CEMC.UWATERLOO.CA | The CENTRE for EDUCATION in MATHEMATICS and COMPUTING

Grade 11/12 Math Circles

Cryptography, Part 1 - Solutions

Exercise Solutions

Exercise 1

(a) Try encrypting the message RETREATIMMEDIATELY using a different number of columns (and possibly a different ordering of columns).

(b) Suppose you have received the ciphertext CTAIMFROPSUUYGHSCNPRYOH, knowing the encryption was done by transposition with four columns, taken in the order first-to-last. What is the decryption of this ciphertext?

Exercise 1 Solution

(a) Answers may vary.

(b) Here, we first count the number of letters in the message, and find that there are 23. Dividing the number of letters by 4 with remainder, we see that there will be 5 full rows of 4, with a final row having the 3 leftover letters.

The columns were taken from first-to-last, so we break the ciphertext into chunks of 6 letters and write them down each column, from first column to last column.

C	R	Y	P
T	O	G	R
A	P	H	Y
I	S	S	O
M	U	C	H
F	U	N	

Reading across the rows now, we recover the plaintext

CRYPTOGRAPHYISSOMUCHFUN

or “cryptography is so much fun”!

Exercise 2

(a) Try encrypting the message RETREATIMMEDIATELY using a shift cipher with a shift number different from 2.

(b) Suppose you receive the ciphertext HIGVCTXMSRMWWLMJXCFYWMRIWW, encrypted using a shift cipher with a forward shift of four letters. What is the decryption of this message?

Exercise 2 Solution

(a) Encryptions may vary. However, you will notice that there are only 25 different shifts that actually disguise the message (one for every letter of the alphabet, minus one for the shift that does nothing). Even worse, if you know someone is using the shift cipher and know even one letter of the plaintext, looking at one letter of the ciphertext is enough to find the shift and decrypt the whole thing. This is not good!

(b) The encryption/decryption table we get for a shift of four letters is the following:

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

Starting with the first ciphertext letter, H, we look up the corresponding plaintext letter, which is D in this case. For the next ciphertext letter, I, the corresponding plaintext letter is E. Continuing in this way over the entire ciphertext, we obtain the plaintext

DECRYPTIONISSHIFTYBUSINESS

Exercise 3

(a) Encrypt the word FALSE using a onetime pad cipher with random key FRQEX.

(b) Suppose you have received the ciphertext EJIKBKYXOS, encrypted with the onetime pad key QWERTYUIOP (the top row of a keyboard). What is the corresponding plaintext?

Exercise 3 Solution

(a) To save space, we omit the encryption details. The ciphertext you obtain in this way is KRBWB. Notice that the distinct plaintext letters L and E are encrypted as the same letter here!

(b) To decrypt, we perform the shift ciphers in reverse on each letter. Doing this, you obtain the plaintext ONETIMEPAD.

Problem Set Solutions

1. In this question, we will explore a couple pitfalls of using transposition ciphers.

   (a) Why is it a bad idea to use a transposition cipher to encrypt a very short message? Is a substitution cipher better in this situation?

   (b) The longer the plaintext becomes, the greater the chance of an error being introduced. In the transposition example, Example 1 of the lesson, where Nick sends

   EMTRMATIIYETDLRAEE

   to Shefaza, suppose he accidentally forgets one of the Ts, sending

   EMRMATIIYETDLRAEE

   instead. Using the same key, what would Shefaza get for a plaintext? Would the same type of problem happen with a substitution cipher?

   (c) Taking this idea further, suppose Nick encrypts a long message using a transposition cipher, and the second half of the message gets lost somehow (maybe it was on a different sheet of paper). Would Shefaza be able to recover any of the plaintext? What if a substitution cipher was used instead?

Solution:

(a) If the message is very short, there aren't many possible ways to scramble it. For instance, if the plaintext is HI, the only possible ciphertext is IH, and decrypting it is obvious even without the key.

A substitution cipher, on the other hand, works really well on short messages. As an extreme case, if you have a plaintext that is one letter long, there's no reasonable way to guess the plaintext when using this cipher.

(b) When Shefaza goes to decrypt the defective ciphertext, she counts that there are 17 letters in the ciphertext, and writes out the letters in five columns, with the last three columns all receiving three letters each, and the first two columns receiving four letters. Inserting the letters in columns, from last to first, we get

R	E	I	M	E
A	T	I	A	M
E	D	Y	T	R
E	L			

Reading across each row, Shefaza gets the "plaintext"

REIMEATIAMEDYTREL

This makes no more sense than the ciphertext did, leaving Shefaza in a conundrum to figure out what Nick meant to say.

If the same type of mistake happened when using a substitution cipher, the plaintext Shefaza gets would look like the one Nick sent, but with that one letter missing. Since all the letters are still in the right order, Shefaza should easily be able to figure out what the message means.

