Experiment 2
Information Security: Encryption/Decryption

The goal is to learn how to encrypt data with standard encrypting algorithms, with a focus on using cryptography to protect sensitive data with a secret key.

Case Study

The goal is to learn how to encrypt data with standard encrypting algorithms, with a focus on using cryptography to protect sensitive data with a secret key.

In cryptography, encryption is the process of encoding messages or information in such a way that only authorized parties can read it. Decryption is the process of taking encoded or encrypted text or other data and converting it back into text that you or the computer can read and understand. Figure 1 shows the basic cryptographic scheme.

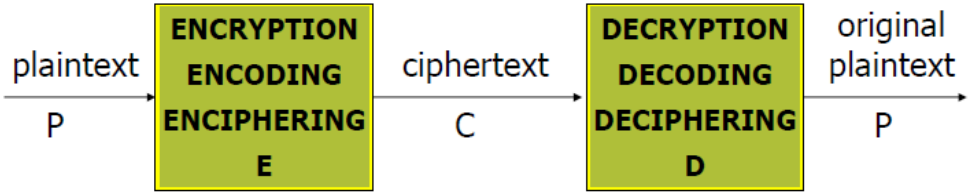

Figure 1 Basic Cryptographic scheme.

Basic types of ciphers include substitution ciphers, transposition ciphers, and product ciphers. In substitution ciphers, letters of plaintext are replaced with other letters by encryption. In transposition ciphers, the order of letters in plaintext are rearranged by encryption. Product ciphers combine two or more ciphers to enhance the security of the cryptosystem.

With regards to keys, there are three cryptosystems: keyless, symmetric and asymmetric cryptosystems. Keyless cryptosystems e.g. Caesar's cipher doesn't have keys and are less secure. Symmetric cryptosystems use the same key in enciphering and deciphering (or one key is easily derived from other). In asymmetric cryptosystems, encipher and decipher are using different keys.

Theory Background

The **Caesar cipher**, also known as a shift cipher, is one of the simplest forms of encryption. It is a substitution cipher where each letter in the original message (called the plaintext) is replaced with a letter corresponding to a certain number of letters up or down in the alphabet. In this way, a message that initially was quite readable ends up in a form that cannot be understood at a simple glance.

For example, here's the Caesar Cipher encryption of a message, using a right shift of 3.

Plaintext: THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG

Ciphertext: QEB NRFZH YOLTK CLU GRJMP LSBO QEB IXWV ALD

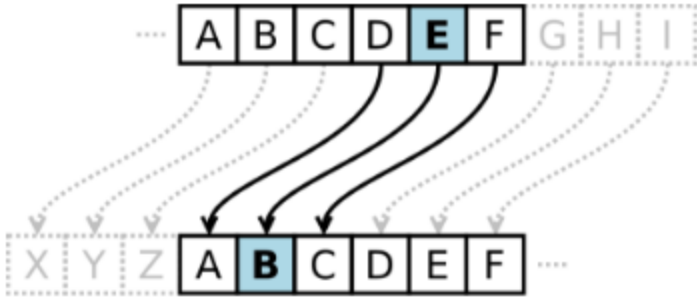

Figure 2 Caesar Cipher.

The **Advanced Encryption Standard (AES)**, also known as Rijndael (its original name), is a specification for the encryption of electronic data established by the U.S. National Institute of Standards and Technology (NIST) in 2001.

AES is based on a design principle known as a substitution-permutation network, combination of both substitution and permutation, and is fast in both software and hardware. Unlike its predecessor DES, AES does not use a Feistel network. AES is a variant of Rijndael which has a fixed block size of 128 bits, and a key size of 128, 192, or 256 bits. By contrast, the Rijndael specification per se is specified with block and key sizes that may be any multiple of 32 bits, both with a minimum of 128 and a maximum of 256 bits.

Basic ops for a round:

1) Substitution –byte level (confusion)
2) Shift row (transposition) –depends on key length (diff.)
3) Mix columns –LSH and XOR (confusion +diffusion)
4) Add subkey–XOR used (confusion)

Experiment Set-up: Configuration

This experiment will require a computer with Linux system.

You can get access to Department of ECE's Linux server through SSH clients, such as "SSH Secure Shell Client" or "Putty".

Download and install SSH client on your PC, and configure as below.

The host name is "linux.ece.ufl.edu".

The user name is your gatorlink username as shown in Figure 3. You will be required to input your password for your gatorlink.

