4. Number Theory (Chapter numbers are from the 7th edition of your textbook)

Number theory has various applications in computer science; we will focus on cryptography.

At the end of these lectures, we will be capable of understanding the basics of a cryptography system, namely the “RSA Public Key Cryptosystem”.

RSA – A Public Key Cryptosystem (Rivest, Shamir, Adleman) 76 MIT

Let p, q be large primes (~200 digits) and e be relatively prime to $(p-1)(q-1)$ and $n = pq$.

Encryption: $C = M^e \bmod n$ e: public encryption key

Decryption: $M = C^d \bmod n$ d: private decryption key

Decryption key d is the inverse of e modulo $(p-1)(q-1)$.

RSA – A Public Key Cryptosystem (Rivest, Shamir, Adleman) 76 MIT

Let *p, q* be large primes (~200 digits) and *e* be relatively prime to $(p-1)(q-1)$ and $n = pq$.

Encryption: $C = M^{e} \bmod n$ *e*: public encryption key

Decryption: $M = C^{d} \bmod n$ *d*: private decryption key

Decryption key *d* is the inverse of *e* modulo $(p-1)(q-1)$.

To understand this, we first need to learn about prime numbers, greatest common divisor algorithms, and modular arithmetic.

4.1/4.2/4.3 Divisibility & Prime Numbers

Divisibility: *(Definition)*

Let a and b are integers s.t. $a \neq 0$, we say "a divides b"

if $\exists c$ integer s.t. $b = ac$ (where a is a factor of b).

Notation: $a \nmid b$

$a \quad b$ a does not divide b.

Theorem 1: Let a, b, c be integers.

1. If $a \mid b \wedge a \mid c$ then $a \mid (b + c)$.
2. If $a \mid b$ then $a \mid bc \quad \forall c$.
3. If $a \mid b \wedge b \mid c$, then $a \mid c$.

Theorem 1: Let a, b, c be integers.

1. If $a \mid b \wedge a \mid c$ then $a \mid (b + c)$.
2. If $a \mid b$ then $a \mid bc \quad \forall c$.
3. If $a \mid b \wedge b \mid c$, then $a \mid c$.

Proof:

1. If $a \mid b \wedge a \mid c$ then $\exists k_1, k_2$ integers s.t.

$$b = k_1 a \wedge c = k_2 a \Rightarrow b + c = (k_1 + k_2)a \Rightarrow a \mid (b + c).$$

Theorem 1: Let a, b, c be integers.

1. If $a \mid b \wedge a \mid c$ then $a \mid (b + c)$.
2. If $a \mid b$ then $a \mid bc \quad \forall c$.
3. If $a \mid b \wedge b \mid c$, then $a \mid c$.

Proof:

1. If $a \mid b \wedge a \mid c$ then $\exists k_1, k_2$ integers s.t.
$b = k_1 a \wedge c = k_2 a \Rightarrow b + c = (k_1 + k_2)a \Rightarrow a \mid (b + c)$.

3. $\exists k_1, k_2$ s.t. $b = k_1 a$, $c = k_2 b = k_2 k_1 a \Rightarrow a \mid c$.

2. Prove as an exercise

Prime Number: *(Definition)*

An integer $p > 1$ is called *prime* iff the only positive factors of p are 1 and p.
If p is not prime, then it is *composite*.

Prime Number: *(Definition)*

An integer $p > 1$ is called *prime* iff the only positive factors of p are 1 and p.
If p is not prime, then it is *composite*.

Prime numbers were of interest,

for philosophical reasons (ancient)
for practical reasons (today)
(such as cryptography)

How to find prime numbers? How to devise an efficient algorithm to determine whether a given integer is not? These are important problems in number theory.

Hunt for the largest prime:

The integer $2^p - 1$, where p is prime,
is called **Mersenne Prime** if it is prime.

e.g.

$2^2 - 1 = 3$, $2^3 - 1 = 7$, $2^5 - 1 = 31$ are Mersenne primes
$2^{11} - 1 = 2047 = 23 \times 89$ is not a Mersenne prime

Hunt for the largest prime:

The integer $2^p - 1$, where p is prime,
is called **Mersenne Prime** if it is prime.

e.g.

$2^2 - 1 = 3$, $2^3 - 1 = 7$, $2^5 - 1 = 31$ are Mersenne primes
$2^{11} - 1 = 2047 = 23 \times 89$ is not a Mersenne prime

The largest Mersenne Prime:

(as of 2005)	$2^{25,964,951} - 1$	7,816,230 digits
(as of 2007)	$2^{32,582,657} - 1$	9,808,358 digits
(as of 2010)	$2^{43,112,609} - 1$	12,978,189 digits
(as of 2013)	$2^{57,885,161} - 1$	17,425,170 digits
(as of 2017)	$2^{74,207,281} - 1$	22,338,618 digits
(as of 2018)	$2^{82,589,933} - 1$	24,862,048 digits
(as of 2019)	check it out!	

Theorem: *Fundamental Theorem of Arithmetic*

"Every positive integer, greater than 1, is either prime or can be written uniquely as the product of primes."

$$
\begin{aligned}
100 &= 2\times2\times5\times5\\
999 &= 3\times3\times3\times37\\
7 &= 7
\end{aligned}
$$

We will prove this important theorem later…

Theorem: There are infinitely many primes.

Proof (by contradiction):
Assume otherwise that all the primes are: $p_1, p_2, \ldots, p_n$ (with n finite)

Theorem: There are infinitely many primes.

Proof (by contradiction):
Assume otherwise that all the primes are: $p_1, p_2, \ldots, p_n$ (with n finite)

Let $q = p_1 p_2 \ldots p_n + 1$
By the *Fundamental Theorem of Arithmetic*, q is either prime or can be written as a product of primes.

- If q is prime, it is a new prime number so we have contradiction!

Theorem: There are infinitely many primes.

Proof (by contradiction):
Assume otherwise that all the primes are: $p_1, p_2, \ldots, p_n$ (with n finite)

Let $q = p_1 p_2 \ldots\ p_n + 1$
By the *Fundamental Theorem of Arithmetic*, q is either prime or can be written as a product of primes.

- If q is prime, it is a new prime number so we have contradiction!
- If q is **not** prime, we can write it as a product of primes. But no prime p_i divides q since it would mean $p_i \mid 1$, which is not possible for integers larger than 1. Thus q must have another prime divisor $p \neq p_i \quad \forall \mathrm{i}\ 1 \leq i \leq n$. Contradiction!

Theorem: There are infinitely many primes.

Proof (by contradiction):
Assume otherwise that all the primes are: $p_1, p_2, \ldots, p_n$ (with n finite)

Let $q = p_1 p_2 \ldots\ p_n + 1$
By the *Fundamental Theorem of Arithmetic*, q is either prime or can be written as a product of primes.

- If q is prime, it is a new prime number so we have contradiction!

- If q is **not** prime, we can write it as a product of primes. But no prime p_i divides q since it would mean $p_i \mid 1$, which is not possible for integers larger than 1. Thus q must have another prime divisor $p \neq\ p_i\ \ \forall \text{i}\ 1 \leq i \leq n$. Contradiction!

