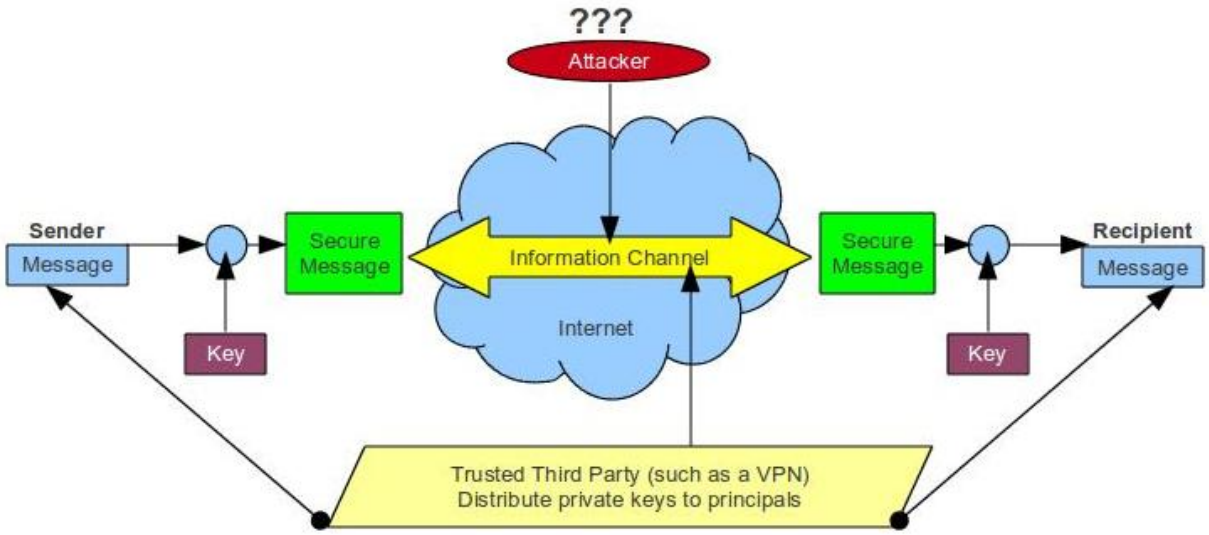
# Introduction to Network Security – Part 8

acarlstein.com/

Posted by Alejandro G. Carlstein Ramos Mejia on December 11, 2010 December 11, 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.

In security, we use a system of key in order to work on encryption and decryption. The most common system used are the Symmetric Key Encryption and the Public Key Encryption




**Symmetric Key Encryption**

In a symmetric  system, one key is used for the encryption of a plaintext to a ciphertext and for the decryption of the ciphertext to a plaintext.

The key must be distributed in a secure way to the sender and the receiver making sure that the key is not disclose since then the communication could be compromised. The possible disclose of the key is one of the disadvantages of this system.

Another disadvantages of this system are:

1. There is no way to prove the message was send by the original sender and not from an intruder.
2. The recipient could change the message and say it came from the sender.

**Public-key Encryption**

In the public key system, normally two keys are generated (pair keys). One key is used to encrypt the message and another key is generated to decrypt the message.

The key that was used for the encryption of the message cannot be used for the decryption and the key used for the decryption of the message cannot be used for the encryption of the plaintext.

One key is the public key which is going to be used for the encryption of the plaintext to the ciphertext and for the verification of the signatures.

The other key is the private key which is going to be used for the decryption of the ciphertext to a plaintext and the generation of signatures.

This system can be used for:

1. Authentication: Verify that the message came from the corresponding sender and the message is received to the corresponding receiver
2. Confidentiality: Create a message that cannot be decrypt by an attacker
3. Authentication and Confidentiality

However, this system still have some main issues such as:

1. Key distribution: In the same way that the symmetric key encryption, there have to be a secure way to distribute keys.
2. Digital Signatures: The way to verify that the message is coming for the sender and not an attacker.

The public-key encryption is considered to be an asymmetric system. This means that those who encrypt the plain-text and/or verify the signatures cannot decrypt the message or create signatures.

In order for a public key encryption to be feasible, it must:

1. Make harder for an attacker to find the key used for the decryption of the ciphertext by just knowing the algorithm and the key used for the encryption of the plaintext.
2. To provide an easy way to decrypt the ciphertext when the key for decryption is used.
3. To provide a way in which either, the private key or public key, can be used for the encryption and the other key used for the decryption of the message. System that implement this policy is called RSA.

This is the way that normally pubic key works:

1. Each user generate a pair of key that will be use for the encryption and decryption.
2. Each user place one key (the public key) to a public register while holding the private key to themselves (the private key is never distributed).
3. In case the private key is change, then the user must generate a new public key that

will replace the older public key.

**Symmetric Key Encryption Versus Public-Key Encryption**

Before we go in deep comparing both encryption systems let clarify some points:

1. The security of both system depend directly on the key/s length. The largest is the key, the harder is to break the cipher.
2. While the public key may provide more security than symmetric key, it produce an overhead. This is the main reason that symmetric key is not considered obsolete with the apparition of the public key encryption.

