Lecture 24: Encryption and number theory

DANIEL WELLER
THURSDAY, APRIL 18, 2019
Agenda

Prime numbers and divisibility
Congruences
Euler’s theorem
Diffie-Hellman key exchange

The traditional clock is a good example of modulo-12 congruences (0 = 12).
To describe modern encryption algorithms like Diffie-Hellman key exchange, we will need to discuss some number theory. In particular, we will explore the factorization of integers, which continues to be a challenging and exciting domain of mathematics.

Recall: integers are discrete numbers we use to count, both positive and negative: -5, 0, 2, etc.

First, we need to consider how we define long division, as a process that produces two results:
  ◦ Quotient
  ◦ Remainder
Defining division

Suppose we have two integers $a$ and $b$. Furthermore, let $b > 0$ (a natural number). Then, there exist *unique* integers $q$ (quotient) and $r$ (remainder) such that:

- $a = bq + r$
- $0 \leq r < b$

We won’t review the proof of the existence or uniqueness of such numbers.

Examples: $a = 34$, $b = 7 \rightarrow q = \_\_\_\_\_\_\_ \quad r = \_\_\_\_\_\_\_ \quad (34 = 7*4 + 6)$

$a = -18$, $b = 5 \rightarrow q = \_\_\_\_\_\_\_ \quad r = \_\_\_\_\_\_\_ \quad (-18 = 5*-4 + 2)$
Division and divisibility

Based on this definition of long division, quotient, and remainder, we can construct a whole theory of integers and factorization. To begin, let us define a relatively simple notion of divisibility:

We say $d$ divides $n$ and write $d \mid n$ if the remainder of $n$ divided by $d$ equals zero. This means that there exists some integer (the quotient) $q$ such that $n = d \times q$. In this case, we also say that $n$ is a multiple of $d$, and that $d$ is a divisor or factor of $n$.

If the remainder is nonzero, then $d$ does not divide $n$, and we write $d \nmid n$.

Example: Does 3 divide 24? How about 7 dividing 18?
Properties of division

As expected, division (and dividing) have lots of useful properties. Here are some of them:

- $d|n$ (a number divides itself)

- $d|n \land n|m \Rightarrow d|m$ (transitivity)

- $d|n \land d|m \Rightarrow d|am + bn$ (linearity)

- $d|n \Rightarrow ad|an$ (scaling both sides)

- The opposite is also true if $a$ is nonzero: $ad|an \Rightarrow d|n$
Properties of division

Here are some other important properties:

- $1|n$ (1 is a divisor of any number)
- $n|0$ (any number is a divisor of zero)
- $0|n \Rightarrow n = 0$ (zero is not a divisor of any other number)
- $d|n \Rightarrow |d| \leq |n|$ (divisors are not any bigger than the number in magnitude)
- $d|n \& n|d \Rightarrow |d| = |n|$ (if two numbers divide each other, they are the same in magnitude)
- $d|n \& d \neq 0 \Rightarrow \frac{n}{d}|n$ (if $d$ is a factor of $n$, so is $n/d$)
Prime numbers

Prime numbers are special, in the sense that they are the basic building blocks of the integers. Let’s define them:

An integer $n > 1$ is *prime* if its only positive divisors are 1 and itself. Otherwise, the number $n > 1$ is *composite*. Notice that we do not consider 1 to be either prime or composite.

Here are some prime numbers: 2, 3, 5, 7, 11, 13, ...

Is 261 prime? How about 73?
Fundamental theorem of arithmetic

While we provide it without proof, the following is one of the most important theorems in number theory as it underlies the concept of factorization:

Every integer $n > 1$ can be expressed as the product of prime factors in only one way, up to the order of the factors.

Example: $60 = 2^2 \times 3 \times 5$

How about 70?
How many primes are there?

Euler proved that there are infinitely many primes.

We’re not going to focus on the proof, but it is important to note that this also means that there are arbitrarily large prime numbers. So if we need a prime greater than $2^{64}$, we can find one given enough time.

Much of the security of modern encryption algorithms stems from working with incredibly large prime numbers. If we could not guarantee their existence, these algorithms would not work!
Common divisors and multiples

So far, we have considered division and factorization with respect to a single number. Where things get really interesting is when we consider divisors/factors of multiple numbers.

