Criptografía

Víctor Bravo, Pedro Buitrago[1]

[1]Fundación Centro Nacional de Desarrollo e Investigación en Tecnologías Libres
Nodo Mérida

CENDITEL, 2013

Licencia de Uso

Copyright (c), 2007. 2013, CENDITEL.
Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".
Una copia de la licencia puede obtenerse en los siguientes sitios en Internet:
http://www.gnu.org/copyleft/fdl.html
http://www.fsf.org/licensing/licenses/fdl.html

Agenda

1. Criptografía
   - Clases de Cifrado
2. Algoritmos Simétricos
   - DES
   - 3DES
   - AES
3. Algoritmos Asimétricos
   - RSA
   - DSA
   - Curva Elíptica
4. Firma electrónica y algoritmos
   - Funciones de una vía
   - Firma electrónica
   - Verificar firma electrónica
   - Notas finales sobre firma electrónica
5. Protocolos SSL, TLS
   - Protocolo SSL

Técnicas Criptográficas

Criptografía

Se pretende ocultar la información contenida en un mensaje; el mensaje mismo está accesible a cualquiera.
Permiten la autenticación, firmas digitales, y verificación de la integridad de un mensaje específico.

$$\text{Cifrar} \rightarrow \text{documento} \rightarrow \text{algoritmo} \rightarrow \text{claves}$$

Técnicas Criptográficas

Criptografía

Cifrar o descrifrar información (digital) utilizando técnicas matemáticas.

Las matemáticas son útiles y fundamentales!

Criptosistema

En general:

un conjunto de algoritmos necesarios para implementar una forma particular de cifrado y descrifrado.

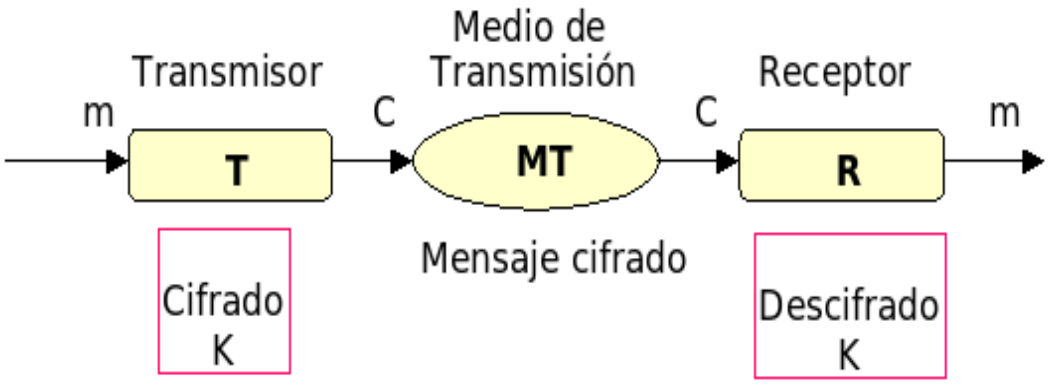
Técnicamente:

una quíntupla (m,C,K,E,D), donde:

- m mensajes sin cifrar (texto plano)
- C posibles mensajes cifrados
- K claves que se pueden emplear en el criptosistema
- E transformaciones de cifrado
- D transformaciones de descifrado

Criptosistema

- $D_k(E_k(m)) = m$



Criptosistema

Misión de un criptosistema:

- integridad
- confidencialidad
- autenticación

Criptosistema

Característica de un criptosistema

- Tiene que tener cumplir con el teorema de Shannon (Teoría de la información) “Secreto Perfecto”

Técnicas Criptográficas

Se reconocen varias clases de cifrado:

