CHAPTER 31 (corrisponde al cap. 29 italiano)

Network Security

Solutions to Review Questions and Exercises

Review Questions

1. A ***nonce*** is a large random number that is used ***only once*** to help distinguish a fresh authentication request from a repeated one.
2. The N^2 problem refers to the large number of keys needed for symmetric key cryptography. For N people, $N \times (N-1)/2$ keys are needed, which is proportional to N^2.
3. Both the ***Needham-Schroeder*** and the ***Otway-Rees*** protocols use a ***KDC*** for user authentication.
4. The ***Kerberos authentication server (AS)*** registers each user and grants each user a user identity and a password. The AS issues a session key for use between the sender and the ticket-granting server (TGS).
5. The ***Kerberos TGS*** issues a ticket for the real server and provides the session key between the sender and the receiver.
6. ***X.509*** is a protocol that describes the certificate in a structural way.
7. A ***certification authority (CA)*** is a federal or state organization that binds a public key to an entity and issues a certificate.
8. A ***long password*** is more immune to guessing than a ***short password***. However, a long password is difficult to remember; it is often written somewhere. This may make it easier for the adversary to steal it.
9. A ***frequently-changed password*** is more secure than a ***fixed password*** but less secure than a ***one-time password.*** However, a one-time password needs more effort from the system and the user. The system needs to check if the password is fresh every time the user tries to use the password. The user needs to be careful not to use the pervious one. A more frequently changed password can be used as an alternative. One solution is that the system initializes the process of changing the password by sending the new password, through a secure channel, and challenging the user to be sure that the right user has received the new password.
10. One way to prevent a ***guessing attack*** on a password is to use long passwords. For example, it is more difficult to guess a 10-digit password than a 4-digit one. Banks

recommend that a customer not use a short PIN (a type of password). In particular, they recommend not using an easily-guessed number such as the birth year. Banks also request a change in the PIN when a stolen bank card is reported and replaced by a new one.

Exercises

11.

a. The algorithm meets the first criteria (***one-wayness***). It is not possible to find the original numbers if the digest is given. For example, if we know the digest is 76, we cannot find the original ten numbers. They can be any set of 10 numbers.

b. The algorithm does not meet the second criteria (***weak collision***). If the digest is given, we can create 10 numbers that hash to the same digest. For example, Eve, without knowing the original set of numbers, can intercept the digest of **51** and create the set {12, 23, 45, 12, 34, 56, 9, 12, 34, 14} and send it with the digest **51** to Bob. Bob is fooled and believes that the set is authentic.

c. The algorithm does not meet the third criteria (***strong collision***). If the digest is given, we can create at least two sets of 10 numbers that hash to the same digest. For example, Alice can create two sets {12, 23, 45, 12, 34, 56, 9, 12, 34, 14} and {12, 23, 45, 16, 34, 56, 9, 12, 34, 10} that both hash to **51**. Alice can send the first set and the digest to Bob, but later she can claimed that she sent the second set.

12.

a. The algorithm meets the first criteria (***one-wayness***). Most of the characters are lost in the process and cannot be reproduced from the digest.

b. The algorithm does not meet the second criteria (***weak collision***). If the digest is given, we can create a message as long as the characters 1, 11, 21,..., 91 are the same as the corresponding characters in the digest. Eve, without knowing the original set of characters, can intercept the digest and create a new set out of the digest and send it with the digest to Bob. Bob is fooled and believes that the set is authentic.

c. The algorithm does not meet the third criteria (***strong collision***). We can easily create two messages in which characters 1, 11, 21,..., 91 are the same but the other characters are different. The digests for both messages are the same. Alice can send the first message and the digest to Bob, but later she can claim that she sent the second set.

13. The possible number of digests is 2^N because each bit can be in one of the two values (0 or 1).

14. It is more probable to find two people with the same birthday than to find a person born on a particular day of the year. For example, in a party of 10 people, we can find the probabilities for the two cases:

a. The probability that a person is born on a particular day (such as February 20) is **0.027** (almost **3** percent)

b. The probability that two or more persons are born in the same day is **0.117** (almost **12** percent)

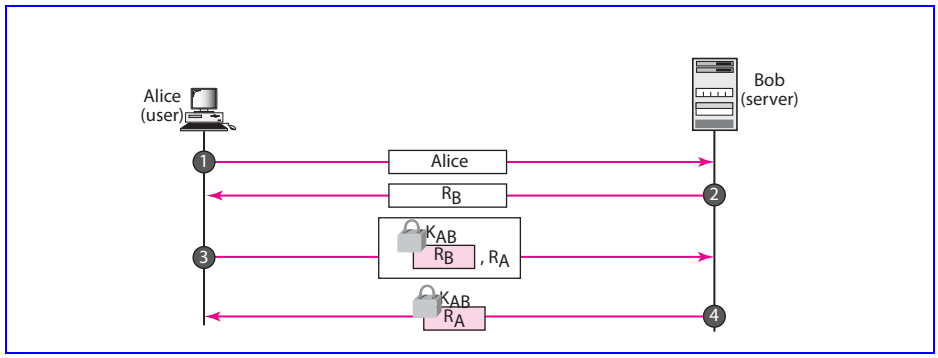
The difference increases sharply when the number of people in a party reaches 20 or more. In the classic birthday probability problem, if there are **23** people in a party, the probability is more than fifty percent that two people will have the same birthday.

