

Lecture 04: Groups and Fields

Definition

A *group*, represented by (G, \circ) , is defined by a set G and a binary operator \circ that satisfy the following properties

- 1 **Closure.** For all $a, b \in G$, we have $a \circ b \in G$
- 2 **Associativity.** For all $a, b, c \in G$, we have $(a \circ b) \circ c = a \circ (b \circ c)$
- 3 **Identity.** There exists an element $e \in G$ such that for all $a \in G$, we have $a \circ e = a$
- 4 **Inverse.** For every element $a \in G$, there exists an element $(-a) \in G$ such that $a \circ (-a) = e$

A Quick Check

- Verify that $(\{0, 1\}^n, \oplus)$, where \oplus is the bit-wise XOR of bits, is a group
 - Closure and Associativity is trivial to verify
 - Show that $\overbrace{00 \cdots 0}^{n\text{-times}}$ is the identity
 - Show that for $a \in \{0, 1\}^n$, the inverse of a is a itself

One-time Pad extended to Arbitrary Groups

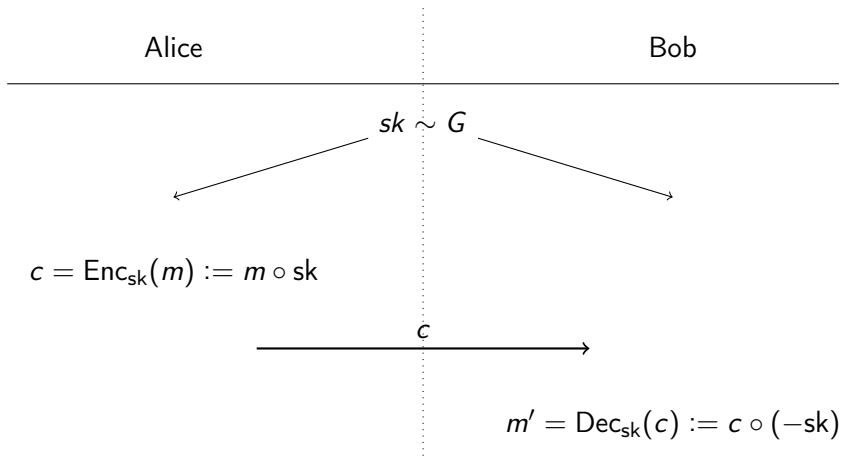


Figure: One-time Pad encryption scheme for the group (G, \circ) .

Verify that the scheme is always correct

Examples I

- Groups can be infinite size. $(\mathbb{Z}, +)$, where \mathbb{Z} is the set of all integers and $+$ is integer addition, is a group (Verify that it satisfies all properties of a group)
- Groups can be finite size. $(\mathbb{Z}_n, +)$, where $\mathbb{Z}_n = \{0, \dots, n-1\}$ and $+$ is integer addition mod n , is a group (Verify that it satisfies all properties of a group)

Examples II

Following are NOT groups. Find which rule is violated.

- (\mathbb{Z}, \times) , where \times is the integer multiplication
- (\mathbb{Z}^*, \times) , where \mathbb{Z}^* is the set of all non-zero integers and \times is the integer multiplication
- (\mathbb{Q}, \times) , where \mathbb{Q} is the set of all rationals and \times is rational multiplication

But (\mathbb{Q}^*, \times) , where \mathbb{Q}^* is the set of all non-zero rationals and \times is rational multiplication, is a group!

Examples III

- Prove that (\mathbb{Z}_p^*, \times) is a group when p is a prime, \times is integer multiplication mod p , and $\mathbb{Z}_p^* = \{1, \dots, p-1\}$
- Prove that (\mathbb{Z}_n^*, \times) is NOT a group when n is NOT a prime, \times is integer multiplication mod n , and $\mathbb{Z}_n^* = \{1, \dots, n-1\}$

Groups need not be commutative.

- Define a group that is not commutative. Hint: Consider G as the set of $n \times n$ full-rank matrices with elements in \mathbb{Q} . Now, define \circ as matrix multiplication.

