Number Theory

Coding Club

Mohammed Alshamsi
2021004826
mo.alshamsi@aurak.ac.ae

American University of Ras Al Khaimah

March 14, 2024

Introduction

- Number Theory?

Introduction

- **Number Theory?** Study of integers, especially positive integers

Introduction

- **Number Theory?** Study of integers, especially positive integers
- **Format?**

Introduction

- **Number Theory?** Study of integers, especially positive integers
- **Format?** You'll see the following:

Introduction

- **Number Theory?** Study of integers, especially positive integers
- **Format?** You'll see the following:
  - Mathematical definitions with examples

Introduction

- **Number Theory?** Study of integers, especially positive integers
- **Format?** You'll see the following:
  - Mathematical definitions with examples
  - Some interesting algorithms

Introduction

- **Number Theory?** Study of integers, especially positive integers
- **Format?** You'll see the following:
  - Mathematical definitions with examples
  - Some interesting algorithms
  - Applications to cryptography

Introduction

- **Number Theory?** Study of integers, especially positive integers
- **Format?** You'll see the following:
  - Mathematical definitions with examples
  - Some interesting algorithms
  - Applications to cryptography
- **Why Math?**

Introduction

- **Number Theory?** Study of integers, especially positive integers
- **Format?** You'll see the following:
  - Mathematical definitions with examples
  - Some interesting algorithms
  - Applications to cryptography
- **Why Math?** Some math background will enable you to make more sophisticated software.

Introduction

- **Number Theory?** Study of integers, especially positive integers
- **Format?** You'll see the following:
  - Mathematical definitions with examples
  - Some interesting algorithms
  - Applications to cryptography
- **Why Math?** Some math background will enable you to make more sophisticated software.
- **These Slides?**

Introduction

- **Number Theory?** Study of integers, especially positive integers
- **Format?** You'll see the following:
  - Mathematical definitions with examples
  - Some interesting algorithms
  - Applications to cryptography
- **Why Math?** Some math background will enable you to make more sophisticated software.
- **These Slides?** See our GitHub repository.
- **If you don't know coding:**

Introduction

- **Number Theory?** Study of integers, especially positive integers
- **Format?** You'll see the following:
  - Mathematical definitions with examples
  - Some interesting algorithms
  - Applications to cryptography
- **Why Math?** Some math background will enable you to make more sophisticated software.
- **These Slides?** See our GitHub repository.
- **If you don't know coding:** Check appendix in main document on GitHub, or find a guide online. (We won't do much in this presentation.)

Introduction

- **Number Theory?** Study of integers, especially positive integers
- **Format?** You'll see the following:
  - Mathematical definitions with examples
  - Some interesting algorithms
  - Applications to cryptography
- **Why Math?** Some math background will enable you to make more sophisticated software.
- **These Slides?** See our GitHub repository.
- **If you don't know coding:** Check appendix in main document on GitHub, or find a guide online. (We won't do much in this presentation.)

Any questions?

Outline

1. Divisibility
   - Division Algorithm
   - Caesar Cipher
   - GCD
   - Prime Numbers
2. Modular Arithmetic
   - Affine Cipher
   - Chinese Remainder Theorem
   - RSA

Divisibility

Definition (Divisibility)

Divisibility

Definition (Divisibility)

A nonzero integer a is a divisor of an integer b if $b = ak$ for some integer k.

Divisibility

Definition (Divisibility)

A nonzero integer a is a divisor of an integer b if $b = ak$ for some integer k.

- When a divides b, we write "$a \mid b$".

Divisibility

Definition (Divisibility)

A nonzero integer a is a divisor of an integer b if $b = ak$ for some integer k.

- When a divides b, we write "$a \mid b$".
- When a does not divide b, we write "$a \nmid b$".

Divisibility

Definition (Divisibility)

A nonzero integer a is a divisor of an integer b if $b = ak$ for some integer k.

- When a divides b, we write "$a \mid b$".
- When a does not divide b, we write "$a \nmid b$".

Example

Divisibility

Definition (Divisibility)

A nonzero integer a is a divisor of an integer b if $b = ak$ for some integer k.

- When a divides b, we write "$a \mid b$".
- When a does not divide b, we write "$a \nmid b$".

Example

- $5 \mid 15$ because $15 = 5 \cdot 3$, and 3 is an integer.

Divisibility

Definition (Divisibility)

A nonzero integer a is a divisor of an integer b if $b = ak$ for some integer k.

- When a divides b, we write "$a \mid b$".
- When a does not divide b, we write "$a \nmid b$".

Example

- $5 \mid 15$ because $15 = 5 \cdot 3$, and 3 is an integer.
- $6 \nmid 15$ because $15 = 6 \cdot 2.5$, and 2.5 is not an integer.

Divisibility

Definition (Divisibility)

A nonzero integer a is a divisor of an integer b if $b = ak$ for some integer k.

- When a divides b, we write "$a \mid b$".
- When a does not divide b, we write "$a \nmid b$".

