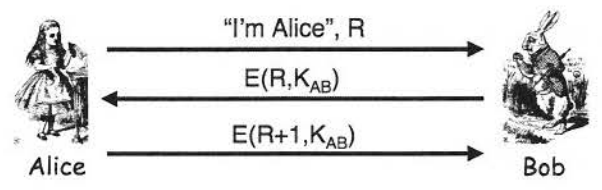
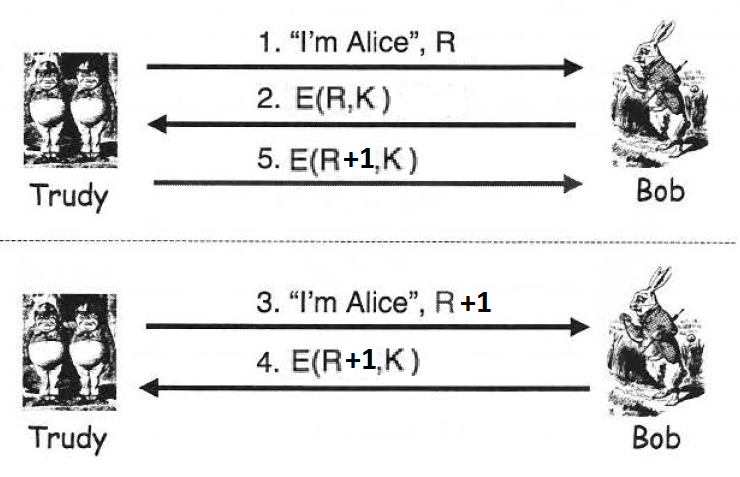
Assignment 9

1. **Consider the following mutual authentication protocol, where KAB is a shared symmetric key.**

****

**Give two different attacks that Trudy can use to convince Bob that she is Alice.**

Attack 1:

****

Attack 2:

Trudy can just record the three messages between Alice and Bob, and then just perform Replay Attack whenever Trudy wants convince Bob that she is Alice.

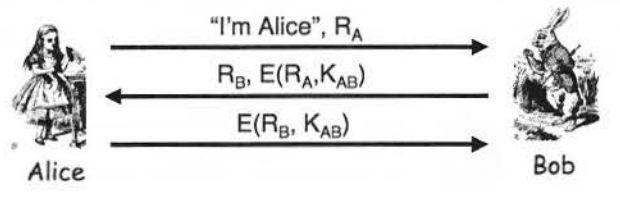
1. **Timestamps can be used in place of nonces in security protocols.**
   1. What is the primary advantage of using timestamps?

The primary advantage of using timestamp is that we can reduce the number of messages because we do not have to send a nonce for them to encrypt, instead we use timestamp from a synchronized clock.

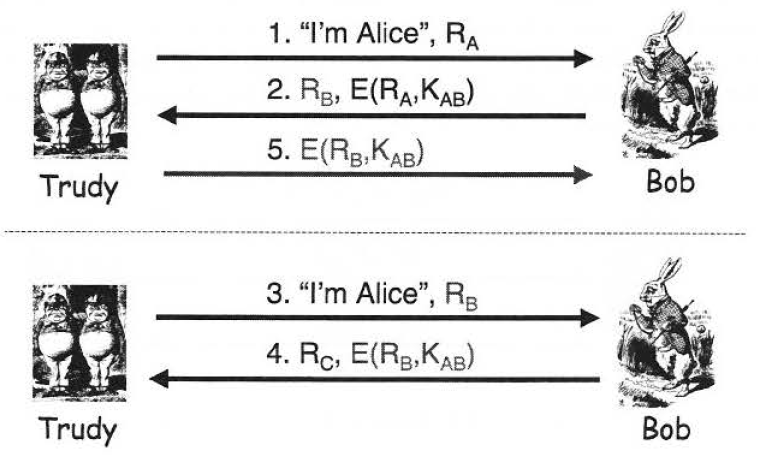
* 1. What is the primary disadvantage of using timestamps?

The primary disadvantage of timestamp is that it makes time a security-critical parameter. By this we mean that time has to always be synchronized and this is not something that we always have, thus when we use timestamp with unsynchronized clocks, many errors will come out.