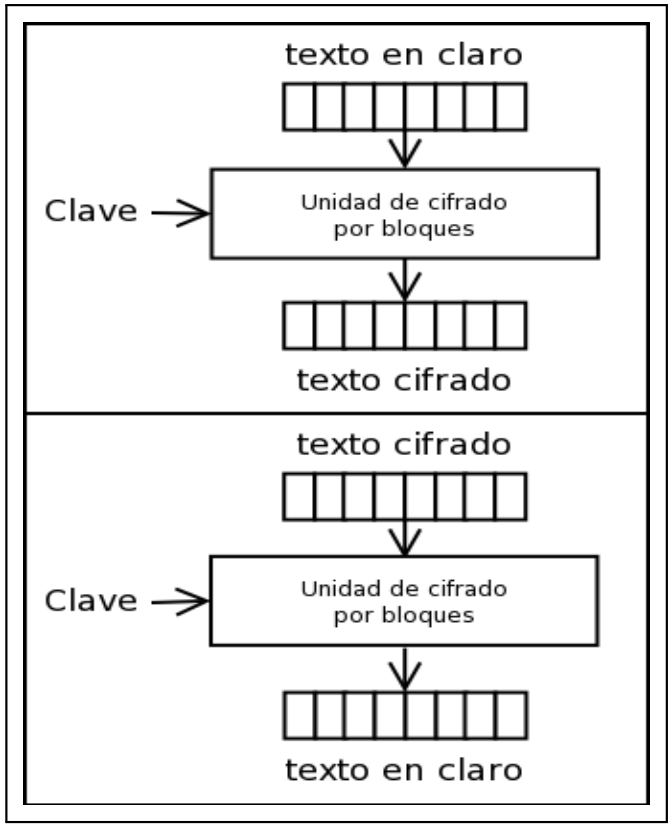
- Cifrado por flujo
- Cifrado por bloques

Cifrado por flujo

- Conocido como Stream cipher
- Realiza el proceso de cifrado convirtiendo el texto plano en texto cifrado bit a bit.
- Utilizan claves de tamano pequeño (128 bits).
- La transformación sobre los bits varía.
- RC4 es uno de los cifrados por flujo más utilizado.
- Se utiliza en aplicaciones donde la longitud del texto plano no es conocida.
- Excepcionalmente más rápido que el cifrado por bloques.

Cifrado por bloques

- Conocido como block cipher.
- Realiza el proceso de cifrado sobre bloques de texto plano de longitud fija.
- La transformación sobre los bits no varía; siempre es la misma.
- DES y AES son ejemplos de cifrado por bloques.

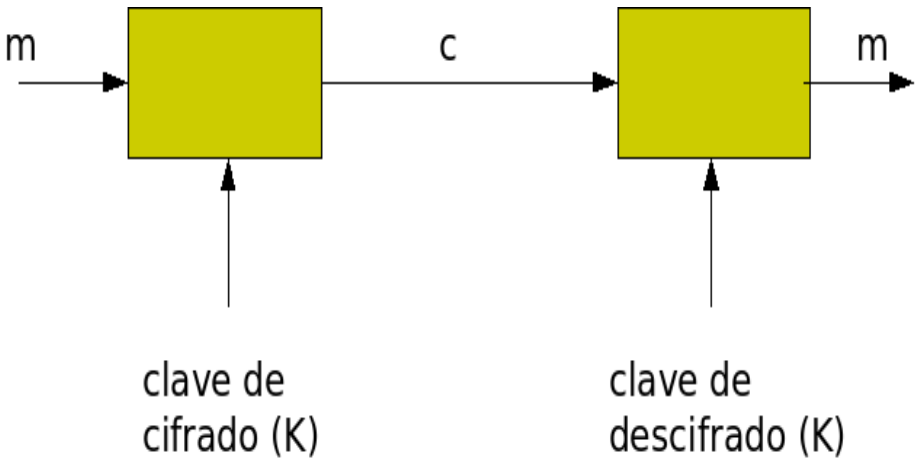


De acuerdo al tipo de clave que utiliza se tiene:

- Cifrado Simétrico (criptografía simétrica)
- Cifrado Asimétrico (criptografía asimétrica)

