MESA Cyber Robot Challenge: Cryptography and Number Bases

Introduction to Cryptography

Suppose you want to tell your friend a secret, something that you don't want anyone else to hear. If you are together, you can whisper the message close to your friend's ear. But suppose that speaking to your friend in privacy is not an option. How can you communicate your message to your friend and prevent others from sharing it?

For thousands of years, people have been finding ways to hide communications from anyone other than the intended recipient. **Cryptography** is the design and use of methods to conceal messages. These methods can rely on very simple techniques like scrambling the letters in the message, or use highly theoretical algorithms based on advanced mathematics.

Cryptography has long been used by militaries and governments to facilitate secret communications, but is now also commonly used within many civilian systems to protect data. For example, most digital communications today travels across the public networks (the Internet). It is virtually impossible to guarantee that an eavesdropper cannot physically access the information traveling over the Internet. However, for many computer applications that use the Internet, such as e-commerce, maintaining the privacy of their information is critical. In e-commerce

(buying products and services over the Internet), a buyer's credit card information must be protected while it is in transit from the buyer to the seller. Cryptography is used to protect information transmitted over the Internet, including emails and Internet telephony. It is also used to protect entire communications infrastructures, such as wireless networks; stored data, from single files to entire hard disks; and computer code, such as computer operating systems.

Let's consider a basic communication scenario. Imagine that Alice wants to communicate secretly with her friend Bob. Alice must somehow transform her message, called **plaintext**, in order to keep the message private. This process is called **encryption**.

Alice uses an encryption algorithm, or **cipher**, to transform her plaintext message into **ciphertext**. Ciphertext is an unreadable form of the original message that hopefully prevents others from eavesdropping. The cipher uses the plaintext together with a **key** that determines exactly how to transform the plaintext into ciphertext. Bob uses the same cipher used by Alice and a key (possibly different from the key used by Alice) to **decrypt** the ciphertext and recover Alice's message.

The key is very important – even if others know the cipher that Alice and Bob use to communicate, they can't read Alice's and Bob's messages without the right key.

Below we discuss three categories of ciphers: **substitution** ciphers, **transposition** ciphers, and **public/private keys**.

Substitution Ciphers

In a substitution cipher, letters in the plaintext are replaced by other letters.

The simplest example is the **Caesar cipher,** so called because Caesar used this technique to communicate secretly with his generals. To encrypt a message using the Caesar cipher, each letter in the plaintext is shifted by the same number of letters in the alphabet. The number of letters to shift is based on the value of the key.

For example, if Alice's plaintext message is HELLO and the key is 3, then the Caesar cipher will generate HELLO+3 = KHOOR. Notice, that each letter in the encrypted message, KHOOR is exactly three letters away from each letter in the plaintext message; K is three after H, H is three letters from E and so on. Bob can retrieve the original message from the encrypted message by simply taking the text he receives and subtracting the value of the key from each letter.

What happens when a plaintext letter appears near the end of the alphabet and “falls off” the end when its value is added to the key? The solution is to “wrap around” to the beginning of the alphabet whenever this occurs. For example, the letter X becomes an B with a key of 4.

Like the Caesar code, the **Vigenère cipher** also shifts letters in the plaintext. However, instead of using the same shift for every letter, it applies different shifts defined by a **keyword**. For example, if the keyword is DOG, consisting of the 4th, 15th, and 7th letters of the alphabet, then the first letter of the message is shifted by 4, the second letter is shifted by 15, the third by 7, the fourth by 4 (here we return to the beginning of the keyword), the fifth by 15, and so on. Using this cipher and the keyword DOG, Alice’s HELLO message becomes LTSPD. Decryption involves sequentially subtracting the keyword shift values.

A final example of a substitution cipher is the **Polybius Square cipher**. In this cipher, messages are encrypted and decrypted using a 25-letter **key square** like the one below:

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

To encrypt a message, replace each letter of the plaintext message with the two letters denoting its row and column. For example, the plaintext letter R is replaced with the pair of letters IE because R lives on row I and column E in the key square. The plaintext message “RETURN AT DAWN” becomes

Plaintext:	RETURN AT NOON
Ciphertext:	IEGEJDHAIEGC IAJD GCHEHEGC

Decrypting a message simply involves taking the ciphertext letters in pairs and replacing them with the plaintext letter belonging to their row and column. For example, the ciphertext pair of letters HD decrypts to the plaintext letter S because S lives on row H and column D.

Transposition Ciphers

In a **transposition cipher** the letters of the plaintext message are rearranged in a different and usually quite complex way, but the letters themselves are left unchanged.