(c) With a transposition cipher, the loss of half the message is probably fatal – Shefaza will not be able to figure out anything Nick was trying to say. As an analogy, imagine you only have half the pieces to a jigsaw puzzle and are trying to figure out what the puzzle looks like fully assembled...

Of course, if a substitution cipher was used instead, the first half of the ciphertext corresponds to the first half of the plaintext, so Shefaza could still recover the first half of the message.

2. Using this encryption table,

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

try encrypting the following plaintext:

It was a bright day in April, and the clocks were striking thirteen.

(This is the first line from George Orwell's famous novel, *1984*.)

Solution: For this question, all we need to do is encrypt each letter of the plaintext, one at a time, using the given table. When we do this, the ciphertext turns out to be

XSUGBGZJXFHSCGEXMGWJXQGMCSHNIQYILBUNJNBSJXLXMFSHXJSNNM

3. Suppose you have intercepted the following ciphertext, which you know for sure has been encrypted using a shift cipher:

ALKLQDLTEBOBQEBMXQEJXVIBXADLFKPQBXATEBOBQEBOBFPKLMXQEXKAIB
XSBXQOXFI

What is the corresponding plaintext?

Solution: Probably the easiest way to do this is to start with the first five letters of the ciphertext, ALKLQ, and keep shifting all the letters forward one place in the alphabet until something sensible emerges. Doing this gives us

BMLMR
CNMNS
DONOT

Already, this last one makes sense: it could be ”Do not”. Applying this shift to all the ciphertext letters reveals the plaintext. Putting punctuation and spacing back in, it reads

Do not go where the path may lead, go instead where there is no path and leave a trail.

4. (a) Using the monoalphabetic substitution cipher, how many keys are possible?

   (b) Suppose you wanted to try breaking the monoalphabetic substitution cipher by brute force, trying every key until you get a sensible message. Let's be generous and say you managed to convince a billion of the Earth's inhabitants to help you on your quest. If every one of you could try one key per minute, roughly how many years would it take to try all the keys? For simplicity, assume each year has 365 days. (Let's also assume people live forever and can spend all of their time checking keys.)

   (c) The current age of the universe is estimated to be 13.772 billion years. Roughly how many times the current age of the universe would it take those billion people to finish checking the keys?

Solution:

(a) There are 26 letters that A could be replaced with in the ciphertext. After this choice is made, there are 25 remaining letters that B could be replaced with. This leaves 24 choices for the encryption of C, and so on. In total, there are

$$26 \cdot 25 \cdot 24 \cdot 23 \cdot \cdots \cdot 2 \cdot 1$$

possible keys when using this cipher. For those familiar with factorial notation, the answer can be re-written as 26!. Converted to scientific notation, this is about 4.03×10^{26}. If you like, you can subtract off the key that doesn't change any letters, but it doesn't really change the answers to the other parts of this question.

(b) In this situation, you can check 1 000 000 000 keys per minute. Since there are 60 minutes in an hour, 24 hours in a day, and 365 days in a year, you could manage to check

$$1\,000\,000\,000 \cdot 60 \cdot 24 \cdot 365 = 525\,600\,000\,000\,000$$

keys every year. If we divide the total number of keys by the number of keys we can check per year, we get

$$\frac{26!}{525\,600\,000\,000\,000} \approx 767\,297\,300\,469$$

years! That seems like a long time (we'll see just how long in the next part).

(c) Here, we take the answer from part (b) and divide through by 13 772 000 000 years, the approximate age of the universe:

$$\frac{767\,297\,300\,469}{13\,772\,000\,000} \approx 55.7.$$

So, even with a billion people to help you check keys, you'd all still be going for almost 56 times the current age of the universe. No wonder people thought the monoalphabetic substitution cipher was secure (at least at first)!

5. We mentioned that the onetime pad is only secure if:

   (1) The key is generated completely randomly.

   (2) Every key is used only once.

   The goal of this question is to see what kinds of problems pop up if either rule is broken.

   (a) Rather than using a random key, suppose we form a key by repeating a short word over and over again, such as KINGKINGKINGKING..., looping this until we get to the end of the plaintext. This special case of the one-time pad cipher was mentioned way back in 1553 by Giovan Battista Bellaso, but carries the name *Vigenère cipher* due to a misattribution to a 19th century French man named Blaise de Vigenère.

      i. If you happen to know the length of the repeated word, how can you reduce the problem of breaking the cipher to the problem of cracking a small number of shift ciphers?

      ii. Even if you don't know how long the repeated word is, it is often still possible to find out. Suppose you got your hands on a ciphertext, and saw that "TPC" appears twice, 75 letters apart, and "APSA" also appears twice, 27 letters apart. Based on this, how long is the repeated word that forms the key likely to be?

