Certificados 101

ReD

Defensor de la privacidad

Entusiasta de la tecnología

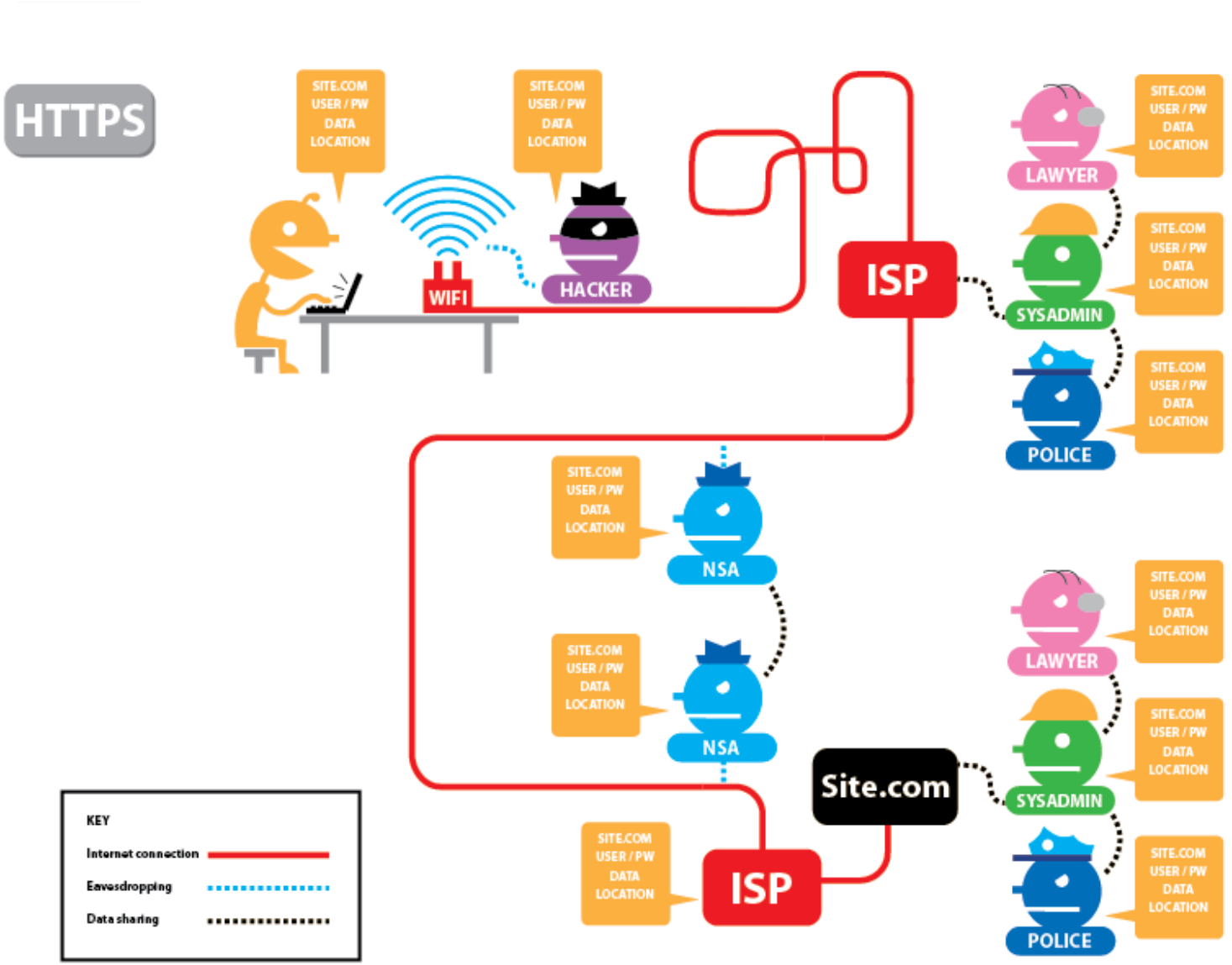
Especialista en seguridad

Cofundador de M4H

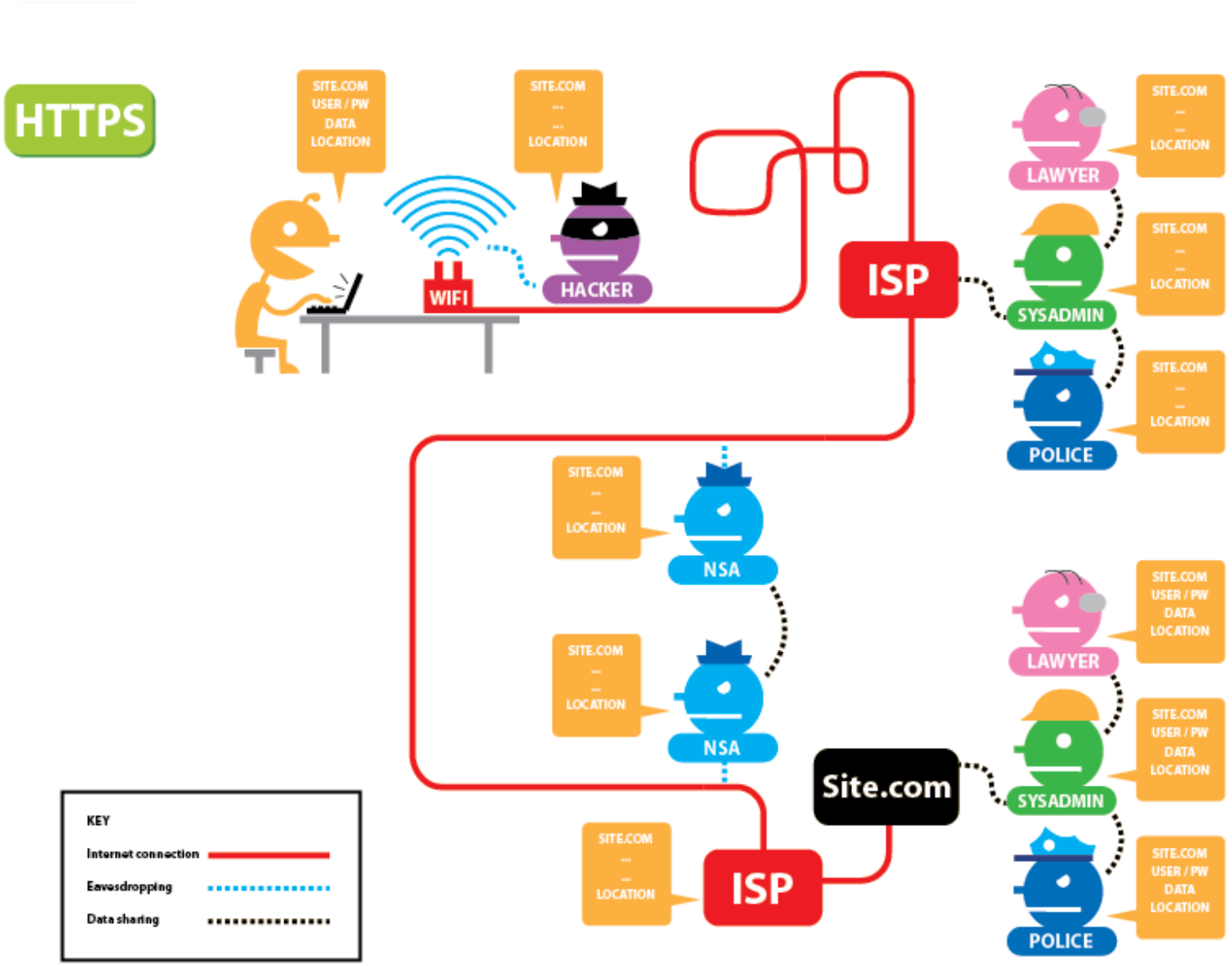
Cofundador de Hacklabs

Formador cuando me dejan

¿Por qué Cifrar? - http



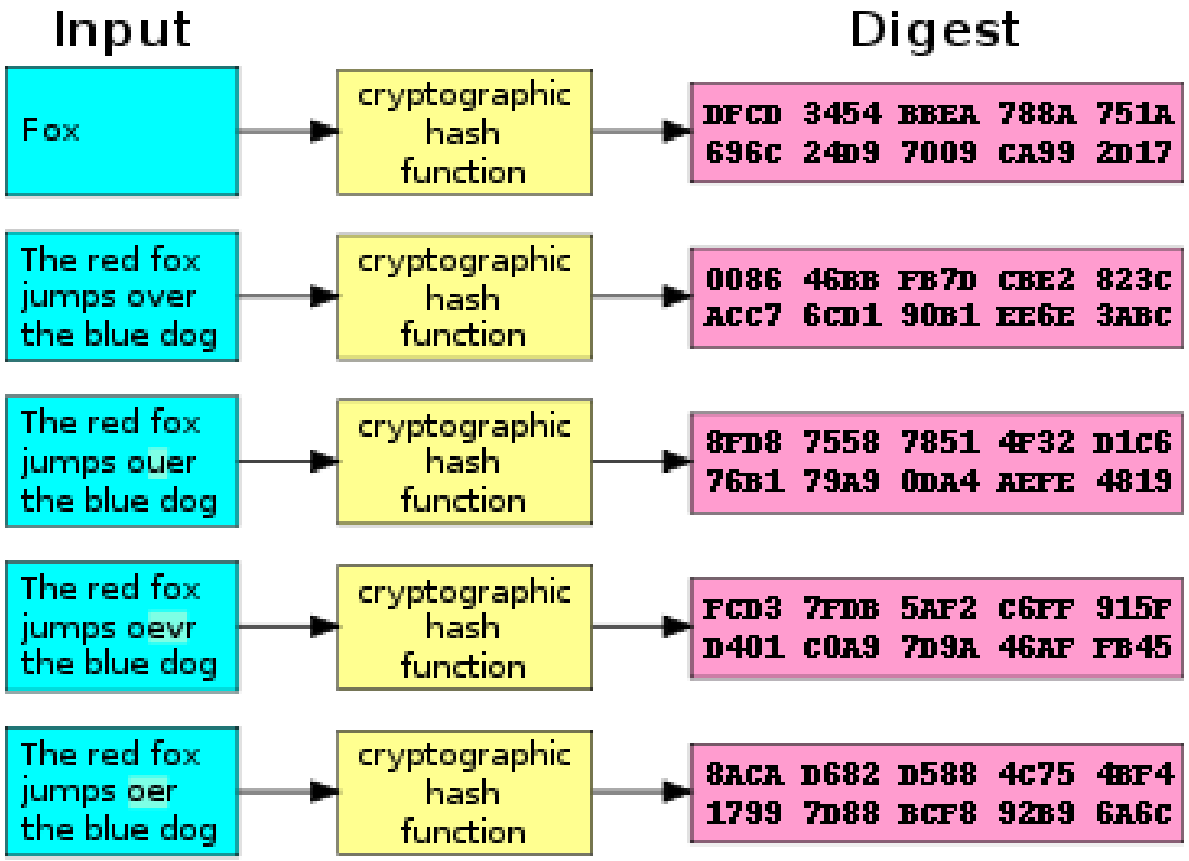
