HW #1 - CNS Sapienza

Daniele Oriana 1457625

29 Oct 2015

20150720-Q2

Diffie-Hellman
Q2.1 [3/30] Describe in detail the Diffie-Hellman exchange of keys.
Q2.2 [3/30] Explain in detail how an attacker can carry out a man-in-the-middle attack and what results it can provide.
Q2.3 [2/30] Suggest a mechanism for securing Diffie-Hellman with respect to man-in-the-middle attacks.

Answer

Q2.1

Diffie Hellman is a cryptographic protocol that can allow to two entities to establish, and subsequently share, a secret key, even in a public context (and therefore potentially insecure). The algorithm which is the basis of this protocol was proposed by Whitfield Diffie and Martin Hellman in 1976 taking a new direction in cryptography. The procedure that characterizes the protocol is the following (we call for convenience the two entities Alice and Bob) :
Alice defines a prime number p and a number g that is the primitive root modulo p. After that Alice randomly chooses a number ”a” in the range [1 ... p-1], calculated the value A= g^{a} mod p and sends to Bob the values A, p and g. At this point Bob randomly chooses a number ”b” in the range [1 ... p-1], calculated the value B= g^{b} mod p and sends it to Alice. In order to obtain the secret key Alice can calculate K_{A}= B^{a} mod p = g^{ab} mod p, in the same way Bob can calculate the secret key K_{B}= A^{b} mod p = g^{ab} mod p. We can see that the two entities involved share the same key in this way. The strength of this approach is that even if an

attacker listens the whole conversation between Alice and Bob he could not calculate the values a and b, because he should compute a discrete logarithm, that is a good candidate to be a one-way function and, for this reason, computationally difficult to calculate. The term one way function denotes a function which is computationally easy to calculate, but its inverse is computationally difficult to calculate. The use of this one way function represents the great innovation introduced by Diffie and Hellman, in fact through this function we can generate a secret key in a public context. The key obtained through this scheme can be later used to encrypt subsequent communications using a symmetrical encryption scheme. At the end of the session between the two entities the secret key is discarded and for every new session a new key is calculated, then Diffie - Hellman is characterized by perfect forward secrecy.

Q2.2

A man in the middle attack is lethal for the cryptographic protocol Diffie - Hellman. An attacker can carry a man in the middle attack if he is able to intercept all messages traveling between the two entities, interposing between them; after the interceptions the attacker can see all the messages and he can also modify them. If we suppose to call the attacker Mallory we have this initially situation:

Figure 1: Initial configuration.

Now when Alice sends her public parameters these are intercepted by Mallory that sends his public parameters to Bob. Later Bob sends B value that is intercepted by Mallory that, once again, sends it to Alice. After this exchange of public keys Alice and Bob believe that they are communicating securely, instead the entire connection is controlled by Mallory, so the attacker has created a man in the middle attack. (Figure 2)

This vulnerability is present because Diffie-Hellman key exchange does not authenticate the participants and therefore Mallory can see al the messages going between Alice a Bob and can also modify these messages.

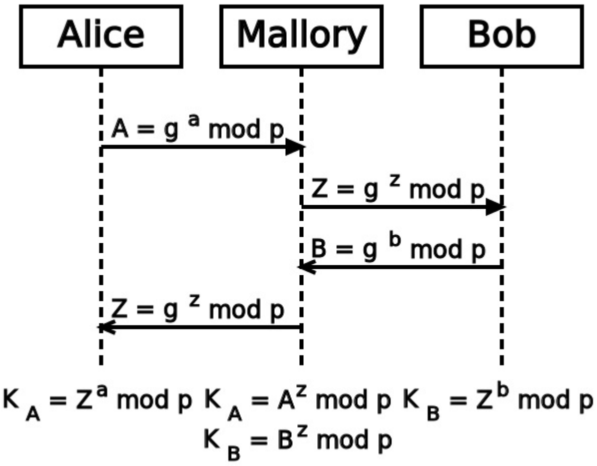

Figure 2: Man in the middle attack.

Q2.3

We have verified that the question is related to matters not yet faced.

20150210-Q1

Data integrity
Q1.1 [3/30] Describe what we mean by data integrity and discuss the use of keyed HMACs for guaranteeing the integrity of a file being transmitted over the network (no other guarantees requested).
Q1.2 [3/30] Suppose you are requested to ensure the integrity of a file but you are only allowed to use AES (and a symmetric key): what can it be done?

Answer

Q1.1

With the concept data integrity we refer to maintaining and assuring the consistency of some data in a determinate context. We can consider data integrity as the opposite of data corruption and data loss. Obviously this concept is often very important, but this importance depends on the context in which we are considering it. For example, in network security, a loss of

data integrity can involve the failure of all the system that is the object of our examination, in fact if an adversary is able to modify a message going between two entities he can obviously change the meaning of the message, creating some important problems.
HMAC is an authentication code that guarantees authentication (as suggested by the name) and data integrity. It is based on a secret key K and on an hash function H. It is characterized by the following procedure: first of all the message to send is divided into some blocks of j length, after that it is choosen a secret key K, at this point if K is longer than j bits it is calculated the hash function of K, that is K'=H(K). K' is called HMAC's key. If |K|=j bits we impose K'=K and if $|K| < j$ bits we impose K'= k + zero padding. After the calculation of K' we can compute the HMAC function in this way: HMAC$_{\mathrm{K}}$ (M,H)=H((K' xor opad)||H((K' xor ipad)||M')), where ipad=00110110 repeated j/8 times, opad=01011100 repeated j/8 times and M' is the suddivision of M in blocks.
This kynd of procedure allows to create data integrity, in fact if an entity A sends to B the pair (M, HMAC$_{\mathrm{k}}$(M)), the other entity B can check the integrity of the data calculating the HMAC$_{\mathrm{K}}$(M) from the message send by A in the pair; if the HMAC$_{\mathrm{K}}$ calculated by B is the same of the HMAC send by A the data integrity is respected.