German University in Cairo
Faculty of Media Engineering and Technology
Mervat Abu-Elkheir
Ahmad Helmy
Mohamed Abdelrazik

CSEN1001: Computer and Network Security

Spring Term 2019

Tutorial 8

Problem 1 - Needham-Schroeder

Consider the following protocol for secret-key exchange

1. $A \rightarrow S : A, B, N_A$
2. $S \rightarrow A : \{N_A, B, K_{A,B}, \{K_{A,B}, A\}K_{B,S} \}K_{A,S}$
3. $A \rightarrow B : \{K_{A,B}, A\}K_{B,S}$
4. $B \rightarrow A : \{N_B\}K_{A,B}$
5. $A \rightarrow B : \{N_B\text{-}1\}K_{A,B}$

a) What is the benefit of using N_A?

In step 1 we use a nonce N_A to prevent a replay attack i.e. A checks that N_A appears in the token returned by S in step 2.

b) Which phase of the protocol implements mutual authentication?

In steps 4 and 5, A and B perform mutual authentication i.e. they prove to each other that they have the session key and that they are participating in this run of the protocol.

Problem 2 - Public-Key Authority

"Eve generates a Public, Private key pair, and sends the Public key PU^e to Bob, claiming that she is Alice." How can this problem be avoided?

Answer

This can be avoided by using a Public-Key authority, the following would happen:

(a) Alice requests Bob's public key PU^b from the authority, sending along the timestamp of the request.

(b) The authority replies back with Bob’s public key PU^e , and the timestamp it received from Alice, both encrypted with the authority’s private key PR^{auth}.

(c) Alice uses Bob’s public key PU^b to encrypt a message to Bob that carries a nonce N_1.

(d) Bob gets Alice’s public key PU^a, the same way as in step (a) and (b).

(e) Bob replies to Alice with N_1 and N_2, N_1 assures Alice that Bob is the one who received the message, not Eve, because Eve wouldn’t have been able to find N_1.

(f) Alice replies to Bob with the N_2 it received, which assures Bob that it is Alice, because Eve cannot find N_2.

(g) Now Alice and Bob have a secure channel between them

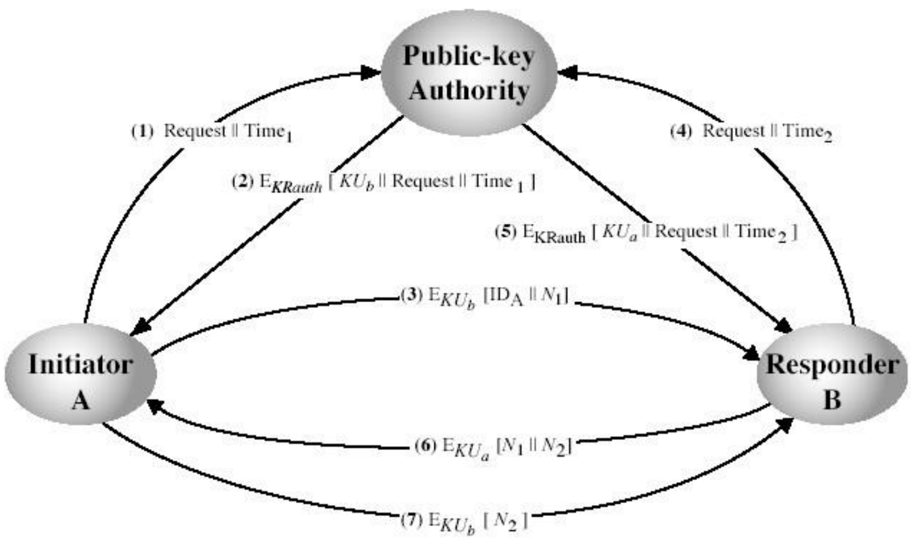

Figure 1: Public-key authority

Problem 3 - Session keys

Communicating with an authority provides more security, at the expense of more wasted time, for this reason, simple session keys were proposed as an alternative. What kind of problems can arise when using Session keys? How can they be avoided?

Answer

Session keys are symmetric keys, and thus much faster to compute an encryption using it than using a public key.

The idea of session keys, is that Alice would send Bob her public key PU^a, and Bob would reply with a session key K_s encrypted with PU^a. See Figure 3.

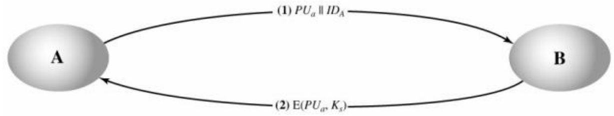

Figure 2: Session-key exchange

The problem here, is that Eve can make a "Man in the Middle" attack.