Example

- $5 \mid 15$ because $15 = 5 \cdot 3$, and 3 is an integer.
- $6 \nmid 15$ because $15 = 6 \cdot 2.5$, and 2.5 is not an integer.
- For all n, $n \mid 0$. (Why?)

Division Algorithm

Theorem (The Division Algorithm)

Division Algorithm

Theorem (The Division Algorithm)

For integers a and m with $m > 0$, there exist unique integers q and r such that

$$a = mq + r,$$

where $0 \leqslant r < m$.

Division Algorithm

Theorem (The Division Algorithm)

For integers a and m with $m > 0$, there exist unique integers q and r such that

$$a = mq + r,$$

where $0 \leqslant r < m$. We may write $a \bmod m$ to refer to this unique r.

Division Algorithm

Theorem (The Division Algorithm)

For integers a and m with $m > 0$, there exist unique integers q and r such that

$$a = mq + r,$$

where $0 \leqslant r < m$. We may write $a \bmod m$ to refer to this unique r.

Example

Division Algorithm

Theorem (The Division Algorithm)

For integers a and m with $m > 0$, there exist unique integers q and r such that

$$a = mq + r,$$

where $0 \leqslant r < m$. We may write $a \bmod m$ to refer to this unique r.

Example

- If $a = 17$ and $m = 5$, $17 = 5 \cdot 3 + 2$. Note that $0 \leqslant 2 < 5$.

Division Algorithm

Theorem (The Division Algorithm)

For integers a and m with $m > 0$, there exist unique integers q and r such that

$$a = mq + r,$$

where $0 \leqslant r < m$. We may write $a \bmod m$ to refer to this unique r.

Example

- If $a = 17$ and $m = 5$, $17 = 5 \cdot 3 + 2$. Note that $0 \leqslant 2 < 5$.
- If $a = -17$ and $m = 5$, $-17 = 5 \cdot -4 + 3$. Also, $-17 \bmod 5 = 3$.

Division Algorithm (Cont.)

In the C programming language, % gives the remainder.

Division Algorithm (Cont.)

In the C programming language, % gives the remainder.

```
int a = 17, m = 5;
int r = a % m;
printf("%d",r);
```

Division Algorithm (Cont.)

In the C programming language, % gives the remainder.

```
int a = 17, m = 5;
int r = a % m;
printf("%d",r);
```

This outputs 2.

Division Algorithm (Cont.)

In the C programming language, % gives the remainder.

```
int a = 17, m = 5;
int r = a % m;
printf("%d",r);
```

This outputs 2.

Note! This isn't the same as a mod m. See this example:

Division Algorithm (Cont.)

In the C programming language, % gives the remainder.

```
int a = 17, m = 5;
int r = a % m;
printf("%d",r);
```

This outputs 2.

Note! This isn't the same as $a \bmod m$. See this example:

```
int a = -17, m = 5;
int r = a % m;
printf("%d",r);
```

Division Algorithm (Cont.)

In the C programming language, % gives the remainder.

```
int a = 17, m = 5;
int r = a % m;
printf("%d",r);
```

This outputs 2.

Note! This isn't the same as $a \bmod m$. See this example:

```
int a = -17, m = 5;
int r = a % m;
printf("%d",r);
```

This prints -2, instead of $3 = -17 \bmod 5$.

Caesar Cipher

- **Encryption:**

Caesar Cipher

- **Encryption:** Transforming a plain text message into cipher text to hide its content.

Caesar Cipher

- **Encryption:** Transforming a plain text message into cipher text to hide its content.
- **Decryption:**

Caesar Cipher

- **Encryption:** Transforming a plain text message into cipher text to hide its content.
- **Decryption:** Reverting the cipher text to plain text.

Caesar Cipher

- **Encryption:** Transforming a plain text message into cipher text to hide its content.
- **Decryption:** Reverting the cipher text to plain text.
- **Key:**

Caesar Cipher

- **Encryption:** Transforming a plain text message into cipher text to hide its content.
- **Decryption:** Reverting the cipher text to plain text.
- **Key:** Determines “parameters” for the encryption and decryption.

Caesar Cipher

- **Encryption:** Transforming a plain text message into cipher text to hide its content.
- **Decryption:** Reverting the cipher text to plain text.
- **Key:** Determines "parameters" for the encryption and decryption.
  - Usually agreed upon by sender and receiver.

Caesar Cipher

- **Encryption:** Transforming a plain text message into cipher text to hide its content.
- **Decryption:** Reverting the cipher text to plain text.
- **Key:** Determines "parameters" for the encryption and decryption.
  - Usually agreed upon by sender and receiver.
- **Caesar Cipher:**

Caesar Cipher

- **Encryption:** Transforming a plain text message into cipher text to hide its content.
- **Decryption:** Reverting the cipher text to plain text.
- **Key:** Determines "parameters" for the encryption and decryption.
  - Usually agreed upon by sender and receiver.
- **Caesar Cipher:** "Shift" alphabet by the key number.

Caesar Cipher (Cont.)

Example

Caesar Cipher (Cont.)

Example

Alphabet shifted by key $k = 3$.

Caesar Cipher (Cont.)

