Facultad de Ciencias

Estudio de la seguridad en los dispositivos IoT (Internet of Things)

(Study of the security in IoT devices)

Trabajo de Fin de Grado
para acceder al

Grado en ingeniería informática

Autor: Mario Abad Diaz

Director: José Ángel Irastorza Teja

Co-Director: Alberto Eloy Garcia Gutierrez

Julio - 2021

RESUMEN

En este documento se presenta un estudio de la seguridad en los dispositivos IoT (Internet of Things).

El internet de las cosas (IoT) está impactando en la forma en la que la sociedad interactúa con el mundo que le rodea. Cada vez más y más "cosas" alrededor de las personas se comunican entre ellas, desde televisores, lavadoras y vehículos hasta monitores de la salud y maquinaria de producción. El IoT promete una gran cantidad de beneficios a primera vista. Sin embargo, es primordial la obtención de la confianza del consumidor en la privacidad y en la seguridad de los dispositivos IoT para que estos puedan alcanzar su máximo potencial. Los datos que manejan los dispositivos IoT contienen información de cada uno de sus usuarios. El gran desafío es proteger esa información.

Para ello es necesario un estudio y diseño de una posible solución que permita securizar la información que envían a las plataformas de gestión teniendo en cuenta todas las características particulares de este tipo de dispositivos

Palabras clave – IoT, seguridad, WSN, criptografía.

ABSTRACT

The aim of this document is to present a study about the security of IoT (Internet of Things) devices.

Internet of things is changing the way that society interacts with the surrounding world. More and more "things" around people communicate between themselves, from tv, washing machines or vehicles to health monitoring systems or production machines. IoT promises a great amount of benefits at a first glance. Nevertheless, obtaining the consumer trust on privacy and security of the IoT devices is a primoridial necessity before the iot can reach its maximum exponential. The data that IoT devices handle contain information from each user. The great challenge is to protect that information.

In order to overcome this problem it is needed a study and a design of a possible solution that allow a securization of the information that is sent to the management platforms taking into account all the considerations that this devices require.

Keywords – IoT, security, WSN, criptography.

ÍNDICE

Resumen **3**

Abstract **3**

Índice de Figuras **6**

Índice de Tablas **7**

1. Preámbulo **9**

1.1 Introducción 9

1.2 Motivación y objetivos 10

1.3 Estructura del documento 10

2. Seguridad en el mundo IoT **13**

2.1 Dispositivos IoT en el mercado 13

2.2 Redes de sensores (WSN) 15

2.2.1 Características principales de las redes de sensores 16

2.3 Consideraciones en las redes de sensores WSN 17

2.4 Vulnerabilidades y amenazas de seguridad en las redes de sensores WSN 19

2.4.1 Ataques pasivos 19

2.4.1.1 Eavesdropping 19

2.4.1.2 Destrucción de nodos 20

2.4.1.3 Disfunción de nodos 20

2.4.1.4 Análisis de tráfico 20

2.4.2 Ataques activos 20

2.4.2.1 Ataques en la capa física 20

2.4.2.2 Ataques en la capa de enlace de datos 21

2.4.2.3 Ataques en la capa de red 22

2.4.2.4 Ataques en la capa de transporte 23

2.4.2.5 Ataques en la capa de aplicación 23

2.5 Defensas ante los ataques planteados 24

3. Soluciones basadas en hardware criptográfico **27**

3.1 Características generales que buscar en las soluciones 27

3.1.1 Autenticación 27

3.1.2 Resistencia al tampering 28

3.1.3 Criptografía 28

3.1.3.1 Llaves simétricas 28

3.1.3.2 Llaves asimétricas 29

3.1.3.3 Funciones hash 30

3.2 Solución empresarial 31
3.2.1 Extreme Defender for IoT 32
3.2.2 Adaptador SA201 33
3.2.3 AP150W 34
3.2.4 Conclusión de Extreme Defender 35
3.3 Solución con modulo TPM 36
3.3.1 IRIDIUM9670 TPM2.0 LINUX 37
3.3.2 Conclusión módulos TPM 38
3.4 Solución con chip criptográfico 38
3.4.1 ATSHA204A 39
3.4.2 ATAES132A 40
3.4.3 ATECC608B 41
3.4.4 Conclusión hardware criptográfico 42
3.5 Comparación de soluciones 43
4. Propuesta de solución basada en chip criptográfico 45
4.1 Materiales hardware 45
4.1.1 Selección de chip 45
4.1.2 Selección de plataforma 46
4.1.3 Selección del sensor 47
4.1.4 Otros materiales hardware 47
4.2 Materiales software 47
4.2.1 Librería core Arduino para ESP8266 48
4.2.2 Librería del sensor DHT11 48
4.2.3 Librería de Wifi 48
4.2.4 Librería del chip criptográfico 49
4.3 Conexión de la placa al sensor y al chip criptográfico 49
4.3.1 Conexión chip criptográfico-placa 49
4.3.2 Conexión sensor de temperatura-placa 50
4.4 Funcionamiento 50
4.5 Presupuesto 51
5. Conclusión y líneas futuras 53
Bibliografía 54

ÍNDICE DE FIGURAS

Figura 1 – Número total de conexiones de dispositivos 10
Figura 2 - Dispositivos IoT en el mercado 13
Figura 3 – Estructura de un sensor 14
Figura 4 – Tabla comparativa de Arduino y Raspberry Pi 14
Figura 5 – Estructura red de sensores 15
Figura 6 – Direccionamiento multisalto 17
Figura 7 – Tipos de esquema de comunicación WSN 18
Figura 8 – Tipos de amenazas de seguridad 19
Figura 9 – Tipos de autenticación 28
Figura 10 – Cifrado usando llave simétrica 29
Figura 11 – Algoritmo Diffie-Hellman 30
Figura 12 – Asignación de llaves e intervalos temporales 31
Figura 13 – Esquema de la solución Extreme Defender for IoT 31
Figura 14 – Menú Dashboard 32
Figura 15 – Adaptador SA201 33
Figura 16 – Esquema de conexión SA201 33
Figura 17 – AP150W 34
Figura 18 – Esquema de conexión AP150W 35
Figura 19 – Raspberry Pi con el módulo Iridium9670 TPM2.0 LINUX 37
Figura 20 – Mapeo de pines GPIO 38
Figura 21 – ATSHA204A 39
Figura 22 – Esquema de pines I2C 40
Figura 23 – Esquema de pines I2C + SPI 41
Figura 24 – Comandos disponibles ATECC608B 45
Figura 25 – NodeMCU v3 46
Figura 26 – Sensor DHT11 47
Figura 27 – Instalación de la librería ESP8266 47
Figura 28 – Funciones de la librería ECCX08 48
Figura 29 – Esquema de conexiones 49

ÍNDICE DE TABLAS

Tabla 1 – Defensas ante ataques pasivos 24
Tabla 2 – Defensas ante ataques activos 25
Tabla 3 – Comparación de soluciones 43
Tabla 4 – Presupuesto hardware 51

1. PREÁMBULO

1.1 Introducción

Los dispositivos IoT son piezas hardware programadas para ejecutar una cierta aplicación y transmitir datos a través de una red de forma inalámbrica. Hoy en día se pueden encontrar este tipo de dispositivos allá donde se mire, desde vehículos o casas inteligentes hasta dispositivos de cuidado de la salud. Su crecimiento es cada vez mayor. Según el proveedor de conocimientos de mercado iot analytics [1], el número de conexiones de dispositivos en internet en 2020 es mayor desde dispositivos iot que desde dispositivos no iot, como se puede ver en la *Figura 1*. En temas de dinero, Statista [2] calcula que en 2020 el valor del mercado IoT es de 389 billones de dólares y prevé que para 2030 esta cifra habrá crecido más de un trillón de dólares al igual que prevé un crecimiento en el número de dispositivos IoT que triplica al número de dispositivos IoT conectados actualmente.

Los dispositivos IoT se encargan de generar, transmitir y manejar una gran cantidad de datos. Estos datos pueden contener desde información privada de usuarios hasta ordenes de ejecución remota. Son una información en muchos casos critica, ya que de ella pueden depender dispositivos que, en caso de mal funcionamiento, pueden incluso poner en peligro la vida de sus usuarios. Esto y otros motivos no mencionados hacen que la seguridad de los datos y de los dispositivos iot sea vital para su correcto funcionamiento.

La seguridad en los dispositivos IoT es la familia de técnicas, estrategias y herramientas para proteger a los dispositivos de ser comprometidos. La mayoría de los ataques que han tenido los dispositivos IoT están relacionados con el hecho de que están conectados unos con otros de forma inalámbrica. Esta conectividad permite a los atacantes interactuar con los dispositivos de forma remota abriendo más vulnerabilidades de las normales. A la vez que esto ocurre, industrias como la médica o la automovilística se están expandiendo en el territorio de los dispositivos IoT, haciéndose cada vez más dependientes de estos dispositivos. Esto normalmente no supondría ningún problema. Sin embargo, en esta caso, los dispositivos IoT acarrean más vulnerabilidades que un dispositivo normal. Esto combinado con el hecho de que muchas veces la industria no piensa en el coste de securizar estos dispositivos, hace que se queden abiertos muchos más frentes de ataque.

Los dispositivos IoT ya han sido protagonistas de varios ataques a gran escala. Un ataque [3] que se ha producido recientemente ha resultado en 150.000 cámaras de vigilancia comprometidas, estando algunas de estas cámaras en espacios seguros de compañías como Tesla. Pero sin duda el mayor ataque que se ha realizado hacia los dispositivos IoT hasta la fecha es el llamado Mirai. Mirai es un malware que infecta mayormente dispositivos IoT, para ello se intenta conectar a los dispositivos utilizando las credenciales que pone por defecto el fabricante. Este virus ha formado una botnet, llegando a contar con 600.000 dispositivos [4] con la que ha realizado varios ataques de denegación de servicio (DDoS Distributed Denial of Service). A dia de hoy esta botnet permanece en activo con un numero de dispositivos desconocido.

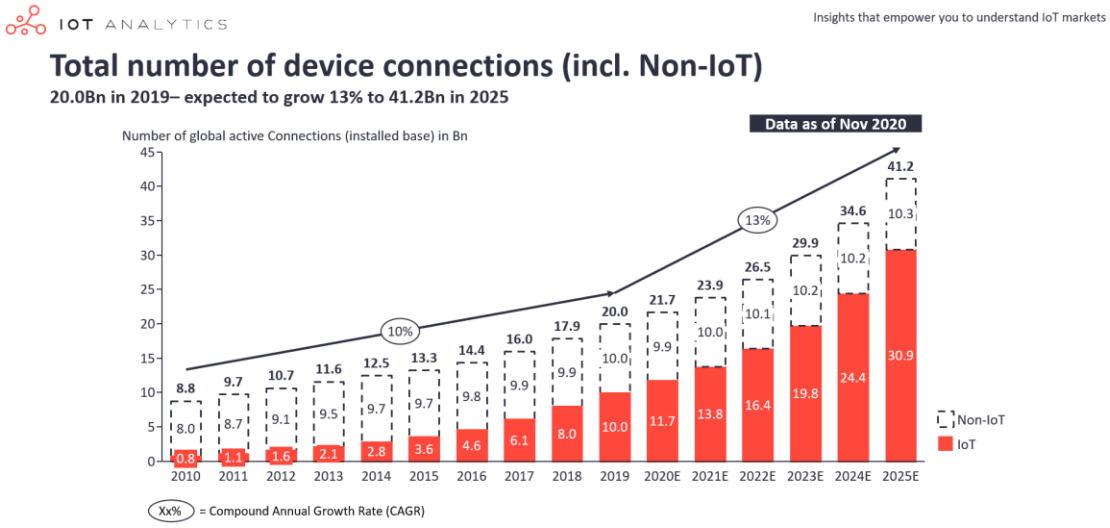


Figura 1 – Número total de conexiones de dispositivos Fuente: **[5]**

1.2 Motivación y objetivos

El proyecto tiene como objetivo principal encontrar una solución a los problemas de seguridad que se pueden encontrar en los dispositivos IoT de hoy en día.

