Hashing: Técnicas y Hash para la Protección de Datos

Samuel Sánchez, Pablo Domínguez, Luis Velásquez, samuel.sanchez1@utp.ac.pa, pablo.dominguez2@utp.ac.pa, luis.velasquez@utp.ac.pa.
Profesor Asesor: Ingeniera Tejedor – Morales, María Yahaira
Universidad Tecnológica de Panamá, C.R. de Coclé, Grupo de Investigación SoftSolution Group, maria.tejedor@utp.ac.pa.

Resumen- En la informática existen diferentes campos, el artículo trata un tema puntual de la estructura de datos denominado hashing; el cual está relacionado con la criptografía para trazar el desarrollo de las funciones hash conocidas también como resumen criptográfico.
Empezamos definiendo el concepto del hashing, seguido las técnicas, especificando como trabaja cada una de ellas.
Las técnicas hash se establecen como métodos que mediante una ecuación fueron creados para generar posiciones en una tabla (arreglo) que será la que contendrá datos.
Surge, por tanto, la necesidad de comprender las diferentes funciones como: SHA-1, SHA-3, SHA-256. La siguiente sección propone el concepto, su campo de acción, ventajas y debilidades a analizar.
La revisión bibliográfica nos permite presentarles una comparación de las técnicas hashing, destacar que al ser distintas ninguna es mejor que otra, cada una de ellas puede presentar una colisión.
Tenemos un amplio número de técnicas a nuestra disposición para el manejo y solución de colisiones, entre las que destacamos las más eficientes: Prueba linear, Doble dirección, entre otras.
Curiosamente, también se observó una útil herramienta para demostrar el resultado y procedimientos de la generación de claves de las funciones Hash.

Palabras claves- Concepto de Hashing, criptografía, Manejo y Solución de Colisiones, Hash, Técnicas del Hashing

Abstract– In computer science there are different fields, the article deals with a specific topic of the data structure called hashing; which is related to cryptography to trace the development of hash functions also known as cryptographic summary.
We start by defining the hashing concept, followed by the techniques, specifying how each of them works.
Hash techniques are established as methods that by means of an equation were created to generate positions in a table (array) that will contain data.
Therefore, there is a need to understand the different functions such as: SHA-1, SHA-3, SHA-256. The following section proposes the concept, its field of action, advantages and weaknesses to analyze. The literature review allows us to present a comparison of hashing techniques, highlighting that being different is no better than another, each of them can present a collision. We have a large number of techniques at our disposal for handling and collision solutions, among which we highlight the most efficient: Linear test, Double direction, among others.
Interestingly, a useful tool was also observed to demonstrate the result and procedures of the key generation of the Hash functions.

Keywords– Cryptography, Hashing Concept, Handling and Collision Solution, Hash Models, Hashing Techniques

1. INTRODUCCIÓN

Esta investigación documenta un área que, aunque no es reciente, es compleja y por lo tanto llama poco la atención en estudiantes a niveles iniciales de pregrado. Es así que la curiosidad, el querer descubrir, la necesidad de afianzar y comprender qué es Hashing, cómo trabaja, cuál es el objetivo de utilizar estos métodos; nos impulsa a compartir las respuestas a estas interrogantes guiados por revisión de la literatura y el análisis de la misma.

Se plantea cómo pueden generarse las claves, ya que todas no funcionan de manera similar, estas variantes de producirlas pueden generar colisiones; de allí que también se explica cómo resolverlas mediante las funciones de Pliegue, residuo y cuadrado medio.

Sobre la base conceptual anterior se presenta posteriormente un análisis de las funciones hash, cómo son utilizadas o empleadas por las empresas. De igual manera se menciona los modelos SHA-256 y SHA-3 que utilizan las empresas para generar claves y proteger sus archivos. Finalmente, se ofrece una contextualización del hash, dentro de ella un programa que puede utilizarse para experimentar con sus funciones, además la realidad de la empresa Barco Silex que hace uso de unas de las nuevas funciones hash.

El hash en conjunto con la encriptación de la información, provee a estudiantes como nosotros un sin número de oportunidades de desarrollo de aplicaciones. Despertar la inquietud por continuar y aportar es la meta que es aspiramos con este documento.

