its proper divisors. In essence,

$$n = \sigma(n) - n \Longrightarrow 2n = \sigma(n).$$

Remark 9.16. It is not known if there are infinitely many perfect numbers.

Remark 9.17. It is not known if there exists an odd perfect number.

Theorem 9.18. *An even number is perfect iff it is of the form*

$$n = 2^k(2^{k+1} - 1)$$

and $2^{k+1} - 1$ *is prime.*

Proof. We will write n in the form $n = 2^k \cdot m$, where m is an odd integer. Hence

$$\sigma(n) = \sigma(2^k \cdot m) = \sigma(2^k)\sigma(m) = (2^{k+1} - 1)\sigma(m).$$

On the other hand, $\sigma(n) = 2n$,

$$\Longrightarrow 2^{k+1} - 1 \,|\, 2n = 2^{k+1}m \Longrightarrow m = (2^{k+1} - 1)l,\ l \in \mathbb{Z}.$$

$$2n = 2^{k+1}(2^{k+1} - 1)l = \sigma(n) = (2^{k+1} - 1)\sigma((2^{k+1} - 1)l) \Longrightarrow 2^{k+1}l = \sigma((2^{k+1} - 1)l).$$

Assume that $l > 1$. Then

$$\sigma((2^{k+1} - 1)l) \geq 1 + (2^{k+1} - 1) + l + (2^{k+1} - 1)l > 2^{k+1}l,$$

which is a contradiction. Now we will take $l = 1$. Then the equation transforms into

$$2^{k+1} = \sigma(2^{k+1} - 1).$$

This equation is only possible if $2^{k+1} - 1$ is prime. Finally, $n = 2^k(2^{k+1} - 1)$ and $2^{k+1} - 1$ is prime. From here, we just need to check that $n = 2^k(2^{k+1} - 1)$ is perfect, which is trivial. □

Remark 9.19. Prime number of the form $2^{k+1} - 1$ are called Mersenne primes. Only 50 of them are known at the moment. The largest Mersenne prime is

$$2^{77232917} - 1.$$

9.6 More on Euler's phi function

Proposition 9.20.

$$\sum_{d \mid n} \varphi(d) = n.$$

Proof. The LHS and RHS of the equation are multiplicative functions. TO verify that they are the same, we need to compare them at the powers of primes. Let $n = p^k$, then we have

$$\begin{aligned}\sum_{d \mid p^k} \varphi(d) &= \varphi(1) + \varphi(p) + \varphi(p^2) + \cdots + \varphi(p^k)\\ &= 1 + (p-1) + (p^2 - p) + \cdots + (p^k - p^{k-1})\\ &= p^k\\ &= n.\end{aligned}$$

□

Proposition 9.21. Let $n \in \mathbb{Z}^+$, $d \in \mathbb{Z}^+$ where $d \mid n$. Then

$$\#\{a \in \mathbb{Z} : 0 \leq a < n, \, \gcd(a, n) = d\}$$

is equal to $\varphi\left(\frac{n}{d}\right)$.

Proof. $\gcd(a, n) = d$ implies that $a = db$ and $n = de$, where $b, e \in \mathbb{Z}$. Now we can consider $\gcd(b, e) = 1$ (if $\gcd(b, e) = f > 1$, then $df \mid a$ and $df \mid n$ which is a contradiction). Hence

$$0 \leq a < n \Longleftrightarrow 0 \leq db < n \Longleftrightarrow 0 \leq b < e.$$

Therefore b belongs to

$$\{b \in \mathbb{Z} : 0 \leq b < e, \, \gcd(b, e) = 1\} := B.$$

From here we will check that the cardinalities of B and the initial set coincide, which will then imply that

$$\#B = \varphi(e) = \varphi\left(\frac{n}{d}\right) = \#\{a \in \mathbb{Z} : 0 \leq a < n, \, \gcd(a, n) = d\}.$$

□

9.7 Mobius Inversion Formula

Proposition 9.22. Let $F(n) := \sum_{d \mid n} \mu(d)$. Then

$$F(n) = \begin{cases} 1 & \text{if } n = 1 \\ 0 & \text{if } n > 1. \end{cases}$$

Proof. To check that the equation holds $\forall n \in \mathbb{Z}^+$, we just need to check the equation for powers of primes ($n = p^k$) as both functions involved are multiplicative. Evaluating $F(n)$, we have

$$F(n) = F(p^k) = \underbrace{\mu(1)}_{=1} + \underbrace{\mu(p^1)}_{=-1} + \underbrace{\mu(p^2)}_{=0} + \cdots + \underbrace{\mu(p^k)}_{=0} = 0$$

which completes the proof. □

Theorem 9.23 (Mobius Inversion Formula)**.** *Suppose that we have $n \in \mathbb{Z}^+$ and numbers a_d for all divisors d of n. Then the following system of equations*

$$\sum_{e \mid d} x_e = a_d$$

over variables x_e where e runs over all divisors of n has a unique solution

$$x_e = \sum_{n \mid e} \mu\left(\frac{e}{n}\right) \cdot a_n.$$

Proof. We write the system of equations in matrix form $M\mathbf{x} = \mathbf{a}$, where

- $\mathbf{x} = (x_e)_{e \mid n}$ is the vector of unknown numbers
- $\mathbf{a} = (a_d)_{d \mid n}$ is the vector of known numbers
- $M = (m_{d,e})_{d,e \mid n}$ is the matrix where

$$m_{d,e} = \begin{cases} 1 & \text{if } e \mid d \\ 0 & \text{otherwise.} \end{cases}$$

Matrix M is triangular (with zeroes above the diagonal and with ones on the diagonal). Hence, $\det M = 1$, which implies that the system has a unique solution given by

$$\mathbf{x} = M^{-1}\mathbf{a}.$$

Now we need to compute M^{-1}. More specifically, we need to verify that

$$M = P = (p_{e,h})_{e,h \mid n}$$

where

$$p_{e,h} = \begin{cases} \mu\left(\frac{e}{h}\right) & \text{if } h \mid e \\ 0 & \text{otherwise.} \end{cases}$$

Now we compute MP, we need to check that this product is the identity matrix I.

$$MP = \begin{pmatrix} m_{11} & \cdots & m_{1n} \\ \vdots & & \vdots \\ m_{d1} & \cdots & m_{dn} \\ \vdots & & \vdots \\ m_{n1} & \cdots & m_{nn} \end{pmatrix} \begin{pmatrix} p_{11} & \cdots & p_{1h} & \cdots & p_{1n} \\ \vdots & & \vdots & & \vdots \\ p_{n1} & \cdots & p_{nh} & \cdots & p_{nn} \end{pmatrix}$$

The entry (d, h) of MP is equal to

$$m_{d1}p_{1h} + m_{d2}p_{2h} + \cdots + m_{dn}p_{nh}.$$

We can evaluate the above expression:

$$\begin{aligned} m_{d1}p_{1h} + m_{d2}p_{2h} + \cdots + m_{dn}p_{nh} &= \sum_{e \,|\, n} \left\{ \begin{matrix} 1 & \text{if } e \,|\, d \\ 0 & \text{otherwise} \end{matrix} \right\} \left\{ \begin{matrix} \mu\left(\frac{e}{h}\right) & \text{if } h \,|\, e \\ 0 & \text{otherwise} \end{matrix} \right\} \\ &= \sum_{\text{all } e \,|\, n \text{ with } e \,|\, d,\ h \,|\, e} \mu\left(\frac{e}{h}\right) \\ &= \sum_{\text{all } k \text{ with } hk \,|\, d} \mu(k) \\ &= \begin{cases} \sum_{k \,|\, \frac{d}{h}} \mu(k) & \text{if } h \,|\, d \\ 0 & \text{if } h \nmid d \end{cases} \\ &= \begin{cases} 1 & \text{if } \frac{d}{h} = 1 \\ 0 & \text{otherwise} \end{cases} \quad \text{(by the proposition).} \end{aligned}$$