Los dispositivos IoT se encuentran en una fase de expansión masiva en la que están entrando en todo tipo de campos. Muchas empresas están digitalizando a marchas forzadas sus procesos, convirtiéndose cada vez más dependientes de la tecnología. Sin embargo, la seguridad no está siendo siempre tomada en consideración. Como siempre ha ocurrido, las empresas fijan la vista en que los dispositivos cumplan con la función principal con la que son asignados, pero se olvidan de que la seguridad es una parte fundamental para alcanzar un correcto funcionamiento. Esta falta de previsión está causando brechas en la seguridad de empresas en múltiples sectores, desde productos de consumidor hasta en la sanidad.

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

1) Visión de la importancia de la seguridad en los dispositivos IoT

2) Análisis de las redes de sensores

3) Análisis de vulnerabilidades en los dispositivos IoT en las redes de sensores

4) Análisis de las características a buscar en las soluciones

5) Búsqueda de soluciones basadas en hardware criptográfico

6) Propuesta de una solución basada en hardware criptográfico

1.3 Estructura del documento

La redacción del proyecto se ha dividido en cuatro capítulos los cuales se muestran a continuación:

Capítulo 1: Preámbulo. En el presente capítulo se realiza una introducción que describe la situación actual del sector tanto como las motivaciones que han propiciado la realización del proyecto y los objetivos que persigue

Capítulo 2: Seguridad en el mundo IoT. Se presenta la situación actual de los dispositivos IoT en el mercado, se presentan las redes de sensores, sus características, vulnerabilidades y soluciones propuestas por el sector.

Capítulo 3: Soluciones basadas en hardware criptográfico. Se proponen tres soluciones distintas, una propuesta de solución empresarial, una propuesta de solución basada en módulos TPM y una propuesta de solución basada en chips criptográficos. Se concluye haciendo una comparación de las tres.

Capítulo 4: Propuesta de solución basada en chip criptográfico. Se hace una propuesta de una implementación utilizando el chip criptográfico ATECC608B junto con una placa NodeMCU y un sensor de temperatura DHT11.

Capítulo 5: Conclusiones y líneas futuras. Se hace un balance de las soluciones propuestas y se reflexiona a cerca de la dirección en la que avanzaran las soluciones en el futuro.

2. SEGURIDAD EN EL MUNDO IOT

Este capítulo contiene un análisis general de la variedad de dispositivos IoT que hay en el mercado. Se centra en las redes de sensores (WSN Wireless Sensor Network), se hace un análisis de sus vulnerabilidades y de los requerimientos y limitaciones que tienen este tipo de dispositivos. A continuación, se expondrán soluciones propuestas por el sector para resolver estos problemas de seguridad.

2.1 Dispositivos IoT en el mercado

Los dispositivos IoT pueden ser cualquier objeto que se pueda conectar a la red. Actualmente se tiende a intentar conectar toda clase de objetos, que históricamente nunca han utilizado internet, esto son los dispositivos IoT. Un dispositivo IoT puede ser desde una bombilla inteligente hasta el juguete de un niño que es capaz de conectarse a la red. Los dispositivos como los ordenadores, que siempre han estado conectados a la red, también se pueden considerar dispositivos IoT.

Dentro del mundo IoT hay una infinidad de variantes de uso de estos dispositivos. Los dispositivos IoT se utilizan en una gran variedad de campos, en la *Figura 2* se pueden ver algunos ejemplos. Un ejemplo muy claro de la variedad de usos de los dispositivos IoT se puede encontrar en la industria de la automoción. Aquí se pueden encontrar dispositivos de varios tipos como el navegador, que actualmente en la mayoría de los coches está conectado a internet, sensores de aparcamiento o sensores de presión de las ruedas que se conectan a la centralita del coche de forma inalámbrica. Otro claro ejemplo es el de las ciudades inteligentes donde se utilizan sensores lumínicos para el encendido inteligente de las farolas o sensores de humedad para el encendido inteligente de los aspersores.

Como se puede ver, los sensores son sin ningún tipo de duda el uso de dispositivos IoT más extendido. Esto se debe a que en la gran mayoría de aplicaciones de los dispositivos IoT se incluyen sensores que se utilizan para conocer el entorno. Esto se debe a que saber las condiciones que rodean a un sistema abre un gran abanico de funcionalidades, además de poder mejorar el funcionamiento de los sistemas. Dentro del campo de los sensores hay un gran abanico de tipos [6] para medir cualquier clase de evento deseado. Es muy común que se utilicen varios sensores a la vez para obtener una información lo más fiable posible. Por ello la forma más común de despliegue de los sensores es en redes de sensores WSN. El amplio uso de los sensores ha causado una caida en el precio de los mismos [7].

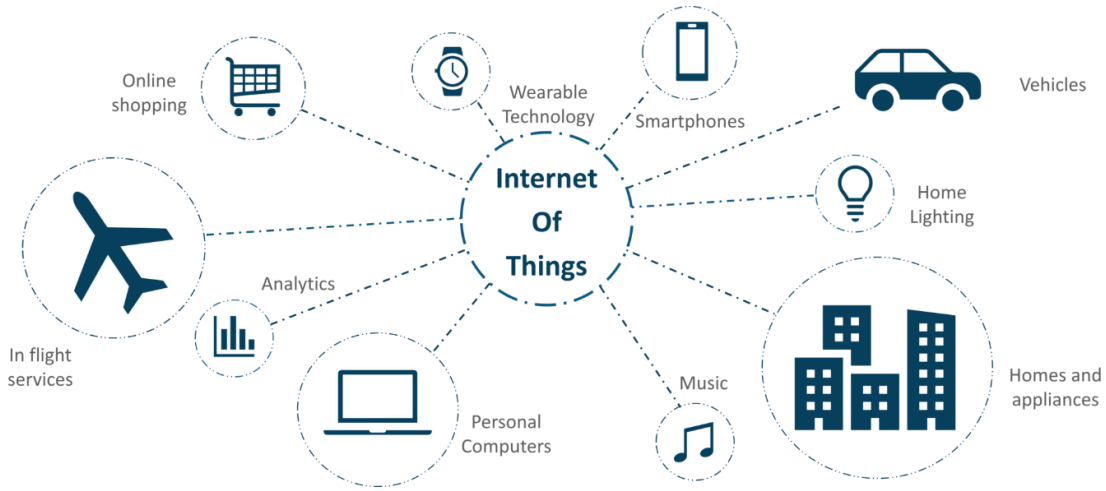


Figura 2 - Dispositivos IoT en el mercado Fuente: **[38]**

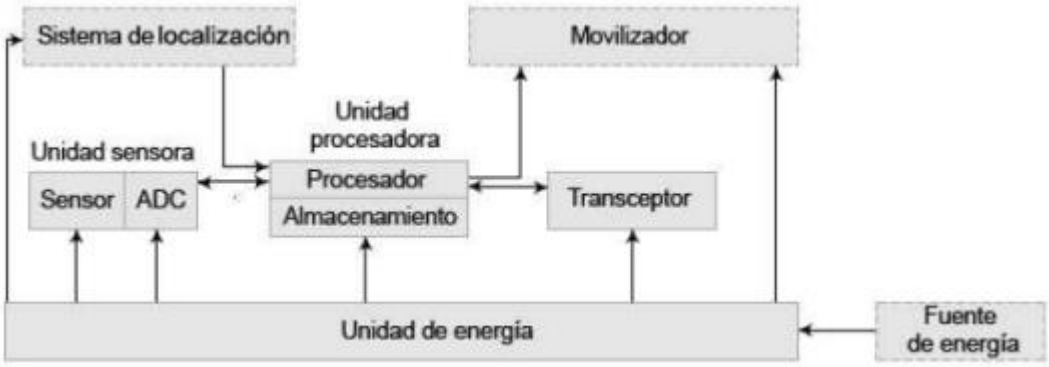


Figura 3 – Estructura de un sensor Fuente: **[39]**

La estructura de un sensor, como se puede ver en la *Figura 3*, consta de cuatro componentes básicos. El primer componente es una unidad sensora que se encarga de tomar las mediciones. Esta unidad puede estar destinada a medir muchos tipos de eventos distintos como temperatura, humedad, luz y movimiento entre otros. El segundo componente es el procesador, este componente es el cerebro del sensor, se encarga de tomar las mediciones utilizando la unidad sensora, procesarlas y enviarlas a través de el transceptor. El tercer componente es un transceptor que se encarga de enviar los datos desde el sensor a un nodo administrador, en el caso de las redes WSN. Puede utilizar varias tecnologias de comunicación distintas como Wifi, Bluetooth o Zigbee. Por ultimo, el cuarto componente basico es la unidad de energía que es la encargada de almacenar y suministrar energía al resto de componentes. Adicionalmente, como se ve en la *Figura 3* pueden haber otros componentes dependiendo de la aplicación que se quiera que tenga el sensor.

La seguridad de los sensores se debe de implementar en las comunicaciones ya que este es el punto mas debil de los sensores. La comunicación entre el transceptor del sensor y el nodo administrador que reciba los datos debe de ser securizada ya que es un gran agujero en la seguridad.

En el mercado existen sensores IoT propietarios de marca que cuentan con una funcionalidad completa integrada en el sensor, sin embargo estos dispositivos no son los mas extendidos. Ademas los sensores

Raspberry Pi	Arduino
Es un mini PC que puede ejecutar múltiples programas al mismo tiempo	Es un micro controlador, parte de un ordenador, que ejecuta un único programa una y otra vez.
Es complicado hacerlo funcionar con batería.	Está pensado para funcionar con batería.
Requiere tareas complejas como instalar librerías y software para interactuar con sensores y otros componentes.	Sus componentes y sensores funcionan de manera integrada.
Es caro en relación a Arduino.	Es barato.
Se conecta fácilmente a Internet con su puerto RJ-45 o con WiFi por USB.	Requiere hardware externo para conectarse a Internet y hay que programarlo utilizando código para que funcione. No está pensado para conectar a Internet.
No tiene almacenamiento, pero puede usar su ranura micro SD para ello.	Puede venir con almacenamiento integrado.
Tiene 4 puertos USB para conectar distintos dispositivos.	Solo tiene un puerto USB Type-B hembra para conectarlo a un PC.
Utiliza procesadores ARM.	Utiliza un procesador de familia AVR.
Debemos apagarlo correctamente para que no haya riesgo de corrupción de archivos.	Es un dispositivo plug and play.
El lenguaje de programación recomendado es Python, pero puede usar C, C++ y Ruby también.	Solo utiliza Arduino y C/C++.

