ALGORITMOS_CRIPTOGRAFICOS

Joan Pablo Alvarado Garfias | Ingeniería en Computación | 704

1. RSA (Algoritmo de Cifrado Asimétrico)

FUNCIÓN: Cifrar, descifrar y firmar digitalmente.

TIPO: Cifrado Asimétrico (Clave Pública/Privada). La seguridad se basa en la dificultad de factorizar números primos grandes.

FUNCIONAMIENTO BÁSICO:

- Generación de Claves: Se eligen dos números primos grandes (,). Se calcula el módulo y la función . Se determina el par de claves :
- Clave Pública: . Se usa para cifrar.
- Clave Privada: . Se usa para descifrar.
- Cifrado (Mensaje M a Cifrado C):
- Descifrado (Cifrado C a Mensaje M):

2. MD5 (Message-Digest Algorithm 5)

FUNCIÓN: Generar una huella digital única e irreversible (*hash*) para verificar la integridad de los datos.

TIPO: Función Hash Criptográfica. No es un algoritmo de cifrado (no es reversible).

FUNCIONAMIENTO BÁSICO:

- Entrada de Longitud Variable: Acepta datos de cualquier tamaño (texto, archivo, etc.).
- Procesamiento: La entrada se divide en bloques de 512 bits. Estos bloques se procesan a través de una serie de cuatro rondas de funciones no lineales y transformaciones bit a bit.
- Salida de Longitud Fija: El resultado es un valor *hash* de 128 bits (representado típicamente como una cadena hexadecimal de 32 caracteres).
- Propiedad Clave: Un cambio mínimo en la entrada resulta en un *hash* completamente diferente. MD5 es vulnerable a colisiones y no debe usarse para aplicaciones de seguridad críticas (como firmas digitales).

3. Base64

FUNCIÓN: Codificar datos binarios en una cadena de texto ASCII imprimible. Permite la transferencia de datos binarios a través de sistemas orientados a texto (como correo electrónico o URLs).

TIPO: Esquema de Codificación Binario-a-Texto. No es un algoritmo de cifrado (no oculta ni protege la información).

FUNCIONAMIENTO BÁSICO:

- Agrupación de Bits: Los datos de entrada se agrupan en bloques de 3 bytes (24 bits).
- División y Mapeo: Cada bloque de 24 bits se divide en cuatro grupos de 6 bits.
- Índice de Caracteres: Cada grupo de 6 bits se mapea a un carácter de un conjunto de 64 caracteres imprimibles (letras mayúsculas, minúsculas, números, + y /).
- Relleno (*Padding*): Si la entrada no es múltiplo de 3 bytes, se utiliza el carácter = para rellenar la salida hasta que sea múltiplo de 4 caracteres.
- Resultado: El resultado codificado es aproximadamente 33% más grande que los datos binarios originales.