MAESTRÍA EN AUDITORIA DE TECNOLOGÍA DE LA INFORMACIÓN

COMPARACIÓN DEL RENDIMIENTO Y NIVEL DE SEGURIDAD EN ALGORITMOS CRIPTOGRÁFICOS LIGEROS PRESENT, CLEFIA, KECCAK Y HIGHT: UNA REVISIÓN SISTEMÁTICA

Propuesta de artículo presentado como requisito para la obtención del título:

Magíster en Auditoría de Tecnologías de la Información

Por la estudiante:
César Alvarito CORONEL GONZÁLEZ

Bajo la dirección de:
Lohana Mariella LEMA MORETA.

Universidad Espíritu Santo
Maestría en Auditoría de Tecnología de la Información
Samborondón - Ecuador
Abril del 2018

Comparación del rendimiento y nivel de seguridad en los algoritmos criptográficos ligeros Present, Clefia, Keccak y Hight: una revisión sistemática.

Comparison of performance and security level in light cryptographic algorithms Present, Clefia, Keccak and Hight: a systematic review.

César Alvarito CORONEL GONZÁLEZ[1]
Lohana Mariella LEMA MORETA[2]

Resumen

El incremento de información generado a través de dispositivos fijos y móviles, expuesta a riesgos y ataques, requiere de mecanismos de encriptación que se adapten a las nuevas exigencias, de ahí la importancia de escoger un algoritmo de encriptación adecuado a nuevas estructuras para proteger la información. El objetivo de este artículo es determinar el nivel de seguridad y rendimiento de los algoritmos criptográficos Present, Clefia, Keccak y Hight a través de una revisión sistemática de literatura que aborda estos temas. La información obtenida determinó que tres cifradores de bloque cuentan con características similares, De ellos Clefia es el que mayor número de ataques resiste, un 50 % comparado con los demás, y salvó Hight los algoritmos tienen diferentes versiones en las que varía su longitud de clave, tamaño de bloque, número de rondas y tamaño de implementación.

Palabras clave: Rendimiento, Nivel de seguridad, Algoritmos Criptográficos ligeros, Revisión sistemática.

Abstract

The increase of information generated through mobile and fixed devices and that is exposed to risks and attacks requires of encryption mechanisms that can be adjusted to the new requirements. Therefore there is the importance of choosing an adequate encryption algorithm that can be adapted to the new structures in order to protect the information. The objective of the present article is to determine the security level and performance of cryptographic algorithms such as Present, Clefia, Keccak, and Hight through a systematic revision of the literature that abords these topics. The obtained information determined that three block ciphers have similar characteristics. Among them, Clefia is stronger than the others because it resists about 50% more of attacks compared with the other algorithms. Finally, with exception of Hight the other algorithms have different versions where the key length, block size, number of rounds and implementation size is variable.

Key words Performance, Security Level, Lightweight Cryptographic algorithms, Systematic revision.

[1] Estudiante de Maestría en Auditoría de Tecnología de Información, Universidad Espíritu Santo – Ecuador. E-mail cecoronel@uees.edu.ec.
[2] Master en Ingeniería del Software. Gerente de Proyectos Software Factory Universidad Espíritu Santo- Ecuador.

INTRODUCCIÓN

El constante desarrollo tecnológico, la adopción de nuevas tecnologías, la aparición de la Internet de las Cosas, entre otras áreas, han incrementado de forma exponencial la cantidad de dispositivos fijos y móviles que hoy en día se encuentran conectados entre sí generando gran cantidad de información. (Evans, 2017). De la mano con este incremento está la cantidad de ataques desarrollados para robar información de estos dispositivos y sistemas, por lo que la necesidad de preservar la confidencialidad e integridad de la información ha derivado en el desarrollo de múltiples técnicas y algoritmos de encriptación. Sin embargo, muchos son los limitantes a la hora de implementarlos; características tales como el tamaño físico del dispositivo, la capacidad de procesamiento, y sobretodo el nivel de seguridad que la información requiere, son los principales obstáculos a vencer durante el diseño y, además representan las principales características a considerar al momento de escoger un algoritmo de encriptación para su implementación. La adopción de Sistemas de Encriptación Avanzada AES (*Advanced Encryption System*) es una técnica que aplica un esquema de bloques en criptografía simétrica, lo cual se presenta como una alternativa que en la mayoría de sistemas ha cubierto la necesidad de cifrado gracias a su resiliencia ante una gran variedad de ataques; más sin embargo, dispositivos muy pequeños tales como *tags,* RFID (*Radio Frequency IDentification)* y sensores son incapaces de adoptar este sistema debido a su reducido tamaño y la limitada capacidad de procesamiento que poseen. (Bogdanov, y otros, 2007)

Los algoritmos criptográficos ligeros surgen entonces como una alternativa para cubrir las necesidades de estos sistemas embebidos. Estos tratan de buscar un equilibrio entre los recursos necesarios para su implementación, desempeño, y sobretodo el nivel de seguridad que ofrecen a la información (Sufyan Salim Mahmood & Imad, 2012). Present, Clefia, Keccaf, Hight, son algoritmos que utilizan el cifrado de bloques e introducen las técnicas de sustitución y permutación para la encriptación (Salas, 2007).

En la siguiente sección se revisa el marco teórico donde se abordará la criptografía y los algoritmos ligeros Present, Clefia, Keccaf; posteriormente a través de una revisión sistemática de varios autores se realiza el análisis de resultados y en la sección final se presentan las conclusiones de esta investigación.

