COLLEGE OF ARTS AND SCIENCES
The University of Alabama at Birmingham

Department of Computer Science

CS 645/745:	Modern Cryptography
Instructor:	Dr. Yuliang Zheng

Secure Messaging

Students are required to build a software package for secure text exchange between two users using the Python Cryptography package:
https://cryptography.io/en/latest/

Part I. Sockets

Write a pair of client-server programs that use "sockets" to establish a communication channel between two users (User A and User B). Information on socket programming in Python can be found at: https://docs.python.org/3/howto/sockets.html

Make sure that

1. User A can use the client software to type in a text on a terminal window and, upon pressing the return key, send the text over the sockets to User B who uses the server software to receive the text.
2. Likewise, User B can type in a text on his terminal and press the return key to send the text over the sockets to User A.
3. your programs should work not only on the same computer but also across the network: The client and the server software can be run on different computers over a LAN or the Internet.

Part II. Message encryption & decryption

Build an encryption program that encrypts a text message using an "authenticated encryption with associated data (AEAD)" mode, and produces a ciphertext as output. Build a second program that, given an encrypted file as input, correctly decrypts it back to the original text.

This assignment requires students to the Ascon light weight cipher as the AEAD method. A Python library for Ascon AEAD encryption/decryption can be found at:
https://github.com/meichlseder/pyascon/tree/5ee786cdc8a74d9c0f7b3c81f99f5dcb5490ca00

The general form of AEAD encryption using Ascon is
$(C, T) = ascon\_encrypt(key, nonce, associateddata, plaintext, variant = "Ascon-128")$,
where $plaintext$ represents a text message. Note that the output contains 2 parts: C is the ciphertext produced by symmetric encryption, and T is an authentication tag.

Likewise, the general form of AEAD decryption is
$M = ascon\_decrypt(key, nonce, associateddata, ciphertext, variant = "Ascon-128")$,
Note that $ciphertext = (C, T)$. Also note that in addition to the key, both encryption and decryption must use the same $nonce$ and the same $associateddata$.

Decide ahead of time the following 3 data items (key, $nonce$, $associateddata$) to be used by both the client and the server.

- A shared secret key key for encrypting and decrypting a text message. It is typically generated uniformly at random or derived from a secure & random seed. It must be kept secret. The length of k is dependent on the mode to be used.
- A $nonce$. For this assignment, let's set $nonce$ to 0. It does not have to be kept secret.
- Associated data (to indicate the context). It does not have to be kept secret. For this assignment, let's use $associateddata$="CS645/745 Modern Cryptography: Secure Messaging".

IMPORTANT:
Do NOT use Python Fernet (https://cryptography.io/en/latest/fernet/) for this assignment!

Part III. Secure transport of texts from User A to User B with a shared key

This part requires students integrate what you have accomplished in the previous 2 parts. First of all, the two users share ahead of time (key_{AB}, $nonce_{AB}$, $associateddata_{AB}$). For this assignment, let's use $associateddata_{AB}$ ="CS645/745 Modern Cryptography: Secure Messaging".

- User A types in a text on a terminal window. Upon pressing the return key, the program encrypts a text using one of the 5 modes for AEAD with the shared secret key key_{AB} together with $nonce_{AB}$ and $associateddata_{AB}$. The resulting ciphertext is then sent to User B over the sockets.
- Upon receiving the ciphertext, User B decrypts it using the same mode for AEAD with the same shared secret key, nonce and associated data.

Part IV. Two way secure text transfer with shared keys

This part further improves the above programs to support 2-way secure communication. That is either user can send a text to the other user in a secure manner. It requires 2 sets of shared keys:

- (key_{AB}, $nonce_{AB}$, $associateddata_{AB}$) for secure transfer of messages from User A to User B, and
- (key_{BA}, $nonce_{BA}$, $associateddata_{BA}$) for secure transfer of messages from User B to User A,

where

- key_{AB} and key_{BA} are generated independently and hence generally different,
- $nonce_{AB}$ and $nonce_{AB}$, can be the same, such as 0, and
- both $associateddata_{AB}$ and $associateddata_{BA}$ take the form of "CS645/745 Modern Cryptography: Secure Messaging".

Submit

1) your code (zipped in one pack),
2) a report detailing
    a) how to use your programs,
    b) your strategies for the design and implementation of the program,
    c) screenshots of real time test runs of your code,
    d) lessons learned,
    e) how to further improve your code, especially in terms of convenience of use, if you revisit it in future.