

CSCI 124/297

Discrete Structures II: Modular Arithmetic

Poorvi L. Vora

November 15, 2006

We start with three simple examples from cryptography to illustrate the need for the mathematics of what is known as modern algebra. We then define modular arithmetic.

The cryptographic problem we examine in this set of notes is as follows: Alice (the sender) wishes to communicate secretly with Bob (the receiver). That is, she does not wish anyone else to overhear the conversation, or read the letter. There is a communication channel linking them – such as a phone line, a data link, or a postal service. The channel is typically insecure. Alice encrypts her message using a key known only to Bob, and uses the insecure communication link to send him the encrypted message. Bob uses the key to decrypt the message.

1 The Shift Cipher

One of the first known ciphers is the shift cipher from the times of Julius Caesar. In this cipher, the letters from A through Z are associated with the numbers 0 through 25. The key is a letter too, and is similarly associated with a number. Encryption is performed letter by letter, by adding the key to each letter in the plaintext message to get the corresponding letter in the ciphertext. This is best explained through an example.

Example 1. Encrypt the sentence “IS CLASS OVER YET?” with the shift cipher, key = “E”. Decrypt the result.

Solution: Convert the letters in the sentence to their numerical values:

IS CLASS OVER YET

8 18 2 11 0 18 18 14 21 4 17 24 4 19

To encrypt, add the value of the key: 4:

12 22 6 15 4 22 22 18 25 8 21 2 8 23

Convert back:

m w g p e w w s z i v c i x

To decrypt with knowledge of the key, subtract 4 from each ciphertext symbol:

First convert to numbers:

12 22 6 15 4 22 22 18 25 8 21 2 8 23

Then subtract 4:

8 18 2 11 0 18 18 14 21 4 17 -2=24 4 19

Then convert back to letters:

IS CLASS OVER YET

Example 2: Decrypt the following ciphertext without knowing the key:

mjrmjrmjrtjpmwjvo

Solution: Try every possible key:

nksnksnksukqnkxkwp

oltoltvtlroylxq

pmupmupmuwmspzyr

qnvqnvqnvxntqanzs

rowrowrowyourboat

Thus a brute force attack, of complexity half the size of the keyspace ($\frac{|G|}{2} = 13$ in Example 2), on average, would result in success at determining the key. The shift cipher is hence extremely weak and is not used in real-world cryptography today.

One may wish to make a brute-force attack somewhat more difficult. One step towards doing this is to use a different key with each letter in the string. Another is to use multiplication instead of addition.

2 Vigenere Cipher

In this cipher, the key is a string of symbols that is repeated. It can be as long as the message.

Example 3 Encrypt the plaintext message:

HO HO HO AND A BOTTLE OF RUM

using the Vigenere cipher and the key:

CHRISTMAS

Solution: The message in numbers is:

7 14 7 14 7 14 0 13 3 0 1 14 19 19 11 4 14 17 20 12

The key, in numbers, repeated to form a string as long as the message, is:

2 7 17 8 18 19 12 0 18 2 7 17 8 18 19 12 0 18 2 7 17

The ciphertext in numbers is:

9 21 24 22 25 33=7 12 13 21 2 8 31=5 27=1 37=11 30=4 16 14 35=9 22 19

In letters, the ciphertext is

J V Y W Z H M N V C I

Unfortunately, for this method to be secure, the key needs to be as long as the message. As the key needs to be sent securely from the sender to the receiver, this is often very impractical.

3 The Affine Cipher

For this cipher, a number x is encrypted as $ax + b$ where a and b form the key.

Example 4. Encrypt the sentence “IS CLASS OVER YET?” with the affine cipher, $a = 3$ and $b = 0$. Can you decrypt the result?

Solution: Convert the letters in the sentence to their numerical values:

IS CLASS OVER YET

8 18 2 11 0 18 18 14 21 4 17 24 4 19

Multiply by 3:

24 54=2 6 33=7 0 2 2 ...

Convert back:

y c g h a c c ...

How would one decrypt with knowledge of the key? If one divided by 3, that would not always give an integer. In this case, what does it mean to divide by 3? To understand this, we need to understand better the mathematical structure behind the numbers we’ve assigned to the alphabet.