Figura 4 – Tabla comparativa de Arduino y Raspberry Pi Fuente: **[40]**

propietarios no suelen contar con formas de securizar las comunicaciones. Las formas mas comunes en las que se encuentran los sistemas de sensores son utilizando las placas Arduino y las placas Raspberry Pi. Esto se debe a que son placas de bajo coste y muy polivalentes que permiten conectar todo tipo de sensores sin tener que fabricar un chip especializado. Estas placas realizan la funcionalidad del procesador, adicionalmente pueden contar con un transceptor integrado. Al ser placas modulares se pueden personalizar para utilizar la tecnologia de red deseada, pudiendo escoger entre Wifi, Bluetooth, Zigbee, cableado u otras tecnologias. Esta modularidad tambien permite la adicion de elementos de seguridad en las comunicaciones.

Dentro del mundo de las placas de desarrollo, las mas comunes son las mencionadas Arduino y Raspberry Pi. Estas placas son utilizadas actualmetne para todo tipo de propositos debido a su polivalencia. Estas marcas a su vez tienen una gran variedad de placas con distintas especificaciones que se ajustan a las caracteristicas computacionales necesarias de la aplicación deseada.

Arduino y Raspberry son dos plataformas muy distintas entre sí. Raspberry Pi es un ordenador completo de baja potencia mientras que Arduino es una plataforma de creacion de codigo abierto basada en hardware y software libre. La principal diferencia es la potencia de computo que ofrecen, siendo la de la Raspberry Pi superior. Esta potencia de computo tambien conlleva un mayor gasto energetico lo que la hace menos adecuada para implementarse como un sensor. Mientras tanto Arduino al ser una plataforma hardware libre cuenta con numerosas implementaciones que se adaptan a cualquier circunstancia. En la *Figura 4* se puede ver una tabla comparativa de las caracteristicas principales de las dos plataformas.

2.2 Redes de sensores (WSN)

Las redes de sensores están muy presentes en el mundo actual. Las industrias las utilizan para monitorizar todos sus espacios, máquinas de producción. Hoy en día incluso se utilizan para monitorizar pacientes en los hospitales.

Los avances tecnológicos de la comunicación inalámbrica han causado que las redes de sensores (WSN) últimamente hayan atraído más atención. Una red de sensores es un conjunto de sensores conectados entre ellos y con un nodo administrador. Estas redes consisten en sensores comunes capaces de ser organizados automáticamente por si mismos de forma autónoma, nodos administradores y un centro de datos como back-end.

El procedimiento de recolección de datos se puede ver en la *Figura 5* y es el siguiente: primero los nodos sensores son los responsables de transmitir datos en tiempo real de su entorno a los nodos

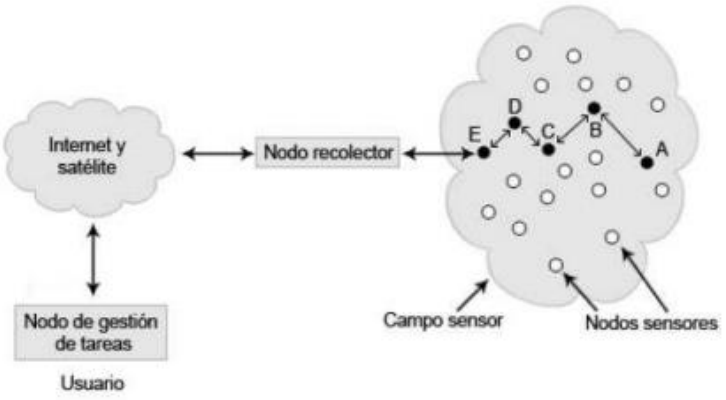


Figura 5 – Estructura red de sensores Fuente: **[39]**

administradores/recolectores. A continuación, los nodos administradores transmitirán esos datos al back-end que puede ser un usuario o un centro de datos. Finalmente, el back-end se encarga de hacer el procesamiento de los datos recibidos. Para todas estas comunicaciones se utilizan protocolos de comunicación inalámbrica tales como Wifi, Zigbee o Bluetooth. De hecho, debido a su propiedad de autoorganización y al no contar con una infraestructura fija, la topología de red cambia de forma dinámica haciendo que el broadcast sea una forma de comunicación muy apropiada.

Las redes de sensores inalámbricas han sido ya utilizadas para múltiples propósitos como uso militar de detección o detección de fuegos forestales. A pesar de ser ampliamente utilizados siguen siendo fácilmente atacadas debido a que utilizan comunicación inalámbrica en modo broadcast y que son físicamente atacables. De esta forma un atacante puede utilizar técnicas de eavesdropping, inyección de paquetes maliciosos o ataques a nodos físicamente.

Generalmente los nodos sensores se centran en proteger la privacidad y la autenticación de nodos. Con la privacidad se busca una confidencialidad total del contenido de los paquetes, permitiendo una comunicación entre nodos sensores y nodos administrador segura. 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.

2.2.1 Características principales de las redes de sensores

En comparación con las formas de comunicación en redes tradicionales, las redes de sensores tienen algunas características que se deben de tener en cuenta cuando se trata con ellas:

- **Son arquitecturas no centralizadas.** En las redes de sensores cada nodo es idéntico a los demás y no es necesario que ningún nodo en concreto este operativo. Esta falta de administración central permite que los nodos se unan o se salgan de la red en cualquier momento. Esto es positivo ya que en caso de que algún nodo falle la red puede seguir funcionando con normalidad haciendo este tipo de red valida incluso en entornos que requieran un equipo con una alta estabilidad.
- **Topología dinámica.** En las redes de sensores se tiene en cuenta que los sensores son desplegados de forma aleatoria (pueden desplegarse en forma de un patrón, pero no necesariamente se tienen que desplegar en orden). También se tiene en cuenta que los sensores se pueden romper, quedar sin batería o incluso ser móviles.
- **Direccionamiento multisalto.** El rango de alcance de cada sensor en la red es limitado. Sin embargo, las redes WSN permiten que, en un supuesto caso de querer comunicarse un nodo X con un nodo Y, que esta fuera de alcance del nodo A, se podría utilizar el nodo Z como intermediario para transmitir los datos entre X y Z. Se pude ver este proceso en la *Figura 6*, en la que el nodo Z habilita la comunicación entre X e Y.
- **Autoorganizados.** Esta característica se debe a que las redes de sensores WSN no tienen una estructura fija. Debido a que no tienen una estructura fija los sensores construyen la red por si mismos cuando comienzan a comunicarse entre ellos utilizando protocolos por capas y algoritmos de distribución. Una vez se construye la red de sensores, se comienzan a recogen los datos y se envían al back-end para que realice su procesamiento.

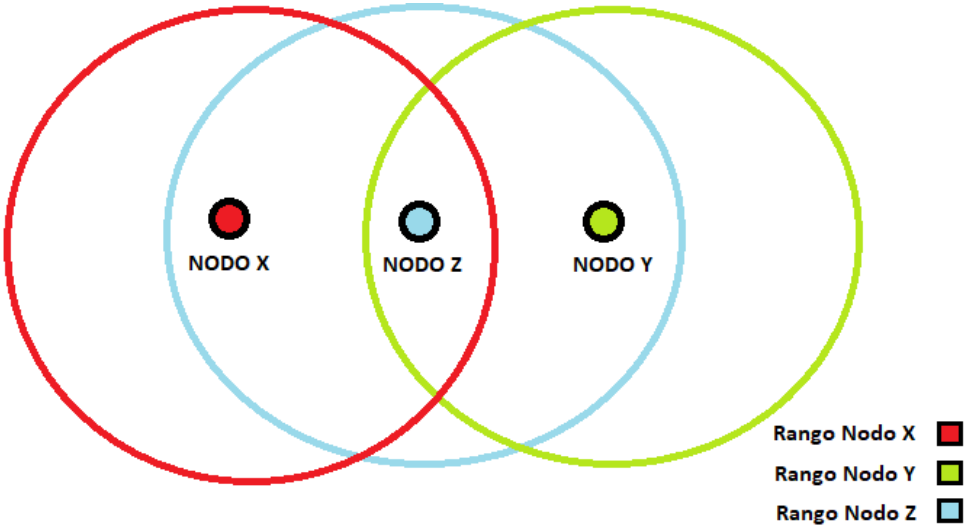


Figura 6 – Direccionamiento multisalto Fuente: Elaboración propia

2.3 Consideraciones en las redes de sensores WSN

Las redes de sensores deben de cumplir con las características descritas en el punto 2.2.1 además de contar con limitaciones debido a el uso de dispositivos IoT que tienen sus propias limitaciones por diseño:

- **Tolerancia a fallos.** Debido a las dificultades que puede causar el entorno en el que se despliegan los sensores, es común que haya problemas en algún nodo. Por ejemplo, los sensores pueden romperse o quedarse sin batería. Estos problemas tienen que ser evitados utilizando estrategias tolerantes a fallos que permitan que siga funcionando la red.
- **Restricciones hardware.** Los sensores debido a los lugares en los que se despliegan, costes y facilidad de manipulación entre otras cosas tienen unas restricciones muy considerables. Al ser dispositivos que no van a estar conectados a una fuente ilimitada de energía deben de utilizar componentes con unas capacidades limitadas para no malgastar energía. Además, los costes también hacen que entre otras cosas se cuente con una capacidad de cómputo reducida. A todo esto, se le debe de sumar el hecho de que los sensores deben de ser empaquetados en carcasas con un volumen reducido.
- **Ahorro de energía.** Cuando un sensor es desplegado para monitorizar algún entorno de interés, pese a funcionar utilizando baterías, es posible que ese sensor deba de aguantar un largo periodo que puede ser de semanas o meses. Por esto mismo, es verdaderamente importante contar con un sistema que permita ahorrar energía. Generalmente el mayor consumo de energía en relación con el tiempo es en el momento de transmitir los mensajes.
- **Comunicación.** En las redes de sensores WSN existente se muestra que hay principalmente tres tipos de comunicación. Como se puede ver en la *Figura 7*, estos tipos son: comunicación multisalto, comunicación por agrupamiento y comunicación directa. La comunicación multisalto es utilizada principalmente por el hecho de que los nodos cuentan con un alcance limitado. Gracias a los nodos vecinos se puede realizar una transmisión de datos a nodos más lejanos que de otra forma no se podría. En la comunicación directa cada nodo transmite sus datos directamente al nodo administrador y este último se encarga de enviarlos al back-end. Por

último, en la comunicación por agrupamiento o clustering los sensores se dividen en varios grupos, asignados cada uno de ellos a un nodo administrador o cluster que se encargara de recolectar los datos.