15. The second and third criteria for a hashing function are closely related to the solution found in problem 14. In the problem we try to related the number of people at the party to the number of days in a year. In a hashing function, we can relate the number of possible messages to the number of possible digests. To understand the problem assume that there are only 10 possible messages (number of people at the party) but there are 365 possible digests.
    a. If a particular digest is given (a particular birthday), the probability that Eve can find one of the ten messages (one of the ten people in the party) is 0.027 (2.7 percent).This is related to the weak collision. The probability is very weak. That is why it is called ***weak collision***.
    b. The probability that Alice can create two or more messages with the same digests is the probability of finding two or more people with the same birthday in a party. If the number of possible messages is 10 and the number of possible digest is 365, this probability is 0.117 or (11 percent). That is why this criterion is called ***strong collision***. The probability is higher. It is more probable that Alice can find two or messages with the same digest than Eve can find a message with a given digest.

    The above discussion leads us to the point that we should worry more about the second criterion that the first. To decrease the probability of both criteria, we need to increase the number of possible digests and the number of possible messages. We need to increase the number of bits in a digest and impose a minimum number of bits on messages.
16. A ***fixed-size digest*** is more feasible. A ***variable-size digest*** needs to be dependent on the length of the message, which makes applying the criteria more difficult and the function itself more involved.
17. The whole idea of a sophisticated hash function such as ***SHA-1*** is that the partial digest of each block is dependent on the partial digest of the previous block and the message on the current block. Each block mingles and mixes the bits in a such a way that changing even one bit in the last block of the message may changed the whole final digest.
18. We can distinguish between the two:
    a. A signed hash normally means first making a hash and then encrypting it with a secret key.
    b. A MAC normally means first concatenating the secret key with the message and then applying the hash function.
19. It is normally both. The entity authentication (based on the PIN) is needed to protect the person and the bank in case the money card is stolen. The message authentication is normally needed for the entity authentication.

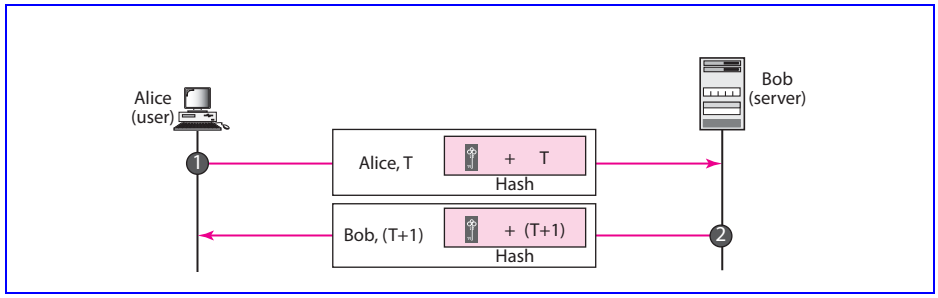
20. Figure 31.1 show one scheme using four messages. In this scheme, Alice, the initiator, needs to authenticate herself before Bob does the same. After the third message, Alice is authenticated for Bob; after the fourth message, Bob is authenticated for Alice. Although, the number of messages can be reduced to three, but (as you can see in textbooks devoted to security) the three-message scheme suffers from some flaws.

Figure 31.1 *Solution to Exercise 20*



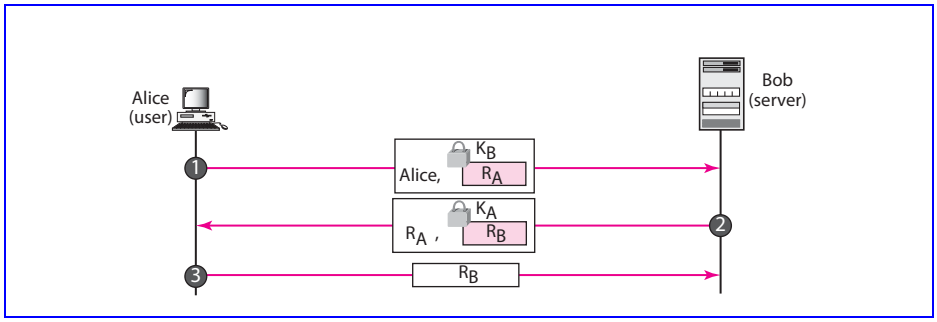
21. Figure 31.2. shows one scheme. Note that the scheme forces Bob to use the timestamp which is related to the timestamp used by Alice (T+1), this ensures that the two messages belongs to the same session.

