Cryptography

Lecture 17

Arkady Yerukhimovich

October 28, 2024

Outline

1. Lecture 16 Review
2. A Modern Cryptography Approach (Chapter 8.Intro)
3. A Brief Intro to Number Theory (Chapter 8.1)
4. A Brief Intro to Group Theory (Chapter 8.1)

Lecture 16 Review

- AES review
- Feistel Networks and DES
- Davies-Meyer Transform

Outline

1. Lecture 16 Review
2. A Modern Cryptography Approach (Chapter 8.Intro)
3. A Brief Intro to Number Theory (Chapter 8.1)
4. A Brief Intro to Group Theory (Chapter 8.1)

Let's Look At Our Assumptions

How to instantiate private-key crypto, so far:

1. Use primitives like PRGs, PRFs, PRPs to build encryption, MACs
2. Instantiate PRPs using block-ciphers

Let's Look At Our Assumptions

How to instantiate private-key crypto, so far:

1. Use primitives like PRGs, PRFs, PRPs to build encryption, MACs
2. Instantiate PRPs using block-ciphers

What's The Problem?

Let's Look At Our Assumptions

How to instantiate private-key crypto, so far:

1. Use primitives like PRGs, PRFs, PRPs to build encryption, MACs
2. Instantiate PRPs using block-ciphers

What's The Problem?

- All block-cipher constructions rely on sequence of random-looking steps (e.g., random permutation, shift bits, taking subsets of bits,...)

Let's Look At Our Assumptions

How to instantiate private-key crypto, so far:

1. Use primitives like PRGs, PRFs, PRPs to build encryption, MACs
2. Instantiate PRPs using block-ciphers

What's The Problem?

- All block-cipher constructions rely on sequence of random-looking steps (e.g., random permutation, shift bits, taking subsets of bits,...)
- No way to formalize a clean (falsifiable) reason for why these are secure

Let's Look At Our Assumptions

How to instantiate private-key crypto, so far:

1. Use primitives like PRGs, PRFs, PRPs to build encryption, MACs
2. Instantiate PRPs using block-ciphers

What's The Problem?

- All block-cipher constructions rely on sequence of random-looking steps (e.g., random permutation, shift bits, taking subsets of bits,...)
- No way to formalize a clean (falsifiable) reason for why these are secure
- Just have to trust that block-ciphers (e.g., AES, DES) are secure

Let's Look At Our Assumptions

How to instantiate private-key crypto, so far:

1. Use primitives like PRGs, PRFs, PRPs to build encryption, MACs
2. Instantiate PRPs using block-ciphers

What's The Problem?

- All block-cipher constructions rely on sequence of random-looking steps (e.g., random permutation, shift bits, taking subsets of bits,...)
- No way to formalize a clean (falsifiable) reason for why these are secure
- Just have to trust that block-ciphers (e.g., AES, DES) are secure

Key Question

How can we build crypto on clean mathematical foundations?

Building Crypto from Mathematical Assumptions

Instead, modern crypto builds cryptographic primitives from clean, mathematical problems (e.g., factoring)

Building Crypto from Mathematical Assumptions

Instead, modern crypto builds cryptographic primitives from clean, mathematical problems (e.g., factoring)

- Allows reducing security of primitives to solving (well studied) mathematical problem

Building Crypto from Mathematical Assumptions

Instead, modern crypto builds cryptographic primitives from clean, mathematical problems (e.g., factoring)

- Allows reducing security of primitives to solving (well studied) mathematical problem
- Problems are (often) studied even outside of crypto

Building Crypto from Mathematical Assumptions

Instead, modern crypto builds cryptographic primitives from clean, mathematical problems (e.g., factoring)

- Allows reducing security of primitives to solving (well studied) mathematical problem
- Problems are (often) studied even outside of crypto
- Assumptions are easy to state and understand

Building Crypto from Mathematical Assumptions

Instead, modern crypto builds cryptographic primitives from clean, mathematical problems (e.g., factoring)

- Allows reducing security of primitives to solving (well studied) mathematical problem
- Problems are (often) studied even outside of crypto
- Assumptions are easy to state and understand
- Can be stated asymptotically, not relying on fixed input/output sizes

Building Crypto from Mathematical Assumptions

Instead, modern crypto builds cryptographic primitives from clean, mathematical problems (e.g., factoring)

- Allows reducing security of primitives to solving (well studied) mathematical problem
- Problems are (often) studied even outside of crypto
- Assumptions are easy to state and understand
- Can be stated asymptotically, not relying on fixed input/output sizes

Added Functionality

We will show next week, how this modern crypto approach leads to the development of *public-key* cryptography.

