Public Key Encryption

Revolutionary advance in encryption

Based on mathematical functions rather than on simple operations on bit patterns

Asymmetric, involving the use of two separate keys. The use of two keys has profound consequences in the areas of confidentiality, key distribution, and authentication

Common misconceptions

Public-key encryption is more secure from cryptanalysis than symmetric encryption.

WRONG!

The security of any encryption scheme depends on:

- The length of the key and
- The computational work involved in breaking a cipher

RIGHT!

Common misconceptions

It has made symmetric encryption obsolete.

WRONG!

On the contrary, because of the computational overhead of current public-key encryption schemes, there seems no foreseeable likelihood that symmetric encryption will be abandoned.

RIGHT!

Common misconceptions

There is a feeling that key distribution is trivial when using public-key encryption, compared to the rather cumbersome handshaking involved with key distribution centers for symmetric encryption.

WRONG!

For public-key key distribution, some form of protocol is needed, often involving a central agent, and the procedures involved are no simpler or any more efficient than those required for symmetric encryption.

RIGHT!

Ingredients of a public-key encryption scheme

Plaintext: Readable message or data that is fed into the algorithm as input

Encryption algorithm: Performs various transformations on the plaintext

Public and private key: Pair of keys that have been selected so that if one is used for encryption, the other is used for decryption. The exact transformations performed by the encryption algorithm depend on the public or private key that is provided as input

Ciphertext: The scrambled message produced as output. It depends on the plaintext and the key

Decryption algorithm: This algorithm accepts the ciphertext and the matching key and produces the original plaintext

Essential steps of PKE

1. Each user generates a pair of keys to be used for the encryption and decryption of messages.
2. Each user places one of the two keys in a public register or other accessible file. This is the public key. The companion key is kept private.
3. If Bob wishes to send a private message to Alice, Bob encrypts the message using Alice's public key.
4. When Alice receives the message, she decrypts it using her private key. No other recipient can decrypt the message because only Alice knows Alice's private key.

Essential steps of PKE

CONFIDENTIALITY

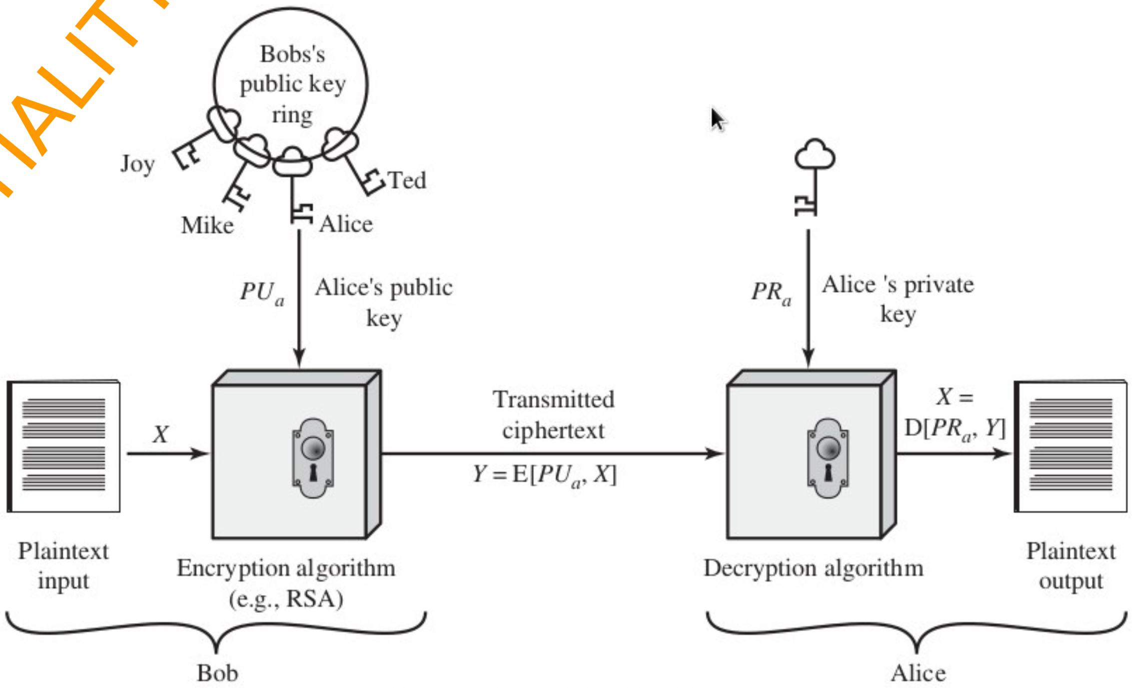


Image extracted from Stallings, W. (2018). Network Security Principles and Practices. *Prentice Hall*. page 69.