1. **The following mutual authentication protocol is based on a shared symmetric key KAB.**

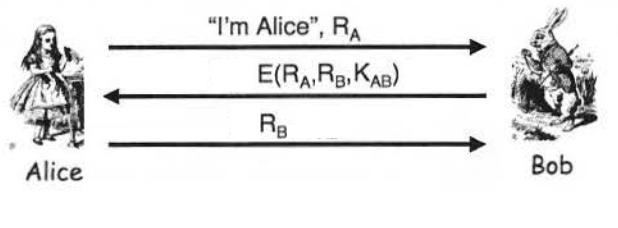
****

**Show that Trudy can attack the protocol to convince Bob that she is Alice, where, as usual, we assume that the cryptography is secure. Modify the protocol to prevent such an attack by Trudy.**

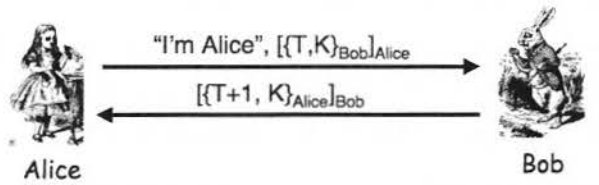
****

Trudy can just use two connections to attack this protocol; first, Trudy gives RA to Bob, Bob computes E(RA, KAB) and sends it with the nonce RB on the second message, now since Trudy cant encrypt RB, she can use RB on a second connection and send it to bob again, and Bob will send back E(RB, KAB), which Trudy can use to send it back as the third message on the first connection.

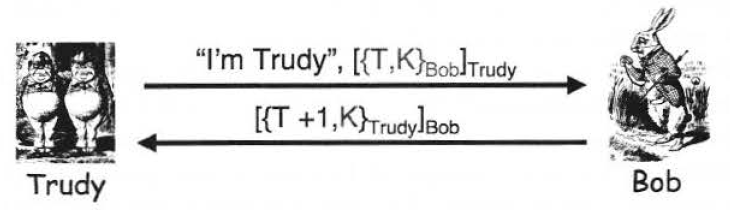
To modify this protocol, we can simply have Bob encrypt both RA and RB on together and ask Alice to send back the correct RB after decrypting it. Illustration of the protocol can be seen below:



1. **Consider the following mutual authentication and key establishment protocol, which employs a timestamp T and public key cryptography.**

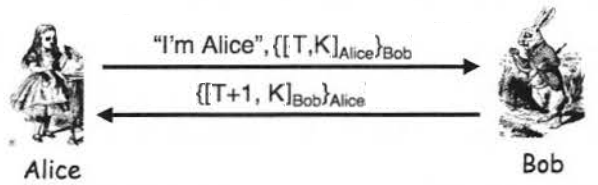
****

**Show that Trudy can attack the protocol to discover the key K where, as usual, we assume that the cryptography is secure. Modify the protocol to prevent such an attack by Trudy.**

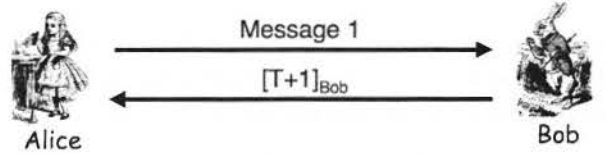


Trudy can simply record the 2 message between Alice and Bob, and then since the first message can be decrypted with Alice’s public key, Trudy can decrypt it and then use her private key to encrypt it and then send it as “I’m Trudy”, [{T,K}Bob]Trudy and then Bob will send back [{T+1,K}Trudy]Bob and then Trudy can use Bob’s public key and then her private key to decrypt and thus obtaining the session key K.

To make this protocol secure, we can simply swap the order of encryption, by first Encrypting with Alice’s private key and then Bob’s public key for the first message, and then on the second message we encrypt with Bob’s private key first and then encrypt with Alice’s private key. Illustration of the modified version shown below:



1. **Consider the following mutual authentication and key establishment protocol, which uses a timestamp T and public key cryptography.**

****

**For each of the following cases, explain whether or not the resulting protocol provides an effective means for secure mutual authentication and a secure session key K. Ignore replay attacks based solely on the clock skew.**

* 1. Message 1: {[T, K]Alice}Bob

This protocol does not provide secure mutual authentication, because in message 1, Alice did not mention her name, thus Bob would not know which public key to use to decrypt the message after decrypting with his private key.

* 1. Message 1:{“Alice”, [T, K]Alice}Bob

This protocol can securely establish a session key and also be able provide secure mutual authentication, because the on message 1 we used Alice’s private key to encrypt T and K and then used Bob’s public key to encrypt “Alice” and the already encrypted message, thus Trudy cannot know the content of the message and bob just sends T+1 encrypted with his private key on the second message to prove his identity to Alice.

* 1. Message 1:“Alice”, {[T, K]Alice}Bob

This protocol is no different from the protocol on part B, it will still be able to securely establish a session key and provide secure mutual authentication.

* 1. Message 1: T, “Alice”, {[K]Alice}Bob

In this protocol Trudy can perform a replay attack because timestamp T is not encrypted and thus she can just change that part of the message and she will be able to perform the replay attack and authenticate as Alice.

* 1. Message 1:”Alice”, {[T]Alice}Bob and let K = h(T)

This protocol can provide a secure mutual authentication but not establish a secure session key, because since K = h(T), Trudy can use Bob’s public key to decrypt message 2, obtaining T+1 which she can get T and then perform h(T) to get the session key.