This value coincides with the (d, h) entry of the identity matrix. Hence $MP = I$ and $M^{-1} = P$. □

Corollary 9.24 (Mobius Inversion Formula for multiplicative functions)**.** Let $f(n)$ and $F(n)$ be two multiplicative functions such that

$$F(n) = \sum_{d \,|\, n} f(d).$$

Then $f(n)$ can be restored from $F(n)$ by the formula

$$f(n) = \sum_{d \,|\, n} \mu\left(\frac{n}{d}\right) F(d).$$

Proof. From applying the Mobius Inversion Formula with $a_d = F(d)$, the proof is complete. □

10 RSA cryptosystem

10.1 The process of the RSA cryptosystem

‘RSA’ comes from the names of the inventors, Rivest, Shamir and Adleman. Description of RSA:

1. Bob chooses two large prime numbers p and q. Bob computes $n = pq$ and then $\varphi(n) = (p-1)(q-1)$. Bob then chooses an encryption exponent e such that $\gcd(e, \varphi(n)) = 1$. Bob computes the decryption exponent $d \equiv e^{-1} \pmod{\varphi(n)}$.
2. Bob publishes the public key (n, e) and keeps $p, q, \varphi(n)$ and d a secret.
3. Alice encodes the message to get a sequence $[m_1, m_2, \cdots]$ where $m_i \in \underbrace{\{0, 1, \cdots, n-1\}}_{\text{alphabet}}$.
4. Alice encrypts the message by computing $[m_1', m_2', \cdots]$, where $m_i' \equiv m_i^e \pmod{n}$.
5. Bob decrypts the message by computing $m_i \equiv (m_i')^d \pmod{n}$.

Now to check that this cryptosystem works, we need to verify

$$(m_i')^d \equiv m_i \pmod{n}.$$

Since we have $m_i' \equiv m_i^e \pmod{n}$ and $d \equiv e^{-1} \pmod{\varphi(n)}$, then

$$de \equiv 1 \pmod{\varphi(n)} \qquad \text{or} \qquad de = k\varphi(n) + 1.$$

Then

$$(m_i')^d \equiv (m_i^e)^d \equiv m_i^{k\varphi(n)+1} \equiv m_i \pmod{n} \qquad \text{(RSA Theorem)}.$$

10.2 Digital signature with the help of RSA

Here, only Alice can encode the message and everyone can decode it. Description of the algorithm:

1. Alice's set-up is the same as before.
2. Alice publishes the public key (n, e), keeping everything else as a secret.
3. Alice encodes the message
$$[m_1, m_2, \cdots];\ m_i \in \{0, 1, \cdots, n-1\}.$$
4. Alice encrypts the message by replacing m_i with $m_i' \equiv m_i^d \pmod{n}$.
5. Bob decrypts the message by replacing m_i' with $(m_i')^e \pmod{n}$.

11 Computational complexity

The general question involved with computing is: How long will it take for a computer to perform computations?

11.1 Elementary bit operations

Computers store numbers in a binary form, in essence,

$$n = (b_{k-1}b_{k-2}\cdots b_1 b_0)_2,$$

where each bit $b_i \in \{0, 1\}$. The numbers from

$$(\underbrace{100\cdots 0}_{k \text{ digits}})_2 = 2^{k-1} \qquad \text{to} \qquad (\underbrace{111\cdots 1}_{k \text{ digits}})_2 = 2^{k-1} + 2^{k-2} + \cdots + 2^1 + 2^0$$

consist of k bits. Given that $n \in \mathbb{Z}^+$, the number of bits required to store n is the unique number $k \in \mathbb{Z}$ such that

$$2^{k-1} \leq n < 2^k \implies k = \lfloor \log_2 n \rfloor + 1.$$

Remark 11.1. The number of bits grows much slower than n.

Let's consider 2 numbers, m, which contains k bits, and n, which contains l bits. Without loss of generality, we can assume that $k \geq l$. Then $m + n$ has either k bits or $k + 1$ bits. Also, mn has either $k + l - 1$ or $k + l$ bits.

Generally, you need k bit operations to compute $m + n$ or $m - n$. $k(l - 1)$ bit operations is needed to compute mn.

Theorem 11.2 (Karatsuba)**.** *Let $M(k)$ be the number of bit operations needed to multiply two k-bit numbers. Then*

$$M(2k) \leq 3M(k) + 10k.$$

Proposition 11.3. Let $l \in \mathbb{Z}^+$. Then the number of bit operations to multiply 2^l-bit numbers can be estimated to be

$$10(3^l - 2^l).$$

Proof. We will prove this by induction on l. If $l = 1$, $10(3^l - 2^l) = 10$, and we can multiply 2-bit numbers in 10 bit operations by long multiplication. Hence this is true for $l = 1$. Now we assume that $M(2^l) \leq 10(3^l - 2^l)$. We are now required to prove that the statement is true for $l + 1$. We have

$$\begin{aligned} M(2^{l+1}) &\leq 3M(2^l) + 10 \cdot 2^l \\ &\leq 3 \cdot 10(3^l - 2^l) + 10 \cdot 2^l \\ &= 10 \cdot 3^{l+1} - 10 \cdot 2^{l+1} \\ &= 10(3^{l+1} - 2^{l+1}) \end{aligned}$$

which completes the proof by induction. □

Proposition 11.4. The number of bit operations to multiply two k-bit numbers can be estimated to be

$$M(k) \leq 30k^{\log_2 3}.$$

Proof. Let l a number such that

$$2^{l-1} < k \leq 2^l.$$

We make k-bit numbers larger by adding zeroes, such that they will become 2^l-bit numbers. Hence

$$M(k) \leq M(2^l) \leq 10(3^l - 2^l).$$

From $2^{l-1} < k \leq 2^l$, we have that

$$l \geq \log_2 k \qquad \text{and} \qquad l < \log_2 k + 1,$$

which implies that

$$l = \lceil \log_2 k \rceil < \log_2 k + 1.$$

Then

$$\begin{aligned} M(k) &\leq 10(3^{\log_2 k+1} - 2^{\log_2 k+1}) \\ &\leq 30 \cdot 3^{\log_2 k} \\ &= 30 \cdot 3^{\log_3 k \cdot \log_2 3} \\ &= 30 \cdot k^{\log_2 3}, \end{aligned}$$

completing the proof. □

11.2 Big O notation

Definition 11.5. Let $f(n)$ and $g(n)$ be two positive valued functions, defined over the positive integers. We say that

$$g(n) = O(f(n))$$

if there exists positive numbers C, N such that

$$g(n) \leq Cf(n), \qquad \forall n \geq N.$$

When we computed the bits required for long division, it came out to be $k(l-1)$. Now if both numbers are k-bit, then this means that the number of bits required is

$$k(l-1) = k(k-1) = O(k^2).$$

Karatsuba requires $30 \cdot k^{\log_2 3}$ bit operations and hence,

$$30 \cdot k^{\log_2 3} = O(k^{\log_2 3}) = O(k^2).$$

Definition 11.6. An algorithm is said to be a polynomial time algorithm if the number of bit operations required to perform it on k-bit numbers is $O(k^a)$ for some $a > 0$.

Proposition 11.7. If

$$\lim_{k\to\infty} \frac{f(k)}{g(k)} = L < +\infty,$$

then $f(k)$ is $O(g(k))$. If

$$\lim_{k\to\infty} \frac{f(k)}{g(k)} = \infty,$$

then $f(k)$ is not $O(g(k))$.