Essential steps of PKE

AUTHENTICATION
DATA INTEGRITY

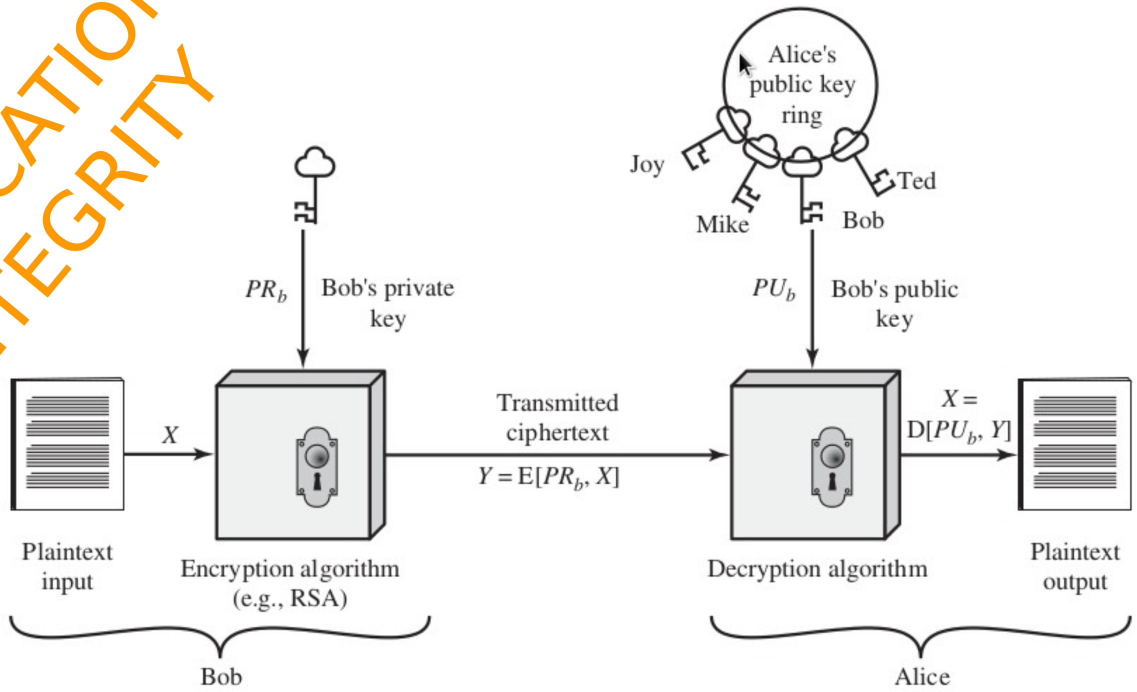


Image extracted from Stallings, W. (2018). Network Security Principles and Practices. *Prentice Hall*. page 69.

Asymmetric encryption algorithms

RSA:

- One of the first public-key schemes. It has since reigned supreme as the most widely accepted and implemented approach to public-key encryption.
- Currently, a 1024-bit key size (about 300 decimal digits) is **not** considered strong enough since 2010.
- 2048-bit key size is recommended and predicted as enough until 2030 by NIST.
- The vast majority of the products and standards that use public-key cryptography for encryption and digital signatures use RSA

Diffie-Hellman Key Agreement:

- A number of commercial products employ this key exchange technique (la red Tor)
- The algorithm itself is limited to the exchange of the keys.

Digital Signature Standard (DSS):

- Published by the NIST.
- The DSS makes use of SHA-1 and presents a new digital signature technique, the Digital Signature Algorithm (DSA).

Elliptic Curve Cryptography (ECC):

- The bit length for secure RSA use has increased over recent years, and this has put a heavier processing load on applications using RSA.
- For electronic commerce sites that conduct large numbers of secure this burden is heavy.
- The principal attraction of ECC compared to RSA is that it appears to offer equal security for a far smaller bit size, thereby reducing processing overhead.