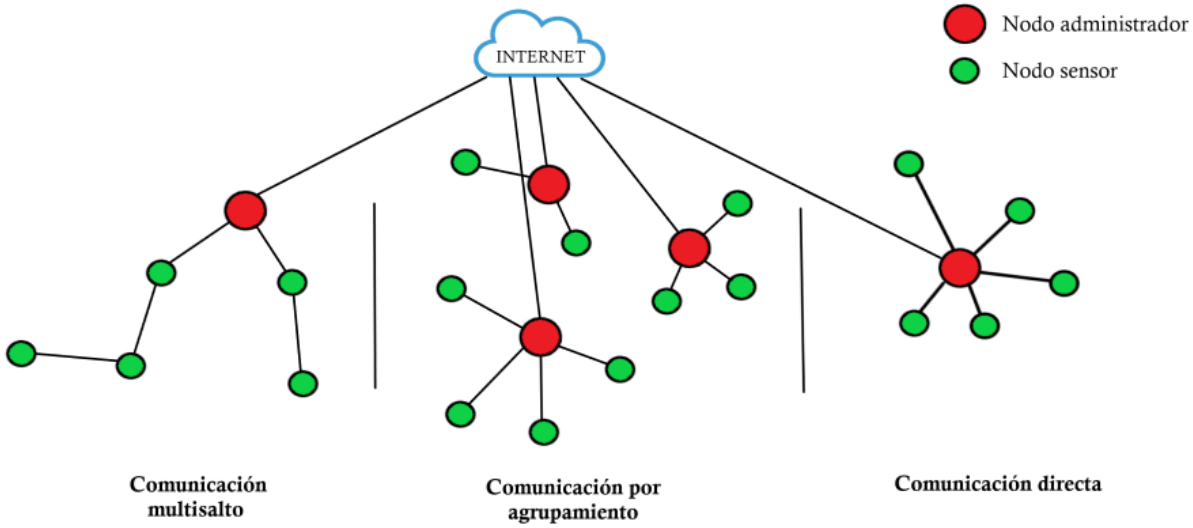


Figura 7 – Tipos de esquema de comunicación WSN Fuente: Elaboración propia

- **Escalabilidad.** Las redes de sensores deben de permitir conectarse a un número ilimitado de nodos. Además, esto se debe de hacerlo de una forma en la que la red siga funcionando con normalidad.
- **Coste.** Dependiendo del objetivo que se quiera alcanzar con la red de sensores, un ejemplo podría ser monitorizar un bosque, puede ser que un gran número de sensores tengan que ser desplegados. Esto hace que el coste total se pueda disparar si no se cuida el coste de cada nodo.
- **Movilidad.** En el caso de las redes de sensores que se comunican por clustering, al tener cada nodo sensor un nodo administrador asignado la movilidad entre clusters se puede volver un problema. No obstante, los nodos sensores en las redes WSN, en comparación con una red ad hoc, son rara vez son movidos de su lugar de despliegue inicial por lo que la movilidad no es un problema grave.
- **Patrón de sueño.** El patrón de sueño o sleep pattern en una técnica imprescindible para aumentar la duración de la red. Consiste en apagar los nodos sensores mientras no están midiendo o transmitiendo, alargando así la duración de su batería. De esta forma se pueden programar los sensores para que mientras unos estén activos otros puedan estar apagados ahorrando batería.
- **Seguridad.** Otro de los desafíos de las redes de sensores WSN (en el que nos centraremos en el documento) es proveer un sistema de seguridad pese a tener unos recursos limitados. Los requerimientos de seguridad constan de autenticación de nodos, confidencialidad de los datos y resistencia a ataques físicos.

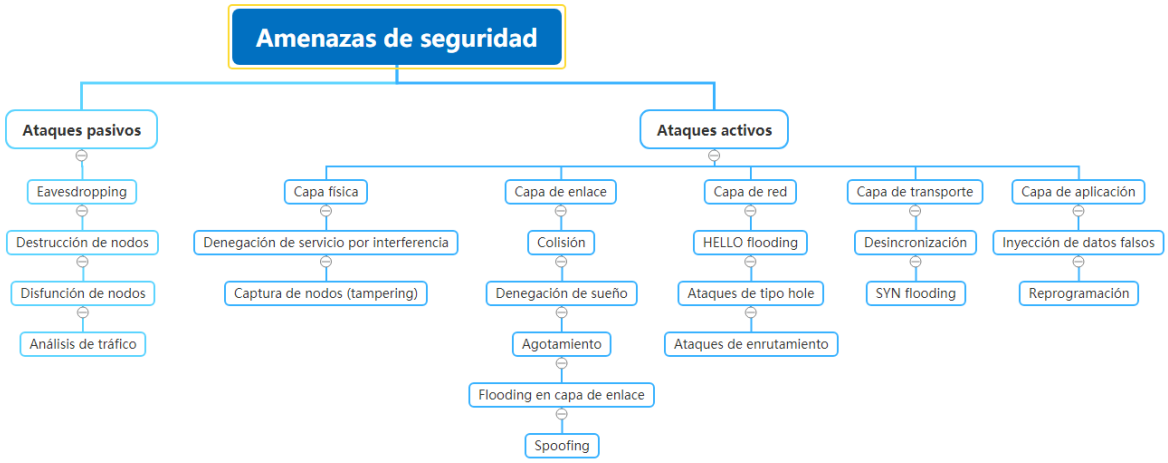


Figura 8 – Tipos de amenazas de seguridad Fuente: Elaboración propia

2.4 Vulnerabilidades y amenazas de seguridad en las redes de sensores WSN

Además de las características y de las consideraciones ya mencionadas, es necesario hacer hincapié en las amenazas a la seguridad más comunes que se pueden encontrar en las redes de sensores. En este subapartado se introducirán varias amenazas a la seguridad que podemos encontrar en una red WSN. La *Figura 8*, muestra un esquema de las amenazas que se van a comentar a lo largo de este subapartado.

2.4.1 Ataques pasivos

Los ataques pasivos se realizan en una forma en la que no pueden ser detectados de ninguna forma. Esto se debe al hecho de que los atacantes no emiten ningún tipo de onda al realizar el ataque. Las redes inalámbricas son más fáciles de interceptar, esto las hace muy susceptibles a ataques como eavesdropping, que puede ser fácilmente realizado escuchando comunicaciones inalámbricas entre sensores de la red WSN sin necesidad de capturar ningún nodo. Los ataques pasivos van principalmente en contra de la confidencialidad de datos.

En los ataques pasivos, el atacante típicamente está escondido mientras intercepta las líneas de comunicaciones para recolectar datos. Este tipo de ataque se puede agrupar en eavesdropping, disfunción de nodos, tampering/destrucción de nodos y análisis de tráfico.

2.4.1.1 Eavesdropping

Eavesdropping quiere decir escucha a escondidas o en este caso recolección de información de forma pasiva. Interceptando las redes se puede conseguir información privada. Las redes inalámbricas son un objetivo perfecto ya que son más fáciles de interceptar, lo que las hace más susceptibles a ataques pasivos. Debido a que las WSN tienen un rango de transmisión pequeño el atacante debe de estar cerca del nodo para poder recolectar información. En este sentido las redes WSN son más seguras a la intercepción de datos respecto a otro tipo de redes de largo alcance. La intercepción de mensajes transmitidos por la WSN puede revelar entre otras cosas la siguiente información: localización física del

nodo, gateways, ids de los mensajes, estampas temporales y otros campos y por supuesto cualquier información que no vaya encriptada.

2.4.1.2 Destrucción de nodos

Consiste como su propio nombre indica en la destrucción física de nodos, esto se puede conseguir de múltiples formas como sobrecarga eléctrica o simple fuerza física.

2.4.1.3 Disfunción de nodos

La disfunción de nodos consiste en causar el mal funcionamiento de un nodo, se puede deber a varios factores desde sensores defectuosos hasta ataques de denegación de servicio como una sobrecarga de los sensores. Esto consiste en generar estímulos de forma artificial para que los sensores realicen un número elevado de mediciones y de envíos de información. Esto además de alterar los datos generados por el sensor hace que desperdicie batería.

2.4.1.4 Análisis de tráfico

El patrón que sigue el tráfico de una red puede ser tan valioso como el contenido que circula por ella para los atacantes. Se puede deducir información importante sobre la topología de una red simplemente analizando los patrones de trafico de paquetes. En una red WSN, los nodos más cercanos a los nodos administradores hacen más transmisiones de paquetes que otros nodos ya que también se encargan de reenviar paquetes a diferencia de los nodos más alejados. Similarmente, el clustering, como ya se ha mencionado previamente, es una herramienta muy importante para las redes WSN y los nodos administradores que actúan de cluster reciben más paquetes que los nodos normales. La detección de clusters es importante ya que en caso de que el atacante detecte uno puede dirigir sus ataques de denegación de servicio o análisis de tráfico hacia ese nodo para conseguir mejores resultados para su propósito. Por medio de técnicas de análisis de tráfico toda esta información se puede obtener. Es más, en contextos de comunicaciones tácticas el silencio en una red puede indicar la preparación de un ataque. Similarmente, un incremento repentino en el tráfico puede indicar el comienzo de un ataque.

2.4.2 Ataques activos

En el caso de los ataques activos, los ataques no son solo contra la privacidad de los datos sino también contra la integridad de los datos. Los ataques activos también buscan acceso no autorizado y aprovechamiento de los recursos de las comunicaciones para perjudicar a la red. Un ataque activo realiza emisiones de ondas detectables por los elementos de la red WSN. En los ataques activos, un atacante afecta al funcionamiento de la red. Este puede ser el objetivo de su ataque y puede ser detectado. Por ejemplo, la red puede ser afectada disminuyendo sus capacidades o incluso dejando de funcionar por completo. El atacante también puede buscar mantenerse encubierto buscando obtener acceso a la red, amenazando así la integridad y/o confidencialidad de la red. Podemos clasificar estos ataques según la capa en la que se producen dentro del modelo OSI: capa física, capa de enlace de datos, capa de red, capa de transporte y capa de aplicación (se consideran sesión y presentación dentro de la capa de aplicación al ser una WSN).

2.4.2.1 Ataques en la capa física

La capa física se encarga principalmente de la selección de la frecuencia y la detección y modulación de la señal. Los ataques más comunes que podemos encontrar en la capa física son denegación de servicio por interferencia y captura de nodos.

2.4.2.1.1 Denegación de servicio por interferencia

En este ataque de denegación de servicio un dispositivo malicioso interfiere una señal generando una señal maliciosa en la misma frecuencia. Esta señal maliciosa hace que se genere más ruido del normal bajando los ratios de señal-ruido haciendo que al receptor le sea imposible recibir los datos de forma correcta. Las señales maliciosas pueden ser generadas continuamente inhabilitando así la comunicación en todo su rango de transmisión o funcionar a intervalos.

2.4.2.1.2 Captura de nodos (tampering)

El atacante se hace con el control del nodo mediante un ataque físico. A diferencia de los ataques pasivos el atacante no busca destruir el dispositivo, busca hacerse con el control. Para Toma el control del dispositivo hay múltiples métodos como enchufar cables en la placa para leer información almacenada o transmisiones en vivo de la red WSN. Además, el atacante puede cambiar el cableado original o cambiar el contenido de las memorias del nodo para utilizar el dispositivo. Capturar el nodo puede hacer que el atacante obtenga claves criptográficas que podrían comprometer toda la red WSN.