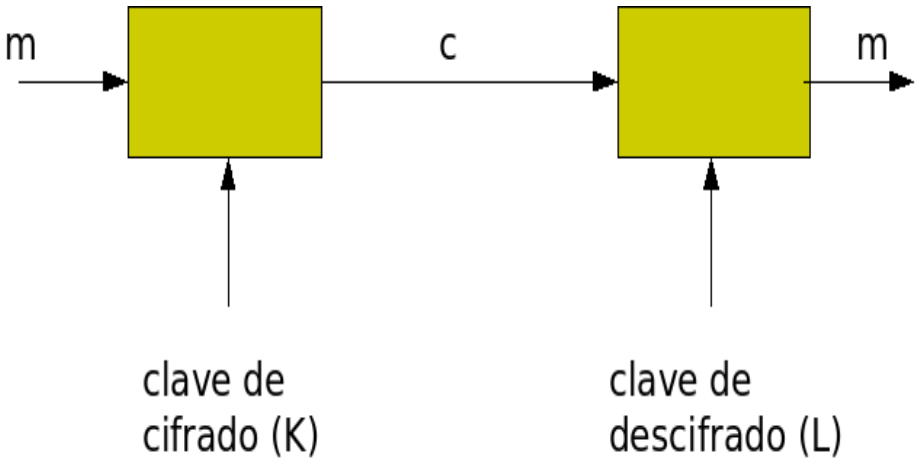
Cifrado Simétrico

- Se utiliza una sola clave para cifrar/descifrar (clave privada).
- Los extremos que se comunican debe conocer la clave privada.
- Considerablemente importante el intercambio de la clave privada.



Cifrado Asimétrico

- Se utiliza una par de claves para cifrar/descifrar.
- Lo que se cifra con una clave se descifra con la otra.
- Utilizan longitudes de claves mayores que cifrado simétrico.
- Utilizan funciones de una vía.
- Más lento que el cifrado simétrico.



Cifrado Híbrido

- Combinación de cifrado simétrico con asimétrico.
- Uso de cifrado asimétrico para compartir la clave secreta simétrica.
- Uso de cifrado simétrico para cifrar el mensaje.
- La clave simétrica se cambia en el tiempo para cada sesión.
- Velocidad en la transmisión con cifrado simétrico.

Algoritmos Simétricos

- Algoritmos utilizados en cifrado simétrico.
- Entre los algoritmos más conocidos están:
  - DES
  - 3DES
  - AES
  - IDEA
  - Blowfish

Data Encryption Standard

- Desarrollado en los años 70.
- Tamaño de clave 56 bits.
- Genera bloques de cifrado de 64 bits.
- Estuvo reconocido como estándar en el FIPS PUB 46
- No se utiliza en aplicaciones dada su pequeña longitud de clave.
- versión mejorada: **3DES**

Triple Data Encryption Standard

- Surge como respuesta ante ataques de fuerza bruta de DES.
- Utiliza tres (3) claves en su operación.
- Tamaño de clave de 128 bits.
- Genera bloques de cifrado de 64 bits.

Advanced Encryption Standard

- Conocido como Rijndael.
- Algoritmo de cifrado por bloques.
- Tamaño de clave 128, 192, 256 bits.
- Genera bloques de cifrado de 128 bits.

Algoritmos Asimétricos

- Algoritmos utilizados en cifrado asimétrico.
- Entre los algoritmos más conocidos están:
  - RSA
  - DSA
  - Curva Elíptica
  - Diffie-Hellman
  - El Gamal

- Creado en 1977 por **R**ivest, **S**hamir y **A**delman y publicado en 1997.
- Se basa en la dificultad de factorizar números primos muy grandes.
- Es considerado uno de los algoritmos criptográficos más exitoso de clave pública/privada.
- Lentitud con respecto a algoritmos simétricos.
- Longitudes de clave mucho mayor que en algoritmos simétricos.
- Utilizado en la firma electrónica y cifrado.
- Tamaños de clave: 512, 1024, 2048, 4096 bits.

Implementación del algoritmo

1. El receptor potencial elige dos números primos (p y q), se calcula $z = p \bullet q$
2. El receptor potencial calcula $\phi = (p-1) \bullet (q-1)$ y elige n tal que $mcd(n, \phi) = 1$; n suele ser primo
3. El par (n, z) forman la clave pública
4. El receptor potencial calcula el número s tal $0 < s < \phi$ y satisface $ns \ mod \ \phi = 1$; el exponente s es la clave privada.

