Utilización de técnicas de programación segura (Programación de servicios y procesos)

Prácticas de programación segura

Listado de buenas prácticas a la hora de escribir código seguro:

- Estar al día en temas de seguridad.
- Precaución en el manejo de datos: limpiar datos, comprobar variables, ficheros, parámetros, URL, etc.
- Reutilizar código que se sabe que es seguro.
- Tener prácticas de de revisión de fallos: revisión por parte de más personas, verificaciones independientes, uso de herramientas de seguridad.
- Configuración correcta de permisos y roles.
- Evitar lo siguiente: rutas relativas, llamar a programas de forma no segura, pensar que el usuario actúa con buena intención, contraseñas inseguras, enviar contraseñas por mail, distribuir información sin seguridad, etc.

Criptografía de clave pública y clave privada

Etimológicamente la palabra Criptología proviene del griego (criptos = oculto y logos = ciencia, tratado) y denota de forma genérica dos disciplinas opuestas pero a su vez complementarias: la criptografía y el criptoanálisis.

La Criptografía diseña procedimientos para cifrar, es decir, para ocultar o enmascarar una determinada información confidencial.

La Criptografía se puede considerar la parte «constructiva» de este proceso mientras que el Criptoanálisis es claramente la parte «destructiva». Sin embargo, ambas disciplinas siempre se han desarrollado de forma conjunta puesto que cualquier procedimiento de cifrado da lugar a un criptoanálisis o, al menos, a un intento de criptoanálisis. El criptoanálisis de define como el conjunto de procedimientos, procesos y métodos empleados para romper un algoritmo criptográfico, desencriptar un texto cifrado o descubrir las claves empleadas para generarlo.

La criptografía se ha definido, tradicionalmente, como el ámbito de la criptología que se ocupa de las técnicas de cifrado o codificado destinadas a alterar las representaciones lingüísticas de ciertos mensajes con el fin de hacerlos ininteligibles a receptores no autorizados.

Ejercicios

- https://www.jesusninoc.com/02/22/ejercicios-de-java-contar-las-veces-que-aparece-una-palabra-en-un-texto/
- https://www.jesusninoc.com/02/01/criptografia-en-powershell/#Anlisis-de-frecuencias
- https://www.jesusninoc.com/02/01/criptografia-en-powershell/#Criptoanlisis

---

Más información

- https://www.jesusninoc.com/02/17/ejercicios-de-seguridad-detectar-mediante-firmas-que-un-fichero-se-ha-

infectado/

- https://www.jesusninoc.com/02/17/ejercicios-de-seguridad-adivinar-un-hash-utilizando-un-fichero-con-hashes-generados/
- https://www.jesusninoc.com/02/28/seguridad-informatica-con-powershell/#Criptografia
- https://www.jesusninoc.com/2017/11/06/creates-a-new-self-signed-certificate/
- https://www.jesusninoc.com/2017/11/10/encrypts-content-by-using-the-cryptographic-message-syntax-format/
- https://www.jesusninoc.com/2017/11/12/decrypts-content-that-has-been-encrypted-by-using-the-cryptographic-message-syntax-format/
- https://www.jesusninoc.com/2017/11/18/exports-a-certificate-to-a-personal-information-exchange-pfx-file/
- https://www.jesusninoc.com/2017/11/20/imports-certificates-and-private-keys-from-a-personal-information-exchange-pfx-file-to-the-destination-store/
- https://www.jesusninoc.com/01/23/cifrar-con-un-algoritmo-sencillo-el-nombre-y-el-contenido-de-un-fichero-de-texto/

Según sean las claves utilizadas en el proceso de cifrado/descifrado, existe una primera clasificación de métodos criptográficos:

Métodos criptográficos de clave simétrica

Son aquellos en los que la clave de cifrado coincide con la clave de descifrado. Esta clave única tiene que permanecer secreta y ser conocida exclusivamente por A y B. Este hecho presupone que emisor y receptor se han puesto previamente de acuerdo en la determinación de la misma o que existe un centro de distribución de claves, que por un canal seguro, la hace llegar a ambos comunicantes. Estos métodos tienen pues que solventar el difícil problema de la distribución de claves.