2.4.2.2 Ataques en la capa de enlace de datos

Los algoritmos en la capa de datos dejan abiertas muchas oportunidades para el atacante, especialmente los protocolos MAC. Esto hace que haya una gran variedad de ataques entre los que podemos encontrar ataques de tipo colisión, denegación de sueño, agotamiento, flooding en la capa de enlace y spoofing.

2.4.2.2.1 Colisión

En este tipo de ataque, el atacante transmite paquetes desde el mismo canal que el nodo de la red cuando el nodo comience a transmitir. De esta forma los paquetes colisionan y hacen que el receptor no reciba el paquete de forma correcta. El paquete recibido no sirve por lo que el receptor lo descarta y pide una retransmisión del paquete. Esta técnica funciona porque modificando solamente un byte ya se puede hacer que el CRC (Cyclic Redundancy Check) no coincida y eso hace el paquete se tenga que descartar por completo. Desde el punto de vista del atacante este ataque es mejor que el de interferencia visto en la capa física, ya que solo se emite en el momento que se está enviando un paquete, esto hace que se consuma menos energía y que sea más difícil de detectar.

2.4.2.2.2 Denegación de sueño

Este ataque consiste, como su propio nombre indica, en no dejar que los nodos puedan dormir para ahorrar energía. Esto el nodo se agote su batería más rápido de lo que debería de ocurrir. Para esta técnica se puede utilizar la técnica de colisión arriba mencionada o mediante un envío constante de señales RTS (Request To Send).

2.4.2.2.3 Agotamiento

Si el ataque de colisión descrito arriba continua de forma ininterrumpida hasta que el nodo objetivo se queda sin energía, ese ataque se llama de agotamiento. Es un ataque simple que solo requiere un nodo o un portátil que pueda transmitir en la misma banda que el sensor.

2.4.2.2.4 Flooding en capa de enlace

Este ataque aprovecha el fairness del acceso al medio. Un nodo malicioso envía una cantidad excesiva de paquetes de datos MAC (Media Access Control) o de paquetes de control MAC a sus nodos vecinos. Con eso el atacante consigue ejecutar un ataque de denegación de servicio. A su vez puede agotar la batería de otros nodos e incluso agotar el ancho de banda del canal.

2.4.2.2.5 Spoofing

En un ataque de spoofing un nodo malicioso falsifica su dirección MAC a partir de la identidad de un nodo legítimo y crea un numero de nodos aparentemente legítimos que utilizan estas identidades en la red. Generalmente el atacante busca falsificar la identidad de un nodo administrador para así recibir todo el tráfico y poder controlarlo.

2.4.2.3 Ataques en la capa de red

En el caso de los ataques a la capa de red, un atacante inyecta un número elevado de paquetes en la red causando congestión en el trafico la red además de elevar los consumos de energía.

2.4.2.3.1 HELLO flooding

Los protocolos de enrutamiento se establecen que los nodos deben enviar mensajes broadcast de tipo HELLO para anunciarse a sus vecinos a un salto de distancia. Los nodos que reciben este tipo de mensaje asumen que están dentro del rango del anunciante. Este ataque aprovecha esta característica utilizando un emisor con un rango de emisión mayor que el de los nodos de la red para que envíe mensajes por la red haciendo creer a los nodos que es su vecino. Si lo emite con una potencia suficiente, podría llegar a convencer a todos los nodos de la red de que es un nodo vecino. Cuando otros nodos envíen paquetes a este nodo malicioso, esos paquetes no los recibirá ningún nodo de la red.

2.4.2.3.2 Ataques de tipo Hole

Dentro de los ataques de tipo hole o agujero podemos encontrar distintas variantes dependiendo de lo que hagan con los paquetes:

- **Sinkhole.** En este caso un nodo malicioso se anuncia en la red como el mejor siguiente salto para mandar paquetes a su destino. Haciendo esto, el nodo consigue terminar recibiendo todos los mensajes que van hacia un nodo administrador. En este caso el nodo sinkhole no descarta los paquetes intentando no ser detectado. Esto deja abiertas muchas oportunidades para otro tipo de ataques.
- **Blackhole.** Un nodo malicioso descarta todos los paquetes que recibe para reenviar. Este ataque es especialmente efectivo si se combina con un ataque de tipo sinkhole ya que podría parar todo el tráfico de la red.
- **Grayhole o reenvío selectivo.** Este ataque es similar al blackhole ya mencionado (de ahí recibe su nombre), pero en vez de descartar todos los paquetes recibidos, el nodo malicioso es más inteligente y descarta paquetes de forma selectiva. De esta forma el atacante espera no ser detectado ya que en el caso del nodo blackhole los vecinos terminan dándose cuenta de que sus paquetes no son recibidos y cambian la ruta.
- **Wormhole.** Este ataque consiste en crear un túnel entre dos nodos maliciosos a través del que se puedan transmitir paquetes de forma más rápida que en la red normal. El objetivo de este ataque es hacer creer a los vecinos de estos dos nodos maliciosos que son vecinos entre ellos. Para conseguir esto los nodos utilizan un canal fuera de banda. El primer nodo malicioso recibe o escucha a escondidas paquetes de su entorno, envía estos paquetes a el segundo nodo malicioso a través del canal fuera de banda que tienen, que es más rápido que del que disponen los nodos normales, entonces el segundo nodo malicioso se los retransmite a sus vecinos. Esto hace creer a los receptores de los paquetes que los emisores son sus vecinos ya que los están recibiendo directamente de ellos. Este ataque es muy difícil de detectar.

2.4.2.3.3 Ataques de enrutamiento

Los ataques de enrutamiento intentan perjudicar a el flujo de paquetes cambiando sus rutas o modificando los paquetes. Hay varios tipos:

- **Desvío.** En un ataque de desvío, el atacante reenvía los paquetes recibidos por una ruta incorrecta. Puede hacer esto, por ejemplo, creando rutas falsas y anunciándoselas a los vecinos, de esta forma se introducirán estas rutas falsas en las tablas de rutas de los vecinos.
- **Partición de red.** En este ataque, una red que está totalmente conectada entre sí se particiona en subredes de forma que a pesar de que haya nodos que estén en la misma red no se puedan comunicar ya que están en distintas subredes.
- **Bucle de enrutamiento.** Los bucles de enrutamiento como su propio nombre indica hacen que unos nodos se reenvíen paquetes entre sí de forma permanente, nunca llegando el paquete a su destino.
- **Información de enrutamiento Modificada o falsificada.** La información de enrutamiento que los nodos se intercambian puede ser alterada por un nodo malicioso de forma que el esquema de enrutamiento se ve perjudicado.

2.4.2.3.4 Ataque Sybil

En este ataque un único nodo presenta múltiples identidades a los otros nodos de la red. Esto causa confusión en la red, los nodos reciben información de enrutamiento contradictoria que genera el atacante. Esto puede reducir la efectividad de la red y puede poner en riesgo otros protocolos de la red como por ejemplo algoritmos basados en votos.

2.4.2.4 Ataques en la capa de transporte

La capa de transporte en el protocolo OSI (Open Systems Interconnection) se encarga de manejar las conexiones de extremo a extremo. En la capa de transporte los ataques se aprovechan de los protocolos que mantienen información en cualquiera de los extremos.

2.4.2.4.1 Desincronización

Un atacante rompe conexiones entre dos nodos desincronizando las transmisiones entre ellos. Un ejemplo de este tipo de ataque es mandar mensajes falsificados con flags defectuosos constantemente a los extremos de la conexión para que finalmente se terminen desincronizando y perdiendo la conexión.

2.4.2.4.2 SYN flooding

En un ataque de flooding el atacante busca agotar la energía o la memoria de un nodo utilizando mensajes espurios. Esto, por ejemplo, en el caso de TCP (Transmission Control Protocol) se puede conseguir mandando mensajes SYN que son peticiones para establecer una conexión sin dejar que se establezca la conexión nunca. Con eso se consigue abrumar el buffer del objetivo.

2.4.2.5 Ataques en la capa de aplicación

Los ataques en la capa de aplicación son generalmente ataques de denegación de servicio. Protocolos como el de localización de nodos, agregación, asociación y fusión de datos o sincronización temporal pueden ser engañados u obstaculizados.

2.4.2.5.1 Inyección de datos falsos

Para influenciar el resultado total de una medición o de una lectura de datos, nodos comprometidos pueden intencionalmente inyectar datos falsos en la red WSN. Es un ataque que no afecta al funcionamiento de la red, pero si a los resultados que produce.

2.4.2.5.2 Reprogramación

De vez en cuando, todos los elementos de una red necesitan ser parcheados o reprogramados, ya bien sea por control de versiones o por una actualización en la funcionalidad. Esto por supuesto también ocurre en las redes WSN y en los dispositivos IoT. El proceso de reprogramación es un proceso crítico. Esto hace que el horario en el que van a ser reprogramados deba de ser cuidadosamente guardado ya que, si un atacante decidiese atacar, simplemente enviando mensajes falsos, cuando se está actualizando un dispositivo, este podría volverse inestable o incluso bloquearse.

2.5 Defensas ante los ataques planteados

La tabla más adelante contiene las defensas propuestas en el sector para solucionar o minimizar las vulnerabilidades que atacan los ataques pasivos.

Ataque	**Defensa**
Eavesdropping	Uso de encriptación de los mensajes, predistribución de llaves
Destrucción de nodos	Uso de hardware resistente al tampering, camuflaje de nodos en el entorno
Disfunción de nodos	Uso de hardware resistente al tampering, camuflaje de nodos en el entorno
Análisis de tráfico	Hay técnicas como Traffic Morphing [8] o Traffic Reshaping [9] pero son demasiado costosas para una red WSN

Tabla 1 – Defensas ante ataques pasivos

La *Tabla 2* contiene las soluciones propuestas en el sector ante los ataques de tipo activo, en ocasiones se puede solucionar por completo la vulnerabilidad y en ocasiones solo se puede minimizar su efecto.

Ataque	**Capa**	**Defensa**
Denegación de servicio por interferencia	Física	Uso de un espectro de comunicaciones amplio
Tampering	Física	Uso de hardware resistente al Tampering
Colisión y Agotamiento	Enlace	Limitar la cantidad de paquetes por dirección MAC, uso de una tecnica de transmisión del tipo TDM que permita limitar el tiempo de acceso al medio que tiene cada transmisor
Denegación de sueño y Flooding	Enlace	Análisis de tráfico y métodos IDS (Intrusion Detection Systems)
Spoofing	Enlace	Utilización de métodos de autenticación de nodos
HELLO flooding	Red	Verificación bidireccional de identidad de nodo [10]
Ataques de tipo Hole y de enrutamiento	Red	Mejorar los protocolos de enrutamiento, para contrarrestar Wormhole concretamente usar llaves basadas en la localización del nodo
Ataque Sybil	Red	Verificación de identidad y autenticación de nodos
Desincronización	Transporte	Uso de autenticación en todos los paquetes recibidos
SYN flooding	Transporte	Uso de protocolos sin conexión (por ejemplo, UDP User Datagram Protocol), desafíos para establecer conexión, aunque requieren un uso extra de recursos
Inyección de datos falsos	Aplicación	Uso de algoritmos de detección de inyección de datos [11]
Reprogramación	Aplicación	Uso de dispositivos con un modo de recuperación.

