CSS 322 – ASSIGNMENT 1

First name: _____ Last name: _____

ID: _____ Total Marks: _____
out of 100

Due Date: Wednesday 27 December 2006, 9am (you can hand in before the start of the lecture)

I certify that, unless otherwise acknowledged, all work carried out in this assignment is my own.

Sign Name: _____ Date: _____

Instructions

- This is an individual assignment. You are not allowed to work in groups on any part of the assignment. Plagiarism will be penalised. (You must sign the statement at the top of this cover sheet).
- The assignment must be handed in by the due date. Late assignments will receive 0 marks.
- The assignment should be neatly handwritten and/or computer generated (for example, Word).
- You must give your calculations, working out, discussion and design decisions. That is, if you only give a correct answer, but no calculations, then you will not receive full marks (in fact, you will receive very few, maybe 0 marks).
- There are many electronic (Internet) resources freely available on this topic, including software tools for calculating answers. If a question asks you to calculate an answer *manually*, then you must not use such software tools – you must complete the steps and show the calculations on paper. However, there is nothing stopping you from using software tools for checking the correctness of your manually calculated answer. If you do use a software tool, then it is your responsibility to determine if a software tool is correct.
- The assignment should be on A4 sheets, with a single staple in the top-left corner. Please do not use plastic sleeves, folders etc.
- You must attach this Cover Sheet (including name, ID and signature) to the front of your assignment.
- If questions include the writing of software, then you must hand in:
    - A printout of the source code (assuming it is less than 10 pages long – if it is longer, please see me).
    - A printout of an example showing the use of the software (e.g. a screen capture or copy of command line input and output).
    - Instructions for compiling and executing the software (include details of the platform you used, such as operating system version, compiler software and version). These instructions must be included in hardcopy in the assignment, as well as a README file with the source code.
    - An email of the executable and source code (and any necessary files, e.g. input, output, makefiles), preferably packaged with ZIP or TAR, and sent to steve@siit.tu.ac.th.
- You cannot hand in your assignment in electronic form (except for source code). For example, I will not accept an emailed Word document – I will only accept a hardcopy (printout) of the assignment.

Question 1 [35 marks]

You have a plaintext message (in binary): 0101 0011 0100 1001 0100 1001 0101 0100

Use your Student ID number to obtain a 10-bit key as follows: Key (in decimal) = ID mod 1024

a) Using Simplified DES (S-DES), and an electronic codebook mode of operation (suitable for S-DES), *manually* encrypt the first block of the plaintext message using your key. You must show and explain all steps in your encryption.

b) Explain the steps you must take to encrypt the entire plaintext message.

c) What is the ciphertext for the entire plaintext message?

d) If you used counter mode with initial counter value of 0, instead of electronic codebook mode in part (a), what is the ciphertext of the entire plaintext message?

e) Explain, with reference to the ciphertext in part (c) and (d), why the electronic codebook mode is not suitable if encrypting long messages.

Question 2 [25 marks]

You have a (single block) ciphertext message (in binary): 1000 1011 0111 1010

Use your Student ID number to obtain a 16-bit key as follows: Key (in decimal) = ID mod 65536

a) Using Simplified AES (S-AES) manually decrypt the ciphertext using your key. You must show and explain all steps in your decryption.

b) Compare AES to DES. For each of the following elements of DES, indicate the comparable element in AES or explain why it is not needed in AES.

    a. XOR of subkey material with the input to the f function

    b. XOR of the f function output with the left half of the block

    c. The f function

    d. Permutation P

    e. Swapping of halves of the block

Question 3 [15 marks]

The following questions require you to use CrypTool v1.4.00 (or latest version). CrypTool is a free download and you can install on lab computers (if not already installed) or your personal computer. If you have trouble accessing CrypTool, please see me as soon as possible.

The questions ask you to perform operations in CrypTool. You only need to include answers to the specific questions in the assignment, including makes notes of key parameter values (for example, keys used) and times. You do not have to include output from every operation.

If parameter values are not specific in the question (for example, key) then you should use your own reasonable values. For example, a key of 0 or 5555 is *not reasonable*. Choose a random key.

The following example files are used as input. You can access them from the course web page: example1.txt, example2.txt, example3.txt.

Use CrypTool to perform the following operations:

a) Encrypt (and then decrypt) the files example1.txt, example2.txt and example3.txt using the following algorithms. For each operation, in your assignment make note of the approximate time it takes as well as the key you used.

- DES with Electronic Code Book
- Triple DES with Cipher Block Chaining
- AES (Rijndael)

b) Generate the following RSA public/private key pairs, using your Last name, First name, and a PIN = 1234. Do not enter an optional key identifier. In your assignment make note of the approximate time it takes to generate keys.

- 1024 bit
- 2048 bit

c) View and print out your certificate generated with the RSA 2048 bit key pair, and include a print out in the assignment.

d) Encrypt (and then decrypt) the files example1.txt, example2.txt and example3.txt using RSA and both your 1024-bit and 2048-bit keys. Make note of the approximate time it takes to perform the operations (there is an option to "Display Encryption Time" – make sure it is selected).

e) Sign the document example1.txt using your 1024-bit RSA key and the algorithms:

- MD5 (using RSA factorisation)
- SHA-1 (using RSA factorisation)

f) List the 6 items (or parameters) included in the signed document. Explain in one or two sentences the meaning or purpose of each item. Print (or copy and paste) one of the signed documents and include in your report (Hint: you can switch between a hexadecimal view and a text view for printing in the "View" menu).

g) Verify the signature of an already signed document and explain what happens when the correct key is used (e.g. 1024 bit) and the incorrect key is used (e.g. 2048 bit). Now modify the signed document and verify using the correct key. Explain what happens and why.

Question 4 [25 marks]

a) Write a program that can encrypt and decrypt using the general Caesar cipher.

b) Write a program that can perform a letter frequency attack on the Caesar cipher without human intervention. Your software should produce possible plaintexts in rough order of likelihood.

Guidelines for programming:

- You can use any language you like (most likely the language that you are most comfortable with). I recommend the C programming language (very easy to write a short command line application, especially in Linux), or Java. You may also consider mathematical languages such as Matlab (if you have had experience with them).
- Apply the KISS principle. KISS = Keep It Short and Simple (or the less politically correct, Keep It Short, Stupid!). You will not get extra marks for fancy programs (e.g. GUIs) – it will just be a waste of your time and my time.
- Use of other people's source code, including examples on the Internet and specialised cryptographic libraries, is strictly prohibited. You must write all the code on your own.
- For the letter frequency attack, you can make use of statistics from the lecture notes or other software (e.g. CrypTool), or obtain your statistics yourself using a sample English text (CrypTool has two good examples).