Criptografía ligera: conceptos, avances y tendencias en seguridad para IoT y dispositivos con recursos limitados

Borgo, Martín Alejandro; Molina, Leandro Rodrigo; Reniero, Oscar Isaías

Universidad Nacional de Entre Ríos

Facultad de Ciencias de la Administración

Licenciatura en Sistemas

martinborgo8@gmail.com, LeandroRodrigoMolina@gmail.com, isa.reniero001@hotmail.com

Abstract. La criptografía ligera (Lightweight Cryptography), es una rama de la criptografía que se enfoca en desarrollar algoritmos eficientes y de bajo consumo para su implementación en sistemas con recursos limitados, como sistemas embebidos, dispositivos IoT (Internet de las cosas) y microcontroladores. La ausencia de medidas de seguridad adecuadas en tales sistemas genera preocupaciones, especialmente cuando se trata de gestionar información sensible, en este contexto la criptografía ligera ofrece soluciones que permiten proteger eficazmente los datos y las comunicaciones, sin sacrificar la eficiencia de los recursos. En este artículo se recopilan los distintos conceptos, avances y tendencias que se fueron dando a lo largo de los años en este campo de la criptografía.

Keywords: Criptografía Ligera, Block Ciphers, IoT, Seguridad.

1 Introducción

La criptografía ligera (rama de la criptografía) es uno de los temas de la actualidad que se encuentra en auge. Es usada en dispositivos donde su poder de cómputo es reducido, a este tipo de dispositivos se les conoce como IoT (Internet of Things o Internet de las Cosas), aunque en sus inicios recibía el nombre de red de sensores. Muchos de estos dispositivos utilizan microcontroladores de muy bajo consumo que solo pueden permitirse una pequeña parte de su cómputo a la seguridad[1]. Esto provoca que los algoritmos clásicos de criptografía avanzada no puedan ser usados, debido a la alta latencia y consumo de energía que presentan en estos dispositivos. La criptografía ligera nos brinda una gran variedad de algoritmos "livianos" que han sido diseñados para garantizar confidencialidad, autenticidad e integridad de los datos en los dispositivos IoT. Son algoritmos desarrollados por los ámbitos académicos, agencias estatales o propietarios, los cuales eran empleados en sus inicios, en el área industrial:

> "Un ejemplo común de este uso es el de las redes de sensores. Estas redes tienen como objetivo conectar grandes cantidades de sensores muy simples a un centro principal. Estos sensores funcionan con baterías y/o generarían su propia energía utilizando, por ejemplo, paneles solares. Los algoritmos criptográficos deben usarse en los canales de comunicación entre los sensores y su centro para proporcionar seguridad, autenticidad e integridad de los mensajes. Sin embargo, debido a la muy baja energía disponible y porque la seguridad es un gasto adicional a la funcionalidad real del dispositivo, los algoritmos criptográficos deben ser lo más 'pequeños' posible." (Biryukov & Perrin, 2017, p. 1)

Con el transcurso del tiempo, la aplicación de estos algoritmos ha permeado diversas áreas, incluyendo el sector médico y el ámbito de los dispositivos electrónicos, tales como televisores y lavadoras, entre muchos otros (J. Eterovic et al., 2019; Gunathilake et al., 2019; Thakor et al., 2020). En consecuencia, en los últimos años se ha experimentado un significativo progreso no sólo en la identificación de vulnerabilidades sino también en la corrección de éstas y en la invención de innovadores métodos de encriptación. Este artículo se enfocará en llevar a cabo un análisis preliminar exhaustivo sobre está subárea de la criptografía, abordando integralmente conceptos, progresos y tendencias actuales en el desarrollo de este tipo de algoritmos.

2 Desarrollo de Trabajo

2.1 Estandarización de la Criptografía Ligera: ISO/IEC 18033 y 29192

Debido a la gran relevancia que estaban adquiriendo este tipo de algoritmos de encriptación, en 2011 se estandarizan gran parte de estos a través de la creación de la ISO/IEC 18033, que en sus diferentes incisos quedan descritos y especificados cada uno de los métodos de encriptación que estaban siendo utilizados a nivel industrial, académico y gubernamental hasta día de hoy. Aunque el paso más grande se dio en 2012 con la salida de la ISO/IEC 29192-1:2012 donde se introduce por primera vez el término criptografía ligera, brindando definiciones y estableciendo una serie de requerimientos de seguridad, implementación y clasificación. De la mano de la estandarización vienen un conjunto de ventajas, de las cuales J. E. Eterovic y Cipriano (2018) mencionan las siguientes:

[1]Es importante tener en cuenta que, aunque la criptografía ligera es útil para dispositivos con recursos limitados, los avances en la tecnología a menudo influyen en lo que se considera "ligero". Ya que lo que se considera ligero en un momento dado podría no serlo en un futuro, a medida que la capacidad de los dispositivos mejore.