Tabla 2 – Defensas ante ataques activos

En estas tablas se puede ver como para garantizar la seguridad de una red de sensores WSN lo más importante es aumentar la resistencia al tampering y reforzar las comunicaciones en la red. Para mejorar las vulnerabilidades anteriormente expuestas, lo más común es utilizar hardware preparado resistir ataques de tipo tampering, mecanismos de autenticación de nodos y criptografía para garantizar la integridad o la privacidad de los datos que viajan por la red.

3. SOLUCIONES BASADAS EN HARDWARE CRIPTOGRÁFICO

En el sector de seguridad IoT ya hay muchas propuestas para securizar los dispositivos. No existe una solución global que se pueda utilizar de forma indiscriminada. Cada una de estas soluciones tiene ventajas e inconvenientes. Estas soluciones han sido creadas con un objetivo en mente, por eso se pueden encontrar soluciones en un gran abanico de precios y características.

En esta sección se comentarán tres tipos de soluciones con distintas que se encuentran en distintos rangos de precios, yendo desde algo de lo más barato que se puede encontrar hasta soluciones de empresas con costes de licencia. Pero antes de ver las soluciones que se proponen es necesario ver qué características puede tener cada solución. En el apartado anterior ya se han comentado las defensas que hay disponibles ante los ataques comentados, pero es necesario ver estas defensas más en profundidad para entender las propuestas.

3.1 Características generales que buscar en las soluciones

En este apartado se van a describir las características necesarias para securizar un dispositivo IoT establecidas en el apartado anterior. Se necesitan mecanismos de autenticación para garantizar la identidad de los dispositivos, hardware resistente al tampering para resistir los ataques físicos y criptografía para proteger las comunicaciones.

3.1.1 Autenticación

La autenticación es un proceso que permite garantizar la identidad de un dispositivo. Es recomendable que los dispositivos IoT sean autenticados antes de poder comunicarse con otros dispositivos de la red. Esto mitiga el riesgo que un atacante pueda falsificar un dispositivo IoT que pueda parecer ser legítimo en la red. Esta característica, como se ha visto en el apartado anterior, es algo que es muy recomendable implementar en cada dispositivo conectado a la red para evitar múltiples tipos de ataques. Hay tres tipos de autenticación en función de cómo se autentiquen los dispositivos como se puede ver en la *Figura 9*.

Estos tres tipos de autenticación se pueden dividir en dos, autenticación por pares y autenticación grupal. En la *Figura 9*(a), se puede ver la autenticación por parejas, en la que el nodo X se autentica con el nodo Y y viceversa. En la autenticación grupal, primero está la autenticación en cluster que se puede ver en la *Figura 9*(b). En este tipo de autenticación los nodos se autentican con el nodo cluster. Finalmente, está la autenticación global, que se puede ver en la *Figura 9*(c), en la que la autenticación de cada nodo es verificada por el nodo administrador y a su vez por el resto de los nodos.

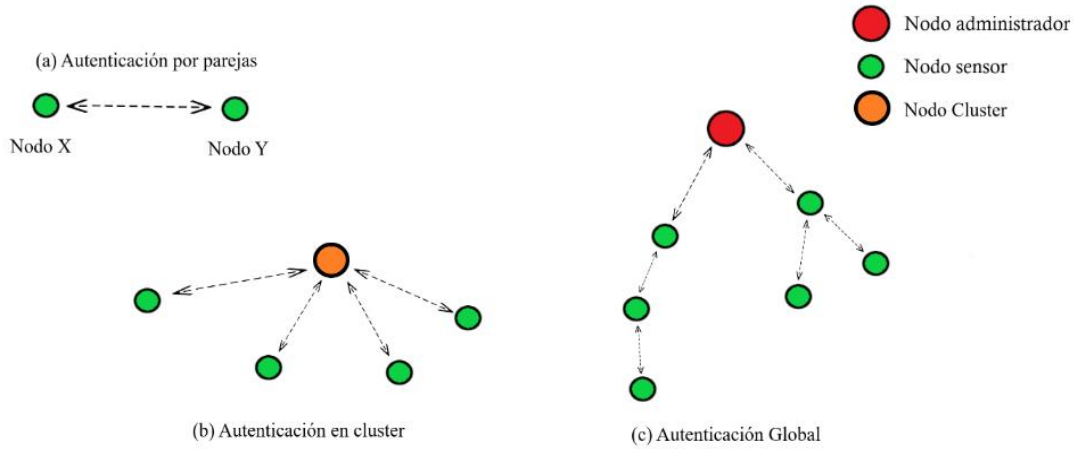


Figura 9 – Tipos de autenticación Fuente: Elaboración propia

3.1.2 Resistencia al tampering

El tampering es el ataque o manipulación física de un dispositivo. Para prevenir los ataques físicos, hay dos formas de abordar la solución, de forma mecánica y eléctrica. Entre los métodos mecánicos podemos encontrar el uso de tornillos especiales. Por ejemplo, antiguamente, en las cabinas telefónicas se utilizaban tornillos especiales con una forma triangular, de esta forma si no tienes acceso a dicha llave no puedes manipular el dispositivo. Los métodos eléctricos consisten principalmente en detectar la intrusión de un atacante. Uno de los métodos utilizados, consiste en utilizar cables con más hilos de los necesarios y conectar estos hilos a una alarma que salta en caso de detectarse manipulación en los cables o en el dispositivo.

3.1.3 Criptografía

Para proteger la privacidad y securizar las comunicaciones, los nodos de la red deben de compartir llaves con cada uno de sus nodos vecinos o la red, dependiendo del tipo de autenticación que se quiera implementar. Por ejemplo, en el caso de la *Figura 9*(a), para la securización de la comunicación entre X e Y se crearía un par de llaves. Mientras tanto en el caso de la *Figura 9*(b) se necesitaría un par de llaves por cada nodo conectado al cluster, en este caso 4 pares de llaves. Dentro de la criptografía se puede clasificar la autenticación de nodos y el establecimiento de llaves en tres tipos: llaves simétricas, llaves asimétricas y funciones hash.

3.1.3.1 Llaves simétricas

La criptografía con llaves simétricas consiste en que el emisor y el receptor comparten la misma llave. Actualmente estándar más popular es el AES (Advanced Encryption Standard). El proceso de encriptación con llave simétrica se puede ver en la *Figura 10*. Consiste en que ambas partes compartan una llave distribuida de forma previa. El emisor con su llave encripta la información a transmitir, en el caso del ejemplo un texto. La información viaja encriptada por la red. Finalmente, el receptor desencripta la información con su llave, que es la misma que la del emisor.

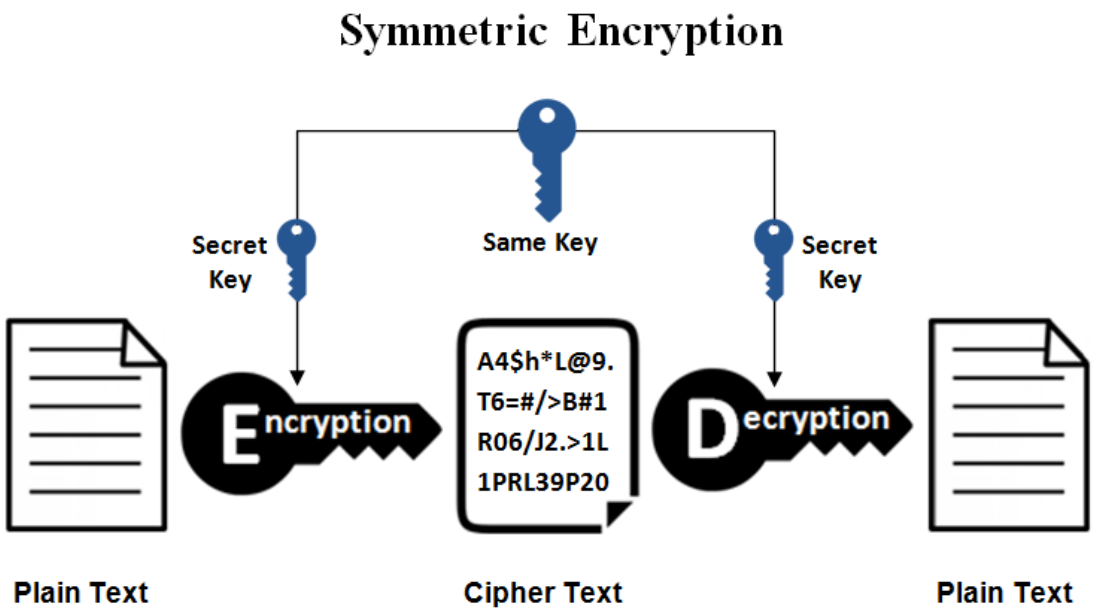


Figura 10 – Cifrado usando llave simétrica Fuente: **[41]**

Esta forma de encriptación tiene un gran problema, debido a que las llaves deben de estar desde el comienzo en ambos lados de la transmisión, para que esta se pueda hacer de forma segura. Una solución a este problema es el uso de una llave maestra, esta llave se distribuiría en el establecimiento de la red a todos los dispositivos. A partir de esta llave maestra se generarían nuevas llaves y como todos los nodos tienen la misma llave maestra, estas llaves serían iguales. Sin embargo, esta solución falla al no tener en cuenta posibles brechas de seguridad, y en una red WSN es algo que va a ocurrir. En este caso cuando un nodo es comprometido, el atacante puede obtener la llave maestra y con ella puede descifrar todas las comunicaciones de la red. Una posible solución a este problema es que los nodos establezcan conexiones con sus vecinos y después borren la llave maestra, aunque esto haría que no se puedan unir nuevos nodos.

Otra forma de distribuir las llaves es que cada nodo almacene las llaves de toda la red, esto solo es válido en redes pequeñas ya que requiere mucha memoria. En el caso de redes grandes, se podría establecer un pool de llaves, con una cantidad suficiente para que cada llave se repita las menos veces posibles. De esta forma se reduce el uso de memoria a costa de un riesgo mínimo en la seguridad.

3.1.3.2 Llaves asimétricas

El cifrado de llaves asimétricas también se conoce como cifrado de llave pública. Esto se debe a que en este tipo de cifrado hay dos tipos de llaves, llave publica y llave privada. La llave publica, como su propio nombre indica es publica, por lo que la conoce todo el mundo y la tiene todo el mundo. Esta llave se utiliza para la encriptación. Los datos encriptados con la llave publica solo pueden ser desencriptados con la llave privada correspondiente. La llave privada solamente la conoce el nodo que recibe los datos.