Implementación del algoritmo

Cifrado

$$c = a^n \ mod \ z$$

Descifrado

$$a = c^s \ mod \ z$$

Implementación del algoritmo

Ejemplo

1. seleccionar p y q; $p = 23$ $q = 31$
2. calcular $z = p \bullet q = 713$
3. calcular $\phi = (p-1) \bullet (q-1) = 660$
4. escoger $1 < n < \phi$; $n = 29$
5. calcular s tal que: $ns \ mod \ \phi = 1$; $29 \bullet 569 \ mod \ 660 = 1$

cifrar

$$c = 572^{29} \ mod \ 713 = 113$$

descifrar

$$a = 113^{569} \ mod \ 713 = 572$$

- Algoritmo de Firma Digital (Digital Signature Algorithm)
- Reconocido como un estándar FIPS del NIST (USA).
- Patentado en USA.
- Sólo puede ser utilizado en firma electrónica (no cifrado).
- Tamaños de clave: 1024 bits.

- Nacimiento a mediados de los 80.
- Siguente algoritmo asimétrico más reconocido después de RSA.
- Util en aplicaciones donde la memoria, el ancho de banda y el poder computacional son limitados.
- Utilizan tamaños de claves más pequeños que RSA: clave de 160 bits de curva elíptica equivale a 1024 bits en RSA.
- Más rápidos que RSA para firmar y descifrar, pero más lentos para verificar firma y cifrar.

Funciones de una vía o funciones hash

La criptografía asimétrica permite identificar al emisor y al receptor del mensaje. Para identificar el mensaje propiamente dicho se utilizan las llamadas funciones ***hash***.

Características:

- entrada variable y salida de longitud fija (128 ó 160 bits).
- no se puede generar la salida sin la misma entrada.
- dos entradas no deberían producir una misma salida.
- cualquier cambio en la entrada, modifica la salida.

Funciones de una vía o funciones hash

Producen una huella digital (fingerprint) o reseña de tamaño fijo a partir de datos de longitud variable.

- MD5 (longitud de 128 bits)
- SHA-1 (longitud de 160 bits)
- SHA-256 (longitud de 256 bits)
- SHA-512 (longitud de 512 bits)

Y todo esto para qué?

Firma electrónica

Mecanismo para verificar la autenticación y la integridad.

Firma electrónica $\rightarrow$ (sellos)

Resolución de problemas de confidencialidad e integridad.

Comparación con la firma autógrafa

La firma electrónica es la autógrafa digitalizada?

NO

La firma electrónica es un método criptográfico que asegura la integridad del documento firmado así como la identidad del firmante.

