effexp.py × EncryptedIM.py × *homework1part3.txt × DHEncryptedIM.py × readme.txt ×

1.
a) The 256 bit key is completely obtainable through brute force due to k16's size. Once k16 is guessed k256 might as well be of size 0 because it will be in plain text for someone else to read.

b) Dean B seems to have no way to verify that he is indeed Dean B. It seems like in this system it is implicity trusted that during key exchanges Dean B has been identified already so we will ignore this glaring problem.

Dean B using both keys can basically act like a diffie hellman go between and create a personally generated prime and base number. Using these Dean B can calculate a value. Dean B then sends the 1.calculated value
2.shared prime
to each of the professors who will compute the message using their keys they shared with Dean B. However, on this last step the attacker could have already got the keys from the first step thus allowing him to calculate the secret message with the
1. calculated value (obtained when Dean B sent it)
2. shared prime (obtained when Dean B sent it)
3. 1 professors key (obtained during intial send via the professors)

Seems like you would have to proform diffie hellman between Dean B and both professors in order to send an authkey between professors to encode the key. The encoded key can be decrypted by Dean B using professor 1's DH authkey and then reencrypted by Dean B using the other DH authkey which professor 2 will be able to decode

2.
To get symmetric key k_a_b we need to perform diffie hellman between Alice and Bob.
In order to make it impossible for Charlie to learn this key the values of the secrets (a) and (b) must not be known by Charlie.
However, Charlie must authenticate Alice and Bob to prevent a active listener who could intercept and send.
However, big C (the calculated value generated by Diffie Hellman during the creation of k_a_c and k_b_c) is known by everyone)
Big A and Big B (the calculated values generated by diffie hellman) should also be known by Charlie.

So Alice and Bob can keep thier secrets they used to calculate Big A and Big B since they are not known to any other party besides themselves.
The shared prime and base number can also be used again because as long as the secrets that Alice and Bob have are not known to anyother party it will be impossible for anyone else to calculate the symmetric key.
Alice and Bob can send eachother big A and big B and generate a symmetric key without an observer or Charlie having any way to calculate the symmetric key based upon only A, B, g, p

The only issue left is Authentication, Charlie can authenticate Alice and Bob if instead of Alice and Bob sending big A and big B directly, they send that through Charlie who will decrypt big A with the symmetric key a,c and then Charlie encrypts Big B with symmetric key b,c sending it to Bob
Charlie will never know the secrets of Alice and Bob
And Eve will never have the symmetric keys k-a-c and k-b-c thus will not be able to decrypt the messages sent between (Charlie and Alice) and (Bob and Charlie) allowing diffie hellman to not be man in the middled

4.
Eve can perform an active attack against this protocol. By intercepting and replacing the calculated and shared key from Alice, Eve can create a symmetric key between Bob while Bob will think he has a symmetric key with Alice.
Then replacing Bob's public key with her own, Eve can create a symmetric key with Alice.
Thus all messages between Alice and Bob can be encrypted and decrypted by Eve and sent to the appropriate party using the appropriate keys.

Bob and Alice are not authenticated so this is a major issue.

Plain Text ▾ Tab Width: 8 ▾ Ln 9, Col 41 ▾ INS

effexp.py × EncryptedIM.py × *homework1part3.txt × DHEncryptedIM.py × readme.txt ×

```
To run the program
python DHEncryptedIM.py -c localhost
in one computer
localhost can be replaced with an IP to leave the computer
python EncryptedIM.py -s
Then type in something an press enter on either screen

This uses a random number on both the server and client to create a authkey between both hosts that cannot be obtained through listening.
```