Centro Universitario de la Defensa
en la Escuela Naval Militar

TRABAJO FIN DE MÁSTER

Técnicas criptográficas ligeras para dispositivos IoT

Máster Universitario en Dirección TIC para la Defensa

ALUMNO: Emilio José Gordillo Vega

DIRECTORES: Javier Vales Alonso

Milagros Fernández Gavilanes

CURSO ACADÉMICO: 2021-2022

Universidade Vigo

Centro Universitario de la Defensa
en la Escuela Naval Militar

TRABAJO FIN DE MÁSTER

Criptografía ligera para dispositivos IoT

Máster Universitario en Dirección TIC para la Defensa
Especialidad de Sistemas y Tecnologías de la Información

Universidade Vigo

RESUMEN

El IoT (Internet Of Things) abre un nuevo paradigma en la sociedad del futuro en el que todo tenderá a estar interconectado, de hecho se estima que se superen los 50.000 millones de elementos conectados a final de esta década.

Con el avance técnico, la interacción de las personas y los sistemas de información, las redes, sensores, sistemas de comunicación y computadoras han disminuido de tamaño, aumentado su capacidad de cálculo y se han abaratado, con lo que se ha pasado de encontrarse sobre una mesa en forma de PC o de teléfono móvil a poder encontrarse en prácticamente cualquier dispositivo, electrodoméstico o, en general, "cosa", ya que es posible encontrarlo en la ropa o, incluso, en el propio cuerpo humano.

No obstante, la información que contienen estos dispositivos debe protegerse, y una forma de hacerlo es con algoritmos criptográficos.

Los algoritmos criptográficos se utilizan diariamente para proteger transacciones de Internet, como los pagos en línea, y ofrecen una manera de verificar el origen de los datos y de evitar que se intercepte la información que contienen, con un alto nivel de confianza.

Sin embargo, el uso de técnicas criptográficas con dispositivos IoT, al tener una baja potencia de cómputo, su cifrado no puede llevarse a cabo con métodos complejos, y se deben utilizar algoritmos de criptografía ligera que permitan un nivel de seguridad aceptable para este tipo de dispositivos.

Este trabajo expondrá algoritmos ligeros de cifrado para dispositivos IoT, probando que son seguros y que deben ser introducidos en estos dispositivos, para que la información que contienen no esté expuesta frente a un ataque informático. Estas técnicas dependerán del software embebido en cada dispositivo, y se deberá aplicar un tipo de criptografía adaptada a cada clase.

PALABRAS CLAVE

Criptografía, IoT, seguridad, ataque criptográfico, criptografía ligera

AGRADECIMIENTOS

Palabras de agradecimiento.

- Cuando llegó la hora de elegir el TFM, pensé en solicitar algún trabajo al profesor de una las asignaturas que más me ha gustado del Máster DIRETIC, se trataba de “Seguridad en Sistemas de Información”. Le pedí al profesor Javier Vales si tenía algo para preparar un TFM de Criptografía y me comentó que sería interesante realizar un trabajo sobre criptografía ligera para dispositivos IoT. En un principio no sabía ni lo que era la “lightweight cryptography” pero tras varios meses de estudio, la verdad es que es una línea de investigación muy novedosa e interesante.

- A mi familia, sobre todo a mi mujer e hijo que han tenido que aguantar el tiempo que me ha llevado realizar este Máster y los demás estudios realizados, y a mis padres que consiguieron con su esfuerzo que terminara mi carrera y pudiera encontrar trabajo.

- Agradecer a mi tutor Javier, que haya estoy pendiente de mi trabajo ayudándome en todo lo que le solicitado.

Contenido

Contenido 5
Índice de Figuras 8
Índice de Tablas 9
Listado de acrónimos 10
1 Introducción y objetivos 12
1.1 Introducción 12
1.1.1 El Internet de las cosas (IoT) 12
1.1.2 Seguridad de la información 12
1.1.3 Criptografía 14
1.1.3.1 Criptografía convencional 14
1.1.3.2 Criptografía ligera 15
1.2 Objetivos 17
2 Estado del arte 18
2.1 El Internet del las cosas (IoT) 18
2.1.1 Definición IoT 18
2.1.2 Campos de aplicación de las redes IoT 19
2.1.3 Seguridad de los dispositivos IoT 21
2.1.4 Protección de los dispositivos IoT 22
2.1.5 El IoT y la Criptografía 23
2.2 Criptografía. 24
2.2.1 Introducción a la criptografía 24
2.2.2 Criptografía clásica 25
2.2.3 Criptografía moderna 25
2.2.3.1 Criptografía simétrica 26
2.2.3.2 Criptografía asimétrica (o de clave pública) 32
2.2.3.3 Funciones Hash 34
2.2.4 Criptografía ligera 36
2.2.4.1 Ejemplos de cifrados de bloques ligeros 37
2.2.4.2 Ejemplos de cifrados ligeros de flujo 43
2.2.4.3 Criptográfía ligera asimétrica (o de clave pública) 47
2.2.4.4. Funciones Hash Ligeras 47
2.3 Ataques Criptográficos 50
2.3.1. Ataque de búsqueda exhaustiva (Exhaustive key search) 50
2.3.2. Criptoanálisis diferencial/lineal (Differential-linear cryptanalysis) 50

2.3.3. Criptoanálisis integral/square/saturation 50
2.3.4. Ataque algebraico (Algebric attack)/Ataque cubo (Cube attack) 51
2.3.5. Ataque Meet-in-the-middle (MITM)/Biclique 51
2.3.6. Ataque de clave relacionada (Related key attack) 51
2.3.7. Ataque de canal lateral (Side Channel)/Ataque de fallos diferenciales (Differential fault attack) 52
2.3.8. Ataque de correlación (Correlation Attack) 52
2.3.9. Ataque distintivo (Distinguishing Attack) 52
2.3.10. Ataque Chosen-IV 52
2.3.11. Ataque de deslizamiento (Slide Attack) 53
2.3.12. Ataque de compensación de tiempo y memoria (Time-Memory Trade-off Attack) 53
2.3.13. Ataque de suposición (Guess and Determine Attack) 53
2.4 Criptografía ligera vs Ultraligera 54
3 Desarrollo del TFM 55
3.1 RFID 55
3.1.1 Objetivos de los ataques en dispositivos RFID 58
3.1.2 Seguridad en los dispositivos RFID 58
3.1.3 Amenazas de seguridad 60
3.1.4 Medidas de seguridad frente a la amenazas de seguridad 62
3.1.5 Clasificación de los protocolos RFID 64
3.1.6 Protocolo de Autenticación SASI 65
3.2 Redes WSN 69
3.2.1 Objetivos de los ataques en WSN 70
3.2.2 Seguridad en las redes WSN 71
3.2.3 Amenazas de seguridad en WSN 71
3.2.4 Medidas de seguridad en WSN 72
3.2.5 Protocolos de seguridad en redes WSN 73
3.2.6 Algoritmo ECDSA para WSN 74
3.3 Smart cards 74
3.3.1 Ejemplos de Smart cards 76
3.3.2 Seguridad en las Smart cards 76
3.3.3 Ataques a las Smart cards 76
3.3.4 Protocolos de seguridad en Smart cards 77
4 Resultados / Validación / Prueba 78
4.1 Ejemplo práctico 1. Sonoboya 78
4.2 Ejemplo práctico 2. Protocolo Ladon 79

4.3 Ejemplo práctico 3. Hidrófono 81
5 Conclusiones y líneas futuras 82
5.1 Conclusiones 82
5.2 Líneas futuras 83
6 Bibliografía 84
Anexo I: CLASE DE ETIQUETAS RFID 89
Anexo II: Clasificacion/taxonomia de los ciberdelincuentes 90
Anexo III: Análisis del rendimiento de los algoritmos de criptografía ligera 92
Anexo IV: Análisis de la seguridad de los algoritmos de criptografía ligera 93
Anexo V: Top 10 algoritmos de criptografía ligera 94

Índice de Figuras

Figura 1-1 Pilares de la Seguridad de la Información (Fuente: Elaboración propia) 13
Figura 1-2 Categorías principales de dispositivos IoT (Tomada de [1]) 15
Figura 2-1 Previsión de los dispositivos conectados a Internet a nivel mundial en 2018, 2025 y 2030 (en miles de millones de unidades) Fuente: Statista 2021 18
Figura 2-2 Descripción y ejemplos de los dominios de aplicación en IoT (tomada de internet) 21
Figura 2-3 Desafíos de la Criptografía en dispositivos IoT (tomada de [1]) 22
Figura 2-3 Proceso de cifra de un mensaje (Fuente: Elaboración Propia basada en [8]) 26
Figura 2-4 Gráfico de Criptografía Simétrica (tomada Libro de Mcgraw Hill) 27
Figura 2-5 AES (tomada de internet) 31
Figura 2-6 Gráfico de Criptografía Asimétrica (tomada Libro de Mcgraw Hill) 33
Figura 2-7 Algoritmos criptográficos ligeros (Fuente: Elaboración propia basada en [2]) 37
Figura 3-1 Partes de un sistema RFID (Fuente: Elaboración propia) 56
Figura 3-2 Protocolo SASI (fabricación propia tomada de [20]) 66
Figura 3-3 Partes de un nodo o sensor inalámbrico (tomada de [13]) 70
Figura 3-4 Smart card (tomada de https://empresayeconomia.republica.com) 75
Figura 4-1 Sonoboya AN/SS0-47 (tomada de internet) 79
Figura 4-2 Marcapasos `St Jude Medical` (tomada de internet) 80
Figura 4-3 Miniature Hydrophone type 8103 (tomada de BRÜEL & KJÆR) 81

Índice de Tablas

Tabla 1-1 Algoritmos de Criptografía ligera (Elaboración propia basada en [2]) 16

Tabla 2-1 Algoritmos criptográficos más populares (tomada de [4]) 24

Tabla 2-2 Comparación entre algoritmos ligeros de bloques (Fuente: Elaboracion Propia basada en [1, 3, 14, 16 y 22]).....42

Tabla 2-3 Comparación entre algoritmos ligeros de flujo (Fuente: Elaboración propia basada en [3]).....46

Tabla 2-4 Algoritmos contenidos en las normas ISO/IEC (tomada de [13]) 46

Tabla 3-1 Aplicaciones IOT con RFID (tomada de [15]) 57

Tabla 3-2 Diferencias entre la criptografía ligera y ultraligera (Fabricación propia basada en de [27]).....54

Tabla 3-3 Comparación entre los Protocolos de Autenticación Ultraligeros (Fabricación propia tomada de [20]) 68

Listado de acrónimos

AES: Advanced Encryption Standard

BAN: Body Area Network

CPU: Central Processing Unit

CRC: Código de Redundancia Cíclica

DES: Data Encryption Standard

DESX: Data Encryption Standard Xor

DESXL: Data Encryption Standard Xor Light

DOS: Denial of Service

DPA: Differential Power Analysis

DSA: Digital Signature Algorithm

DSM: Diffusion Conmutation Mechanism

ECC: Elliptic Curve Cryptograph

ECG: Electrocardiograma

EPC: Electronic Product Code

ET. AL.: Et Alii (y otros)

FSR: Feedback Shift Registers

GB/S: Gigabit por Segundo

GE: Gate Equivalent

GPRS: General Packet Radio Service

GPS: Global Positioning System

GSM: Global System for Mobile communications

HECC: Hyper Elliptic Curve Cryptograph

IBM: International Business Machines

ID: Identification

IDEA: International Data Encryption Algorithm

IDS: IDentification Seudonym

IEC: International Electrotechnical Commission

I2T: Investigación e Ingeniería Telemática

ITS: Intelligent Transportation Systems

KHZ: Kilohercio

KB/S: Kilobyte por segundo

LFSR: Linear Feedback Shift Register

LOPD: Ley Orgánica de Protección de Datos

NLFSR: Non Linear Feedback Shift Register

MAC: Message Authentication Code

MD5: Message Digest Algorithm 5

MHZ: Megahercio

MITM: Man In The Middle

NIST: National Institute of Standards and Technology

NSA: National Security Agency

PIN: Personal Identification Number

PKI: Public Key Infrastructure

RAE: Real Academia Española

RAM: Random Access Memory,

RC4: Rivest Cipher 4

RFID: Radio Frecuency Identification)

ROM: Read-Only Memory

RSA: Rivest, Shamir, Adleman

SASI: Semi-supervised Algorithm Sarcasm Identification

SBOX: Substitution Box

SEAL: Software-Optimized Encryption Algorithm

SHA: Secure Hash Algorithm

SJM: St Jude Medical

SPN: Substitution Permutation Network

SSL: Secure Sockets Layer

TI: Tecnología de la Información

TLS: Transport Layer Security

TPV: Terminal Punto de Venta

UCI: Unidad de Cuidados Intensivos

UMAP: Uniform Manifold Approximation and Projection

WIFI: WIreless FIdelity

WSN: Wireless Sensor Networks

XOR: Exclusive Or

1 INTRODUCCIÓN Y OBJETIVOS

1.1 Introducción

1.1.1 El Internet de las cosas (IoT)

El Internet de las Cosas o IoT (proviene de Internet of the Things) consiste en la interconexión de objetos de uso cotidiano con otros de su alrededor, es decir, se trata de convertir objetos cotidianos en fuentes de información que están conectadas a la red.

Así, el Internet de las cosas es un concepto intangible, es la conexión por ejemplo entre nuestros smartphones y los dispositivos smart que tenemos en casa para controlar la iluminación o el aire acondicionado, es una Raspberry Pi que controla la programación de tu televisor o del lavavajillas, o es un robot aspiradora que configuramos desde el teléfono móvil para que limpie la casa automáticamente cuando no estamos.

Otro dispositivos IoT son los altavoces inteligente tipo Alexa, Siri, Google assistant, etc. que se encuentran conectados a Internet para obtener información e interactúan con otros dispositivos para darnos la capacidad de controlarlos a través de simples comandos de voz. Poco a poco los diferentes dispositivos que se utilizan en las casas, van siendo más y más dispositivos IoT y se interconectan entre sí con la idea de hacernos la vida más fácil.

El IoT tiene enormes ventajas, pero es esencial proteger esos dispositivos de los ataques de aquellas personas que quieren acceder a ellos, y ser capaces de detectar y aislar aquellos que hayan quedado comprometidos. El uso de la criptografía ayuda a proteger estos dispositivos y su autenticación, y proporciona una seguridad sencilla y sólida con la que podemos mantener la integridad de los datos que contienen en su interior.

No obstante, el uso de técnicas criptográficas plantea diversos retos. Uno de ellos es que los algoritmos criptográficos que se emplean para cifrar o descifrar las transmisiones requieren muchos recursos informáticos, lo que supone una carga adicional para este tipo de sistemas, que son, por la propia naturaleza del IoT, extremadamente limitados.

1.1.2 Seguridad de la información

Desde que apareció Internet, hemos escuchado y algunos sentido, como se producen vulneraciones de seguridad informática. La información es a menudo el blanco principal de atacantes, quienes a menudo buscan robar información de las tarjetas de crédito, cuentas de banco, nombres de usuario,

contraseñas e información corporativa. Otros ejemplos son simplemente la pérdida de información, y que ésta no esté disponible en el momento en que se necesita.

Es indispensable conocer los pilares que soportan la seguridad de la información para conocer cómo se pueden producir estos ataques informáticos:

Integridad: El diccionario de la RAE (Real Academia Española) define el término como "Cualidad de íntegro", e integro como "que no carece de ninguna de sus partes". La integridad hace referencia a la cualidad de la información que indica que no ha sido modificada, manteniendo sus datos exactamente igual que cuando fueron generados, sin manipular ni alterar. La integridad de la información se pierde cuando la información se modifica o se elimina en su totalidad o en parte.

Un aspecto relacionado con la integridad es la autentificación, cualidad que permite identificar al que genera la información y que se logra con un acceso correcto del usuario, verificando que es quien dice ser. Para algunos, la autentificación sería el "cuarto pilar" de la Seguridad de la Información.

Confidencialidad: Por confidencialidad entendemos la cualidad de la información para no ser divulgada a personas o sistemas no autorizados, para ello es imprescindible que sea accesible sólo para aquellos que estén acreditados.

¿Cómo se pierde esa confidencialidad? Generalmente, haciendo caso omiso a las recomendaciones de seguridad o no implantando un sistema adecuado; así, cuando compartimos equipos sin eliminar las contraseñas, olvidamos cerrar nuestro usuario, navegamos por paginas no seguras, tiramos un disco duro sin borrar antes sus datos o no ciframos los datos de manera adecuada, la información deja de ser confidencial y dejamos la puerta abierta para que esta información pueda ser divulgada.

Disponibilidad: El tercer y último pilar de la Seguridad de la Información es la disponibilidad, y es posiblemente el término que menos apreciaciones requiere. Por disponible entendemos aquella información a la que podemos acceder cuando la necesitamos a través de los canales adecuados y siguiendo los procesos correctos.

Esta característica, puede en ocasiones chocar frontalmente con la confidencialidad, ya que un cifrado complejo o un sistema de archivado más estricto puede convertir la información en algo poco accesible, esto dependerá del responsable de la seguridad de la información de la empresa u organización.

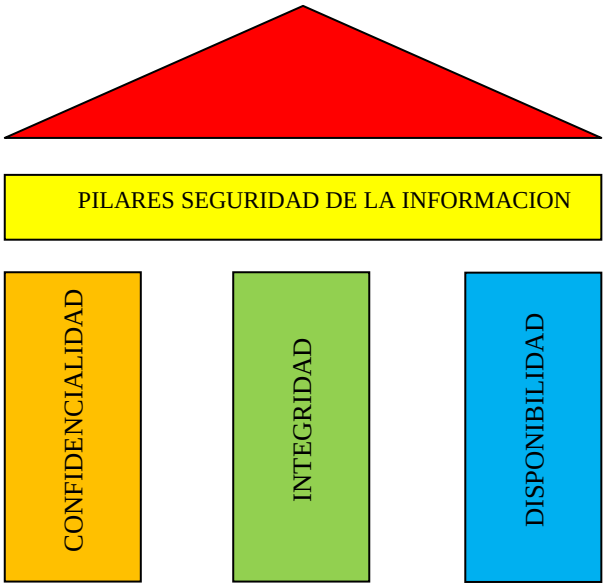


Figura 1-1 Pilares de la Seguridad de la Información (Fuente: Elaboración propia)

1.1.3 Criptografía

La palabra Criptografía proviene del griego "kryptos" que significa oculto, y "graphia", que significa escritura, y su definición según el diccionario es "Arte de escribir con clave secreta o de un modo enigmático". La Criptografía es conjunto de técnicas, que originalmente tratan sobre la protección o el ocultamiento de la información frente a observadores no autorizados. Entre las disciplinas que engloba cabe destacar la Teoría de la Información, la Complejidad Algorítmica y la Teoría de números o Matemática Discreta.

A través de la criptografía, la información puede ser protegida contra el acceso no autorizado, su interceptación, su modificación y la inserción de información extra. También puede ser usada para prevenir el acceso y uso no autorizado de los recursos de una red o sistema informático, y para prevenir a los usuarios de la denegación de los servicios a los que sí están permitidos. En la actualidad, la criptografía se usa para proveer de seguridad de las redes telemáticas, incluyendo la identificación de entidades y su autenticación, el control de acceso a los recursos disponibles en el sistema, la confidencialidad de los mensajes transmitidos, la integridad de estos mensajes y su no repudio[1].

Para este trabajo dividiremos la criptografía en convencional y ligera ya que esta división es pertinente para dispositivos IoT.

1.1.3.1 Criptografía convencional

Los métodos convencionales de criptografía funcionan bien en sistemas con suficiente capacidad de procesamiento, energía, espacio físico y memoria, pero no se adaptan bien a un mundo como el de la IoT donde existen limitaciones energéticas, requisitos de procesamiento o de memoria.

Un esquema de cifrado convencional tiene 5 partes:

- Mensaje a transmitir o texto en “claro”
- Algoritmo de cifrado
- Clave secreta
- Texto cifrado
- Algoritmo de descifrado

La seguridad del cifrado se consigue mediante años de escrutinio y análisis público, y la solución está en mantener segura la clave secreta, **no** el algoritmo. Los algoritmos se clasifican dependiendo de:

- Tipo de operaciones para pasar del texto en claro al texto cifrado
- Numero de claves usadas (Simétrico o Asimétrico)
- La forma en la que se opera con el texto en claro (Flujo o bloques)

Un sistema de cifrado es un par de algoritmos “eficientes” (E, D), tales que:

- Para todo m, k es **D(k, E(k, m)) = m.**
  Con $k \in K$, $m \in M$, $c \in C$ $M = \{0, 1\}^n$: mensajes de longitud n
  $K = \{0, 1\}^s$: claves de longitud s |K|: número de posibles claves

[1] **No repudio en origen:** El emisor no puede negar que envió el mensaje porque el destinatario tiene pruebas del envío.
No repudio en destino: El receptor no puede negar que recibió el mensaje porque el emisor tiene pruebas de la recepción.

Para que un sistema de cifrado tenga el denominado “secreto perfecto” (condición de Shannon), el texto cifrado no puede revelar información sobre el texto plano, siendo imposible que se descifre, pero esto computacionalmente hablando no es factible, porque requeriría que la clave sea tan larga como el mensaje.

Tenemos que considerar que un cifrado pueda romperse[2], pero se supone seguro si el coste de romperlo es mayor que el valor de la información cifrada o el tiempo que se tarda en romperlo es mayor que la vida útil de información cifrada.

Los estándares y protocolos criptográficos convencionales fueron diseñados para servidores y equipos de sobremesa, incluso para tablets y smartphones, pero no aplican a sistemas caracterizados por la escasez de energía, de CPU y de memoria como es el de los dispositivos IoT.

Para superar estos problemas, se proponen métodos de criptografía ligera que ofrecen un compromiso razonable entre seguridad, rendimiento y coste con respecto a la criptografía convencional.

En la última década, se han propuesto y utilizado un gran número de primitivas criptográficas ligeras en dispositivos con recursos limitados. Las organizaciones nacionales como el Instituto Nacional de Estándares y Tecnología (NIST), como las internacionales, como la Comisión Electrotécnica Internacional (ISO/IEC), utilizan una serie de métodos basados en criptografía ligera que se adaptan bien en dispositivos IoT y RFID (Radio Frecuency Identification).

En la figura 1-2 se puede apreciar los dispositivos para cada una de las criptografías:

- Criptografía convencional: Servidores y ordenadores personales, tabletas y teléfonos inteligentes, etc.
- Criptografía ligera: Sistemas embebidos, etiquetas RFID, sensores industriales o nodos de sensores, actuadores, etc.

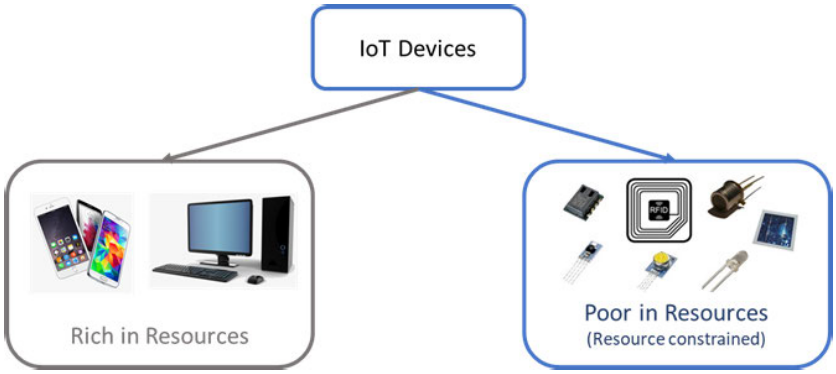


Figura 1-2 Categorías principales de dispositivos IoT (Tomada de [1])

1.1.3.2 Criptografía ligera

La criptografía ligera se define generalmente como la criptografía para dispositivos con recursos limitados, de la que se suelen mencionar como ejemplos, las etiquetas RFID o identificación por radiofrecuencia y las WSN (Wireless Sensor Networks) o redes de sensores inalámbricas.

La criptografía ligera es la unión de dos términos "Light” (ligero) y “weight” (peso), con un algoritmo criptográfico clásico inicial y donde se intenta que funcione bien en dispositivos limitados.

[2] El criptoanálisis es la rama de criptografía que estudia cómo romper códigos y criptosistemas.

Estas limitaciones suelen estar relacionadas con los requisitos de potencia, los equivalentes de puerta (GEs - Gate Equivalents) es decir la complejidad del dispositivo y el tiempo de procesamiento del propio algoritmo en el dispositivo.

El cifrado y descifrado ligero se implementa en plataformas tanto de hardware como de software.

Por ejemplo en el caso de los sistemas embebidos, es habitual ver microcontroladores de 8, 16 y 32 bits, que tendrían dificultades para hacer frente a las demandas en tiempo real de los métodos criptográficos convencionales.

Podemos clasificar la **criptografía simétrica ligera** en cifrados de bloque, de flujo y funciones Hash:

- Cifrados de bloque (Apdo. 2.2.3.1.2): son algoritmos utilizados para cifrar los datos en un intento de reemplazar al algoritmo de cifrado AES con otros algoritmos con menores limitaciones, aunque obteniendo niveles de seguridad comparables. Existen muchos algoritmos que trabajan con tamaños de bloque más reducidos, por ejemplo, 64 bits en lugar de 128; claves más pequeñas, por ejemplo, 80 bits en lugar de 128 o 256; menor número de rondas con un diseño más sencillo; programación de claves más simple e implementaciones mínimas, que no soportan modos de operación complejos. Destacan los algoritmos PRESENT, CLEFIA y SIMON.
- Cifrados de flujo (Apdo. 2.2.3.1.1): se cuentan entre las primitivas más prometedoras para entornos limitados, ya que desde siempre los cifrados en flujo se han orientado a la eficiencia. Estos algoritmos presentan elevados niveles de seguridad con alto rendimiento y bajo coste. Existen numerosas propuestas, entre las que destacan GRAIN, TRIVIUM Y MICKEY [53].
- Funciones hash (Apdo. 2.2.3.3): son algoritmos utilizados para crear resúmenes criptográficamente seguros de los datos. Se han propuesto numerosas funciones ligeras de hash, como PHOTON y SPONGENT, caracterizadas por tamaños más reducidos tanto de salida como de estado interno. Eso sí, con el riesgo de sufrir colisiones y tamaños de mensaje también menores, por ejemplo, de hasta 256 bits.

Los algoritmos de **criptografía ligera asimétrica** que veremos serán BLUEJAY y ECC LIGERA (Criptografía de Curva Elíptica Ligera).

CRIPTOGRAFÍA LIGERA									
Simétrica							**Asimétrica**		
Flujo	Bloque		Función Hash						
GRAIM	AES	CLEFIA	PHOTON	DM-Present	ARMADILLO		BLUEJAY	ECC LIGERA	
TRIVIUM	PRESENT	SIMON	SPONGENT						
MICKEY			LESAMANTA-LW						

Tabla 1-1 Algoritmos de Criptografía ligera (Elaboración propia basada en [2])

1.2 Objetivos

El objetivo que se quiere conseguir con este trabajo es presentar algunos algoritmos de criptografía ligera para IoT y comprobar que su funcionamiento garantiza la seguridad y la protección de la privacidad de los dispositivos IoT, confirmando que la aplicación de algoritmos convencionales suponen un problema cuando se integran en dispositivos con recursos limitados, ya que son incapaces de funcionar en ellos.

Lo anteriormente expuesto, da lugar al planteamiento de unos objetivos más concretos que se perseguirán en este documento:

1) Realizar una enumeración de las limitaciones de recursos de los dispositivos IoT y presentar una serie de campos tecnológicos concretos como ejemplos de esta limitación de recursos.

2) Explicar posibles técnicas criptográficas ligeras que utilicen menos recursos a nivel energético, temporal, de comunicaciones, etc.

3) Explicar cómo se llevan las técnicas de criptografía ligera a los sistemas analizados en el primer objetivo.

4) Presentar una serie de guías que permitan conocer cuando tiene sentido utilizar las técnicas de criptografía ligera o convencional. En caso de utilizar criptografía ligera se presentarán los posibles algoritmos comentando ventajas e inconvenientes.

5) Analizar casos reales en el que tengamos que elegir qué tipo de algoritmo se debe utilizar para garantizar la seguridad de los datos transmitidos por el dispositivo

Metodología:

Lo primero que se plantea en este trabajo (apartado 1) es una introducción a tres conceptos principales, el Internet de las Cosas, la Seguridad de la información y la Criptografía que dividimos en convencional y ligera.

En el apartado 2 se presentan con más detalle los tres conceptos introducidos para exponer algunos de los muchos campos de aplicación de los dispositivos IoT para que principalmente el lector compruebe la importancia de estos mecanismos y algunas generalidades sobre su seguridad, como se pueden proteger y qué relación existe entre estos dispositivos y la Criptografía.

En el apartado 2.2 se hace una presentación extensa sobre los tipos de Criptografía dividiéndola en tres tipos, clásica, moderna y ligera y se relacionan los algoritmos más importantes de cada tipo. También se exponen los ataques criptográficos más significativos.

En el apartado 3 se presentan tres casos en los que se demuestra que la criptografía ligera es útil para garantizar la información transmitida por los dispositivos IoT. Se trata de etiquetas RFID, WSN y Smart Cards.

En el apartado 4 se exponen tres casos prácticos como ejemplos de dispositivos IoT que necesitan proteger sus comunicaciones de posibles atacantes, estos son una sonoboya militar, un marcapasos y un hidrófono.

Por último se exponen las conclusiones de este trabajo, unas posibles línea futuras y la bibliografía utilizada.

2 ESTADO DEL ARTE

2.1 El Internet del las cosas (IoT)

2.1.1 Definición IoT

La definición de la RAE es "Interconexión digital de personas, animales y cosas (electrodomésticos, coches, etc.) con Internet". Un dispositivo IoT puede ser una cosa como una tostadora conectada a Internet (se considera como el primer dispositivo IoT y ya en 1990 se podía controlar su encendido, apagado y tiempo de "tostado"), una persona con un implante cardíaco que transmite estos datos para que sean monitorizados por personal médico, un animal con un chip para poder conocer su ubicación, un automóvil con sensores que alertar al conductor si se aproxima demasiado a otro vehículo o cualquier otra situación natural o artificial que pueda transferir datos a través de una red.

Cada vez más se utilizan dispositivos con conexión a Internet, ya que es una información de interés para la industria (sirve para ofrecer mejores servicios a los clientes o mejorar la toma de decisiones) y para los propios usuarios ya que ofrecen servicios que facilitan su vida.

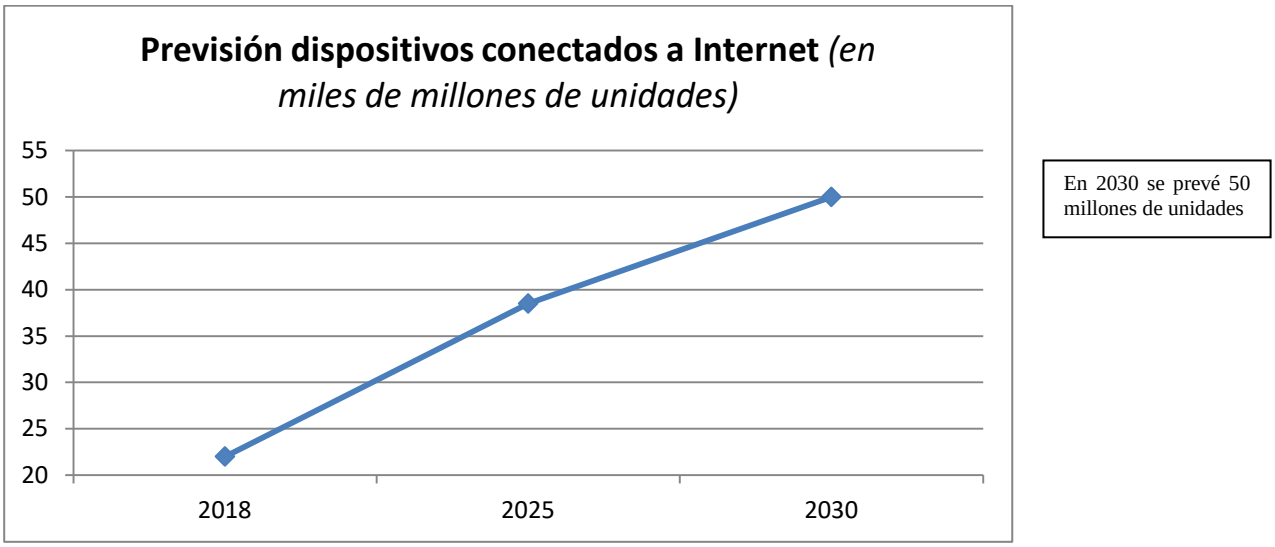


Figura 2-1 Previsión de los dispositivos conectados a Internet a nivel mundial en 2018, 2025 y 2030 (en miles de millones de unidades) Fuente: Statista 2021 [58]

2.1.2 Campos de aplicación de las redes IoT

Las aplicaciones previstas para IoT abarcan una amplia gama de campos que incluyen domótica, cuidado de la salud, vigilancia, transporte, entornos inteligentes y muchos más. Veamos algunos ejemplos en las áreas principales de aplicación:

1. Aeroespacio y aviación.

Utilizando tecnología de seguimiento de equipaje RFID, se consigue evitar la pérdida de maletas, también podemos utilizar el IoT mediante sensores que reconozcan y resuelvan un problema mecánico, incluso antes de que suceda.

