Cryptography and Modular Arithmetic

Cryptography has been important through the ages for sending secret military information. It has also been used by secret societies like the Freemasons. Today, computers and the internet have made cryptography a part of all our lives. Critical information like passwords, on-line purchases, and ATM transactions all use cryptography. Many companies protect their industrial secrets by encoding their files. Companies and individuals often encrypt their email to protect themselves from third party snooping.

We will introduce some simple methods of encoding that use algebra methods, in particular modular arithmetic to encode messages. We refer to the original message as the *plaintext* and the encrypted message as the *ciphertext*.

1 Simple Shift Ciphers

Julius Caesar was one of the first people known to use cryptography to protect messages of military significance (`http://en.wikipedia.org/wiki/Caesar_cipher`). Suetonius describes Julius Caesar's simple cipher in his *Life of Julius Caesar* 56 (tr. J. C. Rolfe):

> There are also letters of his to Cicero, as well as to his intimates on private affairs, and in the latter, if he had anything confidential to say, he wrote it in cipher, that is, by so changing the order of the letters of the alphabet, that not a word could be made out. If anyone wishes to decipher these, and get at their meaning, he must substitute the fourth letter of the alphabet, namely D, for A, and so with the others.
>
> (`http://laudatortemporisacti.blogspot.com/2004/09/secret-writing.html`)

We call this the *Caesar Cipher*. Every letter is shifted over by three. Using our modern alphabet, look up a plaintext letter in the top row of this table and replace that letter with the corresponding letter in the bottom row. To decrypt, look up a cipher text letter in the bottom row and replace it with the corresponding letter in the upper row.

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C

DISCRETE MATH becomes GLVFUHWH PDWK.

More generally, we could shift by any number from 1 to 25, for example, if we shift by 7,

DISCRETE MATH becomes KPZJYLAL THAO.

With a general shift cipher, the number you shift by, e.g. 7, is the *key* to the code. A simple shift cipher *rot13* from "rotate alphabet 13 places", is used on many newsgroups for posting things that might offend some readers. One advantage of rot13 over other shifts is that it deciphers itself. If you shift a letter 13 places and then shift the result 13 places, you are back to the original letter.

1.1 Simple Shift Cipher Links

You can see demos of a number of simple encryption methods at

`http://www.cs.usask.ca/resources/tutorials/csconcepts/1999_3/lessons/L3/SimpleEncryption.html`

and a demo of shift ciphers in particular at

`http://www.louisville.edu/~ahdeso01/applets/Shift.html`.

The Wikipedia has a good explanation of rot13 and its history, in addition to some nice examples of shift ciphers in general:

`http://en.wikipedia.org/wiki/ROT13`

Make sure to look at the "Trivia" section.

A number of encryption toys are based on the simple shift cipher, for example, the **Captain Midnight Secret Decoder Badges** were popular in the mid-1950s and **Ovaltine Secret Decoder Rings** `http://home.teleport.com/~jrolsen/premiums/ovaltine.html` were a hit in 1990. These toys, however, replace plaintext letters with ciphertext numbers.

2 Encoding

Actually, using numbers instead of letters gives us the advantage that we can put math and computers to work to encrypt and decrypt for us. So, the first thing we will do is *encode* our plaintext, that is, replace the letters with numbers by an agreed upon, public method. There are many ways we can do this. Computers mostly use ASCII (American Standard Code for Information Interchange http://www.lookuptables.com/) to represent characters. We will just use the numbers from 0 to 25 for the letters A to Z (or a to z).

A	B	C	D	E	F	G	H	I	J	K	L	M
00	01	02	03	04	05	06	07	08	09	10	11	12
N	O	P	Q	R	S	T	U	V	W	X	Y	Z
13	14	15	16	17	18	19	20	21	22	23	24	25

We have added leading 0s to the single digit numbers so that all the codes are 2 digits long. If we need punctuation, we will use 26 for a space, 27 for a period and 28 for a comma.

