Computer Security Project(s), SoSe 2023

Project 1

(14.99 points)

In what follows, you $\simeq$ your group. Each of you is going to play the roles of Harry/MJ and Bully. As Harry/MJ, you are going to implement a cipher that looks like one-time-pad (OTP). As Bully, you will try to recover or corrupt secret messages with a ciphertext(CT)-only attack.

(i) You will write a program to generate a list of 1024-bit (128 Byte) keys that you keep **secret**. Each key is to be generated in the following fashion: first compute a (random) 16 bit (2 Byte) key k_0. Let k_1 be the 1-bit cyclic shift of k_0 to the right. Let h_0 and h_1 be the SHA-512 hashes of k_0 and k_1, respectively. The concatenation $h_0 \,||\, h_1$ is your key.[1] You may use any cryptographic library of your choice (e.g. OpenSSL).

Compose 10 Plaintext (PT) messages that Harry and MJ might want to communicate and store them in files Plaintext-i.txt (for $i = 1 \dots 10$). Each message should be limited to at most 128 characters. Only english characters are allowed (punctuation allowed).

Now augment your program to carry out OTP based encryption and decryption. In encryption mode, your program should take as input a ($\leq$ 128 char) PT and a 1024-bit key, and compute the CT. Likewise for decryption.

You may convert each message/key to binary or hex using ASCII/UTF to faciliate the XOR operation.

- Encrypt seven PT messages (say $m_1, \dots, m_7$) using OTP, each with a unique key (say $k_1, \dots, k_7$).
- Pick any three messages from the above set $\{m_1, \dots, m_7\}$ and re-encrypt again, each time with a new unique key (k_8, k_9, k_{10}).
- For the remaining three messages m_8, m_9, m_{10} encrypt while re-using one key from the set $\{k_1, \dots, k_7\}$ for each message.

Shuffle the CTs. Re-encode the binary cipher to readable format. Store the results in 13 files, naming them as Ciphertext-i.txt (for $i = 1 \dots 13$). Collect your source code, keys, Plaintexts, Ciphertexts into a tar/zip[2] As a sanity check, run decryption on your opus of CTs and make sure you recover the PTs.

(ii) You will be given the zip from some random group containing CTs and you have to play Bully now. Try to decrypt the set of CTs with your own program, any way you can. The brute-force method would go through each of the 2^{16} keys. Can you do better than brute force? Collect your results in Decrypt-Ciphertext-i.txt (...).

(iii) As Bully, you want to put dirt in the messages. Hopefully, you have recovered some of the PTs and/or keys. Irrespective, manipulate some of the CTs meaningfully and collect your results as Ciphertext-i-dirty.txt (...). Later, the encrypting group will decrypt your modifications.

You must respect the input/output formats.

[1] SHA-512 produces a 512 bit output. By concatenating two such outputs, you get a 1024 bit key. Of course, these aren't ideal keys for OTP (think why?) but will suffice for us. Nobody can brute force 2^{1024} keys in part (ii).

[2] Instructions on using programming languages, submission details, etc come later.