2. Automoción.

Por ahora, el IoT abarca tan sólo sistemas de navegación GPS y otras funciones básicas, pero se están integrando nuevas funcionalidades como seguridad, alerta de accidente y otras similares que llevaran al coche autónomo, es decir sin conductor.

3. Edificios inteligentes.

Por ejemplo, una aplicación de edificio inteligente en el móvil puede dirigirnos a estacionamientos no ocupados.

4. Tecnología médica, Cuidado de la salud.

Se puede conectar IoT con un dispositivo de ECG (un dispositivo que registra el estado del corazón) para recopilar información de un paciente y compartirla con otros centros de atención médica.

5. Vida independiente.

Todo tipo de electrodomésticos forman parte ya de nuestra vida diaria, lavadoras que permiten el encendió/apagado a distancia o que cuando acaban, hacen que las luces parpadean, placas de inducción con recetarios y que permiten manejar la campana extractora, robot aspirador que admite instrucciones por voz, encender y apagar las luces, cambiar su color o regular la intensidad lumínica también por la voz, cubos de basura , que permiten saber cuándo la bolsa está llena, sellarla y reemplazarla automáticamente y un largo etc.

6. Farmacéutico.

Muchos medicamentos deben estar mantenidos bajo ciertas condiciones especiales. Los sensores IoT pueden ayudar enormemente a monitorear en tiempo real y controlar las condiciones bajo las cuales se mantienen los productos farmacéuticos desde su manufactura hasta su comercialización.

7. Retail, Logística, Supply Chain Management

Datalong 16 [12] ha diseñado un pequeño dispositivo que viaja con la mercancía y que permite obtener valores, en tiempo real, sobre la ubicación del envío y la temperatura a la que se encuentra.

Las estanterías inteligentes, aparte de controlar los “huecos” disponibles, avisan de si el peso y las dimensiones del artículo colocado siguen las especificaciones, y también alertan de si existe riesgo de caída del bulto o de accidente del operario.

8. Fabricación, Gestión del ciclo de vida del producto.

Al aplicar las aplicaciones IoT a los sectores logísticos, es posible hacer un seguimiento del inventario, las ubicaciones y otros factores de monitoreo. Desde el lado del consumidor, IoT ofrece muchos beneficios a los como la transparencia, eficiencia, sistemas de fidelización, etc.

9. Compras.

Se puede controlar fácilmente el comportamiento de los clientes, sus necesidades, hábitos de compra, preferencias, etc. De este modo, pueden mantener una mejor experiencia de usuario para aumentar las ventas.

10. Petróleo y gas.

Estas empresas han instalado sensores dentro y fuera de las tuberías para detectar principalmente posibles roturas de tuberías e interrupciones en el proceso. También es importante predecir cuándo las máquinas necesitan mantenimiento.

11. Gestión del medio ambiente.

Por ejemplo para recuperar especies al borde de la extinción rastreando y monitoreando a los animales mediante el uso de collares para el geomapping de su ubicación y de sus hábitos, junto con drones conectados, para rastrearlos.

Programas que permiten a los usuarios encender o apagar las luces y controlar la temperatura y la actividad de la caldera cuando se está fuera de casa, para ahorrar en eficiencia mediambiental.

12. Transporte de personas y bienes.

Por ejemplo el autobús eléctrico de Manchester, que consume un 80% menos de energía y su mantenimiento se controla a través de Internet [55].

13. Trazabilidad de los alimentos.

La trazabilidad de un producto desde su producción hasta la cesta de la compra es clave en la seguridad alimentaria global. Las etiquetas RFID pegadas a los palets permiten la localización de los productos alimenticios en cada etapa de su viaje por la cadena de suministro, desde que sale de los campos de cultivo hasta los lugares de distribución, pasando por las zonas de tratamiento y finalmente las tiendas al por mayor y grandes supermercados.

14. Agricultura y Cría.

Los agricultores colocan sensores en los campos para medir el nivel de humedad y transmiten los datos a un sistema central. Por lo tanto, es posible saber cuándo las plantaciones alcanzan su nivel óptimo de humedad.

El tractor inteligente es un tractor automatizado, es decir, no necesita conductor. En su software existe un mapa con los itinerarios y tareas a realizar. Dispone de GPS, sensores y radares, que le permite circular entre los cultivos.

Un sensor similar al utilizado en la agricultura se puede implantar en los animales para vigilarlos. Ofrecen datos de su ubicación, el estado nutricional y la capacidad reproductiva.

15. Medios, entretenimiento y venta de tickets,

La smartTV es el ejemplo por excelencia, conectada a Internet proporciona servicios de televisión.

16. Seguros.

Por ejemplo en el caso del seguro del automóvil, se utilizarían los sensores para conocer las distancias que recorren, la velocidad media a la que conduce, si tiene algún accidente, etc. Esta información es muy útil para la compañía porque podría seleccionar a los mejores conductores y obtener mayor margen comercial y también no ofrecer sus servicios a los malos conductores.

17. Reciclaje.

Por ejemplo se ha presentado un “contenedor de residuos inteligente” que es capaz de identificar y clasificar los residuos en hasta cuatro categorías: vidrio, papel, plástico y metal basado en IoT.

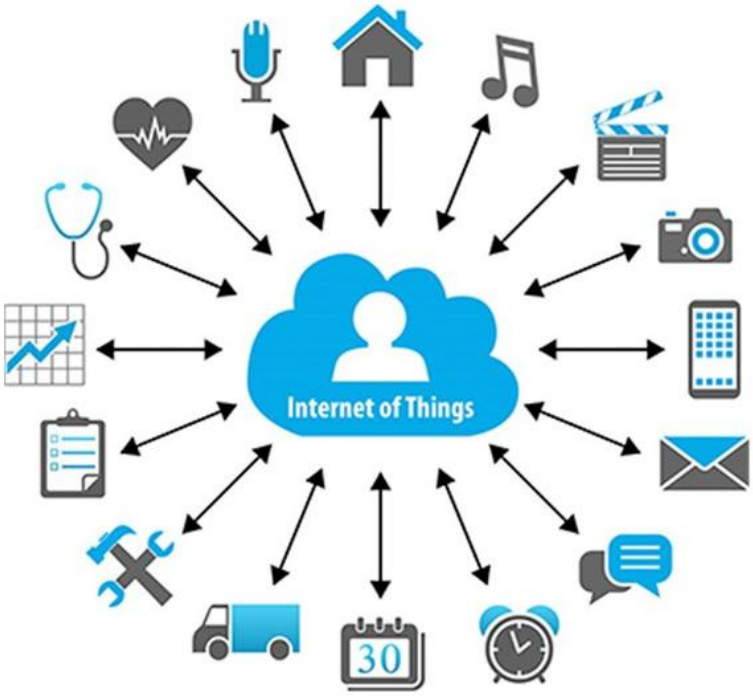


Figura 2-2 Descripción y ejemplos de los dominios de aplicación en IoT (Fuente)

2.1.3 Seguridad de los dispositivos IoT

Los dispositivos IoT como hemos comprobado en el apartado anterior tienen un gran número de aplicaciones. Esta expansión tecnológica provoca que los ciberdelincuentes realicen ataques a la red IoT, porque normalmente no se mantiene la seguridad de todos los dispositivos conectados y hay muchos vulnerables. Podemos clasificar la seguridad de IoT en:

Seguridad y protección de datos: Dado que la mayoría de los dispositivos IoT son inalámbricos y comparten información sensible en redes públicas, son de interés para los ciberdelincuentes que intentan robar esta información. Para evitar esto en lo posible, es necesaria la aplicación de una técnica que asegure el sistema.

Los algoritmos criptográficos son un buen método para garantizar la seguridad de la información en el IoT.

Autenticación y gestión de la identidad: Con objeto de que no pueda unirse cualquier dispositivo a nuestra red, es necesario autentificarlo y comprobar qué identidad tiene, para que precisamente no pueda lograr información de nuestro sistema.

De esta forma cada dispositivo de la red del IoT debe ser capaz de identificar otros dispositivos y autenticarlos. La identificación garantiza la identidad de los dispositivos antes de que se realice la comunicación entre ellos. Un mecanismo que permita a los dispositivos autenticarse mutuamente antes de cada interacción, es fundamental para garantizar su seguridad.

Privacidad: Los dispositivos pueden ser rastreados y esto supone un problema porque se incrementan las amenazas relacionadas con la privacidad. La seguridad de los datos es importante para que no sean utilizados por terceras personas.

A pesar de ello, también deben abordarse las cuestiones relacionadas con la propiedad de los datos. Para que el usuario se sienta cómodo formando parte del sistema IoT hay que tomar las medidas necesarias. La propiedad de la información recogida de los diferentes dispositivos debe establecerse desde un principio. El propietario debe tener garantizado que los datos no se utilizarán sin su consentimiento, especialmente cuando se compartan en Internet.

La privacidad de la información puede garantizarse mediante políticas de privacidad. Por lo tanto, cuando los dispositivos entran en contacto, deben enviar sus respectivas políticas de privacidad y estar de acuerdo con ellas antes de comunicar cualquier información.

2.1.4 Protección de los dispositivos IoT

Cualquier dispositivo, vehículo, aparato o sistema esté conectado a Internet, puede revelar información personal a un posible atacante. Al existir tal cantidad de dispositivos conectados, se presentan más oportunidades por los para los hackers de poner en peligro su seguridad.

Los dispositivos que no disponen de los recursos necesarios para conectarse a Internet cifrando el tráfico, no pueden gestionar certificados digitales o calcular hashes, mecanismos que forman parte de la seguridad que garantiza que un ciberatacante no pueda acceder a dichos dispositivos.

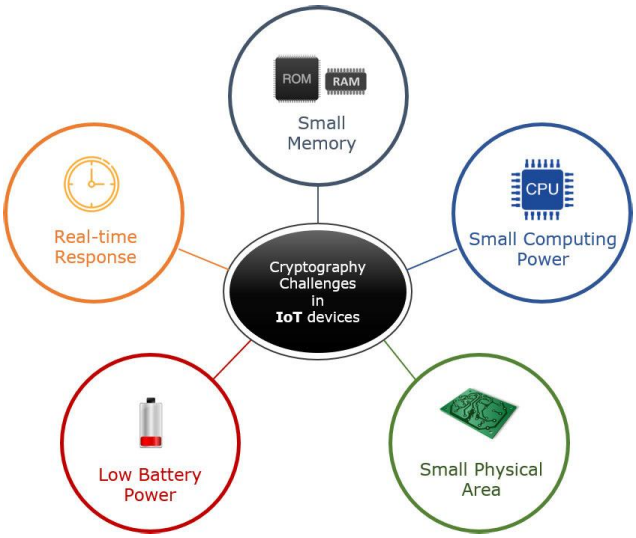


Figura 2-3 Desafíos de la Criptografía en dispositivos IoT (tomada de [1])

Los principales factores que afectan la seguridad de IoT son:

Escalabilidad: la personalización, la escalabilidad y la diversidad son desafíos importantes para la seguridad de los dispositivos. La escalabilidad es la capacidad de un dispositivo para adaptarse a un entorno cambiante según las necesidades que se puedan dar en el futuro. Es muy importante que los dispositivos IoT sean escalables en seguridad, ya que a la velocidad que se propaga esta tecnología, pueden quedarse obsoletos en poco tiempo.

Diversidad: el diseño de seguridad genérico de IoT se está volviendo un problema debido a la diversificación de las plataformas de los dispositivos de IoT. La mayoría de los dispositivos fabricados con fines de IoT no tienen una estandarización adecuada. Además, la conectividad a Internet no puede considerarse como segura.

Vida útil del dispositivo: la vida útil del dispositivo puede variar según las aplicaciones de IoT. Las técnicas de seguridad de datos o dispositivos deben adaptarse y diseñarse en función de capacidad de la batería que debe soportar el dispositivo IoT, así como el procesamiento que genera.

Sistemas de IoT de código abierto: la arquitectura y el software de código abierto se utilizan mucho en los sistemas IoT. La ventaja de utilizar código abierto es que los mismos módulos son implementables en una amplia gama de productos. Pero el inconveniente es que todo el sistema se verá comprometido o se detendrá si se produce un ataque, ya que puede afectar a todos los nodos de la red IoT.

Datos en la nube: el comercio en IoT se suele realizar con datos basados en la nube. Los dispositivos IoT están interconectados para manejarlos desde cualquier sitio, pero esto conlleva numerosos sensores, actuadores y unidades de procesamiento. El sensor está activo la mayor parte del tiempo así que transmite numerosos datos que deben ser almacenados, y se suele hacer en la nube. Este almacenamiento puede ser un problema, de ahí la importancia de contar con un buen sistema de seguridad que permita su protección.

Autenticación y emparejamiento: como hemos comentado, la red IoT está formada por muchos dispositivos, como sensores, etc. que se conectan entre sí formando una red. Cada sensor puede o no estar vinculado a los demás. Si están vinculados entre sí, compartirán datos, que deben estar cifrados, de lo contrario, un atacante puede manipular estos datos. Todos los nodos finales de los sensores están interconectados, por lo que un adversario si afecta a un nodo final, puede manipular todos los demás nodos conectados a la red.

2.1.5 El IoT y la Criptografía

El IoT presenta innovaciones debido a su capacidad inherente de conectar "cosas" inteligentes de un mundo físico a una arquitectura basada en la nube. La protección de los datos y su privacidad son fundamentales en los dispositivos IoT, y además esta tecnología crea nuevos retos de seguridad en materia de seguridad criptográfica, de credenciales y de gestión de identidades.

La criptografía se puede utilizar en varias áreas de una implementación de IoT. Se puede utilizar para proteger los canales de comunicación, por ejemplo, utilizando el protocolo criptográfico TLS (Transport Layer Security) para comunicaciones seguras.

También pueden utilizar la criptografía para cifrar y descifrar los datos del dispositivo, utilizando algoritmos de cifrado de clave única o simétrica, AES, PKI (Infraestructura de Clave Pública) o algoritmos de cifrado de clave asimétrica, como RSA (Rivest, Shamir, Adleman).

A menudo existe un compromiso entre el método de criptografía utilizado y la seguridad general del dispositivo, teniendo en cuenta los recursos con los que se disponen.

Los métodos de criptografía ligera suelen equilibrar rendimiento (throughput) contra el consumo de energía y la GE (Gate Equivalent, espacio físico) pero su seguridad no es tan completa como la que proporcionan los estándares criptográficos principales (como AES y SHA-256). No obstante, estos métodos también deben tener un bajo requerimiento de RAM (para realizar operaciones) y de ROM (datos almacenados en el dispositivo).

Algunas organizaciones de TI (Tecnología de la Información) no emplean la criptografía en IoT porque bloquea la visibilidad, lo que dificulta el análisis de la red y la resolución de problemas. Otros optan por no usarla porque creen que administrarla o configurarla es difícil. También hay algunas que la emplean para proteger sólo una parte de sus dispositivos.

Algunos expertos han refutado estas razones, argumentando que los beneficios de la criptografía superan a sus inconvenientes.

A continuación, se expondrán los distintos algoritmos de criptografía ligera y se comprobarán sus características fundamentales para comprobar si son aplicables a los distintos dispositivos IoT.

2.2 Criptografía.

2.2.1 Introducción a la criptografía

La criptografía ha sido usada a través de los años para mandar mensajes confidenciales cuyo propósito es que sólo las personas autorizadas, puedan entender el mensaje.

Alguien que quiere mandar información confidencial, aplica técnicas criptográficas para poder "esconder" el mensaje (se denomina cifrar o encriptar), envía el mensaje por una línea de comunicación que se supone segura, y después sólo el receptor autorizado puede leer el mensaje "escondido" (se denomina descifrar o desencriptar)

Antes del siglo XIX el uso de sistemas criptográficos se basaba en el uso de métodos (en ocasiones muy ingeniosos) que se mantenían en secreto entre las partes, estos métodos se denominan Clásicos.

Estos métodos cayeron en desuso con las técnicas de criptoanálisis. El criptoanálisis, en su versión más básica, se basa en el estudio frecuencial de un texto: en éste, hay caracteres mucho más frecuentes (por ejemplo, las vocales) que otros (por ejemplo, la 'w', la 'k'). Los textos cifrados conservaban esa distribución, y calculando las frecuencias de aparición de los caracteres en el cifrado, podía estudiarse a qué carácter podía corresponderse en realidad.

Con la aparición de computadores a partir de la segunda guerra mundial, el criptoanálisis se volvió sistemático, y supuso el desarrollo de un nuevo paradigma, con métodos de cifrado más potentes, donde el secreto dejó de ser el algoritmo, y pasó a ser la clave.

Una de las bases de operación de estos métodos fue romper la distribución frecuencial utilizada en el criptoanálisis, haciéndola uniforme. Para ello es necesario codificar cada carácter del texto plano o claro operándolo con la operación OR exclusiva (XOR ó $\oplus$) con un número al azar. Este principio de operación se mantiene en los métodos actuales.

A partir de la segunda mitad de la década de los años 70, nace la criptografía actual. En 1976 se inventa el sistema DES (Data Encryption Standard) y la criptografía se empieza a conocer más ampliamente, principalmente en el mundo industrial y comercial. Posteriormente en 1978 con el sistema RSA, se abre el comienzo de la criptografía en un gran rango de aplicaciones: en transmisiones militares, en transacciones financieras, en comunicación de satélite, en redes de computadoras, en líneas telefónicas, en transmisiones de televisión, etc.

La criptografía se divide en dos grandes ramas, la criptografía de clave privada o simétrica y la criptografía de clave pública o asimétrica, **DES** pertenece al primer grupo y **RSA** al segundo.

Tipo de Criptografía	Algoritmos
Criptografía de clave pública	RSA, ECC, ElGamal, DSA
Criptografía de clave privada	AES, DES, Triple DES, Blowfish, Serpent

Tabla 2-1 Algoritmos criptográficos más populares (tomada de [4])

2.2.2 Criptografía clásica

Desde el inicio de los tiempos, el ser humano ha querido mantener sus secretos y ha buscado mecanismos para no difundirlos a otras personas.

Por ejemplo, los sacerdotes egipcios usaron métodos criptográficos utilizando los jeroglíficos, ya que eran incomprensibles para el resto de la población, los babilonios con su escritura cuneiforme, el método de la escitala utilizado por los espartanos, Julio César que empleó unos de los primeros mecanismos criptográficos para sus comunicaciones militares, Leonardo Da Vinci que escribía de derecha a izquierda y con la mano zurda o Sir Francis Bacon o Edgar Allan Poe, ambos conocidos por su afición a los códigos criptográficos.

Los sistemas de cifrado anteriores a la II Guerra Mundial son los llamados mecanismos clásicos y corresponden a una época anterior al nacimiento de las computadoras.

Los métodos clásicos suelen dividirse en dos tipos:

- Cifrado por sustitución: Letras o grupos de letras que son sistemáticamente reemplazadas por otras letras o grupos de letras.

```
A  B  C  D  E  F  G  H  I  J  K  L  M  N  O  P  Q  R  S  T  U  V  X  Y  Z
Z  Y  X  V  U  T  S  R  Q  P  O  N  M  L  K  J  I  H  G  F  E  D  C  B  A
   Clave: Sustituir del final al principio


   Mensaje     E S T O   E S  U N  E J E M P L O
   Criptograma U G F K   U G  E L  U P U M J N K
```

- Cifrado por transposición: Se cambia el orden de las letras de acuerdo con un esquema bien definido.

```
   Clave: Grupos de 4 letras Transposición:    1234 ----4321

   Mensaje          S I S T E M A S      C L A S I C O S
   Criptograma      T S I S S A M E      S A L C S O C I
```

Así pues podemos afirmar que el paso de la Criptografía clásica a la moderna se da en la II Guerra Mundial, cuando el Servicio de Inteligencia aliado descifra los dos sistemas empleados por el ejército alemán, la máquina ENIGMA (utilizada para los mensajes de las unidades de combate) y el cifrado de Lorenz (utilizada en mensajería de alto nivel).

2.2.3 Criptografía moderna

Los sistemas criptográficos clásicos presentan una dificultad en cuanto a la relación complejidad y longitud de la clave y el tiempo necesario para encriptar y descifrar el mensaje, en cambio los métodos criptográficos modernos no presentan esta dificultad debido a los siguientes factores:

- Velocidad de cálculo: con la aparición de los ordenadores se dispuso de una potencia de cálculo muy superior a la de los métodos clásicos.
- Avance de las matemáticas: con sistemas criptográficos más estables y seguros.
- Necesidades de seguridad: surgieron muchas actividades nuevas que precisaban de la criptología para mantener el secreto.

De esta forma nacieron nuevos y complejos sistemas criptográficos, que se clasifican según el tratamiento del mensaje en:

- Cifrado en bloque: El cifrado se realiza en grupos de bits de longitud fija, llamados bloques.
- Cifrado en flujo: El cifrado se realiza convirtiendo el texto claro en texto cifrado bit a bit, mediante un generador de flujo de clave.

Además, según el tipo de clave utilizada para el cifrado, se clasifican en:

- Criptografía simétrica. Son aquellos en los que los procesos de cifrado y descifrado son llevados a cabo por una única clave.
- Criptografía asimétrica. Son aquellos en los que los procesos de cifrado y descifrado son llevados a cabo por dos claves distintas y complementarias.

Todos los algoritmos criptográficos clásicos son de carácter simétrico, ya que hasta mediados de los años setenta se empezó a utilizar la Criptografía Asimétrica.

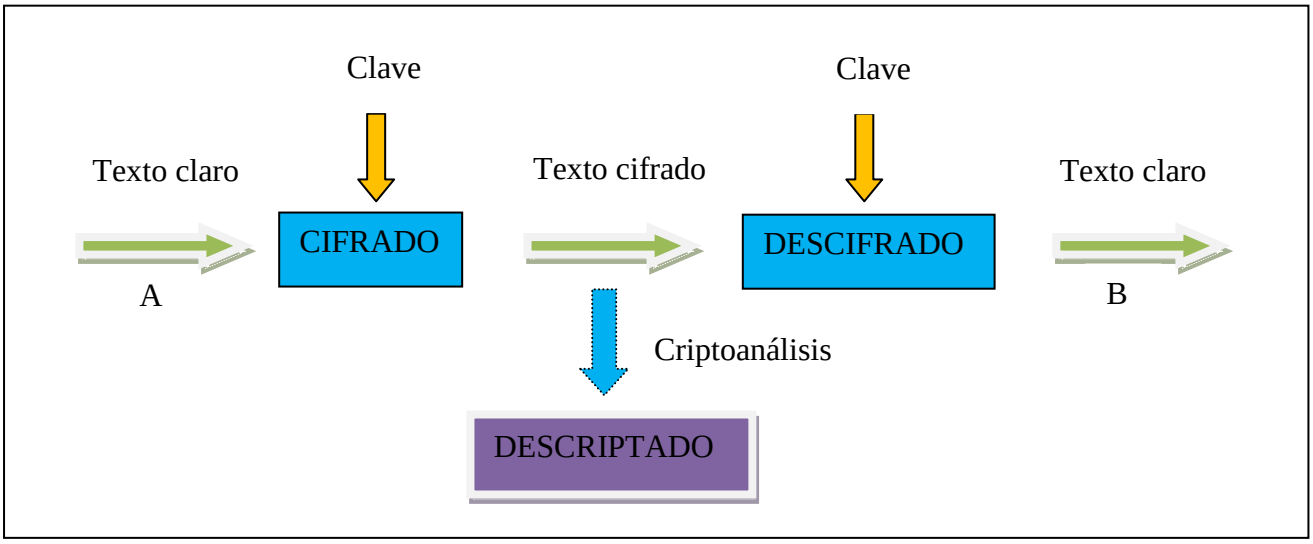


Figura 2-4 Proceso de cifra de un mensaje (Fuente: Elaboración Propia basada en [8])

2.2.3.1 Criptografía simétrica

Es el modelo tradicional y el más utilizado, y el que la mayoría de las personas identifican con el concepto general del "cifrado". Se basa en disponer de una clave secreta, k, que comparten las dos entidades que quieren cambiar información. Existe un algoritmo de cifrado, E, que convierte un texto plano m en un texto cifrado: **C = E(k,m)** y otro de descifrado, D, de tal modo que poseyendo la clave, y sólo poseyéndola, es factible descifrar correctamente el texto cifrado y recuperar el texto plano **D(k, c) = m.**

Se denomina criptografía simétrica porque tanto para cifrar como para descifrar se necesita la misma clave.

Los sistemas de cifrado simétrico presentan dos grandes desventajas: la distribución de las claves (que pueden ser interceptadas) y la dificultad de almacenar y proteger muchas claves diferentes.

En la criptografía simétrica, también llamada criptografía de clave privada, es esencial que la clave se mantenga en secreto porque cualquiera que la tenga puede leer cualquier mensaje que se envíe. Esto

contrasta con la criptografía asimétrica, que utiliza dos claves, una clave privada que está protegida y una clave pública que está disponible para todos.

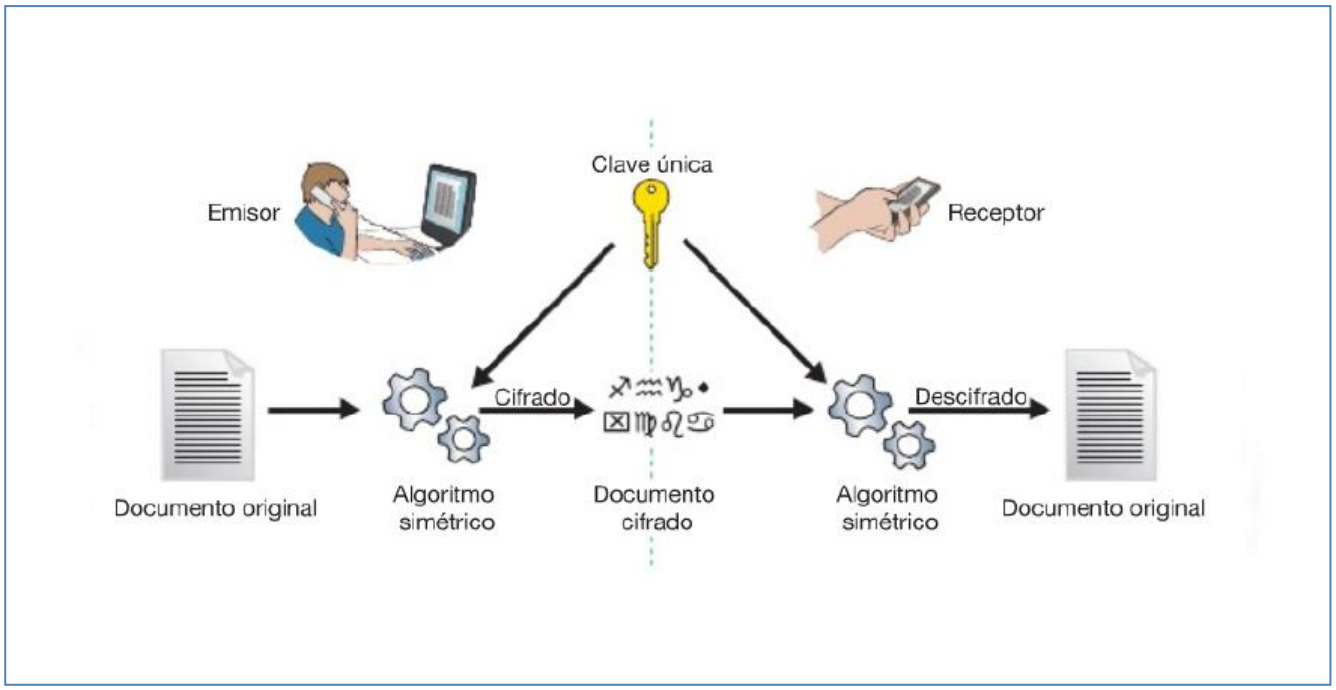


Figura 2-5 Gráfico de Criptografía Simétrica (tomada Libro de Mcgraw Hill)

2.2.3.1.1 Criptografía de flujo

Los algoritmos de cifrado de flujo emplean una secuencia pseudoaleatoria (generada a partir de la clave), y permiten cifrar mensajes de longitud arbitraria, combinando el mensaje con alguna función simple reversible (normalmente el OR exclusivo) con el texto en claro bit a bit, es decir, carácter a carácter. Obviamente estos criptosistemas no proporcionan seguridad perfecta, ya que mientras en el cifrado de Vernam[3] el número de posibles claves era tan grande como el de posibles mensajes, cuando empleamos un generador pseudoaleatorio, tenemos como muchas tantas secuencias distintas como posibles valores iniciales de la semilla tengamos.

La debilidad que existe en los métodos de cifrado de flujo es que si un atacante conoce parte del texto claro, podría sustituirlo por otro sin que lo advierta el destinatario.

Si suponemos que m_i es un trozo del mensaje original conocida por el atacante, y c_i la parte del mensaje cifrado correspondiente a ese trozo, sabemos que:

$$c_i = m_i \oplus o_i$$

siendo o_i el trozo de secuencia pseudoaleatoria que fue combinado con el texto en claro. Haciendo el siguiente cálculo podemos obtener un valor c'_i

$$c'_i = c_i \oplus m_i \oplus m'_i = o_i \oplus m'_i$$

Cuando el destinatario calcule $c'i \oplus o_i$, obtendría el valor falso introducido, m'_i, en lugar del texto en claro original. Esta circunstancia aconseja emplear los métodos de cifrado de flujo en combinación con métodos que garanticen la integridad del mensaje.

[3] **Cifrado de Vernam:** Consiste en emplear una secuencia aleatoria de igual longitud que el mensaje, que se usaría una única vez.

Los generadores que se emplean como cifrado de flujo pueden dividirse en dos grandes grupos, síncronos y asíncronos, dependiendo de los parámetros que se empleen para calcular el valor de cada trozo de la secuencia pseudoaleatoria.

En el cifrado síncrono el generador de números aleatorios no depende del mensaje a cifrar, con lo que el emisor y el receptor deben sincronizar la clave antes de empezar la transmisión y en el caso de pérdida de un bit se pierde el sincronismo. Un generador de secuencia asíncrono o auto-sincronizado es aquel en el que la generación de la clave es función del mensaje utilizado anteriormente, esto tiene la ventaja de permitir la sincronización automática en situaciones de pérdida de sincronismo.

Los cifrados de flujo se utilizan por su velocidad y simplicidad de implementación en hardware y en aplicaciones donde el texto sin formato viene en cantidades de longitud desconocida, como una conexión inalámbrica segura y por lo tanto no podemos aplicar cifrados de bloque.

También son convenientes en las telecomunicaciones, por ejemplo, en una conversación de telefonía móvil la voz se digitaliza (se convierte a un flujo de bits) y se envía cifrada por la red de comunicaciones. Para no ralentizar la conversación, el proceso de cifrado debe ser lo bastante rápido como para no añadir retraso a la comunicación.

Los cifrados de flujo más conocidos son: RC4, SEAL y A5.

1.- **RC4** [36], es un esquema de cifrado de flujo simétrico diseñado por Ron Rivest en 1987. Tiene las ventajas de ser un algoritmo muy simple, rápido de ejecutar, fácil de implementar en hardware y software y con una seguridad relativamente alta. La longitud de la clave es variable, pero generalmente se utilizan 256 bytes. Estas características lo han convertido en uno de los esquemas de cifrado más utilizados del mundo, se usa en algunos de los protocolos más populares como TLS/SSL, sin embargo, RC4 hace tiempo que no es considerado un algoritmo seguro porque por utiliza el mismo esquema de cifrado de WEP (Wired Equivalent Privacy), que hoy en día puede ser violado fácilmente mediante software.

Curiosamente RC4 es usado aún en aproximadamente la mitad de transmisiones TLS que ocurren en el mundo actualmente, desde para consultar el correo hasta para realizar transferencias bancarias.

Ataque a RC4: Itsik Mantin y Adi Shamir, demostraron que el segundo byte de salida de RC4 estaba fuertemente sesgado hacia cero, es un ejemplo clásico de ataque distintivo (Distinguishing Attack) en un cifrado de flujo popular.

2.- **SEAL** [37], es un algoritmo de cifrado simétrico alternativo más rápido que el DES, 3DES y AES propuesto por IBM en 1994. Utiliza una clave de cifrado de 160 bits y tiene un menor impacto en la CPU en comparación con otros algoritmos basados en software ya que su estructura está especialmente diseñada para funcionar de manera eficiente en computadoras con una longitud de palabra de 32 bits.