2. CONCEPTO DE HASHING

Dentro del mundo de las estructuras de datos existe un sinfín de temas que conllevan diferentes métodos, funciones y

aplicaciones para explotar en el contexto de programación; con el objetivo de innovar con soluciones que den soluciones a actividades de la cotidianidad.

Al hablar de hashing estamos refiriéndonos a una tabla que almacena tanto registros como objetos, para después realizar una búsqueda de la cual su prioridad es tener un constante tiempo de recuperación en base a 0 y 1 sin tomar en cuenta qué cantidad de elementos puedan están dentro de una tabla. La manera correcta para poder lograr ese objetivo es crear una tabla de gran medida para abarcar cada elemento que se pueda almacenar y luego guardar cada objeto dentro de una posición con el valor clave del objeto. Si se utiliza Java para implementar Hash la clave sería la que retornaría el método hashCode() de cada uno de los objetos.

De este modo la recuperación de un objeto es directa. En caso de que la clave sea numérica se debería crear una tabla aún más grande que la normal para poder conseguir que cada valor sea un índice válido. Un ejemplo sería las cédulas de cada ciudadano del país. Habría que hacer millones de casillas para que cada índice sea válido [1][2].

3. TÉCNICAS DEL HASHING

Las técnicas hashing o tablas de dispersión se han creado para crear mediante una ecuación y una clave de entrada, como parte fundamental, un dato que será una posición en un arreglo [1].

La función hash cumple con una operación fundamental y es que, si queremos buscar el objeto que guardamos en "x posición", mediante la clave y la ecuación, podremos obtener el archivo guardado, sin importar si los elementos en arreglo están ordenados o no, y sin importar cuál sea el tamaño del arreglo, ya que las técnicas hashing son unos de los métodos de búsqueda más rápido, lo cual es una gran ventaja.

Es posible que ocurra que a la hora de generar una posición en el arreglo o a la hora de buscar un elemento, la posición generada ya existe o nos devuelva un dato no deseado, en caso de la búsqueda, esto se conoce como colisiones. Importante conocer que dada dos claves diferentes k1 y k2 nos deberían dar resultados diferentes.

$$H(K1)=d,\ H(K2)=d \text{ y } K1 \neq k2 \qquad (1)$$

Ingresar una misma posición o destino a más de un dato a guardar, ya sea que las claves de entrada sean distintas, nos pueden generar una misma posición [1].

Además, las técnicas de hashing cuentan con otras funciones que solventan las colisiones por muchas formas, como son: hashing por residuo, pliegue y cuadrado medio. Aunque se debe escoger bien la función a utilizar ya que, pueden ser diferentes procesos, pero se puede presentar una colisión de todas formas.

A. *Funciones del Hash*

- Hashing por residuo o función modular, para utilizar esta técnica es necesario dividir el valor de la clave entre un número y utilizar el residuo de esta como una ruta para el registro [2]. Es considera una función bastante simple en la cual se toma k como clave y N como el tamaño del arreglo para su ecuación que sería la siguiente:

  $$H(K)=(K \bmod N)+1 \qquad (2)$$

  Esto forma se toma el residuo de la operación, es por eso el nombre de la función, y se le suma 1 para una mayor uniformidad en cual será claramente la posición en el arreglo [1].

- Hashing por pliegue, en esta técnica la clave es dividida en varias partes cada una de estas excepto la última tiene la misma cantidad de elementos que tiene la ruta relativa. Estas divisiones luego son unidas una sobre otra y sumadas. El resultado es la ruta relativa. Igual como se trabaja en el método por cuadrado medio el tamaño del espacio de la ruta relativa es potencia 10. El hashing por pliegue define que la clave debe separarse en la misma cantidad de dígitos aunque el último quede con menor dígitos, una vez teniendo separada en dígitos la clave se procede a realizar la suma entre ellos y sumarle uno [1][2].
  La fórmula para el hashing por pliegue es la siguiente:

