1. Classify each of the following as a violation of confidentiality, of integrity, of availability, or of some combination thereof.

a). John peeks at Alice's password when she is logging in.

b). John logs into Alice's account using Alice's password without Alice knowing about it.

c). There is a process running in Alice's machine, which is updating a database from a remote machine. John interrupts the process, results in inconsistent databases.

d). John copies a file from Alice's account and then deletes the file from Alice's directory.

1. Answer:

a). Confidentiality

b). Confidentiality and integrity

c). Integrity

d). Confidentiality and availability

2. Authenticating people is typically based on what you know, what you have, and who you are. Give an example for each of them.

2. Answer:

What you know: password

What you have: smart card

Who you are: biometrics

3. 3 and 65537 are commonly used as the public key. Can they be used as the private key instead? Why or why not?

3. Answer: No. Private key is the key an attacker is trying to figure out. 3 and 65537 are not big enough to counter a brute-force attack starting from zero and counting up. However, they can be used as public key because there is no need to figure out the public key as a public key is *public*.

4. Increasing the key length is one way to increase security of an encryption algorithm against the brute-force type of attack. DES uses 56-bit key, which is not secure, given the modern computing power. Assume that 56-bit key was just sufficient in 1979 when DES was standardized, and assume that the hardware performance improves about 40% per year, then how many bits of a DES key just sufficed in 2003? Until what year would a 112-bit DES key be sufficient?

4. Answer:

40% improvement per year doubles the performance every two years (1.4x1.4≈2). So keys must grow by about 1 bit every two years. (2003-1979)/2+56=68 bit keys sufficed in 2003. (x-1979)/2+56=112, x=2091, so 112-bit key would suffice till year 2091.

5. In an example given in the class, the level of security offered by SHA-256, SHA-384 and SHA-512 is equivalent to that offered by the 128, 192 and 256-bit keys in AES, respectively. Why not SHA-128, SHA-192 and SHA-256? What is the general conclusion you can draw from the example?

5. Answer: in general, the length of message digests should be twice the length of keys in block ciphers to achieve the same level of security. This is because of birthday paradox. It takes $O(2^n)$ to find a message with a given digest, but only takes $O(2^{n/2})$ to find two messages with the same digest. So a message digest needs to be secure against the $O(2^{n/2})$ effort.

6. Most viruses infect your system by implanting themselves into the existing executable files on the disk. Explain how to use a hash algorithm to design a virus detector, which identifies the files that may be infected by viruses.

6. Answer: a virus detector may generate the file digests by applying a hash algorithm on the files and then stores the file digests securely. Then the virus detector periodically computes the file digests and compares them with the stored version. If a virus changes the content of a file, the new digest will be different from the original digest. In this way, a virus detector can detect the modification of a file by a virus.

7.

a) What is a one-time pad?

b) Any good random number generator can be used as a secret-key encryption algorithm. Explain how.

7. Answer:

a) One-time pad is a random bit sequence used to encrypt a message with a simple "XOR" operation. The bits in the one-time pad should be used only once, which leads to the name "one-time" pad.

b) We can use a secret key as the seed of a random number generator to generate a sequence of random numbers, and use the random numbers as one-time pad. To produce the ciphertext, the random numbers (bits) are used to "XOR" with the bits in a message.

8. Consider the following tasks

i) Transmitting data securely over an insecure communication channel

ii) Message integrity check

a) Which task(s) can "Secret key cryptography", "Public key cryptography", and "Hash algorithms" achieve, respectively?

b) Explain how "public key cryptography" achieves i).

8. Answer:

a) Any one of "secret key cryptography", "public key cryptography", and "hash algorithms" can do both tasks.

b) Two communication parties should each have a <public key, private key> pair. Assume Alice and Bob want to communicate with each other over an insecure channel. Alice should know Bob's public key and Bob should know Alice's public key. Alice encrypts her messages using Bob's public key and sends the ciphertext to Bob. Bob decrypts the received messages by using his private key. Similar, Bob uses Alice's public key to encrypt the messages for Alice. Alice decrypts the messages with her private key. Hence, they can transmit messages securely.

9. A keyed hash is a hash function involving a key, which is typically used as the cryptographic checksum for integrity protection. Let MD be a hash algorithm that maps an arbitrary message to a fixed-length message digest. One way to perform a keyed hash is MD(key|message), i.e., hashing the concatenation of the secret key and the message.

a) Explain why this method of performing a keyed hash is not secure.

b) Describe a different approach to perform a keyed hash that is more secure.

9. Answer:

a) The keyed hash, MD(key | message), is not secure because an attacker can append additional information to the message and generates the correct message digest even though the attacker does not know the key. Assume the attacker intercepts the message together with MD(key | message). He/she can concatenate the padding and some addition bits to the end of the message, and then initialize the message digest computation with MD(key | message). Since the key appears only at the beginning and MD(key | message) is the correct intermediate result of hashing, the computation (MD4, MD5 or SHA1) can continue with the appended bits. Therefore, the attack is able to compute the message digest of the modified message correctly without knowing the key.

b) There are several ways to avoid that flaw; HMAC is the provably secure approach. HMAC concatenates the secret to the front of the message, digests the combination, then concatenates the secret to the front of the digest, and digests the combination again. This nested digest with secret inputs to both iterations prevents the attacks of extending a message, which would be possible if we simply digested the key and message once.

