**Data Structure**

Catedrático: Fernandojosé Boiton Tello

Catedrático auxiliar: Luis Angel Tórtola Tejeda

Sharon Anesveth A. Maatens (20190339)

**Hash Function**

El problema que busca solucionar una Hash Function es acelerar la búsqueda. Considere el problema de buscar un array para un valor dado. Si el array no está ordenado, la búsqueda puede requerir examinar todos y cada uno de los elementos del array. Si se ordena el array, podemos usar la búsqueda binaria y, por lo tanto, reducir la complejidad del tiempo de ejecución en el peor de los casos a O(log n).

Podríamos buscar aún más rápido si conocemos de antemano el índice en el que se encuentra ese valor en el array. Supongamos que tenemos esa función mágica que nos dirá el índice de un valor dado. Con esta función mágica, nuestra búsqueda se reduce a una sola sonda, lo que nos da un tiempo de ejecución constante O (1). Dicha función se llama función hash. Una función hash es una función que, cuando se le da una clave, genera una dirección en la tabla, conocida como Hash. Un hash generalmente se muestra como un número hexadecimal.

Una función hash que devuelve un número hash único se denomina función hash universal. En la práctica, es extremadamente difícil asignar números únicos a los objetos. Esto último siempre es posible solo si conoce (o aproxima) el número de objetos a procesar.

Por lo tanto, decimos que nuestra función hash tiene las siguientes propiedades:

* Siempre devuelve un número para un objeto.
* Dos objetos iguales siempre tendrán el mismo número
* Dos objetos desiguales no siempre tienen números diferentes

1. **MD5**

El algoritmo de resumen de mensajes MD5 procesa datos en bloques de 512 bits, divididos en 16 palabras compuestas de 32 bits cada una. La salida de MD5 es un valor de resumen de mensaje de 128 bits.

El cálculo del valor de resumen MD5 se realiza en etapas separadas que procesan cada bloque de datos de 512 bits junto con el valor calculado en la etapa anterior. La primera etapa comienza con los valores de resumen del mensaje inicializados utilizando valores numéricos hexadecimales consecutivos. Cada etapa incluye cuatro pases de resumen de mensajes que manipulan los valores en el bloque de datos actual y los valores procesados ​​desde el bloque anterior. El valor final calculado a partir del último bloque se convierte en el resumen MD5 para ese bloque.

**APLICACIONES:** Verificacion de datos:Los resúmenes MD5 se han utilizado ampliamente en el mundo del software para garantizar la integridad del archivo transferido. Por ejemplo, los servidores de archivos a menudo proporcionan una suma de comprobación MD5 precalculada para los archivos, de modo que un usuario pueda comparar la suma de comprobación del archivo descargado.

**VENTAJAS:** Es más fácil comparar y almacenar estos hashes más pequeños que almacenar un texto grande de longitud variable ademas de que es muy fácil generar un resumen del mensaje original usando este algoritmo.

**DESVENTAJAS:** Vulnerable a ataques: especialmente uno que generó colisiones de hash en un minuto o menos en un portátil estándar y otro que podría generar una colisión en tan solo 10 segundos en un sistema Pentium 4 de 2,66 GHz. Como resultado, el IETF sugirió que los nuevos diseños de protocolo no deberían usar MD5 en absoluto, y que los recientes ataques de investigación contra el algoritmo "han proporcionado razones suficientes para eliminar el uso de MD5 en aplicaciones donde se requiere resistencia a la colisión, como las firmas digitales".

1. **Whirlpool**

WHIRLPOOL es una función de hash diseñada por Vincent Rijmen y Paulo S. L. M. Barreto. El hash ha sido recomendado por el proyecto NESSIE y ha sido adoptado por la Organización Internacional de Estandarización (ISO) y la Comisión Electrotécnica Internacional (IEC) como parte del estándar internacional ISO/IEC 10118-3.

WHIRLPOOL fue desarrollado después del algoritmo de cifrado por bloques Square. Es una construcción Miyaguchi-Preneel basada en una modificación del Advanced Encryption Standard (AES). Dado un mensaje de un tamaño menor de 2^256 bits, devuelve un hash de 512 bits.

Los autores han declarado que no está patentado (ni lo estará). Puede ser usado libremente para cualquier propósito y las implementaciones de referencia son de dominio público.

El nombre del algoritmo es en referencia a la Galaxia Whirlpool en la constelación Canes Venatici.

**APLICACIONES:** Password Storage y Data Integrity Check, ademas de otras aplicaciones criptografica con un input de un bloque de data variable.

