Networks and Systems Security, Assignment 3

Due: March 27, 2019 (2359 hrs.)
Total points: 30

The objective of this assignment is to familiarize you with using OpenSSL library routines. You would need to start with the set-up you used in assignment 1. To this you need to add some new programs, viz. `fput_encrypt', `fget_decrypt', 'fisgn' and 'fverify'.

This is what the programs are supposed to do:

1. `fput_encrypt': Very similar to 'fput', just that it encrypts the contents before storing them to the file. You need to specify the file to be encrypted. The passphrase used for encryption must the one *corresponding to the user* who calls the program. `fput_encrypt' being owned by fakeroot should be able to access the shadow password corresponding to the calling user and should dervice the key and IV, using the password as starting point.

2. `fget_decrypt': Similar to the 'fget' but decrypts and encrypted file and prints it out. One may decrypt files for which he has read privileges. The output maybe saved to a file or dumped directly to stdout. Here the key and IV should be derived from the password corresponding to the user who owns the file.

3. `fsign': This program creates a HMAC signature with the key and IV derived from the file owners shadow password. Everytime a file is created either using `fput' or `fput_encrypt' the signature should be created. The signature should be separate file with the same file name but a '.sign' extension. Again this works only for files where you have the write AND read permissions.

4. `fverify': This program verifies the HMAC signature with the key and IV derived from the owners shadow password. Everytime a file is to be read, if there is a corresponding signature file, it needs to be checked. This is also there for encrypted files which would be decrypted using 'fget_decrypt'. If the signatures mismatch then an error must be reported, else there is no need to report anything. This should work only for files for which the calling user has read permissions.

The rest of the functionality remains the same.

Grading Rubric

- Successful compilation of the programs using Makefile (5 points).
- Correct functioning of the four programs (and use of the appropriate openssl library functions) (10 points).
- Correctly handing three different corner cases. You must supply scripts/programs/inputs to test out these cases (10 points).
- Description of the how you implemented the programs, assumptions and expected outcomes (5 points).

Submission guidelines:

Deadline 1: March 27, 2019 2359 Hrs: No points deducted.
Deadline 2: March 29, 2019 2359 Hrs: 5 points deducted.
Deadline 3: March 31, 2019 2359 Hrs: 10 points deducted.
Submissions after March 31, 2019 (2359 Hrs) **would not** be considered for being graded.