Homework 2 - Sage Basics and Prehistoric Crypto

Cryptography and Security 2017

- You are free to use any programming language you want, although SAGE is recommended.
- Put all your answers **and only your answers** in the provided SCIPER-answers.txt file. This means you need to provide us with Q_{1a1},Q_{1a2}, Q_{1b},Q_{1c}, Q_2, Q_3, Q_4a , Q_4b, Q_4c and Q_5. You can download your **personal** files on `http://lasec.epfl.ch/courses/cs17/hw2/index.php`
- The answers Q_{4b} and Q_5 should be ASCII sentences, the answers Q_{1a1}, Q_{1a2}, Q_{1b}, Q_2 and Q_3 should be integers, and the answers Q_{1c}, Q_{4a} and Q_{4c} should be lists of integers (in the format $[a, b, \ldots, z]$). **Please provide nothing else. This means, we don't want any comment and any strange character or any new line** in the .txt file.
- We also ask you to submit your **source code**. This file can of course be of any readable format and we encourage you to comment your code. Notebook files are allowed, but we prefer if you export your code as a textfile with a sage/python script.
- The plaintexts of most of the exercises contain some random words. Don't be offended by them and Google them at your own risk. Note that they might be really strange.
- If you worked with some other people, please list all the names in your answer file. We remind you that you have to submit your own source code and solution.
- We might announce some typos/corrections in this homework on Moodle in the "news" forum. Everybody is subscribed to it and does receive an email as well. If you decided to ignore Moodle emails we recommend that you check the forum regularly.
- The homework is due on Moodle on **Tuesday the 17th of October** at 22h00.

Exercise 1 Algebraic Computations

This exercise aims practising some computations which you may need to use frequently during the next homeworks.

First, consider the matrix multiplication below which is a representation of a system of linear equations. In your parameter file, you will find integers $m_{11}, m_{12}, m_{13}, m_{14}$ and y_{11}, y_{12}. Find the integers Q_{1a1} and Q_{1a2} and write them in your answer file:

$$\begin{bmatrix} m_{11} & m_{12} \\ m_{13} & m_{14} \end{bmatrix} \begin{bmatrix} Q_{1a1} \\ Q_{1a2} \end{bmatrix} = \begin{bmatrix} y_{11} \\ y_{12} \end{bmatrix}$$

Next, take the integers Q_{1a1} and Q_{1a2} you have just recovered, and compute the integer $Q_{1b} = \gcd(Q_{1a1}Q_{1a2})$ and write it in your answer file.

Last, solve the linear congruence $w_1 \cdot z_1 \equiv u_1 \pmod{Q_{1b}}$ for the unknown z_1 (where the modulus is the value Q_{1b} you have just computed). The integers w_1 and u_1 are given in your parameter file. Write all solutions for $z_1 \in \mathbb{Z}_{Q_{1b}}$ (if any exists) in your answer file under Q_{1c} as a list of integers in the ascending order (e.g., if there are 4 solutions $z_1 \in \{9, 7, 8, 1\}$ write $Q_{1c} = [1, 7, 8, 9]$, if there is no solution $Q_{1c} = \mathsf{None}$).

Exercise 2 Order of Element

In your parameter file, you will find six integers n_2, p_{21}, p_{22}, q_{21}, q_{22} and g_2 where $g_2 \in \mathbb{Z}^*_{n_2}$, $n_2 = p_{21}q_{21}$, $p_{21} = 2p_{22} + 1$ and $q_{21} = 2q_{22} + 1$. Find the order of g_2 in $\mathbb{Z}^*_{n_2}$ and write it under Q_2 in your answer file.

Exercise 3 Easy Diffie-Hellman Group

After the lecture of Cryptography and Security, Alice and Bob decided to use the Diffie-Hellman key agreement protocol that they saw on slide 138 during the lecture to exchange a secret key which will be used to share their answers of future homework. Since they were believing that the computational Diffie-Hellman problem is hard for *any* group, they decided to use $(\mathbb{Z}_{n_3}, +)$.

However, Eve was hearing their discussion and she obtained the generator g_3 they used, the public messages X_3 and Y_3 that Alice and Bob exchanged, and the modulus n_3 by eavesdropping the communication between Alice and Bob. Now, you are Eve.

In your parameter file, you can find four integers g_3, X_3, Y_3 and n_3 where $g_3, X_3, Y_3 \in \mathbb{Z}_{n_3}$. Compute the output Q_3 of the DH key exchange between Alice and Bob and write it in your answer file.

Exercise 4 The Adventures of the Crypto-Apprentice: Simple Substitution

A young apprentice cryptographer just enrolled in the course Cryptography and Security. He was thrilled to learn everything he could about cryptography; after all, it is the coolest subject in the universe and beyond.