1. La libre disponibilidad del algoritmo para su uso.
2. Contar con una descripción detallada de las funciones que lo conforman, como así también del diseño en general.
3. Poder realizar una verificación del funcionamiento y conformidad mediante un grupo independiente de expertos.
4. La existencia de vectores de prueba[2] para corroborar el buen funcionamiento de estos.

2.2 Clasificación y Consideraciones en el Diseño

Wehbe et al. (2022) realiza una clasificación de los distintos algoritmos de encriptación, dividiéndolos en 4 grandes grupos:

1. **Stream ciphers (cifrados de flujo):** Es un cifrado simétrico que opera con una transformación que varía con el tiempo en dígitos individuales de texto plano. Una secuencia de texto plano se cifra usando una secuencia pseudoaleatoria, generada a partir de una clave secreta y un parámetro público. Cada dígito cifrado se obtiene combinando el dígito correspondiente de texto plano con esta secuencia.
2. **Block ciphers (cifrados de bloque):** Es un cifrado simétrico que, para una clave específica k, define un algoritmo de cifrado que convierte un bloque de texto plano de n bits en un bloque de texto cifrado de n bits, y un algoritmo de descifrado correspondiente.
3. **Funciones de hashing criptográficas:** Toman cadenas de entrada de longitud arbitraria y las convierten en cadenas de salida de longitud fija y corta, que es única (en teoría) para cada entrada única.
4. **Algoritmos de criptografía asimétrica (Sistemas de clave pública, como RSA y curvas elípticas):** Utiliza claves diferentes para cifrar y descifrar. Una clave puede ser pública, mientras que su contraparte debe mantenerse en secreto.

En la actualidad se proponen muchos algoritmos criptográficos donde sus diseños varían mucho, y cuya única similitud es la baja potencia requerida para su ejecución. La implementación o diseño de estos algoritmos están divididos en dos: hardware y software. Independientemente de eso, al momento de su diseño se debe hacer hincapié en el consumo de memoria, el tamaño de la implementación (código o circuitería requerida) y la velocidad del algoritmo. Lo que sí varía de un tipo de implementación a otra son los parámetros utilizados para indicar su eficiencia. Más allá de esos aspectos, no importa la implementación específica que se haga, ya que un algoritmo de criptografía ligera debe, idealmente, conseguir un equilibrio entre seguridad, rendimiento y costo. A continuación, mencionaremos los aspectos principales en los que se deben enfocar cada uno de los diseñadores e implementadores de acuerdo con el tipo de implementación particular por la que se haya decantado

Si se desea realizar una implementación por hardware, para que este sea eficiente se deben tener en consideración los siguientes puntos:

- Debe minimizar el consumo de memoria, en RAM o el área de puerta en dispositivos con recursos limitados.

[2]Los vectores de pruebas son un conjunto de datos de entrada predefinidos y conocidos, utilizados para probar y verificar el correcto funcionamiento de un algoritmo.

- El tamaño de la implementación debe ser optimizado, es decir, que el diseño debe ser lo más compacto posible para ocupar menos espacio en el chip.
- Debe considerar el acceso a la memoria no volátil y cómo se accede a las claves almacenadas en ella.
- Las operaciones como las permutaciones de bits, que son baratas en hardware, son las más preferidas.
- La estructura de la clave no debe requerir un estado de clave que se actualice en cada ronda, ya que esto sería costoso en términos de área de puerta.

Si la implementación que se desea realizar es de software, para que este sea eficiente se deberá considerar los siguientes puntos:

- Debe minimizar el consumo de memoria, en RAM utilizada.
- Las operaciones como las rotaciones de palabras, que son eficientes en software, pueden ser preferidas.
- Debe considerar la forma en que se accede a las claves y cómo se almacenan en la memoria.
- Las operaciones que son inherentemente costosas en software, como el manejo de bits a nivel individual, deben evitarse o minimizarse.
- Las estructuras de clave que pueden evaluarse .^{en} el momento", es decir, generadas en tiempo real durante la ejecución en lugar de pre-calculadas y almacenadas son preferibles para mantener la eficiencia.