We have contradiction in both cases. So our initial assumption was false. There are indeed infinitely many prime numbers.

Greatest Common Divisor: *(Definition)*

Let a, b be integers, not both zero.
The largest integer d s.t. $d \mid a \wedge d \mid b$ is called g.c.d of a and b.

$$d = \text{gcd}\,(a, b)$$

Greatest Common Divisor: *(Definition)*

Let a, b be integers, not both zero.
The largest integer d s.t. $d \mid a \wedge d \mid b$ is called g.c.d of a and b.

$$d = \gcd(a, b)$$

Definition: Integers $a_1, a_2, \ldots, a_n$ are **pairwise relatively prime** iff $\gcd(a_i, a_j) = 1 \quad \forall i, j,\ 1 \le i < j \le n$.

How to find gcd of two numbers?

<u>One possible way:</u>

$a = p_1^{a_1} p_2^{a_2} \ldots p_n^{a_n}$, $b = p_1^{b_1} p_2^{b_2} \ldots p_n^{b_n}$,where p_i's are prime.

$$\gcd(a, b) = p_1^{\min(a_1,\, b_1)} p_2^{\min(a_2,\, b_2)} \ldots p_n^{\min(a_n,\, b_n)}$$

How to find gcd of two numbers?

One possible way:

$a = p_1^{a_1} p_2^{a_2} \ldots p_n^{a_n}$, $\quad b = p_1^{b_1} p_2^{b_2} \ldots p_n^{b_n}$, where p_i's are prime.

$\gcd(a, b) = p_1^{\min(a_1, b_1)} p_2^{\min(a_2, b_2)} \ldots p_n^{\min(a_n, b_n)}$

e.g.

gcd(120, 500) = ?

$120 = 2^3 \cdot 3 \cdot 5 \qquad\qquad 500 = 2^2 \cdot 5^3$

$\gcd(120, 500) = 2^2 \cdot 3^0 \cdot 5^1 = 20$

How to find gcd of two numbers?

One possible way:

$a = p_1^{a_1} p_2^{a_2} \ldots p_n^{a_n}$, $\quad b = p_1^{b_1} p_2^{b_2} \ldots p_n^{b_n}$, where p_i's are prime.

$$\gcd(a, b) = p_1^{\min(a_1, b_1)} p_2^{\min(a_2, b_2)} \ldots p_n^{\min(a_n, b_n)}$$

e.g.

gcd(120, 500) = ?

$120 = 2^3 \cdot 3 \cdot 5 \qquad 500 = 2^2 \cdot 5^3$

$\gcd(120, 500) = 2^2 \cdot 3^0 \cdot 5^1 = 20$

But how to find the greatest common divisor of two integers on a computer, especially when the integers are very large?

The Euclidean Algorithm

An efficient method for finding the greatest common divisor of two integers.

Example: Find gcd (287, 91)

$287 = 91 \cdot 3 + 14$

$a \mid 287 \wedge a \mid 91 \quad \Rightarrow a \mid 14$

The Euclidean Algorithm

An efficient method for finding the greatest common divisor of two integers.

Example: Find gcd (287, 91)

$287 = 91 \cdot 3 + 14$

$a \mid 287 \wedge a \mid 91 \quad \Rightarrow a \mid 14$

$\therefore$ gcd (287, 91) = gcd (91, 14)

The Euclidean Algorithm

An efficient method for finding the greatest common divisor of two integers.

Example: Find gcd (287, 91)

$287 = 91 \cdot 3 + 14$

$a \mid 287 \wedge a \mid 91 \quad \Rightarrow a \mid 14$

$\therefore$ gcd (287, 91) = gcd (91, 14)

$91 = 14 \cdot 6 + 7$

$\therefore$ gcd (91, 14) = gcd (14, 7) = 7

Example: Find gcd (287, 91)

$287 = 91 \cdot 3 + 14$

$a \mid 287 \wedge a \mid 91 \quad \Rightarrow a \mid 14$

$\therefore$ gcd (287, 91) = gcd (91, 14)

$91 = 14 \cdot 6 + 7$

$\therefore$ gcd (91, 14) = gcd (14, 7) = 7

Lemma:

Let $a = bq + r$, where a, b, q and r are integers. Then gcd(a, b) = gcd(b, r).

Lemma: Let $a = bq + r$, where a, b, q and r are integers. Then gcd(a, b) = gcd(b, r).

Based on this Lemma, we can develop an algorithm:

Let a, b positive integers s.t. $a \geq b$.
Let $r_0 = a$ and $r_1 = b$.

$$\begin{array}{ll} r_0 = r_1 q_1 + r_2 & 0 \leq r_2 < r_1 \\ r_1 = r_2 q_2 + r_3, & 0 \leq r_3 < r_2 \\ \quad . & \\ \quad . & \\ r_{n-2} = r_{n-1} q_{n-1} + r_n, & 0 \leq r_n < r_{n-1} \\ r_{n-1} = r_n q_n & \end{array}$$

$$a = r_0 > r_1 > r_2 > \cdots \geq 0$$

At most in a steps, remainder will be zero.

$\therefore$ gcd (a,b) = gcd (r_0, r_1) = gcd (r_1, r_2)
= ⋯ = gcd (r_{n-1}, r_n) = gcd (r_n, 0) = r_n

Lemma: Let $a = bq + r$, where a, b, q and r are integers. Then gcd(a, b) = gcd(b, r).

Based on this Lemma, we can develop an algorithm:

Let a, b positive integers s.t. $a \geq b$.
Let $r_0 = a$ and $r_1 = b$.

$$
\begin{array}{ll}
r_0 = r_1\, q_1 + r_2 & 0 \leq r_2 < r_1 \\
r_1 = r_2\, q_2 + r_3, & 0 \leq r_3 < r_2 \\
\quad . & \\
\quad . & \\
r_{n-2} = r_{n-1}\, q_{n-1} + r_n, & 0 \leq r_n < r_{n-1} \\
r_{n-1} = r_n\, q_n &
\end{array}
$$

$$a = r_0 > r_1 > r_2 > \cdots \geq 0$$

At most in a steps, remainder will be zero.

$$
\begin{aligned}
\therefore\ \gcd(a,b) &= \gcd(r_0, r_1) = \gcd(r_1, r_2) \\
&= \cdots = \gcd(r_{n-1}, r_n) = \gcd(r_n, 0) = r_n
\end{aligned}
$$

Example:

$155 = 125 \cdot 1 + 30$
$125 = 30 \cdot 4 + 5$
$30 = 5 \cdot 6$
$\therefore$ gcd (155, 125) = 5

Code:

```
int gcd(int a, int b)
{
    int x,y,r;
    x = a;
    y = b;
    while (y != 0){
        r = x % y;
        x = y;
        y = r;
    }

    return x;
}
```

Example:

$155 = 125 \cdot 1 + 30$

$125 = 30 \cdot 4 + 5$

$30 = 5 \cdot 6$

$\therefore$ gcd $(155, 125) = 5$

Code:

```
int gcd(int a, int b)
{
    int x,y,r;
    x = a;
    y = b;
    while (y != 0){
        r = x % y;
        x = y;
        y = r;
    }

    return x;
}
```

Example:

$155 = 125\cdot 1 + 30$

$125 = 30\cdot 4 + 5$

$30 = 5\cdot 6$

$\therefore \gcd(155, 125) = 5$

Complexity is O(log *b*), but we'll study it later.

