CSS 322 – ASSIGNMENT 1

First name: _____ Last name: _____

ID: _____ Total Marks: _____

out of 85

Due Date: Friday 4 January 2007 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 [30 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.

Answer:

(mandatory: attach the calculations)

b) Explain the steps you must take to encrypt the entire plaintext message. (That is, you do not have to show or explain the individual S-DES calculations, only explain how you take the output of S-DES to get the entire ciphertext.)

Explanation:

c) What is the ciphertext for the entire plaintext message? (Again, the individual S-DES operations do not have to be shown).

Answer:

(optional: attach the calculations)

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? Explain the steps (you don't have to give all calculations)

Answer:

Explanation:

(optional: attach the calculations)

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

Explanation:

Question 2 [20 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.

Answer:

(mandatory: attach the calculations)

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

Explanation:

Question 3 [15 marks]

The following questions require you to use CrypTool v1.4.10 (or latest version). CrypTool is a free download (http://www.cryptool.com/) and you can install it on 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 and example2.txt.

Use CrypTool to perform the following operations:

a) Encrypt (and then decrypt) the files example1.txt and example2.txt and using the following algorithms.

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

Record (for each algorithm): the key used; the approximate time for encryption; the approximate time for decryption; explain any differences between the original plaintext and the output of the decryption (hint: View the output in Text, not just as a Hex Dump).

Answers:

example1.txt - DES with ECB

example1.txt - Triple DES with CBC

example1.txt - AES

example2.txt - DES with ECB

example2.txt - Triple DES with CBC

example2.txt - AES

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

Generation time: 1024-bit

Generation time: 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.

Answer:

(attach the certificate)

d) Encrypt (and then decrypt) the files example1.txt and example2.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).

example1.txt - Encryption time: 1024-bit

example1.txt - Decryption time: 2048-bit

example2.txt - Encryption time: 1024-bit

example2.txt - Decryption time: 2048-bit

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..

Explanation:

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.

Explanation:

Question 4 [20 marks]

a) Write a program (in C or Java) that can encrypt and decrypt using the general Caesar cipher. You can assume there are only English characters (a through to z – you can treat uppercase characters as the same as lowercase).

b) Write a program that can perform a letter frequency attack on the Caesar cipher without human intervention. Your software should produce at least 3 possible plaintexts in rough order of likelihood. You can use letter statistics from the lectures notes, from Cryptool or other sources.

Answer:

Send the source code and executables as attachments.

Explain the programming language and any special instructions for compilation below. By default, I will compile with "gcc" or "java" on the command line in Ubuntu Linux (gcc) or Windows XP (using Sun JDK).

You must name the files using your ID according to the scheme illustrated below. You must hand in the following files (assuming your ID is 4912345678):

- `4912345678_encrypt` Executable of the encryption function
- `4912345678_decrypt` Executable of the decryption function
- `4912345678_ attack` Executable of attacking function
- Plus the source code for the above files (using same naming convention).

Submit the files archived into a single file (e.g. RAR, TAR, ZIP) named as your ID. E.g. `4912345678.rar`.

The interface of the programs when run on the command line must be:

```
ID_encrypt <plaintext> <key>
ID_decrypt <ciphertext> <key>
ID_attack <ciphertext>
```

For example:

```
4912345678_encrypt steve b
tufwf
4912345678_decrypt tufwf b
steve
49123456789_attack tufwf
steve
lsdjd
wrgtg
```