¿Como funciona el cifrado cesar?

- Cambia las posiciones del alfabeto:
Las transformacion se menciona como ROT-N.

Ej: ROT-N

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

→ se realiza el commiento N, hacia la letra necesaria, hacia la derecha. desde donde se inicia la letra.

alfabetos posibles para base 32.

- a-z, sin ñ + 0 al 5
- a-z, con ñ + 0 al 4
- a-z, sin ñ + A, B, C, D, E, F

} 1 - 32.

alfabeto ingles base 34 → [a-z][0,7]

- Para descifrar es al revez.

Vigenere

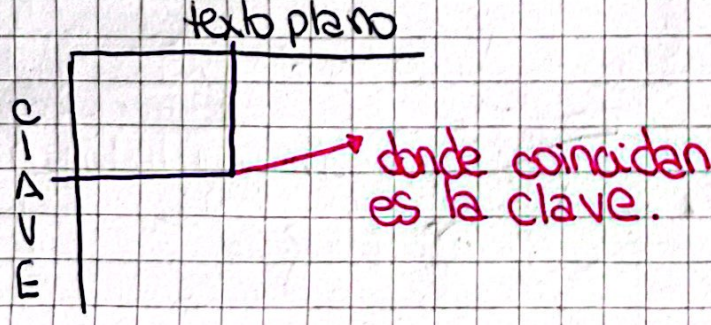
→ Funcionamiento:

- se debe tener una clave y una palabra a decifrar.

ej: MEGA → clave
CRIPTOGRAMAS RESUELTOS → palabra;

se resuelve de la sgte forma:

Criptogramas Resueltos / Palabra
MECAMECAMECA MECAMECAM / clave.



RAIL FENCE O CEASAR BOX

- metodo vulnerable ya que los caracteres permanecen en su estado original, este metodo es vulnerable a un posible ataque de analisis de frecuencias y de anagramas.

Ej: texto dado: HOLA MUNDO
trasposicion: H L M N O
O A U D
texto cifrado: HLMNOOAUD.

BASE 64

- metodo de codificar binario dentro de textos.
- cada 3 bloques de 8 bits son guardados en 4 caracteres de 6 bits, donde los caracteres se encuentran en el rango de [A-Z, a-z, 0-9, +/]{1}.
- signo (=) es el relleno de los bloques.

Ej: Man ⟶ TWFu
codificadas en ASCII son.
M, a y n ⟶ 77, 97, 110.
esto en binarios ⟶ 01001101 = 77
01100001 = 97
01101110 = 110

se unen en un ciclo:
01001101 01100001 01101110

(luego se toman bloques de 6 bits. (forman como max 64)

- luego se cambian a 4 numeros y cl se convierte a su valor en base 64.

Procedimiento final:

texto entrada	M		a		n	
ASCII	77		97		110	
bits	010011	01	011000	01	011011	10
indice (6 bits)	19	22		5		46
RESULTADO (BASE64)	T	W		F		u

Cuando no son exactos, los caracteres "=" se utilizan para rellenar.

sscode01.

• Entropía

- La entropía de un passcode es la cantidad de variación de caracteres dentro de éste.

Se calcula como:

$$H = L \cdot \log_2(W)$$

en donde W es la base utilizada,
L es el largo del string.

- NIST estima que un passcode debería tener un mínimo de 112 bits de entropía.

• El centro nacional de Seguridad Cibernética del Reino Unido (NCSC) recomienda usar tres palabras que sean simples, aleatorias, y sencillas de recordar.

• Microsoft dice que las contraseñas y su complejidad ya no importan, la autentificación multifactor puede evitar el 99,9% de los hackeos

→ Control de acceso

La autentificación tiene que ver con la identidad del usuario, mientras que la autorización tiene que ver con los privilegios del usuario.

SAML → Security Access Markup Language.
IAM → Identity and Access Management
Idp → Identity Provider
└ servicio que almacena y verifica la identidad de los usuarios. Los IdP suelen ser servicios alojados en la nube y a menudo trabajan con proveedores de inicio de sesión único (SSO) para autentificar a los usuarios.

PSD2 → Payment Services Directive.
└ Estándar europeo para pagos.
→ Permite que terceras empresas, también intervengan en los pagos. Conocido como open banking.
→ Establece como obligatorio el 2FA.

SCA → Strong Customer Authentification
↳ obliga a authentificar a la persona que realiza el pago mediante tres maneras posibles, de las que se tienen que asegurar dos de ellas

- tecnicas de Two Factor Authentication
  - ↳ SMS; tecnica mas facil y economica.
  - → Rich Communcation Services (RCS)
  - → FIDO (Fast IDentity Online)
  - → Clave dinamica.
  - → Titan Security Key:
    - ↳ El gran punto fuerte es el chip de seguridad Titan que cuenta con un firmware creado especificamente por Google para comprobar la integridad de las llaves y que la URL a la que accedemos sea valida.

2. Lectura complementaria

Nota: Imágenes y textos (leves modificaciones) tomados de:

- *https://ayudaleyprotecciondatos.es/2020/06/10/cifrado-cesar/ Página consultada el 5 de agosto de 2022.*
- *https://marquesfernandes.com/es/tecnologia-es/que-y-base64-para-que-serve-y-como-funciona/ Página consultada el 5 de agosto del 2022.*

2.1. Cifrado de Cesar

El cifrado de César, también conocido como cifrado por desplazamiento. Es un tipo de cifrado por sustitución en el que una letra en el texto original es reemplazada por otra letra que se encuentra un número fijo de posiciones más adelante en el alfabeto. Por ejemplo:

- Si se desea cifrar el texto "CASA" mediante un ROT-2 (algoritmo de desplazamiento), el resultado del texto cifrado sería "ECUC". Consulte la siguiente ilustración, para mayor comprensión:

Figura 1: Ejemplo cifrado de cesar (Fuente: https://justcodeit.io)

2.2. Cifrado Vigenère

El cifrado Vigenère es un cifrado basado en diferentes series de caracteres o letras del cifrado César formando estos caracteres una tabla, llamada tabla de Vigenère, que se usa como clave. La siguiente imagen representa la tabla de Vigenère considerando como base el alfabeto inglés.