En cuanto al intercambio de las llaves, a diferencia de las llaves simétricas, este intercambio se puede realizar de forma segura. El algoritmo más famoso de intercambio de llaves es el Diffie-Hellman. Este algoritmo permite acordar una clave secreta entre dos dispositivos, a través de un canal no seguro. Como se puede ver en la *Figura 11*, se parte de una situación en la que hay dos números que son públicos a

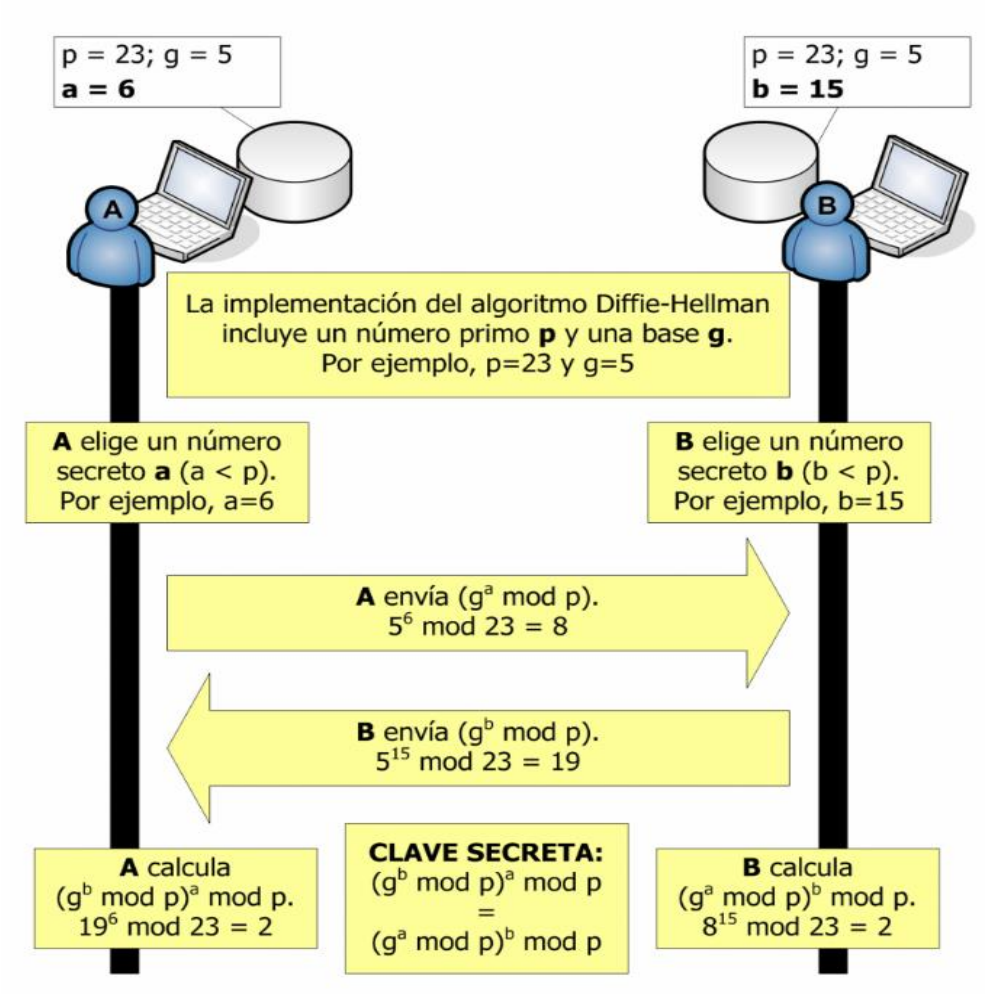


Figura 11 – Algoritmo Diffie-Hellman Fuente: **[42]**

todo el mundo, p que es un número primo y g que es una base. A partir de estos valores el extremo a y el extremo b generan un numero aleatorio menor a el numero primo p, llamados a y b respectivamente. Entonces cada extremo de la conexión realiza la operación $g^{numero\ generado} mod\ p$ y envía el resultado a el otro extremo. A continuación, cada extremo eleva el número recibido a el numero a, en el extremo a, y a b en el extremo b. De esta forma se obtiene una clave simétrica para cifrar las comunicaciones. Utilizando unos números lo suficientemente grandes la clave generada se vuelve imposible de averiguar para el atacante.

Este algoritmo presenta una vulnerabilidad y tiene que ver con el hecho de que no se compruebe la identidad de los extremos. Esto hace que un ataque man in the middle pueda establecer un nodo intermedio que interactúe con cada uno de los extremos haciéndose pasar por ellos. Esto se puede evitar utilizando este algoritmo en conjunto de uno de autenticación.

Sin embargo, este algoritmo es muy pesado en computo para dispositivos IoT por lo que en estos casos se utiliza un algoritmo de curva elíptica ECC (Elliptic Curve Criptography) que funciona de una forma similar al explicado, pero con menos carga computacional.

3.1.3.3 Funciones hash

Las funciones hash son funciones que reciben una entrada y producen un resultado de forma rápida. Sin embargo, el proceso reverso de conseguir la entrada a través de la salida es un proceso tan costoso que se considera inalcanzable. El protocolo más conocido de funciones hash es el SHA (Secure Hash

T_1	T_2	...	T_{n-1}	T_n
K_n	K_{n-1}	...	K_2	K_1

Figura 12 – Asignación de llaves e intervalos temporales
Fuente: Elaboración propia

Algorythm). Esto puede ser aplicable para la autenticación de dispositivos IoT en forma de cadenas de hash. Los nodos de la red generan un valor inicial $h^1(k) = h(k)$, donde k es la llave inicial y $h^1(k)$ representa que la llave inicial k ha sido pasada por la función hash una vez. Entonces h^n puede ser visto como que la llave inicial k ha sido pasada por la función hash n veces, de forma que $h^n(k) = h(h^{n-1}(k))$, donde n = 2,3,4…. Debido a la propiedad de las funciones hash de no ser fácilmente reversibles, la cadena hash puede ser utilizada en orden inverso de generación. De esta forma se puede demostrar que $h^{n-1}(k)$ es auténtico si se demuestra que $h^n(k)$ es auténtico. Esto se puede utilizar en una red de sensores considerando el tiempo que van a estar desplegados, dividiendo ese tiempo en n intervalos, siendo cada intervalo T_n, y asignando a cada intervalo su llave maestra $K_m = h(K_{m-1})$, donde $1 \leq m \leq n$, $K_1 = h(k)$ y k es la llave inicial. En la *Figura 12* se puede ver como quedaría repartido.

Estas llaves maestras, entre otras formas, se pueden utilizar para el autenticado de mensajes añadiendo un código MAC (Message Authentication Code) a los mensajes. Este código MAC se generaría como $MAC = h(k; m)$ donde m es el mensaje enviado protegido por la llave maestra k. Utilizando las llaves maestras una única vez se podría garantizar la autenticidad de los mensajes.

3.2 Solución empresarial

En el mundo empresarial de la informática, la seguridad de los dispositivos IoT es un tema que se debe considerar. Ya hay empresas que han creado servicios para administrar, configurar y monitorizar dispositivos IoT. Entre estas empresas se pueden encontrar gigantes como Amazon con su IoT Device

Figura 13 – Esquema de la solución Extreme Defender for IoT Fuente: **[43]**

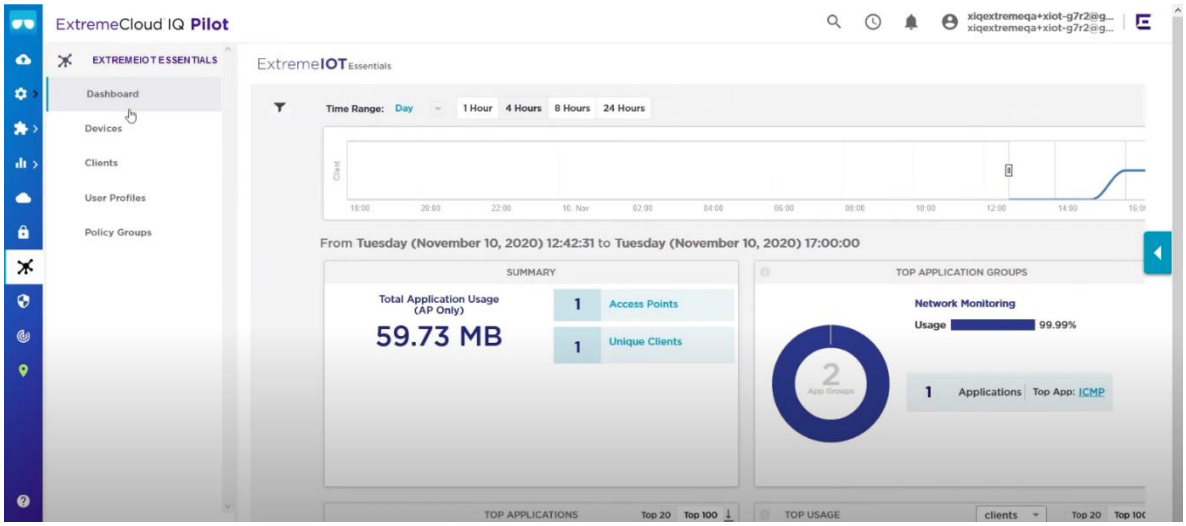


Figura 14 – Menú Dashboard Fuente: **[44]**

Defender o Microsoft con su Azure Defender for IoT. Sin embargo, estas soluciones son más dirigidas a la administración de una red IoT ya securizada.

La empresa Extreme Networks, es una empresa dedicada a la fabricación de infraestructura de red y al desarrollo de software de gestión de estas. Cuenta con una gama de productos que permite securizar la red de una forma sencilla. Su gama de productos cuenta con un software para la gestión y dispositivos hardware para la securización.

Esta solución se centra en securizar dispositivos que tienen capacidades limitadas o que directamente no tienen ningún tipo de seguridad de una forma en la que no se tiene que modificar el dispositivo a securizar. Esto es especialmente útil en dispositivos antiguos o en dispositivos que no son modificables. Además de securizar la red, esta solución se encarga de segmentar la red en zonas diferentes. Esto permite que caso de que se comprometa una subred las otras se ven inalteradas. A su vez se sigue permitiendo la movilidad de dispositivos entre subredes de forma sencilla.

Esta solución funciona utilizando el software Extreme Defender for IoT conectado con los dispositivos IoT mediante el adaptador SA201 (alámbrico) o el punto de acceso AP150W (inalámbrico). En la *Figura 13* se puede ver como quedaría el esquema de la solución.

3.2.1 Extreme Defender for IoT

Este es el software de gestión. Como se puede ver en la *Figura 14*, es una aplicación con una interfaz muy amigable que permite que usuarios que no son especialistas del sector la utilicen con teniendo conocimiento básico de ordenadores. La aplicación cuenta con menús que permiten ver métricas de la red, ver los puntos de acceso conectados, ver los dispositivos conectados a los puntos de acceso, crear grupos de políticas y perfiles de usuario entre otras funcionalidades.

En la *Figura 14* se puede ver la página principal o Dashboard en la que se pueden ver métricas de que están haciendo los dispositivos conectados y monitorizar la red entre otras funciones. Se puede seleccionar el rango temporal que se quiere ver de la información además de ver un historial.