01010101101 SÍ

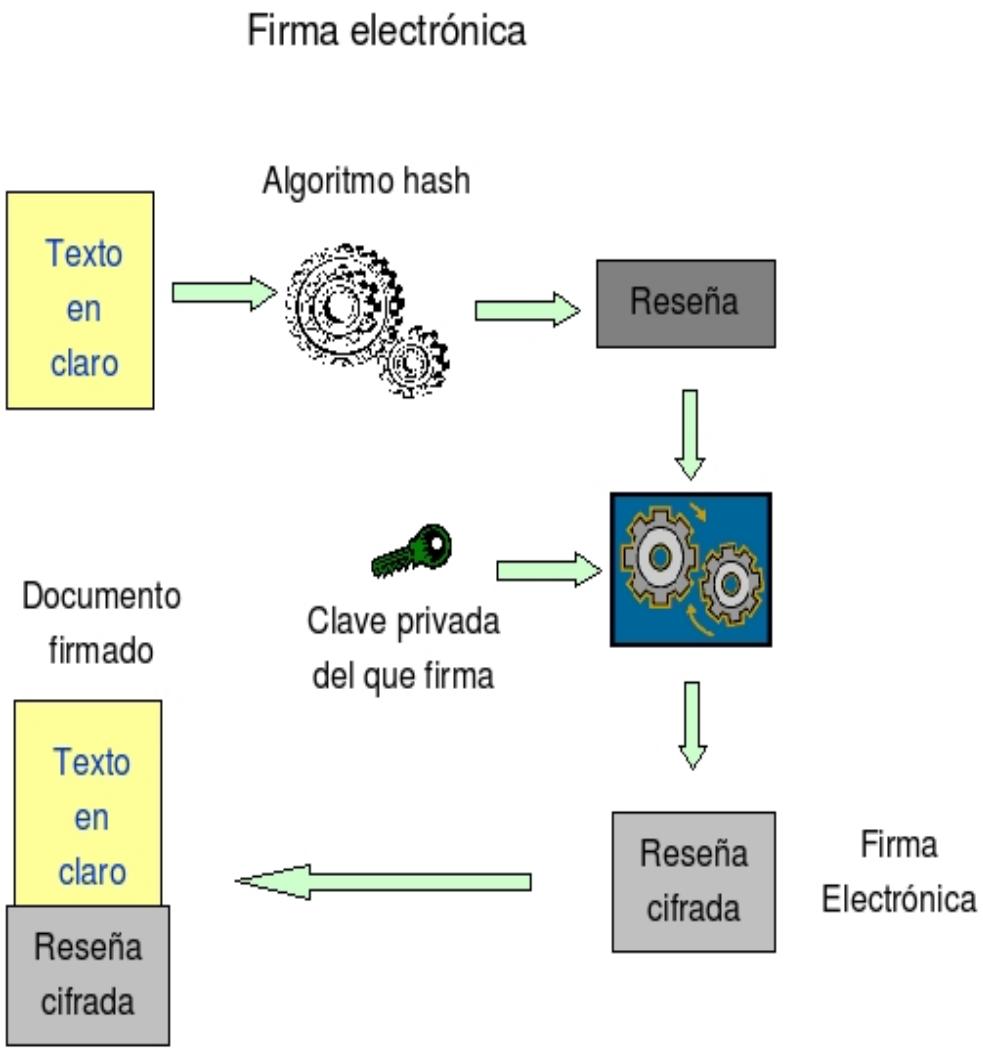
Comparación con la firma autógrafa

La firma electrónica es legalmente equivalente a la firma autógrafa? Podría decirse que sí, pero debe estar validada por una Autoridad de Certificación (Tercero de confianza) que expida un certificado digital que diga que la firma es válida.

Firmar electrónicamente

La firma electronica es el resultado de aplicar cierto algoritmo matematico, denominado función hash, a su contenido. Esta función asocia un valor dentro de un conjunto finito (generalmente los números naturales) a su entrada. Cuando la entrada es un documento, el resultado de la función (resena) es un número que identifica casi unívocamente al texto. Si se adjunta éste número al texto de manera cifrada con el algortimo asimétrico usando la clave privada del que firma.