Here are the differences between symmetric key (conventional) and public key:

1. Symmetric key: Same algorithm using the same key is used for encryption and decrytion.
   Public-key: One algorithm is used for encryption and decryption but a pair of keys are generated. One key is used for the encryption, another is used for the decryption.
2. Symmetric key: Sender and receiver must use the same algorithm and share the same key.
   Public-key: Sender and receiver must use the same algorithm, but each user must create a pair key. One of those keys (the public key) must be distributed from the receiver to the sender. The other key (private-key), the receiver must kept this key and make sure it doesn't not get distributed.

Things that need to be resolve from the point of view of security:

1. Symmetric key: The shared key must be kept in secret
   Public-key: One of the two keys (normally the private key) must be kept in secret.
2. Symmetric and Public-key: It should be very hard for an attacker to decipher a message if there is no information available.
3. Symmetric key: Even do the attacker may have knowledge of the algorithm and have possession of the ciphertext, it should be very hard to obtain the plaintext and/or the shared key.
   Public-key: Even do the attacker may have knowledge of the algorithm, samples of the ciphertext, and the public key, it should be very hard to obtain the plaintext and the other key.

**How to Use Public Key Encryption**

The public-key encryption can be used to provide:

1. Confidentiality: Prevent attackers to know the content of the message
2. Integrity: Prevent attackers for modifying the original message
3. Authentication: To verify that the sender and/or receiver is not an attacker disguising as the sender and/or receiver

4. Digital Signature:  To verify that the message is send by the sender and not the attacker

Confidentiality (secrecy):

1. For a plaintext X where X = [X1, X2, ..., Xn]
2. User A will generate two keys: Public key (PUa) and Private key (PUb)
3. User B will generate two keys: Public key (PUb) and Private key (PUa)
4. For A to send a message to B, A will receive the public key (PUb) from B.
5. User A will encrypt the plaintext (X) using the public key (PUb) from user B with the encryption algorithm (E) to generate the ciphertext (Y).
   Y = E(PUb, X)
6. User B will receive the ciphertext (Y). Using private key (PRb) with the decryption algorithm (D), user B will obtain the plaintext (X).
   X = D(PRb, Y)

Authentication:

1. User A generate a plaintext for user B. User A encrypt the plaintext (X) using the private key (PRa) and the encryption algorithm (E) then user A send the ciphertext (Y) to user B.
   Y = E(PRa, X)
2. User B receive the ciphertext (Y) and using the public key (PUa) with the decryption algorithm (D), user B obtain the plaintext (X).
   X = D(PUa, Y)

Even do this provide authentication and provide safety against the alteration of the message, it does not provide confidentiality because:

1. This Authentication do not prevent from eavesdropping.
2. An attacker can decrypt the ciphertext (Y) using user A public key (PUa).

Since tthe message can be prepare only for user A because it was encrypted by using user A's private key (PUa). this message can be used for the purpose of digital authentication (we can assure the message comes from user A since he provide the public key), and it provide data integrity ( prevention against alteration of the message) since it is impossible to alter the message without the private key (PRa).

Confidentiality and Authentication:

By using the the properties of Confidentiality and Authentication, we can create a scheme that provide more security.

1. User A generates a pair of keys (PUa and PRa) while user B also generates a paid of keys (PUb and PRb)
2. Sending the message: User A uses the private  key (PRa) with the encryption algorithm (E) to encrypt the plaintext (X) to a ciphertext (Y). Then user A uses the

public key (PUb) from user B with the encryption algorithm (E) to encrypt the ciphertext again to a new ciphertext (Z).
Z = E(PUb, E(PRa, X))

3. Receiving the message: User B receive the ciphertext (Z) from user A. User B uses the decryption algorithm (D) with the private-key (PRb) with the ciphertext (Z) to produce ciphertext (Y). Then user B uses the public key (PUa) from user A with the decryption algorithm (D) to decrypt the ciphertext (Y) to the plaintext (X).
X = D(PUa, D(PRb, Z))

**Requirements for Public Key Encryption**

1. It should be easy for user A to generate a pair of keys: Public key (PUa) and private key (PRa).
2. It should be easy for user B to generate a pair of keys: Public key (PUb) and private key (PRb).
3. It should be easy for user A to encrypt the plaintext (M) to a ciphertext (C) using the public key (PUb) from user B.
C = E(PUb, M)
4. It should be easy for user B to decrypt the ciphertext (C) to the plaintext (M) using the private key (PRb).
M = D(PRb, C)
Since C = E(PUb, M) then M = D(PRb, E(PUb, M))
5. It should be very hard for an attacker while knowing the public key (PUb) from user B to guess correctly the private key (PRb) of user B.
6. It should be very hard for the attacker while knowing the public key (PUb) from user B and the ciphertext (C) encrypted with the public key (PUb) to obtain the plaintext (M) send by user A to user B
7. Both keys should be able to be used in either order for the encryption and decryption:
M = D(PUb, E(PRb, M)) = D(PRb, E(PUb, M))

Algorithm such as RSA follow these requirements.