Prime Number Theory

Based on *Cryptography Engineering* by Schneier, Ferguson, Kohno, Chapter 10

Lecture notes of Alexander Wood
awood@jjay.cuny.edu

John Jay College of Criminal Justice

Primes

Before we can discuss public key cryptography we need some mathematical background. Today we discuss **divisibility and primes**.

Divisibility

A number a is a **divisor** of b, denoted $a|b$, if you can divide b by a without a remainder.

Divisibility

A number a is a **divisor** of b, denoted $a|b$, if you can divide b by a without a remainder.

Examples:

- $7|35$
- $1|13$
- $15|15$
- $12 \not| 19$ (here, $\not|$ denotes "does not divide")

Prime Numbers

We say a number is **prime** if it has exactly two positive divisors, one and itself.

The first primes are $2, 3, 5, 7, 11, 13, 17, 19, 23, \dots$. A number which is not prime is called **composite**.

Prime Numbers: Fun Facts

The number 1 is neither prime nor composite.

The number 2 is the only even prime. Why is this?

Divisibility Lemma

Lemma

If $a|b$ and $b|c$, then $a|c$.

Proof.

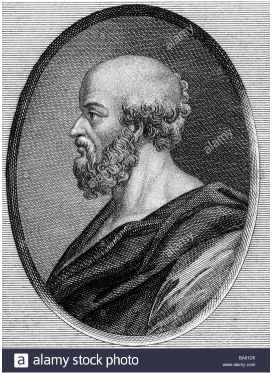
If $a|b$ then $b = ak$ for some integer k. Furthermore, $b|c$ implies that $c = b\ell$ for some integer ℓ. Thus, $c = b\ell = (ak)\ell = a(k\ell)$. This means that a is a divisor of c; hence, $a|c$. $\square$

Mathematical Proofs

What we saw in the previous slide is a **mathematical proof**. It starts with a statement of facts, or assumptions, and proceeds logically step-by-step until the desired conclusion has been reached.

The Sieve of Eratosthenes

Eratosthenes, a friend of Archimedes, was the first person to accuately measure the diameter of the earth 2000 years ago. He created an algorithm now known as the **Sieve of Eratosthenes** which generates prime numbers. Even today it is an excellent algorithm for generating primes.



The Sieve of Eratosthenes

To find all of the prime numbers from 1 up to some integer n, carry out the following:

(1) Write a list of all integers from 2 through n.

(2) Start with $p = 2$, the smallest prime.

(3) Cross out every multiple of two in the list.

(4) Return to the first uncrossed number in your list. Cross out all of its multiples.

(5) Repeat until you reach the end of the list.

Infinity of Primes

What is widely regarded as one of the most beautiful proofs in history was written by the Greek mathematician Euclid, showing that there are an infinite number of primes.

It uses a method called **reductio ad absurdum** or **proof by contradiction**, where you assume the negation of your conclusion is true and show that then some portion of your premise must be false.

Infinity of Primes

Theorem (Euclid)

There are infinitely many prime numbers.

Proof.

Assume to the contrary that there are only a finite number of primes. Because the list of prime numbers is finite, they are enumerable, meaning we can list them. Say that

$$p_1, p_2, \ldots, p_k$$

constitutes a complete list of prime numbers.

Continued on next slide. □

Infinity of Primes

Proof.

Continued.

Consider the number

$$n = p_1 p_2 \cdots p_k + 1.$$

This number cannot be prime, due to our assumption that we have already listed all of the primes. Let d be the smallest divisor of n which is not 1. None of the primes in our list can be equal to d, because n divided by p_i has a remainder of 1 for any $1 \leq i \leq k$. Therefore, d must be a prime value greater than p_k, a contradiction. $\square$

Prime Modulus Arithemetic

Computing values modulo some prime p is the the main application of prime numbers in cryptography. We compute values **modulo some prime** p if we use only the numbers

$$0, 1, 2, \ldots, p - 1.$$

Perform all computations as you normally would, then **reduce** your answer modulo p, denoted $\pmod{p}$.

