CS5285: **Information Security for eCommerce** Autumn, 2018
Problem Set 1 (Due Date: December 2 – 22:59)
Total: 100 points Submit Q1–Q6 hardcopy into my mailbox (typed or written, AC1 6th Floor, Yellow Zone, opposite entrance to CS Department)
Submit Q7–Q8 as instructed in the question.

Questions:

1. **Password File** (2-2-2-2 points):

   (a) In a system, each user has an entry in the system's password file: (y; s) where y is computed as follows and s is a salt.
   i. $y = H(s; password) \oplus password$
   ii. $y = H(s) \oplus H(password)$
   iii. $y = E_s(H(password))$ where E is AES algorithm
   iv. $y = MAC_{password}(s)$ where MAC is CBC-MAC constructed with AES
   List which of the methods above for computing y are effectively secured by the salt against precomputed dictionary attack. Assume that s is adequately long and random.

2. **Replay Attack** (2-2-2-2 points): Determine which of the following protocols are insecure against replay attack, and explain why. h is a hash function.

   (a) $A \rightarrow B$: username, password

   (b) $A \rightarrow B$: username, h(timestamp, password)

   (c) $A \rightarrow B$: username, timestamp, h(password)

   (d) $A \rightarrow B$: username, h(password), h(timestamp)

3. **SSL** (5-5-5 points): Consider the following attacks to internet security and describe how each is countered by a particular feature of SSL. Assume the size of the secret key used by SSL is 128 bits.

   (a) Brute-Force Cryptanalytic Attack: An exhaustive search of the key space for a conventional encryption algorithm.

   (b) Known-Plaintext Dictionary Attack: Many messages contain predictable plaintext, such as the 'HTTP GET' command. An attacker constructs a dictionary containing every possible encryption of the known-plaintext message. The attacker can look up the ciphertext in the dictionary to determine the secret key.

   (c) Replay Attack: Earlier SSL messages are replayed.

4. **Digital Certificates** (2-2-2-2):

   (a) Find out who issued the certificate for `https://mail.google.com` and how long the certificate will be valid.

   (b) Find out or estimate how many certificates (approximately, no need to count them explicitly) your browser contains.

   (c) What is the significance of a CA certificate being contained in the browser?

   (d) The identity of the certificate for the question above is a DNS hostname. Certificates can also be used for signing and encrypting email. For a certificate used for email, what identifier would be used as the identity in the certificate?

5. **IKE** (10-10 points):

   (a) Consider the following key exchange protocol which is similar to IKE Phase 1 Aggressive Mode. p is a large prime number and g is a generator of Z_p^*.

   1. $A \rightarrow B : g^a \bmod p,\ \{\text{“}Alice\text{”}\}_{Bob},\ \{R_A\}_{Bob}$
   2. $A \leftarrow B : g^b \bmod p,\ \{\text{“}Bob\text{”}\}_{Alice},\ \{R_B\}_{Alice},\ \text{proof}_B$
   3. $A \rightarrow B : \text{proof}_A$

   where

   $$\text{proof}_A = h(g^{ab} \bmod p,\ g^a \bmod p,\ g^b \bmod p, \text{“}Alice\text{”})$$
   $$\text{proof}_B = h(g^{ab} \bmod p,\ g^b \bmod p,\ g^a \bmod p, \text{“}Bob\text{”})$$

   Identify the components of challenge and response for A to authenticate B and the components of challenge and response for B to authenticate A. Then modify the protocol so that R_A and R_B can be eliminated without weakening the security of the protocol. In your modification, no additional message protocol, secret keys or signature can be used.

   (b) Consider the following simplified IKE Phase 1 in Aggressive Mode.

   $$A \rightarrow B :\ \text{“}Alice\text{”}, \text{“}Bob\text{”}, g^a \bmod p$$
   $$A \leftarrow B :\ \text{“}Bob\text{”}, \text{“}Alice\text{”}, g^b \bmod p, [g^a \bmod p]_B$$
   $$A \rightarrow B :\ \text{“}Alice\text{”}, \text{“}Bob\text{”}, [g^b \bmod p, g^a \bmod p]_A$$

   $[X]_A$ denotes a signature on message X generated by A. The session key established between A and B is $g^{ab} \bmod p$. Show that this simplified version allows an adversary to impersonate A and/or B.

6. **Password Files** (4-4-4-4 points): Use hashcat to execute a brute-force search to recover the passwords from the three files provided (give some screenshots to prove that you did calculate the answer): file1.txt, file2.txt, file3.txt
   You can download hashcat here: http://hashcat.net/hashcat/
   It is a command line program (no GUI), so you if you would like to run it in Windows you need to do so from the command prompt.

