This is the Myplace service for the 2024/25 session. For the past session, please go to classes 2023/24.

CS808: Computer Security Fundamentals

2.11: Lab: OpenSSL for AES

Done: View

OpenSSL for AES Lab Exercise

OpenSSL is an open source toolkit which provides cryptography functionality such as digital certificates and encryption. In this exercise you will explore using it for AES encryption and decryption.

Those working on Unix based systems are likely to already have OpenSSL pre-installed. You can check by opening a terminal and typing openssl. If you have it installed the terminal will display OpenSSL> before the flashing indicator for command entry.

For those on Windows, it is advised you use Guacamole. Guacamole provides access to a department Linux machine through a web browser. Using a web browser, access https://guacamole.cis.strath.ac.uk/ and log in using your University DS login. Once logged in click "Linux Labs" where you will be prompted again for your login. From there, on the left hand side, select the top icon "terminal". You will then be able to follow the instructions provided. Remember to log out once you are finished.

Steps

Upon loading OpenSSL by typing openssl into a terminal window and hitting enter, you can use the following instructions to complete encryption using the AES algorithm.

1. Create a document in the Documents folder. This can be created by opening the text editor from the left hand screen under 'Activities'. Enter text you wish to encrypt. Save the file with the name plain and extension txt in your Documents folder.

2. Type in the following command replacing abc12345 with your DS username and providing an absolute path for the encrypted output file with appropriate extensions

Use the following command to encrypt your file using AES-256-CBC:

openssl enc -aes-256-cbc -pbkdf2 -a -in /home/abc12345/Documents/plain.txt -out /home/abc12345/Documents/encrypted.ext

aes-256-cbc specifies the implementation of AES encryption we want to use: 256-bit encryption in CBC (Cipher Block Chaining) mode. The -pbkdf2 option is added to provide a more secure key derivation method, you don't need to worry much about this.

-a instructs OpenSSL to encode the result in Base64, meaning the encrypted output will be human-readable and can be opened in a text editor.

-in indicates that the following token is the name of the file to be encrypted.

-out specifies that the next token is the name of the file to be written, which will be the encrypted version.

You will then be prompted for a password – this will be used to generate the encryption key securely, recall the pbkdf2 makes the key generation more secure.

Try to open the encrypted file, which should be located in your documents folder. You should see unreadable, encoded content.

It is recommended to close the terminal window completely and open a new terminal session for the next step. This is because OpenSSL doesn't handle multiple operations well in a single session, and some students have experienced errors when attempting to complete all activities without restarting the terminal.

To decrypt the file created in step 2, open a new terminal and use the following command (adjusting abc12345 to your DS ID):

openssl enc -aes-256-cbc -d -pbkdf2 -a -in /home/abc12345/Documents/encrypted.ext -out /home/abc12345/Documents/decryptedfile.txt

-d specifies decryption mode, and -pbkdf2 ensures the key derivation matches the encryption process. The decrypted content will be written to decryptedfile.txt at the specified path.

-in indicates that the next token contains the name of the file to be decrypted

Once more you will be prompted for the password, and from this it can derive the key for decryption.

Congratulations, you have now encrypted and decrypted using AES.

As noted elsewhere, direct use of AES in its basic form is inadvisable. If making use of OpenSSL or other cryptographic libraries it is important to make sure you are fully aware of the security context.

Last modified: Friday, 27 September 2024, 11:05 AM

◄ 2.10: Article: Making Block Ciphers Usable- Message Authentication Codes

Jump to...

2.12: Cryptography Questions ►

© University of Strathclyde
You are logged in as **Neil Hutton** (**Log out**)
CS808