1. **Consider the following three-message mutual authentication and key establishment protocol, which is based on a shared symmetric key KAB.  
   For each of the following cases, briefly explain whether or not the resulting protocol provides an effective means for secure mutual authentication and a secure session key K.**
   1. Message 1: E("Alice" , K, RA, KAB), Message 2: RA, E(RB, KAB)

This protocol provides a secure mutual authentication and a secure session key, Bob is authenticated by being able to decrypt message one to obtain the session key and also send back RA to prove his identity, while Alice is authenticated by being able to decrypt message 2, and send back RB to prove her identity. Therefore, both Alice and Bob are authenticated and the session key is securely given to both of them.

**IF WE ARE ALLOWED TO PERFORM AN ATTACK ON THE CIPHERTEXT**

This protocol does not provide a secure mutual authentication and a secure session key. In this protocol, Trudy can record all three messages, and since message 2 is E(RB, KAB) and message 3 is RB, Trudy can use these two message to deduce KAB and thus in the future if Trudy wants to authenticate as Alice, she can simply replay message 1, use KAB to decrypt message 2 and send the RB

* 1. Message 1: "Alice", E(K, RA, KAB), Message 2: RA, E(RB, K)

This protocol provides a secure mutual authentication and a secure session key, Bob is authenticated by being able to decrypt message one to obtain the session key and also send back RA to prove his identity, while Alice is authenticated by being able to decrypt message 2, and send back RB to prove her identity. Therefore, both Alice and Bob are authenticated and the session key is securely given to both of them

**IF WE ARE ALLOWED TO PERFORM AN ATTACK ON THE CIPHERTEXT**

This protocol does not provide a secure mutual authentication and a secure session key. In this protocol, Trudy can record all three messages, and since message 2 is E(RB, K) and message 3 is RB, Trudy can use these two message to deduce session key K and thus in the future if Trudy wants to authenticate as Alice, she can simply replay message 1, use session key K to decrypt message 2 and send the RB

* 1. Message 1: "Alice", E(K, RA, KAB), Message 2: RA, E(RB, KAB)

This protocol provides a secure mutual authentication and a secure session key, Bob is authenticated by being able to decrypt message one to obtain the session key and also send back RA to prove his identity, while Alice is authenticated by being able to decrypt message 2, and send back RB to prove her identity. Therefore, both Alice and Bob are authenticated and the session key is securely given to both of them

**IF WE ARE ALLOWED TO PERFORM AN ATTACK ON THE CIPHERTEXT**

This protocol does not provide a secure mutual authentication and a secure session key. In this protocol, Trudy can record all three messages, and since message 2 is E(RB, KAB) and message 3 is RB, Trudy can use these two message to deduce KAB and thus in the future if Trudy wants to authenticate as Alice, she can simply replay message 1, use KAB to decrypt message 2 and send the RB

* 1. Message 1: "Alice", RA, Message 2: E(K,RA,RB,KAB)

This protocol provides a secure mutual authentication and a secure session key, Bob is authenticated by being able to encrypt message 1 into message 2 by using the symmetric key KAB thus proving his identity because he has the symmetric key, while Alice is authenticated by being able to decrypt message 2 by obtaining RB and the session key, Alice sends back RB on message 3 to prove her identity. Therefore, both Alice and Bob are authenticated and the session key is securely given to both of them

1. **Consider the following three-message mutual authentication and key establishment protocol, which is based on a public key cryptography.  
   For each of the following cases, briefly explain whether or not the resulting protocol provides an effective means for secure mutual authentication and a secure session key K.**
   1. Message 1: {"Alice",K,RA}Bob, Message 2: RA, RB

This protocol provides a secure session key but does not provide a secure mutual authentication. Message 1 is encrypted with Bob’s public key, which can be used by anyone, therefore we can say that Trudy will be able to forge Message 1. Message 2 proves that Bob is able to decrypt message 1, thus Bob’s identity is proven, but Alice Identity is still not proven because it requires no work to send back RB on message 3 because it is not encrypted in message 2.

* 1. Message 1: "Alice", { K , RA }Bob, Message 2: RA, {RB}Alice

This protocol provides a secure session key and also a secure mutual authentication. In message 1, Alice specifies that she is Alice and sends the session key and RA encrypted with Bob’s public key. Bob decrypts message 1 with his private key, thus proving his identity, and sends RB encrypted with Alice’s public key on message 2, in order to send back RB­ on message 3, Alice will have to decrypt message 2 with her private key and thus prove her identity. Therefore, both Alice and Bob are authenticated and a session key is securely established.