Proof. From the definition of a limit, $\forall \varepsilon > 0$, $\exists N = N(\varepsilon)$ such that

$$L - \varepsilon < \frac{f(k)}{g(k)} < L + \varepsilon \qquad \text{for } k \geq N.$$

If we take $\varepsilon = 1$, then

$$f(k) < (L+1)g(k) \qquad \text{for } k \geq N(1).$$

Hence, $f(k)$ is $O(g(k))$. Proving the second part of the proposition is trivial. □

Remark 11.8. Big O notation only provides an estimate for $f(k)$ from above, not from below.

12 Computational complexity of some standard algorithms

12.1 Division with a remainder

Now we want an algorithm that considers $a, b \in \mathbb{Z}^+$, where it will find $q, r \in \mathbb{N}$ such that

$$a = qb + r.$$

For k-bit numbers, the algorithm requires less than k subtractions, whereby each subtraction will require up to k elementary bit operations, and less than k comparisons. In total, we will have less than $k^2 + k = O(k^2)$ operations, and hence, our algorithm is polynomial time.

12.2 Computation of gcd

Given numbers $a, b \in \mathbb{N}$, whereby the numbers have less than or equal to k bits, what is the algorithm to find $\gcd(a, b)$?

Now the naive approach will try all the numbers from 1 to $\min\{a, b\}$ and then take the largest element from those set of numbers. This requires taking $\min\{a, b\}$ numbers, which is $O(2^k)$. Trial division of a and b by that number will require $O(k^2)$ operations. In total, the complexity of the algorithm is $O(2^k k^2)$, which is not polynomial time.

Now consider the Euclidean algorithm, that is, given $a, b \in \mathbb{Z}^+$, which are numbers that are at most k bits. Then the algorithm is given by

$$\begin{aligned} a &= q_1 b + r_1 \\ b &= q_2 r_1 + r_2 \\ &\cdots \\ r_{n-3} &= q_{n-1} r_{n-2} + \underbrace{r_{n-1}}_{\neq 0} \\ r_{n-2} &= q_n r_{n-1} + \underbrace{r_n}_{=0}. \end{aligned}$$

Each iteration of the algorithm takes $O(k^2)$ bit operations, meaning that in total, the number of bit operations required is given by $O(nk^2)$.

Proposition 12.1. For each $i \in \{0, \cdots, n-4\}$, we have $r_{i+2} < \frac{1}{2} r_i$.

Proof. Firstly, we will consider when $r_{i+1} < \frac{1}{2} r_i$. Then

$$r_{i+2} < r_{i+1} \leq \frac{1}{2} r_i \implies r_{i+2} < \frac{1}{2} r_i.$$

Now if $r_{i+1} > \frac{1}{2} r_i$, then we have that

$$r_i = q_{i+2} r_{i+1} + r_{i+2} \implies r_{i+2} = r_i - q_{i+2} r_{i+1} \leq r_i - r_{i+1} < r_i - \frac{1}{2} r_i = \frac{1}{2} r_i.$$

□

Corollary 12.2. The number of iterations in the Euclidean algorithm is at most $2k$.

Proof.

$$1 \leq r_{n-1} \leq \frac{1}{2} r_{n-3} < \frac{1}{2^2} r_{n-5} < \cdots < \frac{1}{2^{\lfloor \frac{n}{2} \rfloor}} r_{n-1-2\lfloor \frac{n}{2} \rfloor} \implies 2^{\lfloor \frac{n}{2} \rfloor} < 2^k \implies n \leq 2k.$$

□

In conclusion, we can say that the computational complexity of the Euclidean algorithm is $O(nk^2) = O(k^3)$, and thus, the Euclidean algorithm is polynomial time.

12.3 Computation of a power modulo a number

Suppose that we have an input of three numbers, $a, b, m \in \mathbb{Z}^+$, where each number is less than or equal to k bits long. From this, we want to compute

$$a^b \pmod{m}.$$

When we compute each term of this sequence, the computation will require one multiplication of k-bit numbers, and then we will take the remainder after division by m. This will take $O(k^2)$ bit operations. Since we have k terms, the complexity of this step is $O(k^3)$.

Now we will compute the product of the a^{2^i}'s (from successive squaring) to then make the expression $a^b \pmod{m}$. For this step, we will have less than or equal to $k-1$ multiplications modulo m. The complexity of this step is also $O(k^3)$. Hence, the overall complexity is $O(k^3)$, and thus, our algorithm is polynomial time.

12.4 Checking primality

Now we are given an $n \in \mathbb{Z}^+$, which is up to k bits. The desired task is to check if n is prime (no need for factorisation). Now a fast way to execute this algorithm would be to follow the following steps,

1. Take a random value $a \in \{2, 3, \cdots, n-1\}$
2. Compute $\gcd(a, n)$
3. If the result is not 1, then n is composite
4. If the result is 1, then compute
$$a^{n-1} \pmod{n}$$
5. If the result is not 1, then n is composite
6. If the result is 1, then we will choose a different a and retry all the steps.

Theorem 12.3 (Agrawal-Kayal-Saxena, 2002)**.** *The primality of a number $n \in \mathbb{Z}^+$ can be checked in polynomial time.*

Proof. no thanks. □

Definition 12.4. A number n is called a pseudoprime for the base a if

$$a^{n-1} \equiv 1 \pmod{n}$$

and n is composite.

Definition 12.5. A number n is called a Carmichael number if

$$a^{n-1} \equiv 1 \pmod{n}$$

is satisfied for any $\gcd(a, n) = 1$.

13 Polynomial congruences

We want to solve

$$a_d x^d + a_{d-1} x^{d-1} + \cdots + a_1 x + a_0 \equiv 0 \pmod{m}.$$

Here, x is unknown, $a_0, a_1, \cdots, a_d \in \mathbb{Z}$, $a_d \not\equiv 0 \pmod{m}$. m is called the modulus.

Theorem 13.1 (Principle 1)**.** *We can replace a_i with another coefficient $a_i' \equiv a_i \pmod{m}$. This will not change the set of solutions.*

Theorem 13.2 (Principle 2)**.** *If x is a solution, then $y \equiv x \pmod{m}$ is also a solution.*

Proposition 13.3. Let p be prime, where $p \equiv 3 \pmod{4}$. Then the congruence

$$x^2 \equiv -1 \pmod{p}$$

does not have any solutions.

Proof. Assume that x is a solution of

$$x^2 \equiv -1 \pmod{p}.$$

Now compute

$$x^{p-1} \equiv (x^2)^{\frac{p-1}{2}} \equiv (-1)^{\frac{p-1}{2}} \equiv -1 \pmod{p}$$
$$\implies 1 \equiv -1 \pmod{p},$$

which is a contradiction. □

Proposition 13.4. Let p be prime, where $p \equiv 1 \pmod{4}$. Then the congruence

$$x^2 \equiv -1 \pmod{p}$$

has two solutions modulo p.

