Advanced 1A

Intro to Cryptology

Prepared by Mark on April 21, 2022

Part 1: The Euclidean Algorithm

Definition 1:
The *greatest common divisor* of a and b is the greatest integer that divides both a and b.
We denote this number with $\gcd(a, b)$. For example, $\gcd(45, 60) = 15$.

Theorem 2: The Division Algorithm
Given two integers a, b, we can find two integers q, r, where $0 \leq r < b$ and $a = qb + r$.
In other words, we can divide a by b to get q remainder r.

Theorem 3:
For any integers a, b, c,
$\gcd(ac + b, a) = \gcd(a, b)$

Problem 4:
Find $\gcd(20, 14)$ by hand.

Problem 5:
Using the theorems above, detail an algorithm for finding $\gcd(a, b)$.
Then, compute $\gcd(1610, 207)$ by hand.
Have an instructor check your work before moving on.

Problem 6: Divide and Conquer
If we are given a, b, c, when can we find u, v that satisfy $au + bv = c$?

Part A:
Show that if we find a solution (u, v) to $au + bv = \gcd(a, b)$, we can easily find all other solutions to $au + bv = c$.

Part B:
Using the output of your algorithm* from Problem 5,
- find a pair (u, v) that satisfies $20u + 14v = \gcd(20, 14)$
- find a pair (u, v) that satisfies $541u + 34v = \gcd(541, 34)$

For which numbers c can we find a (u, v) so that $541u + 34v = c$?
For every such c, what are u and v?

*Your solution to Problem 5 is called the *Euclidean Algorithm*

Part 2: Modular Arithmetic

Definition 7:
We say that a, b are equivalent mod m if m divides $a - b$.
If a is equivalent to b mod m, we write $a \equiv b \pmod{m}$.
You can think of b as the remainder of $a \div m$:

$32 \equiv 2 \pmod{6}$
$4 \equiv 4 \pmod{6}$
$-2 \equiv 4 \pmod{6}$

Problem 8:
Complete the following:

$87 \equiv ? \pmod{12}$ (Your answer should be between 0 and 12)
$13 \equiv 2 \pmod{?}$
$? \equiv 1 \pmod{9}$

Definition 9:
The inverse of a mod m is an integer $a^\star$ so that
$a \times a^\star \equiv 1 \pmod{m}$.
Note that not every a has an inverse mod m.

Theorem 10:
a has an inverse mod m iff $\gcd(a, m) = 1$
The proof of this theorem is left as a challenge problem.

Problem 11: Déjà vu?
Find the inverse of $20 \pmod{14}$, if one exists.
Find the inverse of $34 \pmod{541}$, if one exists.

Problem 12:
In general, how can we find the inverse of $a \pmod{p}$?
(Assume p is prime.)

Part 3: Symmetric Cryptosystems

Definition 13:
The goal of cryptography is to establish private communication between two parties over a public channel. The rest of this handout tries to achieve this goal, using the tools we've developed in the last two sections.

In this handout, a "symmetric cryptosystem" consists of the following:
- A public prime number p (Ideally, a *big* prime number).
- k, a secret key that is shared between both parties. This is NOT public.
- $E_k(m) = c$, a function that uses key k to encrypt message m into a ciphertext c.
- $D_k(c) = m$, a function that uses key k to decrypt a ciphertext c into message m.

- Of course, $D_k(E_k(m)) = m$.

We have a good reason for picking a prime p. A prime base guarantees that every* integer has an inverse mod p. Review Theorem 10 and convince yourself that this is true.

We'll assume that the secret key k has been shared beforehand. How such a k is created is beyond the scope of this handout, but those that are curious may look up "Diffie-Hellman Key Exchange" (Computerphile offers a pretty good introduction).

One may wonder why we care about secretly exchanging numbers. Those of you with experience in computing may have an answer: any information—text, images, etc—may be represented as a number. For example, we can encode the 26 letters of the alphabet as the numbers $1 - 26$. Such mappings are called "encodings."