Outline

Preliminaries and Notation

- $\mathbb{Z}$ - set of Integers

Preliminaries and Notation

- $\mathbb{Z}$ - set of Integers
- For integer n, $||n|| = \lfloor \log n \rfloor + 1$ is the number of bits to represent n
  - We require efficient algorithms to run in time $poly(||n||)$

Preliminaries and Notation

- $\mathbb{Z}$ - set of Integers
- For integer n, $||n|| = \lfloor \log n \rfloor + 1$ is the number of bits to represent n
  - We require efficient algorithms to run in time $poly(||n||)$
- $a|b$ – a divides b ($\exists c \in \mathbb{Z}$ s.t. $ac = b$)

Preliminaries and Notation

- $\mathbb{Z}$ - set of Integers
- For integer n, $||n|| = \lfloor \log n \rfloor + 1$ is the number of bits to represent n
  - We require efficient algorithms to run in time $poly(||n||)$
- $a|b$ – a divides b ($\exists c \in \mathbb{Z}$ s.t. $ac = b$)
- If $a|b$ and $a > 0$, $a \notin \{1, b\}$ then a is a *factor* of b

Preliminaries and Notation

- $\mathbb{Z}$ - set of Integers
- For integer n, $||n|| = \lfloor \log n \rfloor + 1$ is the number of bits to represent n
  - We require efficient algorithms to run in time $poly(||n||)$
- $a|b$ – a divides b ($\exists c \in \mathbb{Z}$ s.t. $ac = b$)
- If $a|b$ and $a > 0$, $a \notin \{1, b\}$ then a is a *factor* of b
- Positive integer $p > 1$ is *prime* if it has no factors

Preliminaries and Notation

- $\mathbb{Z}$ - set of Integers
- For integer n, $||n|| = \lfloor \log n \rfloor + 1$ is the number of bits to represent n
  - We require efficient algorithms to run in time $poly(||n||)$
- $a|b$ – a divides b ($\exists c \in \mathbb{Z}$ s.t. $ac = b$)
- If $a|b$ and $a > 0$, $a \notin \{1, b\}$ then a is a *factor* of b
- Positive integer $p > 1$ is *prime* if it has no factors
- An integer $p > 1$ that is not prime is *composite*

Preliminaries and Notation

- $\mathbb{Z}$ - set of Integers
- For integer n, $||n|| = \lfloor \log n \rfloor + 1$ is the number of bits to represent n
  - We require efficient algorithms to run in time $poly(||n||)$
- $a|b$ – a divides b ($\exists c \in \mathbb{Z}$ s.t. $ac = b$)
- If $a|b$ and $a > 0$, $a \notin \{1, b\}$ then a is a *factor* of b
- Positive integer $p > 1$ is *prime* if it has no factors
- An integer $p > 1$ that is not prime is *composite*

Fundamental Theorem of Arithmetic

All positive integers $n > 1$ can be expressed uniquely (up to ordering) as $n = \prod p_i^{\ell_i}$ for primes p_i

Greatest Common Divisor (gcd)

Definition

For $a, b \in \mathbb{Z}$, $gcd(a, b) = c$ s.t. c is the largest integer so that $c|a$ and $c|b$

Properties of gcd:

Greatest Common Divisor (gcd)

Definition

For $a, b \in \mathbb{Z}$, $gcd(a, b) = c$ s.t. c is the largest integer so that $c|a$ and $c|b$

Properties of gcd:

1. If $c|ab$ and $gcd(a, c) = 1$, then $c|b$
    - If p is prime, then $p|ab$ implies that $p|a$ or $p|b$

Greatest Common Divisor (gcd)

Definition

For $a, b \in \mathbb{Z}$, $gcd(a, b) = c$ s.t. c is the largest integer so that $c|a$ and $c|b$

Properties of gcd:

1. If $c|ab$ and $gcd(a, c) = 1$, then $c|b$
   - If p is prime, then $p|ab$ implies that $p|a$ or $p|b$
2. If $a|N$, $b|N$, and $gcd(a, b) = 1$ then $ab|N$

Greatest Common Divisor (gcd)

Definition

For $a, b \in \mathbb{Z}$, $gcd(a, b) = c$ s.t. c is the largest integer so that $c|a$ and $c|b$

Properties of gcd:

1. If $c|ab$ and $gcd(a, c) = 1$, then $c|b$
   - If p is prime, then $p|ab$ implies that $p|a$ or $p|b$
2. If $a|N$, $b|N$, and $gcd(a, b) = 1$ then $ab|N$
3. If $a, b \in \mathbb{Z}^+$, there exist $X, Y \in \mathbb{Z}$ such that $Xa + Yb = gcd(a, b)$
   - $gcd(a, b)$ is the smallest positive integer that can be written like this