  $$H(K)=digmensig(d1..di)+(d2..di)+1 \qquad (3)$$

  Siendo K la clave, y d1 y d2 la clave en partes, digmensig es la suma de las partes de la clave y por último se le suma 1, la posición generada, se deberá obtener dentro del rango 1 a N tamaño del arreglo.
- Hashing por Cuadrado Medio o Función Cuadrado, para esta técnica la clave de la tabla se eleva al cuadrado, luego se sacan ciertos elementos de la mitad del resultado para formar la ruta relativa. Si se requiere obtener una ruta de un número entonces se

toma en ambos extremos de la clave elevada al cuadrado, tomando ciertos elementos intermedios. Estas mismos deben extraerse para cada clave [2]. Otra alternativa, que se nos presenta de Hashing por cuadrado medio y como su nombre lo dice, dada una clave K se eleva al cuadrado y se toman los dígitos que se encentran en mitad del resultado. Su fórmula sería la siguiente:

H(k)= (k2)+1 (4)

Por ejemplo, digamos que el resultado de una clave k(1234) el resultado al cuadrado sería 1,522,756 teniendo en cuenta que nuestro arreglo es de tamaño 100, tomaremos los dígitos de la mitad, o sea 1,522,756 y será nuestra posición en el arreglo, se preguntará porqué solo dos dígitos, en realidad hay siempre que considerar el tamaño del arreglo, no sobrepasarnos de su tamaño [1].

4. HASH

El proceso de las funciones hash se le conoce como criptografía, es decir que es capaz de transformar cualquier entrada, ya se texto, una imagen jpg, png o un archivo

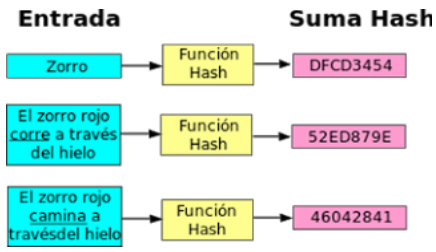

Fig. 1, Proceso criptográfico de las funciones hash

transformarlo a un único código. Pero existen diferentes modelos de algoritmos que nos permiten realizar estos procesos.

De las funciones Hash nacieron algoritmos de seguridad, creadas por empresas, con el objetivo de comparar y encontrar las colisiones, existen varios modelos de seguridad basados en funciones hash como lo son: SHA-1 en el mismo existen diferentes variantes con mejoras. El SHA-1 es uno de los más utilizados por las empresas, pero el mismo tendrá que dejar de usarse debido a que google encontró una colisión a la hora de usar 2 PDF con la misma firma [3][4][5].

A. *UTILIZACIÓN DEL HASH*

Las diferentes funciones Hash que existen son utilizadas más que nada para resguardar la integridad de los archivos de grandes empresas, para certificación que las páginas web, proteger los derechos de autor de libros, películas, y otros contenidos que se encuentran en internet.

Un ejemplo claro del uso de Hash es en las contraseñas, ciertos servicios en línea generan una función hash cuando el usuarios crean sus contraseñas, estos servicios en línea no guardan en texto plano las contraseñas de sus usuarios ya que esto significaría un problema grande en caso de que sufran de algún ataque a sus Base de Datos, generando una función Hash de la contraseña de los usuarios será más difícil de descifrar cuál será la contraseña en texto plano brindando mayor seguridad a los usuarios resguardando su información. En el caso de la protección de los archivos y valorar los derechos de autoría con Películas y Canciones, los autores pueden generar una función Hash de su archivo, ya sea este un libro, película, canción, o cualquier otro, de manera que si alguien de alguna forma logra apoderarse de este archivo y lo desea compartir no le sea tan fácil, ya que ese archivo puede que esté en una lista negra de archivos protegidos [11] [12].

De acuerdo a lo analizado, las empresas que quieran mantener su información a salvo tendrán que cambiarse a las mejoras del SHA-1 que son: SHA-3 Y SHA-256.

- SHA-3, SECURE HASH ALGORITM, fue creado a partir del SHA-2 como una actualización para solventar errores que se encontraban dentro de ese método. Fue creado por diseñados como Guido Bertoni, Joan Daemen, Michaël Peeters y Gilles Van Assche. Debido a que los navegadores como Google Chrome, Internet Explorer entre otros utilizan el método SHA-1 tiene una baja seguridad que estos diseñadores demoraron solo 10 días en descubrir, Google informó que su navegador cambiaría a SHA-3 durante el año 2017 y de igual manera lo realizan otras empresas como Microsoft [6].