MARCO TEÓRICO

Criptografía

La criptografía tiene sus inicios en los años 1900 A.C en Egipto, donde se hallaron los primeros jeroglíficos que contenían variaciones en su escritura que denotaban una serie de transformaciones del texto original. Luego en el año 100 A.C aparece uno de los cifradores más famosos utilizado por Julio Cesar, llamado Cifrador Cesar, el cual usaba la técnica de sustitución para ocultar sus mensajes. El siguiente cifrador conocido es el llamado Vigenere, y ya en el siglo XIX aparece la famosa maquina Enigma, usada durante la Primera Guerra Mundial. A partir de entonces muchas entidades se han dedicado a la creación de nuevos y más robustos sistemas de encriptación (Red Hat, 2017). En (Granados, 2006) se puede encontrar una definición de la criptografía (del griego Krypto y logos, significa el estudio de lo oculto, lo que no se conoce) como la ciencia encargada de buscar métodos capaces de transformar mensajes legibles en ilegibles para su transporte. En conclusion, la criptografía busca métodos que le permitan cifrar los mensajes con códigos secretos que protejan la información durante su transporte, teniendo presente cuatro parámetros importantes: Confidencialidad, Autenticidad, Integridad y Disponibilidad.

Clasificación de la Criptografía.

En la figura 1 se resume la clasificacion de la criptografia, misma que comprende la criptografía clasica, que fue usada hasta la mitad del siglo XX, y la moderna que se divide en simetrica o de clave privada y asimetrica o de clave pública. En la criptografia de clave privada la misma clave es usada tanto en el proceso de encriptacion como en el de desencriptacion. En la criptografia de clave publica se utilizan dos claves, una clave pública para cifrar y otra privada para descifrar. Finalmente la simétrica por la cifra que produce

se clasifica en criptografía en bloque y criptografía en flujo (Granados, 2006).

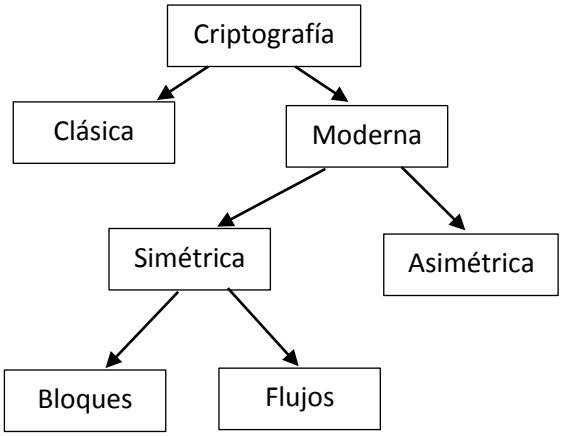


Figura 1. Clasificación de la Criptografía.

Fuente: (Paredes, 2006)

Crifrado en bloque

En la figura 2 se observa el cifrado en bloques, usa una estructura tipo Feistel en la que agrupa los símbolos o caracteres en bloques, para luego transformarlos a traves del uso de funciones y operaciones XOR. Los bloques se cifran sin interesar su posición en el texto ni los bloques adyacentes, por lo que un bloque se puede repetir a lo largo de un mensaje. Los mensajes iguales, cifrados con la misma clave, generan criptogramas iguales (Salas, 2007). Algunos algoritmos como: Present, Clefia, Keccak y Hight que analizaremos en este documento utilizan el cfirado en bloque.

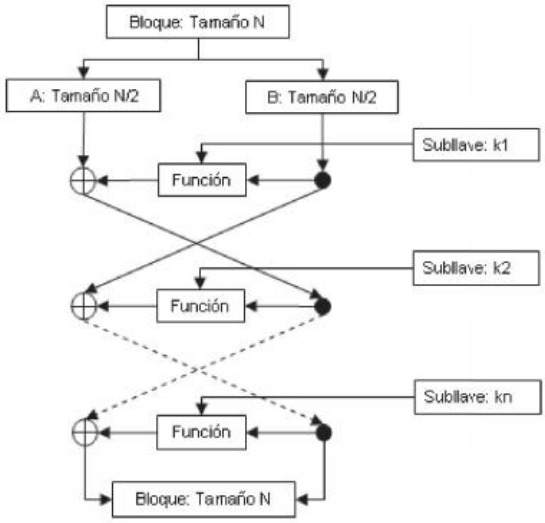


Figura 2. Cifrado por bloques de Feistel.

Fuente: (Paredes, 2006)

Present

Es un algoritmo ligero orientado a ser usado en dispositivos pequeños tales como tags RFID. Fue presentado en el 2007 por Bogdanov et al. Emplea una red de sustituciones y permutaciones en su estructura y contiene 31 rondas. La longitud del bloque es de 64 bits y la longitud de las claves puede ser de 80 y 128 bits. Su diseño está enfocado en su simplicidad, ya que su principal objetivo es lograr su implementación en entornos con recursos limitados. Dentro de las consideraciones del diseño está el hecho de que el nivel de seguridad que demanda es moderado, por lo tanto representa un nivel que podría considerarse adecuado para aplicaciones que no requerirán cifrado de largas cantidades de información (Bogdanov, y otros, 2007).

Clefia

Fue desarrollado en el 2007 por Sony Corporation y la Universidad de Nagoya. Consta de bloques de 128 bits y claves con longitudes de 128, 192 y 256 bits. La estructura en la cual se basa es de tipo Feistel, una característica importante es la implementación de un mecanismo de difusión y transposición (DSM) que brinda mayor inmunidad a varios tipos de ataques. (Sony Corporation, 2007). La estrategia de su diseño es tener un equilibrio en tres funciones fundamentales y necesarias para

cifrados como: seguridad, velocidad y costo de implementación (Sony Corporation, 2010). Finalmente el software de Clefia está dentro de los cifradores de bloque más rápido (Christina Boura, María Naya-Plasencia, Valentín Suder, 2014)

Keccak

Morawiecki & Srebrny (2010) indican que Keccak es una familia de funciones hash cuya construcción es de tipo esponja. Utiliza un sistema de bloque de 7 permutaciones y tiene dos parámetros principales: la tasa de bits (r) y la capacidad (c). La suma de estos dos parámetros indica el tamaño en el que opera Keccak, que usualmente es de 1600 bits. Los valores de la tasa de bits y la capacidad dan el equilibrio entre la velocidad y la seguridad, una mayor tasa de bits hace que la función tenga mayor rapidez pero menos seguridad. (José Luis Gómez Pardo, Carlos Gómez Rodriguez, 2013) .