Su funcionamiento se basa en un proceso inicial en el que se calculan los valores para unas tablas a partir de la clave, de forma que el cifrado propiamente dicho puede llevarse a cabo de una manera realmente rápida. Una característica muy útil de este algoritmo es que no se basa en un sistema lineal de generación, sino que define una familia de funciones pseudoaleatorias, de tal forma que se puede calcular cualquier porción de la secuencia suministrando únicamente un número entero n de 32 bits. Con ese número, junto con la clave k de 160 bits, el algoritmo genera un bloque k(n) de L bits de longitud. De esa forma, cada valor de k da lugar a una secuencia total de L/232 bits, compuesta por la unión de los bloques k(0); k(1)…k(232 − 1).

SEAL se basa en el empleo del algoritmo SHA (Secure Hash Algorithm) para generar las tablas que usa internamente.

3- **A5** [38], es un algoritmo no publicado propuesto en 1994. Tiene dos versiones, la A5/1 fuerte (Europa) y la A5/2 débil (exportada). El uso habitual de este algoritmo lo encontramos en el cifrado del enlace entre el teléfono móvil de un abonado y la estación base en el sistema de telefonía móvil GSM.

Con más de 100 millones de usuarios de telefonía en Europa y otros 100 millones de usuarios en el resto del mundo, el sistema fue atacado en diciembre de 1999 y su futuro es incierto.

Una conversación en el sistema GSM entre A y B se transmite como una sucesión de tramas. Cada trama consta de 228 bits de los cuales los 114 primeros representan la comunicación digitalizada en el sentido A → B, mientras que los restantes 114 bits representan la comunicación digitalizada en el sentido contrario B → A. Cada 4.6 milisegundos se envía una nueva trama. Previamente a la transmisión de cualquier trama se realiza un proceso de sincronización entre A y B.

El generador A5 produce 228 bits pseudoaleatorios de cada trama que se sumarán, bit a bit, mediante la operación XOR con los 228 bits de conversación en claro dando lugar a los 228 bits de conversación cifrada.

2.2.3.1.2 Criptografía de bloques

Los cifrados de bloque se diferencian de los de flujo porque cifran los datos en fragmentos o bloques, en lugar de un carácter a la vez como ocurre en los cifrados de flujo. Un cifrado de bloque generalmente se considera más seguro que un cifrado de flujo porque es más aleatorio, mientras que un cifrado de flujo funciona más rápido cuando el texto es corto.

Una cifrado de bloque sobre el alfabeto binario es una proyección biyectiva E_k: $\{0,1\}^n \rightarrow \{0,1\}^n$ indexada por una clave k. Las cifras de bloque permiten cifrar datos de tamaño arbitrario dividiéndolos en bloques de n bits, añadiendo padding, es decir rellenando todos los bits, si es necesario y aplicando la función Ek.

Su algoritmo tiene la propiedad de combinar una secuencia de caracteres del texto plano con una secuencia de caracteres de la clave, un carácter a la vez, utilizando una función invertible. Así:

$$E_k(m_i) = c_i$$

$$E(_k, _{vi})(m_i) = m_i \oplus k_i$$

$$D(_k, _{vi})(c_i) = m_i$$

$$D(_k, _{vi})(c_i) = c_i \oplus k_i$$

Siendo $\mathbf{E(_k, _{vi})}$ la Función de Cifrado para la clave $_k$ y vector de inicialización v_i; ⊕ la función XOR; m_i el carácter $_i$ del mensaje en claro; k_i el carácter $_i$ de la clave y c_i el carácter $_i$ del texto cifrado.

A su vez $\mathbf{D(_k, _{vi})}$ es la Función de Descifrado. Su finalidad es impedir mensajes iguales sean cifrados idénticamente, así si un atacante estuviese “observando” el canal de comunicaciones no advertiría una retransmisión.

Actualmente los sistemas de cifrado en bloque se clasifican, según el tipo de claves, en dos tipos o familias principales, los de clave simétrica y los de clave pública. En ambos, se trata de mantener la confidencialidad e integridad de los mensajes.

Algunos ejemplos de cifrados de bloque son el Estándar de cifrado de datos (DES), el Estándar de cifrado avanzado (AES), el Algoritmo internacional de cifrado de datos (IDEA) y el sustituto de DES (TWOFISH)

1) **DES** es un cifrado de bloque convencional que utiliza una clave simétrica de 56 bits y fue inventado en los años setenta. Los datos se cifran en bloques de 64 bits pero sólo cincuenta y seis de estos se utilizan efectivamente en el proceso de cifrado/descifrado. Los otros ocho bits sólo se usan para llevar a cabo comprobaciones de paridad.

Cada vez que DES cifra un bloque de texto sin formato de 64 bits, lo ejecuta a través del algoritmo 16 veces o rondas. Debido a la pequeña clave de 56 bits, DES ya no se considera seguro para muchas aplicaciones.

Una de las variantes del DES creadas con el objetivo de incrementar la complejidad de un ataque por fuerza bruta es el Triple DES que utiliza tres claves.

La longitud de la clave puede ser de hasta 168 bits ($56 * 3$). El 3DES se puede implementar con 4.600 GE, lo que requeriría 62 ciclos de reloj para la operación de cifrado. En cambio, se puede encontrar una implementación de la versión del DES que requiere sólo 2.310 GE y que utiliza 144 ciclos de reloj para cifrar un bloque de entrada.

Otra de las variantes del DES que se desarrolló para mejorar su seguridad es el DESX.

A pesar de que con este esquema la longitud de la clave es de 184 bits (64 + 64 + 56), la longitud efectiva es bastante menor y está condicionada por la información que es capaz de obtener el atacante, en concreto, por el número de pares de texto claro y texto cifrado que es capaz de conseguir.

La versión original de DES y también DESX prevén la utilización de cajas S (cajas de sustitución), cuya implementación necesita una gran cantidad de memoria. Para obtener una versión de DES con menos requerimientos de espacio, se creó una versión optimizada conocida con el nombre de DESL, que reemplaza las ocho cajas S originales por una única caja S diseñada de nuevo.

De la combinación del DESL con DESX surgió el sistema de cifra DESXL, que se ha implementado con sólo 2.169 GE. Tanto DESL como DESXL son considerados cifrados ligeros.

En 1997 el NIST no certifica al DES y convoca un concurso público para un nuevo estándar: el AES. En octubre del año 2000 el NIST elige el algoritmo belga RIJNDAEL como nuevo estándar para los algoritmos de cifra en bloque del siglo XXI. Es software de libre distribución y está disponible desde finales del año 2001.

<u>Ataques contra DES [40]:</u>

Ataque de búsqueda exhaustiva: Tiene un tamaño de clave de 56 bits, que no puede afrontar ataques de fuerza bruta con la potencia de cómputo actual.

También ha sufrido ataques de tipo diferencial (el numero de claves se reduce a 2^{47}) y Ataques lineales (probar 2^{43} claves)

2) **AES** fue seleccionado en octubre de 2000 por el NIST para reemplazar a estándar que existía en ese momento DES. AES utiliza tres claves diferentes para realizar múltiples rondas de cifrado en bloques de texto sin formato de 128 bits. Los tamaños de las claves son de 128, 192 y 256 bits de longitud aunque empezó con 56 pero con las primeras pruebas se demostró que, en un tiempo relativamente corto, la potencia de los procesadores podría romper el cifrado más débil y por tanto con menor número de bits en periodos de tiempo cada vez más bajo.

El AES fue diseñado con el requisito de prestar la mayor eficiencia. El AES (128 bits) en modo de sólo cifrado se puede implementar utilizando 3.100 GE y necesita 1.044 ciclos de reloj para realizar el cifrado de un bloque de datos y tiene una naturaleza abierta, lo cual significa que se puede usar tanto en lo público como en lo privado, sea para fines comerciales o no.

Los bloques de 128 bits se organizan en una matriz de cuatro por cuatro con cada byte en una posición. Ocho bits por byte nos dan los 128 bits mencionados y por ello al cifrar la información no se

altera el tamaño de la misma gracias a las matrices. AES es un sistema de sustitución y permutación, el cual debe su alta seguridad gracias a que la clave inicial o semilla le va a servir a través de una fórmula, generar claves nuevas, que al mismo tiempo se utilizarán para codificar los datos.

Entre los tres tipos de cifrados AES la única diferencia es precisamente la longitud de la clave, por lo que si comparamos 128 bits con 256 bits tendremos una clave el doble de larga en este último en lo que a cantidad de bits se refiere. Esto se traduce en que la clave va a tener 2^{256} valores distintos y por tanto el tiempo necesario para descifrarla será mucho más alta, tanto que incluso el ordenador más potente tardaría años en conseguirlo a través de técnicas de descifrado avanzado y el coste en tiempo no compensa a lo que se puede obtener.

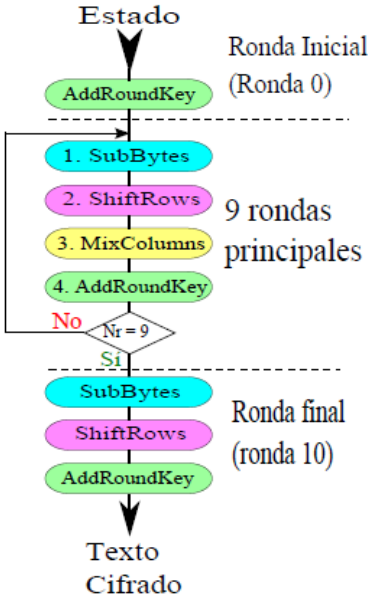


Figura 2-6 AES (Tomada de [62])

El cifrador aplica al Estado cuatro operaciones durante un número determinado de rondas. Dicho número de rondas (Nr) viene definido por la longitud de clave utilizada, siendo Nr = 10 para una longitud de clave de 128 bits, Nr = 12 para 192 bits y Nr = 14 para 256 bits. Las cuatro operaciones realizadas en el cifrado son denominadas: SubBytes. ShiftRows. MixColumns. AddRoundKey.

Ataques contra AES:

Se rompió empleando un ataque Meet-in-the-Middle, combinándolo con un ataque Biclique. [59]

"En el sentido estrictamente académico, el algoritmo está «roto» porque se ha reducido en 2 bits el espacio de claves necesario para calcular la clave por fuerza bruta. Sin embargo, «roto» no significa que no pueda ser usado con seguridad todavía. Por ejemplo, un ataque contra una clave de 128 bits requiere diez millones de años empleando un billón de equipos probando cada uno de ellos un billón de claves. Al reducir en dos bits dicha clave, el tiempo se reduciría a 3 millones de años."

3) **IDEA (*International Data Encryption Algorithm*)**, que se usa ampliamente en Europa y utiliza una clave de 128 bits para realizar ocho rondas de cifrado en bloques de texto sin formato de 64 bits, por lo que es mucho más seguro que DES.

Consiste en ocho transformaciones idénticas (cada una llamada una ronda) y una transformación de salida (llamada media ronda). Gran parte de la seguridad de IDEA deriva del intercalado de tres operaciones: XOR bit a bit, suma módulo 2^{16} y multiplicación módulo $2^{16}+1$.

El algoritmo de desencriptación es muy parecido al de encriptación, por lo que resulta muy fácil y rápido de programar, aportando su longitud de clave una seguridad fuerte ante los ataques por fuerza bruta (prueba y ensayo o diccionarios).

Este algoritmo es de libre difusión para uso no comercial y no está sometido a ningún tipo de restricciones o permisos nacionales, por lo que se ha difundido ampliamente, utilizándose en sistemas como UNIX y en programas de cifrado de correo como PGP.

Ataque contra IDEA:

El ataque por fuerza bruta resulta impracticable, ya que sería necesario probar 1038 claves, cantidad imposible de manejar con los medios informáticos actuales.

No se han informado de debilidades frente al criptoanálisis lineal o algebraico. Se han encontrado algunas claves débiles, hay que evitarlas si queremos más seguridad.

En 2011, el IDEA de 8,5 rondas se rompió mediante un ataque Meet-in-the-middle [63].

En 2012 se rompió utilizando otra variante de MITM, usando una estructura de grafo bipartitos completos, con una reducción de la fuerza criptográfica de aproximadamente 2 bits; sin embargo, este ataque no amenaza, en la práctica, la seguridad de IDEA.

4) **TWOFISH** [64], fue desarrollado por Counterpane Labs y presentado al concurso del NIST que buscaba un sustituto para DES. El tamaño de bloque es de 128 bits y se utiliza una única clave para el cifrado y el descifrado que puede ser de cualquier longitud hasta 256 bits.

Twofish se relaciona con el método de cifrado por bloques anterior Blowfish. Las características distintivas de Twofish son el uso de S-boxes fuertes cuidadosamente seleccionadas diferenciándose de Blowfish que a veces utiliza claves débiles. Twofish coge prestados algunos elementos de otros diseños y utiliza la misma estructura de Feistel que el DES, de esta forma en cada ronda, la mitad del bloque de texto se envía a través de una función F, y luego se aplica XOR con la otra mitad del bloque de texto.

En la mayoría de las plataformas de software Twofish es ligeramente más lento que Rijndael para las claves de 128 bits, pero algo más rápido para las claves de 256 bits y es flexible, y por lo tanto se puede utilizar en aplicaciones de red donde las claves se cambian con frecuencia y en aplicaciones donde hay poca o ninguna RAM y ROM disponibles.

Ataques contra TWOFISH:

No se conoce un ataque contra Twofish salvo la búsqueda de claves por la fuerza bruta.

2.2.3.2 Criptografía asimétrica (o de clave pública)

La criptografía simétrica emplea la misma clave para cifrar y descifrar. Su principal ventaja es su velocidad y por esta razón, es la ideal para trabajar con grandes cantidades de datos. Sin embargo, como comentamos anteriormente, exige que el remitente y el destinatario hayan intercambiado previamente la clave por algún otro medio, lo que constituye un verdadero problema en entornos tan grandes como Internet y también presenta un problema la dificultad de almacenar y proteger muchas claves diferentes.

El problema del conocimiento de la clave secreta que tiene el cifrado simétrico, se resuelve con el cifrado asimétrico y los algoritmos de clave pública.

La criptografía asimétrica utiliza dos tipos de claves: las públicas, que pueden distribuirse abiertamente y entregarse a cualquier persona, y las privadas, que sólo conoce el propietario. La generación de las claves públicas depende de algoritmos criptográficos basados en funciones matemáticas unidireccionales. De esta forma, la clave pública puede distribuirse abiertamente sin comprometer la seguridad, ya que para lograr una seguridad efectiva el requisito es mantener la clave privada.

De esta forma, cualquier persona puede cifrar un mensaje utilizando la clave pública del receptor, pero el mensaje cifrado sólo puede descifrarse con la clave privada del destinatario. Además, los métodos criptográficos garantizan que esa pareja de claves sólo se puede generar una vez, de modo que se puede asumir que no es posible que dos personas hayan obtenido casualmente la misma pareja de claves.

Los sistemas de cifrado asimétrico también presentan dos grandes desventajas: para una misma longitud de clave y mensaje, se necesita mayor tiempo de proceso, las claves deben ser de mayor tamaño que las simétricas y el mensaje cifrado ocupa más espacio que el original.

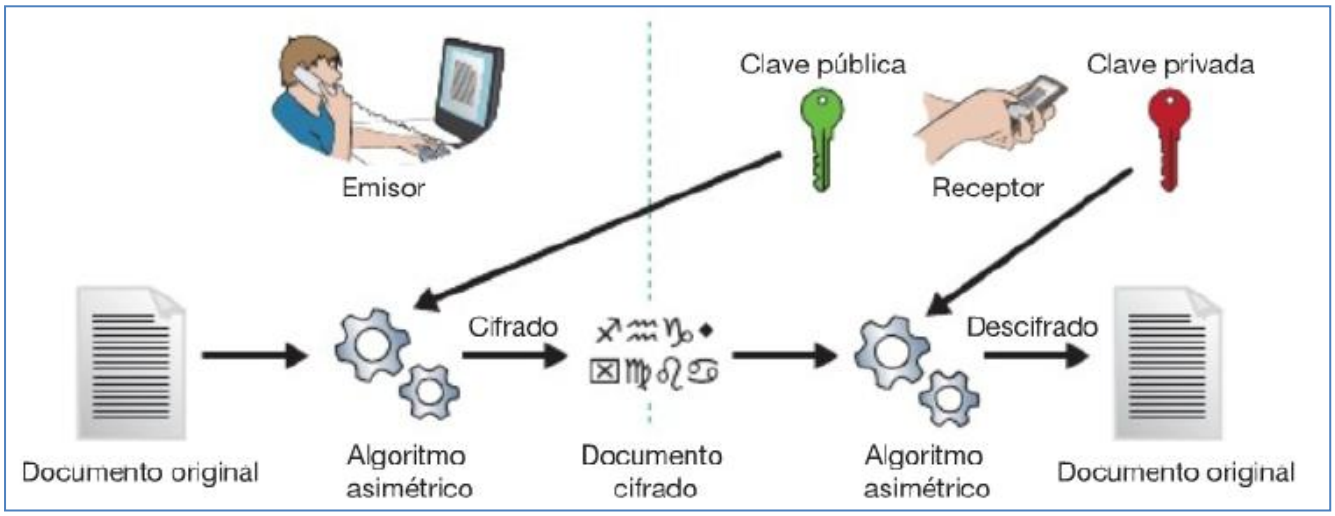


Figura 2-7 Gráfico de Criptografía Asimétrica (tomada Libro de Mcgraw Hill)

Algunos cifrados asimétricos son RSA y Diffie-Hellman.

1) RSA: en criptografía, RSA es un sistema criptográfico de clave pública desarrollado en 1977 por Rivest, Shamir y Adelman (de aquí el nombre RSA).

La seguridad de este algoritmo radica en la dificultad para factorizar grandes números enteros. Los mensajes enviados se representan mediante números y el funcionamiento se basa en el producto (conocido) de dos números primos grandes elegidos al azar y mantenidos en secreto. Actualmente estos primos son del orden de 10^{200}, y se prevé que su tamaño aumente con el aumento de la capacidad de cálculo de los ordenadores.

Como en todo sistema de clave pública, cada usuario posee dos claves de cifrado: una pública y otra privada. Cuando se quiere enviar un mensaje, el emisor busca la clave pública del receptor, cifra su mensaje con esa clave, y una vez que el mensaje cifrado llega al receptor, este se ocupa de descifrarlo usando su clave privada.

El esquema de cifrado utiliza RSA parte de que:

$m^{ed} \equiv m \pmod n$ para m entero.

El descifrado funciona porque $c^d \equiv (m^e)^d \equiv m \pmod n$. La seguridad radica pues en la dificultad de calcular un texto claro m a partir de un texto cifrado cm^e mod n y el parámetro n.

<u>Ataque a RSA:</u>

Se considera seguro, mientras no se halle la manera de descomponer un número grande en producto de números primos. Se cree que la computación cuántica podría realizar estos cálculos y romper el algoritmo.

2) Diffie-Hellman: Este algoritmo permite a las dos partes acordar una clave para cifrar los mensajes que quieren enviarse. La seguridad de este algoritmo depende de la suposición de que es fácil elevar un número a una cierta potencia, pero difícil calcular qué potencia se utilizó dado el número y el resultado.

Se emplea generalmente como medio para acordar claves simétricas que serán empleadas para el cifrado de una sesión.

El funcionamiento de este algoritmo se resume a continuación:

- Sea **q** un número primo muy grande
- Sea ɣ una primitiva de **q**
- Ana elige un número A y transmite XA= ɣ A mod q
- Juan escoge un número B y transmite XB = ɣ B mod q
- Ana calcula la clave de sesión K = (XB)A modulo q
- Juan calcula la clave de sesión K = (XA)B modulo q
- Se establece sesión con la clave K

Ataque a Diffie-Hellman

Su principal vulnerabilidad es ante los ataques Man-In-The-Middle.

2.2.3.3 Funciones Hash

Los algoritmos Hash, o de resumen, constituyen un tipo especial de criptosistemas ya que no utilizan el concepto de clave. Para estos algoritmos existe un nuevo término llamado: fingerprint o huella digital o resumen o hash.

Una función Hash toma un mensaje de entrada de longitud arbitraria y genera un código de longitud fija. La salida de longitud fija se denomina hash del mensaje original.

Los criptosistemas Hash presentan las siguientes características:

- Unidireccionalidad: este concepto significa que deberá ser computacionalmente muy difícil, por no decir imposible, obtener el mensaje M (original).
- Compresión: a partir de un mensaje de cualquier longitud, el hash H(M) debe tener una longitud fija. Normalmente mucho menor.
- Coherente: la misma entrada (mensaje original) siempre deberá producir la misma salida (mensaje original).
- Facilidad de Cálculo: debe ser fácil calcular la función Hash H(M) a partir de un mensaje M.
- Único: casi imposible encontrar dos mensajes que generen el mismo hash.
- Difusión: el resumen H(M) debe ser una función compleja de todos los bits del mensaje M.

A continuación vemos algunos de los principales algoritmos criptográficos de tipo hash:

1) MD5 (Message Digest Algorithm 5, Algoritmo de Ordenación de Mensajes 5): es un algoritmo desarrollado por RSA Data Security, Inc. MD5 es una función hash de 128 bits, que toma como entrada un mensaje de cualquier tamaño y produce como salida un resumen del mensaje de 128 bits.

El primer paso del algoritmo divide el mensaje en bloques de 512 bits. El último bloque o si el mensaje completo es menor a 512 bits, se formatea para tener un tamaño de 512 bits agregando bits "0" más la longitud del tamaño del mensaje.

Además, se tiene un buffer de 4 palabras de 32 bits. Se realizan 4 rondas de procesamiento donde el algoritmo toma bloques de 512 bits de la entrada y los mezcla con los 128 bits del buffer. Este proceso se repite hasta que todos los bloques de entrada han sido consumidos. El valor resultante en el buffer es el hash del mensaje.

Características:

- Irreversibilidad, es decir, el texto original no se puede obtener mediante texto cifrado.
- Inmutabilidad, es decir, el mismo texto original, el texto cifrado obtenido por el algoritmo MD5 es siempre el mismo.
- Hash, es decir, realizar pequeños cambios en el texto original puede provocar que el texto cifrado final se modifique por completo.

Ataque a MD5:

En principio se consideró seguro, pero se ha descubierto algunos métodos sencillos para generar colisiones de hash (cuando dos entradas distintas a una función de hash producen la misma salida).

También se puede atacar con fuerza bruta porque el tamaño del hash (128 bits) es pequeño.

2) SHA: la familia SHA (Algoritmo de Hash Seguro) es un sistema de funciones hash criptográficas relacionadas de la Agencia de Seguridad Nacional de los Estados Unidos y publicadas por el NIST.

El primer miembro de la familia fue publicado en 1993 y fue llamado oficialmente como SHA. Sin embargo, hoy en día, se le llama SHA-0 para evitar confusiones. Existen cuatro variantes que se han publicado desde entonces cuyas diferencias se basan en un diseño algo modificado y rangos de salida incrementados: SHA-224, SHA-256, SHA-384, y SHA-512.

Ataque a SHA [65]:

SHA-1 no se ha encontrado ningún ataque efectivo. No obstante, en el año 2004, un número de ataques significativos fueron divulgados sobre funciones criptográficas de hash con una estructura similar a SHA-1; esto ha planteado dudas sobre la seguridad a largo plazo de SHA-1.

SHA-0 y SHA-1 producen una salida resumen de 160 bits de un mensaje, que puede tener un tamaño máximo de 264 bits, y se basa en principios similares a los usados en MD5.

SHA-2 produce una salida resumen de 256 (para SHA-256) o 512 (para SHA-512) y difiere a SHA-1 en que el algoritmo contempla algunas constante adicionales; así mismo, el tamaño del resumen es diferente al igual que el número de rondas.

SHA-3: En 2015 fue seleccionado por NIST como estándar de hashing para reemplazar al estándar MD5. Es una función de hash criptográfica sobre la base de otras funciones como RadioGatún y Panamá. SHA3 admite salidas de 224, 256, 384 y 512 bits, su código es de dominio público y está definido como estándar en FIPS 202 (SHA-3).

Se cree que dentro de muy poco cuando la capacidad de computación lo permita, se realizan ataques de colisión de prefijos en SHA-1, así que se debe escoger SHA-2 o SHA-3 que son más seguros.

2.2.4 Criptografía ligera

Como hemos visto, la inmensa mayoría de dispositivos IoT desplegados por el mundo cuentan con pocos recursos, pero a su vez, necesitan un mínimo de seguridad en la transmisión de los datos, la criptografía ligera puede ofrecer soluciones de seguridad para estos dispositivos. Para ello es necesaria una implementación de un algoritmo criptográfico que deje la menor huella de software y hardware, el problema radica en la complejidad computacional de estos cifrados.

Así pues, el objetivo de la criptografía ligera es permitir que la información de los usuarios contenida en una amplia gama de dispositivos, como los sistemas de seguridad de los vehículos, los contadores inteligentes, los dispositivos instalados en pacientes de forma inalámbrica, el Internet de las Cosas (IoT), los Sistemas de Transporte Inteligentes (ITS), los sistemas de monitorización, etc., sea segura. Por ejemplo: es posible para un pirata informático provocar un mal funcionamiento de un dispositivo médico afectando la salud de la persona que lo necesita, como son las bombas de insulina o los marcapasos inteligentes.

Los sistemas embebidos (aquellos que se encuentran dentro de un sistema) se despliegan en un amplio abanico de dominios, como las industrias, los dominios privados y públicos, las infraestructuras críticas y las aplicaciones utilizables y portátiles y también tienen limitaciones.

El campo de la criptografía ligera es más ventajoso que el de los algoritmos criptográficos existentes (convencionales), teniendo en cuenta los problemas de seguridad para los dispositivos con recursos limitados. Además, con los avances en el Internet de las cosas (IoT), existe una necesidad persistente de reducir los costes de producción y el tamaño de estas primitivas criptográficas. Por lo tanto, para proporcionar funciones de seguridad de software y hardware resistentes a la manipulación en los sistemas embebidos con recursos limitados como RFID, redes de sensores y dispositivos IoT en general, son necesarias las implementaciones ligeras.

Estas implementaciones proporcionan una arquitectura más escalable y canalizada. Los cifrados requieren las puertas equivalentes mínimas[4], normalmente por debajo de 2000 GE, que son bajas en comparación con las implementaciones de AES que rondan alrededor de 2400-3500, y, por lo tanto, implementables en circuitos más pequeños y con requisitos de potencia mínimos. Un GE pequeño significa que el circuito es más barato y consume menos energía. Por lo tanto, AES es demasiado caro en comparación con los cifrados ligeros que son asequibles para los dispositivos pequeños.

La Criptografía ligera tiene los mismos cifrados que los de la criptografía tradicional: Algoritmos de Clave Pública, Algoritmos de Clave Privada, Algoritmos que trabajan con cifrado en Bloque y flujo, Funciones Hash y Mecanismos de Autenticación (Firma Digital).

Puede estar orientada a Hardware o Software, determinándose parámetros para evaluar y medir las implementaciones que apliquen a este tipo de criptografía. Por ejemplo para hardware se estudian el tamaño de los chips y el consumo de energía que se requiere. Para software, en cambio, se analizan la longitud del código y el uso y consumo de memoria RAM.

Se está avanzando mucho en este tipo de criptografía, pero no todos los algoritmos ligeros son eficientes a la hora de implementarlos.

A continuación se expondrán diversos algoritmos de criptografía ligera que se podrán ser utilizados dependiendo de las características como memoria y espacio físico del dispositivo y dependiendo de los ataques a los que esté expuesto.

[4] **GE:** Una puerta equivalente o GE corresponde al área del chip necesaria para implementar una puerta NAND de dos entradas. Las GE permiten especificar la complejidad de un circuito electrónico independientemente de la tecnología con la que este se haya creado.

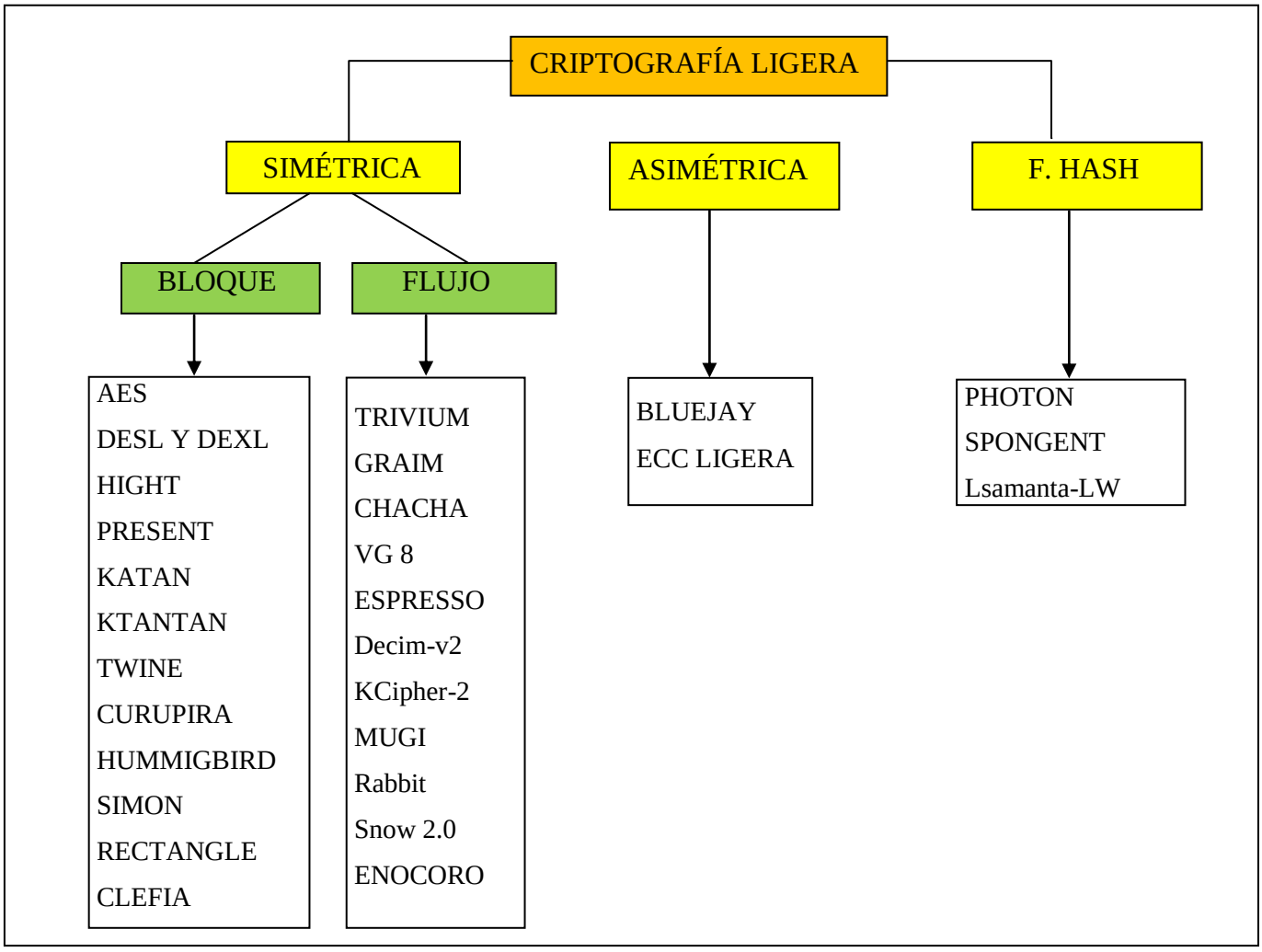


Figura 2-8 Algoritmos criptográficos ligeros (Fuente: Elaboración propia basada en [2])

2.2.4.1 Ejemplos de cifrados de bloques ligeros

1.- AES (Estándar de cifrado avanzado)

Se trata de un cifrado por bloques estandarizado por el NIST que opera con bloques de 128 bits y utiliza un tamaño de clave de 256 bits, 192 bits o 128 bits. El cifrado puede realizarse con diez, doce o catorce rondas en función del tamaño de la entrada. Las matrices de bytes se tratan como matrices de 4 x 4.

La implementación ligera de 128 bits de AES requiere 3400 GE y ofrece un rendimiento de 12,4 kB/s a una frecuencia de 100 kHz [16]. Sin embargo, un ataque Man-In-The-Middle (MITM) sigue siendo una amenaza para el AES ligero.

2.- DESL y DESXL

DESL se basa en el clásico DES pero a diferencia de DES, DESL utiliza una sola S-box[5] en lugar de las 8 S-boxes de DES. Los criterios de diseño de la única S-box de DESL hacen que el DESL sea resistente a los ataques criptoanalíticos más comunes. Esto permite guardar una parte de la ROM para el almacenamiento de tablas.