3.1. Funcionamiento del algoritmo

Para comprender el funcionamiento del algoritmo base64 se empleará una codificación a la palabra **PIPE** como ejemplo:

1. Primero encontraremos en nuestra tabla ASCII el código binario correspondiente de cada letra:

   P:01010000 I:01001001 P:01010000 E:01000101

2. Ahora contamos todos los códigos binarios y los dividimos en grupos de 6 bits:

   010100 000100 100101 010000 010001 010000

   En este ejemplo, faltan 4 dígitos en nuestro último bloque, agregaremos cuatro ceros (0000) a su derecha para completar el grupo de 6 bits.

3. Un punto muy importante en tener en cuenta es que necesitamos grupos completos de 24 bits, si su grupo no se completa, se agrega padding (=). En este caso en particular tenemos un grupo de 24 bits y otro de 12 bits, por lo cual se agregaran 2 padding (=).

4. Ahora que tenemos los grupos de 6 bits, expresaremos los grupos en decimal para asociarlos a la tabla ASCII:

   20 4 37 16 17 16 = =

5. A continuación, buscamos nuestra referencia decimal en la tabla de conversión Base64:

   U E I Q R Q = =

Value	Encoding	Value	Encoding	Value	Encoding	Value	Encoding
0	A	17	R	34	i	51	z
1	B	18	S	35	j	52	0
2	C	19	T	36	k	53	1
3	D	20	U	37	l	54	2
4	E	21	V	38	m	55	3
5	F	22	W	39	n	56	4
6	G	23	X	40	o	57	5
7	H	24	Y	41	p	58	6
8	I	25	Z	42	q	59	7
9	J	26	a	43	r	60	8
10	K	27	b	44	s	61	9
11	L	28	c	45	t	62	+
12	M	29	d	46	u	63	/
13	N	30	e	47	v		
14	O	31	f	48	w	(pad)	=
15	P	32	g	49	x		
16	Q	33	h	50	y		

Figura 3: Tabla base64

2. Lectura complementaria

Nota: Imágenes y textos con leves modificaciones tomados de:

- *https://latam.kaspersky.com/blog/que-es-un-hash-y-como-funciona/2806/ Página consultada el 9 de septiembre del 2022.*
- *https://www.comparitech.com/blog/information-security/what-is-a-collision-attack/ Página consultada el 9 de septiembre del 2022.*

2.1. Función de hash

Una función de hash, o función resumen, es un algoritmo matemático que transforma cualquier bloque arbitrario de datos en una nueva serie de caracteres con una longitud fija. Es decir, que independientemente del largo del string en la entrada, el valor de hash tendrá siempre al misma longitud de salida.

Algunas de sus características son las siguientes:

- **Las funciones hash son deterministas :** cuando la misma entrada se ejecuta a través de una función hash dada, siempre da como resultado exactamente el mismo resultado, cada vez.
- **Las funciones hash son unidireccionales :** es fácil tomar una entrada, ejecutarla a través de la función hash y luego averiguar cuál es el hash coincidente para esa entrada. Sin embargo, no es factible tomar un hash y luego averiguar cuál fue la entrada original del hash. Por 'unidireccional', queremos decir que solo es práctico calcular una función hash en una dirección, no en la otra.
- **No es factible encontrar dos entradas diferentes que den como resultado el mismo resultado :** para garantizar la seguridad en las aplicaciones de las funciones hash, no debe ser factible que un atacante encuentre dos entradas diferentes que produzcan el mismo hash. Cuando se encuentran dos entradas con el mismo hash, se denomina **colisión**

2.2. Problema de cumpleaños

- ¿Cuál es la probabilidad que una persona tenga la misma fecha de cumpleaños que uno?
  $p = \frac{1}{365}$
- Si tengo n personas. ¿Cuántos pares distintos de personas puede generar?
  $\frac{n*(n-1)}{2}$
- La probabilidad que al menos dos personas tengan el mismo día de cumpleaños.
  $1 - (1 - \frac{1}{365})^{(\frac{n*(n-1)}{2})}$

2.3. Ataques a funciones hash

- **Collision attack:** Encuentra 2 mensajes con el mismo hash, es decir implica tomar una entrada determinada y encontrar otra entrada donde ambas entradas den como resultado el mismo hash.
- **Pre-image attack:** Los ataques de preimagen están relacionados con los ataques de colisión, pero implican tratar de encontrar mensajes que den como resultado hashes específicos. En términos muy sencillos implica tomar un hash determinado y luego descubrir la entrada inicial que lo produjo.

2.4. Efecto avalancha

El efecto avalancha se produce cuando al mas mínimo cambio sobre la palabra en texto claro, cambia completamente el hash producido por ella.

2.5. CRC

El CRC o comprobación de redundancia cíclica (CRC) es un código de detección de error. CRC valida los paquetes de información representantes por los mecanismos y los verifica con los datos extraídos, lo que garantiza su exactitud. Características:

- Los bits representan los coeficientes de un polinomio de orden (k-1).
- Donde k es el largo de bits del frame a transmitir
- Suponga un frame 110001 de largo k=6.
- Su polinomio es:
  - $1 \cdot x^5 + 1 \cdot x^4 + 0 \cdot x^3 + 0 \cdot x^2 + 0 \cdot x^1 + 1 \cdot x^0$
  - $x^5 + x^4 + 1$
- Cuando se emplea el método de código polinomial, el emisor y el receptor deben acordar por adelantado un polinomio generador, G(x).
- Para calcular la suma de verificación para una trama con m bits, correspondiente al polinomio M(x), la trama debe ser más larga que el polinomio generador, es decir $len(M(X)) > len(G(x))$

Los pasos a seguir para resolver ejercicios asociados a CRC son:

1. Convertir el polinomio generador a un polinomio binario.
2. Calcular el valor del residuo.
3. Resolver la división, mediante la compuerta XOR.