Revisado Keccak se dispersa de las características de los algoritmos Present, Clefia y Hight.

Hight

Este algoritmo desarrollado en el 2006 por Hong et al. es un cifrador de bloque basado en la estructura de Feistel. Cuenta con bloques de 64 bits de longitud y claves de 128 bits. Su diseño fue enfocado para ser de bajo costo, bajo consumo de energía, ultra ligero, y orientado a hardware. La cantidad de compuertas equivalentes para la implementación física del algoritmo es de 3048 y es mucho más rápido que AES (Hong, y otros, 2006). Kwon et al. (2008) indican que Hight está orientado a un procesador de 8 bits por lo que puede ser implementado en los CPUs incrustados en los sensores WSN (*Wireless Sensor Networks)*, mostrando un rendimiento eficiente en tal ambiente.

Estos algoritmos, y muchos más, han sido desarrollados para su uso dentro de entornos con recursos muy limitados, por lo que el objetivo de este estudio es la realización de una revisión sistemática sobre la literatura existente relacionada con los algoritmos antes mencionados, para conocer las ventajas y desventajas de cada uno al optar por alguno de ellos como parte de la seguridad de un sistema.

METODOLOGÍA.

En una revisión sistemática se desarrolla una exhaustiva revisión de las investigaciones y trabajos realizados hasta la fecha sobre el tópico de interés. En el desarrollo del artículo se utiliza la literatura establecida por (Kitchenham, 2004) para la revisión sistemática, ya que es el mayormente utilizado en desarrollo de software. En la figura 3 se puede observar las etapas de la revisión sistemática y a continuación se definen los pasos para ejecutar esta revisión.

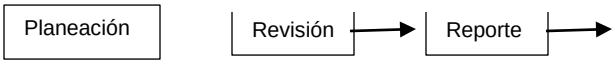


Figura 3. Etapas de la revisión sistemática.

Fuente: (Kitchenham, 2004)

Revisión Sistemática

Problema

La Criptografía es una herramienta importante dentro de la seguridad informática, en este contexto se han desarrollado varios algoritmos criptográficos y en la actualidad los algoritmos ligeros son muy utilizados; sin embargo a la hora de escoger uno de ellos no se conoce de evidencia científica que compare las fortalezas que presentan. Es por eso que se plantea una revisión sistemática de los algoritmos ligeros Present, Clefia, Keccak y Hight, que permita comparar sus estructuras y establecer sus características de rendimiento y nivel de seguridad a la hora de optar por uno de ellos.

Preguntas de Investigación

Q1. ¿Cuáles son características de los algoritmos criptográficos ligeros Present, Clefia, Keccak y Hight, que permitan identificar su rendimiento? Q2. ¿Qué características de

encriptación permiten comparar el nivel de seguridad de los algoritmos criptográficos ligeros Present, Clefia, Keccak y Hight?

Intervención

Estructura, rendimiento y nivel de seguridad de los algoritmos criptográficos ligeros Present, Clefia, Keccak y Hight.

Control

No se dispone de datos iniciales para la revisión sistemática.

Efecto

Factores que permitan comparar la estructura, rendimiento y nivel de seguridad de los algoritmos Present, Clefia, Keccak y Hight.

Medida de resultado

Número de factores identificados para la medición del rendimiento de los algoritmos. Número de factores identificados para medir la seguridad de los algoritmos. Características de la estructura interna de cada algoritmo.

Población

Publicaciones relacionadas con los algoritmos criptográficos ligeros Present, Clefia, Keccak y Hight.

Aplicación

Organizaciones que no tengan implementados algoritmos criptográficos o necesiten mayores exigencias a los ya implementados de acuerdo a su rendimiento y nivel de seguridad. Investigadores que trabajan con algoritmos criptográficos.

Definición de los criterios para Selección de Fuentes y lenguaje de estudio:

Para este proceso se ha definido un conjunto de motores de búsqueda de publicaciones científicas, así como cadenas de búsqueda que servirán para encontrar la información deseada la cual será filtrada según los criterios de inclusión y exclusión que se definirán a continuación. De esta forma los documentos que se obtengan al final serán los que presenten la mayor cantidad de información relevante a este tema de estudio.

Se considerara literatura en inglés o español.

Motores de búsqueda

De acuerdo a la relevancia, calidad de publicaciones e impacto, se determinan los siguientes motores de búsqueda para la revisión.

Tabla 1
Listado de motores de búsqueda

Motores de búsqueda	
ACM Digital Library	**https://www.acm.org/**
IEEE Computer Science Digital Library	**http://www.computer.org/**
Springer Link	**http://www.springerlink.com/**
Science@Direct	**http://www.sciencedirect.com/**
Software Engineering Institute	**https://www.sei.cmu.edu/**
Google Scholar	**http://scholar.google.com**

La mayor parte de estos motores permite la descarga de información de forma gratuita.

Cadenas de búsqueda: Las cadenas de búsqueda (*string search*) que se usaron fueron definidas con relación a las preguntas introducidas al inicio de esta sección, y a las palabras claves definidas en el resumen del presente documento.

