**Tablas rainbow y su uso**

**Pablo Menéndez Suárez**

*UO252406@uniovi.es*

**Resumen**

(Máximo 200 palabras)

1. **Introducción**

A la hora de diseñar una aplicación que requiera de autentificación siempre se plantea un problema cuando se necesita almacenar las contraseñas de los usuarios en una base de datos. Éstas pueden ser almacenadas de varias formas, bien sea en texto plano (nada recomendable), cifradas con un algoritmo bidireccional o cifradas mediante un algoritmo unidireccional dando como resultado un “Hash”. Ésta última es la más usada en sistemas seguros y también la más recomendable., ya que invertir el proceso de cifrado es muy difícil y la única manera de obtener las contraseñas es mediante la fuerza bruta. En este trabajo vamos a ver como apoyándonos en las tablas rainbow, podemos realizar fuerza bruta de una forma mucho más rápida y eficiente.

1. **Tablas Rainbow**
2. **Definición**

Las Tablas Rainbow, son un tipo de tablas pregeneradas utilizadas para la obtención de contraseñas por medio de fuerza bruta que nos permiten ahorrar tiempo de ejecución sacrificando espacio en disco y memoria. Este tipo de tablas almacenan una relación de pares de palabras en texto plano donde, por un lado está la palabra inicial y por otro lado la palabra final. El vínculo formado entre estas dos palabras viene dado por el uso de ambas en la función de resumen y reducción que representa la Tabla Rainbow. Existen diferentes tipos de Tablas Rainbow para cada algoritmo de cifrado, generalmente ocupan muchos gigas o incluso teras. Antes de pasar a hablar del funcionamiento de estas tablas y para comprender bien lo que acabamos de leer, debemos tener claros algunos conceptos.

1. **Función Hash o de resumen**

Una función hash o de resumen es una función que aplica una fórmula matemática a una entrada, normalmente un texto plano, dando como resultado una salida alfanumérica de longitud fija, conocida como hash, que solo puede ser obtenida a través de esa misma entrada.

1. **Función de reducción**

Una función de reducción es una función que transforma un hash en una cadena de texto plano. Es muy importante no interpretar mal este concepto. La función de reducción no es el proceso contrario a una función hash, es decir, aplicar una función de reducción sobre un hash no te va a devolver su cadena original, si no que te va a devolver otra cadena cualquiera, ya que precisamente el objetivo de las funciones hash es que no se pueda realizar una función inversa.

1. **Funcionamiento**
2. **Creación de Tablas Rainbow**

Anteriormente hemos dicho que las Tablas Rainbow almacenan pares de palabras en forma de palabra inicial y palabra final. Pues bien, para la creación de dichas tablas se parte de una palabra inicial, que es un texto plano, y se le realiza una función de resumen obteniendo así su hash. Al hash que se obtiene se le aplica una función de reducción, dando lugar a otra palabra en texto plano a la cual se le volverá a aplicar una función de reducción y así sucesivamente. Este proceso finaliza cuando se ha realizado unas 40.000 veces y lo único que se almacena es la palabra inicial y la final. De esta manera rápidamente nos podemos dar cuenta del enorme ahorro de memoria que se produce, ya que pasa de almacenar 40.000 palabras a únicamente la de entrada y salida. Una vez obtenido el primer par de palabras este proceso se realizará con otra entrada distinta y así sucesivamente tantas veces como se desee. Una vez obtenidos todos los pares de palabras, se ordenan en orden alfabético de acuerdo al valor final, permitiendo así realizar búsquedas binarias que optimicen el trabajo. Cabe destacar que la tabla solo se genera una única vez y una vez generada ya se puede utilizar para la ruptura de múltiples hash.

1. **Algoritmo de las Tablas Rainbow**

Ahora que ya sabemos como se crea una Tabla Rainbow, vamos a ver como ésta realizaría el proceso para la obtención de una clave mediante un hash. El algoritmo que siguen estas tablas es el siguiente:

* 1. Se realiza una función de reducción sobre el hash que tenemos.
  2. Se comprueba si la palabra obtenida se encuentra entre las palabras finales de la tabla. En caso afirmativo saltamos al paso cuarto.
  3. En caso de que la palabra obtenida no se encuentre entre los valores finales de la tabla se le aplica la función de resumen y volvemos al primer paso.

1. **Cositis**
2. **Conclusión**
3. **Referencias**

Apuntes