4 Modular Arithmetic

In assigning numbers to the alphabet, it seems that we are saying that 2 is the same as 28 which is the same as 54. That is, we are saying that two numbers are the same if their difference is divisible by 26. We need not restrict ourselves to 26, and can generalize this to any number m .

Definition 1: $a \equiv b \pmod{m}$ if and only if $m|(b - a)$. If $a \equiv b \pmod{m}$, we say “ a is congruent to b modulo m ”.

For example, $3 \equiv 10 \pmod{7}$, $1 \equiv 3 \pmod{2}$, $5 \equiv -4 \pmod{9}$, etc.

Recall the definition of equivalence relations in CS 123. An equivalence was something like an equality, but not quite an equality. In fact, an equivalence is exactly what we’ve been using.

Theorem: $\equiv \pmod{m}$ is an equivalence relation

Proof: It is:

- *Reflexive*
 $a \equiv a$ because $m|(a - a) = 0$.
- *Symmetric*

$$\begin{aligned}
a &\equiv b \\
\Rightarrow m &| (b - a) \\
\Rightarrow b &= a + km \text{ for } k \in \mathbb{Z} \\
\Rightarrow a &= b - km \text{ for } (-k) \in \mathbb{Z} \\
\Rightarrow m &| (a - b) \\
\Rightarrow b &\equiv a
\end{aligned}$$

- *Transitive*

Suppose $a \equiv b \equiv c$

$$\begin{aligned}
a &\equiv b \\
\Rightarrow m &| (b - a) \\
\Rightarrow b &= a + km \text{ for } k \in \mathbb{Z}
\end{aligned}$$

and

$$\begin{aligned}
b &\equiv c \\
\Rightarrow m &| (c - b) \\
\Rightarrow c &= b + lm \text{ for } l \in \mathbb{Z}
\end{aligned}$$

and combining the above equations,

$$\begin{aligned}
c &= a + (k + l)m \text{ for } (k + l) \in \mathbb{Z} \\
\Rightarrow m &| (c - a) \\
\Rightarrow a &\equiv c
\end{aligned}$$

Note that 10, 3, 17, 24, 87, are congruent among themselves *modulo* 7, because $87 - 17$ or $17 - 10$ or $24 - 87$ are all divisible by 7. In fact, numbers are congruent among themselves when their remainders are the same on division by m . To examine this further, we first need a simple fact.

Theorem: (without proof) Let n and m be two integers. There exist unique integers q and r such that $n = qm + r$, and $0 \leq r < m$. r is often denoted $n \text{ rem } m$.

From the examples it appears that the equivalence partitions the set of integers into m sets, each consisting of integers with the same remainder when divided by m . For example, the equivalence *mod* 26 partitions the set of integers into

26 sets, which we may number A through Z:

$$A = \{\dots - 52, -26, 0, 26, 52, \dots\}$$

$$B = \{\dots - 51, -25, 1, 27, 53, \dots\}$$

.....

$$Z = \{\dots - 27, -1, 25, 51, 77, \dots\}$$

We can show this formally.

Theorem: $a \equiv b \Leftrightarrow a \text{ rem } m = b \text{ rem } m$.

(For example, $10 \bmod 7 = 3$. Also, $-4 \bmod 7 = 3$, and $87 \bmod 7 = 3$, $17 \bmod 7 = 3$, that is, $10 \equiv -4 \bmod 7$, $10 \equiv 87 \bmod 7$, etc.).

Proof: Suppose $a = q_a m + r_a$ and $b = q_b m + r_b$, where $r_a = a \text{ rem } m$ and $r_b = b \text{ rem } m$.

$$\begin{aligned} a &\equiv b \\ \Rightarrow m \mid (b - a) \\ \Rightarrow m \mid (q_b - q_a)m + (r_a - r_b) - n < r_a - r_b < n \\ \Rightarrow m \mid (r_a - r_b) - n < r_a - r_b < n \\ \Rightarrow r_a - r_b &= 0 \\ \Rightarrow r_a &= r_b \end{aligned}$$

and

$$\begin{aligned} r_a &= r_b \\ \Rightarrow b - a &= (q_b - q_a)m + (r_a - r_b) \\ \Rightarrow b - a &= (q_b - q_a)m \\ \Rightarrow m \mid (b - a) \\ \Rightarrow a &\equiv b \end{aligned}$$

Now we see that, while doing any *modulo m* operations on two numbers, say a and c , we can get the same result by doing the operations on two other numbers, say b and d , congruent to a and c respectively. That is, instead of a number $a \bmod m$, we can take any element from its representative class *modulo m*. Hence operations *modulo m* are operations on the entire class representing the respective numbers, and not on the individual numbers themselves.