- Block Cipher Present AND Performance OR Security,
- Block Cipher Clefia AND Performance OR Security,
- Block Cipher Keccak AND Performance OR Security,
- Block Cipher Hight AND Performance OR Security

Selección de literatura relevante

Definición de criterios de inclusión y exclusión de los estudios: Para determinar

aquella literatura que puede contribuir al objetivo de este estudio se definió los siguientes criterios de inclusión y exclusión, que permiten reducir la cantidad de documentos a analizar:

Tabla 2
Criterios de inclusión y exclusión a ser aplicados

Nº	Criterio de inclusión (CI) y exclusión (CE)
CI_1	Incluir artículos cuyo título esté relacionado con los algoritmos Present, Clefia, Keccaf y Hight
CI_2	Incluir artículos cuyo título esté relacionado con las palabras definidas en la cadena de búsqueda (rendimiento y seguridad)
CI_3	Incluir artículos después de la lectura del resumen
CI_4	Incluir artículos luego de su lectura parcial o total
CE_1	Excluir artículos duplicados
CE_2	Excluir artículos que no denoten características de rendimiento y seguridad

Definición de tipos de estudios

Al inicio se tomará todos los artículos relacionados con algoritmos criptográficos ligeros Present, Clefia, Keccak y Hight, posteriormente se seleccionara los que nos permitan identificar características de rendimiento y nivel de seguridad.

Procedimientos para la selección de estudios

El primer criterio a tener en cuenta es el título del artículo, pero en algunos casos si no se tiene la información necesaria, se efectuará la lectura del resumen del artículo relacionado con la investigación, finalmente luego de una selección se hará la lectura completa de los artículos de estudio principales. Esto se encuentra resumido en la tabla 3.

Extracción de la información:

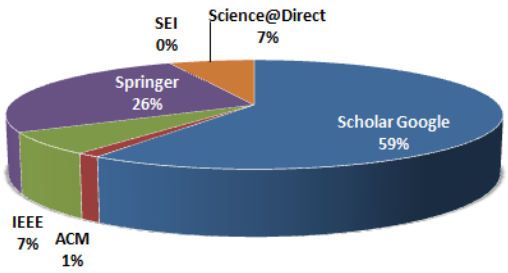
Luego de realizar la búsqueda en los sitios mencionados se obtiene que muchos de los resultados que arroja el buscador Scholar Google se repiten en los demás buscadores. En esta etapa se han recopilado todos los documentos de libre acceso y se han seleccionados aquellos que servirán como base para el estudio (primarios):

Tabla 3
Resultados obtenidos en Search Engines

Search Engine	Fecha de Busqueda	Totales Obtenidos	Artículos Relevantes	Primarios
Scholar Google	13/11/2017	45	31	17
ACM	13/11/2017	1	0	0
IEEE	14/11/2017	5	3	1
Springer Link	14/11/2017	20	19	2
Science @Direct	13/11/2017	5	3	1
SEI	14/11/2017	0	0	0
Totales:		78	27	21

En el gráfico 1 se puede observar el porcentaje de información encontrada en cada motor de búsqueda, donde Scholar Google es de donde más artículos se pudieron extraer.

Grafico 1: Porcentaje de artículos encontrados por buscador



Una vez realizada la recopilación de estos artículos se procede a aplicar los criterios de inclusión y exclusión definidos para identificar de mejor forma los artículos que más aportan a este estudio sistemático. La antigüedad de los artículos data desde el 2006. Esto se debe a que de los 4 algoritmos escogidos, el más antiguo es High que fuera presentado en ese año, por lo tanto toda la documentación anterior a esta fecha no es relevante en este estudio.

La cantidad de artículos seleccionados luego de aplicar los criterios de inclusión y exclusión son 21 y su contenido es analizado en la siguiente sección.

ANÁLISIS DE RESULTADOS

La mayoría de los artículos encontrados son producto de la investigación realizada por universidades de Europa y Asia, siendo notable que la mayoría de los trabajos correspondan a investigaciones en conjunto entre más de un instituto o universidad. Los resultados en su mayoría son experimentales, producto de la implementación de algoritmos y simulaciones para llegar a sus objetivos. A continuación se presentan algunos aspectos que se encontraron en los artículos que permiten realizar una mejor comparación de cada uno de los algoritmos de encriptación.

Para evaluar el desempeño de los algoritmos criptográficos se pueden considerar diferentes aspectos, como los definidos en (Jorstad & Smith, 2017), entre ellos destacan la longitud de la clave usada para encriptación/desencriptacion, la cantidad de ataques a los cuales es resistente y la cantidad de recursos necesarios para su implementación.

Longitud de la clave

Según lo expuesto en (Jorstad & Smith, 2017) un sistema criptográfico es más seguro mientras más larga sea la clave usada ya que así es más resistente a un ataque de fuerza bruta. La longitud de la clave esta expresada en N bits, por lo que la cantidad de posibilidades es igual a 2^N.

Tabla 4
Listado de artículos primarios

1	A Survey of Lightweight-Cryptography Implementations, 2007, Eisenbarth, T.; Kumar, S.; Poschmann, A.; Paar,C.; Uhsadel, L.
2	SAT-based preimage analysis of reduced KECCAK hash functions, 2010, Morawiecki, P.; Srebrny,M.
3	A Statistical Saturation Attack against the Block Cipher PRESENT, 2009, B. Collard, F.-X. Standaert
4	An Enhanced Differential Cache Attack on CLEFIA for Large Cache Lines, 2011, Chester Rebeiro, Rishabh Poddar, Amit Datta, and Debdeep Mukhopadhyay
5	Saturation Attack on the Block Cipher HIGHT, 2009, Peng Zhang1, Bing Sun1, and Chao Li
6	Comparing Performance of Software CLEFIA to Established Block Ciphers on 8-bit Devices, 2011, Rembrand van Lakwijk
7	HIGHT: A New Block Cipher Suitable for Low-Resource Device, 2006, Deukjo Hong, Jaechul Sung, Seokhie Hong, Jongin Lim, Sangjin Lee, Bon-Seok Koo, Changhoon Lee, Donghoon Chang, Jesang Lee, Kitae Jeong, Hyun Kim,
8	Impossible Differential Cryptanalysis of CLEFIA, 2008 Yukiyasu Tsunoo, Etsuko Tsujihara, Maki Shigeri, Teruo Saito,Tomoyasu Suzaki, Hiroyasu Kubo
9	Improved Cache Trace Attack on AES and CLEFIA by Considering Cache Miss and S-box Misalignment, 2010, Xin-jie ZHAO, Tao WANG
10	Improved Impossible Di®erential Cryptanalysis of CLEFIA, 2007, Wei Wang, Xiaoyun Wang
11	Improved Integral Attacks on Reduced-Round CLEFIA Block Cipher, 2011, Yanjun Li, Wenling Wu, Xiaoli Yu, and Le Dong
12	Keccak sponge function family main document, 2010, Guido Bertoni, Joan Daemen, Michael Peets, Gilles Van Assche
13	Lightweight Block Ciphers Revisited:Cryptanalysis of Reduced Round PRESENT, and HIGHT, 2009,Onur ¨ Ozen, Kerem Varıcı, Cihangir Tezcan, and Celebi Kocair
14	Lightweight Block Ciphers: a Comparative Study, 2012,Sufyan Salim Mahmood AlDabbagh ,a, Imad Al

Shaikhli

15 Preimage attacks on the round-reduced Keccak with the aid of differential cryptanalysis, 2013, Pawe l Morawiecki, Josef Pieprzyk, Marian Srebrny, and Micha l Straus

16 PRESENT: An Ultra-Lightweight Block Cipher, 2007, A. Bogdanov, L.R. Knudsen, G. Leander, C. Paar, A. Poschmann, M.J.B. Robshaw, Y. Seurin, and C. Vikkelsoe

17 Automatic Security Evaluation and (Related-key) Dfferential Characteristic Search: Application to SIMON, PRESENT, LBlock, DES(L) and Other Bit-oriented Block Ciphers, 2014, Siwei Sun, Lei Hu, Peng Wang, Kexin Qiao, Xiaoshuang Ma, Ling Song

18 The 128-bit Blockcipher CLEFIA (Extended Abstract), 2007,Taizo Shirai, Kyoji Shibutani, Toru Akishita, Shiho Moriai, and Tetsu Iwata

19 The 128-bit Blockcipher CLEFIA, Security and Performance Evaluations, 2007, Sony Corporation

20 A Lightweight Implementation of Keccak Hash Function for Radio-Frequency Identification Applications, 2010, Elif Bilge KavunTolga Yalcin

21 Multiple Bytes Differential Fault Analysis on CLEFIA, 2010 Xin-jie ZHAO, Tao WANG, Jing-zhe GAO

En cuanto al número de bits que usan en la clave de encriptación, Present implementa dos versiones con claves de 80 y 128 bits (Bogdanov, y otros, 2007), Hight utiliza claves de 128 bits (Hong, y otros, 2006), y Clefia utiliza claves de 128, 192 y 256 bits (Sony Corporation, 2007). En el caso de Keccak al ser un algoritmo de tipo hash no usa clave de encriptación ya que se ejecuta en un solo sentido, esto es, no se puede reversar desde valor obtenido al mensaje original. De esto se puede concluir que Clefia presenta más seguridad en términos de clave de encriptación ya que cuenta con una versión de hasta 256 bits.

Resistencia a ataques

Para evaluar la resistencia de cada algoritmo a los diferentes ataques, los artículos presentan una serie de algoritmos desarrollados para forzar versiones reducidas (menor número de rondas) de cada algoritmo. La razón de esto es que el romper algoritmos de encriptación, dependiendo de la arquitectura y complejidad de los mismos, es un trabajo que consume muchos recursos computacionales, por ejemplo para romper Present se tiene un registro de haber logrado un ataque exitoso hasta en 15 rondas con un total de $2^{35.6}$ pares de texto plano y cifrado (Collard & Standaert).

Tabla 5
Resumen de cantidad de Ataques que resiste cada algoritmo

Algoritmos	Cantidad de ataques conocidos que resiste	Porcentaje
High	11	27,5 %
Clefia	20	50 %
Keccak	3	7,5 %
Present	6	15 %
Total de ataques	40	100 %

Existe una gran variedad de ataques diseñados para romper los algoritmos criptográficos, algunos usan partes del texto original o el texto encriptado para descubrir la clave de encriptación usada, mientras que otros explotan características físicas como el consumo energético, el tiempo de ejecución, etc. para determinar el tipo de algoritmo y la clave usada. En (Houwen) (Gordon, 2015) (KIZHVATOV, 2011) se pueden identificar hasta 24 tipos de ataques que han sido usados para comprometer los sistemas. De los artículos obtenidos se pudo extraer el análisis de resistencia de los algoritmos a varios tipos de ataques, que se resumen a continuación:

Según (Bogdanov, y otros, 2007), (Eisenbarth, Kumar, Poschmann, Paar, & Uhsadel), (Özen, Varıcı, Tezcan, & Kocair) (Wang M. , 2008), la estructura de Present resiste los siguientes ataques:

Tabla 6
Ataques que resiste Present

Ataque
Differential Attacks
Linear Attacks
Algebraic Attacks
Structural Attacks
Key Schedule Attacks
Related Key Attacks

Entre las debilidades del algoritmo destaca el hecho que la capa de difusión presenta

problemas al momento de redistribuir los bits de salida hacia otras S-box diferentes de las iniciales, por lo que se puede determinar los bits de entrada a las S-box en las siguientes rondas y con la aplicación de ciertos algoritmos se pueden determinar las claves usadas en el proceso. En los ataques realizados se registra éxito hasta en 15 de las 31 rondas del algoritmo (Collard & Standaert) y en (Sun, y otros, 2014) se analiza la resistencia de este algoritmo para ataques de tipo related-key differential attack donde se establece que la probabilidad de encontrar una característica diferencial que permita romper el sistema PRESENT-80 tiene una probabilidad de 2^{-60}.