A *common divisor* $d$ of both $m$ and $n$ satisfies both $d \mid m$ and $d \mid n$. The greatest such number is called the *greatest common divisor* (gcd) of $m$ and $n$.

A *common multiple* $x$ of both $m$ and $n$ satisfies both $m \mid x$ and $n \mid x$. The smallest such number is called the *least common multiple* (lcm) of both $m$ and $n$.

Given the $d = \text{gcd}(m,n)$, we have $d \geq 0$, and if $e$ is also a common divisor of $m$, $n$, then $e \mid d$. 
Examples

What is the gcd(3,7)? These are both prime numbers, and we would expect the gcd to be 1. In general if gcd(m,n) = 1, even if m or n are not prime, we say they are relatively prime. Another example: 14 and 15

What about gcd(60,42)? What are the other common divisors of 60 and 42?
Examples

What are common multiples of 6 and 8?

What is the least common multiple?

How about for 15 and 16? What is true about the lcm for two relatively prime numbers?

Observation: $gcd(m,n) \times lcm(m,n) = |mn|$
Besides factorization, congruences are another important element from number theory that we need to describe modern encryption methods.

Suppose we have two numbers $a$ and $b$. We say they are congruent and write $a \equiv b \pmod{N}$ if the remainders of $N$ divided by $a$ and $N$ divided by $b$ are the same.

This means that $(a - b)$ is a multiple of $N$.

Example: $13 \equiv 1 \pmod{12}$
Examples

Some more congruences:

\[ 3 \equiv 8 \equiv -2 \ (mod \ 5) \]
\[ 29 \equiv 5 \ (mod \ 12) \]
\[ -29 \equiv 7 \ (mod \ 12) \]
\[ 4 \equiv -1 \ (mod \ 5) \]
\[ 4^2 \equiv 1 \ (mod \ 5) \]

Note: the congruence relation kind of looks like an equal sign, and they are similar, but they are not the same thing.
Congruence properties

This similarity with equality extends to simple (e.g., linear) manipulations of the congruence “equation”:

1) \( ax + \alpha y \equiv bx + \beta y \) (mod \( m \)) for all integers \( x \) and \( y \)

2) \( a\alpha \equiv b\beta \) (mod \( m \))

3) \( a^n \equiv b^n \) (mod \( m \)) for every positive integer \( n \).

Example: \((m = 7)\)

\[
\begin{align*}
5 & \equiv 5 \\
5^2 = 25 & \equiv 4 \\
5^3 = 5 \times 5^2 & \equiv 5 \times 4 = 20 \equiv 6
\end{align*}
\]
Your turn

Find $7^9 \ (mod\ 11)$

Find $5^6 \ (mod\ 7)$
Euler’s totient function

For $n \geq 1$, the Euler totient function $\varphi(n)$ is the number of positive numbers less than or equal to $n$ that are relatively prime to it.

Example: $\varphi(1) = 1, \varphi(2) = 1, \varphi(3) = 2, \varphi(4) = 2, \varphi(5) = 4, \varphi(6) = 2$

Properties: if $n$ is prime, then $\varphi(n) = n - 1$. For instance, if $n = 5$, $\varphi(5) = 5 - 1 = 4$

If $m$ and $n$ are both prime, then $\varphi(mn) = (m - 1)(n - 1)$

Example: what is $\varphi(15)$?
Euler’s totient function

If a and N are relatively prime, then $a^{\phi(N)} \equiv 1 \pmod{N}$

Example: $N = 5$ is relatively prime to $\{1,2,3,4\}$, so $\varphi(5) = 4$, and $a^4 \equiv 1 \pmod{5}$ for $a = 1, 2, 3, 4$. Check:

\[
\begin{align*}
1^4 &= 1 \equiv 1 \pmod{5} \\
2^4 &= 16 \equiv 1 \pmod{5} \\
3^4 &= 81 \equiv 1 \pmod{5} \\
4^4 &= 256 \equiv 1 \pmod{5}
\end{align*}
\]
Back to encryption

Now, with divisibility, factorization, and congruence, we can turn back to encryption and discuss how we use these ideas to secure information.