Proof. Consider $(p-1)! \pmod{p}$.

$$1 \cdot 2 \cdot 3 \cdot \cdots \cdot \underbrace{(p-4)}_{\equiv -4} \cdot \underbrace{(p-3)}_{\equiv -3} \cdot \underbrace{(p-2)}_{\equiv -2} \cdot \underbrace{(p-1)}_{\equiv -1} \equiv \left(\frac{p-1}{2}\right)! \cdot (-1)^{\frac{p-1}{2}} \cdot \left(\frac{p-1}{2}\right)!$$
$$\equiv \left(\left(\frac{p-1}{2}\right)!\right)^2 \pmod{p}$$

Now we will split all of the elements of $\{1, 2, 3, \cdots, p-1\}$ into pairs (a, a^{-1}). Consider the numbers such that $a \equiv a^{-1} \pmod{p} \iff a^2 \equiv 1 \pmod{p} \equiv a \equiv 1$ or $-1 \pmod{p}$.

$$(p-1)! \equiv 1(-1) \cdot (2 \cdot 2^{-1}) \cdot (3 \cdot 3^{-1}) \cdot \cdots \equiv -1 \pmod{p}.$$

Hence,

$$\left(\left(\frac{p-1}{2}\right)!\right)^2 \equiv -1 \pmod{p}$$

as required. □

From the proof above, we can see that the two solutions of the congruence are

$$x \equiv \left(\frac{p-1}{2}\right)! \pmod{p} \qquad \text{and} \qquad x \equiv -\left(\frac{p-1}{2}\right)! \pmod{p}.$$

Theorem 13.5. *Let p be prime. Consider the congruence*

$$a_d x^d + a_{d-1}x^{d-1} + \cdots + a_1 x + a_0 \equiv 0 \pmod{p},$$

where $a_i \in \mathbb{Z}$, $a_d \not\equiv 0 \pmod{p}$. The set of solutions of this congruence is a union of at most d congruent classes modulo p.

Proof. We will prove this by induction on d. First we sub $d = 1$. Then we have

$$a_1 x + a_0 \equiv 0 \pmod{p}.$$

This congruence has a solution

$$x \equiv -a_0 a_1^{-1} \pmod{p},$$

and hence, the statement is true for $d = 1$. Now we will assume that the statement is true for d, meaning that we will now prove it for $d+1$. Hence

$$f(x) \equiv 0 \pmod{p}, \qquad \deg f = d+1.$$

If there are no solutions to this congruence, then we have nothing to prove. So we consider a solution c such that

$$f(c) \equiv 0 \pmod{p}.$$

Then

$$\begin{aligned} f(x) &\equiv f(x) - f(c) = a_{d+1}(x^{d+1} - c^{d+1}) + a_d(x^d - c^d) + \cdots + a_1(x-c) \\ &\equiv (x-c)(a_{d+1}(x^d c^0 + x^{d-1}c^1 + x^{d-2}c^2 + \cdots x^0 c^d) + a_d(\cdots) + \cdots + a_1) \\ &\equiv (x-c)g(x) \pmod{p}, \end{aligned}$$

where $g(x)$ is a polynomial of degree d.

$$f(x) \equiv 0 \pmod{p} \Longleftrightarrow (x-c)g(x) \equiv 0 \pmod{p}$$

$$\Longleftrightarrow p \mid (x-c)g(x)$$

$$\Longleftrightarrow p \mid x - c \qquad \text{or} \qquad p \mid g(x)$$

$$\Longrightarrow x \equiv c \pmod{p} \qquad \text{(one solution)}, \qquad g(x) \equiv 0 \pmod{p} \qquad (\leq d \text{ solutions}).$$

In total, $f(x) \equiv 0 \pmod{p}$ has $\leq d+1$ solutions. □

14 Primitive roots and discrete logarithms

Consider the congruence

$$x^d \equiv c \pmod{p},$$

where $d \in \mathbb{Z}^+$, $c \in \mathbb{Z}$ and p is prime. This congruence has $\leq d$ solutions.

Theorem 14.1. *Let p be prime, $d \in \mathbb{Z}^+$ such that $d \mid p-1$. Then for any integer $c \not\equiv 0 \pmod{p}$, the congruence*

$$x^d \equiv c \pmod{p}$$

either has no solutions or d solutions.

Proof. Consider the map

$$\begin{aligned} f : \{1, 2, \cdots, p-1\} &\to \{1, 2, \cdots, p-1\} \\ x &\mapsto x^d. \end{aligned}$$

Consider some c in the range (image) of f,

$$c \equiv x^d \pmod{p}.$$

Define $e = \frac{p-1}{d}$. Now we compute c^e,

$$c^e \equiv (x^d)^e \equiv x^{de} \equiv x^{p-1} \equiv 1 \pmod{p}.$$

Hence, any c from the range of f is a solution of $x^e \equiv 1 \pmod{p}$. Now we know that the range of f has $\leq e$ elements. Fixing c from the range of f and looking at x such from the domain of f such that $x^d \equiv c \pmod{p}$. By the theorem, there are $\leq d$ such elements. In total, there are $\leq ed = p-1$ elements in the domain of f. But there are exactly $p-1$ elements in $\{1, 2, \cdots, p-1\}$. Thus, all $\leq$ in the proof become $<$, and there are e elements c in the range of f and for each c from the range,

$$x^d \equiv c \pmod{p}$$

has d solutions. □

14.1 Primes and order

We will recall that if p is prime and $a \in \mathbb{Z}$ such that $\gcd(a, p) = 1$, then the order of a modulo p is the smallest $d \in \mathbb{Z}^+$ such that

$$a^d \equiv 1 \pmod{p}.$$

We know that

1. $a^d \equiv a^{d'} \pmod{p} \iff d \equiv d' \pmod{\mathrm{ord}_p(a)}$
2. $a^d \equiv 1 \pmod{p} \iff \mathrm{ord}_p(a) \mid d$
3. By Fermat's Little Theorem, $a^{p-1} \equiv 1 \pmod{p} \implies \mathrm{ord}_p(a) \mid p-1$.

Theorem 14.2. *Let p be prime, where $d \mid p-1$. Then the number of values $a \in \{1, \cdots, p-1\}$ such that $ord_p(a) = d$, is $\varphi(d)$.*

Proof. Let

$$F(d) := \#\{a \in \{1, \cdots, p-1\} \,:\, \mathrm{ord}_p(a) = d\}.$$

Then

$$\begin{aligned} d &= \#\{a \in \{1, \cdots, p-1\} \,:\, a^d \equiv 1 \pmod{p}\} \\ &= \#\{a \in \{1, \cdots, p-1\} \,:\, \mathrm{ord}_p(a) \mid d\} \\ &= \sum_{e \mid d} \#\{a \in \{1, \cdots, p-1\} \,:\, \mathrm{ord}_p(a) = e\} \\ &= \sum_{e \mid d} F(e). \end{aligned}$$

Applying the Mobius Inversion Formula, we get

$$F(d) = \sum_{e \,|\, d} \mu\left(\frac{d}{e}\right) e = \varphi(d).$$

□

Definition 14.3. Let $m \in \mathbb{Z}^+$. Then $a \in \mathbb{Z}$ is called a primitive root modulo m if $\gcd(a, m) = 1$ and $\text{ord}_m(a) = \varphi(m)$.

Corollary 14.4. Primitive roots modulo a prime p always exist. Moreover, there are $\varphi(p-1)$ of them.

Remark 14.5. The previous corollary is not true for composite values m.

Definition 14.6. Let b be a primitive root modulo a prime p and $a \in \mathbb{Z}$ with $\gcd(a, p) = 1$. Then the discrete logarithm of a modulo p ($\log_{b,p}(a)$) is the value $d \in \{0, \cdots, p-2\}$ such that

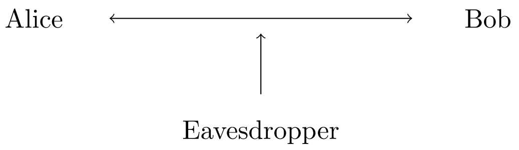
$$b^d \equiv a \pmod{p}.$$

For $a \equiv 0 \pmod{p}$, $\log_{b,p}(a)$ does not exist.

Remark 14.7. For $\log_{b,p}(a)$, the input is a residue modulo p, but the output is a residue modulo $p-1$.

15 Diffie – Hellman key exchange protocol and Elgamal cryptosystem

We now want to establish a shared secret key by only using non secured communication channels.