Métodos criptográficos de clave asimétrica

Son aquellos en los que la clave de cifrado es diferente a la de descifrado. En general la clave de cifrado es conocida libremente por el público, mientras que la de descifrado es conocida únicamente por el usuario sin tener que compartirla con nadie. Estos métodos claramente evitan el problema de la distribución de claves.

Principales aplicaciones de la criptografía

La criptografía es una disciplina con multitud de aplicaciones, muchas de las cuales están en uso hoy en día. Entre las más importantes destacamos las siguientes:

- Seguridad de las comunicaciones.
- Identificación y autentificación.
- Certificación.
- Comercio electrónico.

Seguridad de las comunicaciones

Es la principal aplicación de la criptografía a las redes de computadores, ya que permiten establecer canales seguros sobre redes que no lo son. Además, con la potencia de cálculo actual y empleando algoritmos de cifrado simétrico (que se intercambian usando algoritmos de clave pública) se consigue la privacidad sin perder velocidad en la transferencia.

Identificación y autentificación

Gracias al uso de firmas digitales y otras técnicas criptográficas es posible identificar a un individuo o validar el acceso a un recurso en un entorno de red con más garantías que con los sistemas de usuario y clave tradicionales.

Certificación

La certificación es un esquema mediante el cual agentes fiables (como una entidad certificadora) validan la identidad de agentes desconocidos (como usuarios reales). El sistema de certificación es la extensión lógica del uso de la criptografía para identificar y autentificar cuando se emplea a gran escala.

Comercio electrónico

Gracias al empleo de canales seguros y a los mecanismos de identificación se posibilita el comercio electrónico, ya que tanto las empresas como los usuarios tienen garantías de que las operaciones no pueden ser espiadas, reduciéndose el riesgo de fraudes y robos.

Más información

- https://www.jesusninoc.com/02/04/cifrar-desde-powershell-y-descifrar-desde-php/

Protocolos criptográficos

Un protocolo criptográfico o protocolo de seguridad (también llamado protocolo de cifrado) es un protocolo abstracto o concreto que realiza funciones relacionadas con la seguridad, aplicando métodos criptográficos.

Los protocolos criptográficos se usan ampliamente para transporte de datos seguros a nivel de aplicación.

Un protocolo criptográfico comúnmente incorpora por lo menos uno de los siguientes aspectos:

- Establecimiento de claves.
- Autenticación de entidades.
- Cifrado simétrico y autenticación de mensajes.
- Transporte de datos en forma segura a nivel de aplicación.
- Métodos de no repudio.

Política de seguridad

Los ficheros de política de seguridad sirven para configurar qué permisos están habilitados para el código obtenido de los orígenes de código especificados.

Ejercicio

- https://www.jesusninoc.com/02/26/crear-y-compilar-un-codigo-que-escribe-en-un-fichero-con-la-propiedad-securitymanager-sin-un-fichero-policy-y-utilizando-un-fichero-policy-en-java/

Más información

- https://www.jesusninoc.com/02/26/mostrar-propiedades-java-class-path-java-home-java-vendor-java-version-os-name-os-version-user-dir-user-home-user-name-utilizando-y-sin-utilizar-el-gestor-de-seguridad-en-java/

Programación de mecanismos de

control de acceso

Depende del sistema operativo y lenguaje de programación que se utilicen.

Encriptación de información (cifrado)

El cifrado es un procedimiento que utiliza un algoritmo de cifrado con cierta clave para transformar un mensaje, sin atender a su estructura lingüística o significado, de tal forma que sea incomprensible o, al menos, difícil de comprender a toda persona que no tenga la clave secreta del algoritmo.

Algunos algoritmos que sirven para cifrar son (dependiendo de la tarea que se quiera realizar):

Tipo de Algoritmo	**Algoritmo**
Cifrado Simétrico	TDEA (Triple Data Encryption Algorithm)
	AES (Advanced Encryption Standard)
Algoritmos Asimétricos	DSA (Digital Signature Algorithm)
	ECDSA (Elliptic Curve Digital Signature Algorithm)
	RSA (Criptosistema RSA)
	ECIES (Elliptic Curve Integrated Encryption Scheme)
Intercambio de Clave	DH o DHKA (Diffie-Hellman Key Agreement)
	MQV (Menezes-Qu-Vanstone Key Agreement)
	ECDH (Elliptic Curve Diffie-Hellman)

Tipo de Algoritmo	Algoritmo
	ECMQV (Elliptic Curve Menezes-Qu-Vanstone)
Funciones resumen (Hash)	SHA-2 (Secure Hash Algorithm)
	HMAC (Hash Message Authentication Code)

Veamos detalles de los principales algoritmos.