**VENTAJAS:** Se considera de las más seguridas al dia de hoy por la Organización Internacional de Estandarización (ISO) y la Comisión Electrotécnica Internacional (IEC)

**DESVENTAJAS:** Incluso un pequeño cambio hará que (con una probabilidad de 1-10^(-154)) el resultado de la función sea completamente distinto

1. **SHA-1**

El algoritmo de hash seguro es una función de hash criptográfica diseñada por la NSA de los Estados Unidos. SHA-0 (publicado en 1993) se ha visto comprometido hace muchos años. SHA-1 (1995) produce un valor hash de 160 bits (20 bytes). Por lo general, se representa como un número hexadecimal de 40 dígitos.

Algo así como el 90% de los sitios web que usan encriptación SSL (bloqueo verde) usan un algoritmo llamado SHA-1 para protegerse de ser suplantados. Esto garantiza que cuando vayas a green lock para facebook.com, estés visitando el Facebook real y no le des tu contraseña a un atacante.

Cuando su navegador ve un certificado, puede calcular el SHA-1 para la información de ese certificado y luego compararlo con el SHA-1 firmado que el certificado ofreció como prueba. Debido a que SHA-1 promete “slugs” únicos, el navegador confía en que si coinciden, el certificado en oferta es el mismo que la Autoridad de Certificación firmó.

**APLICACIONES:** Cifrar la conexión del usuario al sitio web y verificar que se haya conectado al sitio web real.

**VENTAJAS:** SHA-1 es realmente rápido (puede hacer millones de hashes en un segundo) y también se puede paralelizar masivamente fácilmente en GPU

**DESVENTAJAS:** Muy debil ante ataques de fuerza bruta: En 2005, los criptógrafos demostraron que SHA-1 podría ser descifrado 2,000 veces más rápido de lo previsto. Su verdadera "muerte" ocurrió en 2010 cuando muchas organizaciones comenzaron a recomendar su reemplazo.

Los tres grandes, Microsoft, Google y Mozilla, dejaron de aceptar certificados SSL SHA-1 en 2017 en sus navegadores, después de múltiples ataques exitosos.

1. **SHA-2**

SHA-2, desarrollado por la Agencia de Seguridad Nacional (NSA), es una función hash criptográfica. SHA-2 incluye cambios significativos de su predecesor, SHA-1. La familia SHA-2 consta de seis funciones hash con resúmenes (valores hash) que son 224, 256, 384 o 512 bits: SHA-224, SHA-256, SHA-384, SHA-512, SHA-512/224, SHA -512/256.

**APLICACIONES:** Cifrar la conexión del usuario al sitio web y verificar que se haya conectado al sitio web real.

**VENTAJAS:** SHA-2 es mucho más complicado y todavía se considera seguro.

**DESVENTAJAS:** SHA-2 comparte la misma estructura y operaciones matemáticas que su predecesor (SHA-1), por lo que es probable que se vea comprometido en el futuro cercano.

1. **RIPEMD**

RIPEMD (RACE Integrity Primitives Evaluation Message Digest) es un grupo de funciones hash desarrollado por Hans Dobbertin, Antoon Bosselaers y Bart Preneel en 1992. La idea de desarrollo de RIPEMD se basa en MD4, que en sí mismo es una función hash débil. Está desarrollado para funcionar bien con procesadores de 32 bits. Tipos de RIPEMD:

* RIPEMD-128
* RIPEMD-160
* RIPEMD-256
* RIPEMD-320

El mensaje se procesa mediante la función de compresión en bloques de 512 bits y se pasa a través de dos flujos de este subbloque utilizando 5 versiones diferentes en las que el valor de la constante "k" también es diferente.

**APLICACIONES:** Password Storage y Data Integrity Check.

**VENTAJAS:** Variacion de “niveles de seguridad” de acuerdo al tipo de RIPEMD que se utilice. Es un reemplazo seguro para las funciones hash de 128 bits MD4, MD5 y RIPEMD (versiones anteriores. Por ejemplo, RIPEMD-160 es más seguro que RIPEMD-128).

**DESVENTAJAS:** RIPEMD-160 parece ser bastante robusto. Sin embargo, apareció después de SHA-1, y es más lento que SHA-1, por lo que tuvo un éxito limitado. Las versiones posteriores parecen ser utlizadas unicamente en “productos niche”.

**Bibliografía**

* <https://www.cs.cmu.edu/~adamchik/15-121/lectures/Hashing/hashing.html>
* <https://searchsecurity.techtarget.com/definition/MD5>
* <https://www.2brightsparks.com/resources/articles/introduction-to-hashing-and-its-uses.html>
* <https://www2.seas.gwu.edu/~poorvi/Classes/CS381_2007/Whirlpool.pdf>
* <https://es.wikipedia.org/wiki/Whirlpool>
* <https://konklone.com/post/why-google-is-hurrying-the-web-to-kill-sha-1>
* <https://blog.jscrambler.com/hashing-algorithms/>
* https://www.tutorialspoint.com/cryptography/cryptography\_hash\_functions.htm