CSE350/550: Network Security (Jan-May 2025)

Programming assignment no. 1 (due 11.55 pm, FRI Jan 31, 2025)

Listed below, you will find brief description of 3 projects, numbered 0 through 2. In groups of 2 you are required to:

a. pick one project (see algorithm below for you to pick a project),
b. complete that project, and
c. submit a report (with a working system) before **11.55 pm, FRI Jan 31, 2025**. The outcome will be evaluated by me and the TAs in an oral presentation that you will make.

Further:

a. You may use any programming language that you are comfortable with, including C, C++, Java, Python, etc. on any platform, Linux, MS Windows, etc., and
b. Do not copy your assignment from another group, or allow others to copy your assignment – be aware it is easy for us to find out (it will also show up in the oral presentation you will make).

Algorithm to pick a project: pick project numbered 0, 1, or 2 as determined by k = (A1+A2) mod 3, where
A1 = last_4_digits_of_entry_no_of_first_student, and
A2 = last_4_digits_of_entry_no_of_second_student.

The submission will consist of four parts:

1. a 2- to 4-page Word or pdf document describing the system you have designed,
2. sample inputs and/or outputs from running the code you have written,
3. the code itself as a separate file, and
4. 5 to 8 slides that you will use to present your work during your presentation to me & TAs.

In each of the projects listed below:

a. You are required to develop executable programs to encrypt, decrypt, and more importantly launch brute-force attack to discover the key,
b. The difference between the three projects are (i) the character set and the manner of crafting the plaintext, (ii) the specific encryption/decryption algorithm you will use, and (iii) the method for launching a brute force attack,
c. In Project 0 the character set is Ω = {A, B, C, D, E, F, G, H} = {000, 001, 010, …, 111}. In Project 1 and Project 2, the character set consists of lower-case English letters, viz. Ω = {a, b, …, z}. In all cases the plaintext you will work with should be "recognizable". To make the text recognizable, the plaintext p should satisfy some property, π, that can be checked by an algorithm or a program that may declare whether π(p) = true or false. This property π can take different forms. **The simplest form where plaintext == original_text ||original_text, where || is the concatenation operator, will not work**. Therefore, you should identify or construct a good hash function Hash(.) that you may use to construct a plaintext, p = (string, Hash(string)), where "string" is the original text. The Hash function should be such that the received or decrypted string can be recognized by an algorithm or program.
d. In all projects, you will:
    i. Prepare 5 plaintexts as above, p = (string, Hash(string)), that can then be encrypted to obtain corresponding ciphertexts using the given & known key;
    ii. The ciphertexts will be decrypted one-by-one using the (**same**) known key to compute the plaintexts. You should then visually inspect the resulting plaintexts to be sure the encrypt and decrypt algorithms/programs are working fine.
e. Limit the length of plain-text as well as size of keys suitably so that the computations can be carried in reasonable time. Necessarily the length of plaintext and the key will be guided more by how much computer time will be used when you launch a brute force attack.
f. Finally, in all projects you will launch a brute force attack by trying out various keys, k1, k2, k3, etc. and decrypt (starting with) the first ciphertext. Once you discover a key say k4 such that the resulting plaintext satisfies the property π, use the same key k4 to decipher the second ciphertext and check if the resulting plaintext also satisfies the property π. If yes, continue with third ciphertexts, etc. ELSE, reject the key, k4, and

restart to discover a new key that will correctly compute all 5 plaintexts (and satisfy property π in all 5 cases).

<u>Project 0:</u> Encryption & decryption using mono-alphabetic substitution of plaintexts where p is formed using symbols from Ω = {A, B, C, D, E, F, G, H} = {000, 001, 010, …, 111}. Encryption uses a table of 8 rows formed using a specific permutation of symbols in Ω, and decryption uses a table that is the inverse of the encryption table. Test your encryption and decryption software on the 5 different plaintexts p of kind p = (string, Hash(string)). Then develop the software to launch a brute-force attack to discover the key. The plaintext should be long enough.

<u>Project 1:</u> Assume Ω = {a, b, …, z}. Encryption & decryption using **poly-alphabetic substitution** of the kind discussed in class. Test your encryption and decryption software on the 5 different plaintexts p of kind p = (string, Hash(string)).Then develop the software to launch a brute-force attack to discover the key. Assume that the key length is known, and is of length 4.

<u>Project 2:</u> Assume Ω = {a, b, …, z}. Encryption & decryption using **transposition** of the kind discussed in class. Test your encryption and decryption software on the 5 different plaintexts p of kind p = (string, Hash(string)).Then develop the software to launch a brute-force attack to discover the key. Here assume that the key length is known to be 9 or less.