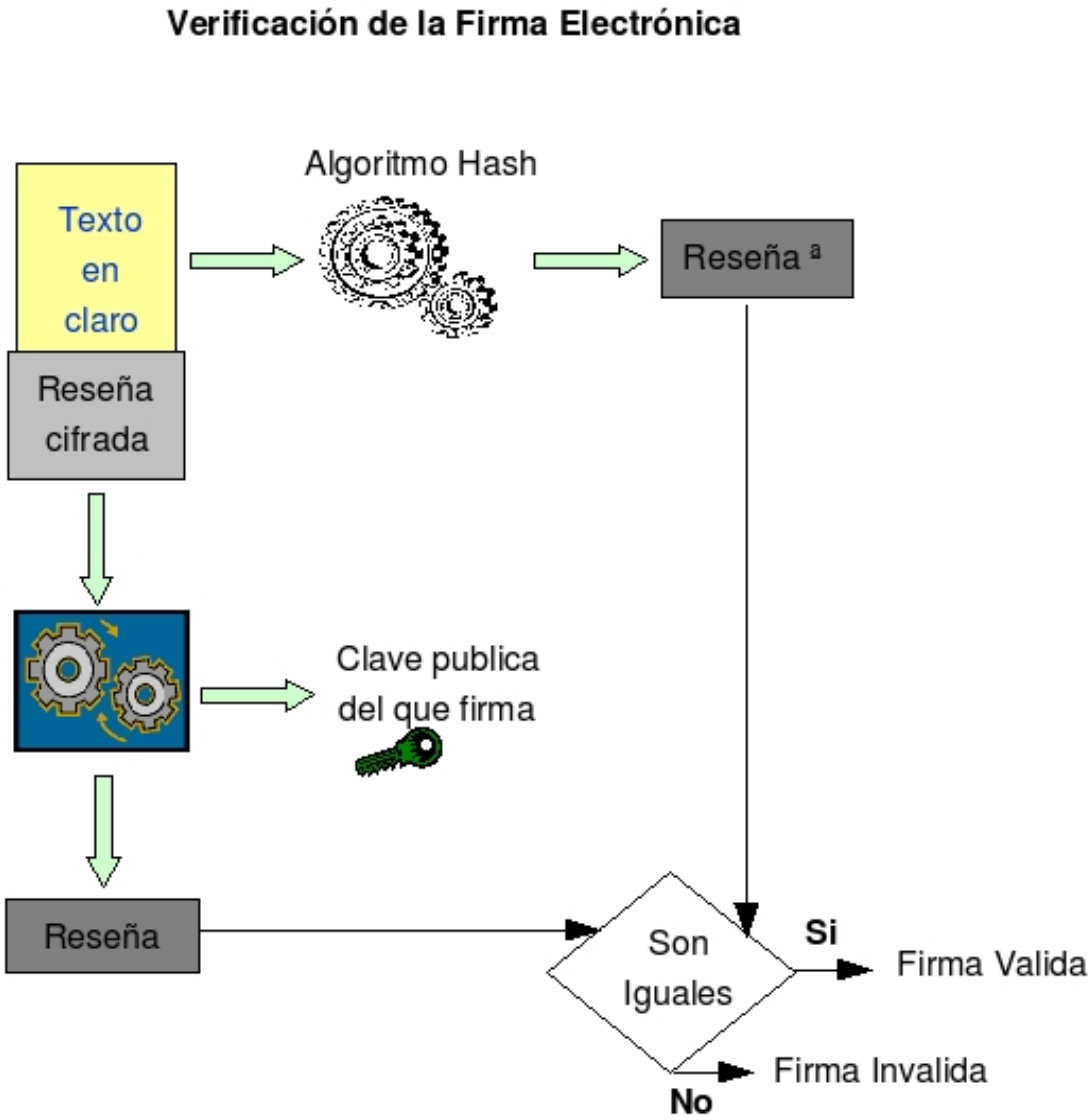
Firmar electrónicamente



Verificar firma electrónicamente

El destinatario debe aplicar de nuevo la función hash al texto en claro y comparar su resultado (resena) con la que ha recibido, que se tiene que descifrar usando la clave pública del firmante. Si ambas son iguales, tiene la seguridad de que el texto no fue modificado una vez que fue firmado y que lo envió la persona poseedora de la clave privada que firmó el texto.

Verificar firma electrónicamente



Notas finales

Firma no válida $\rightarrow$ falla de autenticidad/integridad

Firmar electrónicamente $\rightarrow$ calcular hash y cifrar!

Se pueden firmar claves (integridad)

Se puede firmar todo!

Algunos protocolos criptográficos

Permiten asegurar las conexiones entre extremos a nivel de transporte. Ejemplos:

- SSL
- TLS

Secure Socket Layer

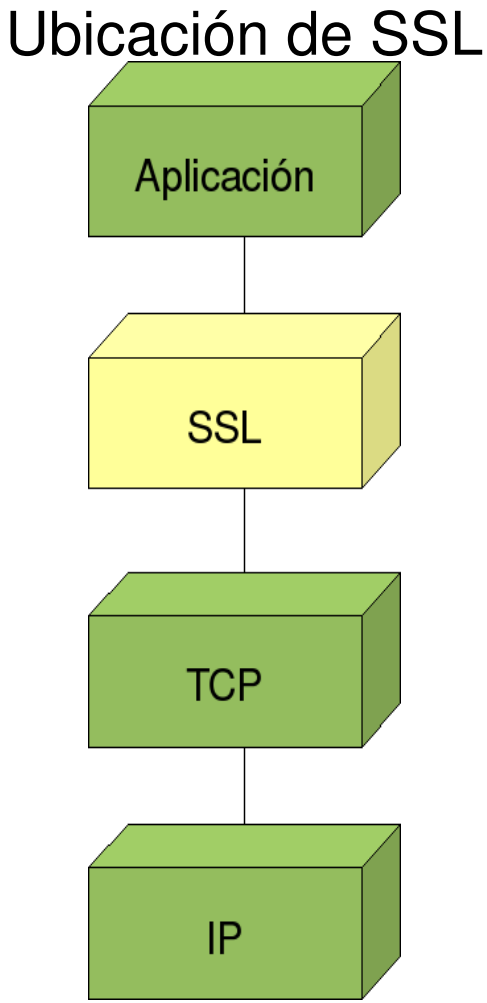
- Protocolo de capa de socket seguro.
- Propuesto por Netscape.
- Versión actual del protocolo: 3.0
- Utilización de certificados digitales.
- Autenticación de servidores y clientes.

Secure Socket Layer

Objetivos del protocolo SSL:

- Seguridad Criptográfica: establecimiento de conexión segura entre partes
- Interoperabilidad
- Extensibilidad
- Eficiencia relativa

Secure Socket Layer



Secure Socket Layer

Establecimiento de comunicación segura con SSL

Transport Layer Security

- Protocolo de Seguridad de la Capa de Transporte.
- Estándar del IETF, desarrollado a partir de SSL.
- Diferencias muy pequeñas con SSL.
- Versión actual del TLS: 1.1 (propuesta la 1.2 en marzo de 2008).

Transport Layer Security

El protocolo TLS sigue tres (3) fases en su operación básica:

- Negociación entre los extremos para especificar soporte de algoritmos.
- Intercambio de claves y autenticación.
- Cifrado simétrico y autenticación de mensaje.

- OpenSSL
- GnuTLS
- NSS

OpenSSL: La herramienta

- OpenSSL es un conjunto de herramientas de código abierto que implementan los protocolos SSL (v2/v3) y TLS (v1).
- Se encuentra disponible para distintas plataformas.
- Versión actual: 0.9.8g
- Se puede instalar desde los códigos fuentes o desde paquetes precompilados para distintas distribuciones Linux.
- http://www.openssl.org

OpenSSL: La herramienta

- Se incluye el programa openssl que permite realizar varias operaciones criptográficas desde una terminal. Se puede utilizar para:
  - Crear claves RSA, DSA, DH.
  - Crear certificados X509, solicitudes de firma de certificados (CSR) y listas de certificados revocados (CRL).
  - Calcular reseñas de mensajes
  - Cifrar y descifrar
  - Manejar correo firmado o cifrado S/MIME.
- Se incluye la librería para desarrollar aplicaciones con soporte de OpenSSL.

Ejercicios: Funciones hash y reseñas

Recuerdan lo que es una función hash?

Ejercicios: Funciones hash y reseñas

Recuerdan lo que es una función hash?

Recuerdan algunos algoritmos utilizados para obtener reseñas?

TIP: puede usar el comando man para obtener ayuda.

Ejercicios: Funciones hash y reseñas (dgst)

- Calcular el hash SHA1 para un archivo llamado archivo.txt:
  - `openssl dgst -sha1 archivo.txt`

Ejercicios: Funciones hash y reseñas (dgst)

- Calcular el hash SHA1 para un archivo llamado archivo.txt:
  - `openssl dgst -sha1 archivo.txt`
  - Editar archivo.txt y agregar un caracter al inicio del archivo. Calcular de nuevo el hash de archivo.txt y comparar.

Ejercicios: Funciones hash y reseñas (dgst)

- Calcular el hash SHA1 para un archivo llamado archivo.txt:
  - `openssl dgst -sha1 archivo.txt`
  - Editar archivo.txt y agregar un caracter al inicio del archivo. Calcular de nuevo el hash de archivo.txt y comparar.
  - Discutir el resultado.