Example

Alphabet shifted by key $k = 3$.

$$\begin{array}{cccccccccc} A & B & C & D & E & \cdots & W & X & Y & Z \\ D & E & F & G & H & \cdots & Z & A & B & C \end{array}$$

Caesar Cipher (Cont.)

Example

Alphabet shifted by key $k = 3$.

$$\begin{array}{cccccccccc} A & B & C & D & E & \cdots & W & X & Y & Z \\ D & E & F & G & H & \cdots & Z & A & B & C \end{array}$$

Message:

I HAVE INVENTED A NEW SALAD, TELL THE GREEKS.

Caesar Cipher (Cont.)

Example

Alphabet shifted by key $k = 3$.

$$\begin{array}{cccccccccc} A & B & C & D & E & \cdots & W & X & Y & Z \\ D & E & F & G & H & \cdots & Z & A & B & C \end{array}$$

Message:

I HAVE INVENTED A NEW SALAD, TELL THE GREEKS.

Replace each letter with its correspondent:

L KDYH LQYHQWHG D QHZ VDODG, WHOO WKH JUHHNV.

GCD

Definition (Greatest Common Divisor)

GCD

Definition (Greatest Common Divisor)

Let a, b, c be integers. If $c \mid a$ and $c \mid b$, then c is a *common divisor* of a and b.

GCD

Definition (Greatest Common Divisor)

Let a, b, c be integers. If $c \mid a$ and $c \mid b$, then c is a *common divisor* of a and b. The largest such c is the greatest common divisor of a and b, and is denoted $\gcd(a, b)$.

GCD

Definition (Greatest Common Divisor)

Let a, b, c be integers. If $c \mid a$ and $c \mid b$, then c is a *common divisor* of a and b. The largest such c is the greatest common divisor of a and b, and is denoted $\gcd(a, b)$.

Theorem (Bézout's Identity)

GCD

Definition (Greatest Common Divisor)

Let a, b, c be integers. If $c \mid a$ and $c \mid b$, then c is a *common divisor* of a and b. The largest such c is the greatest common divisor of a and b, and is denoted $\gcd(a, b)$.

Theorem (Bézout's Identity)

Let a, b, d be integers with $d = \gcd(a, b)$.

GCD

Definition (Greatest Common Divisor)

Let a, b, c be integers. If $c \mid a$ and $c \mid b$, then c is a *common divisor* of a and b. The largest such c is the greatest common divisor of a and b, and is denoted $\gcd(a, b)$.

Theorem (Bézout's Identity)

Let a, b, d be integers with $d = \gcd(a, b)$. For each multiple of d, there exists a pair of integers x, y such that $ax + by$ is equal to this multiple.

The Euclidean Algorithm

Algorithm (Euclidean Algorithm)

The Euclidean Algorithm

Algorithm (Euclidean Algorithm)

Given two integers m and n, find $\gcd(m, n)$.

The Euclidean Algorithm

Algorithm (Euclidean Algorithm)

Given two integers m and n, find $\gcd(m, n)$.

1. [Find remainder.] Divide m by n and let r be the remainder.

The Euclidean Algorithm

Algorithm (Euclidean Algorithm)

Given two integers m and n, find $\gcd(m, n)$.

1. [Find remainder.] Divide m by n and let r be the remainder.
2. [Is it zero?] If r is 0, the algorithm terminates; n is the answer.

The Euclidean Algorithm

Algorithm (Euclidean Algorithm)

Given two integers m and n, find $\gcd(m, n)$.

1. [Find remainder.] Divide m by n and let r be the remainder.
2. [Is it zero?] If r is 0, the algorithm terminates; n is the answer.
3. [Reduce.] Set m to n, then n to r, and go back to Step 1.

Example: Euclidean Algorithm

1. [Find remainder.] Divide m by n and let r be the remainder.
2. [Is it zero?] If r is 0, the algorithm terminates; n is the answer.
3. [Reduce.] Set m to n, then n to r, and go back to Step 1.

Prime Numbers

Definition (Prime Number)

Prime Numbers

Definition (Prime Number)

A prime number p is a positive integer that has no divisors apart from 1 and p.

Prime Numbers

Definition (Prime Number)

A prime number p is a positive integer that has no divisors apart from 1 and p.

$$2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, \ldots$$

The Sieve of Eratosthenes

Algorithm (Sieve of Eratosthenes)

The Sieve of Eratosthenes

Algorithm (Sieve of Eratosthenes)

Generate a list of all prime numbers less than or equal to a positive integer n.

The Sieve of Eratosthenes

Algorithm (Sieve of Eratosthenes)

Generate a list of all prime numbers less than or equal to a positive integer n.

1. [Initialize.] Create a list of consecutive integers from 2 to n. Let $p = 2$.

The Sieve of Eratosthenes

Algorithm (Sieve of Eratosthenes)

Generate a list of all prime numbers less than or equal to a positive integer n.

1. [Initialize.] Create a list of consecutive integers from 2 to n. Let $p = 2$.
2. [Remove composites.] Remove all multiples of p from the list, except p itself.