DESXL es una versión ligera del algoritmo DESX, que opera con bloques de 64 bits utilizando una clave de 184 bits. El cifrado con DESLX se realiza con 16 rondas. A diferencia de DES, DESX realiza transformaciones de los datos de entrada y salida con las subclaves específicas. Al igual que DESL, DESXL utiliza la misma S-box en lugar de 8 S-boxes DESX con lo cual es considerado como algoritmo ligero.

Los requisitos relativamente bajos de DESL/DESXL son sólo el resultado de la reducción de ocho veces los requisitos de ROM para el almacenamiento de tablas (ya que esta es la única diferencia entre DESL/DESXL y los algoritmos clásicos). Los autores de DESL/DESXL afirman que tal reducción de los requisitos es suficiente para utilizar los algoritmos propuestos en dispositivos con recursos limitados. DESXL sólo requiere 2168 GE y ofrece un rendimiento de 44,4 kB/s a 100 kHz de frecuencia [16].

DES ha tenido tres tipos de ataques, de tipo diferencial (numero de claves se reduce a 2^{47}), lineal (probar 2^{43} claves) y de fuerza bruta (probar 2^{56})

3.- HIGHT (High Security and Lightweight)

Es un cifrado de bloques ligero que opera con bloques de 64 bits utilizando una clave de 128 bits que se genera al cifrar y descifrar y utiliza la estructura GFS (General Feistel Structure). El cifrado con HIGHT se realiza con 32 rondas. Esta implementación ligera requiere 3048 GE y ofrece un rendimiento de 188,2 kB/s a una frecuencia de 100 kHz. Este cifrado por bloques es vulnerable a los ataques de saturación.

Ataque a HIGHT:

El vulnerable al criptoanálisis diferencial, al criptoanálisis Lineal, MITM/Biclique y al ataque de clave relacionada [ANEXO IV]

4.- PRESENT

Se considera un algoritmo criptográfico ultraligero que opera con bloques de 64 bits utilizando un tamaño de clave de 80 o 128 bits. El cifrado con PRESENT se realiza en 32 rondas. La capa de sustitución del SPN (Substitution–Permutation Network) utiliza actualmente cajas S de 4 bits para la entrada y la salida. Este cifrado por bloques es conocido por el alto nivel de seguridad que proporciona y por su simplicidad.

PRESENT requiere hasta 1.570 GE cuando se utiliza una clave de 80 bits y hasta 1.884 GE cuando se utiliza una clave de 128 bits, y ofrece un rendimiento de hasta 200 kB/s a una frecuencia de 100 kHz. El cifrado es vulnerable a los ataques diferenciales.

Ataques a PRESENT

Vulnerable a ataque diferenciales, MITM/Biclique, de clave relacionada y de canal lateral/fallos diferenciales. [ANEXO IV]

[5] **S-Box:** (Substitution box o Caja de sustitución) Una S-Box toma un número m de bits de entrada y los transforma en n bits de salida

5.- PRINCE

Es un cifrado ligero conocido por utilizar la propiedad α-reflejo: el cifrado y el descifrado son iguales, pero cada uno utiliza una clave diferente para reducir los requisitos de diseño. Este cifrado opera sobre bloques de 64 bits utilizando una clave de 128 bits, y utiliza la estructura SPN. El cifrado mediante PRINCE se produce en 12 rondas.

Requiere hasta 3491 GE y ofrece un rendimiento de hasta 533,3 kB/s a una frecuencia de 100 kHz. El cifrado es vulnerable a los ataques de reflexión.

Ataques a PRINCE:

Sólo es vulnerable al ataque diferencial. [ANEXO IV]

6.- KATAN Y KTANTAN

KATAN es una familia de cifrados por bloques orientados al hardware muy eficientes con un tamaño de clave de 80 bits. Son seis cifrados divididos en dos tipos, KATAN que se compone de tres cifrados con un tamaño de bloque de 32, 48 o 64 bits (KATAN32, KATAN48 y KATAN64) y KTANTAN que contiene los otros tres cifrados con los mismos tamaños de bloque y es más compacto en hardware, ya que la clave está grabada en el dispositivo (y no se puede cambiar).

El cifrado mediante KATAN y KTANTAN se produce en 254 rondas. La principal diferencia entre los diseños es que KTANTAN necesita aproximadamente la mitad de las puertas equivalentes que KATAN para la implementación hardware con todos los tamaños de bloque.

- Para bloques de 32 bits, KTANTAN necesita 462 EGs, KATAN necesita 802 EGs.
- Para bloques de 48 bits, KTANTAN necesita 588 EGs, KATAN necesita 927 EGs.
- Para bloques de 64 bits, KTANTAN necesita 688 EGs, KATAN necesita 1054 EGs.

KATAN se basa en el registro de desplazamiento de retroalimentación lineal (LFSR). LFSR se utiliza para contar las rondas y para detener el cifrado cuando sea necesario.

KATAN y KTANTAN ofrecen rendimientos de 12,5, 18,8 y 25,1 para kB/s a una frecuencia de 100 kHz al cifrar bloques de 32, 48 y 64 bits, respectivamente. Ambos cifrados son vulnerables a los ataques diferenciales.

Ataque a KATAN: [ANEXO IV]

MITM/Biclique

Ataque a KTANTAN: [ANEXO IV]

Ataque de clave relacionada

7.- TWINE [66]

Es un cifrado de bloques ligero de 64 bits que admite claves de 80 y 128 bits. Está diseñado principalmente para adaptarse a un hardware extremadamente pequeño, pero proporciona un notable rendimiento bajo software embebido. Además, permite una implementación compacta de cifrado y descifrado unificado. Esta característica se debe principalmente al uso de un cifrado Feistel con muchos subbloques.

En cuanto a su seguridad, no se comporta mal en los ataques diferenciales y de saturación que afectan mucho a los cifrados Feistel.

8.- CURUPIRA [43]

El algoritmo es relativamente pequeño ocupando 96 bits, el tamaño de bloque también de 96 bits y acepta varias longitudes de clave fijas, 96, 144 o 192 bits. El bloque de datos se representa como una matriz de 3 x 4 bytes.

Es posible implementar la S-box de Curupira de 8 x 8 bits como una composición de dos S-boxes de 4 x 4 bits, esto reduce los requisitos ROM para almacenar las S-boxes.

El número de rondas no es definitivo, el algoritmo define el número mínimo y máximo de rondas para cada longitud de clave permitida (10 rondas para una clave de 96 bits a 23 rondas para una de 192 bits).

Este cifrado es adecuado para sensores inalámbricos y aplicaciones RFID.

Se ha comprobado que soporta bien los ataques de claves relacionadas porque no tiene claves débiles y los ataques de canal lateral porque utiliza funciones simples como XOR.

9.- HUMMIGBIRD (COLIBRÍ)

Hummingbird cifra los bloques de datos de 16 bits utilizando una clave de 256 bits.

Su arquitectura es original e híbrida ya que combina elementos de cifrado de bloques y secuencias. El procedimiento de cifrado se puede representar como una máquina basada en rotores que funciona continuamente. Cuatro cifrados de bloques internos idénticos desempeñan un papel de rotores virtuales. Realizan un conjunto de operaciones en bloques de datos de 16 bits.

Los requerimientos de recursos relativamente bajos de Hummingbird se pueden lograr debido a operaciones aritméticas y lógicas simples y bloques de datos extremadamente cortos.

En cuanto a su seguridad es vulnerable a varios ataques, la versión mejorada Hummingbird-2 [67] sólo al ataque de Clave relacionada

10. SIMON

SIMON, creado por la Agencia de Seguridad Nacional (NSA) en 2013, está diseñado para optimizar el rendimiento en las implementaciones de hardware aunque también se puede utilizar en las de software. Se introdujo como un cifrado de bloques generalizado que puede utilizarse ampliamente y no es específico de un área concreta.

Su tamaño de bloque y de clave va desde un bloque de 32 bits con una clave de 64 bits, hasta un bloque de 128 bits con una clave de 256 bits.

Fue diseñado para implementarse en un hardware extremadamente pequeño. Es sencillo, flexible y muy ligero, ya que utiliza operaciones a nivel de bit como XOR y AND.

El cifrado de bloques de Simon es un Cifrado Feistel con n bits, por lo que la longitud del bloque es 2n. La longitud de la clave es un múltiplo de n por 2, 3 o 4, que es el valor m. El número de rondas depende del tamaño del bloque y de la clave, por lo tanto, ofrece un mejor rendimiento en hardware que en software.

Una implementación del cifrado Simon se denota como Simon 2n/nm. Por ejemplo, Simon64 / 128 se refiere al cifrado que opera en un bloque de texto plano de 64 bits (n = 32) que usa una clave de 128 bits. El componente de bloque del cifrado es uniforme entre las implementaciones de Simon; sin embargo, la lógica de generación de claves depende de la implementación de 2, 3 o 4 claves. Este cifrado no se ha hecho público.

Seguridad de SIMON:

Es vulnerable a los ataques diferenciales, algebraicos/cubo y al ataque de clave relacionada. (ANEXO IV)

11. **RECTANGLE** [44]

RECTANGLE se implementa utilizando técnicas de corte de bits, para hacerlo más ligero y rápido en sus implementaciones.

Tiene un tamaño de clave de 80 bits y un tamaño de bloque de 64 bits con 25 rondas intermedias hechas de SPN. Cada ronda consiste en una capa S hecha de 16 cajas de sustitución 4x4, una capa P de 3 rotaciones y una clave de ronda añadida de una simple XOR de la matriz de estado y la clave de la ronda.

El diseño de RECTANGLE permite implementaciones ligeras y rápidas utilizando técnicas de corte de bits y proporciona un alto rendimiento.

Las 3 ventajas principales de RECTANGLE:

- Es extremadamente amigable con el hardware. Para la versión de clave de 80 bits, una implementación paralela de un ciclo por ronda sólo necesita 1.600 puertas para un rendimiento de 246 Kbits/seg a un reloj de 100 kHz y una eficiencia energética de 3,0 pJ/bit.
- Logra una velocidad de software muy competitiva entre los cifrados de bloques ligeros existentes gracias a su estilo de corte de bits.
- Debido a una cuidada selección de la caja S y al diseño asimétrico de la capa de permutación, RECTANGLE consigue una muy buena relación seguridad-rendimiento.

Seguridad de RECTANGLE:

Es vulnerable a los ataques de saturación, al de clave relacionada y al de canal lateral y fallos diferenciales. (ANEXO IV)

12. **CLEFIA** [45]

La estructura fundamental de CLEFIA es una estructura Feistel generalizada que consta de 4 líneas de datos, en las que hay dos funciones F de 32 bits por ronda.

Uno de los nuevos enfoques de diseño de CLEFIA es que estas funciones F emplean el Mecanismo de Conmutación de Difusión (DSM): utilizan matrices de difusión diferentes, y se utilizan dos S-boxes diferentes para obtener una mayor inmunidad contra una determinada clase de ataques. En consecuencia, se puede reducir el número de rondas necesarias.

Además, las dos cajas S se basan en estructuras algebraicas diferentes, con lo que se aumenta su seguridad. Otras ideas novedosas son el diseño de programación de claves que es seguro y compacto utilizando también una estructura Feistel generalizada, y es posible compartirla con la parte de procesamiento de datos.

Seguridad de CLEFIA:

Es vulnerable a los ataques de saturación, al de clave relacionada y al de canal lateral y fallos diferenciales al igual que RECTANGLE. (ANEXO IV)

CIFRADOS LIGEROS DE BLOQUE							
ALGORITMO	**Tamaño del bloque**	**Tamaño de la clave**	**Área (GEs)**	**Thoughtput**	**Num. rondas**	**Made of***	**Ataque**
AES	128	128	3100	80	10	SPN	MITM, Related key attack, Ddos, Biclique.
DESL / DESXL	144 / 144	184 / 184	1848/ 2168	High	n	Feistel	---
HIGHT	64	128	3048	188,2	32	GFS	Saturation
PRESENT	64	80/128	1570/ 1882	High	32	SPN	Differential, Integral, DDos, Truncated differential cryptanalysis, Side channel
PRINCE	64	128	3491	>533, 3	12	SPN	Reflection**
KATAN	32/48/64	80	802/9 27/10 54	12,5/ 18,8/ 25,1	254	Flujo	MITM/Biclique
KTANTAN	32/48/64	80	462/5 88/68 8	12,5/ 18,8/ 25,1	254	Flujo	Related key attack
TWINE	64	80/128	1799/ 2285	178	36	GFN	Impossible differential attack on 26th round, Biclique cryptanalysis
CURUPIRA	96	80	---	High	---	Feistel	---
HUMMING BIRD	16	256	2159	High	4	SPN	Varios
SIMON	32 48 64 96 128	64 72/96 96/1289 96/144 128/192/256	739 809 958 955 1234	High	32 36 42/44 52/54 68/69/72	Feistel	Differential fault attacks
RECTANGLE	64	80	1600	High	25	SPN	Slide attack, Related key cryptanalysis, DDos
CLEFIA	128	128/192/ 256	<600 0	268	32	Feistel	Key Recovery, DDos

*Made of: Componente que forma la red

** Reflection: Ataque a un sistema de autenticación que utiliza el mismo protocolo en ambas direcciones

Tabla 2-2 Comparación entre algoritmos ligeros de bloques (Fuente: Elaboración Propia basada en [1, 3, 14, 16 y 22])

2.2.4.2 Ejemplos de cifrados ligeros de flujo

Como se comentó en el apartado de Criptografía Simétrica, los algoritmos de cifrado de flujo emplean una secuencia pseudoaleatoria generada a partir de la clave, y permiten cifrar mensajes combinando el mensaje con alguna función simple reversible (normalmente XOR) con el texto en claro bit a bit.

Los cifrados de flujo ligeros son algoritmos que utilizan una clave de tamaño equivalente a los datos.

Se considera que son potencialmente más compactos, ya que sólo utilizan operaciones de bits. Son más simples y más rápidos en hardware en comparación con los cifrados de bloque. Se construyen utilizando LFSR también NLFSR. Se utilizan en teléfonos móviles, comunicaciones inalámbricas, etc.

1.- TRIVIUM [68]

TRIVIUM fue diseñado en 2005 con las limitaciones de velocidad, seguridad y flexibilidad, y por lo tanto, es muy adecuado para los dispositivos con limitaciones de hardware. Este cifrado utiliza tres registros de desplazamiento o LFSR[6]. Utiliza un vector de inicialización de 80 bits y una clave secreta de 80 bits y tiene un tamaño de estado interno de 288 bits. Necesita unos 2600 GE para su implementación. No es seguro frente al Cube Attack (ataque al cubo).

2.- GRAIN [46]

GRAIN o GRAIN 128 es uno de los cifrados ligeros más aceptados y adecuados para dispositivos con restricciones. Admite una clave de 128 bits, un vector de inicialización de 96 bits y una etapa intermedia de 256 bits. Utiliza LFSRs, NLFSRs y una función booleana que puede implementarse con unos 1300 GE. Tiene la desventaja de un alto retardo de propagación y sus versiones anteriores eran propensas a varios ataques como el de criptoanálisis.

3.- CHACHA [69]

CHACHA, es una familia de algoritmos de cifrado simétricos, que soportan claves de 128 y 256 bits y de alta velocidad creada por Bernstein en 2008. Tiene características similares a Salsa pero con una función primitiva de 8, 12 o 20 rondas mejoradas que aumentan su resistencia al criptoanálisis y mejoran el tiempo por ronda. Así tenemos ChaCha8, ChaCha12 y ChaCha20.

Su código fue publicado y estandarizado y en implementaciones de software, es más eficiente y rápido que AES.

La clave es de 256 bits, el vector de inicialización de 128 bits. Se compone de dos registros de desplazamiento de retroalimentación diferentes (FSR) utilizando operaciones invertibles simples como la suma modular, la rotación de bits y el XOR en cada ronda para actualizar los registros cargados con el flujo de entrada.

[6] **LSFR:** Registro de desplazamiento en el cual la entrada es un bit proveniente de aplicar una función de transformación lineal a un estado anterior para su construcción.

Por lo tanto, el hardware para el cifrado y el descifrado es casi idéntico y resulta muy adecuado para los dispositivos con limitaciones de software.

4.- WG 8 [70]

El WG-8 pertenece a la familia de cifrado de flujo de Welch Gong. Es un LFSR de 20 etapas con una clave de 80 bits y un vector de inicialización. Tiene dos fases de funcionamiento, la de inicialización y la de ejecución. El cifrado consiste en un LFSR con un polinomio de retroalimentación seguido de transformaciones Welch-Gong que generan secuencias de bits con propiedades de aleatoriedad demostrables. Ofrece un mayor rendimiento que otros y necesita menos memoria. Tiene una buena aleatoriedad y es resistente a los ataques. Ofrece un buen rendimiento y tiene un bajo consumo de energía. Sin embargo, no es seguro frente a los ataques de recuperación de claves.

5.- ESPRESSO [71]

ESPRESSO resulta ser el más rápido entre los cifrados ligeros por debajo de 1500 GE. Tiene la ventaja combinada de la configuración Galois y la configuración Fibonacci (métodos para la generación de números aleatorios) del NLFSR. Utiliza una clave de 128 bits y un vector de inicialización de 96 bits que se combinan en la fase de inicialización. Se compone de un NLFSR de 256 bits y una función no lineal de 29 variables. Tiene un retardo de propagación corto y puede ser analizado formalmente. Para su diseño, se ha tenido en cuenta tanto el tamaño del hardware como la optimización de la velocidad. Por lo tanto, ha minimizado la huella de hardware y maximizado el rendimiento. Está especialmente diseñado para aplicaciones 5G con mejor calidad de servicio, ya que soporta una baja latencia de unos pocos milisegundos.

6.- Decim-v2. [47]

Es un algoritmo orientado a Hardware creado en 2005 por Berbain (et al.), 2005. El gobierno, empresas y centros de estudios franceses patrocinaron en la creación de este algoritmo.

Decim-v2 es la versión mejorada a la que fue presentada en el concurso (Decim-v1).

Además existe una versión presentada en 2007, llamada Decim-128 trabaja con una clave y un vector de inicialización de 128 bits cada uno.

Decim-v2 utiliza una clave de 80 bits y un Vector de inicialización de 64 bits. Un estado interno que consiste en un registro de desplazamiento de retroalimentación lineal (LFSR) de 192 bits. Además, Decim-v2 tiene una función de filtro no lineal, un mecanismo Decim irregular (llamado ABSG) y un buffer.

Un texto cifrado/texto plano se obtiene al hacer XOR a un texto plano/texto cifrado, al flujo de claves.

7.- KCipher-2. [48]

También conocido por el nombre K2, es un algoritmo creado por Shinshaku Kiyomoto, Toshiaki Tanaka y Kouichi Sakurai en 2007.

KCIPHER-2 es un algoritmo orientado a Software y muy veloz dada su sencillez. Puede ser implementado en Hardware por sus propiedades de paralelización. Emplea 128 bits de clave y 128 bits de vector de inicialización y hasta el momento no se conocen vulnerabilidades frente a ataques.

Puede cifrar y descifrar entre siete y diez veces más rápido que el AES.

La aplicación del algoritmo KCipher-2 permite niveles más altos de seguridad para una variedad de servicios, incluidos los servicios de contenido multimedia y los servicios de comunicación de banda ancha.

Características:

- El más rápido y ligero del mundo en un teléfono móvil.
- Margen de seguridad excepcionalmente alto.
- Tiene licencia de venta.

8.- MUGI. [49]

Es un algoritmo creado por D. Watanabe, S. Furuya, H. Yoshida, K. Takaragi para la empresa japonesa HITACHI en el año 2001. Es una variante del algoritmo Panamá.

Tiene una clave secreta y un vector de inicialización de 128 bits cada uno. Originalmente orientado a Hardware, también mostró muy buen rendimiento en Software.

Una implementación de velocidad optimizada en hardware alcanza unos 3 Gb/s con 26 Kgates, lo que es varias veces más rápido que AES. Por otro lado, es resistente al ataque de resincronización y al de clave relacionada (*related-key*).

9-Rabbit [50]

Este algoritmo fue creado por Martin Boesgaard, Mette Vesterager, Thomas Christensen y Erik Zenner pertenecientes a la empresa dinamarquesa CRYPTICO A/S.

En su diseño se pueden observar ciertas características de optimización orientado a software.

El algoritmo trabaja con una clave de 128 bits de longitud y un vector de inicialización de 64 bits.

Hasta ahora no se ha revelado ninguna debilidad criptográfica.

Rabbit se diseñó para ser más rápido que los cifrados habituales y para justificar un tamaño de clave de 128 bits para cifrar hasta 2^{64} bloques de texto plano.

Esto significa a un atacante que no conozca la clave, no le debería ser posible distinguir 2^{64} bloques de salida de cifrado.

10.- Snow 2.0. [51]

Se conoce como Snow o Snow 1.0 a la primera versión de este algoritmo. Fue creado por Thomas Johansson y Patrik Ekdahl en la Universidad Lund, Suecia.

El algoritmo trabaja con palabras de 32 bits y permite usar claves de 128 y 256 bits. Fue descubierta una vulnerabilidad y se retiró. Una vez resuelta, sus autores presentaron la versión mejorada bajo el nombre SNOW 2.0.

11.- ENOCORO. [52]

Es una familia de Generadores de Números Pseudoaleatorios, creado en 2007 para la empresa japonesa HITACHI por Dai Watanabe, T. Kaneko (es una variante del algoritmo Panamá).

ENOCORO tiene dos variedades: una de 80 bits y otra de 128 bits conocidas como ENOCORO-80 y ENOCORO-128v2

Tiene buena resistencia contra todo tipo de ataques criptoanalíticos. Además, el coste de implementación es bastante bajo.

CIFRADOS LIGEROS DE FLUJO					
ALGORITMO	**Vector inicialización**	**Tamaño de la clave**	**Área (GEs)**	**Throughput**	**Made of***
TRIVIUM	80	80	2600	High	LFSR
GRAIM	96	128	1300	Low	Both
CHACHA	128	256	750	High	Both
WG 8	80	80	1984	High	LFSR
ESPRESSO	96	128	1500	High	NLFSR**
Decim-v2	64	80	2500	High	LFSR
KCipher-2	128	128	---	High	LFSR
MUGI	128	128	---		Both
Rabbit	64	128	4000	High	---
Snow 2.0	128	128/256	---	Low	LFSR
ENOCORO	64	80/128	2,7k / 4,1k	---	LSFR

*Made of: Componente que forma la red ** NLFSR Non Linear FeedBack Shift Register

Tabla 2-3 Comparación entre algoritmos ligeros de flujo (Fuente: Elaboración propia basada en [3])

La norma ISO/IEC 18033-4:2011 presenta 5 algoritmos de Cifrado de Flujo y la 29192 dos.

ISO/IEC 18033	**ISO/IEC 29192**
Decim-v2	*Enocoro*
KCipher-2 (K2).	
MUGI.	
Rabbit	*Trivium*
SNOW 2.0.	

Tabla 2-4 Algoritmos contenidos en las normas ISO/IEC (tomada de [13])

Las normas ISO/IEC 18033e ISO/IEC 29192 sirven para estandarizar los algoritmos de cifrado de flujo.

Como se ha visto, la mayoría de los cifrados ligeros existentes son específicos de la aplicación y, por lo tanto, existe una gran variedad de ellos. La incorporación de algoritmos sofisticados aumenta la seguridad, pero incrementa su tamaño y limita su propósito. Se generan muchos de estos algoritmos intentando conseguir una mejora en un dispositivo en particular o bien una mejora en el propio algoritmo. Aquí sólo se han enumerado algunos, pero hay muchos más.

2.2.4.3 Criptográfía ligera asimétrica (o de clave pública)

Los enfoques criptográficos asimétricos convencionales no son eficientes porque ninguna de las implementaciones propuestas tienen un tiempo de ejecución razonable.

Los algoritmos de criptografía ligera asimétrica son muy recomendables para los dispositivos con limitaciones de recursos porque son mucho más exigentes desde el punto de vista computacional que sus homólogos simétricos. Existen algoritmos asimétricos convencionales como Rabin/RSA [72] que se basan en el problema de la factorización de enteros, y otros como ECC/HECC (Hyper elliptic curve cryptography), que se basan en el problema del logaritmo discreto de la curva elíptica.

RSA es un algoritmo que requiere muchos recursos y se puede implementar en una optimización ligera. RSA como hemos visto, depende de la selección de dos grandes números primos para encontrar sus claves pública y privada, que suelen estar entre 1024 y 4096 bits de longitud.

Rabin es algo similar a RSA pero más rápido, y se diferencia de RSA en la complejidad de los problemas de factorización de los que depende y en el descifrado, que es menos eficiente.

BluJay es un esquema híbrido basado en Rabin que es adecuado para plataformas ligeras y está basado Hummingbird-2. El cifrado mediante BluJay es significativamente más rápido y ligero que RSA y ECC para el mismo nivel de seguridad.

1.- BLUJAY [54]

Es un criptosistema adecuado para plataformas ultraligeras (un total de 2000-3000 GE) como los microsensores y las etiquetas de autenticación RFID. BlueJay se basa en el criptosistema Rabin y se ha demostrado que romperlo es tan difícil como factorizar. El hardware de cifrado asimétrico del criptosistema híbrido requiere menos de 1000 GE.

BlueJay está pensado para recibir datos de sensores, autenticación en RFID y aplicaciones en las que sólo se requiere la operación de clave pública.

2.- CRIPTOGRAFÍA LIGERA DE CURVA ELÍPTICA LIGERA

La ECC ligera se considera el método de clave pública más eficiente porque requiere menos consumo de energía, un área más pequeña y menos ciclos de reloj.

Se basa en las matemáticas de las curvas elípticas[7] y utiliza claves pequeñas que se generan mediante un algoritmo discreto. Hay varias implementaciones optimizadas para el ECC ligero pero la desventaja es que la mayoría de los diseños requieren más de 10.000 GE.

2.2.4.4. Funciones Hash Ligeras

Se basan en la creación de un "hash" o función resumen con una clave privada que puede ser verificada con una clave pública. Es decir una función que toma datos arbitrarios como entrada y proporciona una salida única e irrepetible. Se utiliza en numerosos campos, como las firmas digitales y las sumas de comprobación.

PHOTON, Spongent y Lesamanta LW son considerados algoritmos estándar para la criptografía ligera, según la ISO/IEC 29192-5:2016 [73].

[7] En matemáticas, las curvas elípticas se definen mediante ecuaciones cúbicas (de tercer grado).

Las funciones hash convencionales como MD5 y SHA1 y otras funciones hash no son convenientes para los dispositivos IoT ya que apenas tienen memoria, ni CPU, ni energía como ya se ha comentado. El NIST por esta razón, recomienda funciones hash ligeras como SPONGENT, PHOTON, Quark y Lesamnta-LW. Estos métodos consumen menos memoria al tener una entrada de sólo 256 caracteres (mientras que las funciones de hash convencionales tienen hasta 264 bits).

Se espera que las funciones hash sean fuertes contra los ataques de colisión, preimagen y segunda preimagen. Hay muchas estructuras de diseño de funciones hash, sin embargo, las funciones hash ligeras se basan en Merkle-Damgárd y Sponge [60].

Un ataque de preimagen (Preimage attack) consiste en encontrar un valor arbitrario cuyo hash colisione con otro hash. Hay dos tipos de resistencia a la preimagen:

- resistencia a la preimagen (débil): para todas las salidas preespecificadas, es computacionalmente inviable encontrar cualquier entrada que tenga un valor hash en esa salida; es decir, dado **y**, es difícil encontrar una **x** tal que **h(x)=y**.
- resistencia de segunda preimagen (fuerte): para una entrada especificada, es computacionalmente inviable encontrar otra entrada que produzca la misma salida; es decir, dado **x**, es difícil encontrar una segunda preimagen $\mathbf{x'} \neq x$ tal que $\mathbf{h(x)=h(x')}$.

La resistencia a la colisión consiste en que no es factible computacionalmente encontrar dos entradas distintas $\mathbf{x, x'}$ que tengan la misma salida; es decir, tal que $\mathbf{h(x)=h(x')}$.

La resistencia a colisiones implica resistencia a la segunda preimagen, pero no garantiza la resistencia a la preimagen. Por el contrario, un ataque de segunda preimagen implica un ataque de colisión.

1.- PHOTON [74]

Es una familia de funciones hash ligeras orientadas al hardware, aunque también es posible implementarlas en software, diseñadas para utilizar la menor memoria interna posible.

Proporciona un equilibrio entre la seguridad del hardware y la velocidad del hash de los mensajes más pequeños que puede ser muy lentos.

No es muy eficaz para grandes cantidades de datos porque la tasa de bits es muy pequeña para minimizar la memoria utilizada.

Está basada en el protocolo AES y es adecuada para dispositivos extremadamente restringidos así que se suele utilizar en dispositivos RFID. Utiliza una construcción de esponja con diferentes tamaños de permutación y hashes de salida: 100. 144. 196. 256 y 288 bits y 80. 128. 160. 224. y 256 bits, respectivamente.

Ataques a PHOTON:

Los ataques a PHOTON son variados, pero ninguno de ellos ha conseguido romperlo. La seguridad de PHOTON se basa en la función esponja que refuerza su seguridad para tasas de bits bajas, porque hay muchas rondas en las que el texto en claro se absorbe por la función esponja o el hash es comprimido por la función esponja. En cada ronda se aplica la permutación y por lo que se añade mucha complejidad.

Resistencia frente al Criptoanálisis diferencial/lineal Como ya se ha mencionado, las primitivas internas de PHOTON están basadas en AES y, por tanto, se podría pensar en romper PHOTON al igual que AES pero PHOTON no requiere ningún material clave en las permutaciones internas y por lo tanto no es posible aprovechar la clave atacando estas permutaciones.

Ataques de rebote y super-sbox Las SBox utilizadas por PHOTON son bastante similares a las permutaciones de PRESENT que no dependen de un determinado secreto. Actualmente, todas las

versiones de PHOTON son bastante seguras contra este tipo de ataque. El atacante sólo puede calcular los resultados de 8 rondas, no se pueden alcanzar más rondas incluso con más potencia de cálculo. El hecho de que en PHOTON no intervenga ninguna clave en las permutaciones internas, impide que se pueda mejorar el ataque utilizando una cierta debilidad de la clave secreta.

Ataque Cubo y ataques algebraicos. Los ataques de testeo de cubos si han funcionado con otros algoritmos hash como Trivium y MD6.

Se hizo una prueba con PHOTON y sólo se recuperaron 3 rondas del algoritmo. En PHOTON se utilizan dos tipos de cajas S, las de AES y las de PRESENT. En el caso de la permutación P_{144}, que se utiliza en PHOTON128/16/16, es necesario resolver unas 9000 ecuaciones en unas 3500 variables.

En comparación con AES, en el que hay que resolver 6.400 ecuaciones en unas 2.500 variables, se supone que PHOTON es bastante seguro contra este tipo de ataques.

Otros criptoanálisis. Recientemente se ha utilizado el ataque de deslizamiento para atacar funciones hash de tipo esponja. Este ataque intenta explotar el grado de autosimilitud de una permutación. En PHOTON todas las permutaciones se hacen diferentes y es imposible realizar el ataque de deslizamiento.

2.- SPONGENT [41]

SPONGENT se basa en la construcción de funciones esponja que realizan una permutación de tipo PRESENT, lo que proporciona robustez.

Presenta cinco tamaños de hash diferentes con tamaños de permutación de 88, 128, 160, 224 y 256 bits con rondas diferentes que calcula códigos hash de 45, 70, 90, 120 y 140 bits respectivamente. Su principal problema de diseño es la serialización para un diseño compacto. SPONGENT se centra en la huella más pequeña. Por lo tanto, el algoritmo de SPONGENT utiliza una función de ronda simple con un S-Box de 4 a 4 bits para minimizar la huella.

Ataques a SPONGENT:

Resistencia contra el criptoanálisis diferencial/lineal: El atacante sólo puede calcular los resultados de 5 rondas, lo que está bastante bien, con lo cual es resistente a este tipo de ataques.

Ataques de colisión: No es posible atacar el número completo de rondas con lo cual se considera resistente.

El ataque de rebote: Resiste el ataque.

3.- LESAMNTA-LW

Es una función hash ligera con un tamaño de permutación de 384 bits que calcula un código hash de 256 bits de longitud.

Proporciona un buen rendimiento en dispositivos con poca memoria que emplean CPUs de 8 bits. Está basada en la construcción Merkle-Damgárd reforzada y tiene un cifrado por bloques basado en algoritmo de cifrado AES con una clave de 128 bits.

Es compacta y rápida, optimizada para aplicaciones ligeras en una gran variedad de entornos, desde dispositivos baratos hasta servidores de gama alta.

Tiene un nivel de seguridad alto. Una función hash ideal de 256 bits proporcionaría un nivel de seguridad de 2^{256} contra ataques de preimagen y LESAMNTA-LW alcanza el nivel de seguridad 2^{120}, que es bastante alto para dispositivos IoT.