In **a simple transposition cipher**, the message is rearranged in a way determined by a keyword known by both the sender and receiver. To encrypt the plaintext, the sender writes the message in a box composed of several rows of a fixed length, ignoring spaces between words and punctuation. Then the message is read out again column by column; however, the columns are chosen in a scrambled order. Both the width of the rows and the order of the columns are defined by the keyword: the number of rows is the length of the keyword, and the column permutation is determined by the alphabetical order of the letters in the keyword. For example, suppose the keyword is ZEBRAS. Since the keyword is 6 letters long, then the rows are of length 6; the column order is: 6 3 2 4 1 5, the alphabetical order of the letters in the word ZEBRAS.

It is possible that there will not be enough letters in the plaintext to fill the last row. Then the message is padded at the end with enough dummy letters to fill this row.

Let's consider this longer message from Alice: "HELLO, HOW IS THE WEATHER THERE?" Let's use the keyword EARLY to encrypt this message using a simple transposition cipher. First, we write the message out in a box with rows of length 5:

E	A	R	L	Y
H	E	L	L	O
H	O	W	I	S
T	H	E	W	E
A	T	H	E	R
T	H	E	R	E

Next, we read the columns in the order indicated by the keyword: 2 1 4 3 5. The ciphertext is "EOHTHHHTATLIWERLWEHEOSERE".

Can you figure out how Bob would decipher this message from Alice? Remember that he knows that Alice used the simple transposition cipher with the keyword EARLY.

Public Key Cryptography

There are two types of cryptography, **symmetric** and **asymmetric**. So far, the ciphers we have discussed have been symmetric, meaning that the encryption and decryption keys are identical. Modern cryptographic systems rely on more sophisticated asymmetric techniques that require two separate keys, one to encrypt the plaintext and one to decrypt the ciphertext. Neither key alone will do both functions. This cryptography system eliminates the need for key exchange, enhancing the security and privacy of the messages.

An asymmetric system commonly in use today is the **public/private key** system. In this system, one of these keys is published (public) and the other is kept private. Public/private key cryptography was invented in the mid 1970's. Examples include the Diffie-Hellman key exchange protocol and the RSA encryption algorithm.

The public/private key system relies on concepts from algebra, such as modular arithmetic. Specifically it relies on the modulo operator, which computes the remainder of the division of two integers. For example, given two integers $a = 5$ and $b = 3$, $a \bmod b$ would be 2, since a divided by b is equal 1 with a remainder of 2.

In this system, messages must be positive integers. Here is a simple example of a public/private key cipher. In order to send a message, Bob chooses any two integers a_1 and b_1, and sets $Z = a_1b_1 - 1$. He then chooses two more integers a_2 and b_2, and sets

$$e = a_2 Z + a_1, \qquad d = b_2 Z + b_1, \qquad n = (de - 1)/Z \;= a_2 b_2 Z + a_1b_2 + a_2b_1 + 1.$$

Note that since $de - nZ = 1$, then $de = 1 \pmod{n}$.

Bob's public key is (n, e), and his private key is (n, d). Anyone, including Alice, can know Bob's public key, but only Bob knows his private key.

To send Bob a plaintext m, Alice computes $c = em \pmod{n}$ to encrypt the message using Bob's public key (n, e). Bob uses his private key to decipher the ciphertext by computing $dc \pmod{n}$. Note that the decryption operation recovers the plaintext, because $dc \pmod{n} = dem \pmod{n} = m \pmod{n}$.

As an example, suppose Bob chooses $a_1 = 3$, $b_1 = 5$, $a_2 = 2$, and $b_2 = 7$. Then $Z = 14$, $e = 31$, $d = 103$, and $n = 228$. Bob's public key is (228, 31). Now suppose that Alice wants to send Bob the numeric message $m = 13$. This could represent anything that Alice and Bob both understand. For example, this could mean the letters "ac". Alternatively, this could be a message that Alice and Bob have previously agreed on to mean "meet me at the mall." Note: the value of Alice's message should be less than n.

To encrypt the message, Alice computes

$$c = em \pmod{n} = 31 \cdot 13 \pmod{228} = 175,$$

and sends this ciphertext to Bob. After receiving this message, Bob computes

$$m = dc \pmod{n} = 103 \cdot 175 \pmod{228} = 13.$$

Challenge: Suppose you are an eavesdropper, and you detect Alice's ciphertext c = 175 sent to Bob. You also know Bob's public key (228, 31). How would you go about trying to determine Alice's original plaintext m?