Diferencias entre MD5 y SHA256.

- SHA256 y MD5 son algoritmos de hash utilizados para producir un resumen unico de datos (HASH).

diferencias:

- **Longitud:** SHA256 produce hash 256 bits.
  MD5 " " 128 bits
- **seguridad:** SHA256 es mas seguro que MD5 en cuanto a resistencia de colisiones y ataques criptograficos.
  MD5 ha demostrado ser vulnerable en cuanto a las colisiones.
- **velocidad:** MD5 es mas rapido de calcular que SHA256.
- **Aplicaciones:**
  - MD5 se utiliza en aplicaciones donde la velocidad es crucial y no la seguridad. ej: comprobar archivos.
  - SHA256 se utiliza en aplicaciones donde se requiere una seguridad robusta. ej: firmas digitales.

Bcrypt:

- Algoritmo de hashing diseñado especificamente para el almacenamiento seguro de contraseñas.
- Es mas lento pero es resistente a ataques por fuerza bruta o diccionario.
- caracteristicas importantes.
  - tiene un parametro de factor de costo ajustable para calcular la cantidad de trabajo necesario para calcular el hash.
  - salt aleatorio, cadena de sal aleatorea unica para cada contraseña hasheada.
- funcionamiento: utiliza un cifrado lento y costoso que se adapta mediante el ajuste de un factor costo, lo que lo hace resistente a los ataques de fuerza ~~prego~~ bruta.

MD5 funciona → toma una entrada de cualquier longitud y produce un hash de 128 bits.
SHA256 funciona → " y produce un hash de ~~128~~ 1256 bits.

comando sed: expresion de sustitucion que se utiliza para sustituir
↳ • se utiliza para realizar operaciones de edicion de texto.

comando 's/$/x/x'
- s: operacion por sustituir
- /: representa el final de la linea
- x: reemplaza 1x al final
- x: vuelve a reemplazar otra x al final.

↳ sed 's/$/x/x' hola → holaxx.

HASH:

- Una función de hash toma un mensaje de cualquier tamaño M y retorna un valor de largo fijo h.
- El mensaje es un flujo de bits
- $h = H(M)$

→ Una función de hash no debe ser reversible:
→ Dado M, es facil calcular h.
→ Dado h, es difícil encontrar un M tal que $H(M) = h$
→ Dado M, es dificil encontrar otro mensaje M' tal que $H(M) = H(M')$.
→ Debe ser resistente a colisiones.
- Es difícil encontrar dos mensajes aleatoreos, M y M' tal que $H(M) = H(M')$.

→ CRC: Codigos de Redundancia Cíclica.

- verifica la integridad de los datos de una transmisión.
- se basa en el principio de que n bits de datos se pueden considerar como los coeficientes de un polinomio.
  Ej: [1 1 0 1] ; $x^3 + x^2 + 1$.

- Procedimiento:

1. Se añaden r bits 0s a la derecha del mensaje (tantos ceros como grados tenga el polinomio generador).
2. se divide el polinomio obtenido por el polinomio G.
3. La eleccion del polinomio generador es escencial si queremos detectar la mayoría de los errores que ocurran.

4. Al comprobar el CRC, solo si el residuo de división es cero, entonces la comunicación estará libre de errores.

- Existen colisiones en CRC.

Lan Manager (LM) / NTLM

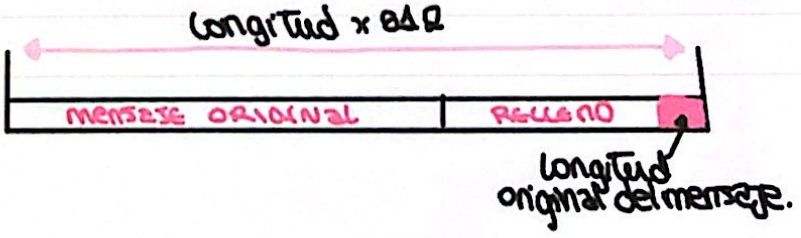
- la contraseña se rellena con ceros o se trunca a 14 caracteres.
- se divide en 2 mitades de 7 caracteres.
- se convierte en una cadena binaria para usar como dos claves DES, insertando un cero después de cada 7 bits.
- Los dos textos cifrados resultantes se concatenan para formar el hash LM de 16 bytes.

MD5: (4 variables, hash 128)

- procesa el mensaje en bloques de 512 bits que se divide en 16 sub-bloques 32 bits c/u.
- Como resultado son 4 números de 32 bits que se concatenan para entregar un valor de hash de 128 bits.
- El mensaje original se expande para tener un largo multiplo de 512 bits - 24.
- El relleno es un bit en 1 seguido de bits en 0.
- finalmente se agregan 64 bits que representan el largo del mensaje, dejando un largo en un multiplo de 512 bits.

→ añadiendo bits.

- se añade un bit 1 y luego tantos bits 0 como sean necesarios para cumplir la condicion(M + Relleno) mod 512 = 448.
- trailer 64 bits.
- 1 a 512 bits de relleno.



