Auto-Evaluation Questions

Authentication and Key Distribution Protocols using Symmetric Encryption and KDCs

The Kerberos Model and Protocol

1. Revise the following authentication and key distribution protocol models explained in the lectures (see the slides or related literature). You must understand the protocol models in each case, trying to identify the advantages of each protocol, compared with the other ones.

   a) Needham-Schroder (Symmetric Encryption) model
   b) Otway-Rees
   c) Yahalom
   Wide-Mouth Frog
   d) Neuman-Stubblebine

2. Explain the concept of Master and Session Keys and how they are used in a protocol for Authentication and Key Distribution, when we only use Symmetric Cryptography.

3. Identify the advantages of the arbitrated process using KDC (Key Distribution Centers) compared to the use of Master Keys and Session Keys in P2P session key establishment processes.

4. Why we need rekeying mechanisms and explain some criteria to fire a rekeying procedure.

5. Explain how we can use Key Tags as Key Selectors for Key Management purposes. Explain of and how we can support this using persistent keystores of type JECKS in the Java JCA/JCE Programming environment.

6. Explain the use of control vectors for Key management purposes and how we can encrypt and decrypt using control vectored keys.

7. Explain what is a Password-Based Encryption (PBEncryption) scheme and its purpose.

8. In a PBEncryption construction we can use Symmetric Block Encryption Algorithms and Secure Hash Functions. What is the purpose of the Secure Hash Function ?

9. List different solutions to store and to manage Keys and explain for each solution their advantages and drawbacks, compared with the use of Java keystores.

10. Define Perfect Forward Secrecy (PFS) and Perfect Backward Secrecy (PBS) and what are the requirements for PFS and PBS in a key distribution and rekeying protocol.

11. When can we say that a generated key is contributive?

12. Can we say that the Key Distribution protocols in the question 1 offer PFS and PBS ? Discuss.

13. Can we say that the Key Distribution protocols in the question 1 offer ways for contributive keys? Discuss.

14. Read the following except of Sherlock Holmes dialogue with Detective Lestrade:

    "We are under great pressure, Holmes." Detective Lestrade looked nervous. "We have learned that copies of sensitive government documents are stored in computers

of one foreign embassy here in London. Normally these documents exist in electronic form only on a selected few government computers that satisfy the most stringent security requirements. However, sometimes they must be sent through the network connecting all government computers. But all messages in this network are encrypted using a top secret encryption algorithm certified by our best crypto experts. Even the NSA and the KGB are unable to break it. And now these documents have appeared in hands of diplomats of a small, otherwise insignificant, country. And we have no idea how it could happen."

"But you do have some suspicion who did it, do you?" asked Holmes.

"Yes, we did some routine investigation. There is a man who has legal access to one of the government computers and has frequent contacts with diplomats from the embassy. But the computer he has access to is not one of the trusted ones where these documents are normally stored. He is the suspect, but we have no idea how he could obtain copies of the documents. Even if he could obtain a copy of an encrypted document, he couldn't decrypt it."
"Hmm, please describe the communication protocol used on the network." Holmes opened his eyes, thus proving that he had followed Lestrade's talk with an attention that contrasted with his sleepy look.
"Well, the protocol is as follows. Each node N of the network has been assigned a unique secret key K_n. This key is used to secure communication between the node and a trusted server. That is, all the keys are stored also on the server. User A, wishing to send a secret message M to user B, initiates the following protocol:

1. A generates a random number *R* and sends to the server his name A, destination B, and E(K_a, *R*).
2. Server responds by sending E(K_b, *R*) to A.
3. A sends E(*R*, *M*) together with E(K_b, *R*) to B.
4. B knows K_b, thus decrypts E(K_b, *R*) to get *R* and will subsequently use *R* to decrypt E(*R*, *M*) to get *M*.

You see that a random key is generated every time a message has to be sent. I admit the man could intercept messages sent between the top secret trusted nodes, but I see no way he could decrypt them."

"Well, I think you have your man, Lestrade. The protocol isn't secure because the server doesn't authenticate users who send him a request. Apparently designers of the protocol have believed that sending E(K_x, R) implicitly authenticates user X as the sender, as only X (and the server) knows K_x. But you know that E(K_x, R) can be intercepted and later replayed. Once you understand where the hole is, you will be able to obtain enough evidence by monitoring the man's use of the computer he has access to. Most likely he works as follows: After intercepting E(K_a, R) and E(R, M) (see steps 1 and 3 of the protocol), the man, let's denote him as Z, will continue by pretending to be A and…

Finish the sentence for Holmes and find yourself his deduction.

15. There are three typical ways to use *nonces* as challenges in a challenge-response protocol. Suppose N_a is a nonce generated by A, A and B share key K, and f() is a function (such as increment). The three usages are

Usage 1

A : B:N_a(2)

B : A: E(K, N_a)

Usage 2

A : B: E(K, N_a)

B : A: N_a

Usage 3

A : B: E(K, N_a)

B : A: E(K, f(N_a))

Describe situations for which each usage is appropriate and to have guarantees anti-message replaying.

16. In Kerberos, when Bob receives a ticket from Alice, how does he know it is genuine?
17. In Kerberos, when Bob receives a ticket from Alice, how does he know it came from Alice?
18. In Kerberos, Alice receives a reply, how does she know it came from Bob (that it's not a replay of an earlier message from Bob)?
19. In Kerberos, what does the ticket contain that allows Alice and Bob to talk securely?
20. Try to explain how Kerberos V5 can support three authentication domains (realms) allowing that clients in any domain can use resources (servers) in any other domain (realm).
21. Explain how the Authentication Forward (for client-delegation) limitation in Kerberos V4 is not present in the Kerberos V5 and how it can be supported.
22. Discover in the Kerberos V4 protocol the Double Encryption limitation, and what is the tradeoff of such design deficiency.
23. What are the advantage of Ticket-Renewal in Kerberos V5 compared with Kerberos V4 and how it is used ?
24. Why the last round message (message 6) of the Kerberos V5 protocol is relevant ? Explain.
25. Present a solution, leveraging on Kerberos V5 for the client and server to generate contributive keys. Is it possible to support also Perfect Forward and Perfect Backward secrecy ? Discuss.