15.1 Algorithm (Diffie – Hellman)

In this algorithm, there are 5 steps:

1. Alice carefully chooses a prime p, a $b \in \{1, 2, \cdots, p-1\}$ and a private key $x \in \{1, 2, \cdots, p-2\}$. From this, she computes $k \equiv b^x \pmod{p}$.
2. Alice sends the triple (p, b, k) to Bob, keeping x a secret.
3. Bob chooses his own private key $y \in \{1, \cdots, p-2\}$ and computes $c \equiv b^y \pmod{p}$.
4. Bob sends to Alice the value c, keeping y as a secret.
5. Alice and Bob compute the common secret key, m:
   - Alice: $m \equiv c^x \equiv b^{xy} \pmod{p}$;
   - Bob: $m \equiv k^y \equiv b^{xy} \pmod{p}$.

The eavesdropper knows the values of p, b, k, c. They need to compute $m \equiv b^{xy} \pmod{p}$ from this information.

It is believed (not proven) that this task requires a solution of the discrete logarithm problem, that is, given p, b, k, we need to find x which solves

$$b^x \equiv k \pmod{p}.$$

This problem is believed to be extremely difficult for large and carefully chosen values of p.

15.2 Elgamal cryptosystem

In the Elgamal cryptosystem, everyone can encrypt the message, but only Alice can decrypt it (like in RSA).

$$\begin{tikzcd}
 & \text{Gob} \arrow[d] & \\
\text{Rob} \arrow[r] & \text{Alice} & \text{Bob} \arrow[l]
\end{tikzcd}$$

In this algorithm, there are 9 steps:

1. As before, Alice chooses a triple (p, b, k) and computes $k \equiv b^x \pmod{p}$.
2. Alice publishes the triple (p, b, k).
3. Bob chooses y and computes $c \equiv b^y \pmod{p}$.
4. Bob encodes the message M such that it is written as a sequence of numbers $\Longrightarrow [M_1, M_2, \cdots, M_d]$, where $M_i \in \{1, \cdots, p-1\}$.
5. Bob computes the shared secret key $S \equiv k^y \pmod{p}$.
6. Bob encrypts the message by computing $M_i' \equiv SM_i \pmod{p}$.
7. Bob sends the following information to Alice: $\langle C, [M_1', M_2', \cdots, M_d'] \rangle$.
8. Alice computes a shared secret key $S \equiv c^x \pmod{p}$.
9. Alice decrypts the message: $M_i \equiv S^{-1} M_i' \pmod{p}$.

15.3 Analysis of the Diffie – Hellman key exchange

Knowing p, b, k, c, we want to find $S \equiv b^{xy} \pmod{p}$. The naive approach to this would be to compute

$$b^0, b^1, b^2, b^3, \cdots \pmod{p}$$

until we find k or c. This method will require up to $\text{ord}_p(b)$ multiplications ($\text{ord}_p(b)$ is maximised if b is a primitive root, and hence, $\text{ord}_p(b) = p - 1$). There is an algorithm (Pohlig-Hellman) which computes the discrete logarithm quickly if all prime divisors of $\text{ord}_p(b) = p - 1$ are small. Now, $p - 1$ should have a large prime divisor. $p - 1$ is even, however, $\frac{p-1}{2}$ is prime. Such a prime is called a *safe prime.*

Conjecture 15.1. There are infinitely many safe primes.

Remark 15.2. For all primes $q \neq 2$, we either have $q \equiv 1 \pmod 4$ or $q \equiv 3 \pmod 4$. Hence, every safe $p = 2q + 1$ (except 5) is congruent to 3 modulo 4.

Proposition 15.3. There are infinitely many primes $p \equiv 3 \pmod 4$.

Proof. Assume that there are finitely many primes

$$p_1, p_2, \cdots, p_d.$$

Consider $N = 4p_1p_2 \cdots p_d - 1$. From this we have $p_i \nmid N$ and $2 \nmid N$. Hence, all prime divisors of n are congruent to 1 modulo 4.

$$-1 \equiv N \equiv q_1q_2 \cdots q_s \equiv 1^s \equiv 1 \pmod 4$$
$$\implies -1 \equiv 1 \pmod 4,$$

which is a contradiction and hence, we have completed the proof. □

Proposition 15.4. There are infinitely many primes which are congruent to 1 modulo 4.

Proof. Assume hat there are finitely many primes

$$p_1, p_2, \cdots, p_d.$$

Consider $N = (2p_1p_2 \cdots p_d)^2 + 1$. From this we have $p_i \nmid N$ and $2 \nmid N$. Hence, all prime divisors of n are congruent to 3 modulo 4.

We will take a prime q such that $q \mid N$. Then $x = 2p_1p_2 \cdots p_d$ is a solution of the congruence

$$x^2 + 1 \equiv 0 \pmod q.$$

But we know that the congruence has no solutions for $q \equiv 3 \pmod 4$, which is a contradiction, and hence, we have completed the proof. □

16 Applications of primitive roots

Given a prime p, how do we find a primitive root modulo p? We know that there are $\varphi(p-1)$ primitive roots modulo p. Hence, the probability that a randomly chosen $b \in \{1, \cdots, p-1\}$ is a primitive root is equal to

$$\frac{\varphi(p-1)}{p-1}.$$

Let $p - 1 = q_1^{\alpha_1} \cdot q_2^{\alpha_2} \cdot \cdots \cdot q_d^{\alpha_d}$ be the prime factorisation of $p-1$. Then

$$\begin{aligned}\frac{\varphi(p-1)}{p-1} &= \frac{(p-1)\left(1-\frac{1}{q_1}\right)\left(1-\frac{1}{q_2}\right)\cdot \cdots \cdot \left(1-\frac{1}{q_d}\right)}{p-1} \\ &= \left(1-\frac{1}{q_1}\right)\left(1-\frac{1}{q_2}\right)\cdot \cdots \cdot \left(1-\frac{1}{q_d}\right).\end{aligned}$$

In theory, if $p-1$ is a product of many primes, this expression can be very small. But in practice, the number of tries for a primitive root is usually ≤ 10.

Now we know that if we want to find a primitive root modulo a prime p, we take candidates randomly from $\{1, 2, \cdots, p-1\}$ and check if they are primitive roots.

Now, if we want to check if $a \in \{1, 2, \cdots, p-1\}$ is a primitive root modulo p, we check if $\text{ord}_p(a) = p - 1$. If this is true, then we can conclude that a is a primitive root modulo p. We know that $\text{ord}_p(a) \mid p - 1$. Therefore, a is a primitive root modulo $p \iff a^d \not\equiv 1 \pmod{p}$ for all proper divisors d of $p-1$. To check this, we need to find the prime factorisation of $p-1$, which may be difficult.

16.1 Solving equations of the form $x^m \equiv c \pmod{p}$

An important property of a primitive root is that if a is a primitive root modulo p, then

$$\{a^0, a^1, a^2, \cdots, a^{p-2}\}$$

is a reduced set of residues modulo p. In other words, $\{a^0, a^1, a^2, \cdots, a^{p-2}\}$ coincides with the set $\{1, 2, \cdots, p-1\}$ in a different order (i.e. $a^0, a^1, a^2, \cdots, a^{p-2}$ are all unique).

Proposition 16.1. Let p be prime, and let a be a primitive root modulo p, where $d \mid p-1$. Then the number a^n is a solution of

$$x^d \equiv 1 \pmod{p}$$

if and only if $n = ke = k\left(\frac{p-1}{d}\right)$.

Proof. Trivial. □

Now consider the congruence

$$x^m \equiv c \pmod{p},$$

where $\gcd(m, p-1) = 1$. Let a be a primitive root modulo p. We will write

$$x \equiv a^p \pmod{p}, \qquad c \equiv a^b \pmod{p}.$$

Then $a^{my} \equiv a^b \pmod{p} \iff my \equiv b \pmod{p-1} \iff y \equiv m^{-1}b \pmod{p-1} \iff x \equiv a^{m^{-1}b} \equiv c^{m^{-1}} \pmod{p}$.