Encoding, going from letters to numbers, and *decoding*, going from numbers back to letters, are different from encrypting (or enciphering) and decrypting (or deciphering). There is nothing secret about encoding and decoding. MATH IS COOL becomes 12001907 0818 02141411 if we leave the spaces or 120019072608182602141411 if we encode the spaces.

What is the original message that encodes to: 1804170413081924 ?

3 The mod Function

The *mod* function has many applications in computer science so we will study it in some detail. It is used for simple and complex cryptography, calendars and clocks, random number generators, and hash tables for a start. We will then use the mod function, to generate shift ciphers and more general *linear ciphers*.

If n is an integer that is greater than 1, and a is any integer, then

$$a \bmod n$$

is the integer remainder when a is divided by n. In fact, $a \bmod n$ is defined when n is negative but we'll restrict our attention to $n > 1$. In this case, $a \bmod n$ is always an integer between 0 and $n - 1$. In Scheme the mod function is given by (modulo a n).

Examples

$17 \bmod 5 = 2$	17 divided by 5 is 3; the remainder is 2.
$8 \bmod 5 = 3$	8 divided by 5 is 1; the remainder is 3.
$55 \bmod 5 = 0$	55 divided by 5 is 11; the remainder is 0.
$4 \bmod 5 = 4$	4 divided by 5 is 0; the remainder is 4.
$37 \bmod 17 = 3$	37 divided by 17 is 2; the remainder is 3.

How do we evaluate $a \bmod n$ when a is negative? Remember that as long as $n > 1$, the values of $a \bmod n$ must be between 0 and $n - 1$. In general, $a \bmod n$ is the unique integer between 0 and $n - 1$ that satisfies $a = q \cdot n + a \bmod n$ for some integer q.

Examples

$-17 \bmod 5 = 3$	$-17 = -4 \cdot 5 + 3$
$-8 \bmod 5 = 2$	$-8 = -2 \cdot 5 + 2$
$-55 \bmod 5 = 0$	$-55 = -11 \cdot 5 + 0$
$-4 \bmod 5 = 1$	$-4 = -4 \cdot 1 + 1$
$-37 \bmod 17 = 14$	$-37 = -3 \cdot 17 + 14$

3.1 Properties of mod

Let n be an integer greater than 1, and let a and b be any integers, then

1. If $a \bmod n = b \bmod n$ then there is an integer k such that $a - b = k \cdot n$.
2. $(a + b) \bmod n = ((a \bmod n) + (b \bmod n)) \bmod n$
3. $(a \cdot b) \bmod n = ((a \bmod n) \cdot (b \bmod n)) \bmod n$
4. $-a \bmod n = n - (a \bmod n)$

Examples

$19 \bmod 8 = 3$ and $51 \bmod 8 = 3$	$51 - 19 = 32 = 4 \cdot 8$
$19 \bmod 5 = 4$ and $7 \bmod 5 = 2$	$(19 + 7) \bmod 5 = 26 \bmod 5 = 1 = (4 + 2) \bmod 5$
	$(19 \cdot 7) \bmod 5 = 133 \bmod 5 = 3 = (4 \cdot 2) \bmod 5$
$37 \bmod 17 = 3$ and $-37 \bmod 17 = 14$	$2 + 14 = 17$

4 Simple Substitution Ciphers

A *simple substitution cipher* is a cryptographic system in which letters (or their codes), are arbitrarily transposed or replaced with other letters (or their codes). The Ceasar Cipher and general Shift Cipher are both simple substitution ciphers. Cryptograms that sometimes appear as newspaper puzzles are also simple substitution ciphers. Each letter is replaced by another letter. We

will study some simple substitution ciphers that can be generated by using the mod or modulo function.