- SHA 256, son nuevas funciones hash que utilizan palabras de 32 bits a 64 bits, la función SHA 256 pertenece a la familia del SHA-2. Esta función participa para la autenticación de software, y para la firma de mensajes [7].

5. COMPARACIÓN DE FUNCIONES HASH

Las funciones de hash como ya sabemos cumplen con operaciones fundamentales de representar de forma compacta, mediante cadenas o arreglos, conjunto determinados de datos en una posición específica en función de búsqueda. Estos a su vez pueden presentar las llamadas colisiones, que pueden

duplicar posiciones de entradas de datos y generar problemas a la hora de la búsqueda.

Las técnicas de cálculo de direcciones: por residuo, por cuadrado medio y por pliegue, son funciones que permiten solucionar algunas colisiones, aunque depende de cada función, y cuál sea la más apropiada según el problema ocasionado por la colisión.

Aunque cada técnica tenga la capacidad de desempeñarse una mejor que otra en situaciones particulares, la técnica del Hash por residuo es considerada la que se desempeña mejor; sin embargo, el término de desempeño va a variar entre cada función, dando a entender que ninguna puede ser mejor que la otra en el sentido estricto de la palabra, y que todo dependerá del contexto en el que se aplique la función. Por ejemplo, el método del medio cuadrado puede desempeñarse mejor en archivos con factores de carga baja.

Mientras que el método de pliegue es más sencillo de calcular, aunque produce resultados erráticos dependiendo de la longitud de la llave a utilizar en la dirección. La mejor función de una con la otra depende de cómo se distribuyen los valores de las llaves [8].

Las Técnica de cálculo de direcciones presentan ventajas y desventajas.

Algunas de las ventajas es que son una de las funciones más rápidas de búsqueda y nos permiten usar valores naturales de la llave, puesto que traducen internamente a variadas direcciones fáciles de ubicar. También nos permite lograr una independencia lógica y física, debido a que los valores de las llaves logran su independencia del espacio de direcciones, en el caso de los índices no requiere almacenamiento de más [8].

Por otro lado, están sus desventajas que son en las funciones de hash de cálculos de direcciones, no se permite usar registros de longitud variable. No clasifica algunos archivos. No permite repetir una misma llave. Sólo permite conectar la posición de una sola llave [8].

Tabla 1. Cuadro comparativo de ventajas y desventajas de cada una de los métodos para calcular direcciones.

Métodos para Calcular Direcciones	Ventajas	Desventajas
Hash por Residuo	Son una de las funciones más rápidas de búsqueda y nos permiten usar valores naturales de la llave. Nos permite lograr una independencia lógica y física, debido a que los valores de las llaves logran su independencia del espacio de direcciones. Es considerada la que mejor se desempeña.	El desempeño va a variar entre cada función. Todo dependerá del contexto en el que se aplique la función. No permite repetir una misma llave. No se permite usar registros de longitud variable.
Medio Cuadrado	Son una de las funciones más rápidas de búsqueda. Puede desempeñarse mejor en archivos con factores de carga baja.	No se permite usar registros de longitud variable. No permite repetir una misma llave. Sólo permite conectar la posición de una sola llave
Hash por Pliegue	Nos permite lograr una independencia lógica y física Es más sencillo de calcular.	No se permite usar registros de longitud variable. No permite repetir una misma llave. Produce resultados erráticos dependiendo de la longitud de la llave a utilizar en la dirección.

6. MANEJO Y SOLUCIÓN DE COLISIONES

A. *Manejo de Colisiones*

El término colisiones tienen un aspecto negativo, pues producen que por ejemplo dos tipos de entrada distintas de una función de hash, produzcan la misma salida en la misma dirección, en vez de diferentes [1].

Sin embargo, existen ciertos puntos positivos a la hora de manejar con una colisión, por ejemplo, si se presenta una colisión repentina, se puede detectar qué tipo de función de hash es más eficiente tratándola.

