PUBLIC KEY CRYPTOGRAPHY:

A BRIEF SNAPSHOT

BY – DEEPAK PAREEK

WHAT IS IT?

This is an overview of the securing and de-securing process that PKI {Public Key Infrastructure} uses to secure files. If you are new to the area of public key cryptography, it may answer some of your questions.

The four security services

When using any form of communication, there are a number of security risks. The main risks are that someone will:

- Intercept a message and read its contents
- Send a message under someone else's name and signature
- Change the contents of a message
- Deny sending a message.

To reduce these risks, four security services have evolved over time:

- **Confidentiality**-evidence that the contents of the message have not been disclosed to third parties
- **Authentication**-a guarantee that a message really has come from the person who claims to have sent it
- **Integrity**-proof that the message contents have not been altered, deliberately or accidentally, during transmission
- **Non-repudiation**-the certainty of knowing that the sender of the message cannot later deny having sent it.

What are Private and Public keys?

In the traditional mail system, your signature, a sheet of letterhead paper, and a sealed envelope provide the four security services. To provide these services electronically, PKI uses a technique called public key cryptography.

In a Public Key system, each user has two pairs of keys (a total of four keys): two Private keys that are kept secret, and two Public keys that are made available to all the other users. PKI uses one pair of keys to encrypt your files (Confidentiality keys) and the other pair to create digital signatures that you use to sign your messages (Authentication keys).

We use two sets of keys because good cryptographic practices dictate the use of different keys for different purposes, i.e. one for confidentiality and another for authentication.

When you sign a file, PKI uses your Private Authentication key to generate a unique digital signature, which it attaches to the file. When you want to encrypt the file, PKI generates a random key which it then encrypts using the Public Confidentiality key *of the person to whom you are sending the message.* This means that only the intended recipients can decipher the file, because they have the corresponding Private Confidentiality keys needed to decrypt the randomly generated encryption key.

Confused?
Don't worry if you don't quite grasp the concept, it's quite difficult to understand. The next section goes into a little more detail, and includes some diagrams that show what happens in the background when you use PKI.

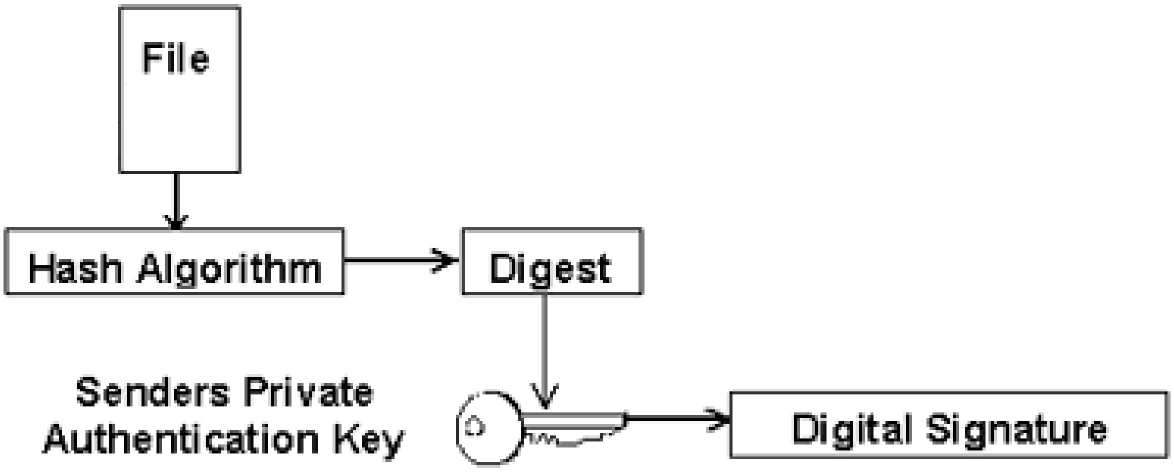
What happens when you secure a file?

Let's say Alice wants to send a file to Bob, but doesn't want Eve eavesdropping and intercepting her message while it passes through the network.

Creating a digital signature

When Alice uses PKI to secure her document, it does the following things:

1. Passes her file through a 'hashing' algorithm that creates a smaller 'digest' of the file. This digest is a number that will be totally different if you changed anything in the file and hashed it again. It's used when de-securing the file to check that the file hasn't changed since it was sent-this provides message integrity.
2. Encrypts this digest using Alice's Private Authentication key. This creates a digital signature for the file.
3. 'Signs' the file by adding the digital signature to the end of the file.



