GUÍA 3: MESSAGE AUTHENTICATION CODES (MAC) Y FUNCIONES DE HASH

Ejercicio 1: Analizar por qué no poseen seguridad los siguientes MAC:

1) $Mac_k(m) = G(k) \oplus m$, donde $G(x)$ es un generador seudoaleatorio.
2) $Mac_k(m) = k \oplus first\_k\_bits(m)$
3) $Mac_k(m) = Enc_k(|m|)$.

Ejercicio 2: Considerar el siguiente algoritmo de código de autenticación de mensaje (MAC).

1. El mensaje m es dividido en bloques de 128 bits cada uno (completando con bits en cero si es necesario)
2. Se efectúa un XOR entre todos los bloques, obteniendo un único bloque de resultado R de 128 bits. La función MAC se aplica luego al bloque R, en lugar de aplicarse a m.

a) Describir por lo menos una manera de encontrar dos mensajes $m \neq m'$ tales que los mensajes aún con diferentes significados tienen el mismo valor de MAC.

b) Implementar el algoritmo y probarlo para la situación propuesta en (a)

Ejercicio 3: Considerar la siguiente función de hash h():

- La función acepta mensajes de cualquier longitud.
- La función retorna una cadena de bits de 32 ceros si la entrada tiene un número par de caracteres, y retorna una cadena de bits de 32 unos si la entrada tiene un número impar de caracteres.

a) ¿es esta una función de hash válida para criptografía? Analizar los tres niveles de seguridad.

b) ¿cuál es la probabilidad de que dos entradas x_1 y x_2 elegidas aleatoriamente colisionen en h()?

Ejercicio 4: (lectura de teoría)

a) Describir la construcción **CBC-MAC** e indicar para qué sirve.

b) Comparar **CBC-MAC** con **CBC-mode** para encripción.

c) ¿Cuáles son, según Katz, las opciones seguras de uso de CBC-MAC?

d) ¿Para qué sirve la "Transformación de Merkle-Darmgard"?

Ejercicio 5: Rutinas de Hashing en OpenSSL

Con el comando dgst se puede obtener el hash (**digesto** o **resumen**) de un mensaje. Luego se puede usar, entre otras cosas, para firma digital.

```
openssl dgst [-sha|-sha1|-mdc2|-ripemd160|-sha224|-sha256|-sha384|-
sha512|-md2|-md4|-md5|-dss1] [-c] [-d] [-hex] [-binary] [-r] [-non-fips-
allow] [-out filename] [-sign filename] [-keyform arg] [-passin arg] [-
verify filename] [-prverify filename] [-signature filename] [-hmac key]
[-non-fips-allow] [-fips-fingerprint] [file...]
```

a) Calcular el hash MD5 de una frase (por ejemplo "hoy es el primer lunes de abril")
b) Calcular el hash SHA-1 de la misma frase.
c) ¿Qué diferencias se observan?

Ejercicio 6:

Se tienen los nombres de alumnos y el hash de sus notas. Decir cuál es la nota de cada alumno. Tener en cuenta que la nota se colocó en números y en letras, en la forma X

nota_en_letras (por ejemplo, 3 tres). Las notas son valores enteros, del 1 al 10. Se usó el algoritmo SHA1:

```
acuña:        1daae8480ce1df09603d3db5388b900e8ce4b880
centurion:    164c22fd426d4215fc47d38964de80100a24f5ff
hernandez:    135fc9d048e923597cc806a51ebdcb1ccac553bf
palacios:     c736e54648efc18698499026ba1779e7785378a2
rossi:        c2fa01c8fdf749547317e985625f2512b2c4e0a6
sanchez:      7c1dfd9e7a101bc419752f623aa2c09352cac070
garcía:       86a76e0399c99c1d5b8c8751b7d5240b24b271f3
zubeldia:     c736e54648efc18698499026ba1779e7785378a2
```