En el menú de Devices o dispositivos, se pueden ver los puntos de acceso que hay en la red asignados a dispositivos IoT, se pueden identificar por un símbolo morado que aparece en el status. Cambiando los filtros se pueden ver todos los puntos de acceso de la red y asignar para dispositivos IoT los puntos de acceso que se quieran.

Figura 15 – Adaptador SA201 Fuente: **[43]**

En el menú Client o cliente se pueden ver los dispositivos cliente, que son los dispositivos conectados a los puntos de acceso. Se muestra la información básica del dispositivo como la dirección IP, la dirección MAC o el estado en el que está el dispositivo. Además, en este menú es donde se asignan las políticas de grupo a cada dispositivo.

En el menú User profiles o perfiles de usuario, se pueden crear perfiles en los que se modifican las reglas del firewall.

En el menú Policy group se pueden crear grupos para agrupar dispositivos y asignarles una misma política. Por ejemplo, se puede crear un grupo que sean los nodos administradores para que a todos ellos se les aplique una misma política de forma rápida y sencilla.

3.2.2 Adaptador SA201

Este dispositivo, de un tamaño similar al de una lata de refresco, permite securizar dispositivos conectados por cable a la red. Se puede ver cómo es físicamente el dispositivo en la *Figura 15*. Para utilizar este dispositivo se requiere de un equipo controlador.

Como se puede ver en la *Figura 16*, el adaptador se conecta al dispositivo que se quiere securizar lo más cerca posible. El adaptador se conecta a un controlador XCC y terminador del túnel Ipsec. Este controlador se conecta con dispositivo con Extreme Defender for IoT. Las conexiones a el adaptador SA201 son utilizando cables RJ45. Este adaptador se puede colocar al lado o dentro del dispositivo si el este es lo suficientemente grande. Este dispositivo trabaja en conjunto con la aplicación Extreme Defender para aplicar las políticas aplicadas por la aplicación.

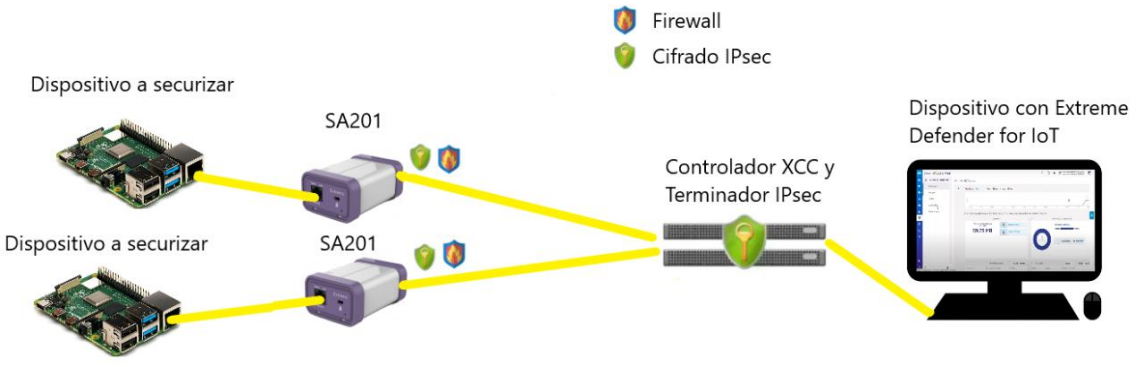


Figura 16 – Esquema de conexión SA201 Fuente: Elaboración propia

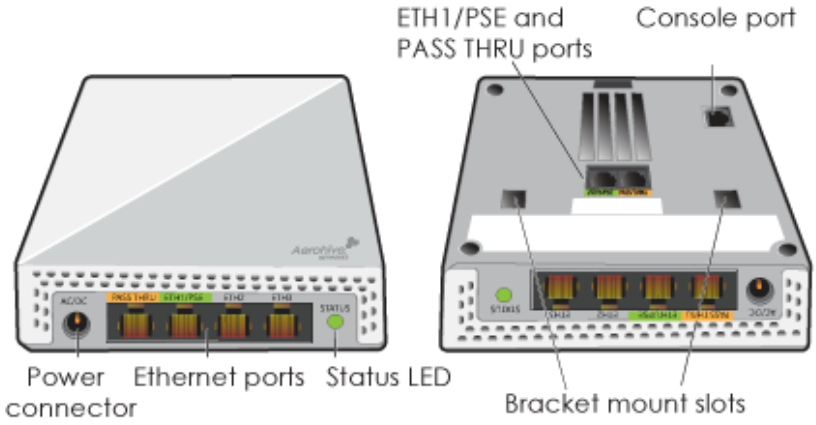


Figura 17 – AP150W Fuente: **[45]**

La seguridad que aporta consiste en monitorizar el tráfico que pasa a través de el con una visibilidad completa desde la capa 2 hasta la 7 del protocolo OSI. Esto permite bloquear todo el tráfico que no esté especificado en el perfil asignado en el software. Además, permite establecer conexiones seguras utilizando túneles Ipsec que hace para que se cifren los datos enviados entre el adaptador SA201 y el controlador. Además, el adaptador cuenta con un firewall. El adaptador tiene tecnología PoE (Power over Ethernet) o alimentación por Ethernet a través del puerto RJ45 lo que hace que no haya necesidad de enchufarlo a una toma de corriente ya que se puede alimentar directamente del cable de internet.

Este dispositivo aporta encriptación en las comunicaciones gracias a las conexiones Ipsec. También proporciona autenticación del dispositivo y un firewall.

Esta solución es similar a la que se elegirá finalmente ya que se conecta al lado del sensor y cifra las comunicaciones, pero solo cumple parcialmente los requerimientos propuestos anteriormente. Esto se debe a que no cuenta con resistencia al tampering. Además, es una solución alámbrica por lo que no es la solución ideal en una red de sensores IoT.

3.2.3 AP150W

El AP150W es un punto de acceso y conmutador con tecnología IoT integrada. Se puede ver el punto de acceso en la *Figura 17*. Este dispositivo proporciona acceso a la red de forma inalámbrica mediante Wifi, BTLE (BlueTooth Low Energy) o ZigBee. También cuenta con cuatro puertos Ethernet que hacen que pueda funcionar de forma cableada si se desea.

Como se puede ver en la *Figura 18* el punto de acceso se coloca entre el dispositivo a securizar y el dispositivo con el software Extreme Defender for IoT. Se puede conectar con ambos extremos de forma inalámbrica o de forma cableada (aunque en la *Figura 18* se ve una conexión cableada con el dispositivo con el software Extreme Defender for IoT esta conexión también podría ser inalámbrica). En este caso, a diferencia del adaptador SA201 no es necesario un controlador ya que este dispositivo no cuenta con

Figura 18 – Esquema de conexión AP150W Fuente: Elaboración propia

IPsec. Gracias a esto el software Extreme Defender for IoT puede estar en otro lugar y conectarse por medio de internet

En cuanto a la seguridad, este punto de acceso proporciona un firewall en las capas 2 a 7 del protocolo OSI. También, cuenta con tecnología PPSK (Private Pre-Shared Key) que permite la autenticación de usuarios con claves personales para cada usuario.

Este punto de acceso únicamente proporciona autenticación de los dispositivos conectados, de las características que se buscan en este proyecto. Esto hace que no sea una solución apta para la securización que se buscada ya que carece de encriptación de los datos y de resistencia al tampering.

3.2.4 Conclusión de Extreme Defender

Estos dispositivos junto con el software de Extreme Defender proporcionan una defensa aceptable ante los ataques pese a ser una solución fácil de desplegar y muy amigable para usuarios inexpertos. Sin embargo, el rango de precios en el que se mueve es elevado. En cuanto al software Extreme Defender, el precio es privado y solo se lo dan a empresas. El adaptador SA201 está listado en la página web cdw por 380$ [12]. El punto de acceso AP150W ha estado listado en Amazon en un rango de 240-220€ [13], ya no está listado, pero se ha mirado el historial de precios que ha tenido. Esos son los precios de los dispositivos sin contar con las licencias que se necesitan. El precio de desplegar una red con varios dispositivos se puede disparar a los miles de euros en cuanto se añaden unos pocos dispositivos, solamente contando el hardware ya que no se conoce el precio del software y de las licencias. Esto ya es un claro primer indicio de que esta solución no está enfocada a la situación que se plantea en este documento.

En el caso de la solución que se quiere plantear para una red de sensores WSN utilizando una placa Raspberry Pi o Arduino, estos dispositivos son compatibles. Sin embargo, el precio es demasiado elevado y no cumple con todos los requisitos de seguridad que se requieren para un despliegue con un buen nivel de seguridad. El adaptador SA201 sí que cuenta con encriptación, pero se conecta de forma alámbrica lo cual no es compatible con una red WSN y tampoco ofrece resistencia al tampering. En cambio, el punto de acceso AP150W ofrece una conexión inalámbrica pero no proporciona ni mecanismos de encriptación ni resistencia al tampering.

Sin embargo, esta solución es apta para securizar dispositivos con unas características muy particulares. Estas características son las de un dispositivo que tiene un precio elevado, no viene securizado de fábrica y no se puede modificar su software o sus componentes. Un ejemplo claro de estos dispositivos son los

aparatos médicos que se pueden encontrar dentro de un hospital. Por ejemplo, un equipo de densitometría ósea que puede costar alrededor de los 30.000€ y que viene con Windows 7 de fábrica. En este equipo fácilmente se podría instalar un firewall o un antivirus entre otras soluciones para aumentar su seguridad. El problema es que estos equipos no pueden ser modificados por motivos de garantía de fábrica ya que son dispositivos críticos y se tiene que garantizar su funcionamiento. Este es el escenario idóneo para esta gama de productos. En este caso por ejemplo se podría utilizar el adaptador SA201 conectado a la máquina, al ser una maquina grande el adaptador se puede guardar en algún compartimento de forma que este más protegido. El adaptador se conectaría con un cable hasta el controlador, siendo este el canal seguro. Finalmente, el controlador se conecta con el dispositivo con el software Extreme Defender for IoT. De esta forma se tendría encriptación en toda la línea de la red.

Con esto se concluye que esta solución empresarial no es adecuada para la securización de una red de sensores WSN.

3.3 Solución con modulo TPM

Los módulos TPM (Trusted Platform Module) son unos módulos encargados de manejar las claves criptográficas dentro del dispositivo. En el caso de la IT tradicional, el encargado de hacer esto es el sistema operativo, tanto Windows como Mac OS o Linux/UNIX entre otros cuentan con una herramienta software que se encarga de este fin. Estos almacenes software cuentan con varias debilidades, por eso mismo se utiliza un módulo TPM que es más robusto.

El uso de los modules TPM es algo que ya se ha utilizado en el pasado en el sector. Windows y Apple ya han utilizado o compatibilizado el uso estos dispositivos anteriormente. Estos módulos están incluidos en algunas placas base actuales. Actualmente se ha anunciado Windows 11 y este nuevo sistema operativo requerirá que el sistema en el que este instalado cuente con un módulo TPM. