Course 7,850,2.00: Cybersecurity Exercises

Instructor: Prof. Katerina Mitrokotsa — Exercises #1
TAs: Florias P., Yu Liujun, Jenit Tomy & Nan Cheng — 23/09/2024

Instructions:

- Complete as many exercises as possible before the next exercise session.
- We will cover (most of) these in Exercise Session 1. Even if you haven't tried to solved them, please review them beforehand.
- Written solutions for all exercises will be available on Canvas before the next lecture.
- The exercises are color-coded by difficulty: easy, medium, hard.

Historical Ciphers

Exercise 1. (Shift ciphers)

Encrypt the sentence below using a shift cipher with *(your-student-number)* mod 26 as a number shift. For example, if your student number is "12-345-678", then the number of shifts you should use is 12345678 mod 26 = 20.

```
Veni, vidi, vici.
```

Exercise 2. (Shift ciphers II)

The following message was encrypted with a shift cipher. Decipher it and write down its' number of shifts.

```
Gur dhvpx oebja sbk whzcf bire gur ynml qbt
```

Exercise 3. (Substitution ciphers I)

A substitution cipher that uses the English alphabet has the following letter permutation as its key:

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

Using this key, the word `SECRET` is encrypted, giving us the word `AJPDJB`, which is again encrypted, and any subsequent word will also be encrypted using the same method. How many different words will we get in the end?

Exercise 4. (Vigenère cipher)

This exercise offers a brief introduction to the simplest polyalphabetic cipher, the Vigenère cipher. You can read more on this on pages 102-104 of "Cryptography and Network Security" by Stallings (also see "Help" at the end of this PDF).

We assign to each letter of the English alphabet a number, as follows: A ↔ 0, B ↔ 1, . . . , Z ↔ 25. Using the last 5 digits of your student number as a key (each number corresponding to a letter), encrypt the plaintext "MAYTHEFORCEBEWITHYOU" using the Vigenère cipher.

Modern Symmetric-key Cryptography

Exercise 5. (First Taste of Randomness)

In cryptography, each cipher has a certain number of possible keys that can be used for encryption. The set of all these possible keys is referred to as the ***key space****.*

In this exercise, we consider a key space $\mathcal{K}$ consisting of 10-bit strings (e.g. `1001110101`*), and we aim to analyse different ways of generating keys.*

A simple method for generating bits is by flipping a coin: if the result is heads (H), the corresponding bit is 0; if it's tails (T), the bit is 1. We can generate longer bit strings by flipping coins multiple times, e.g., by flipping 5 coins, the THTTH outcome will result in the string 10110.

Compute the probability distribution on the key space $\mathcal{K}$ of 10-bit strings (i.e., the likelihood of generating each 10-bit string) for the following key generation methods:

(i) Flip a coin 10 times and associate these outcomes the corresponding bit string.

(ii) Flip a coin 5 times. Then, concatenate the result with itself to get the key.

(iii) Pick an integer from 0 to 1000. The binary representation of the number is the key (fill the empty spaces with 0s).

(iv) Pick an integer from 0 to 2000. The binary representation of the number is the key (fill the empty spaces with 0s, eliminate the most significant bit if the string is longer than 10 bits).

Hint: As a sanity check for each subquestion, you can try to add up all the probabilities over different occurrences to make sure that they add up to 1 (as all probabilities should).

Exercise 6. (OTP)

It was mentioned in lecture that a one-time pad (or more generally, a stream cipher) should never be reused to encrypt multiple messages – thus the name "one-time".

In this exercise you are asked to decrypt messages that were encrypted using ***the same*** *OTP.*

About the plaintext messages: Each message is an English sentence, using English letters, punctuation, symbols and numbers. The start and the end of each message may be in the middle of a word. The used unicode encoding form is 8-bit - each character is represented by one byte, i.e., two hexadecimal digits ranging from `0x00` to `0xFF` (only a subset of these represent English letters, punctuation, symbols, digits, etc.). You can find a table of 8-bit unicode encoding here. Also, recall that in python the function `chr()` maps integers to characters, while `int()` performs the reverse.

About the generated ciphertexts: A collection of 16 hex-encoded ciphertexts can be found on `https://learning.unisg.ch/courses/21231/files/3014546?wrap=1`. These are the result of encrypting 16 plaintext messages with the same one-time pad, which was generated at random and added to the plaintexts by bitwise XORing. Each ciphertext appears on a separate line. The generation script can be found on `https://learning.unisg.ch/courses/21231/files/3014518?wrap=1`.

Complete the following tasks:

(i) Suppose you are unsure if all messages are XORed with the same one-time pad. Devise a method (a *sanity check*) which can give you more confidence that this is indeed the case, and apply it to the above messages.

