Implementación del Hash del cuco en la encriptación de contraseñas para una base de datos

Anthony Bautista[1], Cristopher García[2], Rosa Limachi[3] y Julio Rosales[4]
[1 2 3] Ciencias de la Computación [4] Matemática
Universidad Nacional de Ingeniería, Lima, Perú.
{[1]abautistal, [3]rlimachip}@uni.pe {[2]crisebas100, [4]julio845}@gmail.com

***Resumen*—En este informe se propone una solución particular al problema de la seguridad de las contraseñas en una base de datos, empleando el lenguaje python para la implementación y el algoritmo de hash del cuco para la encriptación.**

I. Introducción

Como administrador de una institución siempre surge el problema de identificar quién y cómo se usa los recursos del sistema. Una solución de este problema sería la creación de un sistema de autenticación de usuarios donde se genera y almacena un nombre de usuario y su respectiva contraseña. Aunque esta solución es válida sigue habiendo el problema de la suplantación de usuarios por el acceso no autorizado de las contraseñas en la base de datos donde se almacenaba. Por ello se tendría que encriptar las contraseñas de una forma que no se pueda obtener la real a partir de la encriptada.

I-A. OBJETIVO

- Encriptación de contraseñas de una base de datos a partir del algoritmo de hash del cuco.

I-B. DEFINICIONES

- Función Hash: Función que a partir de una entrada que suele ser una cadena, genera una salida (cadena) de longitud fija, esta tiene información.
  La colisión hash se produce cuando entradas distintas generan el mismo valor en una función hash.
- Tabla Hash: Estructura de datos que relaciona claves y valores para cada elemento que guarda. Utilizaremos una función hash para transformar la clave de un dato e identificar el lugar que ocupará en la tabla.
- Colisiones Hash: Situación donde al aplicar una función hash a dos entradas distintas generan igual valor.
- Hash del Cuco: Algoritmo que permite resolver las colisiones hash usando dos funciones hash en vez de una.

II. ESTADO DEL ARTE

- Hashing: Técnicas y Hash para la Protección de Datos
  Se compara las diversas técnicas de hashing, donde cada una de estas puede presentar colisiones, con un costo computacional alto. Además, muestra sus diversos usos como en el cifrado de contraseñas, en creación de certificados digitales, cuando ocurre un ataque de base de datos .[1]
- Algoritmo hash y vulnerabilidad a ataques
  Explica el problemas del ciframiento de los datos al aplicar encriptación por hash, donde se propone como solución emplear dos algoritmos como el SHA-1 y RIPEND-160.[2]

III. DISEÑO DEL EXPERIMENTO

El experimento se realizara con el lenguaje python creando 4 módulos:

1. Main: Viene a ser el módulo principal donde se muestra las opciones de registro, validación de usuario y salir. Basta con ejecutar esta función para que el programa funcione.
2. Datos: Almacena las funciones que permiten saber si el usuario que se registra existe, así como validar su contraseña.
3. Cuckoo: Convierte la contraseña ingresada en una sucesión de caracteres, donde al aplicar el algoritmo hash del cuckoo, como en el Algoritmo 2 , permite desordenarla impidiendo su reconocimiento aun obteniendo la fuente donde se almacena las contraseñas.

4. Recepcion_datos: Permite el almacenamiento de usuario y contraseñas en archivos cvs, encriptando la contraseña mediante el módulo Cuckoo. Además, de verificar si el usuario con su respectiva contraseña existen.

Algorithm 1: funcion_hash(funcion,clave)

```
if cont == n then
    aux.append(clave)
for j = 0 to ver do
    suma = suma + ord(i)
switch funcion do
    case 1 do
        returnsuma %n
    case 2 do
        returnn − (suma %n) − 1
```

Algorithm 2: Colocar(clave, tabla, cont, n)

```
if cont == n then
    aux.append(clave)
for j = 0 to ver do
    pos[i] = funcion_hash(i + 1, clave)
    if hashTable[tabla][pos[tabla]]! = 0 then
        save = hashTable[tabla][pos[tabla]]
        hashTable[tabla][pos[tabla]] = clave
        colocar(save, (tabla + 1) %ver, cont + 1, n)
    else
        hashTable[tabla][pos[tabla]] = clave
```

IV. PRUEBA

Ver en el cuaderno de jupyter en el siguiente enlace: https://github.com/crisebas/Cuckoo-Hashing/blob/master/experimentacion.ipynb

REFERENCIAS

[1] http://www.laccei.org/LACCEI2018-Lima/student_Papers/SP96.pdf
[2] http://www.revistasbolivianas.org.bo/scielo.php?pid=s1997-40442009000200026&script=sci_arttext