Cifrado Simétrico

Dentro de estos algoritmos distinguimos dos tipos de algoritmos en función de la cantidad de datos de entrada que manejan a la vez: algoritmos de cifrado por bloques y algoritmos de cifrado de flujo.

Cifrado por bloques

Los algoritmos de cifrado por bloques toman bloques de tamaño fijo del texto en claro y producen un bloque de tamaño fijo de texto cifrado, generalmente del mismo tamaño que la entrada. El tamaño del bloque debe ser lo suficientemente grande como para evitar ataques de texto cifrado. La asignación de bloques de entrada a bloques de salida debe ser uno a uno para hacer el proceso reversible y parecer aleatoria.

Para aplicar un algoritmo por bloques es necesario descomponer el texto de entrada en bloques de tamaño fijo. Esto se puede hacer de varias maneras:

1. **ECB (Electronic Code Book)**. Se parte el mensaje en bloques de k bits, rellenando el último si es necesario y se encripta cada bloque. Para desencriptar se trocea el texto cifrado en bloques de k bits y se desencripta cada bloque. Este sistema es vulnerable a ataques ya que dos bloques idénticos de la entrada generan el mismo bloque de salida. En la práctica no se utiliza.
2. **CBC (Cipher Block Chaining)**. Este método soluciona el

problema del ECB haciendo una o-exclusiva de cada bloque de texto en claro con el bloque anterior cifrado antes de encriptar. Para el primer bloque se usa un vector de inicialización. Este es uno de los esquemas más empleados en la práctica.
3. **OFB (Output Feedback Mode)**. Este sistema emplea la clave de la sesión para crear un bloque pseudoaleatorio grande (pad) que se aplica en o-exclusiva al texto en claro para generar el texto cifrado. Este método tiene la ventaja de que el pad puede ser generado independientemente del texto en claro, lo que incrementa la velocidad de encriptación y desencriptación.
4. **CFB (Cipher Feedback Mode)**. Variante del método anterior para mensajes muy largos.

Más información

- https://www.jesusninoc.com/02/08/cifrar-y-descifrar-con-aes-rijndael-de-256-y-modo-de-operacion-de-unidad-de-cifrado-ecb-desde-powershell/
- https://www.jesusninoc.com/02/08/cifrar-y-descifrar-con-aes-desde-php-utilizando-el-modo-de-operacion-de-cifrado-de-bloques-ecb/

Cifrado de flujo de datos

Generalmente operan sobre 1 bit (o sobre bytes o palabras de 16 ó 32 bits) de los datos de entrada cada vez. El algoritmo genera una secuencia (secuencia cifrante o keystream en inglés) de bits que se emplea como clave. La encriptación se realiza combinando la secuencia cifrante con el texto en claro.

Algoritmos de cifrado simétrico:

AES (Advanced Encryption Standard)

También conocido como Rijndael (pronunciado «Rain Doll» en inglés), es un esquema de cifrado por bloques adoptado como un estándar de cifrado por el gobierno de los Estados Unidos, creado en Bélgica. El AES fue anunciado por el Instituto Nacional de Estándares y Tecnología (NIST) como FIPS PUB 197 de los Estados Unidos (FIPS 197) el 26 de noviembre de 2001 después de un proceso de estandarización que duró 5 años. Se transformó en un estándar efectivo el 26 de mayo de 2002. Desde 2006, el AES es uno de los algoritmos más populares usados en criptografía simétrica.

