

# Computational Number Theory

Chanathip Namprempre

Department of Computer Science  
Reed College

# Agenda

1. Important sets of numbers, Euler Phi function, Greatest Common Divisor
2. Groups, exponentiation
3. Group order
4. Running time of computation on large numbers
5. Subgroups
6. Order of a group element
7. Subgroup generated by an element, subgroup order
8. Generators
9. Discrete Logs
10. Cyclic groups
11. DL and friends: DL, CDH, DDH
12. Elliptic curve groups
13. Diffie-Hellman key exchange protocol

# Important sets of numbers and Euler Phi function

$$\mathbf{Z} = \{\dots, -2, -1, 0, 1, 2, \dots\}$$

$$\mathbf{N} = \{0, 1, 2, \dots\}$$

$$\mathbf{Z}^+ = \{1, 2, \dots\}$$

$$\mathbf{Z}_N = \{0, \dots, N-1\}$$

$$\mathbf{Z}_N^* = \{a \in \mathbf{Z}_N : \gcd(a, N) = 1\}$$

$$\phi(N) = |\mathbf{Z}_N^*|$$

- Try  $\gcd(12, 20), \gcd(21, 16)$
- Try  $\mathbf{Z}_{15}, \mathbf{Z}_{15}^*, \phi(15)$

## Quotient Remainder Theorem

For any  $a, N \in \mathbf{Z}$  with  $N > 0$ , there exists unique  $q, r \in \mathbf{N}$  such that

- ▶  $a = Nq + r$  and
- ▶  $0 \leq r < N$

- ▶ We say  $a \text{ div } N = q$  and  $a \text{ mod } N = r$ .
- ▶ We also say

$$a \equiv b \pmod{N}$$

if and only if  $a \text{ mod } N = b \text{ mod } N$ .

# Groups

## Group $(G, \text{op})$

Let  $G$  be a non-empty set, and  $\text{op}$  be a binary operation on  $G$ . We say that  $G$  is a group with respect to  $\text{op}$  iff the following properties hold:

**Closure** For all  $x, y \in G$ ,  $x \text{ op } y \in G$ .

**Associativity** For all  $x, y, z \in G$ ,  $(x \text{ op } y) \text{ op } z = x \text{ op } (y \text{ op } z)$ .

**Identity** There exists  $1 \in G$  such that, for every  $x \in G$ ,  
 $x \text{ op } 1 = 1 \text{ op } x = x$ .

**Invertibility** For every  $x \in G$ , there exists  $x^{-1} \in G$  such that  
 $x \text{ op } x^{-1} = 1$ .

Try  $(\mathbf{Z}, +)$ ,  $(\mathbf{N}, +)$ ,  $(\mathbf{Z}^+, +)$ ,  $\mathbf{Z}_N$ , MOD-ADD,  $\mathbf{Z}_N$ , MOD-MULT,  $\mathbf{Z}_N^*$ , MOD-MULT

# Exponentiation

- ▶ If we write op as  $\cdot$  and represent the identity element as 1, then exponentiation is written and means what you naturally think it means.
  - ▶  $a^0 = 1$
  - ▶  $a^n = a \cdot \dots \cdot a$  altogether  $n$  terms
  - ▶  $a^{-n} = a^{-1} \dots a^{-1}$  altogether  $n$  terms
  - ▶  $a^{i+j} = a^i \cdot a^j$
  - ▶  $a^{-1} = (a^i)^{-1} = (a^{-1})^i$

# Computational shortcuts

To compute  $5 \cdot 8 \cdot 10 \cdot 16 \bmod 21$ ,

- ▶ don't do this:  $5 \cdot 8 \cdot 10 \cdot 16 = 6400$  then modulo 21
- ▶ do this instead:  $((5 \cdot 8 \bmod 21) \cdot 10 \bmod 21) \dots$

# Group orders

## Group order

The **order** of a group  $G$  is  $|G|$ .

Try  $\mathbf{Z}_{21}^*$ . Write out all the elements in this group and count them.

## Fact

If  $G$  is a group of order  $m$  and if  $a \in G$ , then  $a^m = 1$ .