¿Por qué Cifrar? - https



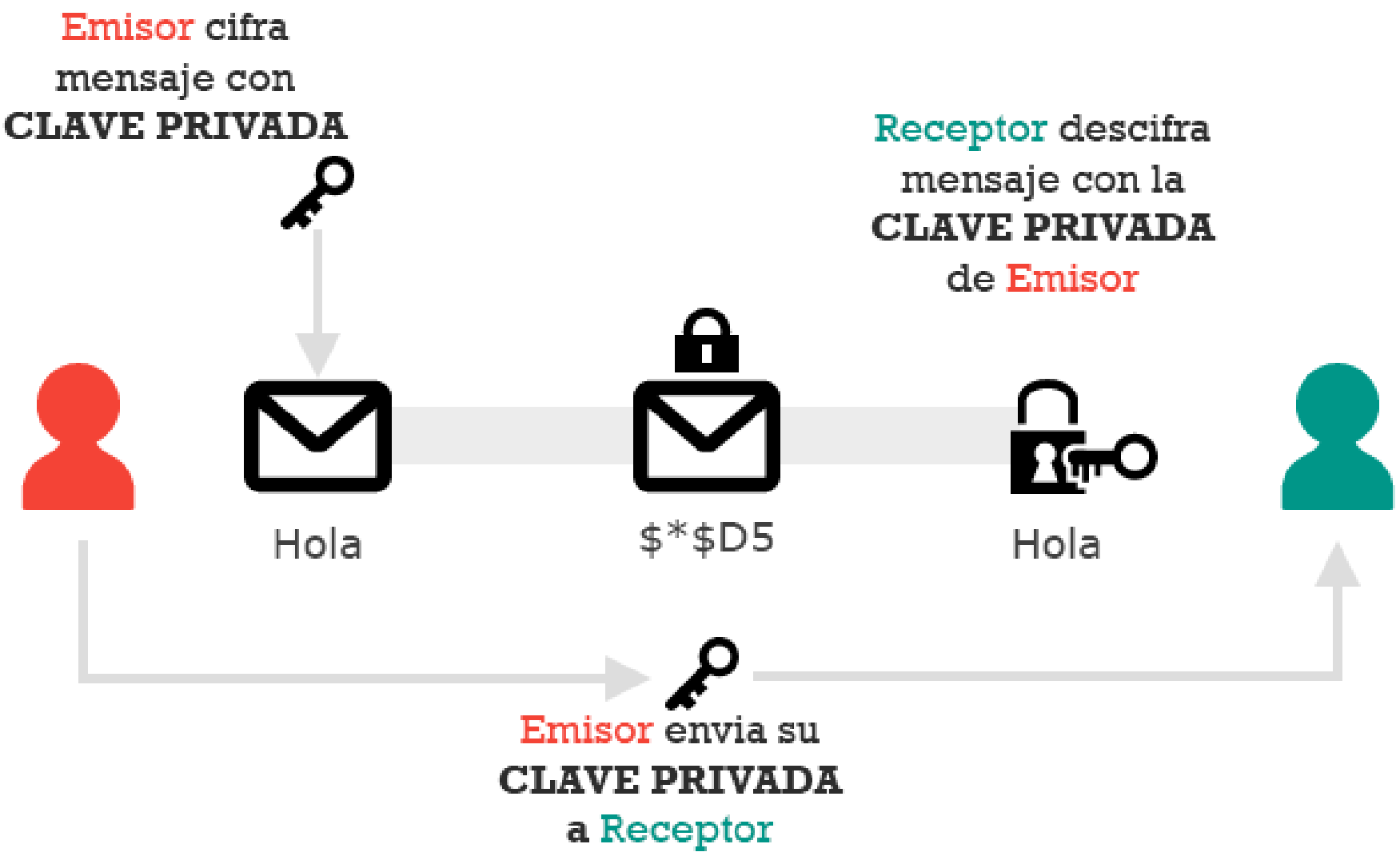
Conocimientos previos

Hash o firma

- **MD5, SHA1, SHA256...**

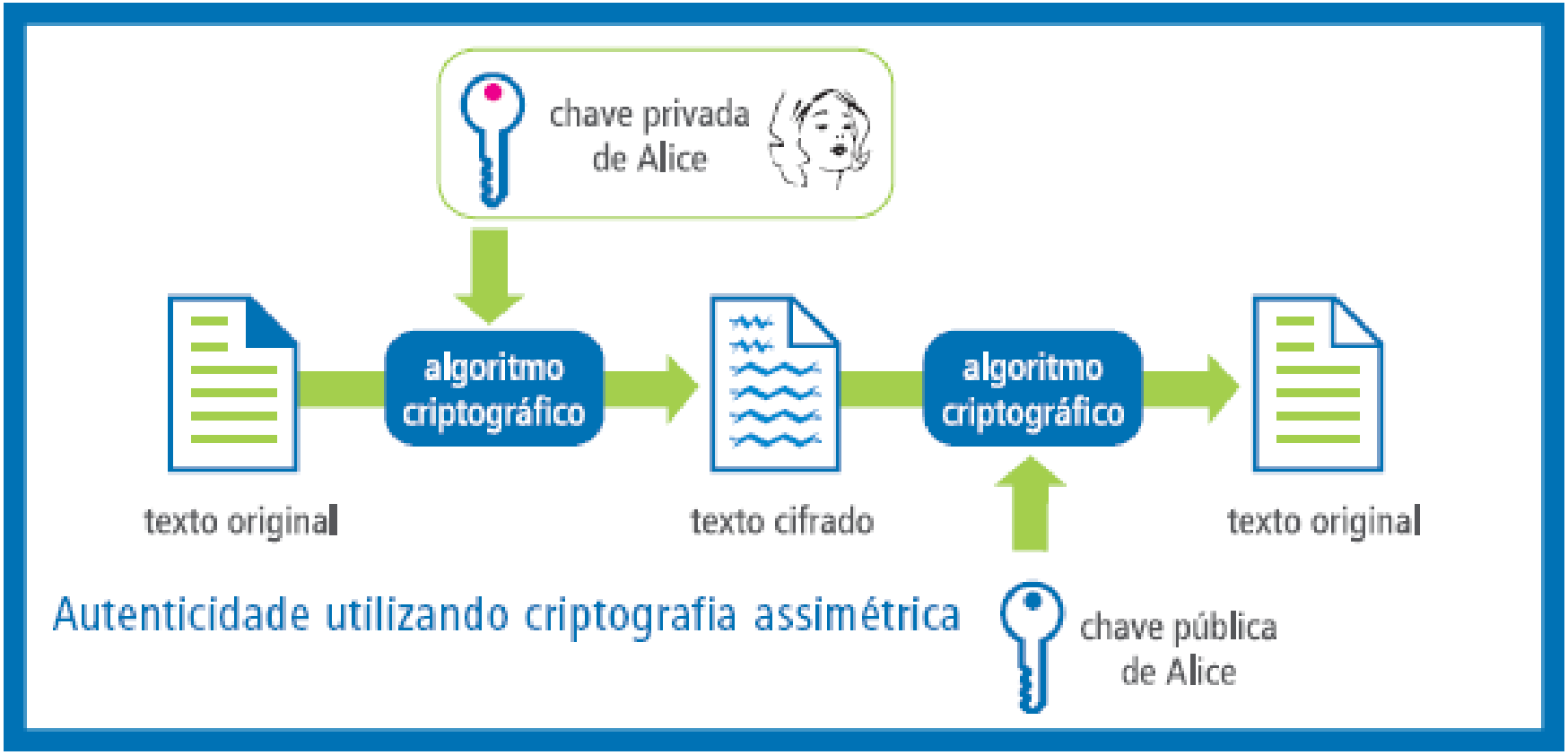


Criptografía simétrica



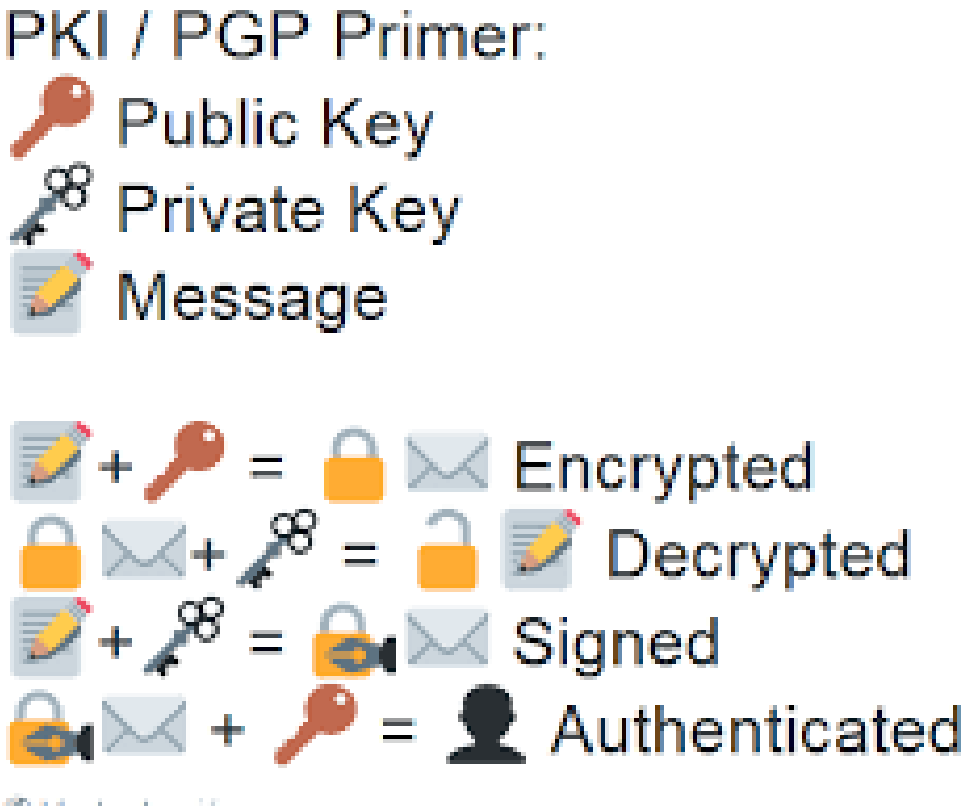
Criptografía asimétrica

- **Clave pública**
- **Clave privada**

