ITIS 6200/8200 Principles of Information Security and Privacy

Jinpeng Wei

Spring 2018

Homework 2

Due time: February 2nd, 2018 before 11:00 am

1. We have a symmetric encryption algorithm $E_K(M)=C$. Here K is the secret key, M is the plaintext, and C is the ciphertext. We (and the attacker) know that the key length is 256 bits. The attacker eavesdrops on the communication line and gets a copy of the ciphertext C1. Now the attacker decides to conduct the brute force attack and try every possible key to get the plaintext M1. Let us assume that there is only one possible M1 and if the attacker sees it, he will know that this is the correct one. The attacker has 1,000,000 machines, with each machine having the capabilities to try 5,000,000 decryptions of C1 with different keys per second. If one machine finds the right key, it will automatically notify the attacker. Now please answer, how many years (roughly) does the attacker need to try 30% of the keys? Note that Google has around 2 million servers. Also, check the Internet and see what the expected life time of the Sun is. Can you crack the key before that?

   Ans:

   1. Key length =256 bits. The number of possible combination = 2256 bits.
   2. Calculation done by attacker: 5000000*1000000 = 5*1012 keys/second.
   3. Total years if attacker tries 30% of the keys = (2256 *30)/ *1012*60*60*24*365*100) = **2.2*1056 years**

2. Bob has a public-private key pair (pub_Bob, pri_Bob). Alice needs to send some information to Bob. She wants to make sure that when Bob opens the message, he can verify that this is from Alice but not anyone else. So she sends out the message as: [Alice, E_{pub_Bob}(message)] to Bob. Basically, she first sends out her name in clear text, then encrypts the message with Bob's public key. Please discuss, can an attacker M impersonate Alice and send out a packet in Alice's name? How can he do it? Here we assume that M also has the public key of Bob. For the same question, if Alice sends out [E_{pub_Bob}(Alice, message)], can M still impersonate Alice? (Here Alice puts her name in the encryption.)

   Ans:

   Case1: The attacker here has the public key of Bob and since Alice has not encrypted her name but sent in plain text, the attacker can impersonate Alice. Anybody with the public key of Bob can send him a message. The attacker M can capture the message, remove it and send a new message to Bob and it is not possible to know if the message is really from Alice or not.

   Case2: Here the attacker cannot impersonate Alice as Alice has encrypted her name and the message which can be decrypted by Bob's private key. As the attacker does not have Bob's private key the attacker cannot capture the message, change it and send some other message to Bob.