EN 2011 un equipo de investigadores descubrieron la primera vulnerabilidad en el estándar de cifrado AES (Advanced Encryption Standard) que reduce la longitud de la llave efectiva del algoritmo en dos bits. Esto significa que las longitudes usuales de la llave de 128, 192 y 256 bits son reducidas a 126, 190 y 254 bits. Los investigadores emplearon un ataque Meet-in-the-middle, un enfoque que hasta ahora ha sido utilizado principalmente con algoritmo de hashing, combinándolo con un ataque «Biclique». Su método permitió a los investigadores calcular la llave de un simple par de texto plano / texto cifrado, de forma más rápida que lanzando un ataque de fuerza bruta en el espacio de claves completo.

Más información

- https://www.jesusninoc.com/08/18/cifrar-utilizando-una-clave-secreta/
- https://www.jesusninoc.com/08/20/descifrar-utilizando-una-clave-secreta/
- https://www.jesusninoc.com/02/05/cifrar-y-descifrar-con-aes-desde-java-con-clave-aleatoria/
- https://www.jesusninoc.com/02/05/cifrar-y-descifrar-con-aes-desde-java-con-clave-anadida/

- https://www.jesusninoc.com/02/05/almacenar-la-clave-secreta-generada-con-el-algoritmo-aes-en-un-fichero-en-java/
- https://www.jesusninoc.com/02/08/cifrar-y-descifrar-con-aes-desde-powershell/
- https://www.jesusninoc.com/02/08/cifrar-con-aes-desde-powershell/
- https://www.jesusninoc.com/02/08/descifrar-con-aes-desde-powershell/
- https://www.jesusninoc.com/02/08/cifrar-y-descifrar-con-aes-desde-powershell/
- https://www.jesusninoc.com/02/08/cifrar-y-descifrar-con-aes-rijndael-de-256-y-modo-de-operacion-de-unidad-de-cifrado-ecb-desde-powershell/
- https://www.jesusninoc.com/02/08/cifrar-y-descifrar-con-aes-desde-php-utilizando-el-modo-de-operacion-de-cifrado-de-bloques-ecb/
- https://www.jesusninoc.com/02/10/cifrar-y-descifrar-con-clave-clave-secreta-aes-que-se-ha-transmitido-mediante-cifrado-asimetrico-rsa-en-java/

DES

El DES (Data Encription Standard o Estándar de Encriptación de Datos) es el nombre del documento FIPS (Federal Information Processing Standard) 46-1 del Instituto Nacional de Estándares y Tecnología (NIST) del Departamento de Comercio de Estados Unidos. Fue publicado en 1977. Es el algoritmo de cifrado simétrico más estudiado, mejor conocido y más empleado del mundo.

El principal problema que tiene DES es que tiene una clave de cifrado corta, solo 56 bits. Se dice que fue así para permitir a las NSA (Agencia de Seguridad Nacional) desencriptar fácilmente documentos.

El ataque típico sobre el algoritmo DES es la fuerza bruta, se

trata de un ataque sencillo y fácil de llevar a cabo. Como medida de defensa existe una variante del DES que vemos a continuación 3DES.

Más información

- https://www.jesusninoc.com/03/15/cifrar-y-descifrar-con-des-desde-powershell-con-clave-aleatoria/
- https://www.jesusninoc.com/03/15/cifrar-y-descifrar-con-des-desde-powershell-con-clave-anadida/
- https://www.jesusninoc.com/03/15/descifrar-un-mensaje-cifrado-con-des-comprobando-contrasenas-de-un-diccionario/

Triple-DES

Consiste en encriptar tres veces una clave DES. Esto se puede hacer de varias maneras:

- DES-EEE3: Tres encriptaciones DES con tres claves distintas.
- DES-EDE3: Tres operaciones DES con la secuencia encriptar-desencriptar-encriptar con tres claves diferentes.
- DES-EEE2 y DES-EDE2: Igual que los anteriores pero la primera y tercera operación emplean la misma clave.

Más información

- https://www.jesusninoc.com/01/24/cifrar-y-descifrar-con-tripledes-desde-powershell/
- https://www.jesusninoc.com/03/15/descifrar-un-mensaje-cifrado-con-triple-des-comprobando-contrasenas-de-un-diccionario/

RC2

El RC2 es un algoritmo de cifrado por bloques de clave de tamaño variable diseñado por Ron Rivest de RSA Data Security (la RC quiere decir Ron’s Code o Rivest’s Cipher).