Applications for Public-Key Cryptosystems

Algorithm	Digital Signature	Symmetric Key Distribution	Encryption of Secret Keys
RSA	Yes	Yes	Yes
Diffie-Hellman	No	Yes	No
DSS	Yes	No	No
Elliptic Curve	Yes	Yes	Yes

Requirements for Public-Key Cryptography

1. It is computationally easy for a party B to generate a pair (public key PU_b, private key PR_b)
2. It is computationally easy for a sender A, knowing the public key and the message to be encrypted, M, to generate the corresponding ciphertext:

$$C = E(PU_b, M)$$

3. It is computationally easy for the receiver B to decrypt the resulting ciphertext using the private key to recover the original message:

$$M = D(PR_b, C) = D[PR_b, E(PU_b, M)]$$

4. It is computationally infeasible for an opponent, knowing the public key, PU_b, to determine the private key, PR_b
5. It is computationally infeasible for an opponent, knowing the public key, PU_b, and a ciphertext, C, to recover the original message, M.
6. Either of the two related keys can be used for encryption, with the other used for decryption.

$$M = D[PU_b, E(PR_b, M)] = D[PR_b, E(PU_b, M)]$$

GnuPG

GnuPG is a complete and free implementation of the OpenPGP standard as defined by RFC4880 (also known as PGP)

GnuPG allows:

- Encrypt and sign your data and communication
- Features a versatile key management system
- GnuPG, also known as GPG, is a command line tool with features for easy integration with other applications

Asymmetric encryption with GPG

Key generation:

```
mjsantof@Hopper:~/cifrado$ gpg --gen-key
gpg (GnuPG) 2.2.5; Copyright (C) 2018 Free Software Foundation, Inc.
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.


Note: Use "gpg --full-generate-key" for a full featured key generation d


GnuPG debe construir un ID de usuario para identificar su clave.


Nombre y apellidos:
```

Asymmetric encryption with GPG

To generate enough entropy:

```
mjsantof@Hopper:~/cifrado$ apt-get install rng-tools
mjsantof@Hopper:~/cifrado$ gpg --gen--key &
mjsantof@Hopper:~/cifrado$ sudo rngd -r /dev/urandom
```

Entropy is the way how we measure the randomness in the sequence of bits that comprise the key

Asymmetric encryption with GPG

Listing keys:

```
$ gpg -k
/home/mjsantof/.gnupg/pubring.gpg
---------------------------------
pub   2048R/8BC6A1DC 2014-10-06
uid   María José Santofimia Romero (Generacion de claves para el ejercicio de clase)
<MariaJose.Santofimia@uclm.es>
sub   2048R/8188FA80 2014-10-06
```

main and
subkey,
both RSA

Asymmetric encryption with GPG

Listing private keys:

```
$ gpg --list-secret-keys
/home/mjsantof/.gnupg/pubring.gpg
---------------------------------
sec   rsa2048 2017-10-04 [SC] [caduca: 2019-10-04]
      FE2CFDE7A28F124234AD3F4B26A1A839A2C06689
uid           [  absoluta ] Maria Jose Santofimia <mariajose.santofimia@gmail.com>
ssb   rsa2048 2017-10-04 [E] [caduca: 2019-10-04]
```

Asymmetric encryption with GPG

Export and send the public key:

```
$ gpg --output Kpub.gpg --export 8BC6A1DC
$ gpg --armor --export mariajose.santofimia@uclm.es > mykey.asc
$ gpg --export-secret-key -a "María José Santofimia Romero" > Kpriv.key
```

Upload a public key to key server:

```
$ gpg --send-keys --keyserver pgp.rediris.es 8BC6A1DC
gpg: enviando clave 8BC6A1DC a hkp servidor pgp.rediris.es
```

Import public and private keys from file or key server:

```
$ gpg --import Kpub.gpg
$ gpg --keyserver pgp.rediris.es --recv-keys 1B1A1428
$ gpg --allow-secret-key-import --import Kpriv.key
```

Using PGP keys

Confidentiality:

Encrypting using the recipient public key

```
$ gpg --encrypt --armor --recipient
8BC6A1DC lesson2-slides.pdf
```