Cryptography Sources

http://williamstallings.com/Crypt-Tut/Crypto%20Tutorial%20-%20JERIC.html
http://www.math.washington.edu/~koblitz/crlogia.html
http://practicalcryptography.com/ciphers/classical-era/polybius-square/
http://en.wikipedia.org/wiki/Transposition_cipher
http://library.thinkquest.org/04oct/00451/trancipher.htm
http://en.wikipedia.org/wiki/Modulo_operation

Introduction to Number Base Conversions

One might ask, "How are number bases relevant to me?" You may or may not know that you have been using numbers with bases regularly. The number base system that you are likely most comfortable with, called the **decimal** system, is a base 10 number system and can represent any quantity using only combinations of the ten symbols 0 through 9. But other number base systems exist, too. For example, the base 5 number system can represent any quantity using only combinations of the five symbols 0 through 4. The base 20 number system can represent any quantity using combinations of the symbols 0 through 9 and the symbols A through J (20 symbols in all). The rise of computers and computer graphics has increased the need for knowledge of how to work with different (non-decimal) base systems, particularly **binary** systems (i.e., base 2, where the only symbols available to use are 0 and 1) and **hexadecimal** systems (i.e., base 16, where the 16 available symbols are the symbols 0 through 9 followed by the symbols A through F). It is often important to be able to convert number representations between different number bases to understand how a computer processes data. This document focuses on how to convert non-decimal (i.e., non-base 10) numbers to decimal (base 10) numbers.

Background Information

To represent a numerical **quantity** in a particular base, we write our quantity as n_b where n is the **number** and b is the **base**.

Example: 12_{10} is read "12 base 10". (This is the twelve that we are used to.)

The number n is written as a string of **symbols** drawn from a set of no more than b distinct symbols. It is traditional to use the following set of symbols:

0 1 2 3 4 5 6 7 8 9 A B C D E F G H (...and so on)

These symbols represent the quantities zero, one, two, etc. as in the following table:

symbol	quantity
0	zero
1	one
2	two
…	…
9	nine
A	ten
B	eleven
C	twelve
…	…

For a base b, only the first b symbols are allowed to be used. So, for example, if $b = 10$ (i.e., the decimal system), the only allowed symbols are 0, 1, 2, 3, 4, 5, 6, 7, 8, and 9. If $b = 4$, the only allowed symbols are 0, 1, 2, and 3.

The numerical quantity represented by a number and its base is determined by the base to the power of the **position**:

Example: $628_b = 6 \times b^2 + 2 \times b^1 + 8 \times b^0$

So if we look at the symbol 8, it is in zeroth position (i.e., zero spots left of the far right) which is why we have 8 multiplied by the base b raised to the 0th power. The symbol 2 is in the first position (i.e., one spot left of the far right), which is why we have 2 multiplied by the base b raised to the 1st power. The symbol 6 is in the second position (i.e., two spots left of the far right), which is why we have 6 multiplied by the base b raised to the 2nd power. If more symbols were present, this pattern of interpretation would continue. In the end, the final quantity represented by the number and base is the sum of all the quantities represented by the individual symbols in the number.

Examples of Base 10 Conversion

Example 1

Convert 31_8 to base 10.

1. First, identify the position of each of the symbols. In this case, 1 is in position zero and 3 is in position one. If the number were 431_8, the 4 would be in position 2, and so on.
2. Next, for each position, raise the base b to the power of the position.
3. Then, multiply the base 10 value of each symbol (i.e., what we think of as the normal value associated with each symbol: e.g., 3 = three, 8 = eight, C = twelve, and so on) to the results from step 2.
4. Finally, add all the results together to get the final value.

	position one	position zero
Step 1	3	1
Step 2	$8^1 = 8$	$8^0 = 1$
Step 3	$3 \times 8^1 = 24$	$1 \times 8^0 = 1$
Step 4	$24 + 1 = 25$	

Result: 31_8 is equal to 25 in base 10.

Example 2

Convert 16_9 to base 10

	position one	position zero
Step 1	1	6
Step 2	$9^1 = 9$	$9^0 = 1$
Step 3	$1 \times 9^1 = 9$	$6 \times 9^0 = 6$
Step 4	$9 + 6 = 15$	

Result: 16_9 is equal to 15 in base 10.

Example 3

Convert $52E_{16}$ to base 10

	position two	position one	position zero
Step 1	5	2	E (=14)
Step 2	$16^2 = 256$	$16^1 = 16$	$16^0 = 1$
Step 3	$5 \times 16^2 = 1280$	$2 \times 16^1 = 32$	$14 \times 16^0 = 14$
Step 4	$1280 + 32 + 14 = 1326$		

Remember that the symbol E represents a value of 14.

Result: $52E_{16}$ is equal to 1326 in base 10.

Number Base Resources

There are many great online tools to assist you in learning how to convert numbers to different bases. The following were used to create this help sheet:

http://www.purplemath.com/modules/numbbase.htm
http://www.cs.umd.edu/class/sum2003/cmsc311/Notes/Data/toBaseTen.html
http://wartex8.com/tutorials/base_conversion.php (contains YouTube video)