17 Algorithms for factorisation and DLP

17.1 Factorisation

We have a problem: Given a composite number n, find one of its non-trivial factors. One way to approach this would be to conduct trial division. We will try small primes p between 2 and $\sqrt{n}$. By the prime number theorem,

$$\#\{p \text{ is prime} : p \leq N\} \approx \frac{N}{\ln N}$$

and hence, the number of primes to test is approximately

$$\frac{\sqrt{n}}{\ln \sqrt{n}}.$$

Note that this algorithm is not polynomial time in $k = \log_2 n$. Another method to solve this problem would be to take a sequence of values a and compute $\gcd(a, n)$ (which can be done quickly by EEA). If $\gcd(a, n) > 1$, then this will give us a non-trivial divisor of n.

Now a question arises, how many tries do we expect if we choose $a \in \{1, 2, \cdots, n-1\}$ randomly? For simplicity, $n = pq$ for prime p, q, where $p, q \approx \sqrt{n}$, $\gcd(a, n) > 1$. Hence, a is a multiple of p or a multiple of q. Thus, the probability is

$$\frac{1}{p} + \frac{1}{q} \approx \frac{2}{\sqrt{n}}.$$

The expected number of tries is approximately $\frac{\sqrt{n}}{2}$, which is not better than trial division.

Now since we have had poor methods to solve this problem, we now have a better method known as the Pollard-Rho Method (1975): We compose the sequence $t_i \in \{0, 1, \cdots, p-1\}$ as follows:

$$\begin{aligned} t_0 &= 1 \\ t_{i+1} &\equiv t_i^2 + 1 \pmod{n}. \end{aligned}$$

Note that every element is between 0 and $n-1$, hence, the sequence has to repeat after $\leq n$ terms. However, in many cases, it repeats much quicker.

Now in the Pollard-Rho method, we expect $t_i \equiv t_j \pmod{p}$ for some $0 \leq i < j \leq \text{constant} \times \sqrt{p}$. We compute

$$\gcd(t_i - t_j, n)$$

for $0 \leq i < j \leq \text{constant} \times n^{\frac{1}{4}}$. The problem is, the computation require $O(n^{\frac{1}{2}})$ computations of gcd, which is not better than trial division. The solution to this is in the following proposition.

Proposition 17.1. Assume that $t_i \equiv t_j \pmod{n}$ for some $i < j$. Then there exists an integer l, $i \leq l < j$ such that

$$t_l \equiv t_{2l} \pmod{n}.$$

Proof. Denote $m_i = j - i$. Then $i, i+1, i+2, \cdots, j-1$ are m succesive integers and exactly one of them (say l) is a multiply of m. Then

$$\begin{gathered} t_i \equiv t_j \pmod{n} \Longleftrightarrow t_i \equiv t_{i+m} \pmod{n} \\ \Longrightarrow t_{i+1} \equiv t_{j+1} \pmod{n} \Longleftrightarrow t_{i+1} \equiv t_{i+1+m} \pmod{n} \\ \Longrightarrow \cdots \\ \Longrightarrow t_k \equiv t_{k+m} \pmod{n}, \qquad \forall k \geq i. \end{gathered}$$

Now applying this congruence to $k = l$, we get

$$t_l \equiv t_{l+m} \equiv t_{l+2m} \equiv \cdots \equiv tl + l = t_{2l} \pmod{n}.$$

□

Now the algorithm for the Pollard-Rho method is as follows:

- Input: $n \in \mathbb{Z}^+$, n is composite.

1. Set $l = 0$, $t_0 = 1$.

2. Compute $t_{l+1} \equiv t_l^2 + 1 \pmod{n}$.
3. Compute $t_{2(l+1)} \equiv (t_{2l}^2 + 1)^2 + 1 \pmod{n}$.
4. Compute $\gcd(t_{2(l+1)} - t_{l+1}, n)$.
    - If the gcd is equal to 1, then increase l by 1 and go back to step 2.
    - If the gcd is larger than 1, then we find some divisor $d > 1$ of n.

Remark 17.2. d may be equal to n. In this case, try trial division or repeat the Pollard-Rho method with different parameters.

Remark 17.3. Expected complexity of the Pollard-Rho method is $O(n^{\frac{1}{4}})$.

17.2 Discrete logarithm problem: Naive approach

No polynomial time algorithms for the Discrete Logarithm Problem are known, which means that the Diffie-Hellman and Elgamal cryptosystem are secure. The fastest algorithm is number field sieving, which can find the discrete logarithm for numbers p which are 160-200 digits long. [Note: We assume that $N = \mathrm{ord}_p(b)$ is known. Computing N may be a hard problem.]

17.3 Baby-step/Giant-step algorithm

The problem now is that we have a prime p, with $a, b \in \{1, 2, \cdots, p-1\}$, where $b^x \equiv a \pmod{p}$. We also know that $N = \mathrm{ord}_p(a)$. Our aim is to find x.

To do this we first let $M = \lceil\sqrt{N}\rceil$. We will write $x = My + z$ with $y, z \in \{0, 1, \cdots, M-1\}$. Then

$$b^x \equiv a \pmod{p} \Longleftrightarrow b^{My+z} \equiv a \pmod{p}$$
$$\Longrightarrow b^z \equiv (b^{-M})^y \cdot a \pmod{p}.$$

Here, will introduce the baby-step/giant-step algorithm:

1. Compute the list of
$$b^0, b^1, b^2, \cdots, b^{M-1} \pmod{p}.$$
2. Compute $b^{-M} \pmod{p}$
3. Compute
$$(b^{-M})^0 \cdot a, (b^{-M})^1 \cdot a, \cdots \pmod{p}$$
until we find a coincidence with the first list.

Then we have

$$(b^{-M})^y \cdot a \equiv b^z \pmod{p} \qquad \text{and} \qquad x \equiv My + Z \pmod{N}.$$

Now the complexity of this algorithm can be calculated by: Algorithm takes $M-1+2+M-1 = 2M$ operations modulo p, which is $O(M) = O(\sqrt{N})$.

17.4 Pohlig-Hellman algorithm

Let $N = q_1^{\alpha_1} \cdot q_2^{\alpha_2} \cdot \cdots \cdot q_r^{\alpha_r}$, where q_i are distinct primes. The idea behind the Pohlig-Hellman algorithm is to compute x modulo $q_1^{\alpha_1}$, $q_2^{\alpha_2}$, $\cdots$, $q_r^{\alpha_r}$ separately, and the use the Chinese Remainder Theorem.

Let $d \,|\, N$,

$$b^x \equiv a \pmod p \Longrightarrow (b^{\frac{N}{d}})^x \equiv a^{\frac{N}{d}} \pmod p.$$

We have $\operatorname{ord}_p(b^{\frac{N}{d}}) = d$. Then we use a naive approach for the baby-step/giant-step algorithm to solve

$$(b^{\frac{N}{d}})^x \equiv a^{\frac{N}{d}} \pmod p$$

with $N' = \operatorname{ord}_p(b^{\frac{N}{d}}) = d$.