Modular Arithmetic

In some situations we do care about the remainder.

e.g.

When will be the next quiz?

Ans: 169 hours from now.

Modular Arithmetic

In some situations we do care about the remainder.

e.g.

When will be the next quiz?

Ans: 169 hours from now.

Definition: Modulo

$$a \bmod m = r \quad \text{iff} \quad a = mq + r \qquad 0 \le r < m$$

Modular Arithmetic

In some situations we do care about the remainder.

e.g.

When will be the next quiz?

Ans: 169 hours from now.

Definition: Modulo

$$a \bmod m = r \quad \text{iff} \quad a = mq + r \qquad 0 \le r < m$$

Definition:

If $a \equiv b \pmod{m}$, then a is *congruent* to b modulo m.

Modular Arithmetic

In some situations we do care about the remainder.

e.g.

When will be the next quiz?
Ans: 169 hours from now.

Definition: *Modulo*

$$a \text{ mod } m = r \quad \text{iff} \quad a = mq + r \qquad 0 \le r < m$$

Definition:

If $a \equiv b \pmod{m}$, then a is *congruent* to b modulo m.

Note that

$$a \equiv b \pmod{m} \quad \leftrightarrow \quad (a \text{ mod } m) = (b \text{ mod } m)$$

$$a \equiv b \pmod{m} \quad \leftrightarrow \quad m \mid a - b$$

Modular Arithmetic

In some situations we do care about the remainder.

e.g.

When will be the next quiz?
Ans: 169 hours from now.

Definition: Modulo

$$a \bmod m = r \quad \text{iff} \quad a = mq + r \qquad 0 \le r < m$$

Definition:

If $a \equiv b \pmod{m}$, then a is *congruent* to b modulo m.

Note that

$$a \equiv b \pmod{m} \quad \leftrightarrow \quad (a \bmod m) = (b \bmod m)$$

$$a \equiv b \pmod{m} \quad \leftrightarrow \quad m \mid a - b$$

Theorem:

$$a \equiv b \pmod{m} \quad \leftrightarrow \quad a = b + km, \qquad k \text{ is some integer}$$

Theorem:

If $a \equiv b \pmod{m}$ and $c \equiv d \pmod{m}$ then

$a + c \equiv b + d \pmod{m}$ and $ac \equiv bd \pmod{m}$.

We will now see a series of theorems and lemmas, that will help us understand the well-known RSA cryptosystem:

***Theorem* 1:**

Let $a, b > 0$, then $\exists s,t$ integers such that $\gcd(a,b) = sa + tb$.

***Theorem* 1:**

Let $a, b > 0$, then $\exists s,t$ integers such that $\gcd(a,b) = sa + tb$.

We won't prove this theorem formally, but the example below shows us that by **reversing** the Euclidean algorithm, we can always find such integers s and t.

e.g.

gcd (22, 6) = 2

$$
\begin{aligned}
22 &= 3\cdot 6 + 4 \\
6 &= 1\cdot 4 + 2 \\
4 &= 2\cdot 2
\end{aligned}
$$

$$
\begin{aligned}
2 &= 6 - 1\cdot 4 \\
&= 6 - (22 - 3\cdot 6) \\
&= 4\cdot 6 - 22
\end{aligned}
$$

***Theorem* 1:**

Let $a,\ b > 0$, then $\exists s,t$ integers such that $\gcd(a,b) = sa + tb$.

We won't prove this theorem formally, but the example below shows us that by **reversing** the Euclidean algorithm, we can always find such integers s and t.

e.g.

gcd (22, 6) = 2

$$\begin{aligned} 22 &= 3\cdot 6 + 4 \\ 6 &= 1\cdot 4 + 2 \\ 4 &= 2\cdot 2 \end{aligned}$$

$$\begin{aligned} 2 &= 6 - 1\cdot 4 \\ &= 6 - (22 - 3\cdot 6) \\ &= 4\cdot 6 - 22 \end{aligned}$$

This theorem has two important consequences, namely *Lemma* 1 and *Lemma* 2:

***Lemma* 1**:

If a, b, $c > 0$ integers s.t. gcd $(a,b) = 1$ and $a \mid bc$,
then $a \mid c$.

***Lemma* 1:**

If $a,\ b,\ c > 0$ integers s.t. gcd $(a,b) = 1$ and $a \mid bc$,
then $a \mid c$.

Proof:

$\exists s,t\ \ \gcd(a,\ b) = 1 = sa + tb$ (by Thm 1 above)

$\Rightarrow sac + tbc = c$ (multiply both sides by c)

$\Rightarrow a \mid tbc$ and $a \mid sac$ (by Thm 1 of Section 4.1 and also since $a \mid bc$)

$\therefore\ a \mid c$

Lemma 1 leads to an important theorem:

***Theorem* 2:**

Let $m > 0$ and a, b, c integers.
If $ac \equiv bc \pmod{m}$ and $\gcd(c, m) = 1$, then $a \equiv b \pmod{m}$.

Lemma 1 leads to an important theorem:

***Theorem* 2:**

Let $m > 0$ and a, b, c integers.
If $ac \equiv bc \pmod{m}$ and $\gcd(c, m) = 1$, then $a \equiv b \pmod{m}$.

Proof:

Since $ac \equiv bc \pmod{m}$, $m \mid ac - bc = c\,(a - b)$
By Lemma 1, $\gcd(c, m) = 1 \Rightarrow m \mid a - b$

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

Lemma 1 leads to an important theorem:

***Theorem* 2:**

Let $m > 0$ and a, b, c integers.
If $ac \equiv bc \pmod{m}$ and $\gcd(c, m) = 1$, then $a \equiv b \pmod{m}$.

Proof:

Since $ac \equiv bc \pmod{m}$, $m \mid ac - bc = c(a - b)$
By Lemma 1, $\gcd(c, m) = 1 \Rightarrow m \mid a - b$

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

Remark: You can **not** simply eliminate equal factors from both sides of the congruence as in usual arithmetic!

***Lemma 2*:**

If p is a prime and $p \mid a_1a_2 \ldots a_n$, where each a_i is integer, then $p \mid a_i$ for some i.

***Lemma 2*:**

If p is a prime and $p \mid a_1a_2 \ldots a_n$, where each a_i is integer, then $p \mid a_i$ for some i.

(Proof can be done based on Lemma 1 using induction, see Exercise 60 in Chapter 4.1 of 6th edition or Exercise 52, Chapter 5.1, 7th edition)

Using *Lemma* 2, we can prove the *Fundamental Theorem of Arithmetic*:

“Every positive integer, greater than 1, is either prime or can be written uniquely as the product of primes,”

Using *Lemma* 2, we can prove the *Fundamental Theorem of Arithmetic*:

“Every positive integer, greater than 1, is either prime or can be written uniquely as the product of primes,” but we’ll prove this only **partly**:

Factorization of an integer into primes is unique.