Decryption using the recipient private key

```
$ gpg -d lesson2-slides.pdf.asc >
lesson2-slides.pdf
```

Using PGP keys

Authentication (Digital Signature):

My private key

Sign:

```
$ gpg -u 8188FA80 --output lesson2-slides.pdf.gpg --sign lesson2-slides.pdf
```

Recipient's public key

Encrypt:

```
gpg --encrypt --armor --recipient 8BC6A1DC lesson2-slides.pdf.gpg
```

Verify and decypher signed files:

```
$ gpg -d lesson2-slides.pdf.gpg.asc > lesson2-slides.pdf.gpg

$ gpg --verify lesson2-slides.pdf.gpg

gpg: Firmado el lun 06 oct 2014 23:41:21 CEST usando clave RSA ID 8BC6A1DC
gpg: Firma correcta de "María José Santofimia Romero (Generacion de claves para el ejercicio de clase) <MariaJose.Santofimia@uclm.es>"

$ gpg --output /tmp/lesson2.pdf --decrypt /tmp/lesson2.pdf.gpg
```

Ejercicio

1. Crea un par de claves (pública y privada) y ponla pública en el servidor de claves de Red IRIS

    Usa tu correo de la UCLM y tu nombre y apellidos

2. Escribe un mensaje en un archivo de texto y garantiza la integridad y confidencialidad del mensaje

    Combina el uso de claves pública y privada

3. Envíame por correo el resultado

    Recuerda que utilizaré tu correo electrónico para buscar tu clave pública en Red IRIS

Asymmetric Encryption Algorithms

RSA

The most widely accepted and implemented approach to public-key encryption.

RSA is a block cipher in which the plaintext and ciphertext are integers between 0 and n – 1 for some n.

Currently, a 2048-bit key size is considered strong enough for virtually all applications.

Asymmetric Encryption Algorithms

DIFFIE-HELLMAN KEY AGREEMENT

The purpose of the algorithm is to enable two users (no previous contact nor authentication required) to securely reach agreement (using a non-secure channel) about a shared secret that can be used as a secret key for subsequent symmetric encryption of messages.

The algorithm itself is limited to the exchange of the keys.

Tor network and ssh use this algorithm.

Asymmetric Encryption Algorithms

Verify which key exchange algorithms are supported by ssh

```
$ ssh -Q kex
diffie-hellman-group1-sha1
diffie-hellman-group14-sha1
diffie-hellman-group14-sha256
diffie-hellman-group16-sha512
diffie-hellman-group18-sha512
diffie-hellman-group-exchange-sha1
diffie-hellman-group-exchange-sha256
ecdh-sha2-nistp256
ecdh-sha2-nistp384
ecdh-sha2-nistp521
curve25519-sha256
```

Asymmetric Encryption Algorithms

DIGITAL SIGNATURE STANDARD (DSS)

DSS uses an algorithm that is designed to provide only the digital signature function.

Unlike RSA, it cannot be used for encryption or key exchange.

Three algorithms are considered under the standard: DSA, RSA, ECDSA

Asymmetric Encryption Algorithms

ELLIPTIC CURVE CRYPTOGRAPHY (ECC)

The principal attraction of ECC compared to RSA is that it appears to offer equal security for a far smaller bit size, thereby reducing processing overhead.

On the other hand, although the theory of ECC has been around for some time, it is only recently that products have begun to appear and that there has been sustained cryptanalytic interest in probing for weaknesses. Thus, the confidence level in ECC is not yet as high as that in RSA.

Not very common for web applications but famous for being used for Bitcoins (hash of an ECDSA public key)

SSH

Create a SSH key pair with ssh-keygen.

Add public key to .ssh/authorized_keys in the remote host.

Disable root login.

Disable password-based login.

Digital Signature and Key Management

Aspects related to the use of public-key encryption:

- The secure distribution of public keys
- The use of public-key encryption to distribute secret keys
- The use of public-key encryption to create temporary keys for message encryption

Digital Signature

Suppose that Bob wants to send a message to Alice:

Bob uses a secure hash function, such as SHA-512, to generate a hash value for the message and then encrypts the hash code with his private key, creating a digital signature.

Bob sends the message with the signature attached.

