Caso 3, informe

Sergio Franco (202116614), Sergio Oliveros (202123159), Gabriel Dicelis (201920847)

1. IMPLEMENTACIÓN DEL PROTOTIPO

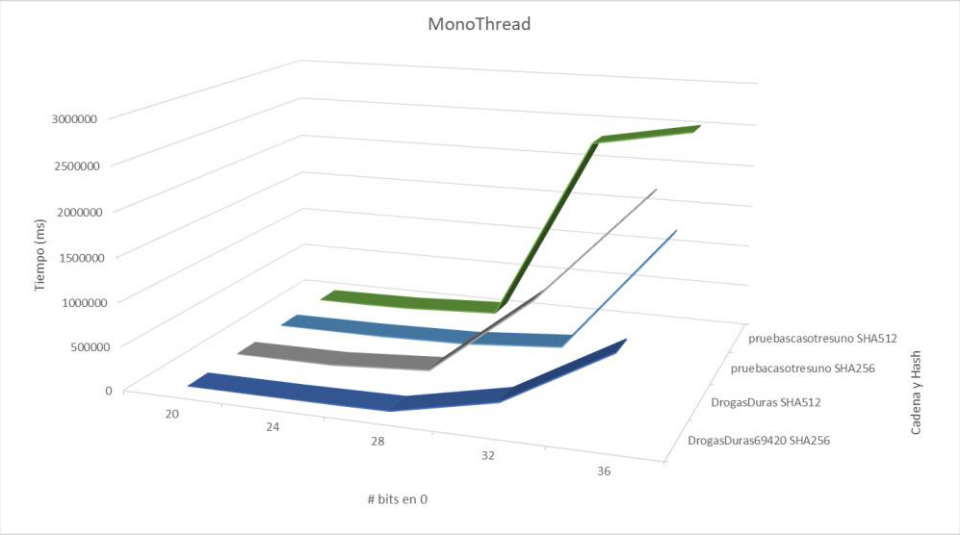
1.1. Tablas de datos

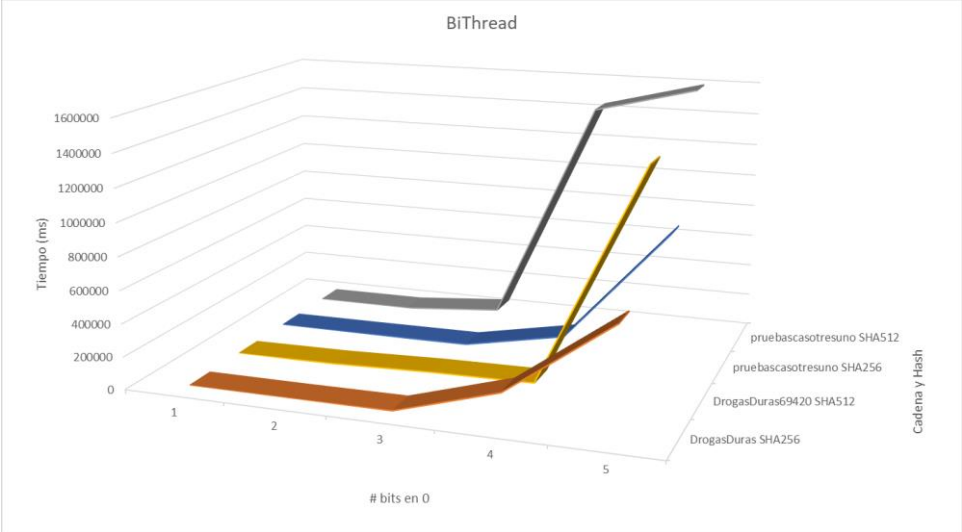
Cadena	Algoritmo	Núm. Threads	Condición	(v) encontrado	Tiempo (ms)
DrogasDuras69420	SHA256	1 Thread	20	lngf	37
			24	ujkks	1068
			28	komlnb	15517
			32	gsshrxl	262625
			36	No econtró	927862
		2 Thread	20	lngf	104
			24	majzbcj	592
			28	mdaxrqh	4389
			32	qnpfvbk	189794
			36	No econtró	648590
	SHA512	1 Thread	20	wzzm	160
			24	bhgqgq	7000
			28	abdxsef	81679
			32	mgxjqyo	1016854
			36	No econtró	2236466
		2 Thread	20	wzzm	130
			24	mcegzul	6819
			28	mgxjqyo	26193
			32	mgxjqyo	34977
			36	No econtró	1376402