Finally, you will notice that the encryption schemes that follow can only take a limited range of inputs. Indeed, even the cyphers in use today have a limited input size. A simple (though possible insecure) way to overcome this limitation is to split the message into "blocks" of a desired size, and encrypt each independently.

*except those $\equiv 0 \pmod{p}$, of course

Problem 14: Multiplication mod p

Consider the cryptosystem where

- p is a prime (for this problem, fix $p = 11$)
- k is an integer
- $E_k(m) = k \times m \pmod{p}$
- $D_k(c) = k^\star \times c \pmod{p}$

Part A:

Encrypt $m = 8$ with $k = 5$.
Decrypt $c = 3$ with $k = 9$.
In other words, find $E_5(8)$ and $D_9(3)$

Part B:

Using this cryptosystem, Nikita sends two messages to Sanjit with the same key.
Looking over Sanjit's shoulder, you find that $E_k(9) = 8$ and $E_k(2) = 3$.
What key did they use?
This is called a *known plaintext attack*. With a good cryptosystem, it will be very difficult to solve this problem.

Part C:

If you only know one message and its corresponding ciphertext, can you find the encryption key?
If you know many ciphertexts encrypted with the same key, can you find the key used to create them?
What range of values can this system effectively encrypt? Justify all answers.

Problem 15: The Affine Cipher
Consider the cryptosystem where

- p is a prime (for this problem, fix $p = 541$)
- $k = (k_1,\ k_2)$ is a tuple of two integers
- $E_k(m) = k_1 \times m + k_2 \pmod{p}$
- $D_k(c) = k_1^\star \times (c - k_2) \pmod{p}$

Part A:
Encrypt $m = 204$ with $k = (34,\ 71)$.
Decrypt $c = 431$ with $k = (34,\ 71)$.

Part B:
Now, let $p = 601$. You know two plaintext-ciphertext pairs:
$(m_1,\ c_1) = (387,\ 324)$
$(m_2,\ c_2) = (491,\ 381)$
Find (k_1, k_2)

Part C:
If you only know one message and its corresponding ciphertext, can you find the encryption key?
If you know many ciphertexts encrypted with the same key, can you find the key used to create them?
What range of values can this system effectively encrypt? Justify all answers.

Part 4: Challenge Problems

Problem 16:
Prove Theorem 10:
a has an inverse mod m iff $\gcd(a, m) = 1$
Hint: To prove an iff statement, prove each direction seperately:
Assume that the left side is true and show that left $\implies$ right,
then do the reverse.

Problem 17:
The Euclidean Algorithm (From Problem 5) can be written as follows:

Assume $a > b$. Set $e_0 = a$ and $e_1 = b$.
Let $e_{n+1} = \text{remainder}(r_{n-1} \div r_n)$
Stop when $e_k = 0$. Then, $\gcd(a, b) = e_{k-1}$.

Let F_n be the n^{th} Fibonacci number. ($F_0 = 0$; $F_1 = 1$; $F_2 = 1$; ...)
Show that if the Euclidean algorithm requires n steps for an input (a, b), then $a \geq F_{n+2}$ and $b \geq F_{n+1}$.
(In other words, show that the longest-running input of a given size is a Fibonacci pair.)

Problem 18: Chinese Remainder Theorem
There are certain things whose number is unknown. If we count them by threes, we have two left over; by fives, we have three left over; and by sevens, two are left over. How many things are there?

Problem 19:
Show that if p is prime, $\binom{p}{i} \equiv 0 \pmod{p}$ for $0 < i < p$.

Problem 20: Fermat's Little Theorem
Show that if p is prime and $a \not\equiv 0 \pmod{p}$, then $a^{p-1} \equiv 1 \pmod{p}$.
You may want to use Problem 19.
Hint: It may be easier to show that $a^p \equiv a \pmod{p}$

[Note on Problem 17] This proof can be used to show that the Euclidean algorithm finishes in logarithmic time, and it is the first practical application of the Fibonacci numbers.