Modular Arithmetic

For example,

$$(25 \mod 7) = 4.$$

When you divide 25 by 7, you end up with a multiple of 4, which is its value reduced modulo 7. You can also compute modular reductions by adding/subtracting multiples of the modulus:

$$25 = 25 - 21 = 4 \pmod 7$$

$$100 = 100 - 70 = 30 = 30 - 28 = 2 \pmod 7$$

$$-5 = -5 + 17 = 12 \pmod{17}$$

Note that our final answer must always be a positive value x where $0 \leq x < p$.

Multiplication

Multiplication is carried out similarly; first multiply as you normally would, then reduce modulo p.

- $(5 \cdot 3 \mod 7) = 1$
- $6 \cdot 9 = 54 = 54 - 43 = 11 \pmod{43}$
- $5 \cdot (-2) = -10 = -10 + 17 = 7 \pmod{17}$
- $((-7) \cdot 3 \mod 19) = 17$

Multiplication

Theorem

For any prime p, $(p-1)^2 = 1 \pmod p$.

Proof.

Observe:

$$\begin{aligned}(p-1)^2 &= p^2 - 2p + 1 \pmod p \\ &= 1 \pmod p\end{aligned}$$

$\square$

Exponentiation

We exponentiate again by computing as we regularly would, then reducing modulo p.

- $4^2 = 8 = 1 \pmod 7$
- $4^6 = 4096 = 1 \pmod 7$
- $19^96 = 1 \pmod{97}$
- $8^8 = 1 \pmod 9$

Are we noticing a pattern?

Fermat's Little Theorem

Fermat's Little Theorem makes reduction of exponents modulo a prime much more quick.

Theorem

For a prime value p and any integer a,

$$a^{p-1} = 1 \pmod{p}.$$

Equivalently,

$$a^{p} = a \pmod{p}.$$

Fermat's Little Theorem

For example, say we wish to compute $(2^{1939} \mod 89)$.
Observe that $1939 = (22)(88) + 3$. Therefore,

$$\begin{aligned}
2^{1993} \mod 89 &= 2^{(22)(89)+3} \\
&= (2^{22})^{88} \cdot 2^{3} \\
&= 2^{3} \\
&= 8 \pmod{89}
\end{aligned}$$

Useful Tips For Computations Modulo p

- You can add/subtract any multiple of p from your number.
- Results should always lie within the range $0, 1, \ldots, p - 1$.
- Any value raised to the $(p - 1)$th power reduces to 1 modulo p.

Groups and Finite Fields

Mathematicians call the set of numbers modulo a prime p a **finite field** and often refer to it as the “mod p” field. Different notations for this field you might encounter include $\mathbb{Z}_p$, $\mathrm{GF}(p)$, or $\mathbb{Z}/p\mathbb{Z}$.

The branch of mathematics called **abstract algebra** has large areas devoted to the study of prime fields.

Groups

A **group** is a mathematical structure consisting of a set of numbers together with some binary operation. A group is **closed under addition**, contains an **identity** element, and every element has an **inverse.**

Group theory is another wide and important branch of abstract algebra. This class requires only some very basic introductory group theory.

Additive Group Modulo p

In this class, we do not need to make this too complicated! We consider the group $\mathbb{Z}_p$, which is the set of numbers $0, 1, 2, \ldots, p-1$, together with addition. Denote this group $(\mathbb{Z}_p, +)$, the **additive group modulo p**.

In the additive case we could also consider a group $(Z_n, +)$, where n is a composite modulus.

The Group $\mathbb{Z}_p$

Let's verify the group properties for $\mathbb{Z}_p$, also denoted $(\mathbb{Z}_p, +)$.

- *Closed under addition?* Adding any two elements in $\mathbb{Z}_p$ and reducing modulo p yields an element in $\mathbb{Z}_p$. Therefore, $\mathbb{Z}_p$ is **closed under addition.**
- *Contains identity?* The **identity element** in $(\mathbb{Z}_p, +)$ is 0. This means that for any a in $\mathbb{Z}_p$, we have that

$$a + 0 = 0 + a = a \pmod{p}.$$

- *Contains inverses?* The **inverse** of an element a in $(\mathbb{Z}_p, +)$ is $(-a \mod p)$. This means that:

$$a + (-a) = (-a) + a = 0 \pmod{p}.$$

Multiplicative Group Modulo p

The **multiplicative group modulo** p is denoted $\mathbb{Z}_p^*$. This is the group composed of the set of numbers $1, 2, \ldots, p-1$ with the operation of addition modulo p. Note that 0 is not included here.

Note that Z_p^* is a group if and only if p is prime.

The Group $(\mathbb{Z}_p, +)$

Let's verify that $\mathbb{Z}_p^*$ is a group.

- *Closed under addition?* Multiplying any two elements in $\mathbb{Z}_p^*$ and reducing modulo p yields an element in $\mathbb{Z}_p^*$. Therefore, $\mathbb{Z}_p^*$ is **closed under multiplication.**
- *Contains identity?* The **identity element** in $\mathbb{Z}_p^*$ is 1. This means that for any a in $\mathbb{Z}_p^*$, we have that

$$a \cdot 1 = 1 \cdot a = a \pmod{p}.$$

- *Contains inverses?* The **inverse** of an element a in $\mathbb{Z}_p^*$ is some value b such that This means that:

$$ab = ba = 1 \pmod{p}.$$

  Some number b satisfying this property always exists when p is prime; we skip the proof of this for now.

Subgroups

A **subgroup** of a group consists of some subset of elements from the group which form a group themselves.

Subgroups Examples

Consider the group $(Z_8, +)$. This group contains several subgroups. Subgroups include $\mathbb{Z}_2$ and $\mathbb{Z}_4$.

This grop has a subgroup consisting of the elements $\{0, 4\}$ under addition, as well as the subgroup consisting of the elements $\{0, 2, 4, 6\}$ under addition. How can we convince ourselves that these are subgroups?

Subgroups Examples

The multiplicative group $\mathbb{Z}_p^*$ has subgroups as well. Consider $\mathbb{Z}_7^*$ which consists of $\{1, 2, 3, 4, 5, 6\}$ under multiplication.
Subgroups include:

- $\{1, 6\}$
- $\{1, 2, 4\}$

How can we check that these are, in fact, subgroups? Does $\{1, 5\}$ form a subgroup under multiplication modulo 7? Why or why not?

Subgroups in Cryptography

Many cryptographic protocols make use of subgroups. We've discussed addition, subtraction, multiplication, and exponentiation modulo n.

The two remaining properties we must discuss are **division** and **logarithms** modulo n.

Division: The GCD Algorithm

The **greatest common divisor (GCD)** of two numbers a and b is the largest value k such that $k|a$ and $k|b$. This is often denoted $(a, b) = k$.

For example, the GCD of 24 and 30 is 6, also written $(24, 30) = 6$.

Division: The GCD Algorithm

Euclid wrote an algorithm which computes the GCD of two numbers. This algorithm is still used today and has important applications in cryptography.

Division: The GCD Algorithm

Input: Positive integers a and b.
Output: k, where $(a, b) = k$.

(1) While $a \neq 0$:
    - $(a, b) := (b \pmod a, a)$

(2) Return b.

The GCD Algoirthm

For example, say our inputs are 24 and 30. The algorithm runs as follows:

Loop 0: $(a, b) = (24, 30)$

Loop 1: $(a, b) = (6, 24)$

Loop 2: $(a, b) = (0, 6)$

And the algorithm returns 6.

The GCD Algoirthm

For example, say our inputs are 12 and 45. The algorithm runs as follows:

Loop 0: $(a, b) = (12, 45)$

Loop 1: $(a, b) = (9, 12)$

Loop 2: $(a, b) = (3, 9)$

Loop 3: $(a, b) = (0, 3)$

And the algorithm returns 3.

Coding Exercises: Euclid’s GCD Algorithm

Write Python code which carries oud Euclid’s GCD algorithm.

Least Common Multiples

The **Least Common Multiple (LCM)** of two numbers is the smallest number that is a multiple of both of them. Find the LCM of *a* and *b* with the following formula:

$$\mathrm{LCM}(a, b) = \frac{ab}{(a, b)}$$

Next Class

Next class we will look at **Euclid's Extended Algorithm**, how to divide and find logarithms modulo p, as well as theorems describing how to find all of the subgroups of $\mathbb{Z}_p^*$.

On Wednesday we will have a quiz over all of this information.

References

- *Cryptography Engineering* by Schneier, Ferguson, Kohno, Chapter 10