- ▶ Try  $5^{12} \in \mathbf{Z}_{21}^*$
- ▶ Try  $8^{12} \in \mathbf{Z}_{21}^*$



## Group orders (cont.)

### Corollary

If  $G$  is a group of order  $m$  and if  $a \in G$ , then for any  $i \in \mathbf{Z}$ ,

$$a^i = a^{i \bmod m}.$$

► Try  $5^{74} \bmod 21$

## Running time of computation on large numbers

Operating on large numbers, we no longer assume that basic operations (like addition and multiplication) take constant time. The running time needs be measured in the size of the inputs.

| Algorithm      | Input     | Output           | Time              |
|----------------|-----------|------------------|-------------------|
| INT-DIV        | $a, N$    | $q, r$           | quadratic         |
| MOD            | $a, N$    | $a \bmod N$      | quadratic         |
| EXT-GCD        | $a, N$    | $(d, a', N')$    | quadratic         |
| MOD-ADD        | $a, b, N$ | $a + b \bmod N$  | linear            |
| MOD-MULT       | $a, b, N$ | $ab \bmod N$     | quadratic         |
| MOD-INV        | $a, N$    | $a^{-1} \bmod N$ | quadratic         |
| MOD-EXP        | $a, n, N$ | $a^n \bmod N$    | cubic             |
| $\text{EXP}_G$ | $a, n$    | $a^n \in G$      | $O( n )$ $G$ -ops |

## Extended gcd

EXT-GCD( $a, N$ ) returns  $(d, a', N')$  such that

$$d = \gcd(a, N) = a \cdot a' + N \cdot N' .$$

Examples: EXT-GCD(25,10), EXT-GCD(10,25),  
EXT-GCD(250,12)

# Extended Euclidean algorithm

Assume that  $(a, N) \neq (0, 0)$ . We know that

$$\gcd(a, N) = \gcd(N, a \bmod N) .$$

So we can compute GCD using a recursive algorithm as follows.

Algorithm **EXT-GCD**( $a, N$ )

  If  $N = 0$  then Return  $(a, 1, 0)$

$(q, r) \leftarrow \text{INT-DIV}(a, N)$

$(d, x, y) \leftarrow \text{EXT-GCD}(N, r)$

$a' \leftarrow y ; N' \leftarrow x - qy$

  Return  $(d, a', N')$

The running time is  $O(|a| \cdot |N|)$ , so quadratic time.

# Modular Inverse

For  $a, N$  such that  $\gcd(a, N) = 1$ , we want to compute  $a^{-1} \bmod N$ , i.e.,

unique  $a' \in \mathbf{Z}_N^*$  such that  $aa' \equiv 1 \pmod{N}$ .

Let  $(d, a', N') \leftarrow \text{EXT-GCD}(a, N)$ , then

$$d = 1 = \gcd(a, N) = a \cdot a' + N \cdot N'.$$

Since  $N \cdot N' \equiv 0 \pmod{N}$ , we have  $a \cdot a' \equiv 1 \pmod{N}$ .

Algorithm MOD-INV( $a, N$ ):

$(d, a', N') \leftarrow \text{EXT-GCD}(a, N)$

    Return  $a' \bmod N$

# Modular Exponentiation

Let  $G$  be a group and  $a \in G$ . For  $x \in \mathbf{N}$ , compute

$$a^x = \underbrace{a \cdots a}_x$$

Suppose we compute  $a^x$  by collecting one term at a time, i.e.,

$y \leftarrow 1$

For  $i \in \{1, 2, \dots, x\}$  do

$y \leftarrow y \cdot a$

Return  $y$

What is the running time of this algorithm?

# Square-and-Multiply Exponentiation

Suppose the binary length of  $x$  is 4, so  $x$  has the form  $b_3b_2b_1b_0$ . Then,

$$\begin{aligned}x &= 2^3b_3 + 2^2b_2 + 2^1b_1 + 2^0b_0 \\ &= 8b_3 + 4b_2 + 2b_1 + b_0\end{aligned}$$

To compute  $a^x$ , we square and multiply iteratively as follows:

$$\begin{aligned}y_4 &= 1 \\ y_3 &= y_4^2 \cdot a^{b_3} = a^{b_3} \\ y_2 &= y_3^2 \cdot a^{b_2} = a^{2b_3+b_2} \\ y_1 &= y_2^2 \cdot a^{b_1} = a^{4b_3+2b_2+b_1} \\ y_0 &= y_1^2 \cdot a^{b_0} = a^{8b_3+4b_2+2b_1+b_0}\end{aligned}$$

We get  $a^x = y_0$ . What is the running time of this algorithm?

# Subgroups

## Subgroup

If  $G$  is a group and  $S \subseteq G$ , then  $S$  is a **subgroup** of  $G$  iff  $S$  is a group under  $G$ 's operation.

- Try  $G = \mathbf{Z}_{11}^*$  and  $S = \{1, 2, 3\}$ . Is  $S$  a subgroup?



# Order of a group element

Let  $G$  be a finite group.

## Order of a group element

The **order** of  $g \in G$ , denoted  $o(g)$ , is the smallest integer  $n \geq 1$  such that  $g^n = 1$ .