4.1 Shift Cipher

Once we have coded the letters A, ..., Z, a general shift cipher with shift k can be described by:

$$n \to (n + k) \bmod 26.$$

or by

$$n \to (n + k) \bmod 29.$$

if we encode and encipher space, "." and "," as well as the letters A $\cdots$ Z. If we want our encrypted message to look like letters, possibly with punctuation, we decode the shifted codes to for our ciphertext. Here's an example.

MATH IS COOL becomes 12001907 0818 02141411 if we just encode the letters. If we shift by 15, we get

Letter	M	A	T	H	I	S	C	O	O	L
Coded	12	00	19	07	08	18	02	14	14	11
Shifted	1	15	8	22	23	7	17	3	3	00
Decoded	A	P	I	W	X	H	R	D	D	A

If we receive the message "APIWXHRDDA" and know that the shift key is 15, We just reverse the procedure above to decrypt our message, code the letters, shift by -15 which is the same as $+11$ mod 26, decode the result.

Cipher Letter	A	P	I	W	X	H	R	D	D	A
Coded	1	15	8	22	23	7	17	3	3	00
Shifted	12	00	19	07	08	18	02	14	14	11
Letter	M	A	T	H	I	S	C	O	O	L

4.2 Linear Ciphers

We can create somewhat more complex simple substitution ciphers by using linear functions along with mod instead of just adding a constant and then using mod. Let's work again with just the 26 letters. Ingeneral, we choose two constants m and k then generate a *linear cipher* is given by

$$a \to (m \cdot a + k) \bmod 26.$$

Lets look at an example with $m = 5$ and $k = 11$.

Letter	A	B	C	D	E	F	G	H	I	J	K	L	M
Code	00	01	02	03	04	05	06	07	08	09	10	11	12
Moved	11	16	21	00	05	10	15	20	25	04	09	14	19
Cipher	L	Q	V	A	F	K	P	U	Z	E	J	O	T
Letter	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
Code	13	14	15	16	17	18	19	20	21	22	23	24	25
Moved	24	03	08	13	18	23	02	07	12	17	22	01	06
Cipher	Y	D	I	N	S	X	C	H	M	R	W	B	G

This seems to make a pretty good simple substitution cipher. Two different letters always have two different cipher letters so an enciphered message should be decipherable. No letter is enciphered by itself so the message won't be trivial to read. Given the table, it is pretty easy to decipher a message. Can you decipher this message? QHKKBXOLBXMLTISFX

There are a few questions we should think about when we make a simple linear cipher.

1. What values of m and k make good linear ciphers if the alphabet has 26 characters?
2. What if the alphabet has 29 characters, e.g. with space, "." and "," included?
3. What if the alphabet has 128 ASCII characters?
4. Can we say anything in general for an alphabet of n characters?
5. Can the person receiving our message decipher it without reconstructing the table, i.e. with just knowing n, m, and k? This will be important if n is large.

To answer these questions, we need to understand more about mod and the arithmetic it induces.

5 Modular Arithmetic

Once we fix and integer n greater than 1, the properties of mod, we cited above, allow us to talk about arithmetic mod n on the set $\mathbb{Z}_n$ of integers from 0 to $n-1$. We define

$$a + b = (a + b) \bmod n$$

$$a \times b = (a \times b) \bmod n$$

Consider these + and × tables for arithmetic mod 3.

+	0	1	2
0	0	1	2
1	1	2	0
2	2	0	1

×	0	1	2
0	0	0	0
1	0	1	2
2	0	2	1

Arithmetic mod 3 has some very nice properties. If a, b, and c are in $\mathbb{Z}_3$ (the set $\{0, 1, 2\}$) then

closure: $a + b$ and $a \times b$ are in $\mathbb{Z}_3$

commutativity: $a + b = b + a$ and $a \times b = b \times a$

associativity: $(a + b) + c = a + (b + c)$ and $(a \times b) \times c = a \times (b \times c)$