Por otra parte, los estudios sobre Hight revelan que es resistente a los siguientes ataques, según (Deukjo, y otros, 2006), (Özen, Varıcı, Tezcan, & Kocair) (Zhang, Sun, & Li, 2009):

Tabla 7
Ataques que resiste Hight

Ataque
Linear attacks
Differential attacks
Truncated Differential
Impossible differential Cryptanalysis
Saturation Attacks
Boomerang Attack
Interpolation and Higher Order Differential Attack
Algebraic Attack
Slide and Related Key Attack

El ataque más fuerte que se puede encontrar en estos documentos es Related Key Differential Attack en 31 rondas.

Clefia ha sido bastante estudiado en cuanto a su seguridad se refiere y se pueden encontrar referencias de esto en (Sony Corporation, 2007) (Shirai, Shibutani, Akishita, Moriai, & Iwata) (Rebeiro, Poddar, Datta, & Mukhopadhyay, 2011) (Tsunoo, y otros) (Wang & Wang) (ZHAO & WANG, Improved Cache Trace Attack on AES and CLEFIA by Considering Cache Miss and S-box Misalignment) (Li, Wu, Yu, & Dong, 2015). Aquí se puede encontrar que Clefia, es resistente los siguientes ataques:

Tabla 8
Ataques que resiste Clefia

Ataque
Differential Attacks
Truncated differential attacks
Linear Attacks
Truncated linear attack
Impossible Differential attack
Integral attack
Saturation attack
Algebraic Attack
Related Key Attacks
Boomerang Attacks
Amplified Boomerang Attacks
Rectangle Attack
Key recovery attack
Gilbert-Minier Collision Attack
High order Differential, Interpolation Attack
XSL Attack
Chi square Attack

En (Rebeiro, Poddar, Datta, & Mukhopadhyay, 2011) se describen algunas formas de atacar a este algoritmo, y cierta debilidad cuando este es implementado en sistemas con caches de gran tamaño. Sin embargo no brinda un dato específico sobre si se ha logrado romper el algoritmo con las propuestas presentadas. Adicional a esto en (ZHAO, WANG, & GAO, Multiple Bytes Differential Fault Analysis on CLEFIA) se expone que al tener una estructura tipo Feistel el sistema es débil ante ataques de tipo multiple byte faults.

En cuanto a Keccak, este presenta fortaleza frente a los siguientes ataques (Morawiecki1, Pieprzyk, Srebrny, & Straus):

Tabla 9
Ataques que resiste Keccak

Ataque
Side Channel Attack
Collision Attack
Preimage Attack

En (Pawed Morawiecki, Marian Srebrny, 2010) se ha estudiado el ataque de preimagen donde solamente se ha tenido éxito en versiones muy

reducidas de Keccak con pocas rondas y pocos bits de entrada.

Tamaño de implementación del algoritmo

Dentro de los artículos primarios descritos anteriormente se han encontrado algunas propuestas para la implementación física de los algoritmos, donde se han experimentado algunas arquitecturas y opciones para reducir el área de implementación y número de compuertas lógicas. La mayoría de implementaciones se hacen sobre procesadores de 8 bits debido a sus características como tamaño, consumo energético, simplicidad, etc, que los hacen ideales para entornos limitados como los sensores y tags para los cuales los algoritmos objetos de este estudio han sido diseñados

Present: La cantidad de compuertas lógicas equivalentes requeridas para su implementación es igual a 1570 para la versión de 80 bits y 1884 para la versión de 128 bits. (Mahmood AlDabbagh & Al Shaikhli, 2012)

Hight: La única versión de 128 bits requiere de un área de implementación igual a 2608 compuertas. (Sufyan Salim Mahmood & Imad, 2012)

Clefia: En la versión de 128 bits el área de implementación del algoritmo es igual a 5979 compuertas equivalentes, para 192 bits es 8536 compuertas equivalentes y para la versión de 256 bits el tamaño es igual a 8482 compuertas. (Sufyan Salim Mahmood & Imad, 2012).

Keccak: Dependiendo de la versión del algoritmo que se use, sus requerimientos en cuanto a compuertas lógicas equivalentes para la implementación varían dependiendo de la tasa de bits requerida y la capacidad ya que el número de rondas es directamente proporcional a estos parámetros. Por ejemplo en el caso de Keccak-f{r=1024, c=576} se requieren hasta 19K GE`s, y en la versión Keccak-f{200} y Keccak-f{400} se han utilizado hasta 2.52K GE`s (Yalcin & Bilge Kavun, 2010) (Bertoni, Daemen, Peeters, & Van Assche). En este caso el tamaño ira en función de los requerimientos del sistema, recordando que existe un balance entre la tasa de bits que se puede usar, la capacidad, y el nivel de seguridad ya que mientras mayor es el bit-rate-tasa de bits, más rápido es el algoritmo pero su seguridad se ve disminuida.

En la tabla 10 podemos observar las principales características de los algoritmos ligeros de encriptación Present, Clefia y High.

CONCLUSIONES

Los algoritmos criptográficos ligeros han sido diseñados para satisfacer las necesidades de seguridad en dispositivos pequeños tales como sensores, tags de radiofrecuencia, dispositivos médicos, etc., cuya utilización se ha visto altamente incrementada con la aparición de tecnologías como Internet de las Cosas. Lo que motiva a desarrollar y usar estos algoritmos es la necesidad de preservar la seguridad, confidencialidad y disponibilidad de la información que generan y reciben ante el incremento de ataques para comprometer y robar la información.