Now there is one last trick with the Pohlig-Hellman algorithm. We first let q^k be a divisor of N from the prime factorisation of N. To find $x \pmod{q^k}$, we compute $x \pmod q$, $x \pmod{q^2}$, $\cdots$. Let

$$x = (y_m y_{m-1} \cdots y_1 y_0)_1,$$

which is the expansion of x in base q, which is identical to

$$x = y_m q^m + y_{m-1} q^{m-1} + \cdots + y_1 q + y_0; \qquad y_i \in \{0, 1, \cdots, q-1\}.$$

Notice that $x \pmod{q^i}$ is $y_{i-1}q^{i-1} + \cdots + y_1 q + y_0$. Now the idea is that, we will compute $x_0, x_1, \cdots, x_k$ where

$$x = y_{i-1}q^{i-1} + \cdots + y_1 q + y_0,$$

i.e. $x_i \equiv x \pmod{q^i}$. We will start off with $x_0 = 1$. Given x_i, we can compute x_{i+1}:

$$x_{i+1} = y_i q^i + x_i; \qquad y_i \in \{0, 1, \cdots, q-1\}.$$

Now we have

$$\begin{aligned} b^x &\equiv a \pmod p \\ (b^{\frac{N}{q^{i+1}}})^{x_{i+1}} &\equiv a^{\frac{N}{q^{i+1}}} \pmod p \\ \Longleftrightarrow (b^{\frac{N}{q^{i+1}}})^{y_i q^i + x_i} &\equiv a^{\frac{N}{q^{i+1}}} \pmod p \\ \Longleftrightarrow (b^{\frac{N}{q}})^{y_i} &\equiv a^{\frac{N}{q^{i+1}}} (b^{\frac{N}{q^{i+1}}})^{-x_i} \pmod p, \end{aligned}$$

which can be computed. In this case, we have another discrete logarithm problem for y_i, which is between 0 and $q-1$. This can be found through a naive approach in $O(q)$ steps, or by the baby-step/giant-step algorithm in $O(\sqrt{q})$ steps.

17.5 Complexity of the Pohlig-Hellman algorithm

Consider the prime factorisation of N:

$$N = q_1^{\alpha_1} \cdot q_2^{\alpha_2} \cdot \cdots \cdot q_d^{\alpha_d}.$$

Then solving the discrete logarithm problem requires α_1 DLP's for the order q_1, α_2 DLP's for the order q_2 and so on until α_d DLP's for the order q_d.

18 Lagrange Interpolation Formula in modular arithmetic

This formula is used to share a secret key between n people so that at least k people are needed to reveal the key.

18.1 The formula

In $\mathbb{R}$, consider (x_1, y_1), (x_2, y_2), $\cdots$, (x_{k+1}, y_{k+1}). Then there exists a unique polynomial

$$P(x) = a_k x^k + a_{k-1}x^{k-1} + \cdots + a_1 x + a_0$$

such that

$$\begin{aligned} P(x_1) &= y_1 \\ P(x_2) &= y_2 \\ &\cdots \\ P(x_{k+1}) &= y_{k+1}. \end{aligned}$$

Theorem 18.1. *Let $(x_1, x_2, \cdots, x_{k+1})$ be distinct integers modulo a prime p. Let $(y_1, y_2, \cdots, y_{k+1}$ be another list of integers. Then there exists a unique polynomial*

$$P(x) = a_k x^k + a_{k-1}x^{k-1} + \cdots + a_1 x + a_0$$

with $a_i \in \{0, 1, \cdots, p-1\}$, such that

$$\begin{aligned} P(x_1) &\equiv y_1 \pmod p \\ P(x_2) &\equiv y_2 \pmod p \\ &\cdots \\ P(x_k) &\equiv y_k \pmod p \\ P(x_{k+1}) &\equiv y_{k+1} \pmod p. \end{aligned}$$

Proof. We will firstly prove the uniqueness of this theorem. Consider two polynomials, $f(x)$ and $g(x)$, both satisfying the stated conditions. Then

$$f(x_i) - g(x_i) \equiv 0 \pmod p \qquad \text{for } i \in \{1, \cdots, k+1\}.$$

In other words, $f(x) - g(x) \equiv 0 \pmod p$ has at least $k+1$ solutions. Since the degree of $f(x) - g(x)$ is $\leq k$ and from the theorem (number of roots mod p is less than the degree of the polynomial), we have

$$f(x) - g(x) \equiv 0 \pmod p \iff f(x) \equiv g(x) \pmod p.$$

Now to prove the existence, we will use the Lagrange Interpolation formula,

$$\begin{aligned} P(x) &= \sum_{i=1}^{k+1} y_i \prod_{j \neq i} \frac{x - x_j}{x_i - x_j} \\ &\equiv y_1 \frac{(x-x_2)(x-x_3)\cdots(x-x_{k+1})}{(x_1-x_2)\cdots(x_1-x_{k+1})} + \cdots + y_{k+1}\frac{(x-x_1)(x-x_2)\cdots(x-x_k)}{(x_{k+1}-x_1)\cdots(x_{k+1}-x_k)} \pmod p. \end{aligned}$$

Checking the theorem from here is trivial. □

18.2 Splitting the secret key

Problem: We want to share some secrete key between n people so that $\geq k$ of them are needed to work out the secret key. Our algorithm is given by:

1. Choose a large prime p such that $p > n$.
2. Randomly choose $a_0, a_1, \cdots, a_{k-1} \in \{0, 1, \cdots, p-1\}$.
3. Let $f(x) = a_{k-1}x^{k-1} + \cdots + a_1 x + a_0$. We will provide the person i with the value $f(i) \pmod{p}$ where $1 \leq i \leq n$.

Any k people can combine their information, use the Lagrange Interpolation Formula and hence find $f(x)$. For any less than k people, $f(x)$ can not be found.

19 Square roots in modular arithmetic

19.1 The case where we have an odd prime p

Definition 19.1. Let $a \in \mathbb{Z}$, $a \not\equiv 0 \pmod{p}$. We call a quadratic residue (QR) if the congruence

$$x^2 \equiv a \pmod{p}$$

has solutions. Otherwise, we call a quadratic non-residue (NR). [Note: $a \equiv 0 \pmod{p}$ is neither QR nor NR modulo p.]

To check if a is QR modulo p, we use the help of primitive roots. Let b be a primitive root modulo p. Then

$$a \equiv b^d \pmod{p} \qquad \text{and} \qquad x \equiv b^y \pmod{p},$$

where d can be derived via the DLP and y is unknown. We have

$$\begin{aligned} x^2 \equiv a \pmod{p} &\iff b^{2y} \equiv b^d \pmod{p} \\ &\iff 2y \equiv d \pmod{p-1}. \end{aligned}$$

Now if d is even, then $d = 2d$. We have

$$y \equiv d_1 \pmod{\frac{p-1}{2}} \qquad \text{and} \qquad x \equiv \pm b^{d_1} \pmod{p}$$
$$\implies a \text{ is QR.}$$

If d is odd, then

$$\underbrace{2y}_{\text{even}} \equiv \underbrace{d}_{\text{odd}} \pmod{\underbrace{p-1}_{\text{even}}}$$
$$\implies \text{no solution} \implies a \text{ is NR.}$$

Now we have a problem, this problem requires to find a primitive root modulo p and to solve the DLP. In general, this is very difficult, and almost impossible for large p. The solution to this is in the next proposition.

Proposition 19.2. Let $a \in \mathbb{Z}$, $a \not\equiv 0 \pmod{p}$. Then if a is QR, then

$$a^{\frac{p-1}{2}} \equiv 1 \pmod{p}.$$

a is NR if

$$a^{\frac{p-1}{2}} \equiv -1 \pmod{p}.$$

Proof. Let a be QR modulo p. As we showed before, $a \equiv b^{2d} \pmod{p}$.

$$\Longrightarrow a^{\frac{p-1}{2}} \equiv (b^{2d_1})^{\frac{p-1}{2}} \equiv (b^{d_1})^{p-1} \equiv 1 \pmod{p}.$$

Now let a be NR modulo p. Then

$$a \equiv b^d \pmod{p} \qquad \text{and} \qquad d = 2d_1 + 1.$$

$$a^{\frac{p-1}{2}} \equiv (b^{2d_1+1})^{\frac{p-1}{2}} \equiv b^{\frac{p-1}{2}} \cdot (b^{d_1})^{p-1} \equiv b^{\frac{p-1}{2}} \pmod{p}.$$

The value of $b^{\frac{p-1}{2}}$ is a solution of $x^2 \equiv 1 \pmod{p}$. Hence $b^{\frac{p-1}{2}} = \pm 1 \pmod{p}$. As b is a primitive root, then

$$b^{\frac{p-1}{2}} \equiv -1 \pmod{p}.$$

□

We will now introduce the square root problem, that is, given $a \in \{1, 2, \cdots, p-1\}$ is QR modulo p, solve the congruence

$$x^2 \equiv a \pmod{p}.$$

From our previous notes, we can solve this with the help of primitive roots modulo p and discrete logarithms. But this method takes too long. Now if $p \equiv 3 \pmod{4}$, then we have $\frac{p+1}{4} \in \mathbb{Z}$.