10.

a) Describe the advantages of CBC over ECB.

b) Explain how CBC can be used for: 1) integrity protection only, 2) both integrity and privacy

10. Answer:

a) The same block repeating in the plaintext will not cause repeats in the ciphertext.

b) For the integrity protection, the ciphertext of the last block of the plaintext is used as the crypto checksum, also called MAC (message authentication code) or CBC residue. The sender sends the plaintext message together with the CBC residue to the receiver, which performs the CBC on the received message and uses the ciphertext of the last block to verity the CBC residue.

To achieve both encryption and integrity protection, there need two passes of CBC with two different keys. The first pass calculates the CBC residue, and the second pass encrypts the message together with the CBC residue.

You need to know how CBC encryption operates!

11.

a) Describe how Diffie-Hellman negotiates a common secret between two remote parties.

b) How about three or more parties?

c) Explain why Diffie-Hellman is subject to the Man-in-the-Middle attack.

11. Answer:

a) For two remote parties, Alice and Bob, they should first agree on a large prime number **p** and a number **g** that is less than p. These two numbers can be sent to each other in clear text or publish in a public place both of them can access. Then Alice and Bob each chooses a 512-bit number at random and keeps it secret ($\mathbf{S_A}$ for Alice, $\mathbf{S_B}$ for Bob). Each raises g to her/his secret number, mod p ($\mathbf{T_A} = g^{SA} \bmod p$, $\mathbf{T_B} = g^{SB} \bmod p$), and the result is called the public number. Then they exchange the public numbers with each other. Finally, each raises the received public number to her/his secret number, that is, Alice computes $T_B^{SA} \bmod p$, and Bob computes $T_A^{SB} \bmod p$. Alice and Bob reached the same secret $g^{SASB} \bmod p$, which is then used as the shared key.

b) Similar to the two-party case, now every party will send his/her public number to every other party, so the resulting shared key will be $g^{SASBSC\cdots} \bmod p$

c) Diffie-Hellman is vulnerable to the Man-in-the-Middle attack, because two individuals agree on a shared secret key with no authentication. When Alice receives T_B from an insecure media, there is no way for her to know for sure whether the number came from Bob or from someone else pretending to be Bob. Consequently she will establish a secret key with whoever transmits T_B. A third person may thus claim to be Bob and establish a shared secret with Alice; he may then claim to be Alice and establish a shared secret with Bob; and finally he simply relays the conversation between Bob and Alice, knowing the full conversation and without being detected.

You can draw diagrams instead of describing by words. However, you need to include every important detail described above in the diagram.

12.

a) What is the advantage(s) of using a KDC (Key Distribution Center) rather than having every two principals in the system sharing a secret key?

b) What secret information should be pre-configured in a KDC and in each principle?

c) What information should be included in a ticket and how should a ticket be encrypted?

You need also know the KDC-based authentication procedure: Lecture 7, Slide 20.

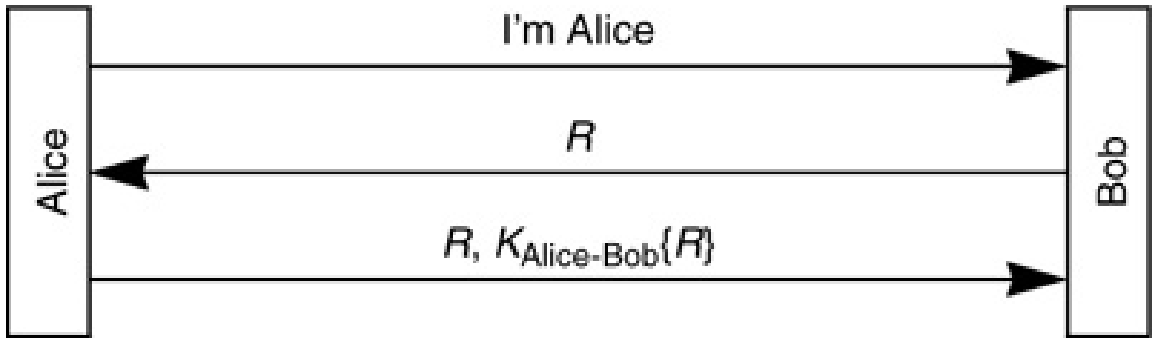
12. Answer:

a) When the secret key cryptography is used, each pair of principals (computers or users) may need to authenticate each other, which means every principal has to share a different secret with every other principle. N x (N – 1) / 2 keys are needed in total for a network consisting of N principals. When a new principal is added to the system, the keys have to be securely distributed to all existing principals in the system. For a large system with thousands of principals, this is not acceptable. KDC (Key Distribution Center) is needed to solve the problem of efficient key management. Every principal shares a master key only with KDC; when a new principal is added into the system, only one key is configured between the new principal and KDC

b) A secret key needs to be pre-configured between each principal and the KDC.

c) Ticket contains a session key, an expiration time, and the sender's identity (e.g., name), encrypted by the receiver (Bob)'s master key.

13. Suppose we are using a three-message mutual authentication protocol, and Alice initiates contact with Bob. Suppose we wish Bob to be a stateless server, and therefore it is inconvenient to require him to remember the challenge he sent to Alice. Let's modify the exchange so that Alice sends the challenge back to Bob, along with the encrypted challenge. So the protocol is:



Is this protocol secure? Explain. How to make it secure?

13. Answer:

No. It is subject to the replay attack. An eavesdropper can replay Alice's messages at any time. If Bob does remember his current challenge, he won't know that the response is to a previous challenge. We can make it secure by Bob sending timestamps as the challenge (assuming the attacker cannot replay the messages fast enough so that the timestamp in the message is acceptable to Bob).