   The new version of hashcat you need to specify the processor option. You can use the -I option to see compatible processors, then use -D option to specify using CPU (not GPU). You can also download and use this older executable file if the new version does not work: `https://www.dropbox.com/s/xkwa3mh7h8xuc1z/hashcat-cli64.exe?dl=0`

   General instructions for using hashcat
   `http://hashcat.net/wiki/doku.php?id=hashcat`

   Details on brute force attack here:
   `http://hashcat.net/wiki/doku.php?id=mask_attack#example1`

   Information about the password files

   - File1 and file2 has six 5-character passwords (from the set A-Z,a-z,0-9). MD5 is the hash function used.
   - File 3 is the same but has six 6-character passwords
   - File1 uses the same salt for all entries (all the salt values are the same, equal to 0 so the stored value is h(0,pwd))
   - File2 uses a 8-bit salt (the stored value is h(s,pwd))
   - File3 uses the same salt for all entries (the same as file 1), but the password is now 6 characters long.

   While hashcat is running you can press [s]tatus and it will show you a progress update.

   (a) Recover the passwords in file1 and file 2
   (b) Recover the passwords in file 3
   (c) How much longer should it take to recover file2 compared to file1. Did the result support the theory?
   (d) How much longer did it take to find the 6 character passwords?

7. **PGP eMail** (5-5-5 points): In this problem we will ask you to familiarize yourself with PGP and ultimately send an encrypted and signed message using PGP to the tutor (see below for more details).
   Unless you are already familiar with using PGP, we suggest you use Thunderbird (a free email client) and Enigmail to accomplish this. The setup of this will require the following steps:

   - Download and install GnuPG (a free and open-source implementation of PGP).
   - Download and install Thunderbird
   - Download and install the Enigmail plugin for Thunderbird (a plugin to enable the use of PGP in Thunderbird)

   You can find more detailed instructions on the installation process in the Quick Start guide of the Enigmail plugin here: `https://www.enigmail.net/documentation/Installation_of_Enigmail`.
   Once you have successfully setup the required software, please complete the following assignments:

   (a) Create your own PGP keypair by using the Enigmail plugin in Thunderbird (Menu "OpenPGP" → "Key Management" → "Generate" → "New Key Pair"). If you have used PGP before and already have a key pair, you can skip this step.

   (b) Import the course public PGP key into Enigmail. You can find the key tsgexercise_cert.asc on the following URL:
   `https://www.dropbox.com/s/txqjtdyy613iorx/tsgexercise_cert.asc?dl=0`

   (c) Send an email to email address tsgexercise@gmail.com, encrypted with the course public key and signed with your newly created key. The subject of the email should include "CS5285 PS2 EncryptedMail XXXXXXXX" where "XXXXXXXX" is your 8-digit student id. Make sure you include your public key as an attachment in the message (Enigmail allows to easily send public keys by going to "OpenPGP" → "Key Management", then right-click on a key and select "Send Public Keys by Email").

   NOTE: The points for tasks (a) and (b) will be awarded when completing task (c), so make sure to complete task (c).

8. **PGP Web of Trust** (5-5 points): In this question we assume you have created your own PGP key pair as described in question 1. Here we will ask you to complete several tasks to illustrate the web of trust in the trust model of PGP (Anarchy model).
   In PGP, you certify that you trust a particular key by signing the key with your own private key, after making sure that it is in fact the right key (by verifying the fingerprint of the key together with the owner of the key).

   (a) Have your PGP key certified mutually (i.e., you certify their key and they certify your key) by at least one other participant of the course CS5285. The best way to do this is as follows:

   - Lookup the fingerprint of your PGP key. Using Enigmail go to "OpenPGP" → "Key Management", then double-click on your own key and look for the field "Fingerprint". Write down the fingerprint on a slip of paper.
   - Send your public key to the person with whom you are going to certify your keys. The person should also send you their public key, so that you have a copy of their public key on your computer.
   - When you meet the person the next time (or at any time of your convenience), give the slip of paper with the fingerprint of your key to said person. You should also receive a slip of paper with the fingerprint of their key.
   - Using Enigmail on your computer, verify the fingerprint of the key from your partner with the fingerprint on the slip of paper they gave you. If the fingerprint is correct (*and only then*), certify their key by signing it with your own key. You can do this by going to "OpenPGP" → "Key Management" then right-click on their key and select "Sign Key".
   - Because the signature only exists on your computer so far, you have to send the public key of your partner back to them. You should also receive your own key from your partner together with the new signature made by your partner. Use Enigmail to import it back on your own computer, which will import the signature made by your partner.

   If you like, you can repeat this step and mutually certify keys with other people in the course CS5285, but make sure that you always verify the fingerprint of a public key before signing it.

   (b) Once you have finished certifying your key (with at least 1 partner) and they have sent your public key back to you, so that the signatures made by other people showup when examining your own key on your computer, send your public key (to email address tsgexercise@gmail.com). As the subject of the email include "CS5285 PS2 PGPKey XXXXXXXX" where "XXXXXXXX" is your 8-digit student id.

   NOTE: Unless you complete task (b) by sending your key signed by a partner to the tutor, no points can be awarded for task (a)!