El algoritmo trabaja con bloques de 64 bits y entre dos y tres veces más rápido que el DES en software. Se puede hacer más o menos seguro que el DES contra algoritmos de fuerza bruta eligiendo el tamaño de clave apropiadamente.

El algoritmo está diseñado para reemplazar al DES.

Blowfish

Es un algoritmo de cifrado por bloques de 64 bits desarrollado por Scheiner. Es un algoritmo de tipo Feistel y cada rotación consiste en una permutación que depende de la clave y una sustitución que depende de la clave y los datos. Todas las operaciones se basan en o-exclusivas sobre palabras de 32 bits. La clave tiene tamaño variable (con un máximo de 448 bits) y se emplea para generar varios vectores de subclaves.

Este algoritmo se diseñó para máquinas de 32 bits y es considerablemente más rápido que el DES.

Cifrado asimétrico

La criptografía de clave pública fue inventada en 1975 por Whitfield Diffie y Matin Hellman. Se basa en emplear un par de claves distintas, una pública y otra privada. La idea fundamental es que las claves están ligadas matemáticamente pero es computacionalmente imposible obtener una a partir de la otra.

Este tipo de algoritmos tienen dos aplicaciones fundamentales:

1. **Encriptación**. Si un usuario A quiere mandar un mensaje a

otro usuario B, lo encripta usando la clave pública de B. Cuando B lo recibe lo desencripta usando su clave privada. Si alguien intercepta el mensaje no puede descifrarlo, ya que no conoce la clave privada de B (de hecho, ni tan siquiera A es capaz de desencriptar el mensaje).

2. **Firmas digitales**. Si B encripta un mensaje usando su clave privada cualquiera que tenga su clave pública podrá obtener el texto en claro correspondiente; si alguien quiere hacerse pasar por B tendrá que cifrar el mensaje usando la misma clave privada o no se descifrará correctamente con la clave pública de B. Lo que B ha hecho es firmar digitalmente el mensaje. El proceso de desencriptar con una clave pública un mensaje firmado se denomina verificación de firma.

El problema fundamental de este tipo de algoritmos es la distribución de las claves; aunque la clave pública se puede distribuir libremente (A la puede enviar por correo o decírsela a B por teléfono), nos queda el problema de la suplantación (C le puede dar su clave pública a B haciéndose pasar por A). Para solventar estos problemas se emplean autoridades certificadoras y certificados digitales.

Algoritmos de cifrado asimétrico:

RSA (Criptosistema RSA)

Es un sistema criptográfico de clave pública desarrollado en 1979, que utiliza factorización de números enteros. Es el primer y más utilizado algoritmo de este tipo y es válido tanto para cifrar como para firmar digitalmente.

El algoritmo a día de hoy es seguro, tiene una pequeña vulnerabilidad que sería explotable utilizando la potencia de la computación cuántica, permitiría en teoría realizar la descomposición del producto de dos número primos muy grandes de manera sencilla.

Más información

- https://www.jesusninoc.com/02/05/cifrar-y-descifrar-con-rsa-desde-java-generando-clave-privada-y-clave-publica/
- https://www.jesusninoc.com/02/05/cifrar-y-descifrar-con-rsa-utilizando-ecb-desde-java-generando-clave-privada-y-clave-publica/

Diffie-Hellman

El algoritmo de Diffie Hellman es un algoritmo de clave pública que permite el intercambio seguro de un secreto compartido. Generalmente se emplea junto con algoritmos de cifrado simétrico, como método para acordar una clave secreta. El algoritmo no se puede usar para encriptar conversaciones o firmas digitales.

Funciones resumen (Hash)

Un algoritmo de resumen de mensajes o función de dispersión criptográfica es aquel que toma como entrada un mensaje de longitud variable y produce un resumen de longitud fija. En inglés el resumen se llama message digest, digest o hash y el algoritmo message digest algorithm o one way hash algorithm.

Funciones de dispersión:

MD2, MD4 y MD5

Los tres son algoritmos de resumen de mensajes (el MD viene de Message Digest) desarrollados por Rivest.