Some observations:

◦ Computing factors of a large number requires lots of computation.

◦ This is something that gets easier with quantum computing – Shor’s algorithm can use quantum bits (qubits) to factor large numbers much more rapidly. There is a huge concern that current encryption methods based on factorization no longer will be secure with the advent of quantum computers.

◦ Generating large prime numbers (or verifying a large number is prime) is also hard.
Secret key generation

A “one way function” is one whose output can be computed easily, but is difficult to invert. For instance, if we have $g$, $A$, and prime $p$, then computing $g^A \mod p$ is easy via congruences.

- However, finding $A$ from the result is hard.
- This is called the discrete logarithm problem.

Example: $5^{12} \mod 23 = ?$

How to find $5^? \mod 23 = \text{(result)}$

There are only a few possible function outputs (23 of them), so it probably will take just a few tries to find a suitable exponent. However, for a large prime number, this approach becomes problematic.
Recall the three pass protocol

We want to share information in a secure way between users A and B.

Three pass protocol:
- User A encrypts data x with key A: $x \rightarrow y$
- User B encrypts encrypted data y with key B: $y \rightarrow z$
- User A decrypts encrypted data z with key A: $z \rightarrow u$
- User B decrypts result u with key B: $u \rightarrow x$

We want to ensure that the decryptions can be done in the same order, not reverse order. Can this be done?
Diffie-Hellman key exchange

Let’s suppose Alice (A) and Bob (B) want to exchange keys secretly, but publicly (no armored trucks driving around!), how can they use number theory to do this?

Let’s say Alice and Bob share some public info (e.g., g = 7 and p = 11). These are fixed.

Then, suppose Alice has a secret number (A = 3). Bob also has another secret (B = 6).

Alice wants to tell Bob her message A = 3; Bob also wants to tell Alice his message B = 6.
Diffie-Hellman key exchange

To start, Alice doesn’t transmit A directly. Instead, she computes

$$\alpha = g^A \pmod{p}$$

At the same time, Bob computes a similar expression for B:

$$\beta = g^B \pmod{p}$$

Example:

$$\alpha \equiv 7^3 \equiv 7 \times 49 \equiv 35 \equiv 2$$

$$\beta \equiv 7^6 \equiv (7^2)^3 \equiv 5^3 \equiv 5 \times 25 \equiv 15 \equiv 4$$
Diffie-Hellman key exchange

From these, Alice and Bob exchange $\alpha$ and $\beta$. These, but not A and B, would be observable by an eavesdropper (Eve).

Alice computes $K = \beta^A \pmod{p}$, and Bob computes $K = \alpha^B \pmod{p}$. These keys are the same (and contain information from A and B), so they have succeeded in sharing a common key that they can use to share additional information.

$$4^3 \equiv 4 \times 16 \equiv 20 \equiv 9 \pmod{11}$$
$$2^6 \equiv 64 \equiv 9 \pmod{11}$$

What about Eve? Since Eve doesn’t know A or B, Eve cannot compute $K$ from the public information $g, p, \alpha, \text{ and } \beta$. 
Diffie-Hellman key exchange

Correctness: do Alice and Bob compute the same key?

\[ \alpha = g^A \mod p \quad \beta = g^B \mod p \]

Alice: \( K = \beta^A \mod p \)

\[ \beta \equiv g^B \pmod{p} \Rightarrow \beta^A \equiv g^{AB} \pmod{p} \]

Bob: \( K = \alpha^B \mod p \)

\[ \alpha \equiv g^A \pmod{p} \Rightarrow \alpha^B \equiv g^{AB} \pmod{p} \]

Therefore,

\[ K = \beta^A \mod p = \alpha^B \mod p \]
Your turn

g = 5, p = 17, A = 4, B = 3

Compute $\alpha$, $\beta$, and $K$: 
Diffie-Hellman key exchange

This key exchange method will allow the sharing of a secret key that it generates on the fly, but it is not so useful for transmitting specific pieces of information. For that, we will turn to asymmetric encryption techniques next time.
Announcements

Next class: Public key encryption