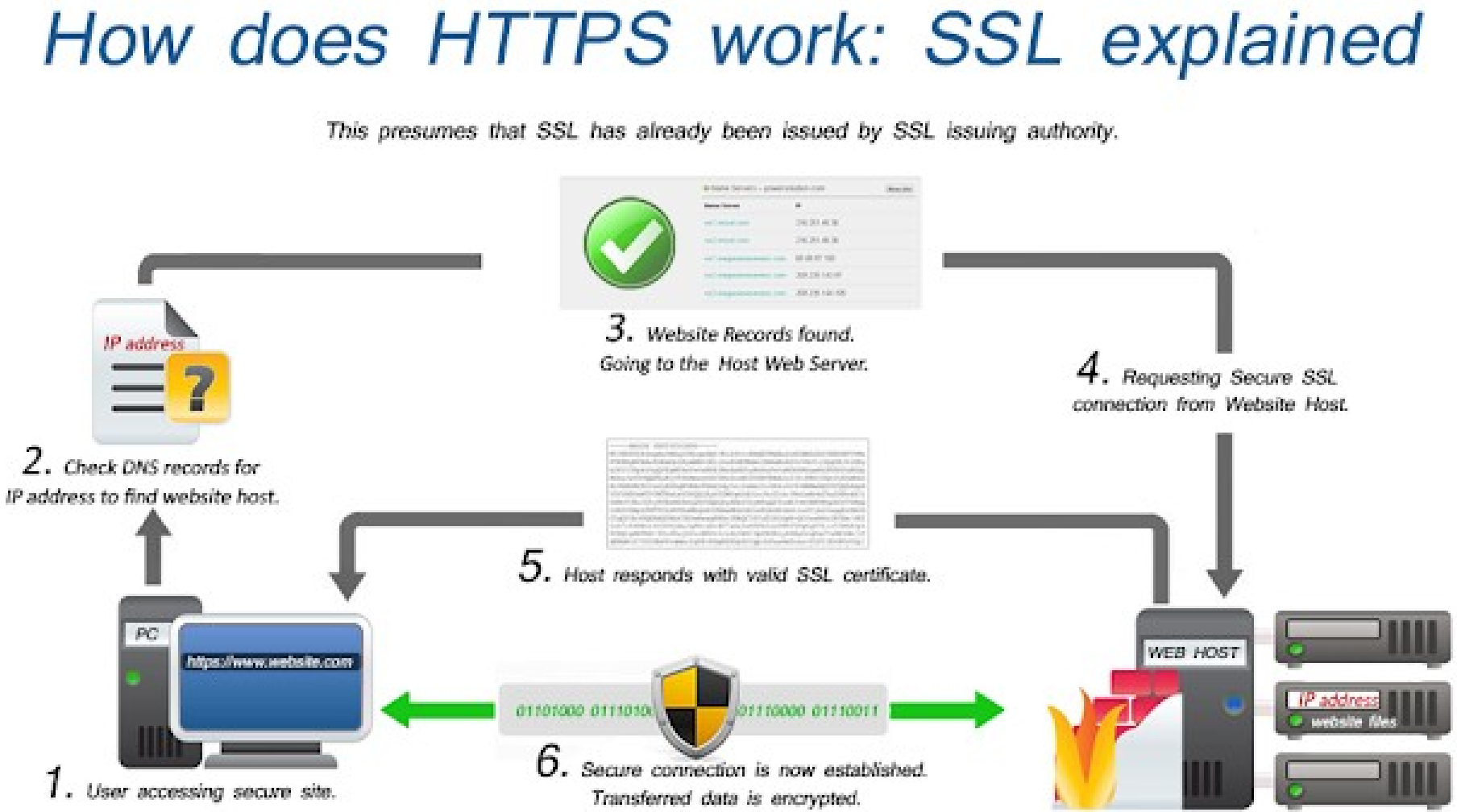


Criptografía asimétrica

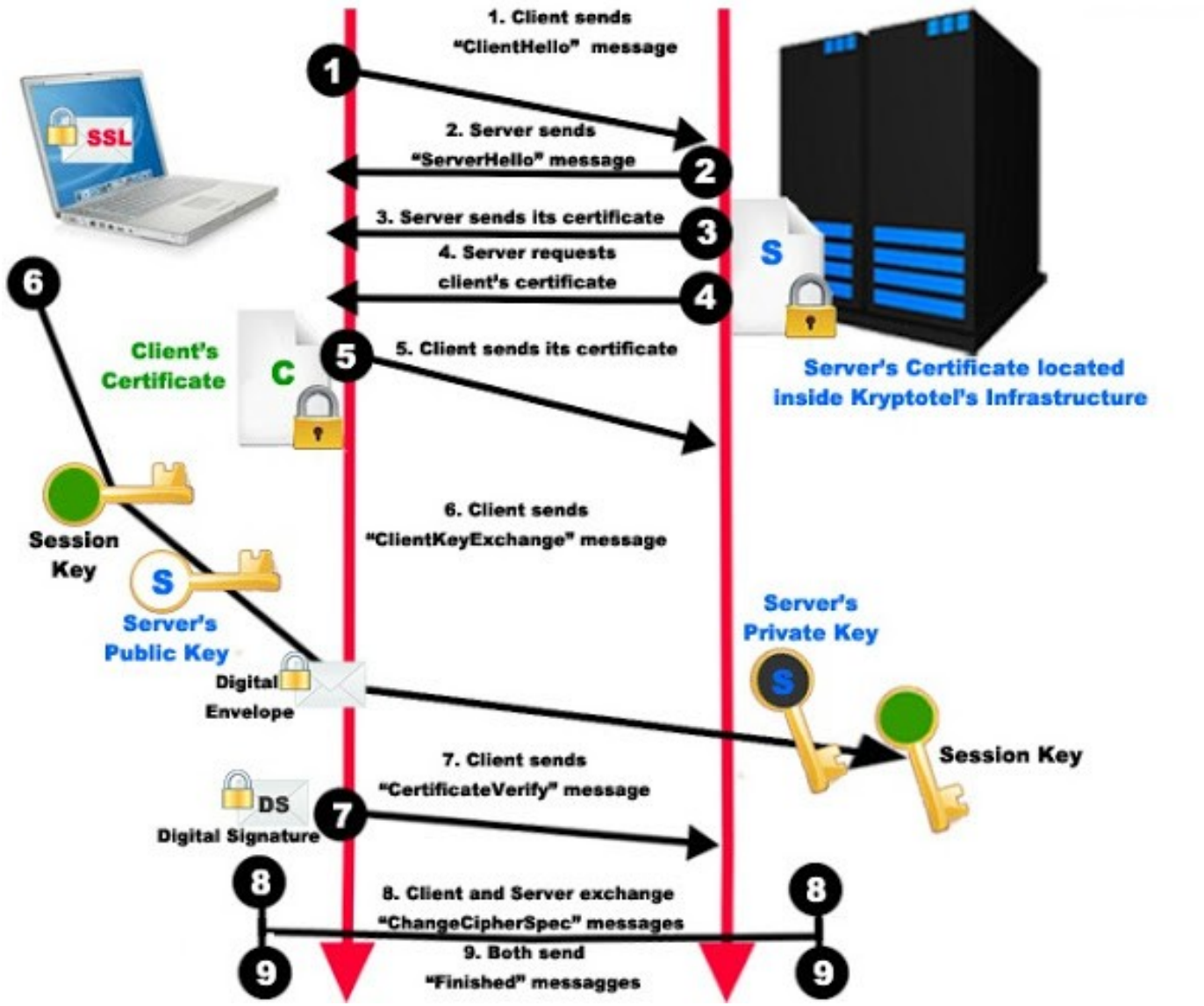
- **Clave pública**
- **Clave privada**



HTTPS


How does HTTPS work: SSL explained
This presumes that SSL has already been issued by SSL issuing authority.
3. Website Records found.
Going to the Host Web Server.
4. Requesting Secure SSL
