Anexo: Algoritmos criptográficos

26/11/2022

Introducción

Algoritmos sugeridos en la resolución 1699/22 del CIN

- Algoritmos de hash
  - SHA1
  - Familia de algoritmos SHA2: SHA-224, SHA-256, SHA-384, SHA-512, SHA-512/224, SHA-512/256.
  - Familia de algoritmos SHA3: SHA3-224, SHA3-256, SHA3-384, SHA3-512
  - SHAKE128 and SHAKE256
- Algoritmos de cifrado y descrifrado
  - AES
  - 3DES
- Algoritmos de firma digital
  - DSA
  - RSA
  - ECDSA

Problemas con estas sugerencias

- Varios de los algoritmos sugeridos **no deben usarse**.
  - SHA-1
  - 3DES
- En otros casos la gama de algoritmos sugeridos es muy amplia, y no queda claro cuál conviene usar.
- En los algoritmos de cifrado no se especifica el modo de operación, que es extremadamente importante.

Objetivo

- Recomendar algoritmos concretos.
- Considerar los casos de uso típicos de una universidad.
- Pueden no ser aplicables en otros casos de uso (p. ej. IoT).

Algoritmos de cifrado simétrico

Algoritmos de cifrado simétrico

- Se utilizan para el cifrado de mensajes y archivos.
- Requieren el uso de una clave secreta, conocida por el emisor y receptor.
- Suelen clasificarse en
  - *cifrados de bloques*
  - *cifrados de flujo*

Longitud de la clave

- Recomendada: 256 bits
- Nunca inferior a 128 bits.

Generación de la clave

La clave puede generarse

- A partir de la salida de un generador aleatorio de bits criptográficamente seguro.
- A partir de un protocolo de acuerdo de claves (*Key Agreement Protocol*)
- A partir de una contraseña mediante una función de derivación de claves basada en contraseña (PBKDF, *Password Based Key Derivation Function*).

Tamaño del bloque

- El tamaño más habitual es 128 bits.
- **No** deben utilizarse cifradores de bloques con bloques de 64 bits, tales como DES, Triple-DES, Blowfish o IDEA.

Modos de operación

- Los cifradores de bloques requieren el uso de un *modo de operación*.
- Usar modos de operación que brinden cifrado autenticado (AEAD: *Authenticated Encryption with Associated Data*):
  - GCM
  - OCB
  - CCM.
- Evitar los modos de operación de sólo confidencialidad:
  - CBC
  - CFB
  - OFB
  - CTR.
- **Nunca** utilizar ECB.

Recomendación

Sólo deben utilizarse algoritmos que permitan cifrado autenticado.

Algoritmos recomendados:

- ChaCha20-Poly1305
- AES-GCM (No cifrar mensajes de más de 64 GB).

No usar

- DES
- RC4

Algoritmos de cifrado asimétrico

Algoritmos de cifrado asimétrico

- También llamados de *clave pública* se utilizan como mecanismos de encapsulamiento de clave o para firma digital.
- **No** se utilizan para cifrar mensajes.
- Implican la existencia de un par de claves, una *clave pública* y una *clave privada*. La clave privada no puede derivarse de la clave pública.
- Por lo general, se basan en la existencia de cierto problema matemático de difícil resolución. En la actualidad, los algoritmos más difundidos están basados en *curvas elípticas*.

Firma digital

Firma digital

- En el caso de la firma digital, el marco normativo tiene más peso que el aspecto técnico.
- Desde el punto de vista técnico *no* es recomendable utilizar RSA, pero en algunos casos su uso será necesario.

Recomendación

Utilizar Ed25519.

Evitar

- RSA
- ECDSA

No utilizar

- DSA
- RSA-PKCS1v1.5

Longitud de clave

En el caso de tener que usar RSA o DSA, las claves no deben ser de longitud inferior a 2048 bits.

Intercambio de claves

Intercambio de claves

- Un algoritmo de intercambio de claves permite a dos corresponsales ponerse de acuerdo en una clave secreta mediante el intercambio de mensajes públicos.
- Se trata de una primitiva que no debería ser utilizada en forma directa, sino en el marco de un protocolo existente (como por ejemplo TLS).

Recomendación

Utilizar X25519

No usar

- Diffie-Hellman convencional
- ECDH no efímero

Cifrado asimétrico

Cifrado asimétrico

- Si bien no es posible cifrar un mensaje con un algoritmo asimétrico, es posible cifrar con un algoritmo simétrico con una clave aleatoria y encapsular la clave con un algoritmo asimétrico.
- Una combinación habitual es ChaCha20-Poly1305 combinado con Curve25519.

Recomendación

- Mediante el uso de una biblioteca
  - Usar `box()` de la biblioteca NaCl o `crypto_box()` de la biblioteca libsodium.
  - No utilizar bibliotecas de bajo nivel, como OpenSSL, Java, BouncyCastle, etc.
- Mediante el uso de una aplicación
  - Usar age.
  - Evitar el uso de GPG.

No usar

RSA

Funciones de hash

Funciones de hash

Las funciones de hash proporcionan integridad, y suelen formar parte de otros algoritmos (firma digital, derivación de claves, MAC).

Recomendación

Usar SHA-256

No usar

- MD5
- SHA-1

Códigos de autenticación de mensajes (MAC)

Códigos de autenticación de mensajes (MAC)

Se utilizan cuando se requiere autenticación bajo un secreto compartido, y no se requiere confidencialidad.

Recomendación

Usar HMAC-SHA-256.

No usar

- HMAC-SHA1
- HMAC-MD5

Gestión de contraseñas

Gestión de contraseñas

Se trata de algoritmos para almacenar contraseñas, o derivar claves a partir de contraseñas.

Recomendación

Usar, en el siguiente orden de preferencia:

1. scrypt
2. argon2
3. bcrypt
4. PBKDF2

No usar

Funciones de hash.

Seguridad de sitios web y aplicaciones cliente-servidor

Recomendación

Usar TLS 1.3, con OpenSSL y LetsEncrypt.

No usar

TLS 1.1 o anterior.

Generación de bits aleatorios

Generación de bits aleatorios

La mayor parte de los protocolos requieren la utilización de bits aleatorios con calidad criptográfica.

Recomendación

- Usar los servicios provistos por el sistema operativo.
  - En Unix/Linux: Utilizar `/dev/urandom` o llamadas al sistema como `getrandom()` o `getentropy()`.
  - En Windows, usar `BCryptGenRandom()`.
- Distintos lenguajes de programación proporcionan acceso a esos servicios. Por ejemplo, Java provee la clase `java.security.SecureRandom`, y Python el módulo `urandom`.

No usar

- Los generadores de números aleatorios estándares de los distintos lenguajes de programación. Estos suelen utilizar algoritmos que no tienen finalidades criptográficas, como por ejemplo Mersenne-Twister.
- `/dev/random`
- Generadores en espacio de usuario.
- El RNG de OpenSSL.