Information Security Technologies COMP607
Assignment 1 (20%)

Instructions:

1. Type or write your answers neatly on A4 paper. The assignment must be done in English. It must be your own work. Show all working. Do not copy material from anywhere without appropriate referencing of the source as it will be penalised.
2. Where appropriate, take screenshots of your work and paste into your submission.
3. Make sure you write your name, student ID, and the question number on your submission.
4. Begin each question on a fresh page, if possible.
5. Marks will be deducted for untidy work.
6. Date due: refer to Canvas
7. Submit your answers (in pdf format), files, etc. in *https://canvas.aut.ac.nz/Assignments* Do not submit as one zipped file.

Notes: Some of the questions require you do the tasks using Linux commands. If you don't have Linux, you can use the server at *scopius.aut.ac.nz*. To do this, open a Windows Powershell and at the $ prompt, type `ssh username@scopius.aut.ac.nz.` Your username is your AUT login name and password is your day of birth, e.g. *01apr*

To copy the file from your directory in the Linux server to your local PC, you can use the WinSCP application.

Where required, to access files in *https://scopius.aut.ac.nz*, username/password: *student/student.* The files are also availaible in scopius /home/pub directory.

You should also use a scientific calculator capable of doing modulo math. You can use the genius math tool in the scopius server by typing at the $ prompt, `genius`

Questions/Tasks:

1. (a) The following cipher text is obtained using a rail-fence method. Using brute force, determine the key and the plaintext message in English? (5 marks)

AAEHDSGNMBTTAOHTODESTRNOAIOIEGB

(b). Using a text editor, create a textfile called *secretLetter.txt* containing the text "*It's no use going back to yesterday, because I was a different person then*". Encrypt this file using the openssl toolset with your AUTlogin name e.g. *xyz1234* as the encryption key.

(i) Use AES 192 bit key, ECB mode, name the encrypted file s*ecretLetter_aes.enc*
(ii) Use DES OFB mode, name the encrypted file *secretLetter_des.enc*

Take screenshots of your work.
Submit screenshots and both files. (5 marks)

2. Consider a cryptosystem where the user enters a key in the form of a password.

   a. Assume a password consists of 8 letters, where each letter is encoded by the ASCII scheme (7 bits per character). What is the size of the key space? (2 marks)
   b. What is the corresponding key length in bits? (2 marks)
   c. Assume that most users use only 26 lowercase letters from the alphabet instead of the full 7 bits of the ASCII encoding. What is the corresponding key length in bits in this case? (2 marks)
   d. At least how many characters are required for a password in order to generate a key length of 128 bits in case of letters consisting of (4 marks)
      (i). 7- bit characters?
      (ii). 26 lowercase letters from the alphabet?

3. The following ASCII bits (8 bits per character) is obtained using a stream cipher to encrypt an English plaintext message. (The spaces are inserted for readability). The encryption key is a single alphabet character (8 bits in ASCII). Using frequency analysis, or otherwise, obtain the plaintext. State and explaing the weakness in the way this cipher is used. (10 marks)

```
00010111 00011111 00011111 00001110 00011011 00001110 00011111 00010110 00011111
00001100 00011111 00010100 00011011 00010111
```

4. Use md5 to check for file integrity.
   Using your browser, go to https://scopius.aut.ac.nz. Click on Cryto/Information Securities Technologies/Software directory. Use `student` for both username and password.

   There are two putty-0.70 msi files, one of which is corrupted. The md5 sum of the good file is also available.

   Download both versions of the putty-0.70 files. Find the one that is good.

   (a) Run the md5sum on the downloaded file and save the output to a file called putty.md5, e.g.
   ```
   $ md5sum puttyputty-0.70-installer.msi
   ```
   (6 marks)

   (b) Now compare your new md5 with that on the server.
   Which file is the good copy? (4 marks)

5. There are two files in Scopius server in *home/pub* folder (and also in *https://scopius.aut.ac.nz* in info directory), called n*otice1.txt* and *notice2.txt* and their related HMACs. One of the notices is fake. You know that the authentic copy was made by the person who has the secret key *comp607* shared with you. You are required to determine which copy is authentic.