→ Inicializando el buffer:

- buffer de 4 registros de 32 bits c/u y se representan como A, B, C, D en formato hexadecimal. (A = 0x01234567)

→ Procesando el mensaje:

- 4 funciones auxiliares, (F, G, H e I), con 3 entradas de 32 bits c/u, (X, Y, Z), obteniendo como salida una palabra de 32 bits. teniendo 4 funciones no-lineales son:

$F(X,Y,Z) = (X \wedge Y) \vee ((\neg X) \wedge Z)$; $G(X,Y,Z) = (X \wedge Z) \vee (Y \wedge (\neg Z))$

$H(X,Y,Z) = (X \otimes Y \otimes Z)$; $I(X,Y,Z) = Y \otimes (X \vee (\neg Z))$.

- Al finalizar los ciclos, quedan en los buffer, A, B, C y D y el resultado final del proceso.
- Como los buffer son de 32 bits cada uno, entonces obtenemos el hash de 128 bits del mensaje M.

→ efecto avalancha

- un cambio pequeño en el mensaje, nos entrega un mensaje totalmente distinto.

SHA (Security Hash Algorithm)

- version antigua SHA-0, version nueva SHA-1.
- SHA-1 es un estandar para USA y esta definida en la RFC 3174.

→ ¿Como funciona?

- 5 variables (A, B, C, D, E).
- Genera un hash de 160 bits.
- mas resistente a ataques de fuerza bruta., pero es mas lento.

RIPEMD-160 (comunidad europea)

- ISO/IEC 10118:2003 dedicated hash-functions.
→ Para cada una de estas 7 funciones hash, especifica un metodo de padding, valores de inicializacion, parámetros, secuencia de funciones a utilizar en las rondas constantes.

SHA-512

- SHA-512/224 y SHA-512/226 truncan a 224 y 256 bits
- SHA-512 es mas rapido que SHA-256 en las CPU de 64 bits.

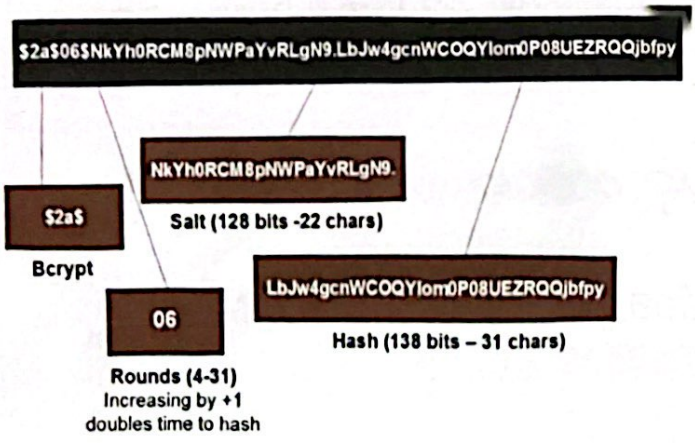
SHA-3

- requerimientos minimos:
→ hashes de 224, 256, 384, 512 bits.
→ tamaño mensaje $2^{64}-1$ bits
- puntos a considerar: costo en HW, velocidad, energia.

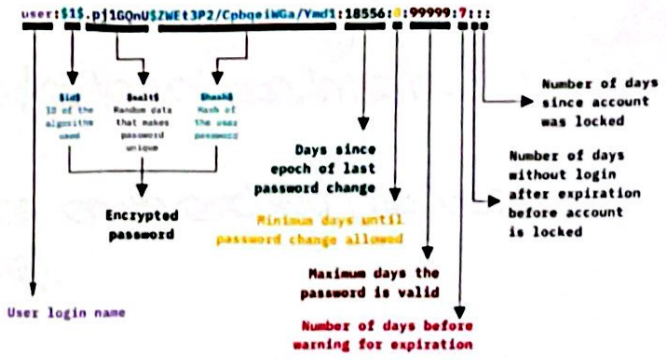
Blowfish Crypt

- los algoritmos pueden correr hasta 100 veces mas rapido en GPU vs CPU, pero bcrypt son iguales o mas lentos en GPU vs CPU.

Bcrypt.



mkpasswd -m sha-512 criptografia seguridad



Razonando la seguridad

→ input secreto (pepper) cuando se almacenan passwords.
→ se guarda por separado en algún otro medio, como un modulo de seguridad de HW.

Key Derivation Function

→ toma una password y genera n bits que pueden ser usados como una llave de cifrado.

Light Weight Cryptography Environment

SHAKE (Security Hash Algorithm and KECCAK).
→ proporciona un recurso, eficiente con el consumo de energía, ya que no requiere grandes recursos físicos ni logicos.

Hashes para otros escenarios

imphash

→ tecnica en la que los valores hash se calculan basandose en los nombres de las bibliotecas/funciones importadas y su orden particular dentro del ejecutable. Si los archivos se compilaron a partir del mismo codigo fuente y de la misma manera, esos archivos tenderan a tener el mismo valor imphash.

humanhash

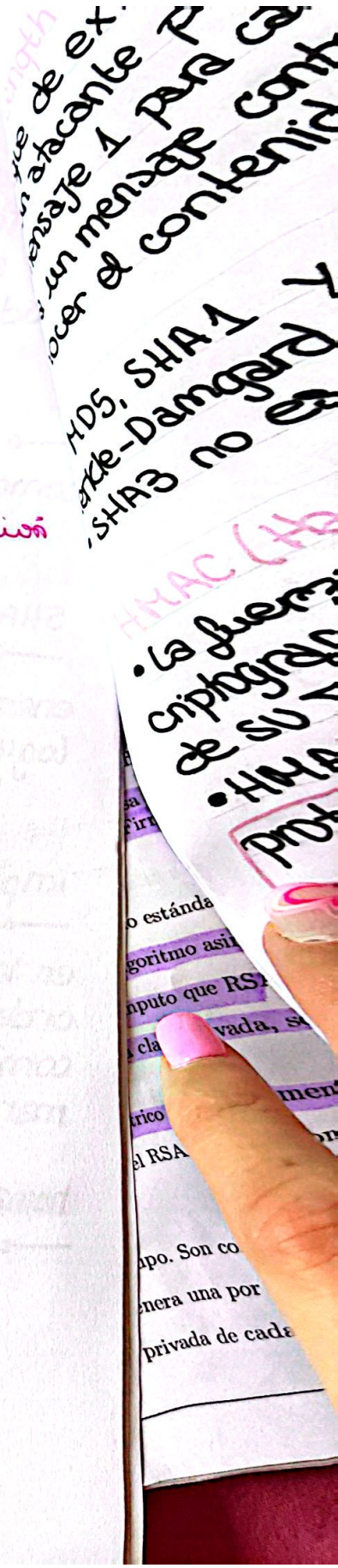
→ facilita que un usuario pueda leer un hash.

fuzzy hashing → evidencian cambios en un archivo (no usan efecto avalancha).

HMAC: Algoritmo de Authentification.

- utiliza
  - funcion hash
  - clave secreta
  - autenticidad de mensaje.
- funcionamiento:

1. se combina el mensaje con la clave secreta utilizando una funcion de combinacion perfecta.
2. se aplica una funcion hash al resultado de la combinacion.
3. el resultado se utiliza como un codigo de authentificacion que se envio junto con el mensaje
4. se realiza lo mismo (receptor) para obtener un codigo de autentificacion
5. Si los codigos coinciden, el mensaje es autentico.



Length Extension Attack

- Ataque de extensión de longitud es un tipo de ataque en el que un atacante puede utilizar hash(mensaje 1) y la longitud del mensaje 1 para calcular hash(mensaje 1 || mensaje 2) para un mensaje controlado por el atacante, sin necesidad de conocer el contenido del mensaje 1.

- MD5, SHA1 y SHA2 estan basados en la construccion de Merkle-Damgard son susceptibles a este tipo de ataque.
- SHA3 no es susceptible

HMAC (Hash-based message authentication code).

- La fuerza criptografica del HMAC depende de la potencia criptografica de la funcion de hash subyacente, el tamaño de su salida de hash y el tamaño y calidad de la llave.
- HMAC-SHA1 y HMAC-MD5 se han utilizado dentro de los protocolos IPSEC y TLS.
  - → ipad es el inner padding / opad es el outer padding.

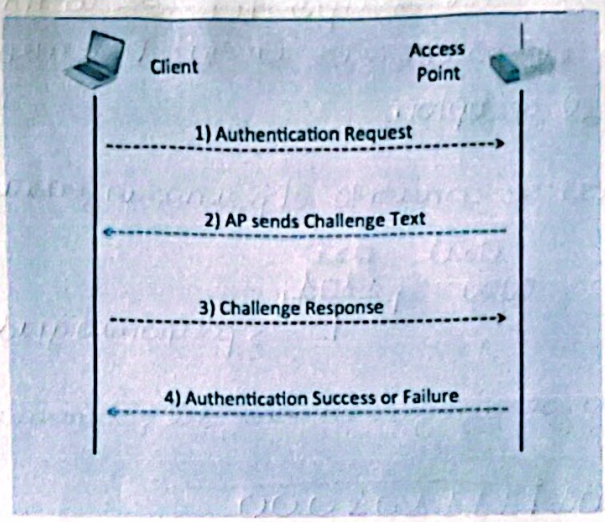
Challenge-Response Authentication Mechanism.

- CRAM-MD5 basado en HMAC-MD5
- Da soporte a Simple Authentication and Security Layer, que es usado en la autentificacion de protocolos como SMTP, POP, IMAP, LDAP, etc.
- Se utiliza para evitar el envio de contraseñas en texto plano.

TEMA

CRAM-MD5 CHALLENGE:

1. El cliente solicita la autenticación
2. El servidor envía un desafío en base64 al cliente.
3. El cliente convierte la cadena a hexadecimal y calcula HMAC-MD5 sobre el mensaje, a partir de la contraseña del usuario. Luego concatena el nombre de usuario y un espacio al mensaje. Finalmente, convierte todo a base64 y se le envía el mensaje al server.
4. El servidor utiliza el mismo método para calcular la respuesta esperada. Si la respuesta dada y la respuesta esperada coinciden, la autenticación se ha realizado con éxito.



DIGEST-MD5

- es mas fuerte, ya que CRAM-MD5 solo incluye datos aleatorios del servidor mientras que DIGEST-MD5 incluye datos aleatorios tanto al cliente como el servidor.

lookup table / rainbow table

- Son tablas formadas por una dupla de palabras, donde cada dupla ha sido formada por la accion de 40 mil veces una funcion de reduccion y resumen.

CRC

Un equipo emisor desea enviar la cadena de datos M(x) = 111101 a un equipo receptor. Para la detección de errores de transmisión, en la construcción del campo FCS de la trama, se usara el algoritmo CRC usando el polinomio 0x0D. Determine la secuencia del FCS que sera enviado al receptor.

→ 1° paso: se convierte el polinomio generador G(x):

0x0 0xD
0000 1101 → polinomio grado 3., $x^3 + x^2 + 1$.

→ 2° paso: se agregan los ceros del polinomio, dependiendo el grado del polinomio

```
1101 | 1111010000
     - 1101
       0010011
         1101
       - ----
         01000
           1101
         - ----
           01010
             1101
           - ----
             01110
               1101
             - ----
               0011
```

M(x) = 111101011, CRC es 0011 o 0x03

[implica que el transmisor envia el sgte tren de datos]

1. INTRODUCCIÓN

FIRMA

La grafoscopía es una disciplina que se centra en la identificación y autenticación de documentos y firmas.

La firma digitalizada es la conversión del trazo de una firma en una imagen. La firma digitalizada se considera firma electrónica simple, pero no ofrecen ninguna garantía respecto a la identidad del firmante. Además, las firmas digitalizadas se pueden falsificar muy fácilmente. Paradójicamente este tipo de firmas son las más utilizadas por la mayoría de las personas para firmar, y dar su consentimiento, en muchos documentos y contratos. Permite identificar al firmante en un documento electronico incluyendo el aspecto gráfico de la firma minuscrita.

Tipos de firmas electrónicas (equivalentes electronico de las firmas manuscritas) que existen:

- La firma electrónica o firma electrónica simple FES
- La firma electrónica avanzada FEA

Estos tipos de firmas electrónicas se diferencian principalmente por sus distintos niveles de seguridad, por su capacidad de garantizar, o no, la integridad de los documentos que se firman y por su capacidad de identificar, o no, al firmante.

La firma digital, mecanismo criptografico que permite al receptor de un mensaje firmado digitalmente, identificar la entidad originada de dicho mensaje y confirman que el mensaje no ha sido alterado desde que fue firmado por el originador. En cuanto a sus propiedades se puede decir que son unicas, infalsificables, verificables, innegables, viables.

Clasificaciones:

- **Seguridad de dispositivos fisicos:** dispositivo que se dice que es resistente a modificaciones. Para esto, el firmante tiene una smart card que puede ser solo cifrar con una secret key k1 y cada verificador tiene una smart card que puede solo descifrar con una secret key k2 de forma que lo cifrado en k1 solo se puede ser verificado por k2. k1 y k2 pueden ser iguales o distintas.
- **Criptografia de clave simetrica:** Esquemas basados en funcion de un solo sentido. Tienen como desventaja el tamañao de las key, de las firmas y del hecho que pueden ser usados un numero fijo de veces. Algunos ejemplos son: firmas de Desmedt, firmas de lamport-diffie, simetric key robin.
- **Criptografia de clave asimetrica:** Se basa en el concepto de funciones de un solo sentido con trampa. Los mas importantes son RSA, DSA, ESING, Firmas con curvas elipticas.

DSS: Digital Signature Standar

- Sistema de firma digital adoptado como estándar por la organización de estándares de los EE.UU. (NIST).
- Utiliza la función hash SHA-1 y el algoritmo asimétrico DSA (Digital Signature Algorithm).
- Requiere mucho más tiempo de cómputo que RSA.
- El algoritmo DSA consiste en una clave privada, sólo conocida por el que envía el documento (el firmante), y una clave pública.
- El DSA es un algoritmo asimétrico que únicamente se puede utilizar con firma digital.
- Utiliza más parámetros que el RSA y así se consigue un grado de mayor seguridad ↗ DSA.

Parametros DSA

- Kg claves públicas de grupo. Son comunes y públicas para un grupo de usuarios.
- Ku clave pública. Se genera una por usuario a partir de las, Kg y es pública.
- Kp clave privada. Es privada de cada usuario, se genera a partir de las anteriores.

- k número aleatorio. Se genera uno para cada firma.
- s y r. Son dos palabras de 160 bits que forman la firma de un texto.
- El número k permite que el mismo texto del mismo usuario no genere siempre la misma firma.

Synchronizing Key Servers (SKS) es vulnerable a ataques de inundación de certificados como consecuencia de su diseño de solo escritura. Esto permite a un atacante envenenar certificados OpenPGP, añadiendo una gran cantidad de firmas "spam"que no pueden ser eliminadas. El envenenamiento hace que GnuPG no pueda importar certificados desde los servidores de claves SKS, provocando la denegación del servicio.

Para proteger la clave privada primero utilizan bcrypt para crear un hash de la contraseña, utilizando un salt generada aleatoriamente que difiere para cada usuario. El resultado se utiliza para encriptar la clave privada con AES-256.

CAI: Content Authenticity Initiative

Gracias a firmas, la información no puede ser modificada y de hacerlo aparecerá un aviso de que el contenido ha sido cambiado.

2. PGP: Pretty Good Privacy

Programa de encriptación privada y autenficación para la comunicacion de los datos, es usado para firmar, encriptar y descriptar textos, emails, archivos, directorios y algunas particiones de disco e incrementar la seguridad de la comunicación de e-mail.

2.1. Algoritmo

1. PGP genera solo una sola key de sesion aleatoria muy compleja de un solo uso que no se puede descifrar.
2. Esta clave de sesion se cifra utilizando la key publica del destinatario. Dicha clave se asocia a la persona , que puede compartirla con cualquiera de quien desea recibir mensajes.
3. El remitente transmite su clave de sesion y el destinatario puede utilizar su key privada para descifrar y leer el mensaje.

La arquitectura de PGP muestra la versión de PGP para poder interpretarlo. El crecimiento mas grande debido a la vulnerabilidades. Como EFAIL en el cifrado 2E2 que filtra el plano de correos cifrados. Y en 2019 vuelve a crecer por que Synchoronizing key servers (KSK) es vulnerable a ataques de inundación de certificados.

Fake PGP Signatures: Claves PGP que permitirian a un tercero acciones como leer y enviar correos con la identidad del desarrollador.

Aplicaciones de PGP en e-mail: Enigemail, GnuPG, Thunderbird.

Yubikey NEO: USB con NFC y MIFARE. Es compatible con las passwords de uso, la funcionalidad de las smart keys, incluyendo OpenPGP y PIV, ademas de U2F.

OpenPGP.js: Para proteger la key privada e usan bcrypt para crear el hash de password, utilizando un salt aleatoreo que cambia por usuario. El resultado se utiliza para encriptar la key privada con AES-256, esto es PRONTONMAIL.

Ejercicio 3

Usted está dando una ayudantía del ramo de criptografía y seguridad en redes, y le preguntan mencionar las ventajas y desventajas, junto con usos específicos de los diferentes tipos de firmas electrónicas. (Al menos).

3.1 Opción de respuesta

En el ámbito de la criptografía y la seguridad en redes, las firmas electrónicas se han convertido en herramientas fundamentales para garantizar la autenticidad, integridad y no repudio de documentos digitales.

Firma electrónica simple (FES)

Ventajas

- **Facilidad de uso:** No requiere de software o hardware especializado.
- **Bajo costo:** Su implementación es relativamente económica.
- **Útil para transacciones de bajo riesgo:** Sirve para identificar al firmante y verificar la integridad del documento.

Desventajas

- **Bajo nivel de seguridad:** No garantiza el no repudio del firmante.
- **Susceptible a falsificaciones:** Puede ser replicada con mayor facilidad.
- **Validez limitada:** Su aceptación legal puede estar restringida a ciertos ámbitos.

Casos de uso

- Firmar correos electrónicos personales.
- Autorizar documentos internos de baja importancia.
- Agilizar trámites administrativos de bajo riesgo.

Firma electrónica avanzada (FEA)

Ventajas

- **Alto nivel de seguridad:** Brinda mayor garantía de no repudio e integridad del documento.
- **Validez legal:** Es reconocida legalmente en la mayoría de los países.
- **Admite firma de documentos complejos:** Permite firmar contratos, escrituras y otros documentos relevantes.

Desventajas

- **Mayor costo:** Requiere de un certificado digital y un dispositivo seguro para su uso.
- **Complejidad técnica:** Su implementación puede ser más compleja que la FES.
- **Menos accesible:** No todos los usuarios cuentan con los medios para obtener una FEA.

Casos de uso

- Firmar contratos electrónicos.
- Realizar trámites gubernamentales en línea.
- Presentar declaraciones de impuestos.
- Participar en licitaciones públicas.
- Firmar documentos notariales.

Anexos

Firma Electrónica

La Firma Electrónica es un equivalente digital de la firma manuscrita que permite a las personas y organizaciones firmar documentos electrónicos de manera segura y confiable. Se basa en la criptografía para garantizar la autenticidad, integridad y no repudio de los documentos firmados.

¿Cómo funciona la Firma Electrónica?

El funcionamiento de la Firma Electrónica se basa en un certificado digital y una clave criptográfica privada. El certificado digital es un documento electrónico que vincula la identidad del titular a la clave criptográfica privada. La clave privada se utiliza para generar la firma electrónica, mientras que el certificado digital se utiliza para verificar la validez de la firma.

Beneficios de la Firma Electrónica

- **Seguridad:** Garantiza la autenticidad, integridad y no repudio de los documentos firmados.
- **Comodidad:** Permite realizar trámites en línea sin necesidad de desplazarse presencialmente.
- **Agilidad:** Agiliza los procesos administrativos y comerciales.
- **Legalidad:** Tiene el mismo valor legal que una firma manuscrita.
- **Eficiencia:** Reduce el uso de papel y los costos asociados.

Firma Electrónica Simple (FES) → clave criptografica.

La Firma Electrónica Simple (FES) es un tipo de firma electrónica que ofrece un nivel de seguridad menor que la Firma Electrónica Avanzada (FEA), pero es más fácil de obtener y usar. Se basa en un certificado digital que vincula la identidad de una persona a una clave criptográfica, la cual se utiliza para firmar documentos electrónicos.

Firma Electrónica Avanzada (FEA) → clave criptografica privada

La Firma Electrónica Avanzada (FEA) es un tipo de firma electrónica que cumple con los más altos estándares de seguridad y confiabilidad. Se basa en un certificado digital que vincula la identidad de una persona a una clave criptográfica privada, la cual se utiliza para firmar documentos electrónicos.

Firma Digitalizada

En el ámbito de la firma electrónica, el término "Firma Digitalizada" se utiliza comúnmente para referirse a la representación digital de la firma manuscrita del titular. En otras palabras, es la imagen escaneada de la firma autógrafa que se incorpora a un documento electrónico para asociarlo a la identidad del firmante.

Diferenciación entre Firma Digitalizada y Firma Electrónica

Es importante destacar que la Firma Digitalizada no es lo mismo que la Firma Electrónica. La Firma Electrónica, como se ha explicado anteriormente, se basa en criptografía y ofrece un nivel de seguridad superior, mientras que la Firma Digitalizada no cuenta con las mismas garantías.

Limitaciones de la Firma Digitalizada

Debido a su naturaleza, la Firma Digitalizada presenta algunas limitaciones en cuanto a su seguridad y confiabilidad:

- **Falta de autenticación:** No garantiza que la firma haya sido efectivamente realizada por el titular.
- **Susceptibilidad a falsificaciones:** Puede ser fácilmente falsificada mediante técnicas de edición de imágenes.
- **Falta de integridad:** No asegura que el documento no haya sido modificado desde que se firmó.
- **No repudio:** No permite al firmante negar haber firmado el documento.

caciones de la Firma Digitalizada

A pesar de sus limitaciones, la Firma Digitalizada sigue siendo utilizada en algunos casos, principalmente cuando se requiere una representación visual de la firma manuscrita para fines de formalidad o costumbre. Entre sus aplicaciones comunes se encuentran:

- **Contratos:** En algunos casos, se puede utilizar la Firma Digitalizada para firmar contratos electrónicos, siempre que las partes involucradas acuerden su validez y se implementen medidas adicionales de seguridad.
- **Documentos internos:** La Firma Digitalizada puede ser utilizada para firmar documentos internos de una organización que no requieran un alto nivel de seguridad.