Hint: Consider the range of values you get when XORing characters.

(ii) Decipher the ciphertexts and extract the secret key.

Hint: One pausible approach is to perform the following:

1. First, think on the following questions: *What is the relation between characters of different sentences at the same location? Which types of characters occur the most frequently? Is it reasonable to guess that in most locations all characters of different sentences come from a small set of frequent characters?* Then, write a script that creates a set of valid key guesses, guessing each byte of the key individually, one at a time.
2. For key bytes with only one possible value, insert those values into the key and decrypt sections of the ciphertexts. How accurate and thorough is this initial guess?
3. Refine your key byte guesses by checking which values lead to coherent plaintexts across all ciphertexts.

Help - Additional Information

Question 1: What is "mod"?

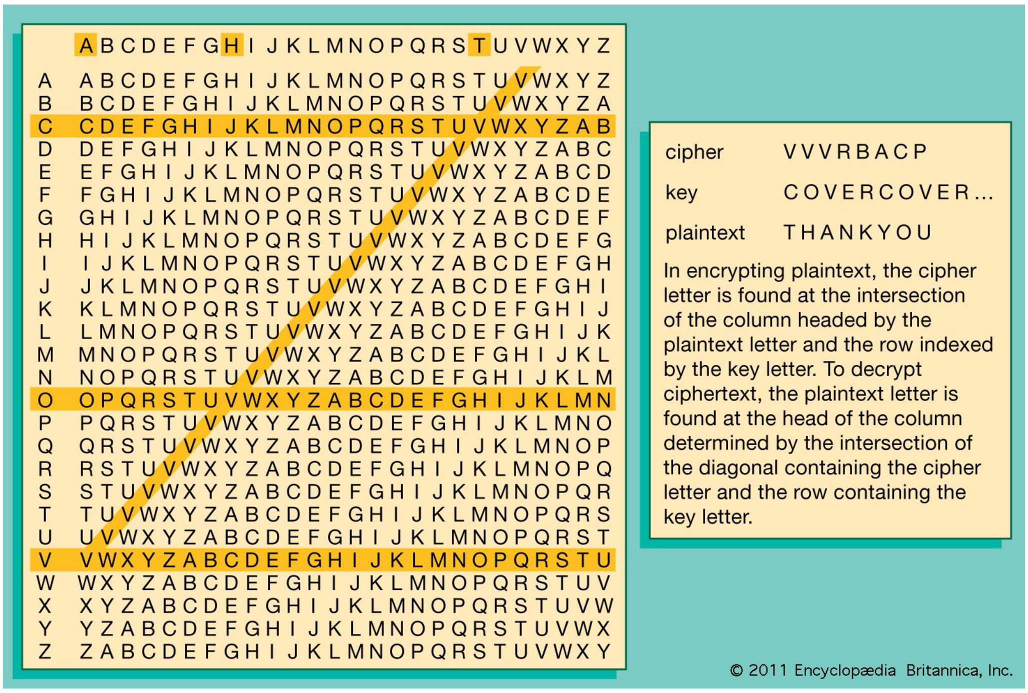
Suppose a is an integer and n is a positive integer.
We define $a \bmod n$ to be the remainder when a is divided by n. We call n the *modulus*.

For example, $21 \bmod 5 = 1$ since $21 = 4 \cdot 5 + 1$.

More details on the concept can be found on Section 2.3 of Stallings' "*Cryptography and Network Security: Principles and Practice*". We will return to this concept and explore it in greater detail in future lectures.

Question 2: What is the Vigenère cipher?

When encrypting with the Vigenère cipher, each letter of the plaintext is encrypted using a different shift cipher with a number of shifts determined by the corresponding key-letter. If the key is smaller than the plaintext, then it is repeated until the whole message is covered.
- The Vigenère cipher is more resistant to frequency analysis than simpler ciphers, but it can still be broken with modern cryptographic techniques.
The encryption-decryption processes are nicely explained by the following illustration by Encyclopædia Britannica:



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

cipher	VVVRBACP
key	COVERCOVER...
plaintext	THANKYOU

In encrypting plaintext, the cipher letter is found at the intersection of the column headed by the plaintext letter and the row indexed by the key letter. To decrypt ciphertext, the plaintext letter is found at the head of the column determined by the intersection of the diagonal containing the cipher letter and the row containing the key letter.

© 2011 Encyclopædia Britannica, Inc.

More details on the concept can be found on Section 3.2, "Polyalphabetic Ciphers" of Stallings' "*Cryptography and Network Security: Principles and Practice*". Also, this 3-minute video provides a nice explanation of the cipher, https://www.youtube.com/watch?v=zNO4PTlg62k.