Greatest Common Divisor (gcd)

Definition

For $a, b \in \mathbb{Z}$, $gcd(a, b) = c$ s.t. c is the largest integer so that $c|a$ and $c|b$

Properties of gcd:

1. If $c|ab$ and $gcd(a, c) = 1$, then $c|b$
   - If p is prime, then $p|ab$ implies that $p|a$ or $p|b$
2. If $a|N$, $b|N$, and $gcd(a, b) = 1$ then $ab|N$
3. If $a, b \in \mathbb{Z}^+$, there exist $X, Y \in \mathbb{Z}$ such that $Xa + Yb = gcd(a, b)$
   - $gcd(a, b)$ is the smallest positive integer that can be written like this
4. $gcd(a, b) = gcd(b, [a \bmod b])$ if $a, b > 1$ such that $b \not| \ a$

Euclidean Algorithm

Goal

Given Integers a, b find $c = gcd(a, b)$.

$GCD(a, b)$:

Euclidean Algorithm

Goal

Given Integers a, b find $c = gcd(a, b)$.

$GCD(a, b)$:

1. If $b|a$, return b

Euclidean Algorithm

Goal

Given Integers a, b find $c = gcd(a, b)$.

$GCD(a, b)$:

1. If $b|a$, return b
2. Else, return $GCD(b, [a \bmod b])$

Euclidean Algorithm

Goal

Given Integers a, b find $c = gcd(a, b)$.

$GCD(a, b)$:

1. If $b|a$, return b
2. Else, return $GCD(b, [a \bmod b])$

Extended Euclidean Algorithm:

- Also lets you find X, Y such that $Xa + Yb = gcd(a, b)$

Euclidean Algorithm

Goal

Given Integers a, b find $c = gcd(a, b)$.

$GCD(a, b)$:

1. If $b|a$, return b
2. Else, return $GCD(b, [a \bmod b])$

Extended Euclidean Algorithm:

- Also lets you find X, Y such that $Xa + Yb = gcd(a, b)$

Both of these are poly-time in $||a||$ and $||b||$

Modular Arithmetic

Notation: For Integers a, b, N

- $[a \bmod N]$ - a modulo N, (e.g. $[15 \bmod 7] = 1$)

Modular Arithmetic

Notation: For Integers a, b, N

- $[a \bmod N]$ - a modulo N, (e.g. $[15 \bmod 7] = 1$)
- $a = b \bmod N$ if $[a \bmod N] = [b \bmod N]$
    - We say, a is *congruent* to $b \mod N$

Modular Arithmetic

Notation: For Integers a, b, N

- $[a \bmod N]$ - a modulo N, (e.g. $[15 \bmod 7] = 1$)
- $a = b \bmod N$ if $[a \bmod N] = [b \bmod N]$
  - We say, a is *congruent* to $b \mod N$
- Note that:

$$
\begin{aligned}
a = [b \bmod N] &\implies a = b \bmod N \text{, but} \\
a = b \bmod N &\;\not\Longrightarrow\; a = [b \bmod N]
\end{aligned}
$$

Modular Arithmetic

Notation: For Integers a, b, N

- $[a \bmod N]$ - a modulo N, (e.g. $[15 \bmod 7] = 1$)
- $a = b \bmod N$ if $[a \bmod N] = [b \bmod N]$
  - We say, a is *congruent* to $b \mod N$
- Note that:

$$\begin{aligned} a = [b \bmod N] &\implies a = b \bmod N, \text{ but} \\ a = b \bmod N &\;\;\not\Longrightarrow\;\; a = [b \bmod N] \end{aligned}$$

Example: $8 \neq [3 \bmod 5]$, but $8 = 3 \bmod 5$

Modular Arithmetic

Congruence Relation

Congruence $\mod N$ is an *equivalence relation* that obeys standard rules of arithmetic.

If $a = a' \mod N$, and $b = b' \mod N$ then:

- $a + b = a' + b' \mod N$
- $ab = a'b' \mod N$

Modular Arithmetic

Congruence Relation

Congruence mod N is an *equivalence relation* that obeys standard rules of arithmetic.

If $a = a' \mod N$, and $b = b' \mod N$ then:

- $a + b = a' + b' \mod N$
- $ab = a'b' \mod N$

Example: $[654321 \cdot 54301 \mod 100] = [21 \cdot 1 \mod 100] = 21$

Modular Arithmetic

Congruence Relation

Congruence mod N is an *equivalence relation* that obeys standard rules of arithmetic.