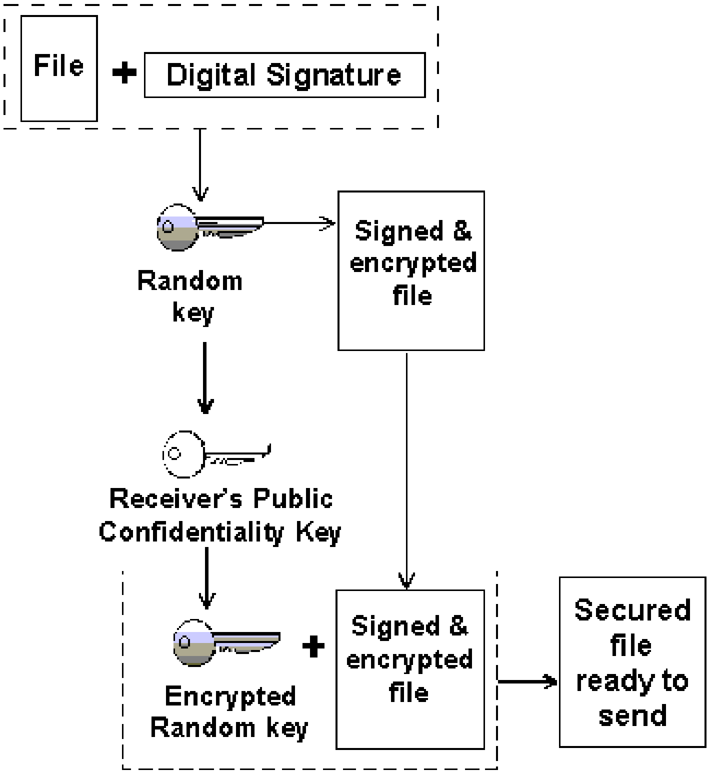
This diagram illustrates the process of creating digital signature

This process is similar to signing a paper letter, and provides authentication, message integrity, and non-repudiation of the sender.

Encrypting the file

PKI then generates a random key and uses it to encrypt the original file and the digital signature, making the message confidential. The randomly generated key is then encrypted using Bob's Public Confidentiality Key and attached to the file.

Alice can now send the signed and encrypted file to Bob as an attachment to an email message. This process is similar to sealing a letter in an envelope to make it confidential.



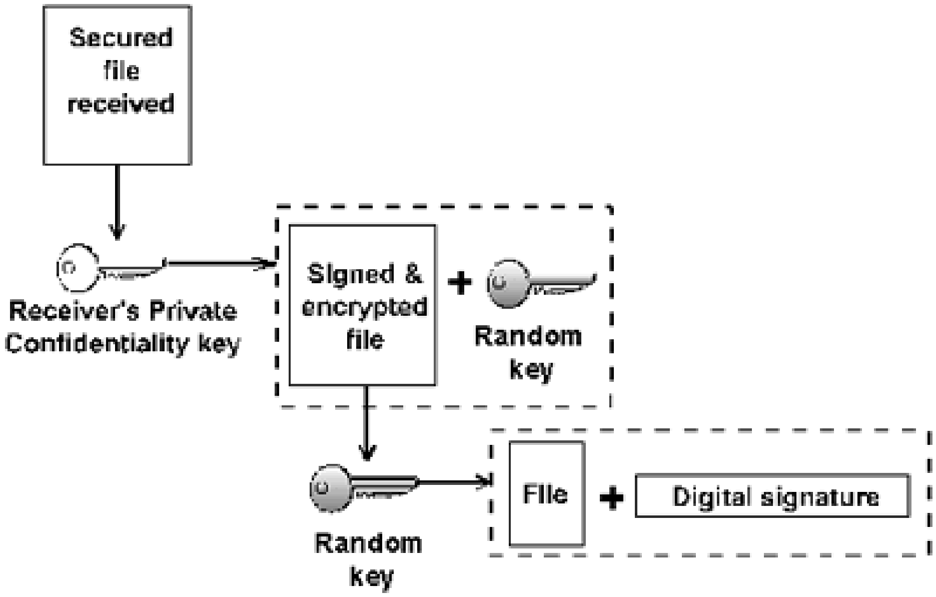
This diagram illustrates the process of encryption

What happens when you de-secure a file?

When Bob receives the file from Alice, he de-secures it and checks the signature.