Using *Lemma* 2, we can prove the *Fundamental Theorem of Arithmetic*:

“Every positive integer, greater than 1, is either prime or can be written uniquely as the product of primes,” but we’ll prove this only **partly**:

Factorization of an integer into primes is unique.

Proof:
Assume two different prime factorizations:
$n = p_1 p_2 \ldots p_s$ and $n = q_1 q_2 \ldots q_t$

Remove all common primes:
$p_{i_1} p_{i_2} \cdots p_{i_u} = q_{j_1} q_{j_2} \cdots q_{j_v}$ $u, v > 0$

Using *Lemma* 2, we can prove the *Fundamental Theorem of Arithmetic*:

"Every positive integer, greater than 1, is either prime or can be written uniquely as the product of primes," but we'll prove this only **partly**:

Factorization of an integer into primes is unique.

Proof:

Assume two different prime factorizations:

$n = p_1 p_2 \ldots p_s$ and $n = q_1 q_2 \ldots q_t$

Remove all common primes:

$p_{i_1} p_{i_2} \cdots p_{i_u} = q_{j_1} q_{j_2} \cdots q_{j_v}$ $u, v > 0$

by *Lemma* 2, p_{i_1} divides q_{j_k} for some k.

Since a prime number can not divide another prime, we have contradiction!

$\therefore$ Factorization is unique

Using *Lemma* 2, we can prove the *Fundamental Theorem of Arithmetic*:

"Every positive integer, greater than 1, is either prime or can be written uniquely as the product of primes," but we'll prove this only **partly**:

Factorization of an integer into primes is unique.

Proof:

Assume two different prime factorizations:

$n = p_1 p_2 \ldots p_s$ and $n = q_1 q_2 \ldots q_t$

Remove all common primes:

$p_{i_1} p_{i_2} \cdots p_{i_u} = q_{j_1} q_{j_2} \cdots q_{j_v}$ $\quad u, v > 0$

by *Lemma* 2, p_{i_1} divides q_{j_k} for some k.

Since a prime number can not divide another prime, we have contradiction!

$\therefore$ Factorization is unique

(Existence of factorization will be proved later).

4.4 Solving Linear Congruences

$ax \equiv b \pmod{m}$ $m > 0$, a, b integers
x is a variable

How to find $\boldsymbol{x}$ that satisfy this congruence?

Definition:

$\bar{a}$: *inverse* of a in modulo m
s.t. $a\bar{a} \equiv 1 \pmod{m}$

Definition:

$\bar{a}$: *inverse* of a in modulo m
s.t. $a\bar{a} \equiv 1 \pmod{m}$

Theorem:

If a and m are relatively prime, and $m > 1$, then $\bar{a}$ exists.
Furthermore it is unique (in modulo m).

Definition:

$\bar{a}$: *inverse* of a in modulo m

s.t. $a\bar{a} \equiv 1 \pmod{m}$

Theorem:

If a and m are relatively prime, and $m > 1$, then $\bar{a}$ exists.
Furthermore it is unique (in modulo m).

Proof: (Existence)

$\gcd(a, m) = 1$

$\exists s,t \quad sa + tm = 1$ (by Thm 1)

$\Rightarrow \quad sa + tm \equiv 1 \pmod{m}$

since $tm \equiv 0 \pmod{m}$

$sa \equiv 1 \pmod{m}$

$\therefore \quad s = \bar{a}$

Definition:

$\bar{a}$: *inverse* of a in modulo m
s.t. $a\bar{a} \equiv 1 \pmod{m}$

Theorem:

If a and m are relatively prime, and $m > 1$, then $\bar{a}$ exists.
Furthermore it is unique (in modulo m).

Proof: (Existence)

$\gcd(a, m) = 1$

$\exists s,t \quad sa + tm = 1$ (by Thm 1)

$\Rightarrow \quad sa + tm \equiv 1 \pmod{m}$

since $tm \equiv 0 \pmod{m}$

$sa \equiv 1 \pmod{m}$

$\therefore \quad s = \bar{a}$

Remark: Inverse does not exist if $\gcd(a, m) \neq 1$.

e.g. Find the inverse of 3 modulo 7.

Solution:

Since gcd (3, 7) = 1, inverse exists.
$7 = 2 \cdot 3 + 1 \Rightarrow -2 \cdot 3 + 1 \cdot 7 = 1$
$\therefore$ –2 is an inverse of 3 mod 7.

Also of 5, –9, 12, so on.

Solution for linear congruence:

$ax \equiv b \pmod{m}$
$\Rightarrow \bar{a}\, a\, x \equiv \bar{a}\, b \pmod{m}$
$\Rightarrow x \equiv \bar{a}\, b \pmod{m}$

Solution for linear congruence:

$ax \equiv b \pmod{m}$

$\Rightarrow \bar{a}\, a\, x \equiv \bar{a}\, b \pmod{m}$

$\Rightarrow x \equiv \bar{a}\, b \pmod{m}$

e.g.

$$3x \equiv 4 \pmod{7}$$

$$\begin{aligned} \therefore\ x &\equiv -2 \cdot 4 \pmod{7} \\ &\equiv -8 \equiv 6 \pmod{7} \end{aligned}$$

e.g. Solve $75x \equiv 5 \pmod{13}$

e.g. Solve $75x \equiv 5 \pmod{13}$

Solution:

Since gcd (75, 13) = 1, inverse exists.

$75 = 13 \cdot 5 + 10 \Rightarrow 13 = 10 \cdot 1 + 3 \Rightarrow 10 = 3 \cdot 3 + 1$ (by Euclidean algorithm)

e.g. Solve $75x \equiv 5 \pmod{13}$

Solution:

Since gcd (75, 13) = 1, inverse exists.

$75 = 13 \cdot 5 + 10 \Rightarrow 13 = 10 \cdot 1 + 3 \Rightarrow 10 = 3 \cdot 3 + 1$ (by Euclidean algorithm)

Reversing the steps of the Euclidean algorithm:

$1 = 10 - 3 \cdot 3 = 10 - 3 \cdot (13 - 10 \cdot 1) = (75 - 13 \cdot 5) - 3 \cdot (13 - 75 + 13 \cdot 5)$

$= 4 \cdot 75 - 23 \cdot 13$

$\therefore$ Inverse of 75 modulo 13 is 4.

e.g. Solve $75x \equiv 5 \pmod{13}$

Solution:

Since gcd (75, 13) = 1, inverse exists.

