Q1. **Explain how the round function of AES provides diffusion.**

The AES round function is make up of 4 different components: The Byte Substitution layer, the Shift Row layer, the Mixed Column layer, and lastly the Key Addition layer. Throughout each round, the round function achieved diffusion through the Shift Row and Mixed Column layers.

AES takes in a 16 bytes input and separates it into 4 different 4-byte blocks. We can label 16 of those individual byte A0, A1, … , A15.

Shifted Row

1. Shift Row provides diffusion by distributing members of a 4-byte block evenly across different 4-byte blocks. Therefore at the end of a Shift Row process, all individual bytes will attached to a different byte block together with members of other byte blocks.
2. Thus any changes within a block can be spread to other blocks.
3. Hence achieving diffusion across each of the 4 different blocks

Mixed Column

1. Mixed Column provides diffusion by taking the output from shift row and further spread any changes in one byte of a single 4-byte block to all of the bytes within the same block.
2. This is achieved through a linear multiplication with a pre-determined hex 4x4 matrix.
3.

$$\begin{pmatrix} C_0 \\ C_1 \\ C_2 \\ C_3 \end{pmatrix} = \begin{pmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{pmatrix} \cdot \begin{pmatrix} B_0 \\ B_5 \\ B_{10} \\ B_{15} \end{pmatrix}$$

C0 = 02 * B0 + 03***B5** + 01*B10 + 01*B15
C1 = 01 * B0 + 02***B5** + 03*B10 + 01*B15
C2 = 01 * B0 + 01***B5** + 02*B10 + 03*B15
C3 = 03 * B0 + 01***B5** + 01*B10 + 02*B15

4. If any bit in byte ‘B5’ was changed, then through the linear multiplication, the changes will be reflected in all of the ‘Mixed Column’ outputs.
5. Thus achieving diffusion within a 4-byte block.

All this partner with 10 different rounds for a 128-bits key, then any changes will be propagated to all other 4-byte blocks through the shift row and mixed column.

When a 16-byte block of plaintext is encrypted with AES, how many rounds of AES are executed before every byte in the input (the plaintext) has influenced every byte in the 16-byte block as it is being processed?

2 rounds.

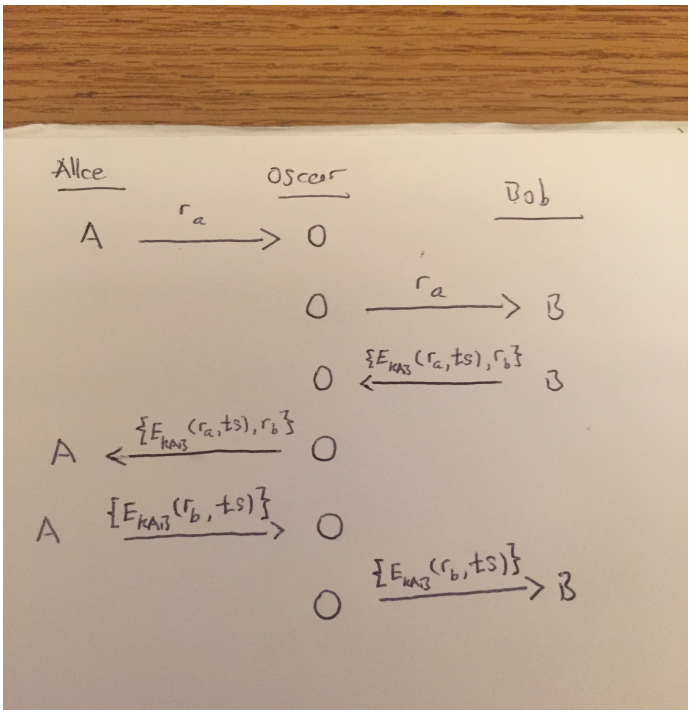
In the first round, a byte, B_i will affect all the other 3 bytes in its respective 4-byte block during the mixed column stage. Then on the next round, each of the 4 affected bytes in the block will move to separate 4-byte blocks during the shift row stage. Finally in the mixed column stage of the second round, the remaining 12-bytes will be affected by $B_{i.}$

Q2:
Suppose AES (or any block cipher) is used to encrypt data. Which mode of encryption (from lecture 1 slides/readings) is best suited for each of the following two scenarios and why? If more than one mode is suitable, pick one and explain why.