connection from Website Host.
IP address
2. Check DNS records for
IP address to find website host.
5. Host responds with valid SSL certificate.
PC
https://www.website.com
WEB HOST
IP address
website files
01101000 0111010
01110000 0110011
1. User accessing secure site.
6. Secure connection is now established.
Transferred data is encrypted.

HTTPS

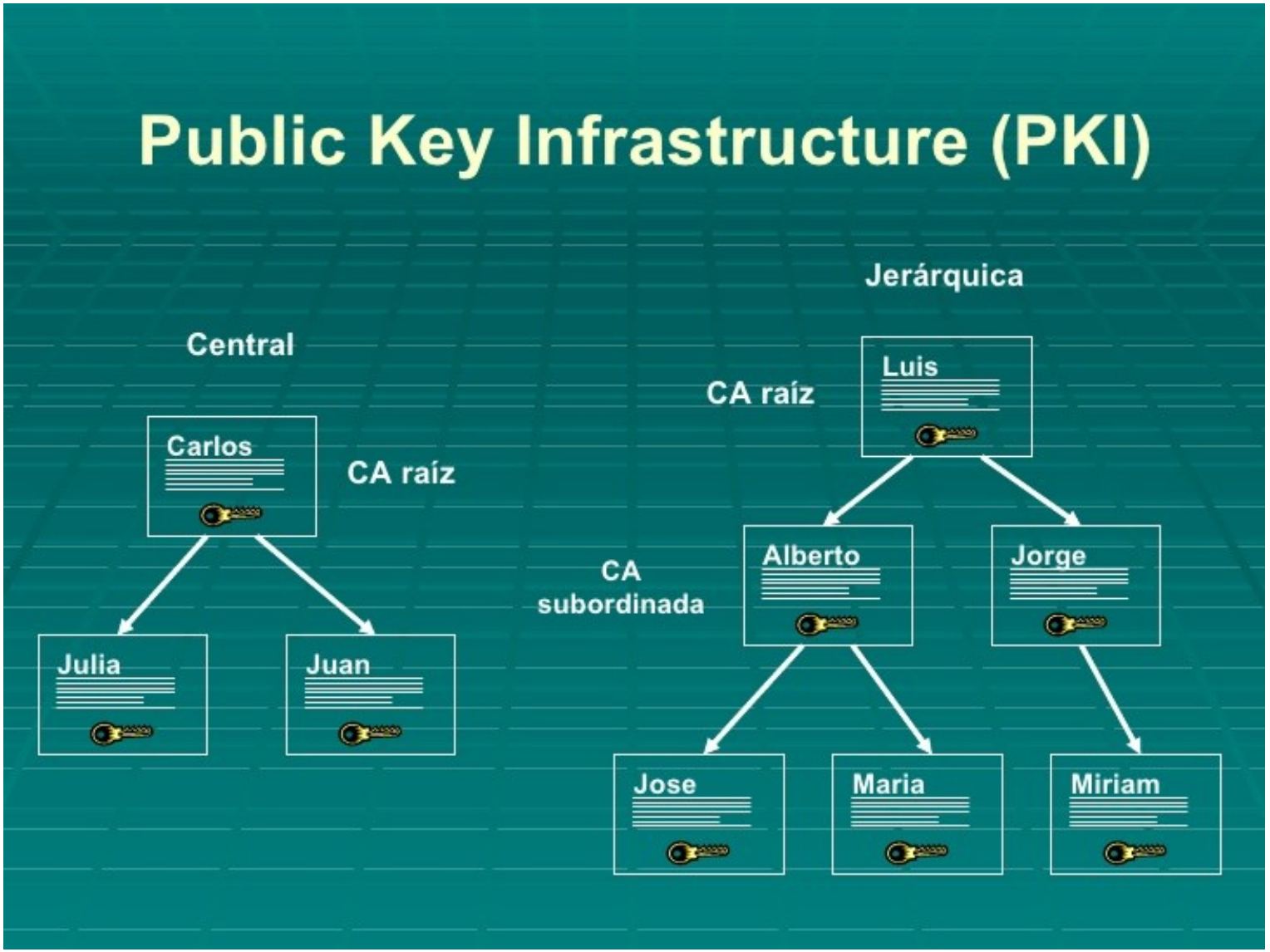

1. Client sends
"ClientHello" message
SSL
2. Server sends
"ServerHello" message
3. Server sends its certificate
4. Server requests
client's certificate
S
6
Client's
Certificate
C
5
5. Client sends its certificate
Server's Certificate located
inside Kryptotel's Infrastructure
Session
Key
6. Client sends
"ClientKeyExchange" message
S
Server's
Public Key
Server's
Private Key
Digital
Envelope
S
Session Key
7. Client sends
"CertificateVerify" message
DS
Digital Signature
7
8
8. Client and Server exchange
"ChangeCipherSpec" messages
8
9
9. Both send
"Finished" messagges
9