Firma Digital

En el contexto de las transacciones electrónicas, la Firma Digital, también conocida como Firma Electrónica Avanzada (FEA), se ha convertido en una herramienta fundamental para garantizar la seguridad, confiabilidad y autenticidad de los documentos digitales. A diferencia de la firma manuscrita tradicional, la Firma Digital se basa en criptografía para vincular la identidad del firmante al contenido del documento firmado, proporcionando un nivel de protección superior contra falsificaciones, manipulaciones y repudio.

Componentes clave de la Firma Digital

- **Certificado Digital:** Documento electrónico que vincula la identidad del titular a una clave criptográfica única.
- **Clave Privada:** Clave criptográfica secreta conocida únicamente por el titular del certificado digital, utilizada para generar la firma digital.
- **Clave Pública:** Clave criptográfica asociada a la clave privada, accesible públicamente en el certificado digital, utilizada para verificar la validez de la firma digital.

Funcionamiento de la Firma Digital

- **Firmado:** El firmante utiliza su clave privada para generar una firma digital única para el documento electrónico.
- **Adición de la firma:** La firma digital se adjunta al documento electrónico, creando un paquete firmado digitalmente.
- **Verificación:** El destinatario del documento utiliza la clave pública del firmante (disponible en el certificado digital) para verificar la autenticidad e integridad de la firma digital y el documento.

Obtención de la Firma Digital

La Firma Digital se puede obtener a través de Prestadores de Servicios de Certificación (PSC) acreditados por la Subsecretaría de Economía y Empresas de Menor Tamaño. El proceso de obtención generalmente implica:

- **Solicitud de certificado digital:** El solicitante presenta su identificación personal y completa un formulario de solicitud.
- **Validación de identidad:** El PSC verifica la identidad del solicitante presencialmente o a través de medios electrónicos confiables.
- **Emisión del certificado digital:** El PSC emite el certificado digital y entrega al solicitante la clave privada y las instrucciones de uso.

Características de la Firmas

- **Autenticidad:** Garantiza que la firma proviene del titular del certificado y no ha sido falsificada.
- **Integridad:** Asegura que el contenido del documento no ha sido modificado desde que fue firmado.
- **No repudio:** Permite al firmante no negar haber firmado el documento.

- cifre ZZZ, utilizando padding ISO/IEC 7816-4., función expansión y relleno 64 bits.

→ fórmulas $L_{i+1} = R_i$, $R_{i+1} = L_i \otimes F(R_i \otimes K_i)$ } encriptar / $R_i = L_{i+1}$, $L_i = R_{i+1} \otimes F(L_{i+1} \otimes K_i)$ } desencriptar

→ S-Box ⟹ entran 6 y salen 4
↳ osea dividido en 6 bits, 8 cajas.

→ Padding ISO 0x80 y puros 0 después.
↳ 4 bits.

RONDA 1:
entonces cifrar ZZZ, con clave 0xFFFFFFFFFFFFFFFF } 16·4 = 64 bits.
entonces mensaje = "ZZZ"
0x5A5A5A8000000000 } siempre 64 bits.
Luego el mensaje se divide en 2:

0x5A5A5A80 | 00000000
L_i | R_i

después XOR entre R_i y K agregando padding para lograr los 48 bits.

0x000000008000 → 48 bits → 4·12 = 48
⊗ 0xFFFFFFFFFFFF
0xFFFFFFFF7FFF
se divide

$2^3 2^2 2^1 2^0$
8 = 1 0 0 0
F = 1 1 1 1
7 = 0 1 1 1

0xFF	0xFF	0xFF	0xFF	0x7F	0xFF
11111111	11111111	11111111	11111111	01111111	11111111

como input:

S1	S2	S3	S4	S5	S6	S7	S8
111111	111111	111111	111111	111111	110111	111111	111111

como output

S1	S2	S3	S4	S5	S6	S7	S8
1111	1111	1111	1111	1111	1101	1111	1111

lo entregamos a hexadecimal

0xffffffdff

y se hace el ⊗ entre 0xFFFFFDFF
0x5A5A5A80
0xA5A5A77F

$2^3 2^2 2^1 2^0$
5 = 0 1 0 1
F = 1 1 1 1
A = 1 0 1 0

D = 1101
A = 1010
7 = 0111

A = 1 0 1 0
F = 1 1 1 1
5 = 0 1 0 1

Prime ronda deja como mensaje 0x00000000A5A5A77F

RONDA 2:

Se toma el mensaje:

0x00000000	A5A5A77F
Li	Ri

XOR entre:

$$\begin{array}{r} \overbrace{\text{0xA5A5A77F8000}}^{48} \\ \text{0xFFFFFFFFFFFF} \\ \hline \text{0x5A5A5880 7FFF} \end{array}$$

8 4 2 1
$2^3\ 2^2\ 2^1\ 2^0$

A = 1010
F = 1111
S = 0101

S = 0101
F = 1111
A = 1010

0x5A	0x5A	0x58	0x80	0x7F	0xFF
0101 10/10	0101/1010	01/011000	100000/00	0111/1111	11/11 1111

Luego dividimos con el input de 6 bits en S-Box.

S1	S2	S3	S4	S5	S6	S7	S8
010110	100101	101001	011000	100000	000111	111111	111111

Luego el output con 4 bits.

S1	S2	S3	S4	S5	S6	S7	S8
0101	1001	1010	0110	1000	0001	1111	1111

0x59A681FF

Luego un XOR entre

$$\begin{array}{r} \text{0x59A681FF} \\ \text{0x00000000} \\ \hline \text{0x59A681FF} \end{array}$$

dejando como resultado de la segunda: 0x 59A681FF A5A5A77F