The Sieve of Eratosthenes

Algorithm (Sieve of Eratosthenes)

Generate a list of all prime numbers less than or equal to a positive integer n.

1. [Initialize.] Create a list of consecutive integers from 2 to n. Let $p = 2$.
2. [Remove composites.] Remove all multiples of p from the list, except p itself.
3. [Iterate.] If there is an integer greater than p in the list, set p to be the smallest such integer, and go to Step 2. Otherwise, terminate; all numbers in the list are prime.

Example: Sieve of Eratosthenes

Initialize

	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50
51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70
71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	89	90
91	92	93	94	95	96	97	98	99	100

Example: Sieve of Eratosthenes (Cont.)

$p = 2$

	2	3		5		7		9	
11		13		15		17		19	
21		23		25		27		29	
31		33		35		37		39	
41		43		45		47		49	
51		53		55		57		59	
61		63		65		67		69	
71		73		75		77		79	
81		83		85		87		89	
91		93		95		97		99	

Example: Sieve of Eratosthenes (Cont.)

$p = 3$

	2	3		5		7			
11		13				17		19	
		23		25				29	
31				35		37			
41		43				47		49	
		53		55				59	
61				65		67			
71		73				77		79	
		83		85				89	
91				95		97			

Example: Sieve of Eratosthenes (Cont.)

$p = 5$

	2	3		5		7			
11		13				17		19	
		23						29	
31						37			
41		43				47		49	
		53						59	
61						67			
71		73				77		79	
		83						89	
91						97			

Example: Sieve of Eratosthenes (Cont.)

$p = 7$

	2	3		5		7			
11		13				17		19	
		23						29	
31						37			
41		43				47			
		53						59	
61						67			
71		73						79	
		83						89	
						97			

Optimization: We can stop if $p > \sqrt{n}$.

Theorems About Primes

Theorem (Euclid's Lemma)

Theorems About Primes

Theorem (Euclid's Lemma)

If a prime number p divides the product ab of two integers a and b, then p must divide at least one of a or b.

Theorems About Primes

Theorem (Euclid's Lemma)

If a prime number p divides the product ab of two integers a and b, then p must divide at least one of a or b.

Theorem (Fundamental Theorem of Arithmetic)

Theorems About Primes

Theorem (Euclid's Lemma)

If a prime number p divides the product ab of two integers a and b, then p must divide at least one of a or b.

Theorem (Fundamental Theorem of Arithmetic)

Every integer greater than 1 can be represented uniquely as a product of prime powers.

Relatively Prime Numbers

Definition (Relatively Prime Numbers)

Relatively Prime Numbers

Definition (Relatively Prime Numbers)

Let a and b be integers. If $\gcd(a, b) = 1$, then a and b are said to be relatively prime.

Relatively Prime Numbers

Definition (Relatively Prime Numbers)

Let a and b be integers. If $\gcd(a, b) = 1$, then a and b are said to be relatively prime.

Definition (Euler's Totient Function)

Relatively Prime Numbers

Definition (Relatively Prime Numbers)

Let a and b be integers. If $\gcd(a, b) = 1$, then a and b are said to be relatively prime.

Definition (Euler's Totient Function)

Let n be an integer. $\phi(n)$ counts how many of the positive integers up to n are relatively prime to n.

Relatively Prime Numbers

Definition (Relatively Prime Numbers)

Let a and b be integers. If $\gcd(a, b) = 1$, then a and b are said to be relatively prime.

Definition (Euler's Totient Function)

Let n be an integer. $\phi(n)$ counts how many of the positive integers up to n are relatively prime to n.

Proposition

Relatively Prime Numbers

Definition (Relatively Prime Numbers)

Let a and b be integers. If $\gcd(a, b) = 1$, then a and b are said to be relatively prime.

Definition (Euler's Totient Function)

Let n be an integer. $\phi(n)$ counts how many of the positive integers up to n are relatively prime to n.

Proposition

- Whenever n is prime, $\phi(n) = n - 1$.

Relatively Prime Numbers

Definition (Relatively Prime Numbers)

Let a and b be integers. If $\gcd(a, b) = 1$, then a and b are said to be relatively prime.

Definition (Euler's Totient Function)

Let n be an integer. $\phi(n)$ counts how many of the positive integers up to n are relatively prime to n.

Proposition

- Whenever n is prime, $\phi(n) = n - 1$.
- For any two relatively prime numbers m and n, $\phi(mn) = \phi(m)\phi(n)$.

Recap

- We defined divisibility and went over the division algorithm.

Recap

- We defined divisibility and went over the division algorithm.
- Caesar Cipher.

Recap

- We defined divisibility and went over the division algorithm.
- Caesar Cipher.
- Greatest Common Divisor, Bézout's Identity, and the Euclidean Algorithm.

Recap

- We defined divisibility and went over the division algorithm.
- Caesar Cipher.
- Greatest Common Divisor, Bézout's Identity, and the Euclidean Algorithm.
- Prime numbers and the Sieve of Eratosthenes.

Modular Arithmetic

Definition (Congruence Modulo m)

Definition (Congruence Modulo m)

For integers a, b, m, if $m \mid (a - b)$, then we say that a is congruent to b modulo m, and write $a \equiv b \pmod{m}$.