2.3 Algoritmos Criptográficos Ligeros: Tendencias y Desafíos

Biryukov y Perrin (2017) enumeran algunas de las tendencias que se están adoptando en el diseño de nuevos algoritmos criptográficos ligeros, donde se mencionan las operaciones no lineales, operaciones lineales y se discute sobre el esquema de llaves.

Las operaciones no lineales son realmente necesarias en cualquier algoritmo de encriptación, ya que brindan una mayor protección contra ciertos ataques, concretamente los ataques de criptoanálisis[3], los ataques de fuerza bruta[4] y los ataques de canal lateral[5]. Dentro de los mecanismos utilizados en la criptografía ligera se encuentran las tablas de consultas (Look-up Table), las cuales son una estructura de datos que, suele ser implementada a través de cajas de sustitución[6] (S-Boxes), estas se utilizan para realizar operaciones no lineales de manera más eficiente, y debido que pueden ser implementadas utilizando S-Boxes, permiten que los algoritmos que utilizan estas tablas puedan ser eficientemente implementados tanto por hardware como por software. Se menciona también a los algoritmos basados en segmentos de bits, que utilizan S-Boxe de manera distinta, permitiéndole realizar operaciones bit a bit como AND y XOR, en palabras de w bits. Dado que esta tarea requiere un número limitado de operaciones lógicas, los algoritmos que utilizan esta técnica

[3]El criptoanálisis se centra en estudiar y analizar los sistemas criptográficos para poder vulnerar su seguridad, algunas de las técnicas usadas son el reconocimiento de patrones de redundancia de los textos, el averiguar características del algoritmo de cifrado, entre otras.

[4]En los ataques de fuerza bruta se trata de descubrir la clave utilizada para cifrar la información, probando todas las combinaciones posibles, es un mecanismo útil cuando se sabe que las llaves de cifrado son cortas.

[5]Un ataque de canal lateral se centra en el análisis de la información indirecta que se filtra durante el proceso de cifrado o descifrado, con el objetivo de descubrir información sensible o claves de seguridad. Esta información puede ser la variación en el consumo eléctrico, el tiempo de respuesta del dispositivo, entre otros.

[6]Las S-Boxes son funciones booleanas vectoriales no lineales, que toman un número determinado de bits de entrada y los transforman en otro número de bits de salida, que no necesariamente tiene que ser la misma cantidad de bits.

se destacan por su eficiencia en la implementación por software y hardware. Por último, se encuentran los algoritmos basados en ARX (adición, rotación y XOR), que, debido a su simplicidad, eficiencia y a la gran dispersión en los datos que se consigue con ellos, junto con el hecho de que la adición a nivel de software es extremadamente eficiente, hace que los algoritmos basados en ARX se encuentran entre los mejores en rendimiento para microcontroladores.

Las operaciones lineales, por otro lado, tienen un rol fundamental, ya que proveen de difusión a los datos. Entre los mecanismos mencionados por los autores se encuentran las matrices de máxima distancia separable (MDS Matrix), las cuales proveen una dispersión adecuada de los datos, al mismo tiempo que minimiza el costo computacional, aumentando la protección contra criptoanálisis lineales y diferenciales. La permutación de bits es otro mecanismo que está siendo altamente utilizado en ciertas plataformas, debido a que, en términos generales, su implementación por hardware se puede realizar a bajo costo, con el inconveniente que por el momento son un poco costosas de realizarse a nivel software, es por esa razón que son mayormente empleados por dispositivos RFID y similares. Por último, nombran a los algoritmos que se basan en rotación y XOR (RX), estos brindan una difusión apropiada, siendo barato de implementar tanto por software como por hardware, acarreando casi todas las ventajas de los algoritmos basados en ARX, pero con un menor costo.

Los autores finalizan hablando de las claves de encriptación, y debido a su elevado consumo en términos de RAM, hace que los diseñadores opten por implementar estos algoritmos utilizando claves de corta extensión, dejándolos expuestos a ataques de llaves relacionadas[7]y ataques de fuerza bruta. Si bien en la actualidad existen una serie de algoritmos que se jactan de ser inmunes a este tipo de ataques, esto lo consiguen añadiendo más rondas al momento de encriptar los datos o a través de esquemas de llaves más complejos, en ambos casos este tipo de alternativas no es muy preferida a la hora de crear este tipo de algoritmos. Por esta razón nacen los distintos esquemas de llaves, cuya principal función es diversificar y distribuir la clave principal en subclaves que se utilizan en diferentes rondas o etapas del cifrado para mejorar la seguridad y la resistencia a diversos tipos de ataques. Aunque al momento de la implementación se puede emplear este mecanismo de diversas formas:

- Utilizar una variante más sencilla del esquema de llaves para solventar los problema de seguridad referidos a ataques relacionales.
- Proporcionar una mayor seguridad mediante un esquema de llaves mucho más complejo.
- Generar subclaves a partir de una clave maestra, tomando diferentes grupos de bits de la clave maestra.
- Utilizar la función de rotación para actualizar el estado de la clave original.

