CS411/507 Homework #4
Hakan Buğra Erentuğ 23637

1.) The problem with RSA_OESB is the key and R value size is too low that makes brute force valid option. Therefore, two for loop to find R and message is enough. Message have to be between 1000 and 9999 and the possible options for R is 128 number. Therefore we can encrypt each R and PIN pair and trying to find correspondence to the ciphertext. Please check RSA_OAEB.py for details. The PIN is 5377 and R value is 157.

2.) The design flaw is upper bound of k value is too low that enables exhaustive search. All we have to do is search k value one by one since g p and r values are given. Please check Elgamal.py for details. K value founded 17106. To find message we need to find "temp = mod_inv(h^k,p)". Since h and p values are already given, we were able to calculate temp variable. Also I want to remind that message is (temp*t %p). The message is "**My favorite machine at the gym is the vending machine".**

3.) Ticket Granting Server have to provide the identity of the server for security in responding process. (EKCG (S, KCS),EKSG (C, TS3, LT, KCS)) stands for respond and EKSG (C, TS3, LT, KCS) is the ticket between client and server. This helps authorize the client into to server. EKCG (S, KCS) is used for the server's identity otherwise any service is not possible. Of course, this ticket can be used for all services for its lifetime. In that case client would be able to access any service with the ticket that he granted from the ticket granting service. If the identity of the server was not included, then Cliff can use the ticket for any server or multiple servers for which he did not have authorization to access.

4.) To achieve forward secrecy they have to obey station and station protocol. Forward secrecy means using long-term public key to secure channel whereas, using sessions key for the communication. In station by station protocol Alice and Bob have their own secret keys lets say x and y. However, they both know the public keys a^y and a^x public keys. Alice and Bob can take power these public keys with their own private keys and obtain same result T. In that way they can both hash the T value with concativate of Params and reach the same result. In that way they can sign this result and decrypt to same result and verify the authentication. The citation is ch11_handout.pdf page 7.

5.) To calculate alfa we have to use the formula alfa = $(sihj - sjhix)(sjrix - sirj)^{-1} \pmod q$. There is no active usage of k1 and k2 however the coeffcients between them which is 2 have to be used to generate alfa. Because the x value is 2 and alfa value is 482061878283805054797834433118192109249147480995. Also, I added two verification to verify the messages for message1 and message2. Please check DSA.py for details.