Modular Arithmetic

Definition (Congruence Modulo m)

For integers a, b, m, if $m \mid (a - b)$, then we say that a is congruent to b modulo m, and write $a \equiv b \pmod{m}$.

Example

Modular Arithmetic

Definition (Congruence Modulo m)

For integers a, b, m, if $m \mid (a - b)$, then we say that a is congruent to b modulo m, and write $a \equiv b \pmod{m}$.

Example

- $9 \equiv 21 \pmod{6}$ because $6 \mid (21 - 9)$.

Modular Arithmetic

Definition (Congruence Modulo m)

For integers a, b, m, if $m \mid (a - b)$, then we say that a is congruent to b modulo m, and write $a \equiv b \pmod{m}$.

Example

- $9 \equiv 21 \pmod{6}$ because $6 \mid (21 - 9)$.
  According to the Division Algorithm, $21 = 6 \cdot 3 + 3$ and $9 = 6 \cdot 1 + 3$.
  Remainders are the same!

Modular Arithmetic

Definition (Congruence Modulo m)

For integers a, b, m, if $m \mid (a - b)$, then we say that a is congruent to b modulo m, and write $a \equiv b \pmod{m}$.

Example

- $9 \equiv 21 \pmod{6}$ because $6 \mid (21 - 9)$.
  According to the Division Algorithm, $21 = 6 \cdot 3 + 3$ and $9 = 6 \cdot 1 + 3$.
  Remainders are the same!
- $-17 \equiv 4 \pmod{7}$ because $7 \mid (4 - (-17))$.

Properties

Proposition (Modular Arithmetic)

Properties

Proposition (Modular Arithmetic)

Suppose that $a \equiv b \pmod{m}$. Then, the following is true for all integers k.

Properties

Proposition (Modular Arithmetic)

Suppose that $a \equiv b \pmod{m}$. Then, the following is true for all integers k.

- $a + k \equiv b + k \pmod{m}$.

Properties

Proposition (Modular Arithmetic)

Suppose that $a \equiv b \pmod{m}$. Then, the following is true for all integers k.

- $a + k \equiv b + k \pmod{m}$.
- If $c \equiv d \pmod{m}$, then $ac \equiv bd \pmod{m}$.

Properties

Proposition (Modular Arithmetic)

Suppose that $a \equiv b \pmod{m}$. Then, the following is true for all integers k.

- $a + k \equiv b + k \pmod{m}$.
- If $c \equiv d \pmod{m}$, then $ac \equiv bd \pmod{m}$.
- $a^k \equiv b^k \pmod{m}$.

Properties

Proposition (Modular Arithmetic)

Suppose that $a \equiv b \pmod{m}$. Then, the following is true for all integers k.

- $a + k \equiv b + k \pmod{m}$.
- If $c \equiv d \pmod{m}$, then $ac \equiv bd \pmod{m}$.
- $a^k \equiv b^k \pmod{m}$.

Definition (Modular Multiplicative Inverse)

Properties

Proposition (Modular Arithmetic)

Suppose that $a \equiv b \pmod{m}$. Then, the following is true for all integers k.

- $a + k \equiv b + k \pmod{m}$.
- If $c \equiv d \pmod{m}$, then $ac \equiv bd \pmod{m}$.
- $a^k \equiv b^k \pmod{m}$.

Definition (Modular Multiplicative Inverse)

Given relatively prime integers a, m, there exists an integer a^{-1} such that $a^{-1}a \equiv 1 \pmod{m}$.

Properties

Proposition (Modular Arithmetic)

Suppose that $a \equiv b \pmod{m}$. Then, the following is true for all integers k.

- $a + k \equiv b + k \pmod{m}$.
- If $c \equiv d \pmod{m}$, then $ac \equiv bd \pmod{m}$.
- $a^k \equiv b^k \pmod{m}$.

Definition (Modular Multiplicative Inverse)

Given relatively prime integers a, m, there exists an integer a^{-1} such that $a^{-1}a \equiv 1 \pmod{m}$. We call a^{-1} the modular multiplicative inverse of a.

Affine Cipher

- Generalization of the Caesar Cipher.

Affine Cipher

- Generalization of the Caesar Cipher.
- First multiply modulo 26, then shift (add modulo 26).

Affine Cipher

- Generalization of the Caesar Cipher.
- First multiply modulo 26, then shift (add modulo 26).

Algorithm (Affine Cipher Encryption)

Affine Cipher

- Generalization of the Caesar Cipher.
- First multiply modulo 26, then shift (add modulo 26).

Algorithm (Affine Cipher Encryption)

1. [Choose key.] Choose an integer $0 < a < 26$ relatively prime to 26, and any integer $0 \leqslant b < 26$.

Affine Cipher

- Generalization of the Caesar Cipher.
- First multiply modulo 26, then shift (add modulo 26).