La tabla 1 ofrece en forma de ejemplo, algunos de los algoritmos más populares que utilizan cada una de las técnicas antes mencionadas. Hay que tener en cuenta que algunos algoritmos pueden estar en más de una categoría debido a que existen diferentes implementaciones de este.

- Tabla de consultas (Look-up Table): TC
- Segmentos de bits (Bit-slice): ASB
- ARX (adición, rotación y XOR): ARX

[7]Los ataques de claves relacionadas son un tipo de ataque criptográfico en el cual se intenta comprometer la seguridad en el cifrado aprovechándose de la relación existente entre las diferentes claves utilizadas en el sistema. Estos ataques se basan en el hecho de que ciertos algoritmos criptográficos pueden exhibir vulnerabilidades cuando se utilizan claves relacionadas de alguna manera específica.

- Matrices de máxima distancia separable (MDS Matrices): MDS
- Permutación de bits (Bit Permutations): PB
- Rotación y XOR (RX): RX
- Utilizan una variante sencilla del esquema de llaves: AV
- Utilizan esquemas de llaves más elaborados: AEE
- Generan subclaves a partir de una clave principal: ASP
- Cambian el estado de las claves a partir de una función de rotación: AR

Mecanismos en el diseño de algoritmos criptográficos ligeros									
Operaciones no lineales			**Operaciones lineales**			**Esquema de llaves**			
TC	**ASB**	**ARX**	**MDS**	**PB**	**RX**	**AV**	**AEE**	**ASP**	**AR**
AES	Noekeon	TEA	AES	PRESENT	3-Way	Noekeon	SEA	XTEA	Noekeon
PRESENT	PRIDE	XTEA	CLEFIA	TWINE	Noekeon	FLY	EPCBC	Noekeon	SEA
Piccolo	Fantomas	HIGHT	LED	LBlock	ITUbee	GIFT	LBlock	HIGHT	EPCBC
PRINCE	RoadRunner	LEA	Zorro	Lilliput			TWINE	LED	SIMECK
	FLY	SPARX		RECTANGLE			SPARX	Fantomas	FLY
	Mysterion			RoadRunner				Midori	SPARX

Tabla 1: Algoritmos que usan mecanismos de diseño

3 Resultados Obtenidos y Esperados

En la sección 2.2 se clasificó todas las formas existentes de encriptación. Si bien todos estos esquemas son utilizados para una amplia gama de propósitos, cuando hablamos de algoritmos ligeros, el esquema asimétrico y las funciones hash suelen ser descartadas, ya que las primeras, en términos generales, son muy costoso y las segundas son empleadas para otro tipo de propósitos como la validación de datos o la construcción de primitivos criptográficos. Es por eso que a la hora de diseñar un algoritmo ligero se opta por los esquemas de encriptación simétrica, es decir, modelos basados en Block cipher o Stream cipher, la tabla 2 muestra las principales diferencias entre estos dos modelos:

En general se utilizan algoritmos basados en Block Cipher cuando el tamaño de los datos es fijo o es conocido de antemano, algunos de sus usos son, por ejemplo, en el protocolo SSL/TLS que utilizan cifrado de bloque para proteger la comunicación entre el cliente y el servidor. Por otro lado, se utiliza un cifrado de flujo cuando los datos a cifrar poseen un tamaño muy variable e imposible de predecir, protocolos Wi-Fi como el WEP o WPA e incluso las comunicaciones satelitales utilizan este tipo de algoritmos. Thakor et al. (2020), menciona otras de las áreas donde se emplean este tipo de algoritmos, entre ellas:

- En dispositivos eléctricos que se utilizan en el hogar, como televisores inteligentes, refrigeradores y otros dispositivos similares, que implementan algoritmos como SIMON, SPECK, Piccolo y TWINE.
- En el ámbito de la logística y la gestión de la cadena de suministro mediante la tecnología RFID (Identificación por Radiofrecuencia), que utilizan algoritmos como SIMON, SPECK, Piccolo y PRINCE.