Ataques a LESAMNTA-LW:

Según los estudios realizados [42], el algoritmo resiste los ataques diferencial/lineales, los ataques con claves relacionadas y los ataques de colisión.

2.3 Ataques Criptográficos

Los algoritmos criptográficos tienden perder efectividad con el tiempo debido al avance de la de la velocidad y potencia de los equipos de computación y todos son vulnerables a los ataques de fuerza bruta. Los ataques buscan encontrar la clave utilizada en el proceso de cifrado y descifrado. Cada ataque tiene un método para descubrirla. Vamos a ver los principales.

2.3.1. Ataque de búsqueda exhaustiva (Exhaustive key search)

El que denominamos normalmente como ataque de fuerza bruta. Consiste en buscar la clave probando todas las combinaciones posibles hasta encontrar aquella que permite el acceso.

Existen diferentes tipos de ataque de fuerza bruta, como el “credential stuffing”, el ataque de diccionario, el ataque de fuerza bruta inverso o el ataque de “password spraying”. Generalmente, los ataques de fuerza bruta tienen mayor éxito en los casos en los que se utilizan contraseñas débiles o relativamente fáciles de predecir.

El “credential stuffing” o relleno de credenciales, el atacante utiliza combinaciones de nombres de usuario/contraseña que tiene en su poder y que se filtraron en algún momento, sobre todo teniendo en cuenta que muchos usuarios suelen reutilizar las contraseñas en más de una cuenta.

El ataque diccionario prueba todas las palabras posibles que el atacante almacena en un diccionario. Estas palabras pueden ser de todo tipo y pueden incluir nombres, lugares y otras combinaciones, muchas de ellas obtenidas por filtraciones previas y que los usuarios empleaban porque eran fáciles de recordar.

En el ataque de fuerza bruta inverso se comienza con una contraseña conocida (como las contraseñas filtradas disponibles en línea) y busca millones de usuarios hasta que se encuentra una coincidencia.

El “password spraying” consiste en probar la misma contraseña en muchas cuentas. Es efectivo en las cuentas donde se utilizan contraseñas simples y predecibles, como "123456".

2.3.2. Criptoanálisis diferencial/lineal (Differential-linear cryptanalysis)

El Criptoanálisis diferencial trata de encontrar alguna relación entre el texto claro y el texto cifrado. Se parte informaciones que se envían en texto en claro para ver si se puede encontrar relaciones en el cifrado.

El Criptoanálisis lineal trata de encontrar correlaciones entre la clave, el texto claro y el texto cifrado obtenido en la salida del sistema criptográfico basado en un cifrado en bloque.

2.3.3. Criptoanálisis integral/square/saturation

Integral: Suele aplicarse a los cifrados de bloque basados en redes de sustitución-permutación. Fue un ataque dedicado contra Square, por lo que se conoce comúnmente como el ataque Square. Selecciona conjuntos de textos sin formato, de los cuales una parte se mantiene constante y otra parte varía con todas las posibilidades posibles.

2.3.4. Ataque algebraico (Algebric attack)/Ataque cubo (Cube attack)

Los ataques algebraicos intentan deducir la clave secreta mediante la resolución de ecuaciones no lineales en las que se incluyen el mensaje, el texto cifrado y los bits de la clave. Esta idea se remonta al trabajo fundamental de Shannon[8]. Un ataque algebraico típico tiene dos fases:

- El atacante establece un número suficiente de ecuaciones no lineales de bajo nivel o estructurales
- Posteriormente se recuperan los bits clave resolviendo las ecuaciones.

El primer paso sólo tiene que hacerse una vez para un cifrado. Los métodos de resolución de ecuaciones más utilizados son la linealización y los algoritmos de base de Gröbner.

La linealización resuelve el sistema resultante de ecuaciones no lineales sustituyendo los términos no lineales por nuevas variables. Entonces, el nuevo sistema resultante es puramente lineal y puede resolverse.

En general, es difícil encontrar ecuaciones de bajo nivel a partir de un algoritmo de cifrado bien diseñado. Teniendo en cuenta la dificultad de resolver ecuaciones no lineales, la mayoría de los algoritmos son naturalmente inmunes a los ataques algebraicos triviales.

Ataque Cubo: Explotan las ecuaciones implícitas de bajo nivel en los algoritmos criptográficos.

A grandes rasgos, una función criptográfica es vulnerable a los ataques de cubo si un bit de salida puede representarse como un polinomio de grado suficientemente bajo.

Funcionan sumando un valor de bit de salida para todos los valores posibles de un subconjunto de bits de entrada públicos (por ejemplo, texto plano elegido o bits IV), elegidos de manera que la suma resultante sea una combinación lineal de bits secretos. La aplicación repetida de esta técnica da un conjunto de relaciones lineales entre bits secretos que se pueden resolver para descubrir estos bits.

2.3.5. Ataque Meet-in-the-middle (MITM)/Biclique

MITM: El atacante se sitúa entre las dos partes que intentan comunicarse e intercepta los mensajes enviados para conseguir la clave.

Biclique: Un ataque biclique es una variante del MITM en el que se mejora el cifrado de bloques para encontrar la clave secreta.

2.3.6. Ataque de clave relacionada (Related key attack)

Consiste en que el atacante puede observar el funcionamiento de un cifrado bajo varias claves diferentes cuyos valores inicialmente se desconocen, pero donde el atacante conoce alguna relación matemática vinculada con las claves. (El protocolo WEB se rompió con un ataque de clave relacionada)

[8] **Shannon:** La probabilidad de que un mensaje, dentro de un conjunto de mensajes posibles, sea recibido, es un valor matemático definido y medible.

2.3.7. Ataque de canal lateral (Side Channel)/Ataque de fallos diferenciales (Differential fault attack)

Un ataque de canal lateral tiene como objetivo recopilar información o influir en la ejecución del sistema buscando vulnerabilidades que sirvan para conseguir el material que permita romper el sistema.

El análisis de fallos consiste en producir un error computacional que modifique el resultado en la salida. Al perturbar activamente la actividad de un dispositivo, puede producir un comportamiento defectuoso dependiente de los datos.

La información filtrada por un cálculo defectuoso puede explotarse utilizando varios métodos, basados en enfoques analíticos y estadísticos.

- Ataques de análisis de fallos diferenciales
- Ataques de análisis de fallos estadísticos
- Ataques de análisis de fallos algebraicos

2.3.8. Ataque de correlación (Correlation Attack)

Los ataques de correlación son una clase de ataques para romper cifrados de flujo cuya secuencia de claves se genera al combinar la salida de varios registros de desplazamiento de retroalimentación lineal (LFSR) utilizando una función booleana.

Los ataques de correlación explotan una debilidad estadística que surge de una mala elección de la función booleana.

Son posibles cuando existe una relación lineal entre la salida de un LFSR individual en el generador de flujo de claves y la salida de la función booleana que combina el estado de salida de todos los LFSR.

Con algo de conocimiento previo de las claves (podemos sacarla del texto sin formato, ya que los dos textos simplemente se combinan mediante XOR), o aplicando la fuerza bruta sobre con los registros relacionados, permitiría al atacante forzar la clave para ese LFSR individual y el resto del sistema por atacarlo por separado.

2.3.9. Ataque distintivo (Distinguishing Attack)

Un ataque distintivo es un algoritmo de prueba que ofrece un comportamiento no aleatorio en un criptosistema. Esto puede proporcionar cierta información al atacante.

En el caso de los cifrados por bloques, consiste en ser capaz de identificar alguna característica distintiva de la salida y podría conducir a un ataque que revele información sobre la clave del cifrado. Por ejemplo, el Criptoanálisis Diferencial de DES identifica una característica distintiva de las primeras rondas de DES, y luego utiliza esa característica para verificar las estimaciones sobre los bits de la clave.

En el caso de un cifrado de flujo, en el Ataque a RC4, se comprobó que el segundo byte de salida estaba fuertemente sesgado hacia cero.

2.3.10. Ataque Chosen-IV

Los cifrados de flujo combinan una clave secreta con un vector de inicialización (IV) para producir una secuencia pseudoaleatoria que de vez en cuando se resincroniza.

Un ataque de Chosen-IV se basa en encontrar determinados IV que, tomados en conjunto, revelen información sobre la clave secreta. Se suelen elegir varios pares de IV y las diferencias en los flujos de claves generados se analizan estadísticamente en busca de una correlación lineal y/o una relación booleana algebraica.

Si la elección de determinados valores del vector de inicialización expone un patrón no aleatorio en la secuencia generada, este ataque computa algunos bits y, por tanto, acorta la longitud efectiva de la clave.

2.3.11. Ataque de deslizamiento (Slide Attack)

El ataque de deslizamiento hace frente a la evidencia empírica de que después de un número elevado de rondas, incluso un cifrado relativamente débil se vuelve muy fuerte.

Se apoya en el ejemplo de DES, donde romper 16 rondas ya es una tarea muy difícil, por no hablar de las 32-48 rondas (por ejemplo, el doble o el triple DES). Por tanto, para el atacante resulta más fácil buscar nuevas técnicas que sean independientes del número de rondas de un cifrado. Este ataque funciona de tal manera que hace que el número de rondas de un cifrado sea irrelevante.

Este ataque explota desde la programación de claves a la explotación de las propiedades generales de un cifrado, analizando sus puntos débiles para romperlo.

El ataque más común es que las claves se repitan de forma cíclica.

2.3.12. Ataque de compensación de tiempo y memoria (Time-Memory Trade-off Attack)

La idea general de este ataque es reducir el tiempo que se realiza utilizando un ataque de fuerza bruta. Consiste en realizar un esfuerzo de precomputación en la tabla de búsqueda una sola vez para reducirlo.

Este tipo de ataque es muy difícil, por lo que la mayoría de los sistemas de cifrado y cifrado en uso no fueron diseñados para resistirlo.

2.3.13. Ataque de suposición (Guess and Determine Attack)

Está basado en el ataque de compensación de tiempo y memoria.

Un ataque de suposición intenta obtener los estados de todas las celdas del sistema de cifrado completo adivinando (de ahí su nombre) inicialmente el contenido de algunas y comparando la secuencia de claves resultante con la secuencia de claves en curso. El cifrado se "rompe" cuando se ha determinado un estado interno completo a partir de los valores adivinados.

Los ataques de suposición son uno de los ataques generales que han sido eficaces contra algunos cifrados de flujo.

2.3.14. Ataques de rebote (rebound attack) y super-sbox

El Rebound Attack es un tipo de ataque estadístico a las funciones hash, utilizando técnicas como el criptoanálisis rotacional y diferencial para encontrar colisiones.

2.4 Criptografía ligera vs Ultraligera

Tras la presentación de todos los algoritmos ligeros, se ha comprobado que es necesario dividir la criptografía ligera en dos partes.

En [27] se recoge esta una clasificación interesante que analiza las características de los algoritmos para encuadrarlos en una categoría u otra. Esta clasificación queda recogida en la siguiente tabla:

	Criptografía Ultraligera	**Criptografía Ligera**
Tamaño de bloque	64 bits	≥ 128 bits
Nivel de seguridad	≥80 bits	≥ 128 bits
GEs	<2000	>2000
Precio Circuito	Relativamente bajo	Precio más elevado
Ataques relevantes	baja complejidad de datos/tiempo	igual que la criptografía normal
Plataforma prevista	circuito dedicado (ASIC, RFID...)	microcontroladores, cpu´s de gama baja
Resiliencia[9]	importante	importante
Funcionalidad	una por dispositivo, por ejemplo, autentificación	cifrado, autentificación, hashing...
Conexión temporal	sólo a un centro determinado	permanente, a una red global

Tabla 2-5 Diferencias entre la criptografía ligera y ultraligera (Fabricación propia basada en [27])

[9] **Resilencia:** Capacidad de los datos para estar disponibles para usuarios o aplicaciones

3 DESARROLLO DEL TFM

Como se comentado, los dispositivos IoT presentan una serie de limitaciones de recursos, pero a su vez es necesario protegerlos para que los consumidores puedan confiar en ellos, ya que contienen información útil para los ciberdelincuentes. En este desarrollo se tratarán tres casos en los que se intentará demostrar que la criptografía ligera es útil para garantizar la información transmitida por los dispositivos IoT.

3.1 RFID

RFID es un sistema de identificación parecido al código de barras pero que en vez de utilizar la imagen para identificar una etiqueta colocada en un producto, utiliza las ondas de radio para comunicarse con un microchip, normalmente a distancias cortas y sin contacto, que puede estar montado sobre gran cantidad de soportes, como por ejemplo un tag o etiqueta RFID, una tarjeta o un transpondedor. De esta forma, la etiqueta RFID puede aplicarse o incorporarse a un producto, animal o persona, con el fin de identificarlo.

Las etiquetas RFID pueden ser activas o pasivas. Las pasivas no requieren alimentación eléctrica interna, ya que se alimentan de la propia señal, que mediante la antena, activa el circuito. Las etiquetas activas disponen de una fuente de alimentación propia (una pequeña batería), o van incluidas en un dispositivo, por ejemplo un teléfono móvil.

Las etiquetas RFID activas alcanzan rangos de lectura mucho mayores que las etiquetas pasivas. Pueden ser leídas a distancias superiores a los 100 metros y su tamaño es mayor. Pueden incorporar una pequeña memoria adicional para almacenar información (normalmente 96 bits de memoria, pero pueden oscilar entre 2 y 1000 bits). No obstante su gran inconveniente es que su funcionamiento está ligado a su fuente de alimentación, por lo que su tiempo de vida se limita al de la batería, aunque en algunos casos pueden llegar hasta 10 años.

Las etiquetas pasivas son mucho más baratas de fabricar y por eso existen muchas más en el mercado. Sin embargo, a pesar de la ventaja del coste y del tamaño, hay diferencias notables en características como rango de lectura (6 o 7 metros máximo), imposibilidad de añadir funciones adicionales y baja capacidad de almacenamiento de datos, que hacen que en determinados casos se deban utilizar etiquetas activas.

El sistema funciona de forma que el lector lanza una señal RF (Radio Frecuencia), la etiqueta RFID se activa y emite una respuesta. El código de identificación es único y la comunicación por

radiofrecuencia necesita la integración de una antena RF, cuyas características dependen de la banda de frecuencia en la que opere el sistema. Por lo tanto se necesita un transmisor y un receptor.

Las comunicaciones no protegidas entre las etiquetas y los lectores a través de un canal inalámbrico pueden revelar información sobre las etiquetas, por lo que se necesita un mecanismo de autenticación para evitar que los dispositivos queden expuestos y los atacantes puedan entrar en la red que forman las etiquetas.

La arquitectura del sistema RFID integrado en una red IoT consta de tres componentes principales: la etiqueta del Código Electrónico de producto (EPC), el lector y la base de datos.

- La etiqueta es un chip electrónico de bajo coste con el número de identificación único (ID).
- El lector identifica cada etiqueta asociada al sistema mediante la recepción del ID a través del canal inalámbrico.
- La base de datos apoya al lector en un proceso de identificación almacenando los atributos de todas las etiquetas asociadas al sistema RFID.

Figura 3-1 Partes de un sistema RFID (Fuente: Elaboración propia)

Las etiquetas RFID han abierto la puerta a múltiples posibilidades y cada día surgen nuevas aplicaciones que las contemplan.

En general, podemos emplear esta tecnología sobre cualquier aplicación de identificación. A continuación se exponen algunos ejemplos:

1) Sector logístico, la tecnología RFID permite conectar todas las fases de la cadena de abastecimiento, desde la producción al inventario y la distribución de los productos. Algunas de sus ventajas son:

- Registro automático de entrada y salida de vehículos, asignación automatizada de ubicaciones o plazas de aparcamiento, identificación en tiempo real y automatizada de la carga de los camiones y generación en tiempo real de documentación de recogida o de entrega.
- Precisión de tiempos de llegada del transporte urbano combinándola con la tecnología GPS y GPRS
- Control de accesos. Por ejemplo cuanto tiempo lleva un producto en almacenes.

2) Cadena de suministro, RFID permite identificar familias de productos de forma única. Se pueden conocer las existencias reales de los productos, las devoluciones que se realizan, identificar clientes por promociones o descuentos, etc. Es particularmente útil en la trazabilidad necesaria en áreas como la seguridad alimentaria y control de medicamentos.

3) Producción, la principal aplicación de RFID en el entorno de producción se encuentra en el etiquetado de productos en el punto de origen. De esta forma, RFID permite una estricta supervisión de

cada una de las etapas de producción, respetando, a su vez, la trazabilidad de cada producto, y facilitando la gestión de cualquier sistema de control de calidad.

4) La Industria Automovilística, usa los sistemas RFID para hacer el seguimiento de los vehículos a lo largo de la cadena de montaje. Otro ejemplo es que los fabricantes de automóviles puedan identificar sus neumáticos en cada coche, para así poder cambiar los neumáticos más eficazmente.

5) La Trazabilidad de equipaje en aeropuertos, permite que las compañías aéreas puedan localizar con exactitud el equipaje de los pasajeros en tiempo real.

6) Peaje automatizado, cuando un vehículo pasa por un punto de control genera un registro que, al ser procesado, generará a su vez un cobro automático para el conductor.

7) Tarjetas identificadoras o inteligentes, para acceder a determinados lugares, realizar transferencias bancarias a través de un TPV (Terminal Punto de Venta) o pagar el transporte público.

8) Gestión de bibliotecas, RFID optimiza el proceso de inventariado además de conocer existencias, prestamos, etc.

9) Médico-sanitarias, para la identificación de pacientes y la gestión del material hospitalario.

10) Supermercados. No hace falta descargar la compra y el ticket se automatizaría.

11) Monitorización de ganado, permiten identificar al animal y garantizar su trazabilidad. Además, también permiten motorizar datos vitales como, por ejemplo, la temperatura del animal que pueden ser interesantes para evitar enfermedades.

Campos IoT	**Definición**
Smart Home Mobile IoT basado en RFID móvil	Se trata de un sistema de servicio doméstico inteligente para beneficiar al usuario en términos de coste, consumo de energía y facilidad
RFID e IoT para el control de la asistencia	Se trata de un sistema de control de asistencia en tiempo real al que pueden acceder varias partes, es decir, profesores, estudiantes y padres.
Sistema híbrido de Harvard	El sistema utiliza las etiquetas RFID para el seguimiento de equipos, camas pacientes y bebés de la UCI.
Sistema de identificación positiva de pacientes	El sistema facilita la identificación del paciente y acelera acceso a los datos del paciente.
Sistema de transfusión Intel	Este sistema identifica las bolsas de sangre, los receptores y el personal. El objetivo de este sistema es mejorar la seguridad de la transfusión de sangre.

Tabla 3-1 Aplicaciones IoT con RFID (tomada de [15])

El incremento de esta tecnología de identificación por radio frecuencia en muchas industrias, hace que los hackers hayan encontrado vulnerabilidades y realicen ataques contra estos sistemas.

3.1.1 Objetivos de los ataques en dispositivos RFID

Los objetivos de cada ataque pueden ser muy diferentes. Muchos sistemas de información se centran únicamente en la protección de los datos transmitidos por los dispositivos, sin embargo, a la hora de diseñar los sistemas RFID, hay que tener en cuenta otros factores como que pueden ser rastreados o que estos datos que se transmiten puedan ser manipulados.

Como ya se ha comentado, su uso es variado y se pueden utilizar para el etiquetado de prendas de ropa y calzado, tarjetas de transporte, pasaportes, etc. o en otros ámbitos como el sanitario, para el seguimiento de pacientes o el control de medicamentos y muestras. Si esta información es rastreada, sería de mucho interés para un ciberdelincuente, ya que podría saber donde ha estado la persona o qué medicamentos tiene que tomarse. Por ejemplo, si nuestro coche llega una tarjeta RFID y está activa, se puede realizar un mapa de los sitios por donde hemos pasado.

A su vez, un atacante podría, por ejemplo, manipular la etiqueta de un artículo reduciendo su precio. Además, puede darse el caso de que los datos pueden ser transmitidos de forma segura y la base de datos no haya sido manipulada porque el artículo existe, pero el fraude se lleva a cabo porque una parte del sistema ha sido modificada (el precio). Por lo tanto, para que un sistema sea seguro, hay que tener en cuenta todos sus componentes. Si no tenemos en cuenta una parte, el nivel de seguridad del resto de componentes podría comprometer la seguridad de todo el sistema.

En el ejemplo, el ataque puede realizarse con un sólo artículo, pero se podría realizar otro para impedir que se produzcan ventas al anular todas las etiquetas, o por ejemplo, introducir información corrupta en la base de datos para que no pueda operar. Algunos ataques, que veremos en el apartado 3.1.4 como la jaula de Faraday (Faraday cage) o el la interferencia activa (active jamming), no tienen que nada ver con la tecnología inalámbrica y la transmisión de datos. Otros ataques se centran en no permitir el acceso físico, e ignoran los datos. Algunos implican el cruce fraudulento de fronteras, el robo de identidad a partir de pasaportes electrónicos, etc.

Debido al crecimiento de esta tecnología, la seguridad de los sistemas RFID es muy importante. Los requisitos de seguridad deseados de los sistemas RFID incluyen la confidencialidad, la integridad y la disponibilidad como base y algunas propiedades que se relacionan a continuación.

3.1.2 Seguridad en los dispositivos RFID

El objetivo de la seguridad en estos dispositivos es que las etiquetas solamente revelen su identidad a los lectores de RFID autorizados, de modo que los objetos puedan ser rastreados sólo por los dispositivos acreditados. Sin embargo, por razones de privacidad las etiquetas no deben revelar su identidad hasta que el lector se haya autenticado; por tanto, el lector debe autenticarse ante las etiquetas antes de hacer cualquier otra cosa.

Es evidente que no todos los dispositivos RFID necesitan el mismo nivel de seguridad, es necesario analizar y evaluar cada sistema (sensibilidad de los datos, como se producen las pérdidas, su criticidad, etc.) para determinar los requisitos exactos de confidencialidad, integridad y disponibilidad. Por ejemplo, los requisitos de seguridad de las etiquetas utilizadas en los pasaportes electrónicos no deberían ser iguales a los empleados en la cadena de suministro (es decir, una etiqueta que cumpla con la clase 1 de generación 2 del EPC) (Ver ANEXO I).

Para disponer de un dispositivo RFID seguro, deberemos tener en cuenta las siguientes propiedades:

• Confidencialidad. La información debe ser accesible sólo para las personas autorizadas para el acceso. Se debe evitar que terceros sean capaces de acceder a la señalización intercambiada entre los lectores y las etiquetas. Además, aparte de la confidencialidad, es importante señalar que la tecnología RFID permite el seguimiento de los artículos y que desde el punto de vista de la persona, es importante que evitemos ser rastreados. Sin embargo, hay empresas que pueden aprovechar esta propiedad para controlar los movimientos de sus productos en las cadenas de suministro, aumentando la productividad de sus procesos.

• Integridad. Hay que garantizar que los mensajes transmitidos entre el lector y la etiqueta no se modifican. Además, algunos sistemas proporcionan la autenticidad de los mensajes. En ocasiones, el destinatario puede incluso demostrar que un mensaje fue originado por el supuesto remitente y no es no es una falsificación (no repudio). Un ejemplo de este tipo de ataque es el ataque de suplantación de identidad o spoofing.

• Disponibilidad. La disponibilidad de un sistema RFID consiste en saber si (o con qué frecuencia) el sistema puede ser utilizado con normalidad por los usuarios. Este factor determinará el rendimiento y el nivel de escalabilidad del sistema RFID. Los ataques DoS son amenazas habituales contra la disponibilidad (por ejemplo, interferir la señal de radio o impedir el funcionamiento normal de las etiquetas de proximidad mediante algún tipo de etiqueta que impida su lectura).

A parte de los tres pilares de la seguridad de la información, los sistemas RFID deben garantizar el número máximo de estas propiedades, aplicándolas si se considerar necesarias dada la variedad de campos en las que son utilizadas:

• Identificación. La propiedad de identificación permite a un lector reconocer una etiqueta a partir de la salida emitida por esta. Cuando se fabrican las etiquetas RFID, reciben un identificador único que se escribe en la ROM (memoria sólo de lectura) de la etiqueta, y por lo tanto es difícil cambiarlo. Se puede conseguir la identificación lector/etiqueta y viceversa sin utilizar la criptografía, pero esto puede dar como resultado una fuga de datos porque se envían en claro, lo que podría dar lugar, por ejemplo, a ataques de rastreo como se verá más adelante.

• Autenticación. Cuando un lector recibe los datos de una etiqueta, no puede saber si estos datos son de una etiqueta válida o no a menos que se añada un sistema de validación. Al igual pasa al revés, cuando una etiqueta recibe datos de un lector.

Para asegurar que las comunicaciones se hacen entre dispositivos válidos, hay que incorporar un mecanismo de autenticación en el sistema que permita garantizar que un lector sólo aceptará los datos de una etiqueta y que una etiqueta sólo aceptará los datos de un lector, si pueden asegurar su validez.

• Privacidad. Como las etiquetas llevan un identificador único, se pueden producir problemas de privacidad al utilizar dispositivos RFID. Así, por ejemplo, como hemos comentado, se podrían seguir los movimientos de una persona o de un objeto que lleve una etiqueta RFID adherida. Las necesidades específicas de privacidad dependerán en gran medida de la aplicación concreta que se esté empleando en las etiquetas RFID. Las amenazas más relevantes a la privacidad personal son:

- Accesos no permitidos a las etiquetas: Éstas pueden contener datos personales, como nombres, fechas de nacimiento, direcciones, etc. Información que puede ser interesante para un ciberdelincuente.
- Rastreo de las personas y/o de sus acciones, gustos, etc.: Una persona con una etiqueta RFID, puede usarla para pagar en establecimientos, transportes públicos, accesos a recintos, etc., podría ser observada y clasificada.
- Uso de los datos para el análisis de comportamientos individuales: Utilizando técnicas de “minería de datos”, este análisis permitiría definir perfiles de consumo basados en las preferencias de los clientes.

• Indistinguibilidad. La indistinguibilidad es una propiedad muy relacionada con la privacidad. Decimos que una etiqueta tiene indistinguibilidad si un atacante que realiza una escucha pasiva no es capaz de distinguir entre dos etiquetas diferentes sólo observando sus salidas.

• Seguridad hacia adelante (*forward security*). Se trata de una extensión de las propiedades de autenticidad e indistinguibilidad que garantiza que dichas propiedades se mantienen para transacciones pasadas cuando un atacante es capaz de manipular una etiqueta en un momento determinado. Por ejemplo, imaginemos que se tira a la basura una etiqueta RFID que disponía de mecanismos para garantizar la autenticidad y la indistinguibilidad una vez acabada su vida útil. En ese momento, un atacante puede recuperar la etiqueta, manipularla y obtener los valores secretos que contiene.

Si incluso con esta información, el atacante sigue sin ser capaz de distinguir entre las salidas de dos etiquetas que registró en el pasado (una de las cuales pertenecía a la etiqueta comprometida), entonces decimos que tiene seguridad hacia adelante.

• Delegación y restricción. Estas propiedades son necesarias en aplicaciones en las que las etiquetas son reutilizadas por varios propietarios. En estas aplicaciones, se quiere que el propietario original pueda delegar el derecho de rastrear una etiqueta en un nuevo propietario, al asegurar que, una vez delegados los derechos, el propietario original pierde la capacidad de rastrearla.

• Prueba de existencia. La prueba de existencia es una propiedad que permite garantizar la existencia de una etiqueta particular en una localización concreta, en un tiempo determinado y con un conjunto de otras etiquetas particulares. Esta propiedad es necesaria, por ejemplo, en aplicaciones en las que se asignan etiquetas RFID a los diferentes componentes que forman parte de una cadena de suministro, de manera que varios lectores distribuidos a lo largo de la cadena puedan controlar su funcionamiento. En este caso, es interesante que el lector sea capaz de detectar que una serie de componentes se encuentran juntos en un espacio en un momento determinado (por ejemplo, si estos componentes se tienen que combinar para formar una sola pieza).

• Límite de distancia. Con el fin de dificultar los ataques de *relay* (que describiremos más adelante), se puede intentar limitar la distancia entre una etiqueta y un lector. Para ello, se limita el tiempo de ida y vuelta (*round trip time*) de los intercambios entre el lector y la etiqueta.

• Sincronización. En protocolos basados en máquinas de estados (donde las diferentes partes van cambiando de estado a medida que avanza el protocolo), un atacante puede provocar que el protocolo no se complete con éxito perturbando o retardando las comunicaciones entre la etiqueta y el lector, es decir, provocando una desincronización. La propiedad de sincronización permite a una etiqueta y a un lector volver a sincronizarse después de haberse desincronizado durante la ejecución de un protocolo.

3.1.3 Amenazas de seguridad

Dado su bajo coste y lo útil que resulta la identificación de un objeto sin contacto físico, los dispositivos RFID se seguirán haciendo cada vez más populares. Sin embargo, esto también puede generar muchos más problemas ya que son vulnerables a ciertos ataques que evidentemente también se producen en otros sistemas de información.

Como se ha mencionado, la tecnología RFID puede ser un sustituto de la tecnología de códigos de barras, sin embargo, pueden existir mayores riesgos si se produce algún fallo. Por ejemplo, si un lector de códigos de barras falla, se puede realizar una introducción manual para leer la etiqueta (nos pasa mucho cuando pasamos artículos por los lectores cuando vamos al supermercado), pero cuando falla una lectura de RFID no es posible, si además si procesa un gran volumen de artículos puede que no nos demos cuenta en un primer momento y tengamos que arreglarlo a posteriori, con lo que esto puede suponer, con lo cual hay tener muy en cuenta las amenazas de seguridad.

Vamos a clasificar estas amenazas según su objetivo, de acuerdo con (Khattab, A. (et al). 2017 y Espejo, C., 2018 [18]): en físicas, de canal o del sistema.

Amenazas físicas en RFID

Son aquellas que utilizan medios físicos para atacar al sistema RFID, concretamente a las etiquetas: desactivándolas, modificándolas o imitándolas.

- Deshabilitar etiquetas temporal o permanentemente. El atacante, para deshabilitar permanentemente una etiqueta, puede por ejemplo aportar una onda de alta energía que la haga inutilizable para siempre. Para desactivar la etiqueta temporalmente, se puede por ejemplo generar una señal en el mismo rango que el lector para evitar que las etiquetas se comuniquen, lo que se conoce como bloqueo activo.
- Ataque de reescritura del lector, una etiqueta falsa consigue comprometer la memoria de un lector. Los ataques de este tipo pueden romper la identificación, la autenticación o la privacidad de un sistema RFID.
- Ataque de reescritura de la etiqueta. En un ataque de reescritura de la etiqueta, el atacante reescribe los contenidos de la memoria de una etiqueta utilizando un lector falso. Este tipo de ataques se puede prevenir exigiendo a las etiquetas que autentiquen los lectores antes de permitir reescribir contenido o bien desplegando mecanismos de bloqueo de memoria.
- Duplicar exactamente una etiqueta copiando su número de identificación único para que la falsificada actúe como la original. (Se suele utilizar en tarjetas de crédito)
- Pickpocketing RFID. El pickpocketing es una técnica que consiste en la sustracción de diversos objetos de pequeño tamaño (carteras, relojes, teléfonos móviles, cinturones, etc.), aplicado al robo de información de dispositivos RFID, se relaciona con la acción de sustraer de manera pasiva la información contenida en la tarjeta o dispositivo RFID sin que la víctima lo note.
- Denegación de servicio y destrucción. En este ataque, el atacante intenta destruir físicamente una etiqueta para dejarla inutilizable o bien lleva a cabo un ataque de denegación de servicio que impide que los demás usuarios accedan al sistema.

Amenazas de canal en RFID

Las amenazas de canal se refieren a los ataques dirigidos a la comunicación entre el lector y una etiqueta debido a que es inalámbrica. Estos problemas de seguridad de comunicación suelen conducir a filtraciones de privacidad.

- Ataque de escaneo (ataque pasivo): ocurre cuando un atacante escucha los datos que se intercambian entre la etiqueta y el lector, normalmente para descubrir información secreta. Esta amenaza es más probable que sucedan en los sistemas que trabajan con UHF (*Ultra High Frequency*) pues cubren más distancia de lectura.
- *Eavesdropping*: también conocido como *sniffing* o *snooping attack*, se produce cuando un lector RFID no autorizado (diferencia con el espionaje) escucha las conversaciones entre una etiqueta y un lector y luego obtiene datos importantes. Es necesario que el hacker conozca los protocolos específicos y la información de la etiqueta y el lector con lo que no es fácil de implementar.
- Ataques de rastreo. En un ataque de rastreo, el atacante persigue a una etiqueta (o a la persona u objeto que lleva dicha etiqueta) para obtener datos como los sitios donde va, cuando suele comprar, cuando saca dinero del cajero, etc.
- Ataque de repetición (*replay attacks)*: cuando un dispositivo malicioso reproduce la información clave que escucha en la comunicación entre lector y etiqueta. La aplicación

más útil ocurre cuando el dispositivo ilegal reproduce la autentificación entre lector y etiqueta para vulnerar la verificación del sistema. Es por lo que los ataques de repetición permiten romper algunos sistemas de autenticación.