Los tres toman un mensaje de longitud arbitraria y generan un resumen de 128 bits. El MD2 está optimizado para máquinas de 8 bits, mientras que el MD4 y MD5 son para arquitecturas de 32

bits. El código para los tres algoritmos se puede encontrar en los RFCs 1319, 1320 y 1321.

Hay una problemática que sufren las funciones de hash que son las colisiones y al igual que todos los algoritmos, también son susceptibles de sufrir un ataque de fuerza bruta. Para los algoritmos de tipo hash también hay los siguientes ataques: ataque de diccionario, ataque de cumpleaños, uso de rainbow tables.

SHA (Secure Hash Algorithm)

Los algoritmos de hash seguro1 son una familia de funciones de hash criptográficas publicadas por el Instituto Nacional de Estándares y Tecnología (NIST) como un estándar federal de procesamiento de información (FIPS) de EE. UU; que incluyen:

- SHA-0
- SHA-1
- SHA-2
- SHA-3

Conjunto de funciones hash criptográficas (SHA-224, SHA-256, SHA-384, SHA-512) diseñadas por la Agencia de Seguridad Nacional (NSA) y publicada en 2001 por el Instituto Nacional de Estándares y Tecnología (NIST) como un Estándar Federal de Procesamiento de la Información (FIPS).

Para SHA-0 y SHA-1 existen ataques típicos, que al igual que todos los algoritmos son susceptibles de sufrir ataque por fuerza bruta, además se añaden ataques de diccionario, ataque de cumpleaños (estos dos últimos ataques también se pueden aplicar para SHA-2).

Una investigación conjunta entre Google y el instituto holandés CWI ha conseguido desarrollar una técnica para generar dos ficheros PDF diferentes con el mismo hash SHA-1. La posibilidad de crear colisiones de hash, pone de manifiesto, según Google, la necesidad de dejar de utilizar el

algoritmo SHA-1 para el cálculo de hashes, ya que supondría poner en entredicho la validez del mismo para garantizar la autenticidad y la integridad de la información.

Más información

- https://www.jesusninoc.com/02/24/funcion-hash-sha1-sobre-un-fichero/
- https://www.jesusninoc.com/02/02/hash-a-password-with-sha-512/
- https://www.jesusninoc.com/02/17/ejercicios-de-seguridad-adivinar-un-hash-utilizando-un-fichero-con-hashes-generados/
- https://www.jesusninoc.com/02/24/poc-para-detectar-colision-en-sha1-con-powershell/

Códigos de autentificación de mensajes

Un código de autentificación de mensaje (message authentication code o MAC) es un bloque de datos de tamaño fijo que se envía con un mensaje para averiguar su origen e integridad. Son muy útiles para proporcionar autentificación e integridad sin confidencialidad. Para generar MACs se pueden usar algoritmos de clave secreta, de clave pública y algoritmos de resumen de mensajes.

Un tipo de MAC muy empleado en la actualidad es el código de autentificación de mensaje resumido (hashed message authentication code o HMAC). Lo que hacemos es generar el MAC aplicando una función de dispersión criptográfica a un conjunto formado por un mensaje y un código secreto.

Firma digital

La firma digital es un item que responde del origen e integridad de un mensaje. El que escribe un mensaje lo firma usando una clave de firmado y manda el mensaje y la firma digital. El destinatario usa una clave de verificación para comprobar el origen del mensaje y que no ha sido modificado durante el tránsito.

Una firma digital se compone de datos que permiten asegurar la identidad del firmante y la integridad. Los algoritmos que garantizan la integridad son algoritmos de resumen de mensajes y consisten en transformar mensajes de tamaño variable a textos cifrados de tamaño fijo sin emplear claves. Se emplean para convertir mensajes grandes en representaciones más manejables.

Es importante aclarar que firmar no es cifrar. Firmar significa garantizar nuestra identidad como autores del envío. Cifrar significa ocultar el contenido para que sólo el destinatario real del mensaje pueda leerlo.

Pasos para firmar:

1. El emisor genera un hash resumen del mensaje.
2. El hash se cifra con la clave privada del emisor y el resultado se conoce como firma digital que se envía adjunta al mensaje.
3. El emisor envía el mensaje junto con la firma digital al receptor, es decir el mensaje firmado.