Decrypting the file

When the secured file arrives, Bob decrypts the random encryption key using his own Private Confidentiality Key, and then uses the random key to decipher the file itself. This produces the original un-encrypted file, plus the digital signature. It also means that the message was confidential, because Bob is the only person who has the Private Confidentiality Key needed to decrypt the random key used to encrypt the file.

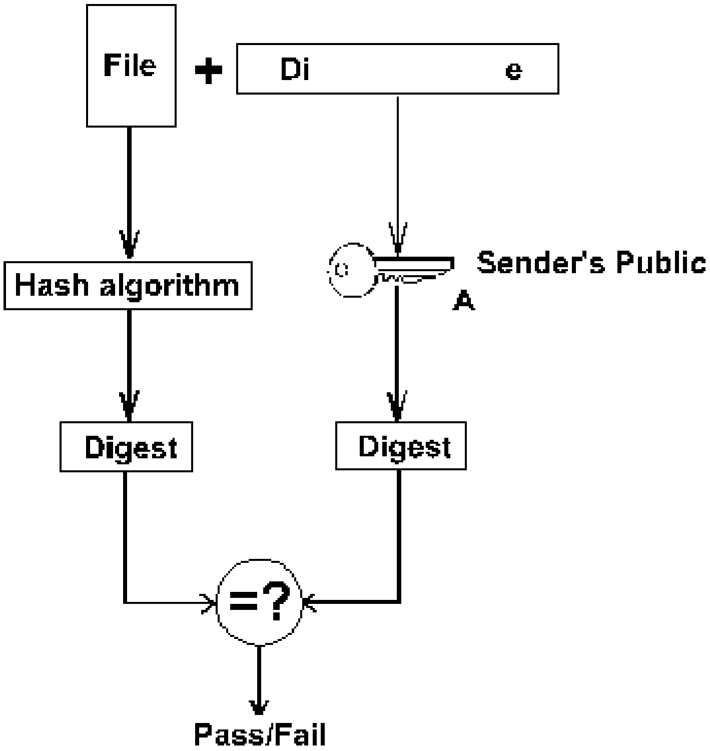


This diagram shows the decryption process

PKI then passes the message through the same hash algorithm Alice used to create the digest, and passes the digital signature through Bob's Public Authentication Key. This provides two hash results.

Checking the signature

PKI then compares the hash results. If they don't match exactly, Bob knows that the message has either changed since Alice sent it, or has been sent by an impostor. Bob therefore knows not to trust its contents.



This diagram shows process at receiver's end to check the signature after the message has been decrypted

If the two hash results match exactly, Bob knows that the message has:

- **Authenticity**, because Alice is the only person who has the Private Key that created the digital signature
- **Integrity**, because if Eve had intercepted the message and made any changes to its contents, the hash results would not match
- **Non-repudiation**, because Alice is the only person who has the Private Key used to create the digital signature, so she can't claim that someone else sent the message.

Note that, to send a secure message using a Public Key system, Alice must have Bob's Public Confidentiality Key so she can encrypt the file, and Bob must have Alice's Public Authentication Key so he can check her digital signature.

PUBLIC KEY AND CERTIFICATION AUTHORITIES

The Public Key system makes it possible for two parties to communicate securely without either having to know or trust the other party. This is possible because a third party that both the other parties' trust identifies them, and certifies that their keys are genuine.

Certification Authority

This third party is called the Certification Authority, or CA. By placing their trust in the CA, the other parties don't need to trust each other because the CA guarantees that they are who they claim to be. The CA does this by registering each user's identification information, and issuing them with a set of Private keys and a set of Public Key Certificates.

Public Key Certificate

A Public Key Certificate is a file that contains some of the user's identifying information and their Public Key. The CA signs this certificate with its own Private Authentication Key to verify that the information it contains is correct. Each user has a copy of the CA's Public Authentication Key, and if they want to check the authenticity of someone else's certificate, they can check the CA's signature on the certificate.

PKI does this check automatically whenever you import a certificate. This way you can always be sure that the person with whom you are communicating using PKI really is who they claim to be.

Each user thus has the following keys:

- **Private Authentication** key for generating digital signatures
- **Public Authentication** key that other people use to verify that user's digital signature
- **Public Confidentiality** key that other people use to encrypt files they send to the user
- **Private Confidentiality** key used to decrypt messages sent to them that have been encrypted with their Public Confidentiality key.

About Author:

Deepak Pareek, FINATECH INDIA

Deepak Pareek is a seasoned Financial Technology Expert. He has worked with a wide range of organisations. Deepak can be consulted for your next IT project. Contact for additional details.