$75 = 13\cdot 5 + 10 \Rightarrow 13 = 10\cdot 1 + 3 \Rightarrow 10 = 3\cdot 3 + 1$ (by Euclidean algorithm)

Reversing the steps of the Euclidean algorithm:

$1 = 10 - 3\cdot 3 = 10 - 3\cdot(13 - 10\cdot 1) = (75 - 13\cdot 5) - 3\cdot(13 - 75 + 13\cdot 5)$

$= 4\cdot 75 - 23\cdot 13$

$\therefore$ Inverse of 75 modulo 13 is 4.

$$75x \equiv 5 \pmod{13}$$
$$4\cdot 75\, x \equiv 4\cdot 5 \pmod{13}$$
$$x \equiv 20 \equiv 7 \pmod{13}$$

$\therefore$ $x = 7$ is a solution and so are 20,33, 46….

e.g. Solve $75x \equiv 5 \pmod{13}$

Solution:
Since gcd (75, 13) = 1, inverse exists.
$75 = 13{\cdot}5 + 10 \Rightarrow 13 = 10{\cdot}1 + 3 \Rightarrow 10 = 3{\cdot}3 + 1$ (by Euclidean algorithm)

Reversing the steps of the Euclidean algorithm:
$1 = 10 - 3{\cdot}3 = 10 - 3{\cdot}(13 - 10{\cdot}1) = (75 - 13{\cdot}5) - 3{\cdot}(13 - 75 + 13{\cdot}5)$
$\quad = 4{\cdot}75 - 23{\cdot}13$
$\therefore$ Inverse of 75 modulo 13 is 4.

$$\begin{aligned} 75x &\equiv 5 \pmod{13} \\ 4{\cdot}75\,x &\equiv 4{\cdot}5 \pmod{13} \\ x &\equiv 20 \equiv 7 \pmod{13} \end{aligned}$$

$\therefore \quad x = 7$ is a solution and so are 20,33, 46….

Remark: If an inverse exists, the linear congruence has a unique solution in modulo m.
However, if an inverse does not exist, there may still be a solution! (e.g. $2x \equiv 4$ mod 6)

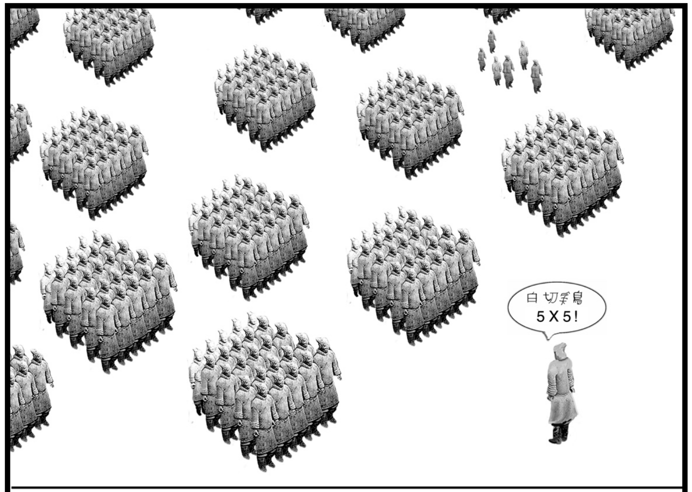
Remark: Inverse of an integer in modulo *m*, if exists, can always be found by reversing the Euclidean algorithm.

In fact there is an efficient algorithm to do this, which is called extended Euclidean algorithm (see Exercise 30, Chapter 4.3, 7th edition).

The Chinese Remainder Problem

The original problem was

How many soldiers are there in Han Xin's army? – If you let them parade in rows of 3 soldiers, two soldiers will be left. If you let them parade in rows of 5, 3 will be left, and in rows of 7, 2 will be left.



$N \equiv 6 \pmod{25}$

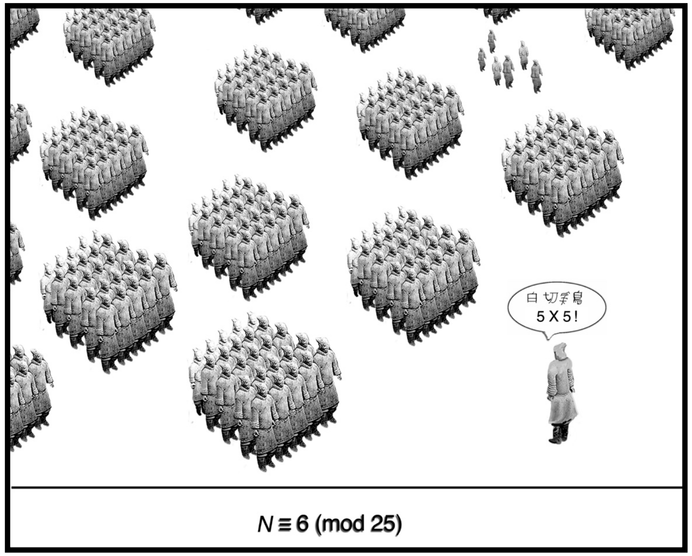
The Chinese Remainder Problem

The original problem was

How many soldiers are there in Han Xin's army? – If you let them parade in rows of 3 soldiers, two soldiers will be left. If you let them parade in rows of 5, 3 will be left, and in rows of 7, 2 will be left.

$x \equiv 2 \pmod 3$
$x \equiv 3 \pmod 5$
$x \equiv 2 \pmod 7$

What is x then?



The Chinese Remainder Theorem

Let m_1, m_2, ..., m_n be pairwise relatively prime positive integers. The system

$$
\begin{aligned}
x &\equiv a_1 \pmod{m_1}\\
x &\equiv a_2 \pmod{m_2}\\
&\ \ \ \ \ .\\
&\ \ \ \ \ .\\
x &\equiv a_n \pmod{m_n}
\end{aligned}
$$

has a **unique** solution in modulo $m = m_1 \cdot m_2 \cdot \ldots\ m_n$.

The Chinese Remainder Theorem

Let m_1, m_2, ..., m_n be pairwise relatively prime positive integers. The system

$$\begin{aligned} x &\equiv a_1 \pmod{m_1} \\ x &\equiv a_2 \pmod{m_2} \\ &\ \ \cdot \\ &\ \ \cdot \\ x &\equiv a_n \pmod{m_n} \end{aligned}$$

has a **unique** solution in modulo $m = m_1 \cdot m_2 \cdot \ldots\ m_n$.

e.g.

Since 3, 5 and 7 are pairwise relatively prime in the previous example, by Chinese Remainder Thm., there is only one solution for x between $0 \leq x < 105$.

OK, but what is this x?

The Chinese Remainder Theorem

Let $m_1, m_2, \ldots, m_n$ be pairwise relatively prime positive integers. The system

$x \equiv a_1 \pmod{m_1}$

$x \equiv a_2 \pmod{m_2}$

.

.

$x \equiv a_n \pmod{m_n}$

has a **unique** solution in modulo $m = m_1 \cdot m_2 \cdot \ldots\ m_n$.

Solution:

Let $M_k = m / m_k$ for $k = 1, 2, \ldots, n$.

Hence gcd $(m_k, M_k) = 1$, and

$\exists y_k$ inverse of M_k s.t. $M_k y_k \equiv 1 \pmod{m_k}$ by the previous theorem.

