Assignment 6
Gunnar Yonker

1.

(1)

Digital certificates are a way to show a proof of identity that is used to verify the authenticity of a public key. It is a more secure and efficient way of verifying a user's identity during the key exchange process. This is because digital certificates are a standardized and automated way of verifying the authenticity of a public key by using a trusted third party called Certificate Authority(CA). The owner of the public key is verified by CA and then CA acts as a trusted intermediary that then vouches for the authenticity of the public key and ensures that only authorized parties have access to that key. It is a scalable system because it is automated which means that it is a quick and efficient process. Using a digital certificate also eliminates some of the risk of errors when distributing the keys manually such as user error when entering or sending the keys. When using digital signatures and secure hash functions, it helps to ensure that the security of the key cannot be reduced through tampering.

(2)

A receiver can verify that the public key included in the digital certificate is issued by the CA and has not been modified by following a few steps. First the receiver would receive a copy of the digital certificate which would contain the key that they want to verify. The user then would obtain the CA's public key from the CA's website or a directory, as long as the source is a trusted source. The user would then use the CA's public key to verify the digital signature on the digital certificate, this would then ensure that the digital signature was created by the CA using it's private key. This would then verify the integrity of the digital certificate and the authenticity of it that it was not tampered with and was created by the CA. The receiver now can use the public key that is contained in the digital certificate to have secure communication with the sender knowing that the sender is authentic and all communications are authentic and not tampered with.

2.

(1)

Defend Against Replay Attacks

1. Host sends packet requesting connection.

2. Host A sends a message to the KDC requesting a session key and includes a unique random number in the message, R1.

The KDC generates a session key, Ks, and creates a ticket containing the session key and the unique random number. The ticket is encrypted using the KDC's private key and sent to the Host A.

3. The KDC sends the session key to Host B encrypted with Host B's public key, along with a new unique random number, R2.

Host B sends a message back to the KDC that includes the random number, R2, and a message authentication code created using the session key and the unique numbers, R1 and R2. The KDC verifies the MAC address and if it is valid, sends a confirmation message to Host A that the session key has been distributed.

Assignment 6
Gunnar Yonker

4. Buffered packet transmitted.

(2)

The protocol messages are used to achieve the key distribution and defend against replay attacks in step 2 because of the use of the unique random number which ensures that the message cannot be replayed by the attacker. The session key and random number are encrypted using the KDC's private key and sent to the host. Host A is then able to decrypt that message and access the session key. In step 3, the KDC sends the session key to Host B encrypted with a second unique random number, R2. A secure channel should be used when sending this message. Then host B sends back a message including the random numbers and a MAC created using the session key. The use of the unique random number here also defends against a replay attack from a third party. The KDC then verifies the MAC that was created using the session key and if it is valid, sends a confirmation message to Host A that the session key has been distributed. The use of the MAC is to ensure that the message has not been tampered with during transmission and can be considered authentic and valid. By following these steps the hosts can be sure that they are securely distributing the session key and that they are communicating with each other and no third party acting in the middle. By using the unique random numbers, it helps to prevent an attacker using the replay attack.

3.

(1)

Step 1: A sends E(PUb,[N1 || IDA]) to B, N1 is a nonce generated by A and IDA is the identity of A. E(PUb) means that A encrypted a message using B's public key. This is so that A can then authenticate themselves to B by proving that it has the private key corresponding to the public key that B possesses. The nonce is used to ensure that the message cannot fall victim to a replay attack because the nonce will be different every time.

Step 2: B sends E(PUa,[N1 || N2]) to A, N2 is a nonce generated by B. PUa is A's public key, and E(PUa) means that B encrypted the message using A's public key. This is so that B can authenticate themselves to A by proving that it has the private key that corresponds to the public key that A possesses. The use of N1 ensures that the message is sent in response to the message sent in step 1 and the use of N2 ensures that the message cannot be replayed in a replay attack.

Step 3: A sends E(PUb, N2) to B. This message is sent to confirm that A has received B's message in step 2 and to prove that A has the private key that corresponds to the public key that B possesses. N2 is used so that the message cannot be replayed using a replay attack.

Step 4: A sends E(PUb, E(PRa, Ks)) to B, in this message Ks is a session key that is generated, PRa is A's private key. Now that A and B have both verified their identities to each other, this message is sent to establish a shared session key. B is able to decrypt the message and obtain the shared session key.

(2)

Typically, a key distribution process should be initiated whenever there is a new session started. In the case of my project this would be when the client connects to the server to authenticate and then log in. If the client is disconnected either through 5 or more login attempts, connection error, or manually disconnecting after their reservation the key distribution process should be initiated again with the new

connection. In any of these situations the original connection would be ended, and with the new connection the secure key exchange process should take place again to verify the identify of the two parties and to establish a session key. If there was a situation where a higher level of security was needed, there could be a more frequent key distribution process to make the communication more secure such as in an environment where financial data is being handled and every transaction requires the key initiation process. In our case, our project is less sensitive so using the key initiation process once per connection should be secure enough in this situation based off of the above situations when a re-connection is needed there should be a new key distribution process.