When Alice receives the message plus signature, she:

1. calculates a hash value for the message;
2. decrypts the signature using Bob's public key; and
3. compares the calculated hash value to the decrypted hash value.

If the two hash values match, Alice is assured that the message must have been signed by Bob

Digital Signature

```
$ gpg --print-md sha512 msg.txt
msg.txt: DBA0F4D0 9891206E 5100791C D857D9E0 8A6DF6F4 F5A539D9 F28B555F
F4E7EE43
         CD6F80EB 03117535 990EC78F 8C42AE7B 5C49A9F0 A25FB964 FE56E183
03E6D107

$ gpg -u 1E2676949711645 50A34C35F30EC347912B80243 --output msg_signed --sign
msg.txt
$ gpg --output msg_original.txt --decrypt msg_signed
```

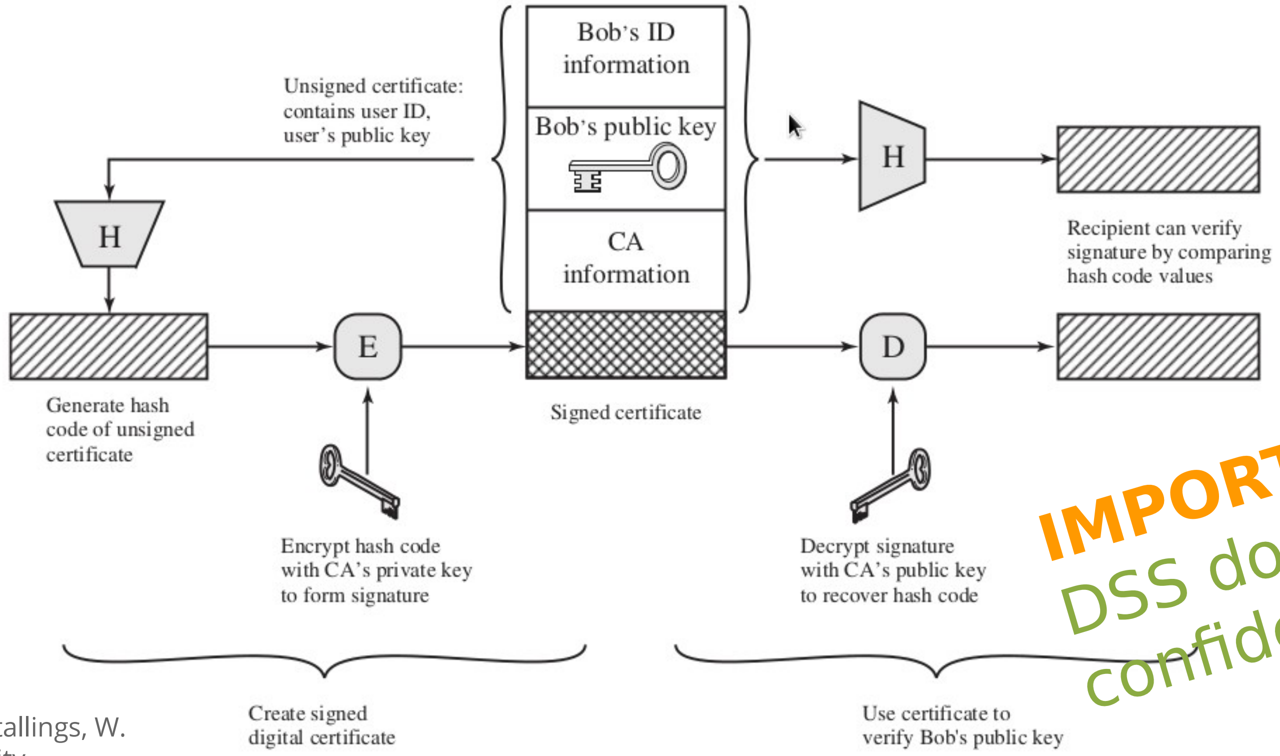
My private key

Public-Key Certificates

The public key is public and can be broadcasted

- Weakness: Anyone can forge such a public announcement
- Solution: The public-key certificate: public key + user ID
- The certificate also includes some information about the third party plus an indication of the period of validity of the certificate
- A user can present her public key to the authority in a secure manner and obtain a signed certificate. The user can then publish the certificate.
- Anyone needing this user's public key can obtain the certificate and verify that it is valid by means of the attached trusted signature

Public-Key Certificates



IMPORTANT: DSS does not sup confidentiality

Image extracted from Stallings, W. (2014). Computer Security Principles and Practices. *Prentice Hall*. page60.