Tratándose de una colisión en una tabla direccionada en una misma ubicación se puede elegir entre métodos de direccionar abiertamente, o por encadenamiento. Ambos tienen la ventaja de que manejan las colisiones más rápido buscando desde el origen de la misma [1].

Sin embargo, tenemos entendido que las colisiones representan un obstáculo a la hora de representar tipos determinados de datos y estas, aunque tengan muchos métodos para tratarlas, se encuentran presente en todos los tipos arreglos de llaves indeterminado número de veces.

Estas colisiones impiden el correcto direccionamiento de entradas de llaves, por ende, hacen que en el momento del ordenamiento de la búsqueda sea más lento.

También existen casos en los que solo devuelve datos y es imposible detectar en qué punto se encuentra el origen de los valores colisionados [1].

B. *Solución de Colisiones*

La manera de resolver el problema de las colisiones es reservar una casilla por clave. Es decir, que se correspondan una a una con las posiciones del arreglo. Pero esto puede tener un alto costo de memoria. Por lo tanto, deben analizarse otras alternativas que permitan equilibrar el uso de memoria con el tiempo de búsqueda [1].

Existen métodos para solventar las colisiones, como lo es el método de Resignación en la cual trabaja de tres formas:

- Prueba Lineal: el direccionamiento abierto se basa en hacer un recorrido dentro de la tabla a partir del índice de la tabla en donde se produce la colisión, se recorre todo hasta encontrar una posición que este vacía y almacenarla en ese lugar dependiendo si lo que se busca es almacenar de no encontrar ningún espacio con el índice buscado o espacio para almacenar se regresara a la posición inicial como si se tratara de un ciclo. Para resolver este tipo de colisiones se recomienda siempre dejar espacios de más en el tamaño de las tablas en el caso de almacenar datos dentro de una tabla hash [1].

- Prueba Cuadrática: Es un método similar al de la prueba lineal lo que varía seria la forma en la que se asigna los valores, la variación de esta distribución nos dará mejores resultados a la hora de obtener las posiciones colisionadas [1].

- Doble Dirección: este proceso consiste en crear una posición alternativa en la tabla al obtener una colisión aplicando una de las funciones hash, ya sea al buscar una posición o al ingresar en una posición vacía. Para detener este proceso es necesario encontrar una posición vacía o el elemento que se estaba buscando. La función aplicada a la nueva llave puede ser o no ser la misma que se le aplica a la llave que produce una colisión [1].

- Diferencias entre Prueba Lineal y Doble Dirección: Ambos son métodos muy parecidos. Pero en los casos en que la cantidad de colisiones son bajas la prueba lineal tiende agruparlas, mientras que los de doble dirección tienden a dispérsalos por los espacios que tenga la tabla, la doble dirección tiene un comportamiento casi perfecto con factores de mayores colisiones mientras que los lineales son malísimos con estos tipos de colisiones. Con el método de doble dirección podemos llegar a tener búsquedas exitosas [1].

Otros métodos para solucionar las colisiones son los siguientes:

- Arreglos anidados, este método consiste en que la posición que produce colisión tenga otro vector para guardar esta posición, pero este método parece fácil, sin embargo, es realmente inútil debido a la cantidad de espacios que habría que crear cada vez que se produce una colisión dentro de los arreglos creados sería imposible tener un cálculo adecuado de cuanto espacio tener [1].
- Encadenamiento, este método consiste en que cada elemento contenga un enlace a una lista para que de esta manera sea manera sea más fácil almacenar los valores colisionados este método es muy eficiente debido al dinamismo [1].

7. CONTEXTUALIZANDO EL HASH

Existen programas que nos facilitan el uso de los modelos Hash, y uno de ellos es el programa llamado File Checksum Utility, el programa se puede descargar de forma gratuita de su página oficial, de dos formas, una versión que se instala en la computadora y la otra versión que es totalmente portable [9]. El programa hace uso de los modelos hash: SHA-1, SHA-256, SHA-512, RIPEMD. El sistema cuenta con una interfaz bastante agradable, como se muestra en la Figura 2.