X509

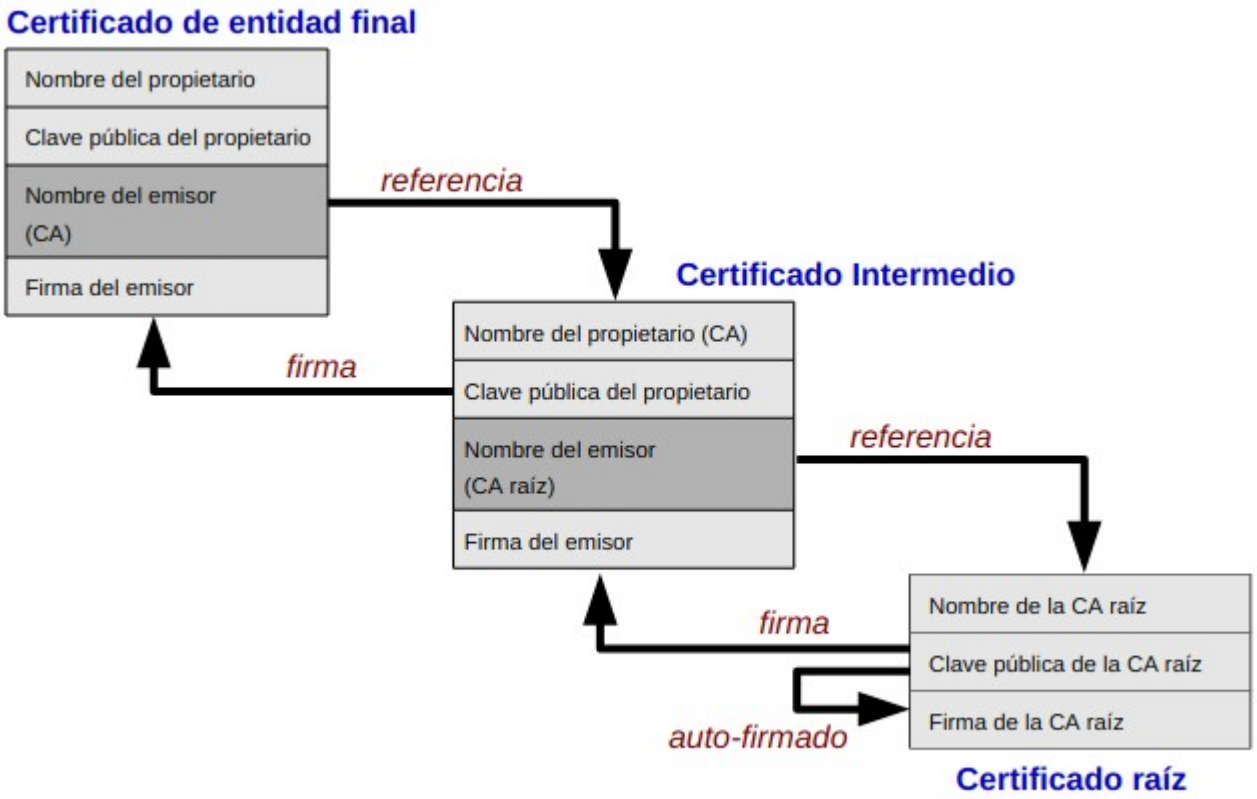
Estándar x509

- **Infraestructuras de claves públicas (PKI)**
- **Versión 3**
- **Define empleando el lenguaje ASN.1**
- **Ley 59/2003, Artículo 8 sobre la firma electrónica max cert 5 años**

Sistema CA (Autoridad certificadora)



Sistema CA (Autoridad certificadora)



Extensiones usadas

- **CSR (Certificate Signing Request)**
- **CRT (CeRT)**
- **KEY**

- **Proceso de generación de certificados KEY + CSR + Firma---> CRT**

- **Tamaño .CSR < .CRT**

Ficheros implicados

-----BEGIN CERTIFICATE REQUEST-----

-----END CERTIFICATE REQUEST-----

-----BEGIN RSA PRIVATE KEY-----

-----END RSA PRIVATE KEY-----

-----BEGIN ENCRYPTED PRIVATE KEY-----

-----END ENCRYPTED PRIVATE KEY-----

-----BEGIN CERTIFICATE-----

-----END CERTIFICATE-----

Codificación de ficheros

- **DES (Distinguish Encoding Rules)**
- **PEM (Privacy Enhanced Mail) BASE64**

Problemas de nomenclatura

- nombrecert.csr.pem
- nombrecert.key.pem
- nombrecert.crt.pem

- nombrecert.csr.des
- nombrecert.key.des
- nombrecert.crt.des

Tipos de certificado - Servidor

Identifica a un servidor ante un cliente en comunicaciones mediante SSL

Tipos de certificado - Cliente

- **Autentificación por certificado**
- **Dirigido a una persona física**

Tipos de certificado
Autofirmado

- **CN del emisor y el CN del sujeto son iguales**
- **openssl x509 -in <ficherocrt.crt.pem> -inform PEM -noout -subject -issuer**

Sistema CA (Autoridad certificadora)

Externa:

- **Incluidas en los navegadores**
- **CA raíz y CA subordinadas**

  CA → CA → CA ...

Sistema CA
(Autoridad certificadora)

Propia:

- **Importar ca ca-root.crt o ca-bundle.crt**
- **Pasos:**
  - Crear nuestra CA y su certificado
  - Crear certificado SSL firmado por CA
  - Crear los certificados de cliente (opt)
  - Lectura de los datos SSL desde nuestra app.