Verificate Certificate


Solicitar verificación - Sede - Mozilla Firefox

Solicitar verificación - ...

https://www.sede.fnmt.gob.es/certificados/persona-fisica/verificar-estado/solicitar-verificacion

Buscar

Sede Electrónica
Real Casa de la Moneda
Fábrica Nacional
de Moneda y Timbre

Certificados | Trámites

Inicio > Certificados > Persona Física > Verificar estado > Solicitar verificación

Persona Física
Obtener Certificado Software
Obtener Certificado con Android
Obtener Certificado con DNIe
Verificar estado
Solicitar verificación
Renovar
Anular
Certificado de Representante
Administración Pública
Certificados de componente
Soporte Técnico

Solicitar verificación

funcionando correctamente.

Con su certificado podrá acreditarse ante los servicios ofrecidos por las entidades que admitan el uso de los certificados digitales emitidos por la Fábrica Nacional de Moneda y Timbre.

Le rogamos no obstante que verifique la exactitud de los datos que le mostramos a continuación y que su nombre, apellidos y NIF coinciden exactamente con su DNI. En caso de ser incorrecto alguno de estos datos deberá revocar su certificado actual y solicitar uno nuevo aquí

Información sobre la identidad (valores personales)

Identificador	Valor
Nombre	MARIA JOSE
Primer apellido	SANTOFIMIA
Segundo apellido	ROMERO
NIF	05920360E
Dirección de correo electrónico	mariajose.santofimia@uclm.es

Información sobre las claves (valores técnicos)

Identificador	Valor
Número de serie del certificado	5D7827A96B5C4181559E5B68C04AB8A4
Autoridad emisora	CN=AC FNMT Usuarios, OU=Ceres, O=FNMT-RCM, C=ES
Propietario	CN=SANTOFIMIA ROMERO MARIA JOSE - 05920360E, GIVENNAME=MARIA JOSE, SURNAME=SANTOFIMIA ROMERO,

Fecha y Hora Oficial
17/10/2018
06:33:06

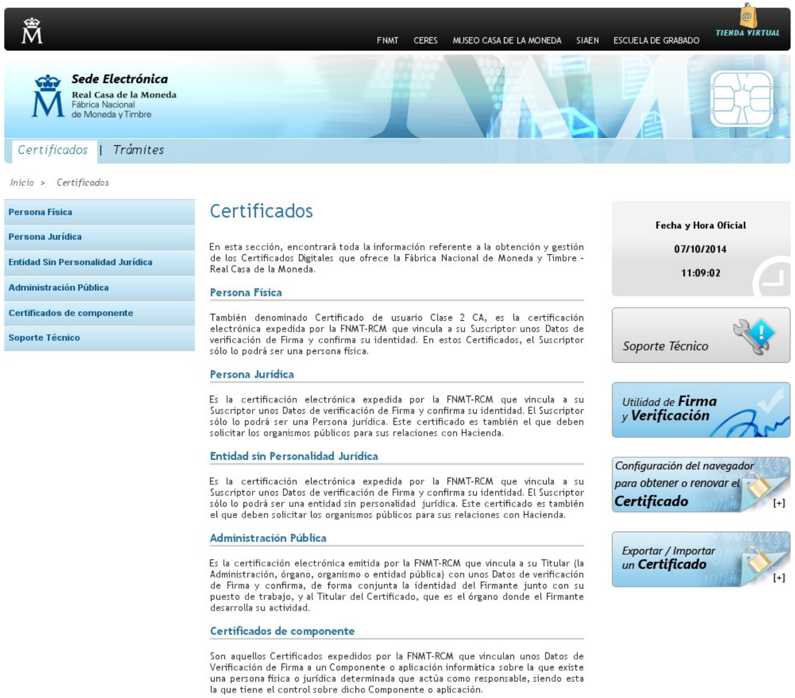
Información Destacada

Configuración del navegador para obtener o renovar el Certificado [+]

Exportar / Importar un Certificado [+]

Atención a Usuarios

Spanish Certificate Authority



Symmetric Key Exchange Using Public-Key Encryption

Symmetric encryption requires the two parties to share the secret key. One approach is the use of Diffie-Hellman key exchange.