   Change to your working directory and copy theses files into it.

   ```
   $ cp /home/pub/notice*.* .
   ```

(a) Display both files.

(b) Use openssl tools to determine their HMACs and determine which is the authentic one. First, generate the HMAC for each one using the shared key, *comp607*

```
$ openssl dgst -hmac sharedkey notice1.txt
```

Compare with the given HMACs, *notice1.hmac.txt* and *notice2.hmac.txt*

```
$ cat notice1.hmac.txt
```

Which notice is the authentic version? (10 marks)

6. In your Linux working directory, create a text file using one of the following:
Using a text editor such as `pico` enter or copy and paste some text into it, and and save.

```
$ pico test1.txt
```

Alternatively you can do as follows:

```
$ echo "This is some text file" > test1.txt
```

View the file:

```
$ cat test1.txt
```

(a) Obtain the md5 hash of your file: `(To get help: $ md5sum -h)`

```
$ md5sum test1.txt
```

(2 marks)

(b) Make a small change in your file such as adding a space, dot, etc.
Save the file a with a new name, e.g. "`test2.txt`" and obtain the new md5 hash.
Compare them. Are they same, different, or very different? Do this visually:

```
$ cat test1.txt
$ cat test2.txt
```

(4 marks)

(c) Make an MD5 integrity check hash for the file `test1.txt`

```
$ md5sum test1.txt > test1md5.txt
```

Verify the integrity of `test1.txt`:

```
$ md5sum -c test1md5.txt
```

(4 marks)

7. RSA algorithm. *Alice* wish to send a message *M = 999999* to *Bob* by encrypting it with *Bob's* public key. Use the following parameters to obtain the RSA keys for *Bob*. Assume that *Alice* is able to obtain *Bob's* public key securely.

a). Use $p = 677$, $q = 1409$, obtain suitable values for *Bob's* private key *d* and public key *e* exponents. What is *Bob's* public key?

What is ciphertext that *Alice* obtains by encrypting *M* using *Bob's* public key?
Show how *Bob* can decrypt this ciphertext correctly using his private key. (5 marks)

b). Use p = 1289, q =1*453*, M = 5*45454*. Calculate *Bob's* private and public keys exponents d and e. The public key exponent e should be a small number.
Bob 'signs' the message M by encrypting it with his private key. What is the 'signature' s, obtained by *Bob*?
When *Alice* obtains the message M and the signature s, show how *Alice* would verify the signature. (5 marks)

8. Describe how *Alice* and *Bob* are able to exchange a secret key using the DH algorithm.
(a) Demonstrate the process by using generator $g=2$, and prime modulus $n=$ 4787
(b) Which party, Alice, Bob, both, or none, can determine the final value of the shared key?
(10 marks)

9. The DH algorithm can also be used for encryption as well using the ElGamal scheme. Demonstrate this encryption scheme using a numerical example as follows.

Alice wish to encrypt a secret message, $M = 215$ to *Bob*. They have chosen the parammeters and private keys as follows:
Bob: private key $b = 231$, generator $G=2$, prime modulus $p = 443$.
Alice: private key $a = 198$

*D*emonstrate how the scheme works by showing what each party computes and sends to each other, showing clearly the cipher texts, and the decrypted messages.
(i) using the above numbers for M, a, b
(ii) using your own choice of numbers for M, a, b (10 marks)

10. Use OpenSSL tools to generate a certificate request and a self signed certificate. Use your own name, with correct information for country, province, city, email address, etc., with a validity period of 2 years.
(a) An X509 certificate request: *yourName.csr*
(b) An X509 self signed certificate: *yourName.crt*

Submit both *.csr* and *.crt* files. (10 marks)