The Chinese Remainder Theorem

Let m_1, m_2, …, m_n be pairwise relatively prime positive integers. The system

$$\begin{aligned} x &\equiv a_1 \pmod{m_1} \\ x &\equiv a_2 \pmod{m_2} \\ &\;\;. \\ &\;\;. \\ x &\equiv a_n \pmod{m_n} \end{aligned}$$

has a **unique** solution in modulo $m = m_1 \cdot m_2 \cdot \ldots\, m_n$.

Solution:

Let $M_k = m / m_k$ for $k = 1, 2, \ldots, n$.

Hence gcd $(m_k, M_k) = 1$, and

$\exists y_k$ inverse of M_k s.t. $M_k\, y_k \equiv 1 \pmod{m_k}$ by the previous theorem.

A solution can then be given as:

$x = a_1 M_1 y_1 + a_2 M_2 y_2 + \ldots + a_n M_n y_n$ Why?

The Chinese Remainder Theorem

Let $m_1, m_2, \ldots, m_n$ be pairwise relatively prime positive integers. The system

$$
\begin{aligned}
x &\equiv a_1 \pmod{m_1}\\
x &\equiv a_2 \pmod{m_2}\\
&\;\;\vdots\\
x &\equiv a_n \pmod{m_n}
\end{aligned}
$$

has a **unique** solution in modulo $m = m_1 \cdot m_2 \cdot \ldots\, m_n$.

Solution:

Let $M_k = m / m_k$ for $k = 1, 2, \ldots, n$.

Hence gcd $(m_k, M_k) = 1$, and

$\exists y_k$ inverse of M_k s.t. $M_k\, y_k \equiv 1 \pmod{m_k}$ by the previous theorem.

A solution can then be given as:

$x = a_1 M_1 y_1 + a_2 M_2 y_2 + \ldots + a_n M_n y_n$ Why?

Since $M_i \equiv 0 \pmod{m_k}$ whenever $i \neq k$ and $M_k\, y_k \equiv 1 \pmod{m_k}$,

$x \equiv a_k M_k y_k + 0 + \ldots + 0 \equiv a_k \pmod{m_k}$.

The Chinese Remainder Theorem

Let $m_1, m_2, \ldots, m_n$ be pairwise relatively prime positive integers. The system

$x \equiv a_1 \pmod{m_1}$

$x \equiv a_2 \pmod{m_2}$

.

.

$x \equiv a_n \pmod{m_n}$

has a **unique** solution in modulo $m = m_1 \cdot m_2 \cdot \ldots\ m_n$.

Solution:

Let $M_k = m / m_k$ for $k = 1, 2, \ldots, n$.

Hence gcd $(m_k, M_k) = 1$, and

$\exists y_k$ inverse of M_k s.t. $M_k y_k \equiv 1 \pmod{m_k}$ by the previous theorem.

A solution can then be given as:

$x = a_1 M_1 y_1 + a_2 M_2 y_2 + \ldots + a_n M_n y_n$ Why?

Since $M_i \equiv 0 \pmod{m_k}$ whenever $i \neq k$ and $M_k y_k \equiv 1 \pmod{m_k}$,

$x \equiv a_k M_k y_k + 0 + \ldots + 0 \equiv a_k \pmod{m_k}$.

Remark: Uniqueness can be proved using *proof by contradiction*.

Example:

$$x \equiv 2 \pmod{3}$$
$$x \equiv 3 \pmod{5}$$
$$x \equiv 2 \pmod{14} \qquad\qquad x = ?$$

$m = 3 \cdot 5 \cdot 14 = 210$

Example:

$x \equiv 2 \pmod{3}$
$x \equiv 3 \pmod{5}$
$x \equiv 2 \pmod{14}$ $\qquad x = ?$

$m = 3 \cdot 5 \cdot 14 = 210$

Note that 3, 5 and 14 are pairwise relatively prime, so we can apply Chinese Rem. Th:

$M_1 = m/3 = 70$, $M_2 = 42$, $M_3 = 15$

$M_1 = 70 \equiv 1 \pmod{3} \qquad y_1 = 1$
$M_2 = 42 \equiv 2 \pmod{5} \qquad y_2 = 3$
$M_3 = 15 \equiv 1 \pmod{14} \qquad y_3 = 1$

$x = a_1 M_1 y_1 + a_2 M_2 y_2 + a_3 M_3 y_3 = 2 \cdot 70 \cdot 1 + 3 \cdot 42 \cdot 3 + 2 \cdot 15 \cdot 1 = 548 \equiv 128 \pmod{210}$.

Note that 128 is the only solution in mod 210, that means there is no other number between 0 and 209, which satisfies the above congruences.

Pseudoprimes

Is there an efficient way to determine whether an integer is prime or not?

Ancient Chinese believed that

$$n \text{ is prime} \leftrightarrow 2^{n-1} \equiv 1 \pmod{n}$$

Pseudoprimes

Is there an efficient way to determine whether an integer is prime or not?

Ancient Chinese believed that

$$n \text{ is prime} \leftrightarrow 2^{n-1} \equiv 1 \pmod{n}$$

Reasons:

i) they observed this holds when n is prime

ii) they couldn't find a composite number for which the congruence holds

Pseudoprimes

Is there an efficient way to determine whether an integer is prime or not?

Ancient Chinese believed that

$$n \text{ is prime} \leftrightarrow 2^{n\text{-}1} \equiv 1 \pmod{n}$$

Reasons:

i) they observed this holds when n is prime

ii) they couldn't find a composite number for which the congruence holds

However they were wrong: e.g., $2^{340} \equiv 1 \pmod{341}$ and $341=11\cdot31$

Hence 341 is a *pseudoprime*!

Pseudoprimes

Is there an efficient way to determine whether an integer is prime or not?

Ancient Chinese believed that

$$n \text{ is prime} \leftrightarrow 2^{n-1} \equiv 1 \pmod{n}$$

Reasons:

i) they observed this holds when n is prime

ii) they couldn't find a composite number for which the congruence holds

However they were wrong: e.g., $2^{340} \equiv 1 \pmod{341}$ and $341=11\cdot31$

Hence 341 is a *pseudoprime*!

But how to compute $2^{340} \pmod{341}$? 2^{340} is too large!

How to compute 2^{340} (mod 341) ? 2^{340} is too large!

One way of doing it: Compute successively 2 mod 341, 2^2 mod 341, 2^3 mod 341, …,
2^{340} mod 341

Note that all computations above are in modulo 341, hence numbers never exceed 340.

2 mod 341 = 2, Compute 2^2 mod 341 = 4, 2^3 mod 341 = 8, 2^4 mod 341 = 16,
2^5 mod 341 = 32,…, 2^8 mod 341 = 256,
$2^9 \equiv 2{\cdot}256 \equiv 512 \equiv 171 \pmod{341} \Rightarrow 2^9$ mod 341 = 171
$2^{10} \equiv 2{\cdot}171 \equiv 342 \equiv 1 \pmod{341} \Rightarrow 2^{10}$ mod 341 = 1, … and so on

How to compute 2^{340} (mod 341) ? 2^{340} is too large!