Algorithm (Affine Cipher Encryption)

1. [Choose key.] Choose an integer $0 < a < 26$ relatively prime to 26, and any integer $0 \leqslant b < 26$.
2. [Encrypt.] For each letter, take its numerical value x. Find the integer $0 \leqslant y < 26$ such that $y \equiv ax + b \pmod{26}$. Replace by the letter corresponding to y.

Affine Cipher Decryption

Assume $b = 0$. We know $y \equiv ax \pmod{26}$.

Affine Cipher Decryption

Assume $b = 0$. We know $y \equiv ax \pmod{26}$.
Since $\gcd(a, 26) = 1$, there must be a^{-1} such that

$$a^{-1}ax \equiv x \equiv a^{-1}y \pmod{26}.$$

Affine Cipher Decryption

Assume $b = 0$. We know $y \equiv ax \pmod{26}$.
Since $\gcd(a, 26) = 1$, there must be a^{-1} such that

$$a^{-1}ax \equiv x \equiv a^{-1}y \pmod{26}.$$

Pairs:

a	1	3	5	7	9	11	15	17	19	21	23	25
a^{-1}	1	9	21	15	3	19	7	23	11	5	17	25

Affine Cipher Decryption

Assume $b = 0$. We know $y \equiv ax \pmod{26}$.
Since $\gcd(a, 26) = 1$, there must be a^{-1} such that

$$a^{-1}ax \equiv x \equiv a^{-1}y \pmod{26}.$$

Pairs:

a	1	3	5	7	9	11	15	17	19	21	23	25
a^{-1}	1	9	21	15	3	19	7	23	11	5	17	25

What if $b \neq 0$, so that $ax + b \equiv y \pmod{26}$?

Affine Cipher Decryption

Assume $b = 0$. We know $y \equiv ax \pmod{26}$.
Since $\gcd(a, 26) = 1$, there must be a^{-1} such that

$$a^{-1}ax \equiv x \equiv a^{-1}y \pmod{26}.$$

Pairs:

a	1	3	5	7	9	11	15	17	19	21	23	25
a^{-1}	1	9	21	15	3	19	7	23	11	5	17	25

What if $b \neq 0$, so that $ax + b \equiv y \pmod{26}$? Then $ax \equiv y - b \pmod{26}$, so

$$x \equiv a^{-1}y - a^{-1}b \pmod{26}.$$

Example: Affine Cipher Encryption

0	1	2	3	4	5	6	7	8	9	10	11	12
A	B	C	D	E	F	G	H	I	J	K	L	M
13	14	15	16	17	18	19	20	21	22	23	24	25
N	O	P	Q	R	S	T	U	V	W	X	Y	Z

AFFINE NOT LINEAR

Choose any a in $\{1, 3, 5, 7, 9, 11, 15, 17, 19, 21, 23, 25\}$, and any $0 \leqslant b < 26$.

Residues

- Suppose you have a set of moduli $m_1, m_2, \ldots, m_k$, and an integer x.

Residues

- Suppose you have a set of moduli $m_1, m_2, \ldots, m_k$, and an integer x.
- "Residues" $u_1 = x$ mod m_1, $u_2 = x$ mod m_2, ...

Residues

- Suppose you have a set of moduli $m_1, m_2, \ldots, m_k$, and an integer x.
- "Residues" $u_1 = x \bmod m_1$, $u_2 = x \bmod m_2$, ...
- Modular representation of x in this system is

$$(u_1, u_2, \ldots, u_k).$$

Residues

- Suppose you have a set of moduli $m_1, m_2, \ldots, m_k$, and an integer x.
- "Residues" $u_1 = x \bmod m_1$, $u_2 = x \bmod m_2$, ...
- Modular representation of x in this system is

$$(u_1, u_2, \ldots, u_k).$$

Example

Residues

- Suppose you have a set of moduli $m_1, m_2, \ldots, m_k$, and an integer x.
- "Residues" $u_1 = x \bmod m_1$, $u_2 = x \bmod m_2$, ...
- Modular representation of x in this system is

$$(u_1, u_2, \ldots, u_k).$$

Example

Three moduli $m_1 = 8, m_2 = 21, m_3 = 5$.

Residues

- Suppose you have a set of moduli $m_1, m_2, \ldots, m_k$, and an integer x.
- "Residues" $u_1 = x \bmod m_1$, $u_2 = x \bmod m_2$, ...
- Modular representation of x in this system is

$$(u_1, u_2, \ldots, u_k).$$

Example

Three moduli $m_1 = 8, m_2 = 21, m_3 = 5$. Let's choose $x = 127$.

Residues

- Suppose you have a set of moduli $m_1, m_2, \ldots, m_k$, and an integer x.
- "Residues" $u_1 = x \bmod m_1$, $u_2 = x \bmod m_2$, ...
- Modular representation of x in this system is

$$(u_1, u_2, \ldots, u_k).$$

Example

Three moduli $m_1 = 8, m_2 = 21, m_3 = 5$. Let's choose $x = 127$. Then $u_1 = 7, u_2 = 1, u_3 = 2$.