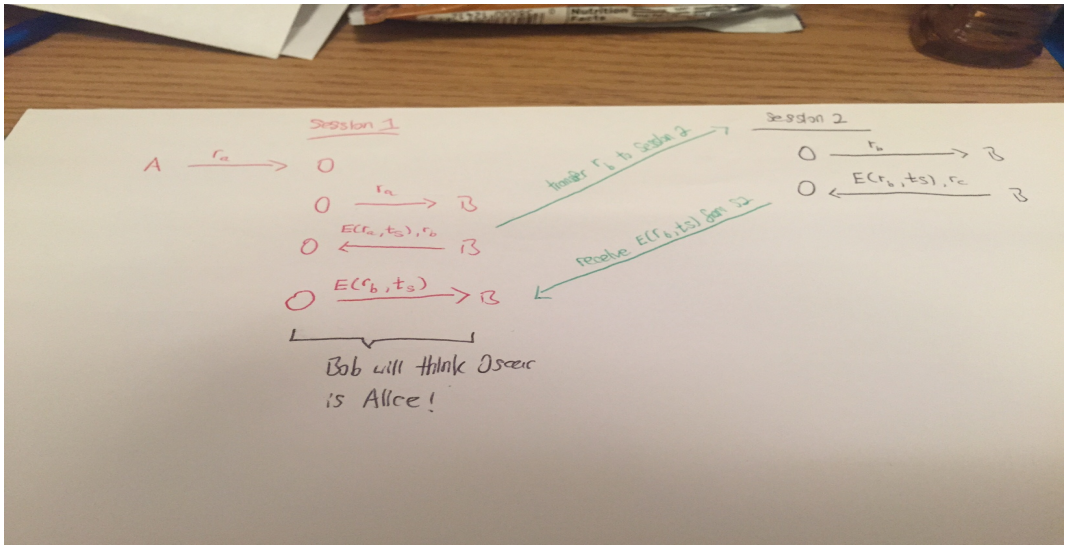
a. Streaming video between a server and client.

CTR Mode:
Since the IV needed for encryption can be generated in advance (thus faster). And any subsequent lose of packet won’t affect later decryption unlike CBC or OFB. Since most video / audio data are transmitted through the network using the UDP protocol.

Q4:

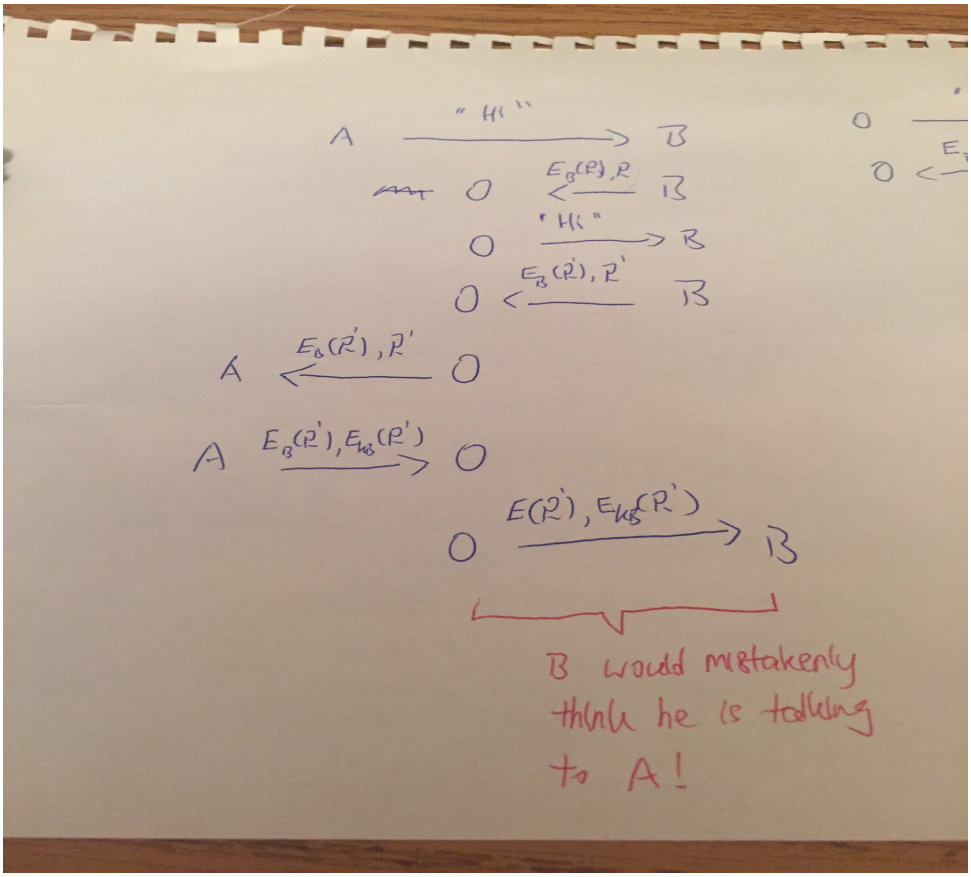


Using the protocol stated in the question, there is no way Alice and Bob can be sure they are communicating with each other. This protocol is susceptible to a “Man in the Middle” attack. An eavesdropper ‘Oscar’ can easily stand between the communication channel and manipulate the exchange. Furthermore, the eavesdropper Oscar can send Bob whatever information it wants to encrypt to Bob and injects fake communication packets to Alice.