Ejercicios: Cifrado simétrico (enc)

- Cifrar el archivo llamado archivo2.txt utilizando el algoritmo 3DES
  - `openssl enc -des3 -salt -in archivo2.txt -out cifrado1.bin`

Ejercicios: Cifrado simétrico (enc)

- Cifrar el archivo llamado archivo2.txt utilizando el algoritmo 3DES
  - `openssl enc -des3 -salt -in archivo2.txt -out cifrado1.bin`
  - Alguna observación al ejecutar el comando?

Ejercicios: Cifrado simétrico (enc)

- Cifrar el archivo llamado archivo2.txt utilizando el algoritmo 3DES
  - `openssl enc -des3 -salt -in archivo2.txt -out cifrado1.bin`
  - Alguna observación al ejecutar el comando?
  - Puede ver el contenido del archivo cifrado1.bin?

Ejercicios: Cifrado simétrico (enc)

- Cifrar el archivo llamado archivo2.txt utilizando el algoritmo 3DES
  - `openssl enc -des3 -salt -in archivo2.txt -out cifrado1.bin`
- Ahora descifrar el contenido del archivo cifrado1.bin
  - `openssl enc -des3 -salt -d -in cifrado1.bin -out descifrado.txt`

Ejercicios: Criptografía de clave pública

- Generar una clave RSA de 1024 bits
  - `openssl genrsa -out miclaveprivada.pem 1024`

Ejercicios: Criptografía de clave pública

- Generar una clave RSA de 1024 bits
  - `openssl genrsa -out miclaveprivada.pem 1024`
  - pruebe: `openssl rsa -in miclaveprivada.pem -noout -text`

Ejercicios: Criptografía de clave pública

- Generar una clave RSA de 1024 bits
  - `openssl genrsa -out miclaveprivada.pem 1024`
- Obtener el componente público de la clave privada generada
  - `openssl rsa -in miclaveprivada.pem -pubout -out miclavepublica.pem`

Ejercicios: Criptografía de clave pública

- Generar una clave RSA de 1024 bits
  - `openssl genrsa -out miclaveprivada.pem 1024`
- Generar una clave RSA de 1024 bits y cifrarla con el algoritmo 3DES
  - `openssl genrsa -out miclave2.pem -des3 1024`

Ejercicios: Criptografía de clave pública

- Generar una clave RSA de 1024 bits
  - `openssl genrsa -out miclaveprivada.pem 1024`
- Generar una clave RSA de 1024 bits y cifrarla con el algoritmo 3DES
  - `openssl genrsa -out miclave2.pem -des3 1024`
- Cifrar archivo3.txt usando el algoritmo 3DES con la clave puública RSA generada (miclavepublica.pem)
  - `openssl rsautl -encrypt -pubin -inkey miclavepublica.pem -in archivo3.txt -out cifrado2.txt`

Ejercicios: Criptografía de clave pública

- Generar una clave RSA de 1024 bits
  - `openssl genrsa -out miclaveprivada.pem 1024`
- Generar una clave RSA de 1024 bits y cifrarla con el algoritmo 3DES
  - `openssl genrsa -out miclave2.pem -des3 1024`
- Cifrar archivo3.txt usando el algoritmo 3DES con la clave puública RSA generada (miclavepublica.pem)
  - `openssl rsautl -encrypt -pubin -inkey miclavepublica.pem -in archivo3.txt -out cifrado2.bin`
- Descifrar el archivo cifrado2.bin con la clave privada RSA generada (miclaveprivada.pem)
  - `openssl rsautl -decrypt -inkey miclaveprivada.pem -in cifrado2.bin -out salida.txt`

Ejercicios: Criptografía de clave pública

- Firmar archivo3.txt con la clave privada (miclaveprivada.pem)
  - `openssl rsautl -sign -inkey miclaveprivada.pem -in archivo3.txt -out firma.bin`

Ejercicios: Criptografía de clave pública

- Firmar archivo3.txt con la clave privada (miclaveprivada.pem)
  - `openssl rsautl -sign -inkey miclaveprivada.pem -in archivo3.txt -out firma.bin`
- Verificar firma en firma.bin con la clave publica (miclavepublica.pem)
  - `openssl rsautl -verify -pubin -inkey miclavepublica.pem -in firma.bin -out original.txt`

Preguntas, comentarios.