   (b) Suppose two teachers use the onetime pad to encrypt answers to true-false tests (just to keep them away from the prying eyes of their students). The plaintexts are just strings of T's and F's. So, for example, the answers to a ten-question test might be TFFFTTFTFF. Now suppose the teachers were negligent and used the same onetime pad key to encrypt the answers to two different tests with the same number of questions. If you can see both ciphertexts, how you can determine which questions on these tests have the same answer, and which don't? (If you've already written the first test, this will give you all the answers for the second one!)

Solution:

(a) i. If we know (for instance) that the word being repeated is four letters long, we can just break the message up into four pieces:

- The first, fifth, ninth, . . . letters.
- The second, sixth, tenth, . . . letters.
- The third, seventh, eleventh, . . . letters.
- The fourth, eighth, twelfth, . . . letters.

Each of these chunks of message has been encrypted with a shift cipher. After taking a guess at each of the four shifts, it helps to look at which four-letter word would produce these shifts. If it comes out as a sensible word (like KING from the example), it is quite likely you've found the key. (But keep in mind that someone might use a gibberish four-letter word just to throw you off the scent.)

ii. We assume that these repeated pieces of ciphertext come from repeated pieces of plaintext encrypted according to the same set of shifts. Let n be the length of the repeated word that forms the key. If "TPC" shows up twice, 75 letters apart, it is reasonable to guess that 75 is a multiple of n. In that case, pieces of plaintext 75 letters apart *are* encrypted by the same collection of shifts. For the same reason, "APSA" showing up twice, 27 letters apart, suggests that 27 is a multiple of n. Let's list out all the factors of these numbers:

Factors of 75: 1, 3, 5, 15, 25, 75.

Factors of 27: 1, 3, 9, 27.

It looks like 1 and 3 are the factors in common, so it seems likely that n is either 1 or 3. But if $n = 1$, then the key is the same letter over and over again (like BBBB. . .), which makes it a shift cipher. We know those are very insecure, so $n = 3$ seems more reasonable.

After we know this information, following the process described in part i. can lead to a decryption. That's why the key in a onetime pad cipher should be totally random!

(b) This is best explained by example. Suppose the onetime pad key is QINTEDOMDW, and the answers to the two tests are:

TFFFTTFTFF
FFTTFTTFFT

Take a look at the corresponding encryptions:

JNSYXWTFIB
VNGMJWHRIP

Notice that some of the ciphertext letters are the same between the two messages, and others are different. In particular, the second, sixth, and ninth letters match, and the rest don't. Notice that the questions with matching letters are exactly the questions on the two tests with the same answer!

In general, all you have to do is compare the two ciphertexts to see which letters are the same. For those letters, the answers to those questions are the same on both tests. For all the rest of the letters, the answers to those questions are different. I don't want to spoil all the fun, so I leave it to you to convince yourself why this is true!

6. One problem with the type of cryptography described in this lesson (*symmetric key cryptography*) is the huge number of keys required to make sure everyone can communicate. Suppose we lived in an alternate universe, where this is the only form of cryptography we have discovered. In this scenario, the Government of Canada maintains a secure database of keys, and any pair of Canadians wishing to communicate securely establishes a key by connecting to the government server and storing their shared key there.

   For simplicity, assume the population of Canada is 38 million people. How many keys would the government need to store to guarantee that any pair of Canadians could communicate securely? If a new Canadian comes along, how many new keys would the government have to store?

Solution: Using the old-style "symmetric key" cryptography, any pair of individuals wishing to communicate would have to exchange a key. So, we need to count the number of pairs that can be formed from 38 million people. Each person in Canada can pair up with any of the 37 999 999 other Canadians, and this is true for each of the 38 million

Canadians. So it *seems* like there should be

$$38\,000\,000 \cdot 37\,999\,999 = 1\,443\,999\,962\,000\,000$$

keys. But we must be careful: we have actually counted each pair twice. For instance, the pairing {Nick, Shefaza} comes up when we count the number of pairs involving Nick, and also when we count the number of pairs involving Shefaza. So the *real* answer is

$$\frac{38\,000\,000 \cdot 37\,999\,999}{2} = 721\,999\,981\,000\,000.$$

That's still a lot of keys! For those familiar with binomial coefficients, another way to write the answer is $\binom{38\,000\,000}{2}$.

If the population of Canada increased even by 1, the 38 000 001st Canadian would need 38 000 000 new keys, one for every person already living in the country. This creates a lot of work for the government!

In the next cryptography lesson, we will describe *public key cryptography*, a modern method of encryption where each person gets two keys to use: one for encryption and one for decryption. Everyone uses the same encryption key to send messages to a given person (made public), and keeps the decryption key private. In a setup like this, the government would only have to store 38 000 000 keys, and if a new Canadian came along, the government would only have to store one additional key. Much better!