- Ataque de retransmisión (*relay attacks*): tiene lugar cuando el atacante utiliza un dispositivo ilegal entre lector y etiqueta de manera que intercepta la información para modificarla o reenviarla directamente. Este tipo de ataques también puede conseguir romper sistemas de autenticación.
- Ataque de modificación del mensaje. El atacante intercepta y modifica la comunicación entre una etiqueta y un lector. Para evitar este tipo de ataques, se añaden mecanismos de control de integridad a los datos que se envían. Los ataques de este tipo pueden romper la identificación, la autenticación o la privacidad de un sistema RFID. Por ejemplo, cuando se trata de información de un paciente y se modifica su medicación, puede acarrear consecuencias irremediables.
- Ataque de transmisión de software malicioso entre dispositivos RFID. Por ejemplo, un lector puede leer código malicioso de una etiqueta contaminada al ejecutar el código y pasar a reescribirlo en otras etiquetas a su alcance, con lo que se propaga el software malicioso.
- Ataque de bloqueo de señal (*Active Jamming*): Consiste en bloquear mediante interferencia la señal de radiofrecuencia que emite el dispositivo RFID y por lo tanto impide la comunicación entre la etiqueta y el lector. Una tarjeta garble (*GarbleCards*) es un dispositivo de seguridad del tamaño de una tarjeta de crédito que funciona distorsionando la señal RFID de tarjetas cercanas y de esta forma permite que la información no se transmita.
- Ataques de canal lateral (*Side channel attacks*). Dadas las características técnicas y físicas de los dispositivos RFID, por ejemplo un atacante puede observar la fuerza del campo electromagnético que se genera o analizar los tiempos de adquisición y procesado de datos para obtener información secreta almacenada en una etiqueta o un lector RFID.

Amenazas del sistema en RFID

Las amenazas del sistema se convierten en los principales ataques de la tecnología RFID y consiste en atacar al protocolo de autentificación y encriptación del sistema.

• Ataque de falsificación o suplantación se produce cuando el atacante utiliza lectores o etiquetas falsos y obtiene información que puede utilizar en su beneficio.

• Rastreo y seguimiento ilegal violando el concepto de privacidad de ubicación cuando el atacante envía consultas y recibe respuesta de una etiqueta en varias ubicaciones.

• **Descodificación del algoritmo de encriptación que utilizan actualmente la mayoría de sistemas RFID para garantizar la confidencialidad, el que nos interesa en este trabajo.**

• Ataque de denegación de servicio (DOS) mediante el cual el atacante bloquea el lector para que no lea las etiquetas y el sistema no funcione.

3.1.4 Medidas de seguridad frente a la amenazas de seguridad

Como ha se ha comentado, la fuga de información en los sistemas RFID es un problema que se produce cuando los datos enviados por las etiquetas revelan información sensible cuando son

consultados por los lectores. Normalmente cuando ocurre esto, los lectores no están autenticados y las etiquetas responden de forma totalmente transparente e indiscriminada.

A continuación se detallan algunos mecanismos que evitan la utilización de los datos transmitidos, las dividiremos en medidas físicas y medidas para garantizar la autenticación:

Medidas físicas para evitar la utilización de los datos:

- El comando "kill": el lector envía un código a la etiqueta que la apaga permanentemente e impide que pueda ser reactivada. Por ejemplo, un supermercado puede utilizar etiquetas RFID para el inventario y gestión de sus productos pero una vez se pase por caja la etiqueta se anularía.
- La jaula de Faraday: se trata de un envase realizado de un metal impenetrable para las señales de radio (o de ciertas frecuencias) que proporciona protección frente a los campos magnéticos y que impide que se pueda leer la tarjeta RFID. Un ejemplo son las bolsas que se venden para meter las llaves del coche y evitar que emitan señales de forma continua para saber si el coche esta cerca.
- Interferencia activa de señales de radiofrecuencia: estos dispositivos difunden activamente señales de radio para bloquear e interrumpir la operación de cualquier lector de RFID cercano. (los inhibidores de frecuencia que utilizamos en vehículos oficiales o a la entrada de los establecimientos militares)
- Etiquetas RFID inteligentes: estas etiquetas protegen mejor la privacidad utilizando métodos criptográficos. (por ejemplo la tarjeta que utilizamos para el DNI que utiliza claves RSA públicas y privadas, clave pública de root CA para certificados card-verificables y claves Diffie-Hellman)
- "Blocker tag": consiste en realizar un "bloqueo selectivo" de lectores RFID mediante etiquetas o tags. Un "blocker tag" es un dispositivo RFID pasivo y barato que puede simular simultáneamente el espectro completo de los posibles números de serie para los tags, ocultando de esta manera los números de serie de otros tags cercanos, y además no interfiriendo con la operación normal de los sistemas RFID. En una tienda, por ejemplo, ayudan a evitar la lectura no deseada de los productos que llevamos encima, pero no afectan al escaneo del inventario de la tienda.
- Etiquetado de etiquetas: en este caso el identificador único de la etiqueta se reetiqueta con otro identificador único. De esta forma la etiqueta puede ser controlada de mejor manera por la organización porque puede adaptarse más a sus intereses. En el caso de la tienda, la etiqueta de un producto puede ser de nuevo etiquetada para que no se pueda leer en otros lectores, únicamente en los de la tienda.

Medidas para garantizar la Autenticación

La autenticación como se ha comentado anteriormente es el proceso mediante el cual un objeto prueba su identidad a la otra parte de la comunicación y proporciona alguna evidencia para garantizar que es válido. Este proceso se implementa a través de soluciones de software. La autenticación en los sistemas RFID es necesaria en dos fases:

- antes de comenzar cualquier comunicación tanto etiqueta como lector deben verificar que están contactando con el dispositivo deseado.

- cuando los datos se intercambian, garantizar que son los legítimos y no han sido modificados.

Además la autenticación debe ser mutua, es decir, la etiqueta necesita saber si el lector es legítimo y se requiere que el lector averigüe si la etiqueta es confiable. De esta forma, la mayoría de los ataques expuestos anteriormente podrían resolverse.

Esta solución brinda beneficios cuando no son posibles las soluciones físicas, las cuales son adecuadas para aplicaciones muy concretas. Como ya se ha señalado, los sistemas RFID disponen de unos recursos limitados que vendrán determinados por el tipo de etiquetas o lectores que se utilicen.

3.1.5 Clasificación de los protocolos RFID

Dada la necesidad de conseguir dispositivos de bajo coste, la complejidad de los algoritmos que se pueden implementar se ve limitada, y por tanto, las soluciones criptográficas aplicables, siendo necesario recurrir a la criptografía ligera o muy ligera. En definitiva, los sistemas RFID que disponen de más recursos, podrán emplear criptografía más compleja para aumentar su seguridad

El coste de las etiquetas es un factor muy importante para su implantación en el mercado. Las etiquetas pasivas son las más comunes por las dimensiones en la que puede ser impresa y por su precio. Para una etiqueta RFID de bajo coste, hay unas 5K-10K puertas lógicas, y sólo 250-3K pueden utilizarse para funciones de seguridad.

Basándonos en el coste computacional y en las operaciones que admiten las etiquetas, se pueden clasificar los protocolos de autenticación RFID en cuatro clases [20].

- La primera clase, denominada “full-fledged class” (clase completa), se refiere a aquellos protocolos que exigen el soporte de de funciones criptográficas convencionales como el cifrado simétrico, funciones criptográficas de un sólo sentido, o incluso los algoritmos de clave pública.
- La segunda clase denominada “simple” es para aquellos protocolos que deben soportar el generador de números aleatorios y la función hash de una sola dirección en las etiquetas.
- La tercera clase denominada protocolos “lightweight” (ligeros) se refiere a aquellos protocolos que requieren un generador de números aleatorios y funciones simples como el Código de Redundancia Cíclica (CRC), que analiza pero no realiza la función hash.
- La cuarta clase denominada “ultralightweight” (ultraligera) se refiere a los protocolos que sólo implican simples operaciones (como XOR, AND, OR, etc.) sobre las etiquetas.

El reto que se nos plantea es diseñar un protocolo eficiente que satisfaga todos los requisitos de seguridad con recursos limitados y si la información se consiguiese por algún medio, que no se pueda descifrar. Además, aunque un adversario conseguir información de una etiqueta en una sesión concreta, no debería ser capaz de tener datos de las sesiones anteriores, de esta forma se mantiene el secreto hacia adelante. Si una etiqueta quedara comprometida no debería llevar a comprometer otras etiquetas del entorno.

La memoria de la etiqueta está restringida a varios cientos de bits, y aproximadamente de 250 a 4.000 puertas lógicas del espacio total de la etiqueta pueden dedicarse a tareas relacionadas con la seguridad.

El problema se solucionaría implementando algunas funciones criptográficas de bajo coste pero consideradas como seguras. Esto aportaría un alto nivel de privacidad en el que no sería necesario ningún cambio sustancial en la comunicación entre la etiqueta y el lector.

A continuación se analizará el primer caso en el que se intentará demostrar que la criptografía ligera o ultraligera es útil para garantizar la información transmitida por los dispositivos IoT, en este caso por un sistema RFID cuyos creadores denominan SASI [20].

3.1.6 Protocolo de Autenticación SASI

Lo más relevante de este protocolo es que proporciona autenticidad e integridad pero no confidencialidad y puede resistir todos los ataques vistos anteriormente. No obstante en el caso de RFID no es esencial la propiedad de confidencialidad a no ser que el tag envíe otra información que no sea su identidad, lo cual pueden hacer algunos protocolos más modernos. Las etiquetas sólo requieren operaciones simples a nivel de bits (AND, XOR, etc.). Estas excelentes características lo hacen muy interesante para los sistemas RFID de bajo coste y de muy bajo coste.

El protocolo se inspira en la familia de protocolos de autenticación mutua ultraligera (UMAP: M^2AP, EMAP, LMAP) e incluye tres entidades: etiqueta, lector y servidor. El canal entre el lector y el servidor final (backend) se supone que es seguro, pero el canal entre el lector y la etiqueta es susceptible de sufrir ataques. Cada etiqueta tiene una identificación estática (ID), y comparte un seudónimo (IDS) y dos claves K1=K2 con el servidor final. La longitud de cada una es de 96 bits ID=IDS=K1=K2.

Para resistir el posible ataque de desincronización (ver apartado 3.1.1.2 Seguridad en los dispositivos RFID), cada etiqueta mantiene en realidad dos entradas de (IDS, K1, K2): una es para los valores antiguos y la otra para los posibles valores siguientes.

El protocolo propuesto por Chien [20] consta de tres etapas:

- Fase de identificación de la etiqueta
- Fase de autenticación mutua
- Fase de actualización del seudónimo y de la clave.

El lector RFID puede preguntar a la etiqueta una o dos veces en la fase de identificación, dependiendo de si se encuentra el IDS de la etiqueta o no. El lector envía primero el mensaje "hola" a la etiqueta, y ésta responderá con su próximo IDS potencial.

El lector utiliza el IDS de respuesta de la etiqueta para encontrar una entrada que coincida en la base de datos, y pasa a la fase de autenticación mutua si hay una entrada que coincide; en caso contrario, volvería a preguntar y la etiqueta respondería con su antiguo IDS.

En la fase de autenticación mutua, el lector y la etiqueta se autentican y actualizan, respectivamente, su seudónimo local y las claves después de que se produzca la autenticación, la etiqueta almacena los valores coincidentes en la entrada ($IDS_{old} \| K1_{old} \| K2_{old}$) y almacena los valores actualizados en la entrada ($IDS_{next} \| K1_{next} \| K2_{next}$). El generador de números aleatorios se necesita únicamente en el lector, y las etiquetas sólo realizan operaciones simples a nivel de bits como XOR ($\oplus$), OR ($\vee$), AND ($\wedge$), suma en mod 2^m (+) y rotación a la izquierda (Rot (x,y))[10].

3.1.6.1 Introducción

Fase de identificación de la etiqueta. Inicialmente, el lector envía un "hola" a la etiqueta que primero responde con su siguiente identificador potencial.

Si el lector puede encontrar una entrada coincidente en la base de datos, pasa a la fase de autenticación mutua; en caso contrario, vuelve a preguntar y la etiqueta responde con su antiguo IDS.

[10] **Rot (x,y);** Función para girar a la izquierda el valor de x con los bits de y

Fase de autenticación mutua. El lector utiliza el IDS para encontrar un registro coincidente en la base de datos. Puede ser el próximo IDS potencial o el antiguo IDS de la etiqueta. A continuación, utiliza los valores coincidentes y dos enteros aleatorios generados n1 y n2 para calcular los valores A, B y C (las ecuaciones de cálculo se especifican en la Fig. 1). A partir de A || B || C, la etiqueta extrae primero n1 de A, extrae n2 de B, calcula K1´ y K2´ y luego verifica el valor de C. Si la verificación tiene éxito, entonces calcula el valor de la respuesta D.

Al recibir D, el lector utiliza sus valores locales para verificar D.

Fase de actualización del seudónimo y de la clave. Después de que el lector y la etiqueta se han autenticado mutuamente, actualizan su seudónimo local y las claves locales.

Este esquema también proporciona la confirmación de los valores de sincronización (K1´, K2´) cuando el lector y la etiqueta se autentican mutuamente con éxito. Esta propiedad lo hace robusto a los posibles ataques de desincronización.

En la Fig. 3-4 se clarifica la secuencia del protocolo.

Lector	1 hola	**Etiqueta**
Identificacion de la etiqueta	⇨	La etiqueta primero responde con el siguiente IDS y si no coincide con la BD, utiliza el viejo IDS
Autenticacion mutua: usa IDS para encontrar un registro	2 IDS ⇦	Extrae n1 de A y n2 de B
A=IDS⊕K1 ⊕ n1 B=(IDS ∨ K2) + n2 K1´=Rot(K1⊕n2,K1) K2´=Rot(K2⊕n1,K2) C=(K1⊕K2´)+(K1´⊕K2)	3 A||B||C ⇨	K1´=Rot(K1⊕n2,K1) K2´=Rot(K2⊕n1,K2) C´=(K1⊕K2´)+(K1´⊕K2)
Verifica D	4 D ⇦	Verifica que C=C´ Si se cumple D=(K2´+ID) ⊕((K1⊕K2) ∨K1´))
Actualización de seudónimos y claves IDS=(IDS + ID) ⊕ (n2⊕K1´)) K1=K1´; K2=K2´		IDS_{old}=IDS; IDS_{next}=IDS=(IDS+ID)⊕ (n2⊕K1´)) $K1_{old}$=K1 ; $K1_{next}$=K1´ $K2_{old}$=K2 ; $K2_{next}$=K2´

Leyenda: ⊕ = XOR ∨ = OR ∧ = AND (+) = Suma en mod 2^m Rot (x,y)=Rotación a la izquierda

Figura 3-2 Protocolo SASI (fabricación propia tomada de [20])

3.1.6.2 Análisis de seguridad

1. Autenticación mutua e integridad de los datos. La etiqueta y el lector pueden autenticarse mutuamente, ya que sólo el lector auténtico que tiene las claves K1 y K2 puede generar los valores A || B || C válidos, y sólo la etiqueta auténtica, que tiene las claves secretas, puede obtener los números aleatorios n1 y n2, y posteriormente generar la respuesta D.

Este protocolo es bastante diferente a los de la familia MAP, porque puede garantizar tanto la autenticidad y la integridad de los mensajes, mientras que sus homólogos no pueden asegurar la autenticidad ni la integridad.

Los cálculos de C y D implican claves secretas actuales, los dos números aleatorios y las nuevas claves. Por tanto, sólo el lector y la etiqueta auténticos tienen la capacidad de generar estos valores.

2. Anonimato de la etiqueta y resistencia al rastreo. El seudónimo de cada etiqueta se actualiza cada vez que se autentifica con éxito, y la operación de actualización implica números aleatorios. Por lo tanto los seudónimos de la misma etiqueta son aleatorios, y el atacante no puede identificar la identidad de la etiqueta y no puede rastrearla.

Por supuesto, si el atacante prueba sucesivamente la misma etiqueta muchas veces y consigue dos autenticaciones válidas, entonces la etiqueta responderá los mismos seudónimos (el antiguo y el nuevo). Esta situación permite al atacante rastrear la misma etiqueta; sin embargo, esta posibilidad es muy difícil que se dé.

3. Confidencialidad de los datos. El cálculo de cada valor de A, B C, y D implica al menos dos valores secretos (incluyendo las claves y los números aleatorios); así, la identificación de la etiqueta y los valores secretos están protegidos de las posibles escuchas.

4. Seguridad futura. La propiedad de seguridad futura significa proteger las comunicaciones anteriores de una etiqueta, incluso suponiendo que la etiqueta pueda ser comprometida algún día en el futuro.

En este esquema, si suponemos que un atacante compromete una etiqueta y adquiere las dos entradas de (ID, IDS, K1 y K2), el atacante no puede deducir los datos secretos y las claves anteriores de la misma etiqueta, porque cada una de las ecuaciones de actualización y los cálculos de A || B || D implican al menos dos valores aleatorios. Por lo tanto, el atacante no puede comprometer las comunicaciones anteriores de la misma etiqueta.

5. Confirmación explícita de la clave y resistencia al ataque de desincronización. Los anteriores esquemas de autenticación RFID que requieren la sincronización de los datos compartidos son vulnerables al ataque de denegación de servicio, porque el atacante puede modificar fácilmente los datos para que el lector y la etiqueta se desincronicen sin que se note.

Pero, en este protocolo, la autenticidad y la integridad de los valores aleatorios (n1 y n2) está garantizada, y las nuevas claves (K1´ y K2´) se confirman explícitamente, porque los cálculos de C y D implican explícitamente estos valores. Así, el atacante no puede cambiar los datos sin ser descubierto. Por supuesto el atacante puede interceptar los datos D enviados por la etiqueta para hacer que la etiqueta actualice sus datos locales mientras el lector no lo hace. Afortunadamente, esto no puede causar problemas a este esquema porque la etiqueta mantiene dos entradas de datos secretos (una es para los posibles valores nuevos y la otra para los valores antiguos), y puede autenticarse con el lector utilizando el valor antiguo.

Hay dos posibilidades para desincronizar los valores compartidos entre las etiquetas y los lectores: una es interceptar la respuesta D de la etiqueta, y la otra es hacer que el lector y la etiqueta utilicen diferentes n1 y n2 para actualizar sus datos locales.

En la primera posibilidad la etiqueta actualizará sus datos locales, pero el lector no. Sin embargo ya que la etiqueta mantiene dos entradas de sus datos locales (una para los valores antiguos y la otra para los nuevos), el lector y la etiqueta pueden autenticarse mutuamente en esta situación, utilizando los valores antiguos, con lo cual no habrá desincronización.

La segunda posibilidad no funciona en este esquema, porque es inviable para el atacante pueda cambiar los números aleatorios sin ser descubierto.

6. Resistencia a los ataques de repetición. El atacante puede reproducir la respuesta D de una etiqueta pero el lector descubrirá que el valor repetido no es válido, porque los números aleatorios n1 y n2 del lector son diferentes e independientes en cada sesión.

Otro escenario de repetición es que un atacante pueda interceptar la respuesta D en una sesión, y luego reproducir un mensaje antiguo A || B|| C (correspondiente a un antiguo IDS) desde el lector. Pero este escenario no cambiará el estado interno de la etiqueta, y el atacante no obtiene información secreta de la etiqueta. Por lo tanto, el atacante no obtiene información secreta ni desincroniza el lector y la etiqueta.

7. Resistencia al ataque del hombre en el medio (Man-In-The-Middle). El ataque del hombre en el medio no funciona en este protocolo, porque este esquema proporciona una integridad fuerte y una autenticación fuerte en los datos A || B || C y D. Cualquier modificación de los valores A o B causará, desde el punto de vista del atacante, cambios imprevisibles en los valores C y D, lo que hace que el atacante no pueda modificar datos sin ser descubierto.

8. Resistencia al ataque de divulgación. En los ataques de divulgación a otros protocolos ultraligeros, un atacante puede modificar ligeramente la información del lector y posteriormente deducir información parcial de la respuesta de la etiqueta.

Sin embargo, el ataque no funciona en este esquema, porque cualquier ligera modificación en la transmisión será detectada.

En la tabla 3-3 se recogen la resistencia de los algoritmos de la familia MAP frente al protocolo SASI. Como se puede observar el protocolo SASI está optimizado frente a este tipo de ataques.

ATAQUES	**LMAP**	**L^2AP**	**EMAP**	**SASI**
Resistencia a los Ataques de resincronizacion	NO	NO	NO	SI
Resistencia al Ataque de divulgacion	NO	NO	NO	SI
Privacidad y anonimato	NO	NO	NO	SI
Autentificacion mutua y seguridad hacia adelante o futura	NO	NO	NO	SI
Número total de mensajes de autenticación mutua	4L	5L	5L	4L
Tamaño de la memoria de una etiqueta	6L	6L	6L	7L
Tamaño de la memoria de cada etiqueta en el servidor	6L	6L	6L	4L
Tipos de operaciones en la etiqueta	$\oplus$, $\vee$, $\wedge$, +	$\oplus$, $\vee$, $\wedge$, +	$\oplus$, $\vee$, $\wedge$	$\oplus$, $\vee$, $\wedge$, +, Rot

Leyenda: L → longitud en bits de un seudónimo o clave. $\oplus$ = XOR $\vee$ = OR $\wedge$ = AND (+) = Suma en mod 2^m
Rot (x,y)=Rotación a la izquierda

Tabla 3-2 Comparación entre los Protocolos de Autenticación Ultraligeros (Fabricación propia tomada de [20])

3.1.6.3 Evaluación del rendimiento

Como hemos podido comprobar, el protocolo es resistente a los ataques propuestos porque así se ha diseñado, pero esto puede conllevar un coste computacional, de transmisión de datos o de almacenamiento elevado que impediría su implementación. A continuación se evaluará su rendimiento.

Coste computacional: la etiqueta sólo realiza operaciones simples bit a bit: XOR, AND, OR, suma en mod 2^m (+) y rotación a la izquierda (Rot (x,y)). Estas operaciones son de muy bajo coste y pueden implementarse de forma eficaz en dispositivos RFID de bajo coste y de muy bajo coste.

Coste de transmisión de datos: sólo tenemos que contar los mensajes en la fase de autenticación mutua, ya que esta fase aporta la mayor parte del coste de la comunicación. En esta fase, la etiqueta y el lector transmiten A || B || C y D, que en total exigen 4x96=384 bits, que es un coste muy bajo.

Coste de almacenamiento de la etiqueta. Para cada etiqueta tenemos un ID estático y dos entradas para (IDS,K1,K2). Por tanto, se necesita una memoria ROM para almacenar la identificación estática de 96 bits, y se requiere una memoria reescribible de 576 bits para almacenar las claves actualizables y el seudónimo, que serían IDS_{old}, IDS_{next}, K1, K2, K1´y K2´, en total 6x96=576 bits.

Se ha comentado que otros esquemas análogos son vulnerables a los ataques de desincronización y los ataques de divulgación completa, donde el ataque de desincronización hace que el lector y la etiqueta estén desincronizados, y puede comprometer toda la información secreta de las etiquetas. Por lo tanto, estos esquemas no logran autenticación mutua, privacidad, confidencialidad y recuperación de datos como SASI.

De este análisis se desprende que el esquema propuesto es muy eficiente y de bajo coste y además como se ha comprobado, ofrece resistencia a todos los posibles ataques enumerados en el Análisis de Seguridad. Este es un ejemplo en el que la criptografía ligera es útil en dispositivos RFID.

3.2 Redes WSN

Una red de sensores inalámbricos (WSN) está formada de pequeños dispositivos autónomos o nodos, con recursos limitados como baja potencia computacional, transmisión de datos limitada y restricciones de potencia, y que utilizan sensores para monitorear condiciones físicas o ambientales, procesan estos datos y los envían por señales de radio.

El consumo es una parte crítica, se necesitan sistemas con el consumo más bajo posible ya que existen muchas limitaciones energéticas, la batería debe durar el máximo tiempo.

Las redes de sensores inalámbricas están de actualidad y se han integrado con otras ciencias como son la medicina, biología, minería, etc., al igual que en aplicaciones tecnológicas militares (detección de trazas y guía para ataques) y civiles (monitorizar la maquinaria de una empresa, detección de fuegos forestales, monitorización de la actividad de los pacientes en los hospitales, monitorizar posibles desastres ambientales o como interfaz entre los usuarios y la tecnología IoT)

Generalmente, las WSN no tienen una estructura fija y se instalan sensores según la demanda que se necesite. Por lo tanto, al formar una estructura dinámicas, permiten la adición o sustracción de nodos después del despliegue, permitiendo así el crecimiento de la red o reemplazar los nodos que tengan fallos.

En muchos casos no existe una estación de monitorización de nodos durante la vida operativa de la red, por lo que una WSN debe contar con mecanismos de autoconfiguración y adaptación en caso de fallos.

Cada nodo tiene una unidad de radio para la comunicación, una pequeña batería, un microcontrolador para la tomas de decisiones, un circuito analógico y uno o más sensores. Los sensores, están formados básicamente por una parte computacional encargada de almacenar y transmitir datos, y una parte sensora que puede estar formada por uno o más sensores, acústicos, sísmicos, infrarrojos, temperatura, presión, etc.

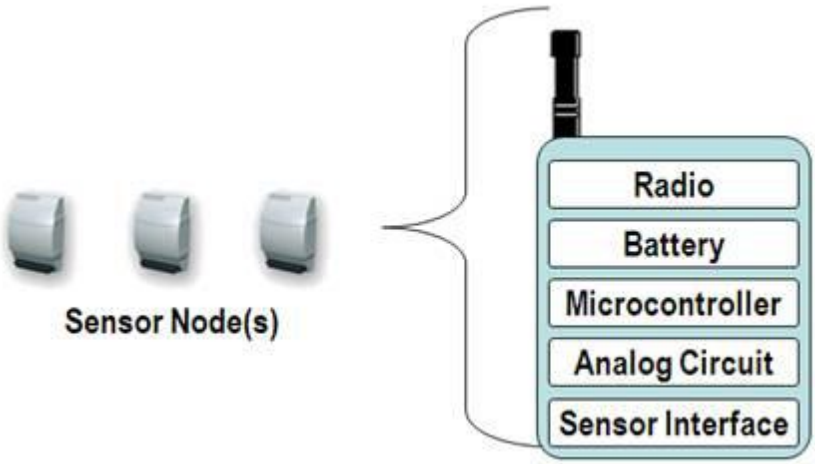


Figura 3-3 Partes de un nodo o sensor inalámbrico (tomada de [13])

Las aplicaciones más comunes de las WSN son:

1) Aplicaciones Militares. Se utilizar para recoger información militar, como por ejemplo rastrear a un enemigo, vigilar un lugar en concreto, o rastrear y clasificar objetivos en movimiento. También se utilizan para protección de propiedades, vigilancia y control de las fronteras.

2) Monitorización ambiental en ambientes cerrados. Con la ayuda de los sensores, se puede optimizar este ambiente. Por ejemplo calentadores o ventiladores que evitan sobrecostes en las empresas o los sensores que existen para predecir terremotos o evitar fuegos. Hoy en día los detectores de humo son comunes en edificios y permiten disparar alertas.

3) Monitorización ambiental al aire libre. Su uso es común en el rastreo de animales. Otras aplicaciones son el estudio de fenómenos naturales y de pronóstico del clima.

4) Agricultura. Para mejorar la eficiencia del trabajo, mejorar el crecimiento de los cultivos, reducir los costes, reducir su impacto ambiental y aumentar la calidad de los productos.

5) Monitorización de la salud. Cuidado de la salud e investigaciones científicas, porque se obtiene una información en tiempo real que permite tomar decisiones inmediatas. El BAN (Body Area Network) es un ejemplo, recolecta información sobre un paciente y si encuentra algo anormal emite una alerta.

3.2.1 Objetivos de los ataques en WSN

En las WSN cada uno de los nodos toman un rol dentro de la red, el nodo final (End-Device) toma la información de los sensores y la envía a un nodo enrutador (Gateway) que procesa la información la almacena o la envía a otro nodo enrutador.

Para las comunicaciones se utilizan protocolos de comunicación inalámbrica como Wifi, Zigbee o Bluetooth. Por este hecho, pueden ser fácilmente atacadas por la comunicación inalámbrica en modo broadcast[11] o de forma física, es decir un atacante puede utilizar técnicas de *eavesdropping* (inyección de paquetes maliciosos) o ataques a nodos de forma física.

Generalmente los sensores se centran en proteger la privacidad y la autenticación de nodos. Junto a la privacidad se busca confidencialidad e integridad total del contenido de los paquetes, permitiendo

[11] **Broadcast:** Difusion masiva de información de un nodo emisor o varios nodos receptores

una comunicación segura entre los sensores y los nodos administradores y que la información no sea modificada o errónea. Mientras tanto con la autenticación, se pretende que los nodos no autorizados no puedan participar en la comunicación de forma fraudulenta pudiendo generar falsos datos o recibiendo información privada.

Los objetivos de los ataques a una WSN son similares a los expuestos en RFID.

3.2.2 Seguridad en las redes WSN

Una WSN es un tipo especial de red que comparte algunas características en común con una red de ordenadores, pero también existen algunas particularidades que son únicas.

Los requisitos de seguridad más importantes en WSN son:

- Confidencialidad: Ver apdo. 3.1.2 aplicado a WSN
- Integridad: Ver apdo. 3.1.2 aplicado a WSN
- Autenticación: Ver apdo. 3.1.2 aplicado a WSN
- Disponibilidad: Ver apdo. 3.1.2 Aplicado a WSN
- Frescura: Los datos enviados deben ser recientes, y el sistema debe garantizar que no se puedan enviar datos antiguos. Esto es importante cuando los nodos dentro de la WSN usan claves compartidas ya que un atacante puede realizar un ataque de repetición (los mensajes antiguos son guardados y luego repetidos por el nodo atacante). Esto se puede solucionar añadiendo un nonce o contador específico de tiempo en cada paquete para poder chequear que sea reciente.
- No repudio: consiste en tener pruebas de la recepción/transmisión del envío por una tercera parte.
- Trazabilidad: información de auditoría tiene que ser guardada y protegida para que las acciones que afecten la seguridad del sistema puedan ser trazadas hasta el responsable.

3.2.3 Amenazas de seguridad en WSN

Las redes de sensores están expuestas a múltiples ataques, pero debemos proteger la autentificación y el envío de datos, que debe ser confidencial.

Los dispositivos sensores suelen están desplegados en grandes áreas, lo que presenta un problema para tenerlos controlados. Además, como ya se ha comentado con los dispositivos RFID, la comunicación inalámbrica permite que un atacante pueda desencadenar ataques sin tener acceso físico al dispositivo.

Los ataques los podemos clasificar en pasivos (si no modifican la información que se transmite, sólo la utilizan) y activos (realizan modificaciones a los datos transmitidos), siendo los principales ataques los siguientes:

Ataque pasivos en WSN:

• Espionaje: el atacante monitoriza el flujo de información en una red pero no intenta modificar su contenido. Obtienen información que les permitirá causar daño a las aplicaciones de los usuarios.

• El análisis de tráfico o Eavesdropping, en el cual el atacante observa el flujo de datos para obtener información acerca de la naturaleza de la comunicación. Las WSN son susceptibles de sufrir este ataque porque numerosos canales de comunicación transmiten datos a través de las redes de sensores.

• Jamming. El ataque de bloqueo consiste en la interferencia de la señal de radiofrecuencia que utilizan los nodos sensores para comunicarse.

Para reducir al mínimo amenazas de este tipo, es necesario usar técnicas que empleen un cifrado fuerte.

Ataque activos en WSN:

• Denegación del servicio (DoS): Este tipo de ataque ocurre a nivel físico, un nodo malicioso envía indiscriminadamente mensajes que consumen el ancho de banda disponible de la red, consiguiendo la indisponibilidad temporal o total del acceso a los recursos de la red.

• Ataque de colisión: Cuando dos nodos sensores intentan transmitir mientras están en la misma frecuencia, en este caso el paquete se descarta y debe retransmitirse.

• Ataques físicos: Es la sustracción física de los nodos de la red para sustraer la información y claves de criptografía.

• SinkHole: Se introduce un nodo malicioso cerca de la estación base (como ruta más eficiente para que sea utilizado por la mayoría de los nodos) para atraer información confidencial.

• Ataque Sybil: El atacante introduce múltiples nodos con identidades ilegítimas o con identidades robadas de la red para que sean considerados parte de la red.