Cadena	Algoritmo	Núm. Threads	Condición	(v) encontrado	Tiempo (ms)

pruebascasotresuno	SHA256	1 Thread	20	ehqqm	255
			24	czuamf	5185
			28	wnzgni	31586
			32	cpkpdne	122221
			36	No econtró	1511816
		2 Thread	20	mabezqe	286
			24	maxdgqv	2492
			28	mcmxzoi	6511
			32	nxrledr	126909
			36	No econtró	827794
	SHA512	1 Thread	20	adkgw	179
			24	ctpped	14578
			28	oryfur	71363
			32	No econtró	2308476
			36	No econtró	2518432
		2 Thread	20	adkgw	296
			24	majaird	1840
			28	mgvomcq	52386
			32	No econtró	1440467
			36	No econtró	1597184

1.2 Gráficas sobre datos:





1.3 Velocidad Procesador

Para un procesador de 2.4GHz su duración de ciclo de reloj sería:

$$T = \frac{1}{f} = \frac{1}{2.4 \cdot 10^9} = 0.4167\ ns$$

Y en promedio generar y evaluar un valor para determinar si cumple o no con la condición buscada toma un tiempo de $12125\ ns$ es decir, 29097.672 ciclos.

Este tiempo se midió con un promedio de los resultados de las mediciones en el siguiente fragmento de código:

```
private boolean generateStringsInRange() {
    boolean verificacion = hash.crearValidarHash(currentV, cadena, ceros);

    while (monitor.getContinuar() && compareOrder(currentV, fin)) {

        if (verificacion)
        {
            monitor.terminar();
        }
        else
        {
            long startTime = System.nanoTime();
            currentV = generateNextString(currentV);
            verificacion = hash.crearValidarHash(currentV, cadena, ceros);
            long endTime = System.nanoTime();
            long timeElapsed = endTime - startTime;
            System.out.println("Execution time in nanoseconds: " + timeElapsed);
        }
    }
    return verificacion;
}
```

1.4 Tiempo Exploración Espacio MonoThread

Para un programa monothread, en el peor caso (explorar todo el espacio de búsqueda) el toal de combinaciones que se tendrían que evaluar es $26\text{^}1 + 26\text{^}2 + 26\text{^}3 + 26\text{^}4 + 26\text{^}5 + 26\text{^}6 + 26\text{^}7 = 8353082582$ y teniendo en cuenta en cuenta que cada combinación toma $12125ns$ en promedio para ser generada y verificada el programa en el peor de los casos tomaría un tiempo de $12125ns \cdot 8353082582 = 101281126306750ns$ en promedio.

2. ANÁLISIS Y ENTENDIMIENTO DEL PROBLEMA

2.1 Sobre la actualidad de los algoritmos de generación de códigos criptográficos de hash

2.1.1 ¿Cuáles se usan hoy día?

Aunque hoy día el algoritmo más seguro es SHA-3, los algoritmos de la familia SHA-2 siguen siendo los más utilizados ya que de igual manera no se les ha encontrado vulnerabilidades, estos son SHA2-224, SHA2-256, SHA2-384 y SHA2-512.

2.1.2 ¿Por qué dejamos de usar aquellos algoritmos que se consideran obsoletos?

Dos algoritmos ampliamente utilizados anteriormente que ya no se consideran se guros son MD5 y SHA-1 debido a vulnerabilidades en sus diseños que permiten a los atacantes encontrar colisiones con relativa facilidad. En el caso de MD5, se descubrió su vulnerabilidad en 2004 y en el caso de SHA1 en 2017.

2.1.3 ¿Qué referencias bibliográficas usaron para responder estas preguntas?

Principalmente publicaciones del Instituto Nacional de Estándares y Tecnología (NSIT) y Google, específicamente *Announcing the first SHA1 collision*[1] de Google y *NIST Transitioning Away from SHA-1 for All Applications*[2], *NIST Comments on Cryptanalytic Attacks on SHA-1*[3] y *CVE-2004-2761 Detail*[4] del NSIT.

2.1.4 ¿Por qué esas referencias tienen autoridad sobre este tema?

El NSIT es una agencia del gobierno de los Estados Unidos que se encarga de la seguridad de la información. El NSIT publica estándares y directrices para la seguridad de la información, incluyendo recomendaciones sobre el uso de algoritmos de hashing.

Google es una de las principales empresas tecnológicas del mundo y utiliza algoritmos de hashing en una amplia gama de productos y servicios. Por ejemplo, Google utiliza algoritmos de hashing para almacenar contraseñas, verificar la integridad de los datos y generar huellas digitales de archivos por lo que ha realizado contribuciones significativas a la criptografía y la seguridad en línea, incluyendo la identificación de vulnerabilidades en algoritmos de hashing existentes y la propuesta de nuevos estándares.

2.2 Caso de uso de blockchain en el contexto de la Universidad de los Andes

2.2.1 Descripción del caso

La blockchain puede utilizarse para almacenar de forma segura y transparente los registros académicos de los estudiantes, como las calificaciones, las notas, los certificados y los títulos. Esto puede ayudar a reducir el fraude académico y a mejorar la confianza en los resultados académicos.

2.2.2 ¿Cuál o cuáles de los cuatro problemas de seguridad, de los estudiados en clase, resuelve blockchain en este caso?

Blockchain es capaz de resolver los cuatro problemas de seguridad. Por ejemplo, si un estudiante afirma haber obtenido una determinada calificación, la blockchain puede utilizarse para verificar esa afirmación. La blockchain también puede utilizarse para prevenir que un estudiante tome dos veces el mismo curso o que se le otorgue una titulación que no haya obtenido.

Por lo tanto, la tecnología blockchain puede proporcionar una solución segura y eficiente para la gestión de registros académicos en una universidad, abordando eficazmente los problemas de espionaje, adulteración, suplantación y repudio.

2.2.3 ¿Cómo los resuelve? Es decir, divida la tecnología en componentes e identifique qué parte, o partes, de la tecnología están involucradas en la resolución del problema y cómo lo resuelven.

Espionaje: Blockchain es una tecnología descentralizada, lo que significa que los datos se almacenan en una red de computadoras en lugar de en un único servidor, esto dificulta que los atacantes obtengan acceso ilegal a los datos.

Adulteración: Blockchain utiliza un sistema de cifrado para proteger los datos, esto hace que sea muy difícil para los atacantes modificar datos sin ser detectados.

Suplantación: Blockchain utiliza un sistema de autenticación para identificar a los usuarios, esto ayuda a evitar que los atacantes se hagan pasar por otras personas o comunidades.

Repudio: Blockchain registra todas las transacciones en cadena de bloques, lo que proporciona información histórica sobre eventos que se puede utilizar para demostrar quién realizó la acción.

2. REFERENCIAS

[1]Google. (2017, 23 febrero). *Announcing the first SHA1 collision*. Google Online Security Blog. https://security.googleblog.com/2017/02/announcing-first-sha1-collision.html

[2]*NIST transitioning away from SHA-1 for all applications* | NIST. (2022, 20 diciembre). NIST. https://www.nist.gov/news-events/news/2022/12/nist-transitioning-away-sha-1-all-applications

[3]*Computer Security Division, Information Technology Laboratory*, National Institute of Standards and Technology, U.S. Department of Commerce. (s. f.). NIST comments on cryptanalytic attacks on SHA-1 | CSRC. https://csrc.nist.gov/News/2006/NIST-Comments-on-Cryptanalytic-Attacks-on-SHA-1

[4]*NVD - CVE-2004-2761*. (s. f.). https://nvd.nist.gov/vuln/detail/CVE-2004-2761