Proposition 19.3. If $p \equiv 3 \pmod{4}$ is prime and a is QR, then the congruence

$$x^2 \equiv a \pmod{p}$$

has solutions $x \equiv \pm a^{\frac{p+1}{4}} \pmod{p}$.

Proof. a is QR modulo p,

$$\Longrightarrow a^{\frac{p-1}{2}} \equiv 1 \pmod{p}$$

$$\Longrightarrow x^2 \equiv a^{\frac{p+1}{2}} \equiv a^{\frac{p-1}{2}} \cdot a \equiv a \pmod{p}.$$

□

Now if we have a general case $p = 2^k m + 1$, $k \in \mathbb{Z}^+$, where m is odd, then we have an algorithm. The algorithm for solving $x^2 \equiv a \pmod{p}$ is as follows:

1. Check if a is QR modulo p by checking $a^{\frac{p-1}{2}} \equiv 1 \pmod{p}$.
2. Find $b \in \{1, 2, \cdots, p-1\}$ so that $\operatorname{ord}_p(b) = 2^k$. The method to do this is: Find a NR modulo p, denoting it as r. We do this by picking random numbers and checking

   $$r^{\frac{p-1}{2}} \equiv -1 \pmod{p}.$$

   We have $\frac{p-1}{2}$ quadratic non-residues in $\{1, 2, \cdots, p-1\}$, meaning that r should be found quickly. Then

   $$b \equiv r^m \pmod{p}.$$

   Now we check

   $$b^{2^k} \equiv (r^m)^{2^k} \equiv r^{p-1} \equiv 1 \pmod{p}$$

   $$b^{2^{k-1}} \equiv (r^m)^{2^{k-1}} \equiv r^{\frac{p-1}{2}} \equiv -1 \pmod{p} \qquad (r \text{ is NR})$$

   $$\Longrightarrow \operatorname{ord}_p(b) \mid 2^k \qquad \text{and} \qquad \operatorname{ord}_p(b) \nmid 2^{k-1}$$

   $$\Longrightarrow \operatorname{ord}_p(b) = 2k.$$

3. We have that $b^0, b^2, \cdots, b^{2^k-2}$ are all roots of 1 of degree 2^{k-1}. On the other hand, a^m is a root of 1 of degree 2^{k-1}. Now we find j which solves

$$b^{2j} \equiv a^m \pmod{p}; \qquad j \in \{0, 1, \cdots, 2^{k-1}-1\}.$$

Since $\operatorname{ord}_p(b^2) = 2^{k-1}$, the Pohlig-Hellman algorithm can solve this quickly.

4. $x \equiv \pm b^j a^{-\left(\frac{m-1}{2}\right)} \pmod{p}$. Now check

$$x^2 \equiv b^{2j} a^{-(m-1)} \equiv a^{m-m+1} \equiv a \pmod{p}.$$

19.2 The case where the modulus m is of the form $m = pq$, where p, q are distinct primes

$x^2 \equiv a \pmod{pq}$ is equivalent to

$$\begin{cases} x^2 \equiv a \pmod{p} \\ x^2 \equiv a \pmod{q}. \end{cases}$$

[Note that the reverse is true by the Chinese Remainder Theorem.]

$x^2 \equiv a \pmod{p}$ has

$$\begin{cases} 2 \text{ solutions if } a \text{ is QR} \\ 1 \text{ solution if } a \equiv 0 \pmod{p} \\ 0 \text{ solutions if } a \text{ is NR.} \end{cases}$$

Therefore, $x^2 \equiv a \pmod{pq}$ has $\begin{Bmatrix} 2 \\ 1 \\ 0 \end{Bmatrix} \times \begin{Bmatrix} 2 \\ 1 \\ 0 \end{Bmatrix}$ solutions. That gives us 4, 2, 1 or 0 solutions mod pq.

19.3 Rabin cryptosystem

In this cryptosystem, everyone can encrypt a message but only Bob can decrypt the message.

$$\text{Alice} \longrightarrow \text{Bob} \longleftarrow \text{Alice}$$

In this algorithm, the steps are:

1. Bob chooses two primes p, q and computes $m = pq$.
2. Bob publishes m as a public key and keeps p and q as a secret.
3. Alice encodes the message as a sequence of residues modulo m

$$[t_1, t_2, \cdots, t_l].$$

4. Alice encrypts the message by replacing

$$t_i \to t_i^2 \equiv S_i \pmod{m}.$$

5. Alice sends the encrypted message $[S_1, S_2, \cdots, S_l]$ to Bob.

6. Bob decrypts the message by solving $t_i^2 \equiv S_i \pmod{m}$ with the help of p and q.

Here Bob will get 4 solutions and the correct solution needs to be guessed. To overcome the problem, we need the help of the next definition and proposition.

Definition 19.4. Let $m = pq$ where p, q are distinct primes. Let $a \in \mathbb{Z}$. We say that a is QR modulo m if it is QR modulo p and QR modulo q.

Proposition 19.5. Define

$$Q_m^* := \{a \in \mathbb{Z} : 1 \leq a \leq m-1 \text{ and } a \text{ is QR modulo } m\}.$$

Let $p, q \equiv 3 \pmod{4}$. The following map

$$\begin{aligned} f : Q_m^* &\to Q_m^* \\ t_i &\to t_i^2 \pmod{m} \end{aligned}$$

is a bijection.

Proof. We know that if a is QR modulo p, then $-a$ is NR modulo p. Hence

$$(-a)^{\frac{p-1}{2}} \equiv (-1)^{\frac{p-1}{2}} \cdot a^{\frac{p-1}{2}} \equiv (-1)^{\frac{p-1}{2}} \equiv -1 \pmod{p}$$
$$\implies -a \text{ is NR modulo } p.$$

Consider $t \in Q_m^*$,

$$\begin{aligned} t &\equiv u \pmod{p} \\ t &\equiv v \pmod{q}, \end{aligned}$$

where u, v are QRs. Then $f(t) \equiv t^2 \pmod{m} \equiv u^2 \pmod{p} \equiv v^2 \pmod{q}$. By solving the congruence

$$x^2 \equiv t^2 \pmod{m},$$

we get 4 solutions

$$\begin{aligned} x &\equiv \pm u \pmod{p} \\ x &\equiv \pm v \pmod{q}. \end{aligned}$$

The only solution $x \in Q_m^*$ is

$$\begin{cases} x \equiv u \pmod{p} \\ x \equiv v \pmod{q} \end{cases}$$

or

$$x \equiv t \pmod{m}.$$

□

It is possible to compute the inverse f^{-1} as follows: If

$$\begin{aligned} s &\equiv u \pmod{p} \\ s &\equiv v \pmod{q} \end{aligned}$$

then

$$\begin{aligned} f^{-1}(s) &\equiv u^{\frac{p+1}{4}} \pmod{p} \\ f^{-1}(s) &\equiv v^{\frac{q+1}{4}} \pmod{q}. \end{aligned}$$