Theorem: If $a \equiv b \bmod m$ and $c \equiv d \bmod m$, then:

$$a + c \equiv b + d \bmod m$$

$$a \times c \equiv b \times d \bmod m$$

Proof: If $a = q_a m + r_a$, then $b = q_b m + r_a$. If $c = q_c m + r_c$, then $d = q_d m + r_d$.

$$\begin{aligned} a + c \text{ mod } m & \\ &= (q_a + q_c)m + r_a + r_c \text{ mod } m \\ &= r_a + r_c \text{ mod } m \end{aligned}$$

Further,

$$\begin{aligned} b + d \text{ mod } m & \\ &= (q_b + q_d)m + r_a + r_c \text{ mod } m \\ &= r_a + r_c \text{ mod } m \\ &= a + c \text{ mod } m \end{aligned}$$

$$\begin{aligned} a \times c \text{ mod } m & \\ &= (q_a m + r_a)(q_c m + r_c) \text{ mod } m \\ &= (q_a q_c m + q_a + q_c)m + r_a r_c \text{ mod } m \\ &= r_a r_c \text{ mod } m \end{aligned}$$

Further,

$$\begin{aligned} b \times d \text{ mod } m & \\ &= (q_b m + r_a)(q_d m + r_c) \text{ mod } m \\ &= (q_b q_d m + q_b + q_d)m + r_a r_c \text{ mod } m \\ &= r_a r_c \text{ mod } m \\ &= a \times c \text{ mod } m \end{aligned}$$

We pick the easiest number to use to represent the class, *modulo* m , of integer a : the remainder when a is divided by m .

Definition 2: $a \text{ MOD } m = a \text{ rem } m$.

Thus, $10 \text{ MOD } 7 = 3$, $3 \text{ MOD } 7 = 3$, $12 \text{ MOD } 7 = 5$, $-3 \text{ MOD } 7 = 4$ and so on.

\mathbb{Z}_m denotes the set of all remainders *modulo* m with the additive and multiplicative operations *modulo* m .

Definition 3: $\mathbb{Z}_m = \{0, 1, 2, \dots, m-1\}$ with operations $+_m$ and \times_m , addition and multiplication *modulo* m respectively, resulting in numbers expressed *MOD* m .

5 The Shift Cipher over \mathbb{Z}_m

We are now in a position to understand the shift cipher and generalize it to \mathbb{Z}_m for any m . Its encryption and decryption rules are:

$$e_K : \mathbb{Z}_m \rightarrow \mathbb{Z}_m$$

$$e_K(g) = g +_m K$$

$$d_K : \mathbb{Z}_m \rightarrow \mathbb{Z}_m$$

$$d_K(g) = g - K \text{ MOD } m$$

Examples:

- $m = 26$: \mathbb{Z}_{26} represents the English alphabet, where each letter is represented by a number *MOD* 26.
- $m = 8$: \mathbb{Z}_8 represents the set of all bytes, represented as numbers *MOD* 8
- $m = 2$: \mathbb{Z}_2 is the set of all bits, represented as numbers *MOD* 2.

6 The Affine Cipher over \mathbb{Z}_m

We can also generalize the affine cipher to \mathbb{Z}_m for any m . We get the following encryption and decryption rules:

$$e_K : \mathbb{Z}_m \rightarrow \mathbb{Z}_m$$

$$e_K(g) = a \times_m x +_m b$$

$$d_K : \mathbb{Z}_m \rightarrow \mathbb{Z}_m$$

$$d_K(g) = ?$$

In the next module, we try to abstract the commonalities between the operations in the shift and affine ciphers.