• Ataque gusano (Wormhole): En este caso se genera un enlace de baja latencia entre dos nodos de la red, por el cual el atacante recolecta la información y la reenvía con cierto retraso para agotar los recursos de la red.

• Tampering: El ataque de manipulación se produce debido a la vulnerabilidad física de los nodos sensores distribuidos en grandes áreas, por lo que son susceptibles de captura, interrupción del circuito, modificación de la configuración o incluso sustitución de un nodo de red por un nodo malicioso. El objetivo es hacer pasar la versión modificada por original.

• El ataque de Reenvío Selectivo: es la participación de un nodo sensor por parte de un atacante que hace que algunos mensajes sean enrutados y otros descartados.

• El ataque Hello Flood, el atacante usa un transmisor de alta potencia para engañar a un gran número de nodos sensores, haciéndoles creer que están cerca cuando no lo están.

• El ataque de suplantación de reconocimiento consiste en difundir información falsa sobre el estado de los nodos sensores vecinos realizada por un nodo sensor malicioso para evitar que los paquetes lleguen a su destino.

• El ataque Flooding consiste en la avalancha de solicitudes a nuevas conexiones con el fin de agotar los recursos de memoria y evitar el cierre de requerimientos legítimos de provisiones.

• El ataque de desincronización se refiere a la interrupción de una conexión existente. El atacante captura mensajes que obligan al remitente a reenviarlos gastando energía innecesariamente.

• Ataque MITM (Man-In-The-Middle): intercepta los flujos de datos de la WSN que se transmiten a través de los canales de comunicación y manipula los mensajes que se envían al operador para falsificar las acciones que debe realizar.

3.2.4 Medidas de seguridad en WSN

Las medidas de seguridad se toman para detectar, prevenir o recuperarse de un ataque. Las técnicas criptográficas son la base de estas medidas siendo la encriptación de la información la más extendida.

Las medidas de seguridad utilizadas en WSN son la firma digital y los protocolos de seguridad. Los certificados digitales no son aplicables a las redes de sensores por su coste según Salinas Y., 2010 [29].

3.2.5 Protocolos de seguridad en redes WSN

Las redes de sensores inalámbricos, como se ha comentado, tienen sensores con limitaciones en el procesamiento y la comunicación. Estos sensores ocupan un área extensa y la comunicación puede ser monitoreada, estando sujetos a la captura o manipulación de datos sensibles por parte de un atacante, por lo que la seguridad y el enrutamiento seguro son fundamentales.

Un ejemplo de técnicas de seguridad que pueden ser aplicadas en WSN son las criptográficas y la detección de intrusos, mecanismos que proporcionan autenticidad, confidencialidad e integridad de la información.

Debido a las limitaciones de las WSN, no todas las soluciones de seguridad diseñadas para redes informáticas convencionales se pueden implementar directamente en una WSN. De hecho, durante mucho tiempo se creyó que la criptografía de clave pública no era apta para WSN porque requería alta potencia de procesamiento, pero a través de estudios de algoritmos de cifrado basados en curvas elípticas, se ha comprobado que dan buenos resultados.

Otros trabajos como Salinas Y., 2010, [29], exponen que "debido a que en el procesamiento y la transmisión de los datos es donde los nodos sensores tienen un mayor consumo de energía (Wang, 2005), y se concluye que la criptografía no es viable para obtener comunicaciones seguras"; además, según se comenta este estudio, "algunos autores también hacen referencia a las limitaciones en cuanto a la energía, comunicaciones y capacidad de procesamiento, para señalar que este tipo de algoritmos criptográficos, no son prácticos para redes de sensores (Eschenauer y Gligor, 2002). Aunque si el hardware del nodo sensor fuera capaz de soportar las operaciones de criptografía asimétrica, entonces si sería un método viable para la distribución de clave (Chan et al., 2004)"

En este último estudio se afirma que la distribución de claves en una red de sensores es impracticable, ya que se desconoce la topología de la red antes del despliegue de los sensores, el rango limitado en las comunicaciones, a las operaciones intermitentes del nodo sensor y a la dinámica de la red (Eschenauer y Gligor, 2002).

Lo que ha podido ocurrir es que estos estudios donde especifican que la criptografía no es el método idóneo para garantizar la seguridad de los datos en WSN, es que se hayan quedado obsoletos, porque en los últimos años se han analizado los algoritmos asimétricos en sistemas embebidos con baja potencia computacional y han dado buenos resultados.

RSA es actualmente el más utilizado entre los algoritmos asimétricos, trabajando desde la dificultad de factorizar números primos grandes, con la ventaja de que está estandarizado y alcanza una eficiencia relativamente buena. Los algoritmos basados en curvas elípticas como ya hemos comentado, pueden ser una alternativa a RSA, y los análisis que algunos autores han realizado, demuestran que es posible lograr buenos resultados con claves más pequeñas.

En 2010 se propuso el algoritmo de clave pública llamado MQQ [75], donde se comprobó que es más rápido que los algoritmos RSA y ECC en sistemas embebidos con baja potencia computacional.

Los algoritmos de clave privada (como AES y Twofish) y los algoritmos de cifrado de clave pública se han utilizado para proporcionar un nivel de seguridad alto, pero debido a los enormes cálculos que implican estos algoritmos, no pueden ser soportados eficientemente por las limitadas capacidades de procesamiento y almacenamiento de los sensores. Este hecho hace que debamos contar con algoritmos de cifrado seguros, con menos cargas computacionales y más flexibles, y aquí entra la criptografía ligera.

La investigación en este campo está creciendo debido a la idoneidad de la Criptografía ligera para los sistemas de redes de sensores inalámbricos y los sistemas embebidos.

Las WSN necesitan garantizar la seguridad, consumir menos energía y conseguir un alto rendimiento. Sin embargo, proporcionar estos tres requisitos en un sólo diseño es una tarea imposible porque sólo dos de los tres requisitos pueden encontrarse en un diseño ligero. Por lo tanto, en el desarrollo de un algoritmo criptográfico se tendrá en cuenta los tres requisitos (seguridad-rendimiento, seguridad-coste, rendimiento-coste)

3.2.6 Algoritmo ECDSA para WSN

A pesar de las altas limitaciones presentes en los nodos sensores, y las vulnerabilidades que poseen estos tipos de redes, mediante algoritmos criptográficos es posible proteger la comunicación y el canal.

Era lógico pensar que existirían algoritmos de criptografía ligera que sirvieran como ejemplo para garantizar los datos que se transmiten de forma segura, pero no ha sido posible encontrarlos.

Se analiza el algoritmo ECDSA (Elliptic Curve Digital Signature Algorithm) que es una modificación del algoritmo DSA que emplea operaciones sobre puntos de curvas elípticas en lugar de las exponenciaciones que usa DSA (problema del logaritmo discreto) y que no podemos definir como criptografía ligera aunque es un algoritmo que requiere números de tamaños menores para brindar la misma seguridad que DSA o RSA.

No obstante, algunos autores [31] han demostrado que usar Criptografía de clave pública en redes de sensores, se puede considerar una realidad y no un concepto teórico, debido a que la ECC aporta mejor rendimiento en cómputo y almacenamiento de claves que otros criptosistemas como el RSA.

Además, trabajar en el dominio de curvas elípticas, y concretamente, usando ECDSA [76] proporciona a la red operaciones relacionadas con firmas digitales, y esto a su vez, permite la posibilidad de dotar al sistema de servicios correspondientes a Infraestructuras de Clave Pública.

Sin embargo, no todas las primitivas de clave pública se reducen a ECC, sino que dependiendo del tipo de arquitectura del nodo y del rol que desempeña en la red, pueden usarse otros tipos de criptosistemas. Un ejemplo, sería el criptosistema de clave pública multivariada, también conocido MQ-esquemas [77], en el caso de que el nodo tuviera suficiente recursos de memoria, como una estación base que tampoco se ha definido como de criptografía ligera.

3.3 Smart cards

Podemos definir una Smart Card o tarjeta inteligente como una tarjeta del tamaño de una tarjeta de crédito que incorpora un chip electrónico.

Podríamos decir también que una smart card es un pequeño ordenador pero no manipulable ya que en la mayoría de los casos contiene una CPU y capacidad de almacenamiento que no se pueden editar, con lo cual aumenta su seguridad. De esta forma, en la propia tarjeta podemos almacenar datos secretos, certificados digitales o claves privadas asociadas a ellos. La tarjeta lleva a cabo por sí misma sus propias operaciones criptográficas, con lo cual es otra ventaja.

Los microprocesadores de tarjetas inteligentes intercambian datos con lectores de tarjetas y otros sistemas a través de una interfaz. La tarjeta inteligente en sí está alimentada por una fuente externa, generalmente el lector de tarjetas inteligentes. No necesita batería.

Las tarjetas inteligentes se comunican con los lectores mediante contacto físico directo o mediante RFID u otro estándar de conectividad inalámbrica de corto alcance. El chip o procesador de la tarjeta contiene datos a los que accede el lector de tarjetas. El procesador de la tarjeta contiene un sistema

operativo básico que permite que la tarjeta contenga, transmita y proteja los datos. El lector de tarjetas pasa los datos de la tarjeta inteligente a su destino, generalmente un sistema de pago o autenticación, a través de una conexión de red.

Figura 3-4 Smart card (tomada de https://empresayeconomia.republica.com)

Las razones por que las empresas utilizan smart cards:

- Proporcionan almacenaje a prueba de falsificaciones y no se pueden manipular.
- Computación aislada y con alta seguridad para la autenticación, firma digital e intercambios de claves con otras partes del sistema.
- Posibilita la portabilidad de credenciales y otra información privada entre ordenadores. Por ejemplo, en el caso de usuarios que se conecten en el trabajo y en casa, o para trabajadores móviles.

Por ello, el uso de tarjetas inteligentes ofrece una serie de ventajas:

- Seguridad multicapas: Un usuario no autorizado no puede iniciar sesión con las credenciales de otra persona con sólo saberse su password: tiene que utilizar una tarjeta física para poder hacerlo y conocer el identificador personal (PIN) asignado a la tarjeta.
- Un 'packet sniffer' no puede interceptar PINs a través de la red, puesto que nunca se transmiten.
- Una smart card sólo puede ser usada por una persona a la vez, haciendo imposible su uso concurrente y se bloquea al introducir mal el PIN tres veces.
- Las tarjetas inteligentes están diseñadas para resistir a intentos de falsificación o manipulación
- Al igual que una tarjeta de débito o crédito, si se pierde o roban una smart card, se puede usar un número de teléfono para cancelarla y activar la emisión de una nueva. Pero a diferencia de las tarjetas de débito o crédito, las tarjetas inteligentes se pueden emitir al momento.
- La tarjeta es portátil y fácil de llevar encima.
- Se pueden configurar las tarjetas inteligentes para conexiones remotas (dial-up o VPNs) mediante el protocolo EAP (Extensible Authentication Protocol).

3.3.1 Ejemplos de Smart cards

Los gobiernos de los principales países del mundo están empezando a implementar políticas para promocionar el uso de tarjetas inteligentes. España es uno de los países que ha apostado más decididamente, principalmente con 3 iniciativas:

1 El DNI electrónico (DNIe).

2 La Administración Electrónica. Se necesitará utilizar certificados digitales o el DNIe.

3 La tarjeta sanitaria. Los datos sanitarios son de los que mayor nivel de protección requieren. La LOPD obliga a extremar su protección.

3.3.2 Seguridad en las Smart cards

Las tarjetas inteligentes se caracterizan por:

- Integridad de los datos: Se utilizan firma digital para garantizar la integridad.

- Autenticación: Se necesita para verificar la identidad de los usuarios, por ejemplo en las transacciones financieras, verificaciones de SIM, etc. Se suele utilizar la criptografía de clave pública.

- Confidencialidad: Es necesaria para mantener los mensajes y la información sensible como el PIN, la información médica, datos académicos, etc., en secreto. Los algoritmos criptográficos simétricos se utilizan para proporcionar esta confidencialidad de los datos.

3.3.3 Ataques a las Smart cards

Aunque son ampliamente utilizadas, existen varias formas en que las Smart Cards puedan ser atacadas, uno de estos ataques es el DPA (Differential Power Analysis).

Cuando se generan claves, se cifra, o se firma digitalmente se realizan operaciones matemáticas que consumen más tiempo y energía. El ataque consiste en conocer el consumo de energía que es usado cuando una Smart Card hace esta operación y así deducir información de la clave privada.

El DPA consiste esencialmente en la diferencia de energía que toma la Smart Card al realizar las diferentes operaciones en el proceso de cifrado y, a partir de ahí, deducir información de la clave privada.

Para poder evitar este ataque, es necesario evitar que puedan tomarse las lecturas de energía necesarias para realizar las operaciones de cifrado.

A continuación se relacionan algunos más:

- Ataque interno: Lo posibilita el propio usuario autorizando el acceso a las tarjetas inteligentes. La parte autorizada puede atacarlas con mayor facilidad, ya que tienen conocimiento sobre el sistema operativo, los nodos de comunicación, etc. En el esquema de autenticación, un usuario tiene contraseña para acceder al servidor. Si la misma contraseña es utilizada por otro servidor, el usuario puede acceder a los datos de otro servidor. Esto conduce a un ataque interno.

- Ataque Man-In-The-Middle: Como hemos visto anteriormente es una forma de escucha activa, y ocurre cuando un atacante se sitúa entre el usuario y el servidor. El atacante tendrá acceso a información sensible de la Smart card sin que el servidor y el usuario lo sepan. Esto puede acarrear fuga de información privada como el PIN, el historial del paciente, la identidad del usuario, etc. datos de interés para un ataque posterior.

- Ataque de repetición: Cuando un usuario solicita al servidor el inicio de sesión transmitiendo la información de inicio de sesión a través de un canal, entonces el adversario accede a esa información,

cambia algunos contenidos de la misma y hace que la información sea adecuada para el inicio de sesión del adversario en el servidor. El usuario se da cuenta porque no puede iniciar la sesión.

- Ataque de suplantación de identidad: El atacante envía el ID del usuario al servidor. El servidor emite una contraseña y envía una tarjeta inteligente para el usuario al atacante. Al tener la contraseña y la tarjeta, el atacante accede a la información del usuario.

- Ataque de denegación de servicio: El adversario envía continuamente solicitudes de inicio de sesión erróneas para que el servidor esté ocupado e impida al usuario autorizado iniciar sesión.

- Ataque de robo de verificación: El atacante roba los datos de datos de verificación y autenticación del servidor. A partir de estos datos robados, el atacante genera los datos al servidor. El atacante envía los datos de comunicación correctos, y entonces se hace pasar por el usuario para la siguiente sesión de autenticación. Para resolver esto, se aplican algoritmos de criptografía en las tarjetas inteligentes.

3.3.4 Protocolos de seguridad en Smart cards

El aumento de las aplicaciones en línea, la verificación remota del usuario y la autenticación segura de las identidades digitales, se ha convertido en un tema que preocupa a las personas o empresas que utilizan esta tecnología. Para resolver estos problemas de seguridad se prefieren las tarjetas inteligentes, para ello en las tarjetas se aplican criptoalgoritmos cuando se envía la información sensible que está almacenada en la tarjeta. Las tarjetas utilizan los algoritmos más seguros y se crean con los chips más fiables, con lo que son virtualmente inviolables.

Iniciar sesión con una smart card ofrece una forma fuerte de autenticación porque usa identificación basada en criptografía y prueba de posesión, al autenticar un usuario a un dominio a través del PIN. Los esquemas de autenticación basados en ECC son altamente seguros contra varios ataques, pero al mismo tiempo requieren un alto poder computacional para los cálculos matemáticos. Por lo tanto, deben utilizarse algunas técnicas de optimización de energía, como el control del reloj, la reducción de la longitud del cable, el diseño de cables, diseño jerárquico, etc.

Cuando las tarjetas son criptográficas, las posibilidades de identificación y autenticación se multiplican, ya que se pueden almacenar de forma segura certificados digitales o características biométricas en ficheros protegidos dentro de la propia tarjeta, de modo que estos elementos privados nunca salgan de la tarjeta, pues las operaciones de autentican a través del propio chip criptográfico.

Hay numerosos algoritmos como RSA, DES, 3DES, AES, SHA, IDEA, RC5 que se utilizan para el cifrado de datos, el Código de autentificación de mensaje (MAC) se suele utilizar como primitiva de autenticación simétrica, y ECC y RSA como funciones Hash para la autenticación asimétrica.

Hay varios estudios en los que se busca la mejor implementación en cuanto al área y rendimiento, teniendo en cuenta las limitaciones de memoria, área y potencia de las de las tarjetas inteligentes. En ellos se puede resumir que DES proporciona la mejor compatibilidad, ECC tiene la mejor eficacia de cifrado entre los algoritmos de clave pública y AES tiene mejor rendimiento y seguridad.

Algoritmos como AES, Triple DES, y Blowfish tienen menos requisitos de memoria y área, pero son más compatibles en aspectos de seguridad que RSA y ECC. Las tarjetas inteligentes suelen aplicar estos cifrados de bloque para la seguridad de los datos, ahora bien, los bits de datos inferiores al tamaño del bloque se rellenan con ceros, lo que conlleva una operación adicional para el relleno aleatorio. Para reducir esta sobrecarga, es mejor aplicar el cifrado de flujo.

ECC se considera el mejor para la autenticación de tarjetas inteligentes, ya que su arquitectura dificulta extraer la información. Los cifrados simétricos ligeros como AES, TRIPLE DES y PRESENT se aplican para el cifrado de datos y son eficientes, pero también se pueden utilizar cifrados de flujo para evitar el rellenado de ceros de los bloques.

4 RESULTADOS / VALIDACIÓN / PRUEBA

4.1 Ejemplo práctico 1. Sonoboya

Se nos presenta el caso de estudio de implementar la criptografía en una red de sonoboyas submarinas intentando minimizar el consumo energético. Las sonoboyas están contenidas cilindros de 125 mm de diámetro, por 915 mm de largo y equipadas con un pequeño radio transmisor de VHF de cerca de 1 vatio de potencia, que transmite en alguno de los 99 canales en la banda de 136 a 173.5 MHz, para enviar la información recibida por su sensor al receptor ubicado en una aeronave.

Las Sonoboyas son del tipo AN/SS0-47 y emiten un pulso en la banda de audio de 12.82 kHz, es decir son activas.

En una aproximación razonable el tiempo mínimo entre pulsos podemos considerarlo de 10,00 kHz

Aplicando el Teorema de muestreo de Nyquist-Shannon[12] [54] tendremos que tomar una tasa de muestreo de 20,00kHz y a 16 bits por muestra tendremos 320.000 bits, es decir 313,42kB.

Asumiendo una compresión razonable de 1:4, el throughput o tasa de transferencia es de 78,35 kB/s

Según la tabla que mostraba el rendimiento de los distintos cifrados [ANEXO III], no podemos aplicar ninguno con tasa inferior a los 78,35 kB/s con lo podremos implementar los cifrados AES, HIGHT, PRESENT, PRINCE Y TWINE.

La sonoboya tiene un consumo activo significativo pero su batería queremos resguardarla al máximo.

Dado que el número de puertas equivalentes es proporcional al consumo y queremos minimizarlo, escogeremos el algoritmo que menos puertas lógicas presente, que en este caso es PRESENT que tiene 1570 GE para una clave de 80 bits o 1884 GE para una clave de 128 bits y consumo de 2,35µw.

[12] **El teorema de Nyquist-Shannon:** Demuestra que la reconstrucción exacta de una señal continua en banda base a partir de sus muestras, es matemáticamente posible si la señal está limitada en banda y la tasa de muestreo es superior al doble de su ancho de banda.

Figura 4-1 Sonoboya AN/SSQ-47 (Fuente)

También cuenta con un buen nivel de seguridad y aunque es vulnerable al ataque de diferencial, podemos asumir que podría implementarse en este dispositivo.

4.2 Ejemplo práctico 2. Protocolo Ladon

En 2017 se puso en riesgo a casi medio millón de personas que portaban un marcapasos 'St Jude Medical' (SJM), debido a que la información que transmitía el dispositivo no poseía ningún tipo de cifrado.

Los marcapasos son dispositivos cardíacos controlados vía radio que controlan el ritmo cardíaco y, de ser necesario, envían una respuesta apropiada para hacer latir al corazón a un ritmo adecuado. También registran posibles anomalías en el ritmo cardíaco, información que monitoriza periódicamente y de forma remota por un médico.

La vulnerabilidad permitía que un usuario no autorizado equipado con un equipo sencillo, pudiera hackearlos de forma remota y reprogramar el marcapasos para agotar su batería o bien, alterar el ritmo de los latidos del paciente.

Figura 4-2 Marcapasos `St Jude Medical` (tomada de Internet)

La compañía publicó una actualización de firmware que parcheaba el agujero de seguridad y evitaba el secuestro del dispositivo. El problema era que no se podía hacer a nivel de usuario y se debía acudir a que el personal médico autorizado realizara el procedimiento.

El Protocolo Ladón, del equipo I2T (Investigación e Ingeniería Telemática) de la doctora Astorga [28] es un ejemplo de un protocolo de seguridad, que precisamente garantiza que la persona que se está conectando es quien dice ser, está autentificado, identificado y tiene autorización y permisos para utilizar el marcapasos, y donde la idea principal es que los datos que se transmiten vayan cifrados y protegidos.

El protocolo se ha confeccionado para que tenga un consumo muy reducido ya que es fundamental no tener cambiar las baterías ya que la sustitución de las baterías supone volver a operar al paciente. Por otra parte, la memoria y la latencia del protocolo también son reducidas.

Ladon, una versión mejorada de Kerberos[13] que amplía el protocolo original con capacidad de autorización y sustituye la necesidad de sincronización del reloj añadiendo al protocolo nonces (números aleatorios usado una sola vez destinados a la autenticación de transferencia de datos entre dos o más partes) especiales de duración limitada. De este modo, aunque todas las entidades necesitan temporizadores, sólo los relojes de los dos servidores que constituyen el centro de distribución de claves deben estar sincronizados entre sí.

La propuesta para esta caso de estudio si nos hubieran encargado solucionar la vulnerabilidad de los marcapasos, es que necesitamos un sistema de cifrado asimétrico para establecer la clave de sesión con un método de criptográfico ligero. El este trabajo se presentaron dos, BLUEJAY y ECC LIGERA. Tendremos que analizar cuál de los dos es más adecuado para este caso.

BLUEJAY es adecuado para plataformas ultraligeras (un total de 2000-3000 GE) como los microsensores y las etiquetas de autenticación RFID. Además romperlo es difícil y está pensado para la autenticación en RFID por lo cual se considera más adecuado que la ECC LIGERA, que aunque es más eficiente, requiere más de 10.000 GE y esto incrementaría el consumo del marcapasos que, como se ha comentado, es un requerimiento esencial en este dispositivo.

[13] **Kerberos** es un protocolo de autenticación, pero no de autorización. Se encarga de identificar a cada usuario, a través de una contraseña sólo conocida por este, pero no determina a qué recursos o servicios puede acceder o no dicho usuario.

4.3 Ejemplo práctico 3. Hidrófono

Los hidrófonos son dispositivos que captan el sonido bajo el agua y lo convierten en señales de audio para posteriormente convertir estas señales en señales eléctricas que se pueden traducirse en datos medibles.

A diferencia del sonar, un hidrófono funciona sólo como un receptor y no emitirá ningún tipo de señal que rebote cuando se encuentre con algún tipo de objeto sólido.

A medida que se recopilan los datos, el hidrófono transmite la información a un panel de control en el barco. El software convierte los datos en gráficos y otras imágenes que se pueden estudiar con más detalle. Dependiendo del espaciado y la frecuencia de las señales de audio captadas por el hidrófono, los resultados pueden identificar un área que el barco de prospección desea explorar con más detalle.

El hidrófono que se va a utilizar en el proyecto es el Miniature Hydrophone Type 8103 de la empresa BRÜEL & KJÆR® Transducers que es más pequeño que tienen. Tiene una frecuencia de hasta 180kHz lo que lleva a una tasa de muestreo de 360kHz muestras. Si consideramos de nuevo 16 bits por muestra, eso supone 5.760 kB. Asumiendo una compresión bastante eficaz de, por ejemplo, un 25% del tamaño original, se tendría una tasa de datos significativa, en el entorno de los 1.440 kB = 1,4 MB.

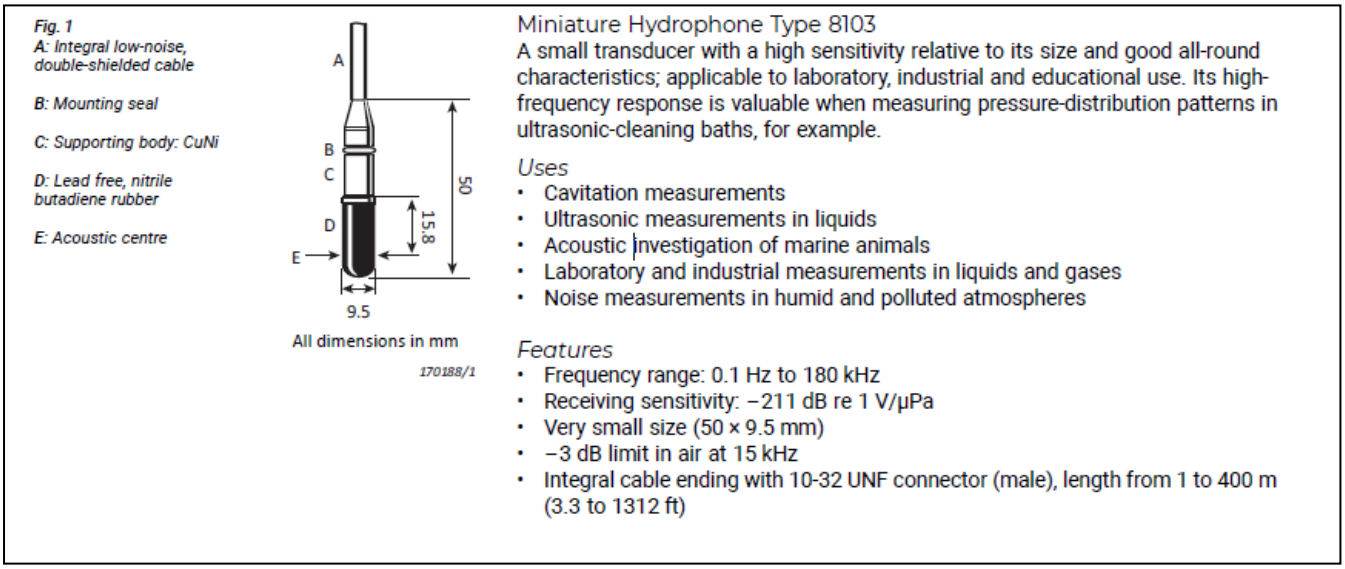


Figura 4-3 Miniature Hydrophone type 8103 (tomada de BRÜEL & KJÆR)

Utilizando la tabla que mostraba el rendimiento de los distintos cifrados (ANEXO III), tenemos que aplicar el cifrado PRINCE que es indicado para tasas superiores a 533,3 kB/s.

El consumo es algo mayor con 3.491 GE pero cuenta con un buen nivel de seguridad y aunque es vulnerable al ataque de reflexión, en este caso no hay problema porque el sistema no emite, sólo recibe datos.

5 CONCLUSIONES Y LÍNEAS FUTURAS

5.1 Conclusiones

Los dispositivos IoT están a la orden del día y seguirán en continuo crecimiento, vivimos en un mundo conectado a Internet y es evidente que se seguirán introduciendo dispositivos de este tipo en el futuro. Este concepto no es teórico, en la práctica existe la necesidad tecnológica de dispositivos que se conecten a Internet aunque cada vez tienen mayores limitaciones.

Estos dispositivos envían y reciben datos de Internet, y la seguridad de estos datos debe garantizarse con objeto de que no caigan en manos de ciberdelincuentes que puedan apropiarse de esta información, y lograr un beneficio en muchos casos monetario, de todo esto.

La criptografía es un método realista y de bajo coste, que puede ayudar a garantizar esta seguridad que necesitamos, pero al ser dispositivos con recursos limitados ya que no poseen ni memoria, ni CPU ni energía, es la llamada Criptografía Ligera la que puede dar respuesta a las necesidades crecientes de seguridad en dispositivos restringidos, porque no es posible aplicar los algoritmos más robustos de la Criptografía convencional al consumir demasiados recursos.

Estos algoritmos ligeros están avanzando mucho, sobre todo en los últimos 5 años como demuestra el esfuerzo de la NIST por estandarizar dichos algoritmos y la cantidad de publicaciones que se confeccionan implementando algoritmos ligeros. De hecho, la Criptografía Ligera está sustituyendo a los algoritmos de criptografía simétrica que son los que más se utilizan en Criptografía reduciendo su coste computacional pero garantizando su seguridad.

Es evidente que los algoritmos ligeros no son resistentes a todos los ataques criptográficos, pero ya que es imposible implementar una criptografía convencional, constituyen muchas veces la única forma de que estos dispositivos dispongan de seguridad.

Sin embargo, el término “ligero” abarca demasiados dispositivos y debería de existir una división que podíamos denominar criptografía ultraligera y criptografía ligera para categorizarlos mejor.

En particular, y aparte del consumo de energía, esta división es necesaria debido a los diferentes niveles de seguridad que exigen estos algoritmos. Hay que elegir bien estos algoritmos para adecuar el algoritmo utilizado al dispositivo.

Se han presentado algunos de los algoritmos más importantes de criptografía ligera y se han evaluado tres casos prototipo que sirven para ilustrar este tipo de tecnología. RFID que proporciona autenticación e integridad en los datos y WSN y Smart Cards que proporcionan además de autenticación e integridad, confidencialidad en los datos.

En RFID el protocolo SASI es un ejemplo de algoritmo de criptografía ligera útil para este tipo de dispositivos. En WSN en cambio no se ha podido encontrar una primitiva criptográfica ligera que solucione sus problemas. En Smart Cards se han encontrado algoritmos de cifrados simétricos ligeros como AES, TRIPLE DES y PRESENT que se aplican al cifrado de datos y dan un buen resultado. También son útiles los cifrados de flujo ligeros para evitar el rellenado de ceros de los cifrados de bloques.

5.2 Líneas futuras

Como líneas futuras, cabría la posibilidad de encontrar un algoritmo ligero para WSN que funcione de forma correcta, y completar el estudio analizando los tiempos de los algoritmos para conocer las diferencias entre implementaciones convencionales y las ligeras.

6 BIBLIOGRAFÍA

[1] Thakor, V.A (et al.), «Cryptography Algorithms for Resource-Constrained IoT Devices», Digital Object Identifier, 2019

[2] Saddkhan, S.B. y Salman, A.O., «A Survey of Lightweight-Cryptography. Status and Future Challenges», International Conference on Advances in Sustainable Engineering and Aplications (ICASEA), 2018

[3] Philip, M.A.,«A Survey On Lightweight Ciphers For IoT Devices», IEEE International Conference on Technological Advancements in Power and Energy (TAP Energy), 2017

[4] Mobahat, H., «Authentication and Lightweight Cryptography in Low Cost RFID», 2ª International Conference on Software Technology and Engineering (ICSTE), 2010

[5] Gunathilake N. A. (et al.), «Next Generation Lightweight Cryptography for Smart IoT Devices: Implementation, Challenges and Applications» IEEE 5th World Forum on Internet of Things (WF-IoT), 2019

[6] Rodríguez Flores L. A., «Arquitectura hardware compacta para criptografía ligera de llave pública» Grado Maestría en Ciencias en la especialidad de Ciencias Computacionales, 2014

[7] Lucena López M. J. «Criptografía y seguridad en computadores» 4ª Edición, Version 0.6.2

[8] Garcia Flores L. A., «Tesis en Maestría en ciencias en la especialidad de ciencias computacionales» Instituto Nacional de Astrofísica, Óptica y Electrónica (INAOE), 2014

[9] Vales Alonso J. «Guía de estudio a la Criptografía » Seguridad en Sistemas de Información. Master en Direccion TIC para la Defensa. Universidad de Vigo, 2020-2021

[10] Huidobro J. M., «Introducción a la protección de la información. “criptografía”», COIT (colegio oficial de ingenieros de telecomunicación)

[11] Fernandez Toledo J., Formacion IT, 21 nov 2020, Disponible: https://jesusfernandeztoledo.com/criptografia-asimetrica/ [Ultimo acceso 31/12/2021]

[12] «Datalong16 presenta su dispositivo para asegurar el estado de los envíos en tiempo real» Disponible: : https://www.cadenadesuministro.es/noticias/datalong16-presenta-su-dispositivo-para-asegurar-el-estado-de-los-envios-en-tiempo-real/ [Ultimo acceso 18/01/2022]

[13] Eterovic J.E. y Cipriano M. J., «Stream Ciphers livianos estandarizados mediante normas internacionales para ser usados en Internet de las Cosas». Sistemas, Cibernética e Informática Volumen 15 - Número 2 - 2018

[14] Pandurang S., «Light Weight Cryptography Schemes for Resource Constraint Devices in IoT» Disponible: : https://www.riverpublishers.com/journal_read_html_article.php?j=JMM/15/1/5 , Revista Multimedia movil, 2020 [Ultimo acceso 31/12/2021]

[15] Khalid M. (et, al), «Ultralightweight RFID Authentication Protocols for Low -Cost Passive RFID Tags» Security and Communication Networks, Article ID 3295616, 2019

[16] Eisenbarth T. (et al.), «A Survey of Lightweight-Cryptography Implementations», IEEE Design & Test of Computers,

[17] Junta de Castilla y León «RFID: Tecnología de identificacion por Radio frecuencia y sus principales aplicaciones». 2007

[18] Espejo, C. «Estudio de las aplicaciones de la tecnología RFID y su grado de implantacion» Trabajo fin de Grado Escuela Tecnica Superior de Ingeniería. Sevilla, 2018

[19] Peris, Pedro. «Lightweight Cryptography in Radio Frequency Identification (RFID) Systems», Tesis Universidad Carlos III. Madrid, 2008

[20] Chien H., «SASI: A New Ultralightweight RFID Authentication Protocol Providing Strong Authentication and Strong Integrity», IEEE Transactions on dependable and secure computing, vol. 4, no. 4., 2007

[21] Juels A., «RFID Security and Privacy: A Research Survey», IEEE Journal on selected areas in communications, vol. 24, no. 2, 2006

[22] Bhardwaj I. (et al.), «A Review on Lightweight Cryptography Algorithms for Data Security and Authentication in IoTs», 4ª IEEE Interntational Conference…, 2017

[23] Buchanan W. J. (et al.), «Lightweight cryptography methods», Journal of Cyber Security Technology, 2018

[24] Khalid M., «Ultralightweight RFID Authentication Protocols for Low-Cost Passive RFID Tags», Security and Communication Networks, 2019

[25] Lee S.M. (et. al.), «Efficient Authentication for Low-Cost RFID Systems», Center for Information Security Technologies (CIST), 2005.