If $a = a' \mod N$, and $b = b' \mod N$ then:

- $a + b = a' + b' \mod N$
- $ab = a'b' \mod N$

Example: $[654321 \cdot 54301 \mod 100] = [21 \cdot 1 \mod 100] = 21$

But, this congruence relation does not necessarily respect division
Example: $3 \cdot 2 = 6 = 15 \cdot 2 \mod 24$, but $3 \neq 15 \mod 24$

Modular Arithmetic

Congruence Relation

Congruence $\mod N$ is an *equivalence relation* that obeys standard rules of arithmetic.

If $a = a' \mod N$, and $b = b' \mod N$ then:

- $a + b = a' + b' \mod N$
- $ab = a'b' \mod N$

Example: $[654321 \cdot 54301 \mod 100] = [21 \cdot 1 \mod 100] = 21$

But, this congruence relation does not necessarily respect division
Example: $3 \cdot 2 = 6 = 15 \cdot 2 \mod 24$, but $3 \neq 15 \mod 24$

Division $\mod N$

Let $b, N \in \mathbb{Z}, b \geq 1, N > 1$, b is invertible $\mod N$ (i.e., can divide by b) if and only if $gcd(b, N) = 1$

Finding Multiplicative Inverse

Goal

For $b, N \in \mathbb{Z}$, find $b^{-1} \bmod N$

Finding Multiplicative Inverse

Goal

For $b, N \in \mathbb{Z}$, find $b^{-1} \bmod N$

Idea: Use extended Euclidean algorithm

- Find $X, Y \in \mathbb{Z}$ s.t. $Xb + YN = gcd(b, N)$

Finding Multiplicative Inverse

Goal

For $b, N \in \mathbb{Z}$, find $b^{-1} \bmod N$

Idea: Use extended Euclidean algorithm

- Find $X, Y \in \mathbb{Z}$ s.t. $Xb + YN = gcd(b, N)$
- Recall that for b^{-1} to exist, $gcd(b, N) = 1$

Finding Multiplicative Inverse

Goal

For $b, N \in \mathbb{Z}$, find $b^{-1} \bmod N$

Idea: Use extended Euclidean algorithm

- Find $X, Y \in \mathbb{Z}$ s.t. $Xb + YN = gcd(b, N)$
- Recall that for b^{-1} to exist, $gcd(b, N) = 1$
- $Xb + YN = 1 \implies Xb = 1 - YN \implies Xb = 1 \bmod N$

Finding Multiplicative Inverse

Goal

For $b, N \in \mathbb{Z}$, find $b^{-1} \bmod N$

Idea: Use extended Euclidean algorithm

- Find $X, Y \in \mathbb{Z}$ s.t. $Xb + YN = gcd(b, N)$
- Recall that for b^{-1} to exist, $gcd(b, N) = 1$
- $Xb + YN = 1 \implies Xb = 1 - YN \implies Xb = 1 \bmod N$
- $X = b^{-1} \bmod N$

Outline

Definition of a Group

A group is a set G with a binary operation $(\cdot)$ such that:

Definition of a Group

A group is a set G with a binary operation $(\cdot)$ such that:

- Closure: $\forall g, h, \in G, g \cdot h \in G$

Definition of a Group

A group is a set G with a binary operation $(\cdot)$ such that:

- Closure: $\forall g, h, \in G, g \cdot h \in G$
- Identity: $\exists$ element $1_G \in G$ s.t. $\forall g \in G, 1_G \cdot g = g \cdot 1_G = g$

Group Theory

Definition of a Group

A group is a set G with a binary operation $(\cdot)$ such that:

- Closure: $\forall g, h, \in G, g \cdot h \in G$
- Identity: $\exists$ element $1_G \in G$ s.t. $\forall g \in G, 1_G \cdot g = g \cdot 1_G = g$
- Inverse: $\forall g \in G, \exists h \in G$ s.t. $g \cdot h = h \cdot g = 1_G$

Group Theory

Definition of a Group

A group is a set G with a binary operation $(\cdot)$ such that:

- Closure: $\forall g, h, \in G, g \cdot h \in G$
- Identity: $\exists$ element $1_G \in G$ s.t. $\forall g \in G, 1_G \cdot g = g \cdot 1_G = g$
- Inverse: $\forall g \in G, \exists h \in G$ s.t. $g \cdot h = h \cdot g = 1_G$
- Associativity: $\forall g_1, g_2, g_3 \in G, (g_1 \cdot g_2) \cdot g_3 = g_1 \cdot (g_2 \cdot g_3)$

Group Theory

Definition of a Group

A group is a set G with a binary operation $(\cdot)$ such that:

- Closure: $\forall g, h, \in G, g \cdot h \in G$
- Identity: $\exists$ element $1_G \in G$ s.t. $\forall g \in G, 1_G \cdot g = g \cdot 1_G = g$
- Inverse: $\forall g \in G, \exists h \in G$ s.t. $g \cdot h = h \cdot g = 1_G$
- Associativity: $\forall g_1, g_2, g_3 \in G, (g_1 \cdot g_2) \cdot g_3 = g_1 \cdot (g_2 \cdot g_3)$

Additional definitions:

Definition of a Group

A group is a set G with a binary operation $(\cdot)$ such that:

- Closure: $\forall g, h, \in G, g \cdot h \in G$
- Identity: $\exists$ element $1_G \in G$ s.t. $\forall g \in G, 1_G \cdot g = g \cdot 1_G = g$
- Inverse: $\forall g \in G, \exists h \in G$ s.t. $g \cdot h = h \cdot g = 1_G$
- Associativity: $\forall g_1, g_2, g_3 \in G, (g_1 \cdot g_2) \cdot g_3 = g_1 \cdot (g_2 \cdot g_3)$

Additional definitions:

- G is *abelian* if commutativity holds: $\forall g, h \in G, g \cdot h = h \cdot g$

Group Theory

Definition of a Group

A group is a set G with a binary operation $(\cdot)$ such that:

- Closure: $\forall g, h, \in G, g \cdot h \in G$
- Identity: $\exists$ element $1_G \in G$ s.t. $\forall g \in G, 1_G \cdot g = g \cdot 1_G = g$
- Inverse: $\forall g \in G, \exists h \in G$ s.t. $g \cdot h = h \cdot g = 1_G$
- Associativity: $\forall g_1, g_2, g_3 \in G, (g_1 \cdot g_2) \cdot g_3 = g_1 \cdot (g_2 \cdot g_3)$

Additional definitions:

- G is *abelian* if commutativity holds: $\forall g, h \in G, g \cdot h = h \cdot g$
- $|G|$ - *order* of G (number of elements in G) - For us $|G| < \infty$

Group Theory

Definition of a Group

A group is a set G with a binary operation $(\cdot)$ such that:

- Closure: $\forall g, h, \in G, g \cdot h \in G$
- Identity: $\exists$ element $1_G \in G$ s.t. $\forall g \in G, 1_G \cdot g = g \cdot 1_G = g$
- Inverse: $\forall g \in G, \exists h \in G$ s.t. $g \cdot h = h \cdot g = 1_G$
- Associativity: $\forall g_1, g_2, g_3 \in G, (g_1 \cdot g_2) \cdot g_3 = g_1 \cdot (g_2 \cdot g_3)$

Additional definitions:

- G is *abelian* if commutativity holds: $\forall g, h \in G, g \cdot h = h \cdot g$
- $|G|$ - *order* of G (number of elements in G) - For us $|G| < \infty$
- Exponentiation in G: $g^x = g \cdot g \cdots g$ (x times)

Group Theory

Definition of a Group

A group is a set G with a binary operation $(\cdot)$ such that:

- Closure: $\forall g, h, \in G, g \cdot h \in G$
- Identity: $\exists$ element $1_G \in G$ s.t. $\forall g \in G, 1_G \cdot g = g \cdot 1_G = g$
- Inverse: $\forall g \in G, \exists h \in G$ s.t. $g \cdot h = h \cdot g = 1_G$
- Associativity: $\forall g_1, g_2, g_3 \in G, (g_1 \cdot g_2) \cdot g_3 = g_1 \cdot (g_2 \cdot g_3)$

Additional definitions:

- G is *abelian* if commutativity holds: $\forall g, h \in G, g \cdot h = h \cdot g$
- $|G|$ - *order* of G (number of elements in G) - For us $|G| < \infty$
- Exponentiation in G: $g^x = g \cdot g \cdots g$ (x times)

Examples:

Group Theory

Definition of a Group

A group is a set G with a binary operation $(\cdot)$ such that:

- Closure: $\forall g, h, \in G, g \cdot h \in G$
- Identity: $\exists$ element $1_G \in G$ s.t. $\forall g \in G, 1_G \cdot g = g \cdot 1_G = g$
- Inverse: $\forall g \in G, \exists h \in G$ s.t. $g \cdot h = h \cdot g = 1_G$
- Associativity: $\forall g_1, g_2, g_3 \in G, (g_1 \cdot g_2) \cdot g_3 = g_1 \cdot (g_2 \cdot g_3)$

Additional definitions:

- G is *abelian* if commutativity holds: $\forall g, h \in G, g \cdot h = h \cdot g$
- $|G|$ - *order* of G (number of elements in G) - For us $|G| < \infty$
- Exponentiation in G: $g^x = g \cdot g \cdots g$ (x times)

Examples:

- The integers, $\mathbb{Z}$, form an abelian group under addition

Group Theory

Definition of a Group

A group is a set G with a binary operation $(\cdot)$ such that:

- Closure: $\forall g, h, \in G, g \cdot h \in G$
- Identity: $\exists$ element $1_G \in G$ s.t. $\forall g \in G, 1_G \cdot g = g \cdot 1_G = g$
- Inverse: $\forall g \in G, \exists h \in G$ s.t. $g \cdot h = h \cdot g = 1_G$
- Associativity: $\forall g_1, g_2, g_3 \in G, (g_1 \cdot g_2) \cdot g_3 = g_1 \cdot (g_2 \cdot g_3)$

Additional definitions:

- G is *abelian* if commutativity holds: $\forall g, h \in G, g \cdot h = h \cdot g$
- $|G|$ - *order* of G (number of elements in G) - For us $|G| < \infty$
- Exponentiation in G: $g^x = g \cdot g \cdots g$ (x times)

Examples:

- The integers, $\mathbb{Z}$, form an abelian group under addition
- The integers, $\mathbb{Z}$, are not a group under multiplication (no inverses)

Group Theory

Definition of a Group

A group is a set G with a binary operation $(\cdot)$ such that:

- Closure: $\forall g, h, \in G, g \cdot h \in G$
- Identity: $\exists$ element $1_G \in G$ s.t. $\forall g \in G, 1_G \cdot g = g \cdot 1_G = g$
- Inverse: $\forall g \in G, \exists h \in G$ s.t. $g \cdot h = h \cdot g = 1_G$
- Associativity: $\forall g_1, g_2, g_3 \in G, (g_1 \cdot g_2) \cdot g_3 = g_1 \cdot (g_2 \cdot g_3)$

Additional definitions:

- G is *abelian* if commutativity holds: $\forall g, h \in G, g \cdot h = h \cdot g$
- $|G|$ - *order* of G (number of elements in G) - For us $|G| < \infty$
- Exponentiation in G: $g^x = g \cdot g \cdots g$ (x times)

Examples:

- The integers, $\mathbb{Z}$, form an abelian group under addition
- The integers, $\mathbb{Z}$, are not a group under multiplication (no inverses)
- $\mathbb{Z}_N = \{1, \ldots, N-1\}$ is a group under addition $\mod N$

Important Properties of Groups

1. $\forall a, b, c \in G$, if $ac = bc$, then $a = b$

Important Properties of Groups

1. $\forall a, b, c \in G$, if $ac = bc$, then $a = b$
   - Proof:

$$\begin{aligned} ac = bc \implies (ac)c^{-1} = (bc)c^{-1} &\implies a(cc^{-1}) = b(cc^{-1}) \\ &\implies a \cdot 1_G = b \cdot 1_G \implies a = b \end{aligned}$$

Important Properties of Groups

1. $\forall a, b, c \in G$, if $ac = bc$, then $a = b$
2. Let $|G| = m$, $\forall g \in G, g^m = 1$

Important Properties of Groups

1. $\forall a, b, c \in G$, if $ac = bc$, then $a = b$
2. Let $|G| = m$, $\forall g \in G, g^m = 1$
   - Proof (for abelian groups):
     Consider $(gg_1), (gg_2), \ldots, (gg_m)$ where $g_1, \ldots, g_m \in G$
     Since $(gg_i) = (gg_j)$ iff $g_i = g_j$ (by [1]), each of the (gg_i) is distinct
     Now, we have that

     $$g_1 \cdot g_2 \cdots g_m = (gg_1) \cdot (gg_2) \cdots (gg_m) = g^m \cdot (g_1 \cdot g_2 \cdots g_m)$$

     First equality holds because the (gg_i) are all possible values in G.
     So, $g^m = 1$

Important Properties of Groups

1. $\forall a, b, c \in G$, if $ac = bc$, then $a = b$
2. Let $|G| = m$, $\forall g \in G, g^m = 1$
3. Let $|G| = m$, then for any $g \in G$ and any $x \in \mathbb{Z}$, $g^x = g^{[x \bmod m]}$

Important Properties of Groups

1. $\forall a, b, c \in G$, if $ac = bc$, then $a = b$
2. Let $|G| = m$, $\forall g \in G, g^m = 1$
3. Let $|G| = m$, then for any $g \in G$ and any $x \in \mathbb{Z}$, $g^x = g^{[x \bmod m]}$
    - Proof:
      Let $x = qm + r$ where $q, r \in \mathbb{Z}$ and $r = [x \bmod m]$

$$g^x = g^{qm+r} = g^{qm} \cdot g^r = (g^m)^q \cdot g^r = 1_G^q \cdot g^r = g^r$$