Tabla 10
Resumen de características de los algoritmos ligeros de encriptación

Características	Clefia	High	Present
Longitud de Clave (bits)	128/ 192/ 256	128	80/128
Tamaño de Bloque (bits)	128	64	64
Numero de rondas	18/22/26	32	31
Tamaño de implementación (GE`s)	5979/ 8536/ 8482	2608	1570/1884

Sin embargo mientras más pequeños son los dispositivos, mayores son las limitaciones para

la implementación de estos sistemas y esto va de la mano con el nivel de seguridad requerido. Los algoritmos ligeros con niveles más altos de seguridad requieren mayores áreas para la implementación de compuertas lógicas y demandan mayor consumo energético. Esto constituye uno de los principales trade-offs ya que se debe buscar un equilibrio entre el nivel de seguridad, tamaño y consumo de recursos según el tamaño del dispositivo.

Entre los algoritmos estudiados en este artículo y que cumplen la característica de ser ligeros están Present, Hight y Clefia, siendo Present el algoritmo con menor resistencia a ataques, según la literatura revisada, por lo que se puede considerar a este como un algoritmo de seguridad moderada a baja. A la par con esta característica está que el área requerida para su implementación es la menor de los tres algoritmos mencionados, por lo que lo hace un candidato ideal para dispositivos con altas limitaciones en cuanto a espacio físico, consumo de energía, y sobretodo donde no se requiera alta seguridad y la cantidad de información a ser procesada sea baja.

Hight es el segundo en cuanto a seguridad se refiere. La cantidad de ataques que puede resistir es considerablemente mayor a los de Present, sin embargo debido a su estructura, el área de implementación es mucho mayor, con 2608 compuertas equivalentes, lo que implica un mayor consumo de energía.

Clefia, por otro lado, se presenta como un candidato fuerte ante al menos 19 ataques conocidos de los cuales se tiene registro en la literatura estudiada, por lo que es el candidato ideal a ser implementado cuando el nivel de seguridad requerido es muy alto. De la mano con su alta resistencia a ataques está también el tamaño de implementación, que en su versión de 256 bits es igual a 8482 compuertas.

Keccak, por otro lado, es un algoritmo de tipo hash, cuya función es generar un digest o cadena de valores según el texto de entrada. Sin embargo este algoritmo no es reversible por cuanto no se puede obtener el texto original partiendo desde el valor cifrado. La utilidad de este tipo de algoritmos está en que permite generar un valor o identificador para una cadena de información, de esta forma se puede validar que la información ingresada sea la correcta comparándola con el valor hash o digest guardado en las bases de datos. Estos algoritmos también son susceptibles a ataques como el ataque de colisión, en el cual dos cadenas de datos distintas pudieran generar el mismo valor hash, sin embargo para Keccak se ha comprobado que el algoritmo es lo suficientemente fuerte para resistir varios ataques.

De la presente revisión se puede concluir entonces que los algoritmos ligeros de bloque descritos presentan diferentes niveles de seguridad y de la mano con la seguridad está el tamaño y consumo que presentan. Para la industria la implementación de uno u otro algoritmo en sus dispositivos implica variación de costos, que son directamente proporcionales a los recursos necesarios para cada algoritmo, y la aplicación que se dará al dispositivo. Adicionalmente la información encontrada en cuanto al nivel de seguridad, permite apreciar el gran interés de la comunidad científica ya que se han realizado muchos esfuerzos para intentar vulnerar cada uno de los algoritmos. Los recursos para probar su fortaleza son muy demandantes ya que se requieren de muchas horas de simulación y equipos de computo con grandes capacidades de procesamiento para vulnerar versiones muy reducidas de cada algoritmo, sin embargo este estudio es un gran apoyo para conocer en la práctica el desempeño de cada uno según la aplicación a la cual se dedique.

Bibliografía

Bertoni, G., Daemen, J., Peeters, M., & Van Assche, G. (s.f.). *Keccak sponge function family main document.* Obtenido de Team Keccak: http://keccak.noekeon.org/

Bogdanov, A., Knudsen, L., Leander, G., Paar, C., Poschmann, A., Robshaw, M., . . . C. (2007). PRESENT: An Ultra-Lightweight Block Cipher. *CHES 2007*.

Christina Boura, María Naya-Plasencia, Valentín Suder. (26 de 09 de 2014). Obtenido de https://hal.inria.fr/hal-01068894/PDF/699.pdf

Collard, B., & Standaert, F. (s.f.). *A Statistical Saturation Attack against the block cipher PRESENT.* Obtenido de Research Gate: https://www.researchgate.net/publication/221208266_A_Statistical_Saturation_Attack_against_the_Block_Cipher_PRESENT?enrichId=rgreq-85c2c044c42d19e1403ae1fdf8dc624a-XXX&enrichSource=Y292ZXJQYWdlOzIyMTIwODI2NjtBUzoxMzAxMzAyNDAzNDgxNjBAMTQwODAzNjg3NjEzOA%3D%

Deukjo, H., Jaechul, S., Seokhie, H., Jongin, L., Sangjin, L., Bon-Seok, K., . . . Seongtaek, C. (2006). HIGHT: A New Block Cipher Suitable for Low-Resource Device. *CHES* , 46-59.