- En el uso de sensores en la agricultura inteligente para monitorear y controlar aspectos de la producción agrícola, en los cuales se aplican algoritmos como SIMON, SPECK, PRESENT y TWINE.
- En el uso de sensores en el campo de la medicina para la monitorización y el seguimiento de la salud de los pacientes, donde se utilizan algoritmos como SIMON, SPECK, PICCOLO y PRESENT.
- En los sistemas industriales en entornos de fabricación, producción y procesamiento, que hacen uso del algoritmo AES.
- La incorporación de sistemas de seguridad y comunicación en la industria automotriz que utilizan los algoritmos Keeloq, Midori, PRINCE, PRESENT y SIMON[8].

Tipo de Algoritmo	Técnicas de encriptación	Uso de la Llave	Velocidad	Uso de Memoria	Modo de operación	Implementación
Block Cipher	Difusión y Confusión.	Utiliza la misma llave para cifrar todos los bloques.	Es más lento debido a que encripta un bloque de caracteres al mismo tiempo.	Requiere más memoria ya que debe almacenar todo el bloque que se está encriptando.	ECB (Electronic Codebook), CBC (Cipher Block Chaining), PCBC (Propagating Cipher Block Chaining), GCM (Galois/Counter Mode), SIV (Synthetic Initialization Vector).	Son más simples de implementar, ya que de acuerdo con el modo de operación por el que se haya optado ya se contará con autenticidad de los datos y con un nivel de seguridad adecuado.
Stream Cipher	Confusión.	Utiliza una llave distinta para cifrar cada byte.	Es más rápido ya que realiza sus operaciones byte a byte.	Solo requiere memoria para el byte que se está encriptando en ese instante.	CFB (Cipher Feedback), OFB (Output Feedback).	Son más complicados de implementar, ya que se debe tener en cuenta otros factores como la sincronización o la autenticidad de los datos, entre otras.

Tabla 2: Diferencias entre Block y Stream cipher

4 Conclusión

5 Referencias

Biryukov, A., & Perrin, L. (2017). State of the art in lightweight symmetric cryptography. *Cryptology ePrint Archive*.

Cipriano, M. J. (2021). *Criptografía liviana e Internet de las Cosas: Confidencialidad de la informaci'on mediante Stream Ciphers estandarizados en las normas ISO/IEC 18033 y 29192* [Tesis doctoral, Universidad Nacional de La Plata].

Eterovic, J., Cipriano, M., García, E., & Torres, L. M. (2019). Criptografía Ligera en Internet de las Cosas para la Industria. *XXV Congreso Argentino de Ciencias de la Computaci'on (CACIC)(Universidad Nacional de Río Cuarto, Córdoba, 14 al 18 de octubre de 2019)*.

Eterovic, J. E., & Cipriano, M. J. (2018). Stream Ciphers livianos estandarizados mediante normas internacionales para ser usados en Internet de las Cosas. *Sistemas, Cibern'etica e Inform'atica*, *15*(2-2018).

[8]Para su mayor conocimiento, en el mismo artículo se enumera una serie de algoritmos ligeros junto con los tipos de ataques a los que han sido sometido cada algoritmo. Los tipos de ataques incluyen ataques de criptoanálisis, ataques de canales laterales, ataques de fuerza bruta, entre otros. Esta información resulta útil para comprender la seguridad de los algoritmos y evaluar su resistencia a diversos tipos de amenazas.

Gunathilake, N. A., Buchanan, W. J., & Asif, R. (2019). Next generation lightweight cryptography for smart IoT devices:: implementation, challenges and applications. *2019 IEEE 5th World Forum on Internet of Things (WF-IoT)*, 707-710.

Harris, S. (2008). Exploring Cipherspace: Combining stream ciphers and block ciphers. *Cryptology ePrint Archive*.

Ismoyo, D. D., & Wardhani, R. W. (2016). Block cipher and stream cipher algorithm performance comparison in a personal VPN gateway. *2016 International Seminar on Application for Technology of Information and Communication (ISemantic)*, 207-210.

Thakor, V. A., Razzaque, M. A., & Khandaker, M. R. (2020). Lightweight cryptography for IoT: A state-of-the-art. *arXiv preprint arXiv:2006.13813*.

Tilborg, H. C. v., & Jajodia, S. (s.f.). Encyclopedia of Cryptography and Security, 2011.

Valea, E., Da Silva, M., Flottes, M.-L., Di Natale, G., & Rouzeyre, B. (2019). Stream vs block ciphers for scan encryption. *Microelectronics Journal*, *86*, 65-76.

Wehbe, R., Armas, A. d., & Barrera, E. (2022). Criptografía liviana para objetos conectados. *XXIV Workshop de Investigadores en Ciencias de la Computaci'on (WICC 2022, Mendoza)*.