El programa una vez ejecutado nos brindará tres opciones, una de esas opciones es calcular la suma de verificación de cada uno de los modelos hash mencionados anteriormente, a

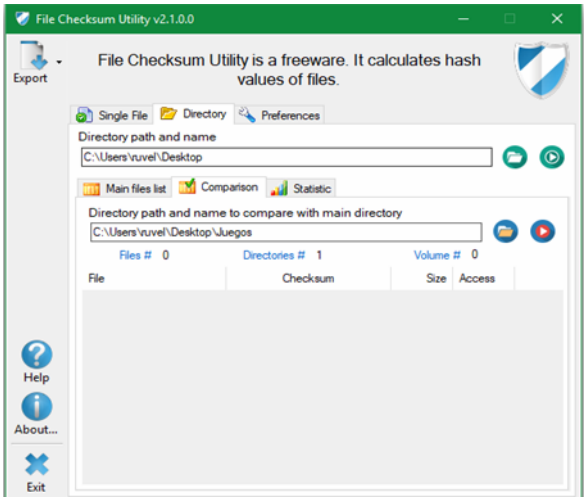

Figura 2. Interfaz de programa File Checksum utility.

partir de un único archivo que puede ser de cualquier formato. La segunda opción permite comparar la integridad entre dos carpetas, además de verificar los archivos descargados, que no haya sido modificado en el proceso de descarga. Y por último se nos presente una sección de configuración del programa.

Uno de los objetivos del programa, es que un usuario tenga la facilidad de una vez generado los diferentes códigos, mediante los diferentes modelos de hash, pueda guardarlos para después, si el usuario así lo desea, pueda verificarlos nuevamente, en caso que los mismos hayan sido modificados, lanzarán un nuevo código, lo cual significa que el archivo o carpeta fue modificado, ya sea por un virus en el sistema o por que alguien modificó el archivo.

Este software nos permite ver más claramente cómo funciona el hash, una vez que nos encontramos en la primera pestaña de "single file" en la parte de "Directory path and name" hacemos clic en el icono de la carpeta y nos lanzara el explorador de Windows, o del sistema operativo que se esté utilizando. A partir de ahí localizar el instalador del programa, luego lo abrimos e instantáneamente deberá generar la clave. En su sitio de descarga podemos visualizar la clave generada del MD5 el cual debe coincidir con el que genera el programa, si son iguales satisfactoriamente no ha sufrido de modificaciones, de lo contrario nos generaría otra clave. El FileCheckSum Utility sirve como un control importante sobre la integridad de los archivos que descargamos de internet o de los que contenemos en nuestro ordenador.

La empresa **Barco Silex** brinda servicios como: núcleos IP de seguridad, encriptación y procesamiento de vídeo, así como en servicios de diseño electrónico [10]. Dicha compañía ha empezado a usar específicamente el modelo de hash SHA-3, para la comprobación de integridad de mensajes, transacciones y datos, además de ser empleada en áreas de comercio electrónico, aplicaciones financieras y otros procesos delicados que tienen que ver con el manejo de dinero por la web [10]. Un gran detalle a destacar de la empresa es que está en constante renovación de sus sistemas, para mantener todos sus procesos seguros tal y como demandan sus clientes.

La forma en que la empresa utilizará El SHA-3 será, por ejemplo, generando un código para los datos guardados, en caso de que el archivo sea modificado intencionalmente o indirectamente, provocará que a la hora de verificar nuevamente el archivo dé como resultado un código diferente al que ya existía, a partir de ahí los encargados en el área trabajarán en ello.

8. CONCLUSIÓN

Actualmente muchas empresas u organizaciones que utilizan tecnologías se ven obligadas a realizar cambios de seguridad en sus plataformas o utilizar un método seguro capaz de evitar fraudes, estafas, etc. Aquí es donde el Hash apuesta por brindar los servicios necesarios para evitar colisiones.

A pesar que son varios los modelos cada día se estudian más para revelar posibles anomalías y de esa forma sustituir por una versión estable. Siendo este el caso del modelo SHA-1 por lo que será remplazado de la mayoría de la página web por defectos de seguridad.

