Cryptography

Lecture 17

Announcements

- HW4 due 4/10/24

Agenda

- More Number Theory!

Other groups over the integers

- We will be interested mainly in multiplicative groups over the integers, since there are computational problems believed to be hard over such groups.
  - Such hard problems are the basis of number-theoretic cryptography.
- Group operation is multiplication mod p, instead of addition mod p.

Multiplication mod p

Example:

$$3 \cdot 8 \; mod \; 13 \equiv 24 \; mod \; 13 \equiv 11 \; mod \; 13.$$

Multiplicative Groups

Is Z_p a group with respect to multiplication mod p?

- Closure—YES
- Identity—YES (1 instead of 0)
- Associativity—YES
- Inverse—NO
  - 0 has no inverse since there is no integer a such that $0 \cdot a \equiv 1 \; mod \; p$.

Multiplicative Group

For p prime, define $Z_p^* = \{1, \dots, p-1\}$ with operation multiplication mod p.

We will see that Z_p^* is indeed a multiplicative group!

To prove that Z_p^* is a multiplicative group, it is sufficient to prove that every element has a multiplicative inverse (since we have already argued that all other properties of a group are satisfied).

This is highly non-trivial, we will see how to prove it using the Euclidean Algorithm.

Inefficient method of finding inverses mod p

Example: Multiplicative inverse of $9\ mod\ 11$.

$$9 \cdot 1 \equiv 9\ mod\ 11$$
$$9 \cdot 2 \equiv 18 \equiv 7\ mod\ 11$$
$$9 \cdot 3 \equiv 27 \equiv 5\ mod\ 11$$
$$9 \cdot 4 \equiv 36 \equiv 3\ mod\ 11$$
$$9 \cdot 5 \equiv 45 \equiv 1\ mod\ 11$$

What is the time complexity?

Brute force search. In the worst case must try all 10 numbers in Z^*_{11} to find the inverse.

This is exponential time! Why? Inputs to the algorithm are $(9,11)$. The length of the input is the length of the binary representation of $(9,11)$. This means that input size is approx. $\log_2 11$ while the runtime is approx. $2^{\log_2 11} = 11$. The runtime is exponential in the input length.

Fortunately, there is an efficient algorithm for computing inverses.

Euclidean Algorithm

Theorem: Let a, p be positive integers. Then there exist integers X, Y such that $Xa + Yp = \gcd(a, p)$.

Given a, p, the Euclidean algorithm can be used to compute $\gcd(a, p)$ in polynomial time. The extended Euclidean algorithm can be used to compute X, Y in polynomial time.

Proving Z_p^* is a multiplicative group

In the following we prove that every element in Z^*_p has a multiplicative inverse when p is prime. This is sufficient to prove that Z_p^* is a multiplicative group.

Proof. Let $a \in Z_p^*$. Then $\gcd(a, p) = 1$, since p is prime.

By the Euclidean Algorithm, we can find integers X, Y such that $aX + pY = \gcd(a, p) = 1$.

Rearranging terms, we get that $pY = (aX - 1)$ and so $p \mid (aX - 1)$.

By definition of modulo, this implies that $aX \equiv 1 \, mod \, p$.

By definition of inverse, this implies that X is the multiplicative inverse of a.

Note: By above, the extended Euclidean algorithm gives us a way to compute the multiplicative inverse in polynomial time.

Extended Euclidean Algorithm Example

Find: X, Y such that $9X + 23Y = \gcd(9,23) = 1$.

$$23 = 2 \cdot 9 + 5$$
$$9 = 1 \cdot 5 + 4$$
$$5 = 1 \cdot 4 + 1$$
$$4 = 4 \cdot 1 + 0$$

$$1 = 5 - 1 \cdot 4$$
$$1 = 5 - 1 \cdot (9 - 1 \cdot 5)$$
$$1 = (23 - 2 \cdot 9) - \big(9 - (23 - 2 \cdot 9)\big)$$
$$1 = 2 \cdot 23 - 5 \cdot 9$$

$-5 = 18\ mod\ 23$ is the multiplicative inverse of $9\ mod\ 23$.

Time Complexity of Euclidean Algorithm

When finding $\gcd(a, b)$, the "b" value gets halved every two rounds.

Why?

Time complexity: $2\log(b)$.

This is polynomial in the length of the input.

Why?

Modular Exponentiation

Modular Exponentiation

Is the following algorithm efficient (i.e. poly-time)?

ModExp(a, m, N) //computes $a^m \ mod \ N$

Set $temp := 1$

For $i = 1$ to m

Set $temp := (temp \cdot a) mod \ N$

return $temp$;

Modular Exponentiation

Is the following algorithm efficient (i.e. poly-time)?