Try  $G = \mathbf{Z}_{11}^* = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$ .

| $i$            | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|----------------|---|---|---|---|---|---|---|---|---|---|----|
| $2^i \bmod 11$ |   |   |   |   |   |   |   |   |   |   |    |
| $5^i \bmod 11$ |   |   |   |   |   |   |   |   |   |   |    |

What are  $o(2)$  and  $o(5)$ ?

## Subgroup generated by $g \in G$

For  $g \in G$ , define

$$\langle g \rangle = \{g^0, g^1, \dots, g^{o(g)-1}\}.$$

$\langle g \rangle$  is a subgroup of  $G$ , and its order is  $o(g)$ .

### Lagrange's Theorem

The order of a subgroup  $S$  of  $G$  always divides the order of  $G$ .

Try  $G = \mathbf{Z}_{11}^*$  and list elements of  $\langle 2 \rangle$  and  $\langle 5 \rangle$ . What are  $o(2)$  and  $o(5)$ ?

# Cyclic Group

## Cyclic group

A group  $G$  is **cyclic** iff there exists  $g$  such that  $\langle g \rangle = G$ .

Such an element  $g$  is called a **generator**.

## Discrete Log problem

Consider  $\mathbf{Z}_{11}^*$ . We know that 2 is a generator of  $\mathbf{Z}_{11}^*$ .

|                |   |   |   |   |   |    |   |   |   |   |
|----------------|---|---|---|---|---|----|---|---|---|---|
| $i$            | 0 | 1 | 2 | 3 | 4 | 5  | 6 | 7 | 8 | 9 |
| $2^i \bmod 11$ | 1 | 2 | 4 | 8 | 5 | 10 | 9 | 7 | 3 | 6 |

We say that, for example,

$$\text{Dlog}_2(8) = 3 .$$

Let  $G = \langle g \rangle$  be a cyclic group. Then, given  $a \in G$ , find

$$\text{Dlog}_g(a) .$$

# Discrete Log Problem: game formulation

Let  $G = \langle g \rangle$  be a cyclic group of order  $m$ , and let  $A$  be an adversary.

## DLP game

Game  $\text{DL}_{G,g}(A)$ :

$$x \xleftarrow{\$} \mathbf{Z}_m$$

$$X \leftarrow g^x$$

$$x' \leftarrow A(X)$$

Return  $(x = x')$

The **dl-advantage** of  $A$  is

$$\text{Adv}_{G,g}^{\text{dl}}(A) = \Pr [\text{DL}_{G,g}(A) \Rightarrow \text{true}] .$$

# Computational Diffie-Hellman problem

Consider  $\mathbf{Z}_{11}^*$ . We know that 2 is a generator of  $\mathbf{Z}_{11}^*$ .

| $i$            | 0 | 1 | 2 | 3 | 4 | 5  | 6 | 7 | 8 | 9 |
|----------------|---|---|---|---|---|----|---|---|---|---|
| $2^i \bmod 11$ | 1 | 2 | 4 | 8 | 5 | 10 | 9 | 7 | 3 | 6 |

Let  $G = \langle g \rangle$  be a cyclic group. Then, given  $X = g^x$  and  $Y = g^y$ , find

$$Z = g^{xy}.$$

# Computational Diffie-Hellman Problem: game formulation

Let  $G = \langle g \rangle$  be a cyclic group of order  $m$ , and let  $A$  be an adversary.

## CDH game

Game  $\text{CDH}_{G,g}(A)$ :

$$x \xleftarrow{\$} \mathbf{Z}_m ; y \xleftarrow{\$} \mathbf{Z}_m$$

$$X \leftarrow g^x ; Y \leftarrow g^y$$

$$Z \leftarrow A(X, Y)$$

$$\text{Return } (Z = g^{xy})$$

The **cdh-advantage** of  $A$  is

$$\mathbf{Adv}_{G,g}^{\text{cdh}}(A) = \Pr [\text{CDH}_{G,g}(A) \Rightarrow \text{true}] .$$

# Decisional Diffie-Hellman problem

Consider  $\mathbf{Z}_{11}^*$ . We know that 2 is a generator of  $\mathbf{Z}_{11}^*$ .

| $i$            | 0 | 1 | 2 | 3 | 4 | 5  | 6 | 7 | 8 | 9 |
|----------------|---|---|---|---|---|----|---|---|---|---|
| $2^i \bmod 11$ | 1 | 2 | 4 | 8 | 5 | 10 | 9 | 7 | 3 | 6 |

Let  $G = \langle g \rangle$  be a cyclic group. Then, given  $X = g^x$ ,  $Y = g^y$ , and  $Z \in G$ , figure out whether

$$Z = g^{xy}.$$



# Decisional Diffie-Hellman Problem: game formulation

Let  $G = \langle g \rangle$  be a cyclic group of order  $m$ , and let  $A$  be an adversary.

