Funciones de Hash en Criptografia

Valentina S. Vispo

Octubre 2023

Índice

1 Funciones de Hash

1.1 De que se trata

Las funciones de hash criptograficas son probablemente las herramientas mas versatiles de todo el arsenal criptografico, y forman parte de infinidad de protocolos

1.1.1 Propiedades

Una funcion de hash es una funcion h que tiene, como minimo, las siguientes propiedades"

1. **Compresion**: h mapea una entrada x de longitud finita arbitraria a una salida $h(x)$ de longitud fija n
2. **Resistencia de preimagen**: Si $H = F(x)$, debe ser dificil, conociendo H y F, averiguar X.
   (a) Dificil de invertir
   (b) Si lograr obtener un texto claro, no es posible de manera directa obtener otros
3. **Resistencia de segunda preimagen**: Si el archivo es modificado produciendo un $X'! = X$, la funcion debe producir $H' = F(X')! = H$. Debe ser dificil, conociendo H, X y F, encontrar un X' tal que $F(X) = F(X')$
   (a) Importante para firmas digitales
   (b) *Resistencia a colisiones para firmas digitales*
      i. Si firmas algo creado por otra persona, la resistencia a segunda preimagen no te protege — Es mas facil generar dos documentos con el mismo hash, conociendo el x.
4. **Resistencia a colisiones**: es dificil encontrar x. x' tal que $x! = x'$ y $h(x) = h(x')$.

1.1.2 Problemas

1. Si se conoce como es la funcion de hash, son facilmente quebrables
2. Almacenamiento de passwords:
   (a) texto claro: si se obtiene acceso a la base de datos, la informacion esta comprometida de manera directa. Si ocurre un ataque interno, ocurre lo mismo.
   (b) cifradas: para cifrar y descifrar requiere una clave, entonces debe estar almacenada, siendo inseguro

1.1.3 Tipos de ataque a las funciones hash

Ataque de preimagen:

- El atacante debe generar multiples mensajes y probar la funcion de hash con cada uno de ellos.
- Para lograrlo se necesita en promedio generar 2^n mensajes.

- Una funcion de hash de 128 bit ya es seguro para el ataque de preimagen. Deberian generarse 2^128 mensajes.

Ataques de colision:

- Mucho mas simple de realizar que los ataques de preimagen.

- Para lograrlo se debe encontrar una colision, y estas requieren en promedio $2^{\frac{n}{2}}$ operaciones.

- Una funcion de hash de 256 bit ya es seguroo para el ataque de preimagen, para que demoren 2^128 operaciones.

- Paradoja del cumplea;os

- Enunciado: Cuantas personas tiene que haber en un grupo para que exista la probabilidad del 50

- Respuesta: 23

- Lo que ocurre esque si hay N personjes hay $\frac{N\cdot(N-1)}{2}$ pares de personas, es decir, del orden N^2

Ataques sobre la funcion de compresion: - El atacante utiliza criptoanalisis diferencial (ataque probabilistico)

- Ataques de multibloque

- Los mensajes que colisionan son:

$M = M_1||M_2$, $M' = M_1'||M_2'$

Ataques de prefijo elegido: - Permiten construir colisiones a partir de dos prejifos distintos elegidor por el atacante

- Es un ataque mas poderoso, permite lograr colisiones agregando bits a mensajes existentes

- Puede extenderse paralograr multicolisiones (Nostradamus attack)

2 Desafio: Segunda preimagen de una función de hash

2.1 Enunciado

El desafío consiste en encontrar una segunda preimagen al resultado de aplicar una función de hash de 24 bits a su dirección de correo electrónico. Usaremos una función que consiste en aplicar SHA-256 y tomar los primeros N bits del resultado. Llamaremos a esta función SHA-256-N.

Cada byte (8 bits) está representado por 2 caracteres hexadecimales, por lo que SHA-256-24 tiene una salida de 6 caracteres, y SHA-256-48 una de 12.

Observaciones

- Para encontrar una preimagen, el esfuerzo necesario es del orden de 2^n, donde n es la longitud del hash. En este caso $n = 24$, por lo que se requiere calcular unos pocos millones de hashes.
- El cálculo no debería tomar más de unas decenas de segundos en una computadora con un procesador reciente.

2.2 Analisis

2.3 Solucion

Notas a tener en cuenta

1. La respuesta será una secuencia de bytes codificada en base64
2. La respuesta contiene los cinco textos cifrados, uno por línea, y codificados individualmente en base64

Pasos

1.

2.3.1 Como correr la solución

python3 challenges/6_Second_Preimage.py

2.4 Herramientas

2.5 Referencias

1. Enunciado ejercicio