ModExp(a, m, N) //computes $a^m \ mod \ N$

Set $temp := 1$

For $i = 1$ to m

Set $temp := (temp \cdot a) mod \ N$

return $temp$;

No—the run time is $O(m)$. m can be on the order of N. This means that the runtime is on the order of $O(N)$, while to be efficient it must be on the order of $O(\log N)$.

Modular Exponentiation

We can obtain an efficient algorithm via "repeated squaring."

ModExp(a, m, N) //computes $a^m \ mod \ N$, where $m = m_{n-1}m_{n-2} \cdots m_1 m_0$ are the bits of m.

Set $s \coloneqq a$

Set $temp \coloneqq 1$

For $i = 0$ to $n - 1$

If $m_i = 1$

Set $temp \coloneqq (temp \cdot s) mod \ N$

Set $s \coloneqq s^2 \ mod \ N$

return $temp$;

This is clearly efficient since the loop runs for n iterations, where $n = \log_2 m$.

Modular Exponentiation

Why does it work?

$$m = \sum_{i=0}^{n-1} m_i \cdot 2^i$$

Consider $a^m = a^{\sum_{i=0}^{n-1} m_i \cdot 2^i} = \prod_{i=0}^{n-1} a^{m_i \cdot 2^i}$.

In the efficient algorithm:

s values are precomputations of a^{2^i}, for $i = 0\ to\ n-1$ (this is the "repeated squaring" part since $a^{2^i} = (a^{2^{i-1}})^2$).

If $m_i = 1$, we multiply in the corresponding s-value.

If $m_i = 0$, then $a^{m_i \cdot 2^i} = a^0 = 1$ and so we skip the multiplication step.

Getting Back to Z_p^*

Group $Z_p^* = \{1, \ldots, p-1\}$ operation: multiplication modulo p.

Order of a finite group is the number of elements in the group.

Order of Z_p^* is $p - 1$.

Fermat's Little Theorem

Theorem: For prime p, integer a:

$$a^p \equiv a \; mod \; p.$$

Corollary: For prime p and a such that $(a, p) = 1$:

$$a^{p-1} \equiv 1 \; mod \; p$$

Generalized Theorem

Theorem: Let G be a finite group with $m = |G|$, the order of the group. Then for any element $g \in G, g^m = 1$.

Corollary of Fermat's Little Theorem is a special case of the above when G is the multiplicative group Z^*_p and p is prime.

Multiplicative Groups Mod N

- What about multiplicative groups modulo N, where N is composite?
- Which numbers $\{1, \dots, N-1\}$ have multiplicative inverses $mod\ N$?
  - a such that $\gcd(a, N) = 1$ has multiplicative inverse by Extended Euclidean Algorithm.
  - a such that $\gcd(a, N) > 1$ does not, since $\gcd(a, N)$ is the smallest positive integer that can be written in the form $Xa + YN$ for integer X, Y.
- Define $Z_N^* \coloneqq \{a \in \{1, \dots, N-1\} | \gcd(a, N) = 1\}$.
- Z_N^* is an abelian, multiplicative group.
  - Why does closure hold?

Order of Multiplicative Groups Mod N

- What is the order of Z_N^*?
- This has a name. The order of Z_N^* is the quantity $\phi(N)$, where ϕ is known as the Euler totient function or Euler phi function.
- Assume $N = p \cdot q$, where p, q are distinct primes.
  - $\phi(N) = N - p - q + 1 = p \cdot q - p - 1 + 1 = (p-1)(q-1)$.
  - Why?

Order of Multiplicative Groups Mod N

General Formula:

Theorem: Let $N = \prod_i p_i^{e_i}$ where the $\{p_i\}$ are distinct primes and $e_i \geq 1$. Then

$$\phi(N) = \prod_i p_i^{e_i - 1}(p_i - 1).$$

Another Special Case of Generalized Theorem

Corollary of generalized theorem:

For a such that $\gcd(a, N) = 1$:

$$a^{\phi(N)} \equiv 1 \, mod \, N.$$

Another Useful Theorem

Theorem: Let G be a finite group with $m = |G| > 1$. Then for any $g \in G$ and any integer x, we have

$$g^x = g^{x \bmod m}.$$

Proof: We write $x = a \cdot m + b$, where a is an integer and $b \equiv x \bmod m$.

- $g^x = g^{a \cdot m + b} = (g^m)^a \cdot g^b$
- By "generalized theorem" we have that

$$(g^m)^a \cdot g^b = 1^a \cdot g^b = g^b = g^{x \bmod m}.$$

An Example:

Compute $3^{25}\ mod\ 35$ by hand.

$$\phi(35) = \phi(5 \cdot 7) = (5-1)(7-1) = 24$$

$$3^{25} \equiv 3^{25\ mod\ 24}\ mod\ 35 \equiv 3^{1}\ mod\ 35$$

$$\equiv 3\ mod\ 35.$$