Residues

- Suppose you have a set of moduli $m_1, m_2, \ldots, m_k$, and an integer x.
- "Residues" $u_1 = x \bmod m_1$, $u_2 = x \bmod m_2$, ...
- Modular representation of x in this system is

$$(u_1, u_2, \ldots, u_k).$$

Example

Three moduli $m_1 = 8, m_2 = 21, m_3 = 5$. Let's choose $x = 127$. Then $u_1 = 7, u_2 = 1, u_3 = 2$. So x can be represented as $(7, 1, 2)$.

Chinese Remainder Theorem

In above example, between 1 and $m_1 m_2 m_3 = 840$ inclusive, 127 is the only number with representation $(7, 1, 2)$!

Chinese Remainder Theorem

In above example, between 1 and $m_1 m_2 m_3 = 840$ inclusive, 127 is the only number with representation $(7, 1, 2)$!

Theorem (Chinese Remainder Theorem)

Chinese Remainder Theorem

In above example, between 1 and $m_1 m_2 m_3 = 840$ inclusive, 127 is the only number with representation $(7, 1, 2)$!

Theorem (Chinese Remainder Theorem)

Let $m_1, m_2, \ldots, m_k$ be positive integers that are relatively prime in pairs.

Chinese Remainder Theorem

In above example, between 1 and $m_1 m_2 m_3 = 840$ inclusive, 127 is the only number with representation $(7, 1, 2)$!

Theorem (Chinese Remainder Theorem)

Let $m_1, m_2, \ldots, m_k$ be positive integers that are relatively prime in pairs. Let $m = m_1 m_2 \cdots m_k$, and let $a, u_1, u_2, \ldots, u_k$ be integers.

Chinese Remainder Theorem

In above example, between 1 and $m_1 m_2 m_3 = 840$ inclusive, 127 is the only number with representation $(7, 1, 2)$!

Theorem (Chinese Remainder Theorem)

Let $m_1, m_2, \ldots, m_k$ be positive integers that are relatively prime in pairs. Let $m = m_1 m_2 \cdots m_k$, and let $a, u_1, u_2, \ldots, u_k$ be integers. Then there is exactly one x such that

$$a \leqslant x < a + m, \quad \text{and} \quad x \equiv u_i \pmod{m_i} \quad \text{for } 1 \leqslant i \leqslant k.$$

Chinese Remainder Theorem

In above example, between 1 and $m_1 m_2 m_3 = 840$ inclusive, 127 is the only number with representation $(7, 1, 2)$!

Theorem (Chinese Remainder Theorem)

Let $m_1, m_2, \ldots, m_k$ be positive integers that are relatively prime in pairs. Let $m = m_1 m_2 \cdots m_k$, and let $a, u_1, u_2, \ldots, u_k$ be integers. Then there is exactly one x such that

$$a \leqslant x < a + m, \quad \text{and} \quad x \equiv u_i \pmod{m_i} \quad \text{for } 1 \leqslant i \leqslant k.$$

a allows for an offset. We took $a = 1$ above, but could choose any value.

RSA

- **Asymmetric encryption** (two keys)

RSA

- **Asymmetric encryption** (two keys)
  - **Public key** shared with anyone, used for encryption

RSA

- **Asymmetric encryption** (two keys)
  - **Public key** shared with anyone, used for encryption
  - **Private key** known only to receiver, used for decryption

RSA

- **Asymmetric encryption** (two keys)
  - **Public key** shared with anyone, used for encryption
  - **Private key** known only to receiver, used for decryption
- RSA's security relies on difficulty of factorizing large primes.

RSA

Algorithm (RSA Encryption)

RSA

Algorithm (RSA Encryption)

1. [Choose key.] Choose two primes p and q, and an integer e such that $(p-1)(q-1)$ and e are relatively prime.

RSA

Algorithm (RSA Encryption)

1. [Choose key.] Choose two primes p and q, and an integer e such that $(p-1)(q-1)$ and e are relatively prime.
2. [Encrypt.] For each letter, take its numerical value x, and replace it with the letter corresponding to $y = (x^e \bmod pq)$.

RSA

Algorithm (RSA Encryption)

1. [Choose key.] Choose two primes p and q, and an integer e such that $(p-1)(q-1)$ and e are relatively prime.
2. [Encrypt.] For each letter, take its numerical value x, and replace it with the letter corresponding to $y = (x^e \bmod pq)$.

Decryption: Find integer d for which $ed \equiv 1 \pmod{(p-1)(q-1)}$. Then take $x = y^d \bmod pq$.

Algorithm (RSA Encryption)

1. [Choose key.] Choose two primes p and q, and an integer e such that $(p-1)(q-1)$ and e are relatively prime.
2. [Encrypt.] For each letter, take its numerical value x, and replace it with the letter corresponding to $y = (x^e \bmod pq)$.

Decryption: Find integer d for which $ed \equiv 1 \pmod{(p-1)(q-1)}$. Then take $x = y^d \bmod pq$. Yes, that's it.

RSA

Algorithm (RSA Encryption)

1. [Choose key.] Choose two primes p and q, and an integer e such that $(p-1)(q-1)$ and e are relatively prime.
2. [Encrypt.] For each letter, take its numerical value x, and replace it with the letter corresponding to $y = (x^e \bmod pq)$.