<u>One way of doing it</u>: Compute successively 2 mod 341, 2^2 mod 341, 2^3 mod 341, …, 2^{340} mod 341

Note that all computations above are in modulo 341, hence numbers never exceed 340.

2 mod 341 = 2, Compute 2^2 mod 341 = 4, 2^3 mod 341 = 8, 2^4 mod 341 = 16,
2^5 mod 341 = 32,…, 2^8 mod 341 = 256,
$2^9 \equiv 2\cdot256 \equiv 512 \equiv 171 \pmod{341} \Rightarrow 2^9$ mod 341 = 171
$2^{10} \equiv 2\cdot171 \equiv 342 \equiv 1 \pmod{341} \Rightarrow 2^{10}$ mod 341 = 1, … and so on

<u>Another (better) way to use (in general)</u>: Compute 2 mod 341, 2^2 mod 341, 2^4 mod 341, 2^8 mod 341, …, 2^{340} mod 341

Yet a better way to compute b^a mod m is to write the prime factorization of m and then to use *Chinese Remainder Theorem* and ***Fermat's Little Theorem*** (if possible):

Fermat's Little Theorem:

If p is prime and a is an integer not divisible by p, then

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

Furthermore for every integer a, $a^p \equiv a \pmod{p}$

Remark: For proof, see Exercise 19 in Chapter 4.4 of your textbook, 7th edition.

To compute 2^{340} (mod 341) ,

$341 = 11 \cdot 31$ (prime factorization)

(i) $2^{10} \equiv 1 \pmod{11}$ by Fermat's Little Theorem
$2^{340} = (2^{10})^{34} \equiv 1 \pmod{11}$

(ii) $2^{30} \equiv 1 \pmod{31}$ by Fermat's Little Theorem
$2^{330} = (2^{30})^{11} \equiv 1 \pmod{31}$
$2^{10} = 2^5 2^5 \equiv 1 \pmod{31}$ since $2^5 = 32 \equiv 1 \pmod{31}$
Hence $2^{340} = 2^{330} 2^{10} \equiv 1 \pmod{31}$

To compute 2^{340} (mod 341) ,

$341 = 11 \cdot 31$ (prime factorization)

(i) $2^{10} \equiv 1 \pmod{11}$ by Fermat's Little Theorem
$2^{340} = (2^{10})^{34} \equiv 1 \pmod{11}$

(ii) $2^{30} \equiv 1 \pmod{31}$ by Fermat's Little Theorem
$2^{330} = (2^{30})^{11} \equiv 1 \pmod{31}$
$2^{10} = 2^5\, 2^5 \equiv 1 \pmod{31}$ since $2^5 = 32 \equiv 1 \pmod{31}$
Hence $2^{340} = 2^{330} 2^{10} \equiv 1 \pmod{31}$

Then by Chinese Remainder Thm, **(i)** $\wedge$ **(ii)** $\rightarrow 2^{340} \equiv 1 \pmod{341}$

To compute 2^{340} (mod 341) ,

$341 = 11 \cdot 31$ (prime factorization)

(i) $2^{10} \equiv 1 \pmod{11}$ by Fermat's Little Theorem
$2^{340} = (2^{10})^{34} \equiv 1 \pmod{11}$

(ii) $2^{30} \equiv 1 \pmod{31}$ by Fermat's Little Theorem
$2^{330} = (2^{30})^{11} \equiv 1 \pmod{31}$
$2^{10} = 2^5 2^5 \equiv 1 \pmod{31}$ since $2^5 = 32 \equiv 1 \pmod{31}$
Hence $2^{340} = 2^{330} 2^{10} \equiv 1 \pmod{31}$

Then by Chinese Remainder Thm, **(i)** $\wedge$ **(ii)** $\rightarrow 2^{340} \equiv 1 \pmod{341}$

(If you have difficulty to understand this last statement, see Exercise 21, Chapter 4.4, 7th edition, or Exercise 23, Ch. 3.7, 6th edition)
You can reason in this way: The integer $x = 2^{340}$ is congruent to 1 in mod 11 and 31. Chinese Remainder Thm states that there exists only one such integer in modulo 341 (that is between 0 and 340), and 1 already satisfies these congruences. So $x = 2^{340} \equiv 1$ (mod 341).

4.6 Cryptography

Earliest known cryptology was used by J. Caesar:
$h(p) = (p + k) \bmod 26$, where p is an integer code for alphabet letters, and k is the key.

YES $\Rightarrow$ AGU (for $k = 2$)

A: 0
B: 1
…

Z: 25

$h(p) = (p + 2) \bmod 26$, where p is an integer code for alphabet letters
not very high level of security!

A better alternative: $h(p) = (ap + b) \bmod 26$
choose a and b s.t. $h(p)$ is 1-to-1.

$h(p) = (p + 2) \bmod 26$, where p is an integer code for alphabet letters
not very high level of security!

A better alternative: $h(p) = (ap + b) \bmod 26$
choose a and b s.t. $h(p)$ is 1-to-1.

Still not a secure encryption scheme:
Generally broken using frequency analysis: Given an encrypted sentence, guess that the most commonly used letter represents “E” since it is the most common letter used in English. Then continue…

$h(p) = (p + 2) \bmod 26$, where p is an integer code for alphabet letters
not very high level of security!

A better alternative: $h(p) = (ap + b) \bmod 26$
choose a and b s.t. $h(p)$ is 1-to-1.

Still not a secure encryption scheme:
Generally broken using frequency analysis: Given an encrypted sentence, guess that the most commonly used letter represents “E” since it is the most common letter used in English. Then continue…

There exist of course much better recent cryptography methods. Next we’ll learn one of them, which is RSA: A “Public Key” Cryptosystem.

First let’s see what “public key” means…

Private **key** vs ***Public*** **key**

Private key cryptology:

e.g.

Encryption: $C = (M + k) \pmod{26}$ M: original message code

Decryption: $M = (C - k) \pmod{26}$ C: encrypted message code

k: private key (used for both encryption and decryption)

Private **key** vs ***Public*** **key**

<u>Private key cryptology</u>:

e.g.

Encryption: $C = (M + k) \pmod{26}$ M: original message code

Decryption: $M = (C - k) \pmod{26}$ C: encrypted message code

k: private key (used for both encryption and decryption)

Everybody knows the encryption method but nobody, supposedly, can get the original message without knowing the private key k.

Problem: How to share the secret key between two parties?

***Private* key vs *Public* key**

Private key cryptology:

e.g.

Encryption: $C = (M + k) \pmod{26}$ M: original message code

Decryption: $M = (C - k) \pmod{26}$ C: encrypted message code

k: private key (used for both encryption and decryption)

Everybody knows the encryption method but nobody, supposedly, can get the original message without knowing the private key k.
Problem: How to share the secret key between two parties?

Public key cryptology:
Encryption and decryption keys are different!

Everybody knows the encryption method and the public encryption key, but nobody can get the original message without knowing the private decryption key.