* 1. Message 1: "Alice", {K}Bob, [RA]Alice, Message 2: RA, [RB]Bob

This protocol does not provide a secure session key and neither a secure mutual authentication because this method is subject to Replay attack. On message 1, Alice encrypts the session key with Bob’s public key, and then encrypt RA with her private key, but the cihpertext {K}Bob is vulnerable to a replacement attack. Bob’s then decrypt message 1 using Alice’s public key and then sends RB encrypted with his private key on message 2. On message 3 Alice just uses Bob’s public key to find RB and send it back.

Although RA and RB can are encrypted with Alice’s and Bob’s private key respectively, it might seem that it proves their identity, but it is not, because in this protocol, Trudy can replay the ciphertexts and she is able to fool either one of them. Trudy can just replay message 1, when she receives message 2, she just have to use Bob’s public key and send it back on message 3, and therefore she is authenticated as Alice.

* 1. Message 1: RA, {"Alice",K}Bob, Message 2: [RA]Bob, {RB}Alice

This protocol provides a secure session key and also a secure mutual authentication. Alice sends RA and {“Alice”, K}Bob on message 1 to Bob, Bob proves his identity by being able to decrypt the message and see who is the sender, Bob encrypts RA with his private key, and uses Alice’s public key to encrypt RB and then send both on message 2. Alice then knows that this is the real Bob because he was able to use his private key to encrypt RA, Alice then has to use her private key to find out RB and then send it on message 3, After Bob receives message 3, he knows Alice identity because only Alice has the private key. Therefore they are mutually authenticated and the session key was securely established.

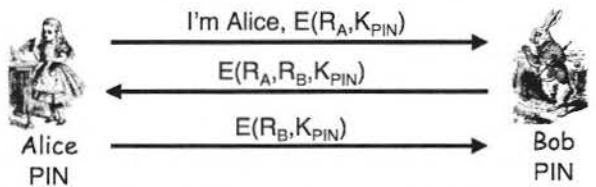
* 1. Message 1: {"Alice",K,RA,RB}Bob, Message 2: RA, {RB}Alice

This protocol provides a secure session key but does not provide a secure mutual authentication. On message 1, Trudy can just forge the message by using Bob’s public key to encrypt RA, RB, “Alice” and K. On message 2, Bob is able to prove his identity by decrypting message 1, and then sends back RA and encrypt RB with Alice’s public key. On message 2, Bob is authenticated and logically we assume that on message 3, Alice is authenticated because she was able to decrypt message 2, but that is wrong because RB is already known because it was determined by the client side. Thus Trudy can easily fool Bob that he is talking to Alice.

1. **A timestamp-based protocol may be subject to a replay attack, provided that Trudy can act within the clock skew. Reducing the acceptable clock skew might make the attack more difficult, but it will not prevent the attack unless the skew is zero, which is impractical. Assuming a nonzero clock skew, what can Bob, the server, do to prevent attacks based on the clock skew?**

By encrypting the timestamp and the session key on the first message as “I’m Alice” {[T,K]Alice}Bob, and then will need to decrypt message 1 with his private key, after getting T and K, bob uses his private key and Alice’s public to encrypt it into {[T,K]Bob}Alice, this protocol provides secure means to mutual authentication as well as a secure session key.

1. **Consider the authentication protocol below, which is based on knowledge of a shared 4-digit PIN number. Here, KPIN = h(PIN).**



**Suppose that Trudy passively observes one iteration of the protocol. Can she then determine the 4-digit PIN? Justify your answer.**

In this protocol, Trudy will not be able to determine the 4 digit PIN, because the 4 digit PIN does not appears in the message, only the hash of the PIN is present, and even if Trudy is able to get the hash of the PIN, she cannot determine the PIN by having the hash because hash is one way and she cannot figure out the PIN by having the hash.

1. **Suppose that in the Fiat-Shamir protocol in Figure 9.32 we have N = 63 and v = 43. Recall that Bob accepts an iteration of the protocol if he verifies that y2 = x \* ve mod N.**
   1. In the first iteration of the protocol, Alice sends x = 37 in message one, Bob sends e = 1 in message two, and Alice sends y = 4 in message three. Does Bob accept this iteration of the protocol? Why or why not?

Yes Bob accepts the iteration, because x = 37, v = 43 and N = 63, thus y2 = 37 \* 43 mod 63 = 16 which is Alice’s third message y = 4.

* 1. In the second iteration of the protocol, Alice sends x = 37, Bob sends e = 0, and Alice sends y = 10. Does Bob accept this iteration of the protocol? Why or why not?

In this iteration, Bob does not accept it because Alice’s third message is y = 10, and since e = 0, we have y2 = 37 \* 1 Mod 63= 37, and the square root of 37 does not equals to 10, therefore the second iteration fails.

* 1. Find Alice's secret value S. Hint: 10-1 = 19 mod 63.

19 \* 4 = 76

762 mod 63 = 43

Therefore, **S = 76.**