Decryption: Find integer d for which $ed \equiv 1 \pmod{(p-1)(q-1)}$. Then take $x = y^d \bmod pq$. Yes, that's it.

Example: RSA

65	66	67	68	69	70	71	72	73	74	75	76	77
A	B	C	D	E	F	G	H	I	J	K	L	M
78	79	80	81	82	83	84	85	86	87	88	89	90
N	O	P	Q	R	S	T	U	V	W	X	Y	Z

KEEP ON KEEPING ON

Euler's Theorem

To explain RSA, we'll need this theorem.

Euler's Theorem

To explain RSA, we'll need this theorem.

Theorem (Euler's Theorem)

Euler's Theorem

To explain RSA, we'll need this theorem.

Theorem (Euler's Theorem)

For integers a and n, if they are relatively prime, then

Euler's Theorem

To explain RSA, we'll need this theorem.

Theorem (Euler's Theorem)

For integers a and n, if they are relatively prime, then

$$a^{\phi(n)} \equiv 1 \pmod{n}, \quad \text{or equivalently } a^{\phi(n)+1} \equiv a \pmod{n}.$$

Euler's Theorem

To explain RSA, we'll need this theorem.

Theorem (Euler's Theorem)

For integers a and n, if they are relatively prime, then

$$a^{\phi(n)} \equiv 1 \pmod{n}, \quad \text{or equivalently } a^{\phi(n)+1} \equiv a \pmod{n}.$$

So $x^{\phi(pq)} \equiv 1 \pmod{pq}$, which implies that $x^{k\phi(pq)+1} \equiv x \pmod{pq}$.

Correctness of RSA Decryption

Given $ed \equiv 1 \pmod{(p-1)(q-1)}$.

Proof.

Correctness of RSA Decryption

Given $ed \equiv 1 \pmod{(p-1)(q-1)}$.

Proof.

We know that $x^{p-1} \equiv 1 \pmod{p}$ and $x^{q-1} \equiv 1 \pmod{q}$.

Correctness of RSA Decryption

Given $ed \equiv 1 \pmod{(p-1)(q-1)}$.

Proof.

We know that $x^{p-1} \equiv 1 \pmod{p}$ and $x^{q-1} \equiv 1 \pmod{q}$.
So $x^{k(p-1)(q-1)+1} \equiv x \pmod{p}$ and $x^{k(p-1)(q-1)+1} \equiv x \pmod{q}$.

Correctness of RSA Decryption

Given $ed \equiv 1 \pmod{(p-1)(q-1)}$.

Proof.

We know that $x^{p-1} \equiv 1 \pmod{p}$ and $x^{q-1} \equiv 1 \pmod{q}$.
So $x^{k(p-1)(q-1)+1} \equiv x \pmod{p}$ and $x^{k(p-1)(q-1)+1} \equiv x \pmod{q}$.
Since $ed \equiv 1 \pmod{\phi(pq)}$, there is k such that $ed = k\phi(pq) + 1$. That is, $ed = k(p-1)(q-1) + 1$.

Correctness of RSA Decryption

Given $ed \equiv 1 \pmod{(p-1)(q-1)}$.

Proof.

We know that $x^{p-1} \equiv 1 \pmod{p}$ and $x^{q-1} \equiv 1 \pmod{q}$.
So $x^{k(p-1)(q-1)+1} \equiv x \pmod{p}$ and $x^{k(p-1)(q-1)+1} \equiv x \pmod{q}$.
Since $ed \equiv 1 \pmod{\phi(pq)}$, there is k such that $ed = k\phi(pq) + 1$. That is, $ed = k(p-1)(q-1) + 1$.
Substitute:

$$x^{ed} \equiv x \pmod{p} \quad \text{and} \quad x^{ed} \equiv x \pmod{q}.$$

Correctness of RSA Decryption

Given $ed \equiv 1 \pmod{(p-1)(q-1)}$.

Proof.

We know that $x^{p-1} \equiv 1 \pmod{p}$ and $x^{q-1} \equiv 1 \pmod{q}$.
So $x^{k(p-1)(q-1)+1} \equiv x \pmod{p}$ and $x^{k(p-1)(q-1)+1} \equiv x \pmod{q}$.
Since $ed \equiv 1 \pmod{\phi(pq)}$, there is k such that $ed = k\phi(pq) + 1$. That is, $ed = k(p-1)(q-1) + 1$.
Substitute:

$$x^{ed} \equiv x \pmod{p} \quad \text{and} \quad x^{ed} \equiv x \pmod{q}.$$

So $x^{ed} \equiv x \pmod{pq}$. ■

That's All!

Most of this was based on the following:

- The Art of Computer Programming (Knuth) — Chapter 4, sections 4.3.2 and 4.5.4
- Concrete Mathematics (Graham, Knuth, Patashnik) — Chapter 4
- Number Theory (Andrews) — Chapters 1 through 4
- Proofs: A Long-Form Mathematics Textbook (Cummings) — Chapter 2
- Handbook of Applied Cryptography (Menezes, Oorschot, Vanstone) — Section 8.2