El receptor qué hace con lo que ha recibido:

1. Separa el mensaje de la firma.
2. Genera un resumen del mensaje recibido usando la misma función hash que el emisor.
3. Descifra la firma con la clave pública del emisor obteniendo el hash original.

Si los resúmenes coinciden quiere decir que no se ha modificado durante la transmisión el mensaje. El sistema de firma digital descansa sobre un pilar fundamental: la autenticidad de la clave pública, para asegurar dicha autenticidad se recurre a las Autoridades de Certificación que aseguran que la firma es de quien dice ser.

Más información

- https://www.jesusninoc.com/02/02/hash-a-password-with-sha-512/
- https://www.jesusninoc.com/02/06/guardar-un-texto-junto-con-el-resumen-hash-sha-en-java/
- https://www.jesusninoc.com/02/06/comprobar-que-un-texto-no-se-ha-danado-o-manipulado-en-java-verificando-el-hash/
- https://www.jesusninoc.com/02/12/firmar-un-mensaje-y-comprobar-que-se-ha-firmado-correctamente-en-java/
- https://www.jesusninoc.com/02/12/firmar-y-verificar-archivos-java-archive-jar-con-jarsigner/
- https://www.jesusninoc.com/02/12/crear-un-certificado-y-firmar-un-script-ps1-desde-powershell/

Algoritmos de firmas digitales:

DSA y DSS

El DSA (Digital Signature Algorithm o Algoritmo Estándar de Firmado) es el algoritmo de firmado digital incluido en el DSS (Digital Signature Standard o Estándar de Firmas Digitales) del NIST Norteamericano. Se publicó en 1994.

El DSA está basado en el problema de los logaritmos discretos y sólo puede emplearse para las firmas digitales (a diferencia del RSA, que también puede emplearse para encriptar). La

elección de este algoritmo como estándar de firmado generó multitud de críticas: se pierde flexibilidad respecto al RSA (que además, ya era un estándar de hecho), la verificación de firmas es lenta, el proceso de elección fue poco claro y la versión original empleaba claves que lo hacían poco seguro.

DSA es un algoritmo lento que al contrario que RSA, no cifra, sino que realiza firmas.

Certificados digitales

Los certificados digitales son el equivalente digital del DNI, en lo que a la autentificación de individuos se refiere, ya que permiten que un individuo demuestre que es quien dice ser, es decir, que está en posesión de la clave secreta asociada a su certificado.

Un certificado de clave pública es un punto de unión entre la clave pública de una entidad y uno o más atributos referidos a su identidad. El certificado garantiza que la clave pública pertenece a la entidad identificada y que la entidad posee la correspondiente clave privada.

Los certificados de clave pública se denominan comúnmente Certificado Digital, ID Digital o simplemente certificado. La entidad identificada se denomina sujeto del certificado o subscriptor (si es una entidad legal como, por ejemplo, una persona).

Ejemplos

- https://www.jesusninoc.com/03/13/how-to-create-a-self-signed-certificate-with-powershell/
- https://www.jesusninoc.com/02/12/crear-un-certificado-y-firmar-un-script-ps1-desde-powershell/
- https://www.jesusninoc.com/02/12/crea-un-certificado-aut

ofirmado-para-firmar-un-fichero-script-o-lo-que-sea-desde-powershell/
- https://www.jesusninoc.com/03/13/crear-un-certificado-pfx-con-openssl-desde-powershell/
- https://www.jesusninoc.com/02/15/crear-un-certificado-en-windows-con-openssl-desde-powershell-powershell-exe/
- https://www.jesusninoc.com/02/12/firmar-y-verificar-archivos-java-archive-jar-con-jarsigner/
- https://www.jesusninoc.com/02/11/socket-ssl-en-java/#Crear_certificado_con_Keytool

---

Protocolos seguros de comunicaciones

Los datos que circulan a través de la red entre aplicaciones son accesibles por terceras personas ajenas a la comunicación, es necesario evitarlo a toda costa mediante el uso de protocolos seguros.

También si importante asegurarse de que los datos no se modifiquen durante el transporte. Los protocolos que aseguran la información a nivel de capa de transporte son SSL y TLS.