## DDH game

Game  $\text{DDH}_{G,g}(A)$ :

$b \in \{0, 1\}$

$x \xleftarrow{\$} \mathbf{Z}_m ; y \xleftarrow{\$} \mathbf{Z}_m$

$X \leftarrow g^x ; Y \leftarrow g^y$

If  $b = 0$  then  $Z \xleftarrow{\$} G$  else  $Z \leftarrow g^{xy}$

$d \leftarrow A(X, Y, Z)$

Return  $(d = b)$

The **ddh-advantage** of  $A$  is

$$\mathbf{Adv}_{G,g}^{\text{ddh}}(A) = 2 \cdot \Pr[\text{DDH}_{G,g}(A) \Rightarrow \text{true}] - 1.$$

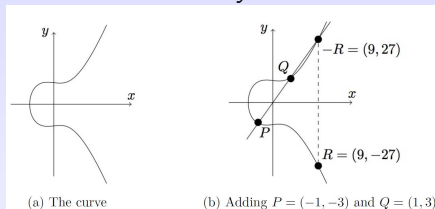
# Hardness of DLP

**Problem:** If we use  $G = Z_p^*$  where  $p$  is a prime, we need to make  $p$  **very large** to ensure that DLP is hard in  $G$ .

**Solution:** Use an elliptic curve group  $E/\mathbb{F}_p$ . It turns out that, for an elliptic curve group, DLP is hard even for **small**  $p$ .

# Elliptic Curve Group

This figure below shows the curve  $y^2 = x^3 - x + 9$  over the reals.



## Elliptic Curve over Finite Field

Let  $p > 3$  be a prime. An **elliptic curve**  $E$  over  $\mathbb{F}_p$ , denoted  $E/\mathbb{F}_p$ , is an equation

$$y^2 = x^3 + ax + b$$

where  $a, b \in \mathbb{F}_p$  satisfy  $4a^3 + 27b^2 \neq 0$ .

Try <https://grau1.de/code/elliptic2/>!

The condition  $4a^3 + 27b^2 \neq 0$  ensures that  $x^3 + ax + b = 0$  does not have a double root. This is to avoid certain degeneracies.

# Finite Field

## Definition

A field is a set  $\mathbb{F}$  together with two binary operations on  $\mathbb{F}$  called addition and multiplication. These operations are required to satisfy the following properties, referred to as field axioms (in these axioms,  $a, b$ , and  $c \in \mathbb{F}$ ):

- ▶ Associativity of addition and multiplication:  
 $a + (b + c) = (a + b) + c$ , and  $a \cdot (b \cdot c) = (a \cdot b) \cdot c$ .
- ▶ Commutativity of addition and multiplication:  $a + b = b + a$ , and  $a \cdot b = b \cdot a$ .
- ▶ Additive and multiplicative identity: there exist two different elements  $0$  and  $1$  in  $\mathbb{F}$  such that  $a + 0 = a$  and  $a \cdot 1 = a$ .
- ▶ Additive inverses: for every  $a$  in  $\mathbb{F}$ , there exists an element in  $\mathbb{F}$ , denoted  $-a$ , such that  $a + (-a) = 0$ .
- ▶ Multiplicative inverses: for every  $a \neq 0$  in  $\mathbb{F}$ , there exists an element in  $\mathbb{F}$ , denoted by  $a^{-1}$ , such that  $a \cdot a^{-1} = 1$ .
- ▶ Distributivity of multiplication over addition:  
 $a \cdot (b + c) = (a \cdot b) + (a \cdot c)$ .

# Diffie-Hellman Key Exchange protocol

Let  $G$  be a cyclic group of order  $m$ , and let  $g$  be a generator.

Alice

Bob

$$x \xleftarrow{\$} \mathbf{Z}_m$$

$$X \leftarrow g^x$$

$$\xrightarrow{X}$$

$$y \xleftarrow{\$} \mathbf{Z}_m$$

$$Y \leftarrow g^y$$

$$\xleftarrow{Y}$$

$$K_A = Y^x$$

$$K_B = X^y$$

# Relative hardness of discrete-log-based problems

Alice

$$x \xleftarrow{\$} \mathbf{Z}_m$$

$$X \leftarrow g^x$$

$$K_A = Y^x$$

Bob

$$y \xleftarrow{\$} \mathbf{Z}_m$$

$$Y \leftarrow g^y$$

$$K_B = X^y$$

$\xrightarrow{X}$

$\xleftarrow{Y}$

Break DH KE

↑

easy

solve DDH

↑

easy

solve CDH

↑

easy

solve DLP