Important Properties of Groups

1. $\forall a, b, c \in G$, if $ac = bc$, then $a = b$
2. Let $|G| = m$, $\forall g \in G, g^m = 1$
3. Let $|G| = m$, then for any $g \in G$ and any $x \in \mathbb{Z}$, $g^x = g^{[x \bmod m]}$
4. Let $|G| = m$, and let $e > 0 \in \mathbb{Z}$. Define $f_e : G \to G$ by $f_e(g) = g^e$.
   If $gcd(e, m) = 1$, then f_e is a permutation over G.
   If $d = e^{-1} \bmod m$, then $f_d = f_e^{-1}$.

Important Properties of Groups

1. $\forall a, b, c \in G$, if $ac = bc$, then $a = b$
2. Let $|G| = m$, $\forall g \in G, g^m = 1$
3. Let $|G| = m$, then for any $g \in G$ and any $x \in \mathbb{Z}$, $g^x = g^{[x \bmod m]}$
4. Let $|G| = m$, and let $e > 0 \in \mathbb{Z}$. Define $f_e : G \to G$ by $f_e(g) = g^e$.
   If $gcd(e, m) = 1$, then f_e is a permutation over G.
   If $d = e^{-1} \bmod m$, then $f_d = f_e^{-1}$.
   - Proof: Enough to prove that f_d is inverse of f_e
     For any $g \in G$, we have:

$$f_d(f_e(g)) = f_d(g^e) = (g^e)^d = g^{ed} = g^{[ed \bmod m]} = g^1 = g$$

Important Properties of Groups

1. $\forall a, b, c \in G$, if $ac = bc$, then $a = b$
2. Let $|G| = m$, $\forall g \in G, g^m = 1$
3. Let $|G| = m$, then for any $g \in G$ and any $x \in \mathbb{Z}$, $g^x = g^{[x \bmod m]}$
4. Let $|G| = m$, and let $e > 0 \in \mathbb{Z}$. Define $f_e : G \to G$ by $f_e(g) = g^e$.
   If $gcd(e, m) = 1$, then f_e is a permutation over G.
   If $d = e^{-1} \bmod m$, then $f_d = f_e^{-1}$.

Cyclic Groups

Notation: Let G be a group such that $|G| = m$

Cyclic Groups

Notation: Let G be a group such that $|G| = m$

- For $g \in G$, define $< g >= \{g^0, g^1, \ldots\}$ – the items generated by g

Cyclic Groups

Notation: Let G be a group such that $|G| = m$

- For $g \in G$, define $< g >= \{g^0, g^1, \ldots\}$ – the items generated by g
- *order* of $g \in G$ is smallest $i \leq m$ such that $g^i = 1$ (Note that $i|m$)

Cyclic Groups

Notation: Let G be a group such that $|G| = m$

- For $g \in G$, define $< g >= \{g^0, g^1, \ldots\}$ – the items generated by g
- *order* of $g \in G$ is smallest $i \leq m$ such that $g^i = 1$ (Note that $i|m$)
- $< g >= \{g^0, \ldots, g^{i-1}\}$ is a *subgroup* of G

Cyclic Groups

Notation: Let G be a group such that $|G| = m$

- For $g \in G$, define $< g >= \{g^0, g^1, \ldots\}$ – the items generated by g
- *order* of $g \in G$ is smallest $i \leq m$ such that $g^i = 1$ (Note that $i|m$)
- $< g >= \{g^0, \ldots, g^{i-1}\}$ is a *subgroup* of G
  - $g^x = g^{[x \bmod i]}$

Cyclic Groups

Notation: Let G be a group such that $|G| = m$

- For $g \in G$, define $< g >= \{g^0, g^1, \ldots\}$ – the items generated by g
- *order* of $g \in G$ is smallest $i \leq m$ such that $g^i = 1$ (Note that $i|m$)
- $< g >= \{g^0, \ldots, g^{i-1}\}$ is a *subgroup* of G
  - $g^x = g^{[x \bmod i]}$
  - $g^x = g^y$ iff $x = y \bmod i$

Cyclic Groups

Notation: Let G be a group such that $|G| = m$

- For $g \in G$, define $< g >= \{g^0, g^1, \ldots\}$ – the items generated by g
- *order* of $g \in G$ is smallest $i \leq m$ such that $g^i = 1$ (Note that $i|m$)
- $< g >= \{g^0, \ldots, g^{i-1}\}$ is a *subgroup* of G
  - $g^x = g^{[x \bmod i]}$
  - $g^x = g^y$ iff $x = y \bmod i$

Cyclic Group

A group G is *cyclic* if $\exists g \in G$ s.t. order$(g) = |G|$. I.e., $< g >= G$.