Los objetivos de los protocolos seguros son varios:

- Seguridad criptográfica. El protocolo se debe emplear para establecer una conexión segura entre dos partes.
- Interoperabilidad. Aplicaciones distintas deben poder intercambiar parámetros criptográficos sin necesidad de que ninguna de las dos conozca el código de la otra.
- Extensibilidad. El protocolo permite la incorporación de nuevos algoritmos criptográficos.
- Eficiencia. Los algoritmos criptográficos son costosos computacionalmente, por lo que el protocolo incluye un esquema de caché de sesiones para reducir el número de

sesiones que deben inicializarse desde cero (usando criptografía de clave pública).

El protocolo SSL

Originalmente diseñado por Netscape para establecer comunicaciones seguras con protocolos como HTTP o FTP. Permite negociar qué algoritmos se van a emplear, intercambiar las claves de encriptación y la autentificación de clientes y servidores.

Existen tres versiones del protocolo, la cuarta es una mejora del SSLv3 y se conoce con el nombre de TLS.

El protocolo TLS (Transport Layer Security)

Es una evolución del protocolo SSL (Secure Sockets Layer).

El protocolo SSL/TLS tiene multitud de aplicaciones en uso actualmente. La mayoría de ellas son versiones seguras de programas que emplean protocolos que no lo son. Hay versiones seguras de servidores y clientes de protocolos como el http, nntp, ldap, imap, pop3, etc.

Existen multitud de implementaciones del protocolo, tanto comerciales como de libre distribución. Una de las más populares es la biblioteca openssl, escrita en C y disponible bajo licencia GNU. Incluye todas las versiones del SSL y el TLS y un gran número de algoritmos criptográficos, algunos de los cuales ni tan sólo son empleados en el estándar TLS. La biblioteca está disponible en el URL http://www.openssl.org. En esa misma dirección se puede encontrar una lista de referencias a otras implementaciones gratuitas y comerciales de los protocolos SSL y TLS y aplicaciones que los emplean.

Ejercicios

Simular el funcionamiento de una VPN

- https://www.jesusninoc.com/01/24/ejercicios-de-seguridad-simular-el-funcionamiento-de-una-vpn-desde-powershell/

Simular el funcionamiento de un protocolo seguro con Cryptographic Message Syntax (TCP y UDP)

- https://www.jesusninoc.com/02/11/ejercicios-de-seguridad-realizar-una-comunicacion-tcp-segura-utilizando-cryptographic-message-syntax/
- https://www.jesusninoc.com/02/11/ejercicios-de-seguridad-realizar-una-comunicacion-udp-segura-utilizando-cryptographic-message-syntax/

Más información

- https://www.jesusninoc.com/11/06/creates-a-new-self-signed-certificate/
- https://www.jesusninoc.com/11/10/encrypts-content-by-using-the-cryptographic-message-syntax-format/
- https://www.jesusninoc.com/11/12/decrypts-content-that-has-been-encrypted-by-using-the-cryptographic-message-syntax-format/
- https://www.jesusninoc.com/11/18/exports-a-certificate-to-a-personal-information-exchange-pfx-file/
- https://www.jesusninoc.com/02/11/socket-ssl-en-java/

Programación de aplicaciones con comunicaciones seguras

Depende del sistema operativo y lenguaje de programación que

se utilicen.

Ejercicios

Sockets SSL en Java

- https://www.jesusninoc.com/02/11/socket-ssl-en-java/

Cifrar tráfico entre cliente y servidor mediante TLS desde PowerShell

- https://www.jesusninoc.com/03/11/cifrar-el-trafico-entre-un-cliente-y-un-servidor-mediante-el-protocolo-tcp-ip-con-un-certificado-x509-autofirmado-en-powershell/
- https://www.jesusninoc.com/03/12/enviar-una-cadena-segura-system-security-securestring-entre-un-cliente-y-un-servidor-cifrando-la-comunicacion-mediante-el-protocolo-tcp-ip-con-un-certificado-x509-autofirmado-en-powershell/
- https://www.jesusninoc.com/03/13/enviar-una-cadena-segura-system-security-securestring-entre-un-cliente-y-un-servidor-cifrando-la-comunicacion-mediante-el-protocolo-tcp-ip-desde-powershell-utilizando-un-certificado-x509-autofirmado/