Figure 31.2 *Solution to Exercise 21*



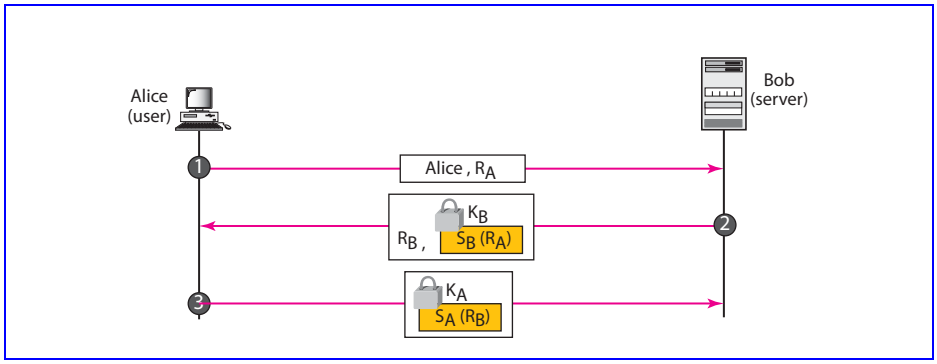
22. Figure 31.3 shows one scheme. In the first message, Alice introduces herself and sends a nonce (R_A) encrypted with Bob's public key. In the second message, Bob decrypts the first message and sends R_A in plain text to authenticate himself. Bob also challenges Alice in the second message by sending his nonce (R_B) encrypted with Alice's public key. In the third message, Alice can authenticate herself by sending Bob's decrypted nonce (R_B). Note that in this scheme, Bob, the server, has been authenticated for Alice, the user, before Alice is authenticated for Bob. However, Bob has not released any information that endangers security.
23. Figure 31.4 shows one simple scheme. Note that in the second message, Bob signs the message with his private key. When Alice verifies the message using Bob's public key, Bob is authenticated for Alice. In the third message, Alice signs the

Figure 31.3 *Solution to Exercise 22*



message with her private key. When Bob verifies the message using Alice's public key, Alice is authenticated for Bob.

Figure 31.4 *Solution to Exercise 23*



24. The ***encryption*** protects the student and the university for the ***first time***. However, the intruder can intercept the encrypted password and replay the process some other times. The intruder does not have to know the password in plaintext; the encrypted password suffices for replaying. The university system cannot determine if the student has encrypted the message again or the intruder is replaying it.
25. The ***timestamp*** definitely helps. If Alice adds a timestamp to the password before encrypting, the university, after decrypting, can check the freshness of the plaintext. In other words, adding a timestamp to a password, is like creating a new password each time.
26. A list of passwords can also help, but the question is how long the list should be. Another problem is that the student must remember to use the next password in the sequence. If she accidentally uses a password out of order, access will be denied.
27. If the ***KDC*** is down, nothing can take place. KDC is needed to create the session key for the two parties.
28.
    a. If the ***AS*** is down, the process cannot start because Alice cannot be authenticated.

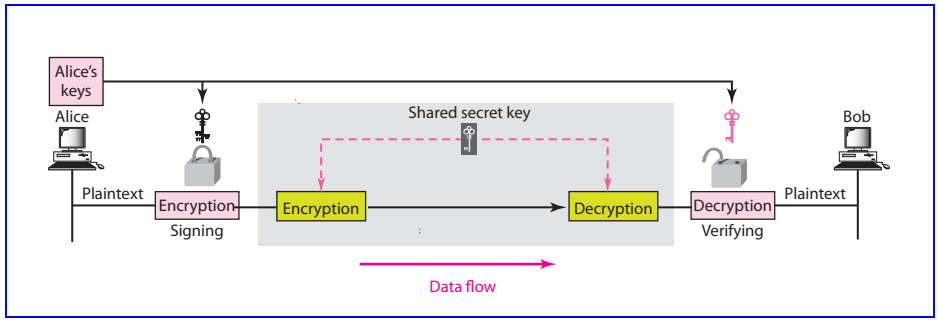
b. If the ***AS*** is running, but the ***TGS*** is down, Alice can be authenticated and get the ticket for TGS, but cannot receive the session key. Alice can apply later and present her tickets to obtain the session key. We can compare the process with air travelling. We need a ticket, but we also need a boarding pass. We can get the ticket if the airline office is open, but we cannot get the boarding pass if the flight is cancelled. We can apply another time, when that particular flight is operational to get the boarding pass.

c. If the ***AS*** and ***TGS*** are running, but the ***main server*** is down, we can get the session key, but we cannot access the main server. Some systems allow the use of a session key in a future time; some do not. The situation is like having the boarding pass to board the air craft. If the flight is delayed, we can wait and apply the boarding pass later. If the flight is cancelled, the boarding passes are probably invalid.

29. If the ***trusted center*** is down, Bob cannot obtain his certificate. Bob still can use his public key if the other party does not ask for a certificate.

30. See Figure 31.5. The shaded area shows the encryption/decryption layer.

Figure 31.5 *Solution to Exercise 30*



31. See Figure 31.6. The shaded area shows the encryption/decryption layer.

Figure 31.6 *Solution to Exercise 31*