No obstante, existen otras alternativas al SHA-1 muy utilizadas como es el SHA-256 y SHA-3 versiones estables en temas de seguridad actualmente y como tal cuentan con un porcentaje muy bajo de error, sin embargo, no las absuelve totalmente de problemas aún no descubiertos.

Ninguna de las técnicas del Hashing es mejor que otra, pero cada una es eficaz en su especialidad. Incluso realizando tareas no propias, pero a su vez restándole características que la limitan.

En el momento en donde una clave se nos duplica y nos enfrentamos a una colisión es necesario tomar en cuenta parámetros para intentar resolver los problemas ocasionado por una llave duplicada. El hecho de encontrar una solución no es un proceso simple, conlleva una serie de paso para aplicar una técnica compleja y corregir el posicionamiento incorrecto.

Sería interesante investigar como ésta tecnología podría revolucionar las diversas áreas de estudios en la actualidad, por lo tanto, su mayor inconveniente ha sido su total enfoque a las criptodivisa/Blockchain, debido a esto, día a día los

expertos ven potencial para implementarla en campos específicos, para resolver problemas o crear nuevas tecnologías.

REFERENCIAS

[1] Ullman, J., Aho, A. y Hopcroft, J. Estructura de Datos y Algoritmos.Addison-Wesley. México. 1988

[2] Víctor Valenzuela Ruz, ”Manual Análisis de Algoritmos”. [Online]. Available: http://colabora.inacap.cl/sedes/ssur/Asignatura%20Indtroduccion%20a%20la%20Programacn/An%C3%A1lisis%20de%20Algoritmo/Manual-Analisis%20de%20Algoritmos_v1.pdf. [Accessed: 03-may-2017]

[3] Maciej Heyman, “SHA-1 Collision Found”. [Online]. Available: http://www.military-technologies.net/2017/03/11/sha-1-collision-found/. [Accessed: 08-apr-2017]

[4] “Secure Hash Algorithm 1 deemed unsafe”. [Online]. Available: http://www.techcentral.ie/secure-hash-algorithm-1-deemed-unsafe/. [Accessed: 8-apr-2017]

[5] Rhiannon Williams, “Amber Rudd did mistake hashing for hashtags, MPconfirms”. [Online]. Available: https://inews.co.uk/essentials/news/technology/amber-rudd-mean-hashing-not-necessary-hashtags-clarifies-mp/. [Accessed: 1-apr-2017]

[6] “Encryption Chat, Secure Hash Algorithm 3 SHA-3”. [Online]. Available: https://www.sha-3.com/. [Accessed: 3-may-2017]

[7] Mediateam Ltd, “Secure Hash Algorithm 1 deemed unsafe”. [Online]. Available: http://www.techcentral.ie/secure-hash-algorithm-1-deemed-unsafe/, [Accessed: 3-may-2017]

[8] J.Lawrence Carter y.Mark N. Wegman, Universal Classes of Hash Functions, Journal of Computer and System Sciences 18, 143-154 (1979), IBM Thomas J. Watson Research Center, Yorktown Heights, New York 10598 Received August 8, 1977; revised August 10, 1978

[9] Rubén Velasco, “Genera un hash de tus archivos y carpetas con File Checksum Utility”. [Online]. Available: https://www.softzone.es/2017/05/06/generar-hash-archivos-file-checksum-utility/. [Accessed: 26-may-2017]

[10] “New SHA-3 hashing IP from Barco Silex helps customers implement future-proof security”. [Online]. Available: https://www.design-reuse.com/news/42032/sha-3-hashing-ip-barco-silex.html. [Accessed: 02-may-2017]

[11] Gutiérrez Pedro, "¿Qué son y para qué sirven los hash?: funciones de resumen y firmas digitales". [Online]. Available: https://www.genbetadev.com/seguridad-informatica/que-son-y-para-que- sirven-los-hash-funciones-de-resumen-y-firmas-digitales. [Accessed:22-jan-2018]

[12] "¿Qué Es Un Hash Y Cómo Funciona?". [Online]. Available: https://latam.kaspersky.com/blog/que-es-un-hash-y-como-funciona/2806/. [Accessed: 22/01/2018]