[26] «Guía de Seguridad de las TIC CCN-STIC 817», Esquema Nacional de Seguridad. Abril 2020 Gestión de ciberincidentes

[27] Biryukov A. y Perrin L., «State of the Art in Lightweight Symmetric Cryptography», Journal of Ambient Intelligence and Smart Environments, 2014

[28] Astorga, J (et. al.), «Securing access to next generation IP-enabled pacemakers and ICDs using Ladon», Journal of Ambient Intelligence and Smart Environments 6(2):157-177, DOI:10.3233/AIS-140250, Marzo 2014

[29] Salinas Y., «Propuesta de configuraciones de seguridad para Redes de Sensores Inalámbricos», Facultad de Ingeniería Eléctrica, Departamento de Telecomunicaciones y Electrónica, Trabajo de diploma, 2010

[30] Eschenauer L. y Gligor V., «A Key-Management Scheme for Distributed Sensor Networks», Proceedings of the ACM Conference on Computer and Communications Security, 2002

[31] Alcaraz C. (et. al), «Análisis de primitivas criptográficas para redes de sensores», VI

Jornadas de Ingeniería Telemática (JITEL07), pp. 401-408, 2007.

[32] Banegas G., «Attacks in Stream Ciphers: A Survey», Department of Computer Science, Federal University of Santa Catarina, 2014

[33] Panasenko S. y Smagin S., «Lightweight Cryptography: Underlying Principles and Approaches», International Journal of Computer Theory and Engineering,Vol. 3, No. 4, Agosto 2011

[34] Shreyas S. y Nagaraja G.S, «An Evaluation of Lightweighted Data Security and Authentication Schemes for IoT Devices», High Technology Letters, ISSN NO: 1006-6748, 2020

[35] Kaur J. (et. al.), «Lightweight Cipher Algorithms for Smart Cards Security: A Survey and Open Challenges», Proceedings of the 4th International Conference on "Signal Processing, Computing and Control", ISPCC- 2k17; IEEE Conference ID: 40546, 21st-23rd Septiembre 2017

[36] «RC4» De Wikipedia, the free encyclopedia, Disponible en: https://es.wikipedia.org/wiki/RC4 [Ultimo acceso 18/01/2022]

[37] Handschuh H. y Gilbert H., «X^2 Cryptanalysis of the SEAL Encryption Algorithm», Published in E. Biham, Ed., Fast Software Encrytion, vol. 1267 of Lecture Notes in Computer Science, pp. 1{12, Springer-Verlag, 1997

[38] «A5/1» De Wikipedia, the free encyclopedia, Disponible en: https://es.wikipedia.org/wiki/A5/1 [Ultimo acceso 18/01/2022]

[39] Leander G. (et. al.), «New lightweight DES variants», Conference Paper, DOI: 10.1007/978-3-540-74619-5_13 · Source: DBLP, Marzo 2007

[40] «Data Encryption Standard» De Wikipedia, the free encyclopedia, Disponible en: https://es.wikipedia.org/wiki/Data_Encryption_Standard [Ultimo acceso 18/01/2022]

[41] Bogdanov A. (et. al.), «SPONGENT: A Lightweight Hash Function», Conference Paper, DOI: 10.1007/978-3-642-23951-9_21 · Source: DBLP, Septiembre 2011

[42] Hirose S., «An AES Based 256-bit Hash Function for Lightweight Applications: Lesamnta-LW», Paper Special Section on Cryptography and Information Security

[43] Simplício M. (et. al.), «The CURUPIRA-2 Block Cipher for Constrained Platforms: Specification and Benchmarking», Conference Paper,, Source: DBLP, Enero 2008

[44] Zhang W. (et. al.), «RECTANGLE: A Bit-slice Lightweight Block Cipher Suitable for Multiple Platforms», SCIENCE CHINA Information Sciences, Vol. 58: 122103(15), doi: 10.1007/s11432-015-5459-7, Diciembre 2015

[45] Shirai T. (et. al.), «The 128-bit Blockcipher CLEFIA (Extended Abstract)», Published in FSE 26, Marzo 2007

[46] Hell M. (et. al.), «Grain-128AEAD - A lightweight AEAD stream cipher», 2019

[47] Berbain C. (et. al.), «DECIM v2», Article DOI: 10.1007/978-3-540-68351-3_11 · Source: OAI, Junio 2008

[48] KDDI Corporation, «Stream Cipher KCipher-2», Document Version 1.2, 2017

[49] Watanabe D. (et. al.), «A New Keystream Generator MUGI», Conference Paper DOI: 10.1007/3-540-45661-9_14 · Source: DBLP, Julio 2002

[50] Boesgaard M. (et. al.), «The Stream Cipher Rabbit», DOI:10.1007/978-3-540-68351-3_7, In book: New Stream Cipher Designs (pp.69-83), June 2008

[51] Ekdahl P. y Johansson T., «A New Version of the Stream Cipher SNOW», Lecture Notes in Computer Science (Selected Areas in Cryptography. Revised Papers), 2003

[52] Pyrgas L. y Kitsos P., «Compact Hardware Architectures of Enocoro-128v2 Stream Cipher for Constrained Embedded Devices», MDPI - Electronics, 2020

[53] Steve Babbage S. y Dodd M., «The stream cipher MICKEY 2.0», Capítulo del libro "New Stream Cipher Designs", Volume 4986, ISBN : 978-3-540-68350-6, 2008

[54] Juhani M. y O. Saarinen O., «The BlueJay Ultra-Lightweight Hybrid Cryptosystem», DOI:10.1109/SPW.2012.11, Conference: Security and Privacy Workshops (SPW), 2012 IEEE, May 2012

[55] Peris-Lopez P. (et. al.), «LMAP: A Real Lightweight Mutual Authentication Protocol for Low-cost RFID tags», Enero 2006

[56] Kitsos P., «Hardware Implementations for the ISO/IEC 18033-4:2005 Standard for Stream Ciphers», Article in Signal Processing, January 2006

[57] «Teorema de muestreo de Nyquist-Shannon», De Wikipedia, the free encyclopedia, Disponible en: https://es.wikipedia.org/wiki/Teorema_de_muestreo_de_Nyquist-Shannon [Ultimo acceso 18/01/2022]

[58] «Un autobús urbano eléctrico que se recarga en tres minutos recorre las calles de Mánchester», Disponible en: https://www.esmartcity.es/2017/10/23/autobus-urbano-electrico-se-recarga-tres-minutos-recorre-calles-manchester [Ultimo acceso 18/01/2022]

[59] «El cifrado AES, ¿está roto o no?», Disponible en: https://www.globalgate.com.ar/novedades-el-cifrado-aes-esta-roto-o-no.html [Ultimo acceso 18/01/2022]

[60] «Merkle–Damgård construction», De Wikipedia, the free encyclopedia, Disponible en: https://en.wikipedia.org/wiki/Merkle%E2%80%93Damg%C3%A5rd_construction [Ultimo acceso 18/01/2022]

[61] «Dispositivos conectados (Internet de las cosas) a nivel mundial de 2018 a 2030», Stadista 2022, Disponible en https://es.statista.com/estadisticas/517654/prevision-de-la-evolucion-de-los-dispositivos-conectados-para-el-internet-de-las-cosas-en-el-mundo/, [Ultimo acceso 19/01/2022]

[62] Garcia M., «Implementación del algoritmo de cifrado AES para bajo consumo sobre FPGA», Universidad Carlos III de Madrid. Departamento de Tecnología Electrónica, Proyecto de fin de carrera, pagina 35

[63] «International Data Encryption Algorithm», De Wikipedia, the free encyclopedia, Disponible en: https://es.wikipedia.org/wiki/International_Data_Encryption_Algorithm [Ultimo acceso 19/01/2022]

[64] «Twofish», De Wikipedia, the free encyclopedia, Disponible en: https://es.wikipedia.org/wiki/Twofish [Ultimo acceso 19/01/2022]

[65] «Secure Hash Algorithm», De Wikipedia, the free encyclopedia, Disponible en: https://es.wikipedia.org/wiki/Secure_Hash_Algorithm [Ultimo acceso 19/01/2022]

[66] Suzaki T. (Et. al), «TWINE: A Lightweight, Versatile Block Cipher», Nec Corporation, Investigacion de criptografía simétrica, Disponible en https://www.nec.com/en/global/rd/tg/code/symenc/twine.html#top [Ultimo acceso 19/01/2022]

[67] Engels E. (Et. al), «The Hummingbird-2 Lightweight Authenticated Encryption Algorithm», Revere security, , Disponible en https://eprint.iacr.org/2011/126.pdf [Ultimo acceso

19/01/2022]

[68] «Trivium (cifrado)», De Wikipedia, la enciclopedia libre, Disponible en: https://en.wikipedia.org/wiki/Trivium_(cipher) [Ultimo acceso 19/01/2022]

[69] «Cifrado Chacha20», De Asecuritysite.com, Disponible en: https://asecuritysite.com/symmetric/chacha [Ultimo acceso 19/01/2022]

[70] Fan X. (et. al.), «WG-8: A Lightweight Stream Cipher for Resource-Constrained Smart Devices», Quality, Reliability, Security and Robustness in Heterogeneous Networks, 2013, Volume 115 ISBN : 978-3-642-37948-2

[71] Yao G., Parampalli U., «Generalized NLFSR Transformation Algorithms and Cryptanalysis of the Class of Espresso-like Stream Ciphers», Computer science Cryptography and Security Cornell University

[72] «Criptosistema Rabin», De Wikipedia, la enciclopedia libre, Disponible en: https://es.wikipedia.org/wiki/Criptosistema_Rabin [Ultimo acceso 19/01/2022]

[73] «ISO/IEC 29192-5», Hash-function, Online Browsing Platform (OBP), Disponible en: https://www.iso.org/obp/ui/#iso:std:iso-iec:29192:-5:ed-1:v1:en [Ultimo acceso 19/01/2022]

[74] Guo J. (et. al..), «The PHOTON Family of Lightweight Hash Functions», Disponible en: https://sites.google.com/site/photonhashfunction/design [Ultimo acceso 19/01/2022]

[75] Faugère J.C., «Analysis of the MQQ Public Key Cryptosystem», Conference Paper · December 2010 DOI: 10.1007/978-3-642-17619-7_13 · Source: DBLP

[76] Kodali R.K., «Implementation of ECDSA in WSN», International Conference on Control Communication and Computing (ICCC), 2013, pp. 310-314, doi: 10.1109/ICCC.2013.6731670

[77] Wolf C. y Preneel B., «Taxonomy of Public Key Schemes Based on the Problem of Multivariate Quadratic Equations», Cryptology ePrint Archive Report, 2007/077

ANEXO I: CLASE DE ETIQUETAS RFID

En total, las distintas clases de etiquetas hasta la fecha son:

• EPC Class 0 (UHF): etiquetas de sólo lectura programadas durante el proceso de fabricación. Rango de alcance de 3 metros. Son las de menor coste.

• EPC Class 1 (HF): etiquetas WORM (Write Once Read Many) con una mínima memoria para almacenar el código EPC. Rango de alcance de 3 metros. Coste muy bajo.

• EPC Class 1 (UHF): etiquetas WORM (Write Once Read Many) con una mínima memoria para almacenar el código EPC. Rango de alcance de 3 metros. Coste muy bajo

• EPC Class 2 (UHF): etiquetas pasivas programables con memoria de usuario, encriptación, etc. Rango de alcance de 3 metros. Coste bajo.

• EPC Class 3 (UHF): etiquetas semipasivas con memoria de usuario, encriptación, etc. Rango de alcance de 100 metros. Coste moderado.

• EPC Class 4 (UHF): etiquetas activas con memoria de usuario, encriptación, etc. Rango de alcance de 100 metros. Coste alto.

Clase de etiquetas RFID (tomada de [17])

ANEXO II: CLASIFICACION/TAXONOMIA DE LOS CIBERDELINCUENTES

Contenido abusivo	**SPAM**	Correo electrónico masivo no solicitado. El receptor del contenido no ha otorgado autorización válida para recibir un mensaje colectivo.
	Delito de odio	Contenido difamatorio o discriminatorio. Ej: ciberacoso, racismo, amenazas a una persona o dirigidas contra colectivos.
	Pornografía infantil, contenido sexual o violento inadecuado	Material que represente de manera visual contenido relacionado con pornografía infantil, apología de la violencia, etc.
	Sistema infectado	Sistema infectado con malware. Ej: Sistema, computadora o teléfono móvil infectado con un rootkit
Contenido dañino	**Servidor C&C (Mando y Control)**	Conexión con servidor de Mando y Control (C&C) mediante malware o sistemas infectados.
	Distribución de malware	Recurso usado para distribución de malware. Ej: recurso de una organización empleado para distribuir malware.
Obtención de información	**Configuración de malware**	Recurso que aloje ficheros de configuración de malware Ej: ataque de webinjects para troyano.
	Escaneo de redes (scanning)	Envío de peticiones a un sistema para descubrir posibles debilidades. Se incluyen también procesos de comprobación o testeo para recopilar información de alojamientos, servicios y cuentas. Ej: peticiones DNS, ICMP, SMTP, escaneo de puertos.
	Análisis de paquetes (sniffing)	Observación y grabación del tráfico de redes.
Intento de intrusión	**Ingeniería social**	Recopilación de información personal sin el uso de la tecnología. Ej: mentiras, trucos, sobornos, amenazas.
	Explotación de vulnerabilidades conocidas	Intento de compromiso de un sistema o de interrupción de un servicio mediante la explotación de vulnerabilidades con un identificador estandarizado (véase CVE). Ej: desbordamiento de buffer, puertas traseras, cross site scripting (XSS).
	Intento de acceso con vulneración de credenciales	Múltiples intentos de vulnerar credenciales. Ej: intentos de ruptura de contraseñas, ataque por fuerza bruta.
	Ataque desconocido	Ataque empleando exploit desconocido.
Intrusión	**Compromiso de cuenta con privilegios**	Compromiso de un sistema en el que el atacante ha adquirido privilegios.
	Compromiso de cuenta sin privilegios	Compromiso de un sistema empleando cuentas sin privilegios.
	Compromiso de aplicaciones	Compromiso de una aplicación mediante la explotación de vulnerabilidades de software. Ej: inyección SQL.
Disponibilidad	**Robo**	Intrusión física. Ej: acceso no autorizado a Centro de Proceso de Datos.
	DoS (Denegación de servicio)	Ataque de denegación de servicio. Ej: envío de peticiones a una aplicación web que provoca la interrupción o ralentización en la prestación del servicio.
	DDoS (Denegación distribuida de servicio)	Ataque de denegación distribuida de servicio. Ej: inundación de paquetes SYN, ataques de reflexión y amplificación utilizando servicios basados en UDP.
	Mala configuración	Configuración incorrecta del software que provoca problemas de

		disponibilidad en el servicio. Ej: Servidor DNS con el KSK de la zona raíz de DNSSEC obsoleto
	Sabotaje	Sabotaje físico. Ej: cortes de cableados de equipos o incendios provocados.
	Interrupciones	Interrupciones por causas ajenas. Ej: desastre natural.
	Acceso no autorizado a información	Acceso no autorizado a información. Ej: robo de credenciales de acceso mediante interceptación de tráfico o mediante el acceso a documentos físicos.
Compromiso de la información	**Modificación no autorizada de información**	Modificación no autorizada de información. Ej: modificación por un atacante empleando credenciales sustraídas de un sistema o aplicación o encriptado de datos mediante ransomware.
Fraude	**Pérdida de datos**	Pérdida de información Ej: pérdida por fallo de disco duro o robo físico.
	Uso no autorizado de recursos	Uso de recursos para propósitos inadecuados, incluyendo acciones con ánimo de lucro. Ej: uso de correo electrónico para participar en estafas piramidales.
	Derechos de autor	Ofrecimiento o instalación de software carente de licencia u otro material protegido por derechos de autor. Ej: Warez.
	Suplantación	Tipo de ataque en el que una entidad suplanta a otra para obtener beneficios ilegítimos.
Vulnerable	**Phishing**	Suplantación de otra entidad con la finalidad de convencer al usuario para que revele sus credenciales privadas.
	Criptografía débil	Servicios accesibles públicamente que puedan presentar criptografía débil. Ej: servidores web susceptibles de ataques POODLE/FREAK.
	Amplificador DDoS	Servicios accesibles públicamente que puedan ser empleados para la reflexión o amplificación de ataques DDoS. Ej: DNS open-resolvers o Servidores NTP con monitorización monlist.
	Servicios con acceso potencial no deseado	Ej: Telnet, RDP o VNC.
	Revelación de información	Acceso público a servicios en los que potencialmente pueda relevarse información sensible. Ej: SNMP o Redis.
	Sistema vulnerable	Sistema vulnerable. Ej: mala configuración de proxy en cliente (WPAD), versiones desfasadas de sistema.
	Otros	Todo aquel incidente que no tenga cabida en ninguna categoría anterior.
Otros	**APT**	Ataques dirigidos contra organizaciones concretas, sustentados en mecanismos muy sofisticados de ocultación, anonimato y persistencia. Esta amenaza habitualmente emplea técnicas de ingeniería social para conseguir sus objetivos junto con el uso de procedimientos de ataque conocidos o genuinos.

Tabla 1.- Clasificación de los Ciberincidentes (tomada de [26])

ANEXO III: ANÁLISIS DEL RENDIMIENTO DE LOS ALGORITMOS DE CRIPTOGRAFÍA LIGERA

Algorithm	Key Size	Block Size	Tech (μm)	Area (GEs)	Power (μW)	Energy (μJ/bit)	Throughput @100KHz (Kbps)	Hardware Effi-ciency (Kbps/KGE)	Key Size	Block Size	ROM (byte)	RAM (byte)	Latency (Cy-cles/ block)	Energy (μJ/bit)	Throughput @4MHz (Kbps)	Software Effi-ciency (Kbps/KB)
AES*	128	128	0.13	2400	2.4	42.38	56.64	23.6	128	128	918	0	4192	16.7	122	132.9
PRESENT*	80	64	0.18	1570	2.35	11.77	200	127.38	128	64	660	0	10792	43.1	23.7	35.91
RECTANGLE	80	64	0.13	1467	1.46	5.96	246	167.68	-	-	-	-	-	-	-	-
MIDORI	128	64	0.09	1542	60.6	1.61	400	259.4	-	-	-	-	-	-	-	-
mCrypton*	128	64	0.35	2594	4.66	138.61	33.51	12.91	96	64	1076	28	16457	68	15.5	14.41
NOEKEON*	128	128	0.35	2604	4.68	1362.21	3.44	1.32	128	128	364	32	23517	95.9	21.7	59.62
ICEBERG	128	64	0.18	5817	8.72	21.81	400	68.76	-	-	-	-	-	-	-	-
PUFFIN-2	80	64	0.18	1083	1.62	314.74	5.2	4.8	-	-	-	-	-	-	-	-
PRINCE*	128	64	0.13	2953	2.95	5.53	533.3	180.59	128	64	1108	0	3614	14.4	70.8	63.9
PRIDE*	-	-	-	-	-	-	-	-	128	64	266	0	1514	6	169	635.34
PRINT^	80	48	0.18	503	0.75	7.54	100	198.8	80	48	6210	48	87272	117.8	2.2	0.35
Klein*	64	64	0.18	1220	1.83	59.18	30.9	25.32	64	64	2980	50	7901	10.6	32.4	10.87
LED#	64	64	0.18	966	1.45	282.55	5.1	5.27	80	64	2164	368	35161	-	7.28	3.36
I-PRESENT	80	64	0.18	2467	370	-	-	-	-	-	-	-	-	-	-	-
EPCBC	96	48	0.18	1008	1.51	124.74	12.12	12.02	-	-	-	-	-	-	-	-
DESL*	56	64	0.18	1848	2.77	62.37	44.4	24.02	56	64	3098	0	8365	33.4	30.6	9.88
TEA*	128	64	0.18	2355	3.53	35.32	100	42.46	128	64	648	24	7408	30.3	34.5	53.24
XTEA*	-	-	-	-	-	-	-	-	128	64	504	0	17514	70	14.6	28.97
Camellia*	128	128	0.18	6511	9.76	33.57	290.1	44.55	128	128	1262	12	64000	256	8	6.34
SIMON*	96	48	0.13	763	0.76	48.32	15.8	20.7	96	48	170	0	594	2.3	323	1900
SEA*	96	8	0.13	2562	2.56	1117.67	2.29	0.89	96	96	426	24	41604	173.7	9.2	21.6
KASUMI*	128	64	0.13	3437	3.44	29.9	115.14	33.5	128	64	1264	24	11939	47.6	21.4	16.93
MIBS^	64	64	0.18	1396	2.09	10.47	200	143.26	64	64	3184	29	49056	66.2	5.2	1.63
LBlock^	80	64	0.18	1320	2	9.9	200	151.51	80	64	976	58	18988	25.6	13.48	13.81
ITUbee*	-	-	-	-	-	-	-	-	80	80	716	0	2607	10.4	122.7	171.37
GOST^	256	64	0.18	1000	1.5	7.5	200	200	256	64	4748	190	10240	13.8	25	5.27
Robin^	-	-	-	-	-	-	-	-	128	128	1942	80	4935	6.6	103.74	53.42
Fantomas^	-	-	-	-	-	-	-	-	128	128	1920	78	3646	4.9	140.42	73.14
CLEFIA*	128	128	0.13	2678	2.67	36.82	76	28.37	128	128	3046	0	28648	114.5	17.8	5.84
PICCOLO^	80	64	0.13	1136	1.13	4.8	237.04	208.66	80	64	966	70	21448	28.9	11.93	12.35
TWINE#	80	64	0.09	1503	1.05	5.91	178	118.42	80	64	1180	140	20505	-	12.48	10.58
SPECK*	96	48	0.13	884	0.88	73.67	12	13.57	96	48	134	0	408	1.6	470.5	3511.19
IDEA*	-	-	-	-	-	-	-	-	128	64	596	0	2700	10.8	94.8	159.06
HIGHT*	128	64	0.35	2608	4.7	24.93	188	72.08	128	64	5718	47	6377	25.5	40.14	7.02
LEA#	128	128	0.13	3826	3.82	50.22	76.19	19.91	128	128	590	32	5231	-	97.8	165.76
KATAN*	80	32	0.13	802	0.8	64.16	12.5	15.58	80	64	338	18	72063	289.2	3.5	10.35
KTANTAN^	80	32	0.13	462	0.46	36.96	12.5	27.05	80	32	10516	614	10233211	13814.8	0.012	0
Hummingbird^	-	-	-	-	-	-	-	-	128	16	1822	82	4637	6.2	13.8	7.57
Hummingbird-2^	128	16	0.18	2159	3.23	40.48	80	37.05	128	16	770	50	1520	2	42.1	54.68

* 8-bit microcontroller, ^16-bit microcontroller, # 32-bit microcontroller

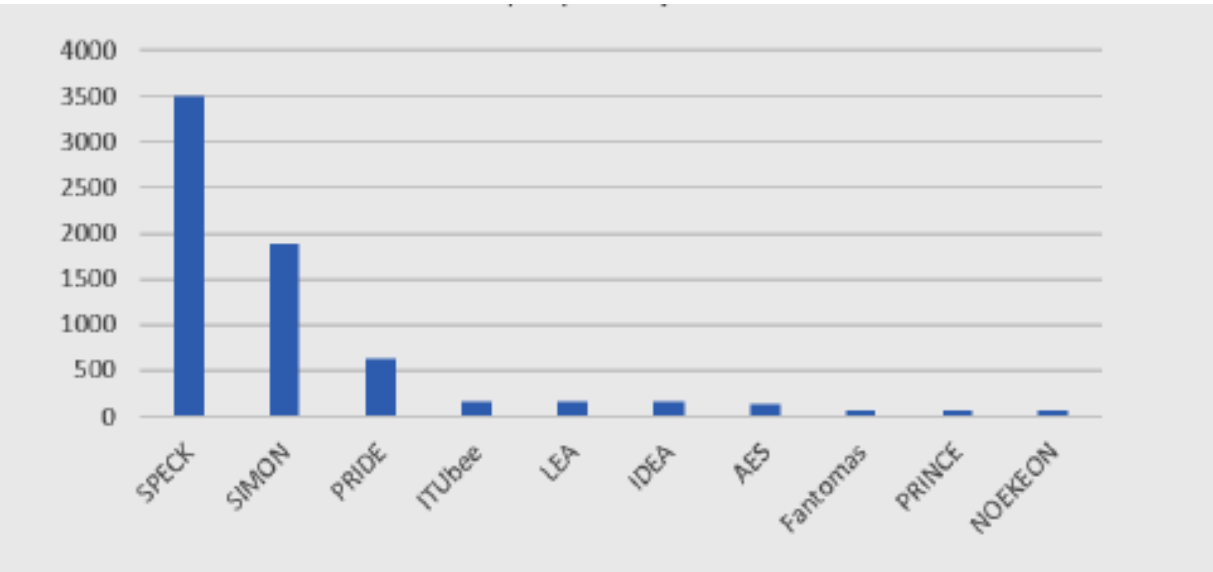
Algoritmos ligeros (tomada de [1])

ANEXO IV: ANÁLISIS DE LA SEGURIDAD DE LOS ALGORITMOS DE CRIPTOGRAFÍA LIGERA

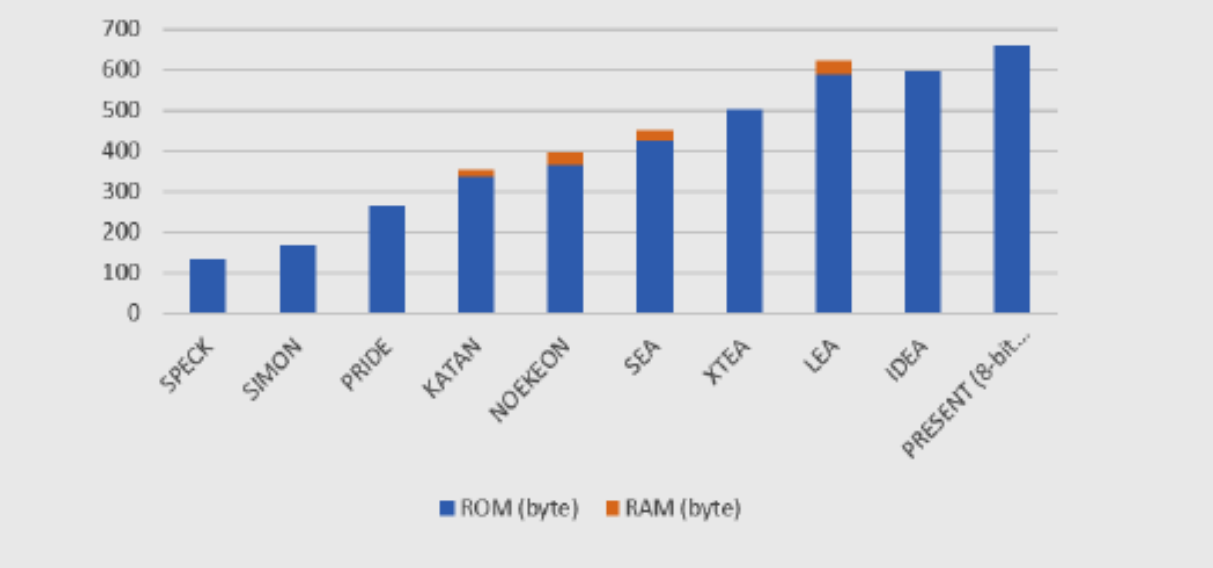
LWC Algorithm	Differential Cryptanalysis*	Linear Crypt-analysis	Integral/Square/ Saturation Cryptanalysis	Algebric/Cube Cryptanalysis	MITM/ Biclique	Related Key attack	Side-Channel/ Differential fault attacks
AES	✓ [134]	–	–	–	✓ [50]	✓ [50]	✓ [135] [136] [137]
PRESENT	✓ [138] [139]	-	-	-	✓ [28]	✓ [140]	✓ [135] [137] [141] [142]
GIFT	✓ [143] [56]	-	✓ [144]	-	✓ [54] [144] [57]	✓ [145] [56] [146]	✓ [147]
SKINNY	✓ [148]	-	-	-	-	✓ [146]	✓ [137] [149] [150]
RECTANGLE	-	-	✓ [53]	-	-	✓ [53]	✓ [53]
MIDORI	✓ [48]	✓ [48]	–	–	✓ [48]	–	–
mCrypton	-	-	-	-	-	✓ [151]	-
NOEKEON	-	-	-	-	-	✓ [65]	-
ICEBERG	✓ [152]	-	-	-	-	-	-
PUFFIN-2	✓ [153]	-	-	-	-	-	-
PRINCE	✓ [154]	–	–	–	–	–	–
PRINT	-	-	-	-	-	✓ [155]	-
Klein	-	-	-	-	✓ [156]	✓ [157]	✓ [158]
LED	✓ [159]	-	-	-	✓ [28]	✓ [77]	-
EPCBC	-	-	-	✓ [23]	-	✓ [23]	-
TEA	–	–	–	–	–	✓ [88] [89]	–
XTEA	-	-	-	-	-	✓ [91]	-
XXTEA	✓ [160]	–	–	–	–	–	–
Camellia	✓ [94]	-	-	-	-	-	✓ [24] [136]
SIMON	✓ [29] [161]	-	-	✓ [30]	-	✓ [30]	-
KASUMI	✓ [17]	–	–	–	–	✓ [18]	–
MIBS	✓ [162]	✓ [162]	-	-	-	-	-
LBlock	✓ [25] [26]	–	✓ [60]	–	✓ [163]	–	–
ITUbee	-	-	-	-	-	✓ [164]	-
GOST	-	-	-	-	✓ [165]	✓ [166]	-
CLEFIA	-	-	✓ [107]	-	-	✓ [107]	✓ [136]
PICCOLO	✓ [167]	-	-	-	✓ [28] [168]	-	-
TWIS	✓ [112]	–	–	–	–	–	–
TWINE	-	-	✓ [84]	-	✓ [169]	-	-
SPECK	✓ [161] [131]	-	-	-	-	✓ [131]	-
IDEA	-	-	-	-	✓ [16]	-	-
HIGHT	✓ [140]	✓ [15]	-	-	✓ [168]	✓ [170]	-
LEA	–	–	–	–	–	–	✓ [27]
KeeLoq	-	✓ [123]	-	✓ [122]	✓ [171]	-	✓ [123]
KATAN	-	-	-	-	✓ [172]	-	-
KTANTAN	-	-	-	-	-	✓ [19]	-
Hummingbird-2	-	-	-	-	-	✓ [173]	-
Hummingbird	Vulnerable to several attacks [127]						

Seguridad algoritmos ligeros (tomada de [1])

ANEXO V: TOP 10 ALGORITMOS DE CRIPTOGRAFÍA LIGERA



Eficiencia en Sofware (tomada de [1])



Eficiciencia en memoria (tomada de [1])