Cyclic Groups

Notation: Let G be a group such that $|G| = m$

- For $g \in G$, define $< g >= \{g^0, g^1, \ldots\}$ – the items generated by g
- *order* of $g \in G$ is smallest $i \leq m$ such that $g^i = 1$ (Note that $i|m$)
- $< g >= \{g^0, \ldots, g^{i-1}\}$ is a *subgroup* of G
  - $g^x = g^{[x \bmod i]}$
  - $g^x = g^y$ iff $x = y \bmod i$

Cyclic Group

A group G is *cyclic* if $\exists g \in G$ s.t. order$(g) = |G|$. I.e., $< g >= G$.

- g is called the *generator* of G

Cyclic Groups

Notation: Let G be a group such that $|G| = m$

- For $g \in G$, define $< g >= \{g^0, g^1, \ldots\}$ – the items generated by g
- *order* of $g \in G$ is smallest $i \leq m$ such that $g^i = 1$ (Note that $i|m$)
- $< g >= \{g^0, \ldots, g^{i-1}\}$ is a *subgroup* of G
  - $g^x = g^{[x \bmod i]}$
  - $g^x = g^y$ iff $x = y \bmod i$

Cyclic Group

A group G is *cyclic* if $\exists g \in G$ s.t. order$(g) = |G|$. I.e., $< g >= G$.

- g is called the *generator* of G

Useful property: If $|G|$ is prime, then G is cyclic. Moreover, all $g \in G$ except 1 are generators

Cyclic Groups

Notation: Let G be a group such that $|G| = m$

- For $g \in G$, define $< g >= \{g^0, g^1, \dots\}$ – the items generated by g
- *order* of $g \in G$ is smallest $i \leq m$ such that $g^i = 1$ (Note that $i|m$)
- $< g >= \{g^0, \dots, g^{i-1}\}$ is a *subgroup* of G
  - $g^x = g^{[x \bmod i]}$
  - $g^x = g^y$ iff $x = y \bmod i$

Cyclic Group

A group G is *cyclic* if $\exists g \in G$ s.t. order$(g) = |G|$. I.e., $< g >= G$.

- g is called the *generator* of G

Useful property: If $|G|$ is prime, then G is cyclic. Moreover, all $g \in G$ except 1 are generators

Examples:

Cyclic Groups

Notation: Let G be a group such that $|G| = m$

- For $g \in G$, define $< g >= \{g^0, g^1, \dots\}$ – the items generated by g
- *order* of $g \in G$ is smallest $i \leq m$ such that $g^i = 1$ (Note that $i|m$)
- $< g >= \{g^0, \dots, g^{i-1}\}$ is a *subgroup* of G
  - $g^x = g^{[x \bmod i]}$
  - $g^x = g^y$ iff $x = y \bmod i$

Cyclic Group

A group G is *cyclic* if $\exists g \in G$ s.t. order$(g) = |G|$. I.e., $< g >= G$.

- g is called the *generator* of G

Useful property: If $|G|$ is prime, then G is cyclic. Moreover, all $g \in G$ except 1 are generators

Examples:

- $\mathbb{Z}_N =< 1 >$

Cyclic Groups

Notation: Let G be a group such that $|G| = m$

- For $g \in G$, define $< g >= \{g^0, g^1, \ldots\}$ – the items generated by g
- *order* of $g \in G$ is smallest $i \leq m$ such that $g^i = 1$ (Note that $i|m$)
- $< g >= \{g^0, \ldots, g^{i-1}\}$ is a *subgroup* of G
  - $g^x = g^{[x \bmod i]}$
  - $g^x = g^y$ iff $x = y \bmod i$

Cyclic Group

A group G is *cyclic* if $\exists g \in G$ s.t. order$(g) = |G|$. I.e., $< g >= G$.

- g is called the *generator* of G

Useful property: If $|G|$ is prime, then G is cyclic. Moreover, all $g \in G$ except 1 are generators

Examples:

- $\mathbb{Z}_N =< 1 >$
- $\mathbb{Z}_p^*$ – Not all $g \in \mathbb{Z}_p^*$ are generators: $< 2 >= \{1, 2, 4\} \neq \mathbb{Z}_7^*$ but, $< 3 >= \{1, 3, 9 = 2, 6, 4, 5\}$ is a generator