In the homework we shall define left and right inverses, and left and right identity. We shall prove interesting properties regarding these inverses and identities.

Generator I

- Consider the group $(\mathbb{Z}_5, +)$
- Note that
 - 2 added 0-times is 0
 - 2 added 1-times is 2
 - 2 added 2-times is 4
 - 2 added 3-times is 1
 - 2 added 4-times is 3
 - 2 added 5-times is 0
 - (and so on)
- We say that 2 generates $(\mathbb{Z}_5, +)$ because we can generate the entire set \mathbb{Z}_5 by repeatedly “+”-ing 2 to itself
- Consider the group (\mathbb{Z}_7^*, \times) . Which elements in \mathbb{Z}_7 generate the group? And which elements do not generate the group?

- We will introduce a shorthand. By a^k , we represent the number $\overbrace{a \circ a \circ \cdots \circ a}^{k\text{-times}}$
- We define $a^0 = e$, the identity of the group

Repeated Squaring Technique

Let g be a generator of a group (G, \circ) . Consider the following algorithm.

- Let $n[0] := g$, the identity of (G, \circ)
- For $i = 1$ to k , do the following:
 - $n[i] := n[i - 1] \circ n[i - 1]$
- At the termination of the algorithm, we have the following $n[0] = g, n[1] = g^2, n[2] = g^4, \dots, n[k] = g^{2^k}$
- Note that we only used the \circ operation only k times in this algorithm to generate this sequence
- Let i be an integer in the range $\{0, \dots, 2^{k+1} - 1\}$
- How to compute g^i using $(k + 1)$ additional \circ operations?
- Note: This gives us an algorithm to compute g^i , where $i \in \{0, \dots, 2^{k+1} - 1\}$ using at most $(2k + 1)$ \circ operations!

Why Repeated Squaring is Efficient?

- Let (G, \circ) be a group generated by g
- Suppose we are interested in computing g^i
- First Algorithm: Multiply g i -times to get g^i . This method takes $O(i)$ time.
- Second Algorithm: Use repeated squaring to compute g^i . This method takes $O(\log i)$ time.
- Why is the first algorithm an exponential-time algorithm?
Why is the second algorithm a polynomial-time algorithm?

Definition

A field is defined by a set of elements \mathbb{F} , and two operators $+$ and \cdot . The field $(\mathbb{F}, +, \cdot)$ satisfies the following properties

- 1 **Closure.** For all $a, b \in \mathbb{F}$, we have $a + b \in \mathbb{F}$ and $a \cdot b \in \mathbb{F}$
- 2 **Associativity.** For all $a, b, c \in \mathbb{F}$, we have $(a + b) + c = a + (b + c)$ and $a \cdot (b \cdot c) = (a \cdot b) \cdot c$
- 3 **Commutativity.** For all $a, b \in \mathbb{F}$, we have $a + b = b + a$ and $a \cdot b = b \cdot a$
- 4 **Additive and Multiplicative identities.** There exists elements $0 \in \mathbb{F}$ and $1 \in \mathbb{F}$ such that for all $a \in \mathbb{F}$, we have $a + 0 = a$ and $a \cdot 1 = a$
- 5 **Additive inverse.** Every $a \in \mathbb{F}$ has $(-a) \in \mathbb{F}$ such that $a + (-a) = 0$
- 6 **Multiplicative inverse.** Every $0 \neq a \in \mathbb{G}$ has $(a^{-1}) \in \mathbb{F}$ such that $a \cdot (a^{-1}) = 1$
- 7 **Distributivity.** For all $a, b, c \in \mathbb{F}$, we have $a \cdot (b + c) = (a \cdot b) + (a \cdot c)$

Examples

- $(\mathbb{Z}_p, +, \times)$ is a field when p is a prime, $+$ is integer addition mod p , and \times is integer multiplication mod p
- $(\mathbb{Q}, +, \times)$ is a field
- The first example mentioned above is a *finite* field, and the second example mentioned above is an *infinite* field
- Size of any finite field is p^n , where p is a prime and n is a natural number
- Additional Reading: If interested, read about how the fields of size p^2, p^3, \dots are defined