Drawback:

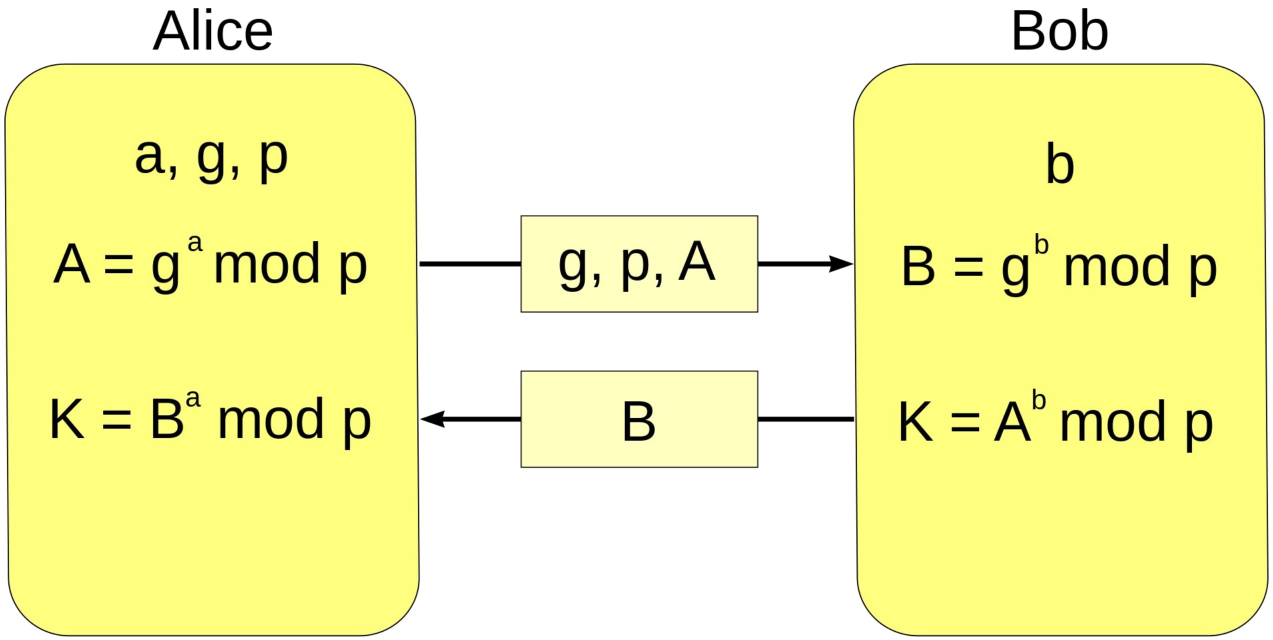
In its simplest form, Diffie-Hellman provides no authentication of the two communicating partners

Solutions:

There are variations to Diffie-Hellman that overcome this problem

There are protocols using other public-key algorithms that achieve the same objective

Symmetric Key Exchange Using Public-Key Encryption



$$K = A^b \bmod p = (g^a \bmod p)^b \bmod p = g^{ab} \bmod p = (g^b \bmod p)^a \bmod p = B^a \bmod p$$

Image extracted from Wikipedia

Digital Envelopes

This can be used to protect a message without needing to first arrange for sender and receiver to have the same secret key