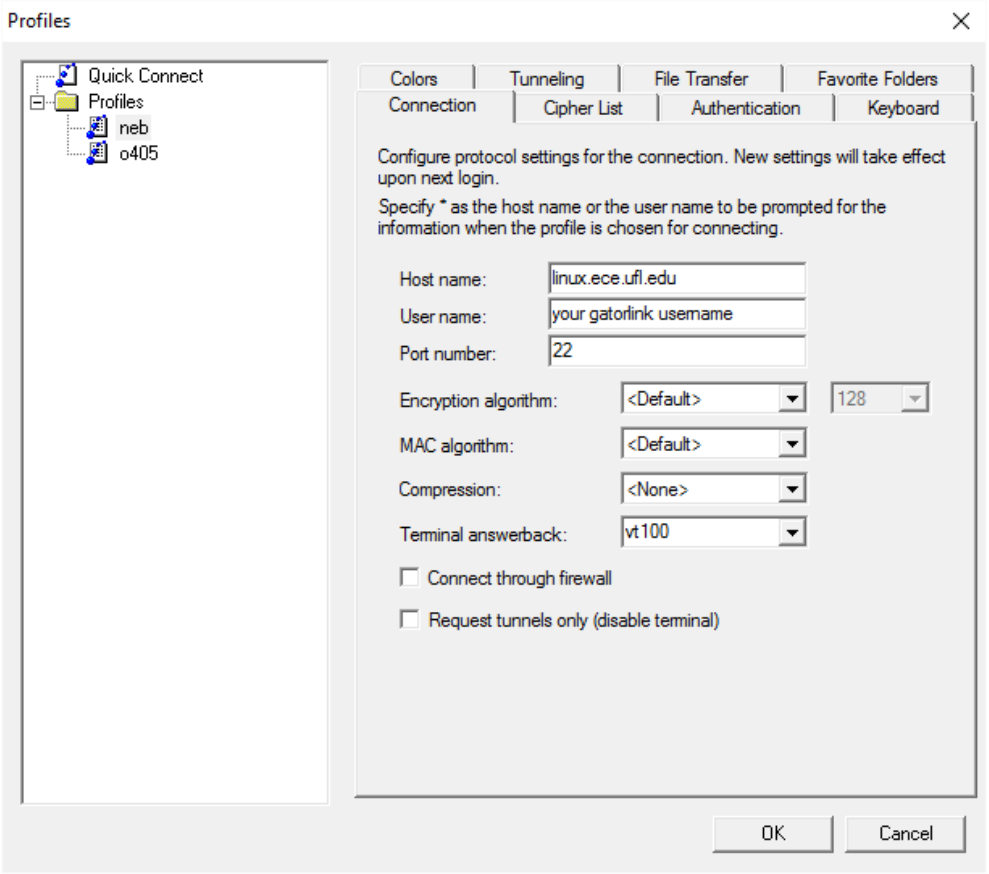

Figure 3 SSH configuration when using SSH Secure Shell Client.

Experiment Set-up: Instructions

Caesar Cipher

Write C code for Caesar Cipher algorithm. Both plaintext and the key will be the inputs. You need to execute it and write the output and answer following questions regarding it.

The important thing is that it must work for any length key and any reasonable amount of data. In a comment at the top of your code submission please write a few sentences for instructions on how to improve the Caesar Cipher.

AES Specific Challenge

1) Download code for AES
2) Copy 3 file to AES from AES code you downloaded and unzipped
   AES-> test.c, TI_aes.c, TI_aes.h
3) Compile the project using the command
   gcc –w test.c TI_aes.c
4) Run the project using the command
   ./a.out
5) Take a screenshot of the output.
6) Open test.c and go to the line starting with
   unsigned char msg[] =
7) Change the text in quotes to “gainesville” or any input 16 characters or less
8) Try compiling and screenshot the result
9) Try changing msg[] to “universityoffloridauniversityofflorida” or any other input of length more than 16 characters
10) Observe output and take a screenshot. Save it.

Measurement, Calculation, and Question

1. Turn in your code for Caesar Cipher.
2. Turn in the screenshot taken in AES Specific Challenge step 5 and answer:
    a. : What is the difference between the plain text and the ciphertext?
3. Turn in the screenshot taken in AES Specific Challenge step 8 and answer:
    a. What happens to the hex values of the characters?
4. Turn in the screenshot taken in AES Specific Challenge step 10 and answer:
    a. Does the text before and after encryption match? If not, what's the difference and why?
    b. Can you change the key length to 192 or 256 bits? Write report where will you make changes. (Hint: Number of rounds will be changed.)
    c. Write a short paragraph describing the advantages and disadvantages of both 192 and 256-bit key lengths.

Optional Follow-up

1) Modify the AES code for any size of message length. Turn in your code and screenshots when executed.

☞ Lab Report Guidelines

1. In your report, attach your code for the Caesar cipher and a screenshot when you are running it.
2. Answer all the questions in the Question part.
3. Attach all the screenshots as required in the Question part.

References and Further Reading

[1] https://learncryptography.com/classical-encryption/caesar-cipher

[2] https://en.wikipedia.org/wiki/Advanced_Encryption_Standard

[3] http://www.tech-faq.com/block-and-stream-ciphers.html

[4] http://www.ti.com/lit/an/slaa397a/slaa397a.pdf

[5] http://crypto.stackexchange.com/questions/20/what-are-the-practical-differences-between-256-bit-192-bit-and-128-bit-aes-enc

[6] http://coolshell.cn/wp-content/uploads/2010/10/rijndael_ingles2004.swf