Campos de un certificado

Certificado

- **Versión**
- **Número de serie**
- **ID del algoritmo**
- **Emisor**
- **Validez**
- **No antes de**
- **No después de**
- **Sujeto CN (Common Name), OU (Organizational Unit), O (Organization) y C (Country).**
- **(nombres alternativos, usos permitidos para la clave, ubicación de la CRL y de la CA,etc.)**
- **Información de clave pública del sujeto**
- **Algoritmo de clave pública**
- **Clave pública del sujeto**
- **Identificador único de emisor (opcional)**
- **Identificador único de sujeto (opcional)**
- **Extensiones (opcional) serverAuth, clientAuth**
- **...**
- **Algoritmo usado para firmar el certificado**
- **Firma digital del certificado**

Campos de un certificado

Certificate

- Data:
- Version: 1 (0x0)
- Serial Number: 7829 (0x1e95)
- Signature Algorithm: md5WithRSAEncryption
- Issuer: C=ZA, ST=Western Cape, L=Cape Town, O=Thawte Consulting cc,
- OU=Certification Services Division,
- CN=Thawte Server CA/Email=server-certs@thawte.com
- Validity
- Not Before: Jul 9 16:04:02 1998 GMT
- Not After : Jul 9 16:04:02 1999 GMT
- Subject: C=US, ST=Maryland, L=Pasadena, O=Brent Baccala,
- OU=FreeSoft, CN=www.freesoft.org/Email=baccala@freesoft.org
- Subject Public Key Info:
- Public Key Algorithm: rsaEncryption
- RSA Public Key: (1024 bit)
- Modulus (1024 bit):
- 00:b4:31:98:0a:c4:bc:62:c1:88:aa:dc:b0:c8:bb:
- 33:35:19:d5:0c:64:b9:3d:41:b2:96:fc:f3:31:e1:
- 66:36:d0:8e:56:12:44:ba:75:eb:e8:1c:9c:5b:66:
- 70:33:52:14:c9:ec:4f:91:51:70:39:de:53:85:17:
- 16:94:6e:ee:f4:d5:6f:d5:ca:b3:47:5e:1b:0c:7b:
- c5:cc:2b:6b:c1:90:c3:16:31:0d:bf:7a:c7:47:77:
- 8f:a0:21:c7:4c:d0:16:65:00:c1:0f:d7:b8:80:e3:
- d2:75:6b:c1:ea:9e:5c:5c:ea:7d:c1:a1:10:bc:b8:
- e8:35:1c:9e:27:52:7e:41:8f
- Exponent: 65537 (0x10001)
- Signature Algorithm: md5WithRSAEncryption
- 93:5f:8f:5f:c5:af:bf:0a:ab:a5:6d:fb:24:5f:b6:59:5d:9d:
- 92:2e:4a:1b:8b:ac:7d:99:17:5d:cd:19:f6:ad:ef:63:2f:92:
- ab:2f:4b:cf:0a:13:90:ee:2c:0e:43:03:be:f6:ea:8e:9c:67:
- d0:a2:40:03:f7:ef:6a:15:09:79:a9:46:ed:b7:16:1b:41:72:
- 0d:19:aa:ad:dd:9a:df:ab:97:50:65:f5:5e:85:a6:ef:19:d1:
- 5a:de:9d:ea:63:cd:cb:cc:6d:5d:01:85:b5:6d:c8:f3:d9:f7:
- 8f:0e:fc:ba:1f:34:e9:96:6e:6c:cf:f2:ef:9b:bf:de:b5:22:
- 68:9f

Revocar certificados

- Hay que llevar un listado de los certificados que hemos creado (número de serie).
- El servidor tiene que tener una lista de certificados en revocación (CRL "Certificate Revocation List")
- Cuando un tercero desea comprobar la validez de un certificado, debe descargar una CRL actualizada desde los servidores de la misma autoridad de certificación que emitió el certificado en cuestión
- Ej apache. SSLCARevocationFile path/a/revocados.crl
- Alternativa OCSP (Online Certificate Status Protocol, no necesitas CRL

JKS (Java KeyStore)

- **Se gestiona con la herramienta keytool.**
  - alias: nombre de referencia al par de claves creado
  - keypass: clave de clave privada del par de claves
  - validity: es el tiempo de validez
  - storepass: clave para acceder al keystore.

- **keytool -list -storepass <claveacceso>**

Comandos útiles

Todo se realiza con openssl

- Ver cadena de comprobacion online
  - openssl s_client -connect HOST:PUERTO
    - Ej GET /index.htm HTTP/1.0
- Para sacar significado a los codigos de error
  - man verify
- Incluyendo certificados
  - openssl s_client HOST:PUERTO -showcerts

Comandos útiles

- Revisar una llave privada
  - openssl rsa -in privateKey.key -check
- Revisar un certificado
  - openssl x509 -in certificate.crt -text -noout
- Sacar propietario y emisor de certificado
  - openssl x509 -in <ficherocrt.crt.pem> -inform PEM -noout -subject -issuer
- **Comando para sacar si hereda de esa key**
  - (openssl x509 -noout -modulus -in <ficherocertificado.crt.pem> | openssl md5 ; openssl rsa -noout -modulus -in <ficherokey.key.pem> | openssl md5) | uniq

Comandos útiles

- Convertir un archivo DER (.crt .cer .der) a PEM
  - openssl x509 -inform der -in certificate.cer -out certificate.pem

- Convertir un cert + key PEM a .pfx .p12
  - openssl pkcs12 -export -out certificate.pfx -inkey privateKey.key -in certificate.crt -certfile CACert.crt

- **Check certificado SSL es válido en CA**
  - openssl verify -verbose -CAfile <ficheroautoridades.crt.pem> -purpose any <ficherocertificado.crt.pem>

Gracias

¿Preguntas?

red@nosoloaulas.com