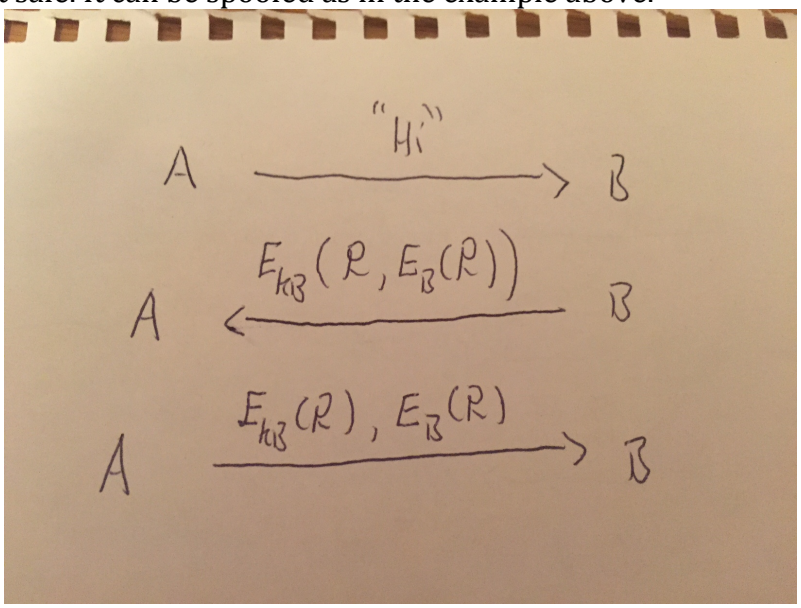


Since both Bob and Alice share a same key, All communication can be done within the encrypted channel. Instead of sending unencrypted ra and rb. Both ra and rb can be encrypted, thus denying Oscar the ability to have Bob encrypt whatever he wants.

Q5:

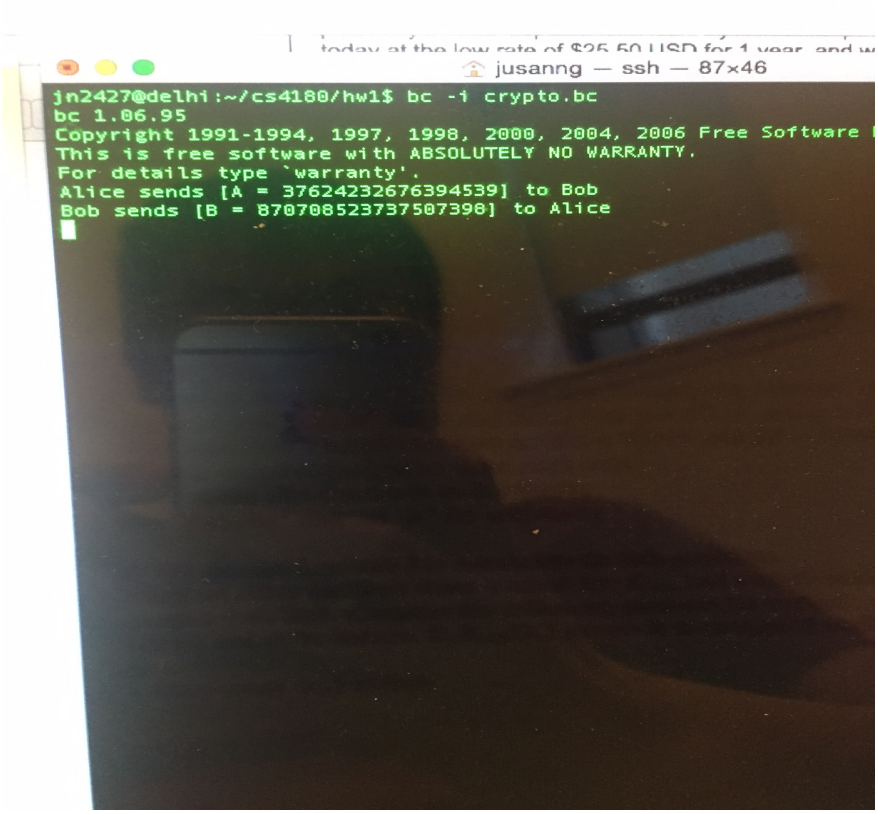


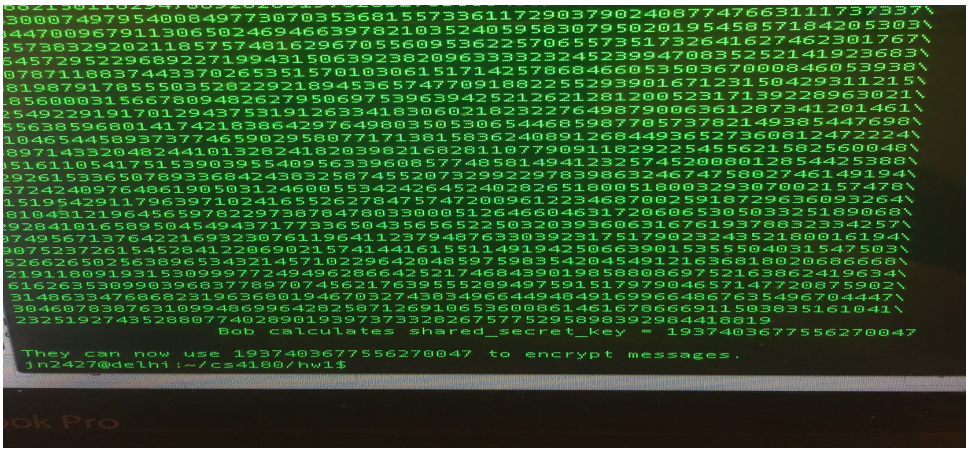
The protocol is not safe. It can be spoofed as in the example above.



Possible fix to the protocol.

Q3:





The calculated shared key using the Diffie-Hellman Key Exchange is:

1937403677556270047

Q6:

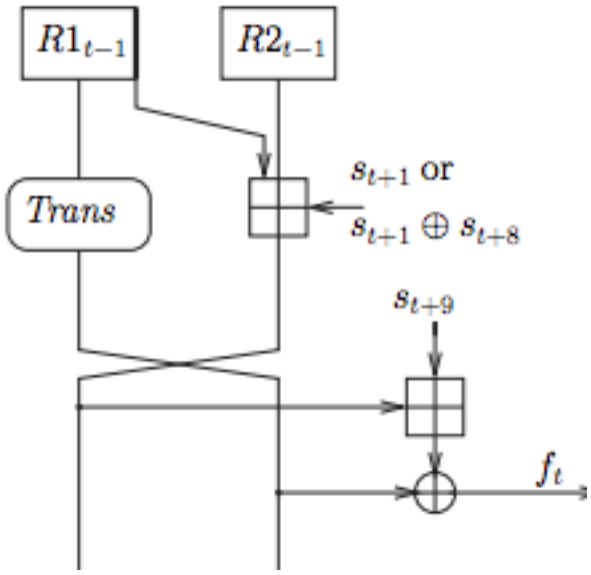
The Sosemanuk crypto-system is a stream cipher that utilizes a block cipher system 'Serpent' under its hood.

Its non-block cipher parts are separated into two portions. One is a Linear Feedback Shift Register (LFSR) and another is a FSM.

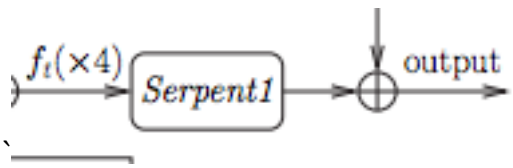
The finite state machine is initialized by the inputs generated by the LFSR. The FSM processes the inputs based on its state, and then send its output to the 'Serpent' block cipher system to be encrypted. The output of the 'Serpent' block cipher is then further XOR with the LFSR outputs to provide further confusion.

The FSM has finite state, thus it is deterministic. But its inputs $R1_t$, $R2_t$, f_t are manipulated by the inputs of the LFSR. The outputs of LFSR together with the state of FSM provide the FSM a degree of probabilistic characters.

$$FSM_t : (R1t-1, R2t-1, st+1, st+8, st+9) \; 7\rightarrow \; (R1t\,, R2t\,, ft)$$



The LFSR will execute 4 rounds; the FSM will save the output of each round. Once the rounds are completed, the FSM will send the save data to the 'serpent' block cipher. Here the output of FSM will be whiten by the specify 'Key'.



We can think of the FSM and LFSR layers of the Sosemanuk crypto-system as the diffusion layer of the standard AES system. And the 'Serpent' block cipher portion of the system as the confusion and whitening portion in the AES system.