(a) Eve can intercept the message sent by Alice to Bob, which is $PU^a||ID_A$.

(b) Eve then sends ID_A along with her own Public key PU^e.

(c) Bob, thinking that he got the message from Alice, will reply with the session key K_s, encrypted with PU^e.

(d) Eve can decrypt this message to get K_s, and can now forward it to Alice after encrypting it with Alice's public key PU^a.

(e) Alice will think that everything went as expected. And start sending the messages encrypted with K_s.

(f) Since Eve knows K_s, she can read all the exchanged messages.

A possible solution, is to first establish a public key, by using two related values N_1 and N_2, the steps proceed as follows (See Figure 4):

(a) Alice sends Bob N_1 and ID_A encrypted with his public key PU^b.

(b) Bob replies with N_1, that he got from Alice, and N_2, that is computed from N_1, encrypted by Alice's public key PU^a.

(c) Alice then sends N_2 back to Bob, encrypted with his public key PU^b, to assure him it's her.

(d) Then Alice sends a session key K_s, encrypted with her private key, and Bob's public key.

(e) Now that Alice and Bob are sure that the K_s is secured and known only to the two of them, they can both use K_s for encryption and decryption (symmetric key).

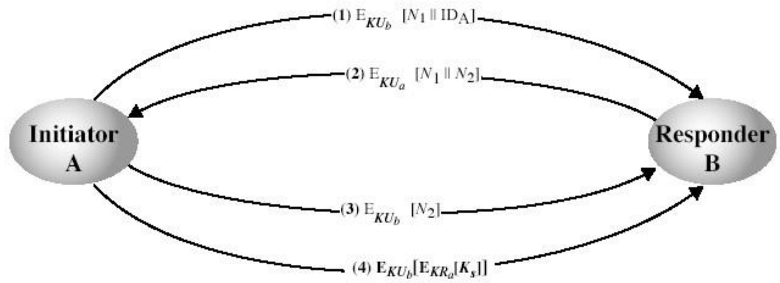

Figure 3: Session-key exchange

Problem 4 - Certificate Authority

Public-Key authority provides greater security, since Alice and Bob deal with a Trusted-Third-Party, rather than directly communicating with each other. However, such communication places a big burden on the authority, because anyone who wants to communicate with Alice, will have to get her Public key PU^a from the authority first. How can this be improved?

Answer

By using a Certificate Authority (CA). In this case, the following would happen:

(a) Alice sends her Public Key PU^a to the CA.

(b) The CA takes care of ensuring that this is the actual Alice and replies back to Alice with $C_A = PR^{CA}\{PU^a\}$ which is her Public Key and a timestamp, both encrypted by the CA's Private key.

(c) Whenever Alice would like to communicate with Bob, she would send the C_A to Bob, Bob can use the CA's Public key PU^{CA} to get decrypt C_A and get Alice's Public key PU^a. The timestamp is used to assure Bob that the certificate isn't old.

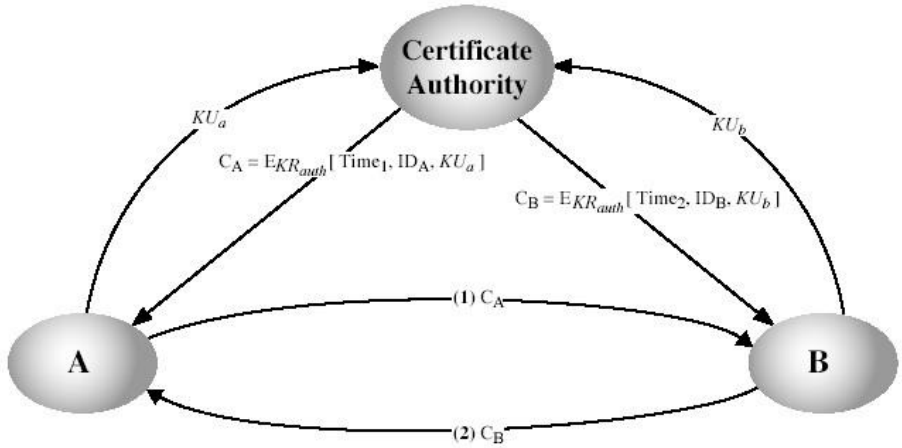

Figure 4: Public-key authority

References

- BINF711 & CSEN1001 Spring 2014
- Questions from the book