Introduction to Network Security – Part 11

Posted by Alejandro G. Carlstein Ramos Mejia on December 15, 2010 December 15, 2010 About Programming / Network Security

NOTIFICATION: These examples are provided for educational purposes. The use of this code and/or information is under your own responsibility and risk. The information and/or code is given 'as is'. I do not take responsibilities of how they are used. You are welcome to point out any mistakes in my posting and/or leave a comment.

Key Distribution Using Public-Key Cryptography

In the previous post, introduction to network security – part 10, we saw three main methods of public-key:

1. Public announcement,
2. Public-key authority, and
3. Public-key certificates

These methods can be used for encryption and decryption of messages (secrecy) and/or authentication.

These methods the disadvantage of being slow; therefore, its common to use symmetric-key encryption for secrecy and distribute using public-key encryption session keys. In this way we use the advantage of the speed of symmetric-key encryption and the security of public-key encryption.

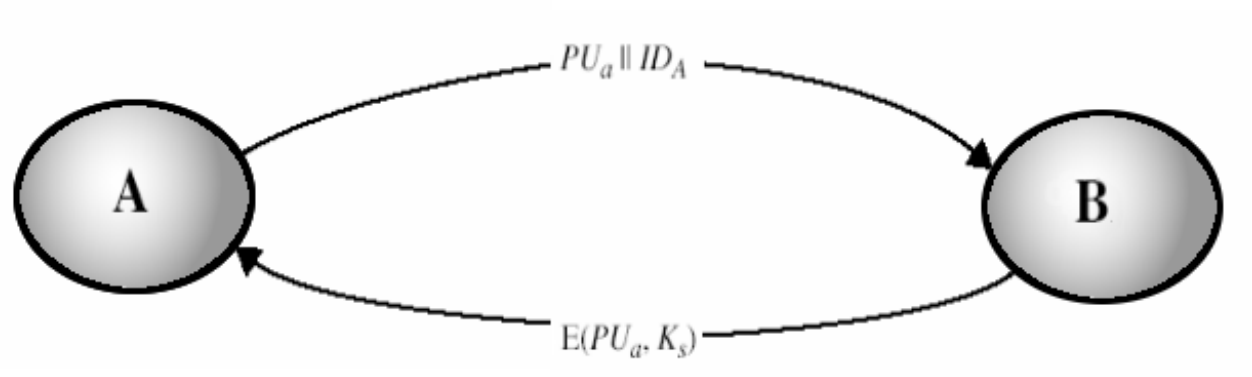
Simple Key Distribution

In 1979, Ralph C. Merkle created his thesis entitled "Secrecy, authentication and public key systems" which let him receive his Ph. D. in Electrical Engineering at Stanford University <http://en.wikipedia.org/wiki/Ralph_Merkle>.

For a key distribution, Merkle proposed:

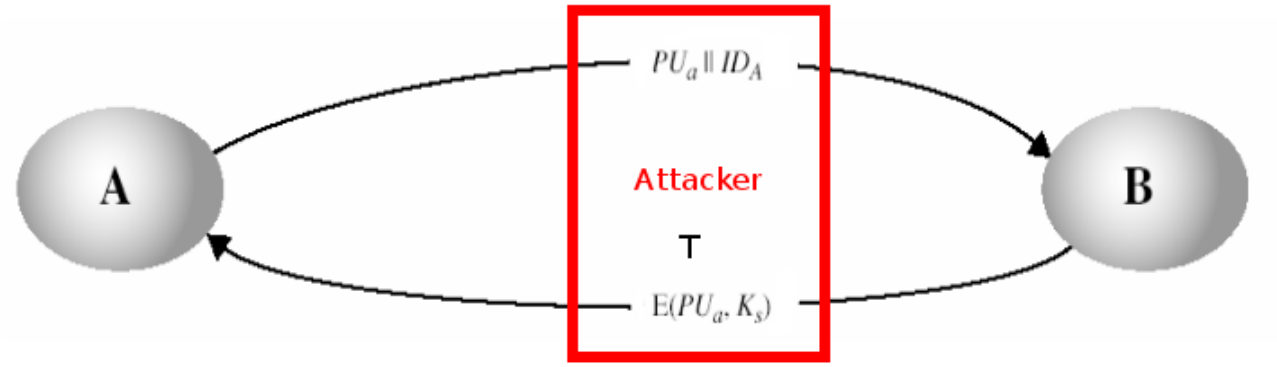
1. User A will generate a new temporaty public key pair, *PUa*
2. User A send the public key, *PUa*, to user B together with its identity, *IDa*
   PUa, IDa
3. User B generate the session key *K*.
4. User B uses the public key, *PUa*, supplied by user A to encrypt the session key *K*.
   Then user B send the encrypted session to user A
5. User A decrypt the message to obtain the session key *K*.
6. User A discards the public key *PUa*
7. User B discards user A's public key, *PUa*.
8. After the exchange of information is complete, user A and B discard the session key

K.