identity +: 0 is an *additive identity* $a + 0 = a$ for all $a \in \mathbb{Z}_3$

identity ×: 1 is a *multiplicative identity* $a \times 1 = a$ for all $a \in \mathbb{Z}_3$

inverse +: Every $a \in \mathbb{Z}_3$ has an *additive inverse* $b \in \mathbb{Z}_3$ such that $a + b = 0$

inverse ×: Every non-zero $a \in \mathbb{Z}_3$ has an *multiplicative inverse* $b \in \mathbb{Z}_3$ such that $a \times b = 1$

distributive law: $c \times (a + b) = (c \times a) + (c \times b)$

Note: The symbol $\in$ means "in" so $a \in \mathbb{Z}_3$ means "a in $\mathbb{Z}_3$."

These properties mean that the set $\mathbb{Z}_3$ with $+$ and $\times$ mod 3 is a mathematical *field*. The real numbers, rational numbers, and complex numbers are also mathematical fields with their regular addition and multiplication.

Now, let's consider these $+$ and $\times$ tables for arithmetic mod 4 on the set $\mathbb{Z}_4 = \{0, 1, 2, 3\}$..

$+$	0	1	2	3
0	0	1	2	3
1	1	2	3	0
2	2	3	0	1
3	3	0	1	2

$\times$	0	1	2	3
0	0	0	0	0
1	0	1	2	3
2	0	2	0	2
3	0	3	2	1

Addition mod 4 has similar properties to addition mod 3. There is an additive identity, 0, and every $a \in \mathbb{Z}_4$ has an additive identity, $0+0 = 1+3 = 2+2 = 0$. But $\mathbb{Z}_4$ is not a field. It does have a multiplicative identity, $a \times 1 = a$ for all $a \in \mathbb{Z}_4$ but 2 does not have a multiplicative inverse. We cannot solve $2 \times b = 1$ or $2 \times b = 3$ in $\mathbb{Z}_4$. In fact, $2 \times 2 \bmod 4 = 0$. We say *2 is a zero-divisor mod 4*. In general, We say $a \in \mathbb{Z}_n$ is a *zero-divisor* mod n if there is a non-zero $b \in \mathbb{Z}_n$ such that $a \times b = 0$.

Can you now say what values of m will be bad for linear ciphers, $a \to (m \cdot a + b) \bmod 26$?

6 Powers mod n

We often have to compute powers of numbers mod n. The RSA Encryption Algorithm (`http://world.std.com/~franl/crypto/rsa-guts.html`) which is widely used in electronic commerce protocols uses high powers of numbers mod n. We can easily compute powers mod n when when the exponent is itself a power of 2 by using the property

$$(a \cdot b) \bmod n = ((a \bmod n) \cdot (b \bmod n)) \bmod n.$$

and the fact that

$$a^{(2^k)} = a^{(2 \cdot 2^{k-1})} = a^{(2^{k-1}+2^{k-1})} = \left(a^{\left(2^{(k-1)}\right)}\right) \cdot \left(a^{\left(2^{(k-1)}\right)}\right) = \left(a^{\left(2^{(k-1)}\right)}\right)^2.$$

The idea is to alternate evaluating mod n and squaring. This is probably best understood by looking at some examples.

Examples

$37^2 \bmod 3 = (37 \bmod 3)^2 = 1^2 = 1$

$115^4 \bmod 7 = (115 \bmod 7)^4 = (3 \bmod 7)^4 = (3^2 \bmod 7)^2 = (9 \bmod 7)^2$

$= (2 \bmod 7)^2 = (2^2 \bmod 7) = (4 \bmod 7) = 4$

$115^{32} \bmod 7 = (115 \bmod 7)^{32} = ((115 \bmod 7)^4)^8 = (4 \bmod 7)^8$

$= (16 \bmod 7)^4 = (2 \bmod 7)^4 = (16 \bmod 7) = 2$

When we study binary representation of integers, we will learn how to put this idea to work to compute powers mod n when the exponents are not powers of 2.