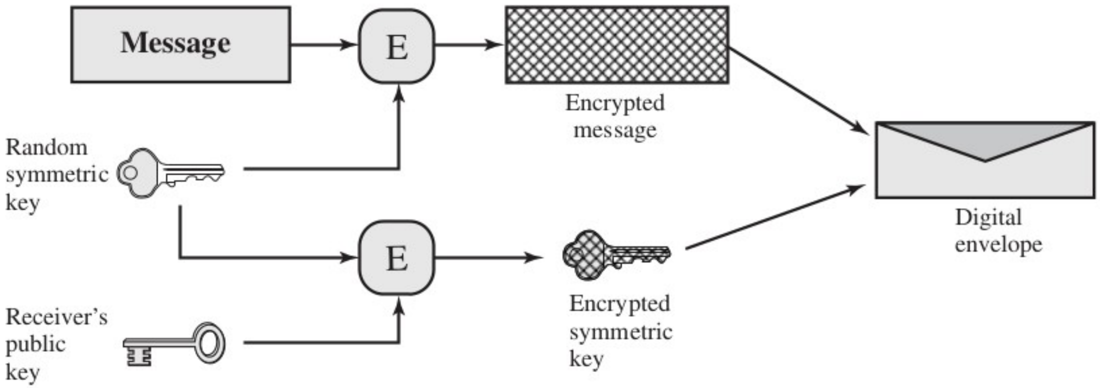


Image extracted from Stallings, W. (2018). Computer Security Principles and Practices. *Prentice Hall*. Page 76.

Digital Envelopes

Only Alice is capable of decrypting the one-time key and therefore of recovering the original message. If Bob obtains Alice's public key by means of Alice's public-key certificate, then Bob is assured that it is a valid key.

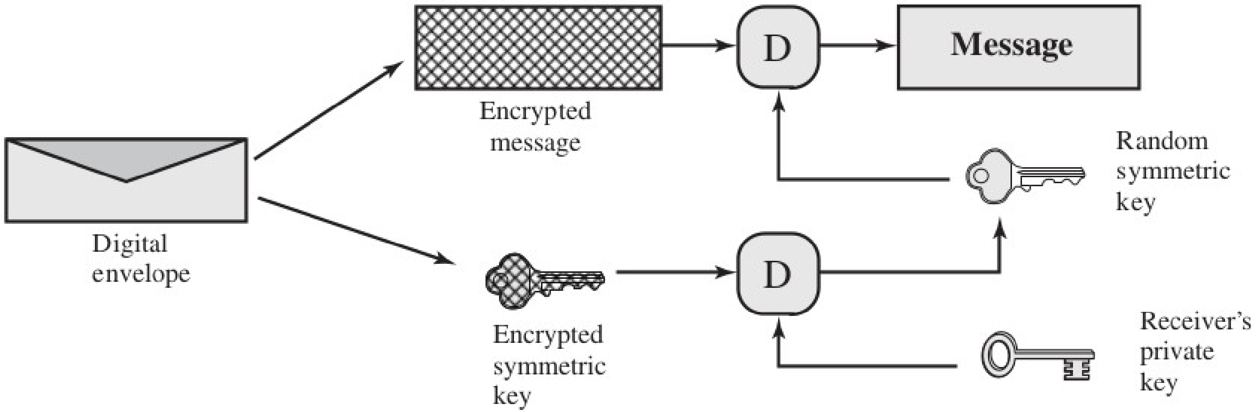


Image extracted from Stallings, W. (2018). Computer Security Principles and Practices. *Prentice Hall*. Page 76.

Digital Envelopes

1. Prepare a message.
2. Generate a random symmetric key that will be used this one time only.
3. Encrypt that message using symmetric encryption the one-time key.
4. Encrypt the one-time key using public-key encryption with Alice's public key.
5. Attach the encrypted one-time key to the encrypted message and send it to Alice.

La firma electrónica en España

Ley 59/2003, de 19 de diciembre, de firma electrónica

Firma electrónica: es el conjunto de datos en forma electrónica, consignados junto a otros o asociados con ellos, que pueden ser utilizados como medio de identificación del firmante.

Firma electrónica avanzada: es la firma electrónica que permite identificar al firmante y detectar cualquier cambio ulterior de los datos firmados, que está vinculada al firmante de manera única y a los datos a que se refiere y que ha sido creada por medios que el firmante puede mantener bajo su exclusivo control.

Firma electrónica reconocida: es la firma electrónica avanzada basada en un certificado reconocido y generada mediante un dispositivo seguro de creación de firma.

La firma electrónica reconocida tendrá respecto de los datos consignados en forma electrónica el mismo valor que la firma manuscrita en relación con los consignados en papel.

Who are Alice and Bob?

- Alice and Bob send messages each other
- Carol, Charlie, Dan, Dave are other participants
- Chuck is an opponent/enemy
- Craig is a password cracker
- Eve is an eavesdropper (passive attacker)
- Maller (active attacker o MiM)
- Oscar (white-hat hacker)
- …