RSA – A Public Key Cryptosystem (Rivest, Shamir, Adleman) 76 MIT

Let p, q be large primes (~200 digits) and e be relatively prime to $(p-1)(q-1)$ and $n = pq$.

Encryption: $C = (M^e \bmod n)$ e: public encryption key

Decryption: $M = (C^d \bmod n)$ d: private decryption key

Decryption key d is the inverse of e modulo $(p-1)(q-1)$.

Inverse exists since $\gcd(e, (p-1)(q-1)) = 1$.

RSA – A Public Key Cryptosystem (Rivest, Shamir, Adleman) 76 MIT

Let p, q be large primes (~200 digits) and e be relatively prime to $(p-1)(q-1)$ and $n = pq$.

Encryption: $C = (M^e \bmod n)$ e: public encryption key

Decryption: $M = (C^d \bmod n)$ d: private decryption key

Decryption key d is the inverse of e modulo $(p-1)(q-1)$.

Inverse exists since $\gcd(e, (p-1)(q-1)) = 1$.

Remark: Almost impossible to find d although one knows e and $n = pq$ since these are very large numbers. No polynomial time algorithm exists for prime factorization.

e.g.

$p = 43, q = 59 \qquad n = 43 \cdot 59 = 2537 \qquad e = 13$

$\Rightarrow \quad \gcd(13, 58 \cdot 42) = 1$

Let $M = \underset{M_1}{1819} \quad \underset{M_2}{1415}$ (STOP)

$C_1 \equiv 1819^{13} \pmod{2537} = 2081$

$C_2 \equiv 1415^{13} \pmod{2537} = 2182$

$d = 937$ (inverse of 13 modulo $42 \cdot 58$)

$M_1 = C_1^{937} \pmod{2537} = 1819$

$M_2 = C_2^{937} \pmod{2537} = 1415$

Use Examples of RSA Cryptosystem

Encryption

Suppose Alice wants to send a message M to Bob. Alice creates the ciphertext C by exponentiating s.t. $C = M^{e} \bmod n$, where e and n are Bob's public keys. She sends C to Bob. To decrypt, Bob also exponentiates but with d s.t. $M = C^{d} \bmod n$; the relationship between e and d ensures that Bob correctly recovers M. Since only Bob knows d, only Bob can decrypt this message.

Use Examples of RSA Cryptosystem

Encryption

Suppose Alice wants to send a message *M* to Bob. Alice creates the ciphertext *C* by exponentiating s.t. $C = M^{e}$ mod *n*, where *e* and *n* are Bob's public keys. She sends *C* to Bob. To decrypt, Bob also exponentiates but with *d* s.t. $M = C^{d}$ mod *n*; the relationship between *e* and *d* ensures that Bob correctly recovers *M*. Since only Bob knows *d*, only Bob can decrypt this message.

Digital Signature

Suppose Alice wants to send a message *M* to Bob in such a way that Bob is assured the message is both authentic, has not been tampered with, and from Alice. Alice creates a digital signature *S* by exponentiating s.t. $S = M^{e}$ mod *n*, where *e* is Alice's private key. She sends *M* and *S* to Bob. To verify the signature, Bob exponentiates and checks whether the message *M* is recovered by $M = S^{d}$ mod *n*, where *d* and *n* are Alice’s public keys.

Use Examples of RSA Cryptosystem

Encryption

Suppose Alice wants to send a message M to Bob. Alice creates the ciphertext C by exponentiating s.t. $C = M^e \bmod n$, where e and n are Bob's public keys. She sends C to Bob. To decrypt, Bob also exponentiates but with d s.t. $M = C^d \bmod n$; the relationship between e and d ensures that Bob correctly recovers M. Since only Bob knows d, only Bob can decrypt this message.

Digital Signature

Suppose Alice wants to send a message M to Bob in such a way that Bob is assured the message is both authentic, has not been tampered with, and from Alice. Alice creates a digital signature S by exponentiating s.t. $S = M^e \bmod n$, where e is Alice's private key. She sends M and S to Bob. To verify the signature, Bob exponentiates and checks whether the message M is recovered by $M = S^d \bmod n$, where d and n are Alice's public keys.

Thus encryption and authentication here take place without any sharing of private keys: each person uses only another's public key or their own private key. Anyone can send an encrypted message or verify a signed message, but only someone in possession of the correct private key can decrypt or sign a message.

RSA – A Public Key Cryptosystem

Let p, q be large primes (~200 digits) and e be relatively prime to $(p-1)(q-1)$ and $n = pq$.

Encryption: $C = (M^e \bmod n)$ e: public encryption key

Decryption: $M = (C^d \bmod n)$ d: private decryption key

Decryption key d is the inverse of e modulo $(p-1)(q-1)$.

But why is C^d equal to M in modulo $n = pq$?

We can show this by using what we have learned so far…..

Why is C^d equal to M in modulo $n = pq$?

By **Fermat's Little Thm,**

$M^{p-1} \equiv 1 \pmod{p}$ assuming p and q does not divide M (p, q are very large)

$M^{q-1} \equiv 1 \pmod{q}$

Why is C^{d} equal to M in modulo $n = pq$?

By **Fermat's Little Thm,**

$M^{p-1} \equiv 1 \pmod{p}$ assuming p and q does not divide M (p, q are very large)
$M^{q-1} \equiv 1 \pmod{q}$

and $C^{d} \equiv (M^{e})^{d} = M^{ed} = M^{1 + k(p-1)(q-1)} \pmod{n}$ since $ed \equiv 1 \bmod (p-1)(q-1)$.

Why is C^d equal to M in modulo $n = pq$?

By **Fermat's Little Thm,**

$M^{p-1} \equiv 1 \pmod{p}$ assuming p and q does not divide M (p, q are very large)
$M^{q-1} \equiv 1 \pmod{q}$

and $C^d \equiv (M^e)^d = M^{ed} = M^{1 + k(p-1)(q-1)} \pmod{n}$ since $ed \equiv 1 \bmod (p-1)(q-1)$.

Hence

$C^d \equiv M\,(M^{p-1})^{k(q-1)} \equiv M \cdot 1 \equiv M \pmod{p}$
$C^d \equiv M\,(M^{q-1})^{k(p-1)} \equiv M \cdot 1 \equiv M \pmod{q}$

Why is C^{d} equal to M in modulo $n = pq$?

By **Fermat's Little Thm,**

$M^{p-1} \equiv 1 \pmod{p}$ assuming p and q does not divide M (p, q are very large)
$M^{q-1} \equiv 1 \pmod{q}$

and $C^{d} \equiv (M^{e})^{d} = M^{ed} = M^{1 + k(p-1)(q-1)} \pmod{n}$ since $ed \equiv 1 \bmod (p-1)(q-1)$.

Hence

$C^{d} \equiv M(M^{p-1})^{k(q-1)} \equiv M \cdot 1 \equiv M \pmod{p}$
$C^{d} \equiv M(M^{q-1})^{k(p-1)} \equiv M \cdot 1 \equiv M \pmod{q}$

Since gcd $(p, q) = 1$, by **Chinese Remainder Thm**

$C^{d} \equiv M \pmod{pq}$