The Man-In-The-Middle Attack

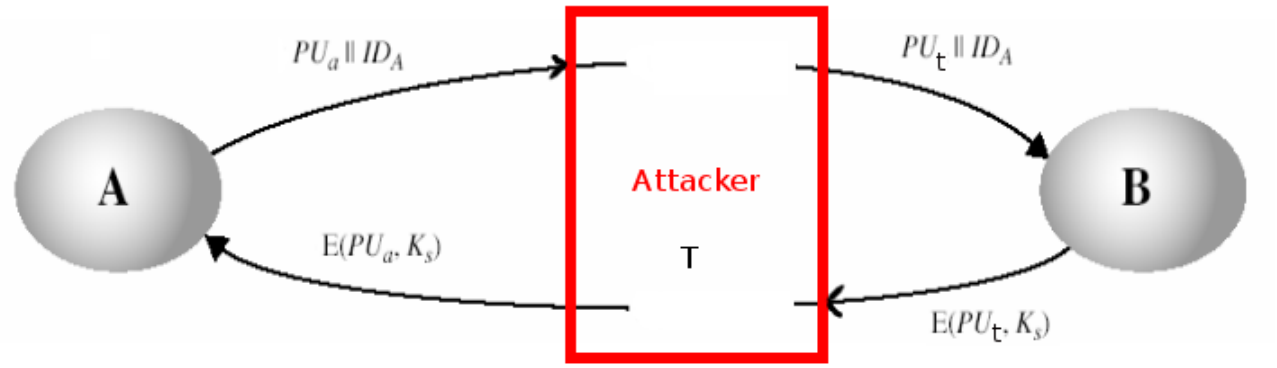
This type of key distribution have a disadvantage. Lets assume that we have an attacker that gets in the middle of the communication in a way that this attacker can intercept the messages and then replay this message, modify this message, or send another different message.



Lets analyse this problem:

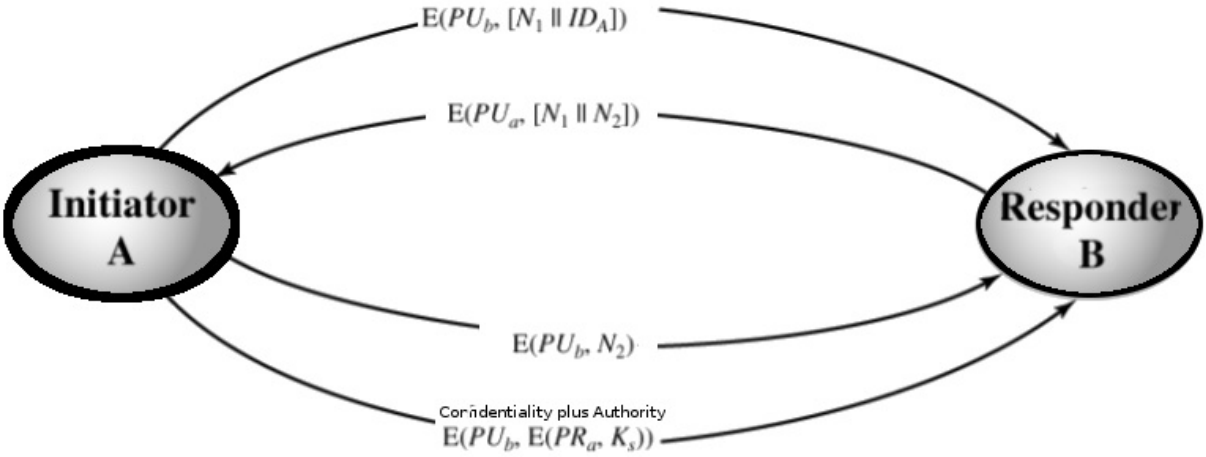
1. User A send a message to user B which holds the public key PUa, and user A's identifier IDa
2. The attacker T intercept this message and create its own pair keys, public key *PUt* and private key *PRt*:
   {PUt, PRt}
3. The attacker T send to user B, its own public key *PUt* together with the user A's identification *IDa* :
   PUt||IDa
4. User B generate a session key *Ks*. Then user B send this session key Ks encrypted using the public-key *PUt* that he received thinking that it came from user A.
   Ciphertext = E(PUt, Ks)
5. The attacker T intercept the message obtaining the session key *Ks* by decrypting the message with his private key *PRt*.
   Ks = D(PRt, Ciphertext) = D(PRt, E(PUt, Ks))

6. Then attacker T send the key session *Ks* to the user A using user A's public key *PUa*
7. Without user A and B knowing, the attacket T obtained the session *Ks* successfully.



Solution to The Man-In-The-Middle Attack

1. The process begins with user A. User A encrypt the message containing the user A identification *IDa* plus a nonce N1 using the user B's public key PUb
2. User B generate a new nonce *N2* and encrypts the message containing user A's nonce *N1* plus a new nonce *N2* using the user A's public key.
3. Since user B is the only one that could decrypted the first message coming from user A plus the new message send from user B to user A will contain the nonce *N1* (given by user A in the first message), user A will know the new message is coming from user B and not an attacker.
4. User A will encrypt nonce *N2* using the public key *PUb* of user B. Then user A will send then encrypted nonce *N2* to user B. In this way, since nonce *N2* was generated by user B, when user B find nonce *N2*, user B will known the message came from user A.
5. User A generate a secret key *Ks*. User A will encrypt first the secret key *Ks* using the private key *PUa* of user A which would provide authentication, and then it will encrypt the output of the encryption with the public key *PUb* of user B to produce a new ciphertext *M* which provide confidentiality.
6. User B decrypt the ciphertext *M* by decrypting the ciphertext *M* using the private key *PUb* of userB, and the result will be decrypted again using the public key *PUa* of user A. In this way the secret key *Ks* is obtained.



Hybrid Key Distribution

Public key encryption is an algorithm that require a lot of processing. In a system that require to distribute session keys thought many users and require a frequently change of session keys, the public key encryption can slow the performance of the system as the load on the system keep increasing. One solution to this problem is to use an hybrid of different key distribution.

In an hybrid key distribution, the key distribution center (KDC) will be in charge of distributing a master key *MK* to each user of the system plus perform the distribution of session keys. Before these session keys are distributed, they will be encrypted by using the master key *MK*. Also, the master key is encrypted using a public key encryption. Since the master key only update in few occasions then the load of the system is reduced.