Eisenbarth, T., Kumar, S., Poschmann, A., Paar, C., & Uhsadel, L. (s.f.). *IEEEXPLORE.* Obtenido de IEEEXPLORE: http://ieeexplore.ieee.org

Evans, D. (14 de 12 de 2017). *www.cisco.com.* Obtenido de CISCO: https://www.cisco.com/c/dam/en_us/about/ac79/docs/innov/IoT_IBSG_0411FINAL.pdf

Gordon, A. (2015). *Oficial (ISC) Book to CISSP CBK.*

Granados, G. (2006). Obtenido de Revista Digital Universitaria: http://www.ru.tic.unam.mx/tic/bitstream/handle/123456789/1105/511.pdf?sequence=1&isAllowed=y

Hong, D., Sung, J., Hong, S., Lim, J., Lee, S., Koo, B.-S., . . . Chee, S. (2006). HIGHT: A New Block Cipher Suitable for Low-Resource Device. *CHES*, 46-59.

Houwen, J. (s.f.). *Information Security Today.* Obtenido de http://www.infosectoday.com/Understanding_Cryptography/Articles/Attacking_Defending_Cryptosystems.pdf

Jorstad, N., & Smith, L. (27 de 11 de 2017). *National Institute of Standards and Technology.* Obtenido de https://csrc.nist.gov/: https://csrc.nist.gov/csrc/media/publications/conference-paper/1997/10/10/proceedings-of-the-20th-nissc-1997/documents/128.pdf

José Luis Gómez Pardo, Carlos Gómez Rodriguez. (2013). Obtenido de file:///C:/Users/Zona/Downloads/SHA-3.pdf

Kitchenham, B. (Julio de 2004). Obtenido de http://www.inf.ufsc.br/~aldo.vw/kitchenham.pdf

KIZHVATOV, I. (07 de 2011). Physical Security of Cryptographic Algorithm Implementations. *Physical Security of Cryptographic Algorithm Implementations*.

Li, Y., Wu, W., Yu, X., & Dong, L. (2015). *Researchgate.* Obtenido de

https://www.researchgate.net/publication/221239649

Mahmood AlDabbagh, S., & Al Shaikhli, I. (2012). Lightweight Block Ciphers: A Comparative Study. *Journal of Advanced Computer Science and Technology Research*, 159-165.

Morawiecki1, P., Pieprzyk, J., Srebrny, M., & Straus, M. (s.f.). *International Association for Cryptologic Research.* Obtenido de International Association for Cryptologic Research: https://eprint.iacr.org/2013/561.pdf

Özen, O., Varıcı, K., Tezcan, C., & Kocair, Ç. (s.f.). *Springer Link.* Obtenido de https://link.springer.com/chapter/10.1007/978-3-642-02620-1_7

Pawed Morawiecki, Marian Srebrny. (05 de 08 de 2010). Obtenido de https://eprint.iacr.org/2010/285.pdf

R. Vivek, J. Roopchand. (01 de 08 de 2012). Obtenido de http://www.ijcta.com/documents/volumes/vol3issue4/ijcta2012030417.pdf

Rebeiro, C., Poddar, R., Datta, A., & Mukhopadhyay, D. (2011). An Enhanced Differential Cache Attack on CLEFIA for Large Cache Lines. *INDOCRYPT*, 58-75.

Red Hat. (4 de 12 de 2017). Obtenido de Red Hat, Inc.: https://access.redhat.com/blogs/766093/posts/1976023

Salas, J. M. (01 de 12 de 2007). Obtenido de http://itzamna.bnct.ipn.mx/dspace/bitstream/123456789/740/1/Tesis%20Juan%20Manuel%20Guzman%20Salas.pdf

Shirai, T., Shibutani, K., Akishita, T., Moriai, S., & Iwata, T. (s.f.). Obtenido de https://iacr.org/archive/fse2007/45930182/45930182.pdf

Sony Corporation. (2007). Obtenido de https://www.sony.net/Products/cryptography/clefia/download/data/clefia-eval-1.0.pdf

Sony Corporation. (29 de 01 de 2010). Obtenido de http://www.cryptrec.go.jp/english/cryptrec_13_spec_cypherlist_files/PDF/22_00espec.pdf

Sufyan Salim Mahmood, A., & Imad, A. S. (2012). Lightweight Block Ciphers: a Comparative Study. *Journal of Advanced Computer Science and Technology Research Vol.2 No.4,*, 159-165.

Sun, S., Hu, L., Wang, P., Qiao, K., Ma, X., & Song, L. S. (2014). Automatic Security Evaluation and (Related-key) Differential Characteristic Search: Application to SIMON, PRESENT,LBlock, DES(L) and Other Bit-oriented Block Ciphers. *Advances in Cryptology – ASIACRYPT 2014*, 158-178.

Tsunoo, Y., Tsujihara, E., Shigeri, M., Saito, T., Suzaki, T., & KuboHiroyasu. (s.f.). *International Association for Cryptologic Research.* Obtenido de International Association for Cryptologic Research: https://www.iacr.org/archive/fse2008/50860404/50860404.pdf

Wang, M. (2008). Differential Cryptanalysis of Reduced-Round PRESENT. *Progress in Cryptography, AFRICACRYPT*, 40-49.

Wang, W., & Wang, X. (s.f.). *International Association for Cryptologic Research.* Obtenido de International Association for Cryptologic Research: https://eprint.iacr.org/2007/466.pdf

Yalcin, T., & Bilge Kavun, E. (2010). A Lightweight Implementation of Keccak Hash Function for Radio-Frequency Identification Applications. *International Workshop on Radio Frequency Identification: Security and Privacy Issues*, 258-269.

Zhang, P., Sun, B., & Li, C. (2009). Saturation Attack on the Block Cipher HIGHT. *CANS* , 77-86.

ZHAO, X.-j., & WANG, T. (s.f.). *International Association for Cryptologic Research.* Obtenido de International Association for Cryptologic Research: https://eprint.iacr.org/2010/056.pdf

ZHAO, X.-j., WANG, T., & GAO, J.-z. (s.f.). Multiple Bytes Differential Fault Analysis on CLEFIA.