Problem 1.

Answer

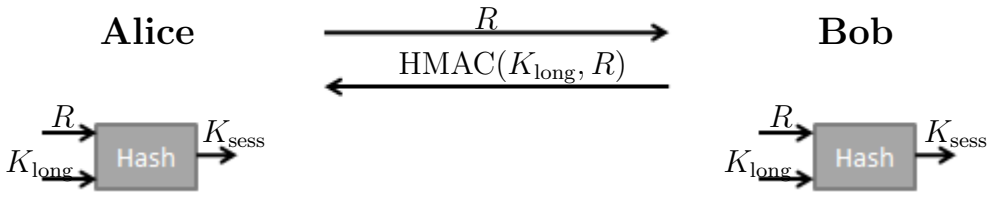
A forgery can be done either by obtaining the private key or by finding a collision in the hash function that is used in the signature. The private key can e.g., be stolen or an attempt to compute it from the public key can be done. (3 points)

Problem 2.

Answer

a) A hash algorithm is suitable. (Other answers, such as MAC or block cipher is acceptable if it is clearly motivated.)

b) Since we want implicit key authentication we wish to be certain that no one else besides a party knowing the long-term key can be able to compute the session key. Unilateral authentication can be obtained through a challenge-response protocol. Assume that Alice wishes to authenticate Bob. A simple protocol providing this would be



(1+2 points)

Problem 3.

Answer

The Hello messages will include random values from both the client and the server and the finished messages will include an encrypted hash of the previous messages in the handshake, including the random values. If an attacker replays messages from e.g., the client side, the finished message will not be correct as it will include the wrong server generated random value. (3 points)

Problem 4.

Answer

a) It provides integrity protection against unintentional modification, but not against an active attacker who can make changes to the header and recompute a valid checksum.

b) It does not change anything. A man-in-the-middle will still be able to make any changes and recompute the hash value. (1.5+1.5 points)

Problem 5.

Answer

a) In Alice's view, the validity of Bob's public key is *full*. The trust is not set implicitly (it is unknown), but has to be decided explicitly by Alice.

b) The confidence is decided by the validity of the key. If Alice decides to fully trust Bob, then the validity is full. If Alice decides to only marginally trust Bob, then there has to

be either one other signature on Charlie's key by someone Alice fully trust or two other signatures by people who Alice has marginal trust in. If Alice does not trust Bob at all then there has to be either one other signature on Charlie's key by someone Alice fully trust or three other signatures by people who Alice has marginal trust in. (1+2 points)

Problem 6.

Answer
There are four IDs associated with the process and they take the following values: Real user ID: Bob
Effective user ID: Alice
Real group ID: Bob's primary group
Effective group ID: Students (3 points)

Problem 7.

Answer
At least the following properties should hold for a correctly stored password:

a) Passwords must be stored in hashed form.

b) Each user must have a unique salt associated with the password. This salt should be used as input to the hash function together with the password.

c) The hash function must be slow so that brute force or dictionary attacks only can test few passwords per second (typically in the order of a few hundred).

(3 points)

Problem 8.

Answer
In Orange book, functionality and assurance is tied together. Each evaluation level sets rules for both. In common criteria, the security target states the functionality of the product and the evaluation level states the assurance. Thus, functionality and assurance is separated in common criteria. (3 points)

Problem 9.

Answer
ss-property is broken since Erika should not be allowed to write to file_d.
***-property** is broken since Alice is reading file_a and writing to file_b at the same time. This is not allowed since $f_O(\text{file\_a}) > f_O(\text{file\_b})$ which means that information from file_a can flow to file_b, i.e., downwards.
ds-property is broken since Charlie is not allowed to append to file_e according to the Access Control Matrix. (3 points)

Problem 10.

Answer
The flaw was that a stream cipher was used in order to compute the response as response = challenge $\oplus$ keystream. By xoring challenge and response the keystream could be used to either authenticate again (using the same IV as before) or to decrypt any traffic using the same IV. (3 points)

Problem 11.

Answer
a) The antivirus, acting as a MitM, will capture traffic before it leaves the computer and also capture returning traffic before it is delivered to the application. When installing the antivirus product, the user also installs a trusted CA certificate with the product's public key. This will allow the MitM to create legitimate certificates on the fly for any requested webpage. The MitM will then act as the webpage to the application and set up its own connection to the real webpage.
b) As a MitM, the antivirus will be able to read all incoming and outgoing traffic in unencrypted form. This will make it easier to detect malware or other types of attacks to the computer it protects. (3+2 points)

Problem 12.

Answer
a) When injecting code into a buffer, it is difficult to know the exact location of where it will end up. The return address should be chosen to be the start of the code and by starting the code with a number of NOP instruction, the attacker gives himself an error margin when guessing the location of the code.
b) A canary is a value inserted between local variable and the return address in the stack. The idea is that when overflowing a buffer in order to overwrite the return address, this will also overwrite the canary. Checking the integrity of the canary before returning from a function will detect an attack.
c) ASLR will randomize the location of the stack, heap and libraries in order to make it virtually impossible for an attacker to correctly guess the value of the return address to use in the attack. (2+2+1 points)

Problem 13.

Answer
We know that $n = 221$ is the product of two primes. Testing a few primes gives us $221 = p \cdot q = 13 \cdot 17$. Thus, $\phi(n) = (p-1)(q-1) = 12 \cdot 16 = 192$. In RSA we have that $e \cdot d \equiv 1 \mod \phi(n) \Rightarrow e \cdot d = k \cdot \phi(n) + 1$, for some integer k. With $k = 2$ we find that $d = (192 \cdot 2 + 1)/7 = 55$.

The next step is to find $2^{55} \mod 221$. This is easy once we note that $2^{2x} \mod n \equiv (2^x \mod n)^2$, and that $55 = 1+2+4+16+32$. Using the table, we have $2^{55} \equiv 2 \cdot 4 \cdot 16 \cdot 120 \cdot 35 \equiv 128 \mod 221$. Note that the last two lines in the table implies that $120 \cdot 35 \equiv 1 \mod 221$ so the calculations becomes very simple. (5 points)

Problem 14.

Answer

a) A MAC computed using a hash function.

b) A block cipher mode where each block is encrypted independently from other blocks.

c) IPsec protocol which offers both integrity and confidentiality.

d) A network connected computer which does not hold any valuable information, but exists in order to draw attention to attackers.

e) An abstract machinery with the purpose of making access control decisions. (5 points)