After the first lecture, the apprentice was studying the course slides. Eager to actually *use* some crypto, the apprentice did not finish the whole chapter, but stopped on slide 25. He thought: "Wow, the number of possible substitution tables is so HUGE! This cipher is surely secure!"[1] And so, he decided to use the simple substitution cipher.

Before starting with the cipher itself, the apprentice implemented a mapping to encode the standard 26 lowercase letter alphabet plus the space symbol as integers. The letter 'a' is mapped to 0, 'b' to 1, ..., 'z' to 25, and the space character ' ' to 26. Write a function `encode` that maps an element from the alphabet to the corresponding integer and a function `decode` that given an integer returns the corresponding element from the alphabet. Using the first function, encode $word_4$ which you can find in your personal parameter file to obtain a list of integers that you should write under Q_{4a} in your answer file. Using the second function, decode the array $encoded_4$ and write the solution under Q_{4b}.

[1] It's a pity that he did not read a couple more slides.

Hint: Have a look at the python functions "ord()", "chr()", and "String.join()".

Happy with his encoding functions, the apprentice turned his attention to the substitution cipher. He first sampled a random permutation $Q_{4c} : \mathbb{Z}_{27} \to \mathbb{Z}_{27}$ that he used as the secret key. Then the crypto-apprentice took a (long) extract $text_4$ from one of his childhood books, stripped it of marks, and encoded it as a list of integers $ints_4$ using the function encode.[2] He then encrypted $ints_4$ using the permutation Q_{4c}, obtaining a ciphertext C_4. Concretely, he computed the list of integers C_4, such that for each $i = 0, \ldots, \mathsf{length}(ints_4) - 1$ we have $C_4[i] = Q_{4c}(ints_4[i])$. He then sent the ciphertext to a fellow crypto-nerd.

You have intercepted the ciphertext. Moreover, the apprentice printed the secret key on paper, and then threw it away. You found the paper, but can only make out mapping of some letters. You decided to recover the secret permutation Q_{4c} to give a friendly lesson to the apprentice.

In your personal file, you will find the ciphertext C_4, and the partial key $hint_4$; it is an incomplete version of Q_{4c} with some of the images replaced by -1. Recover the complete secret permutation Q_{4c} and write it in your answer file as a list of integers. (I.e. if $Q_{4c}(0) = a$, $Q_{4c}(1) = b$, $Q_{4c}(0) = c$ and so on, then $Q_{4c} = [a, b, c, \ldots]$.)

Hint: You can work with "data" (wink-wink) given in the course slides. Also, the space character is more frequent than any letter.

Exercise 5 The Adventures of the Crypto-Apprentice: Badly Generalized Vernam

After the blunder with the simple substitution cipher, the crypto-apprentice discovered (while reading the rest of the first chapter of the Cryptography & Security course) that the Vernam chipher provides perfect secrecy. He was attentive enough to notice that the perfect secrecy is only achieved, if each binary plaintext is encrypted with a fresh, uniform binary key of the same length. The apprentice also saw, that Vernam cipher can be generalized to work over larger alphabets than simple bits. The apprentice decided that he will send a message encrypted with the generalized Vernam cipher to a friend. Except he didn't really pay attention to all the details that are needed for the generalized Vernam to be secure.

The apprentice decided to use the set $\mathbb{Z}_{27}$ as alphabet for his generalized Vernam cipher. He composed his message Q_5, and encoded it into a list of integers $ints_5$ using the function encode from Exercise 4.[3] Remembering that the Vernam cipher required a key composed of independently uniformly distributed bits, the apprentice then sampled a uniform binary key k_5, such that k_5 is a list of $\mathsf{length}(ints_5)$ integers where every element $k_5[i]$ is sampled uniformly and independently from the set $\{0, 1\}$. Finally, the apprentice cryptographer computed the ciphertext C_5, which is a list of integers such that $C_5[i] = ints_5[i] + k_5[i] \bmod 27$ for $i = 0, \ldots, \mathsf{length}(ints_5)$.

You have intercepted the ciphertext, and are determined to recover the plaintext. You make a (very good) guess, that the message only contains "normal" English words, which are all listed in the file `dictionary.txt` provided by us, and spaces. In your parameter file, you will find the list of integers C_5. Recover the original plaintext Q_5 and write it in your answer file. (This means, that you have to provide an English phrase in ASCII characters!)

Hint: The phrase Q_5 is as meaningful/grammatically correct, as it gets.

[2] Thus the extract $text_4$ is composed of lower-case alphabetic characters and spaces only.

[3] Again, the message Q_5 only consists of lower-case alphabetic characters and spaces.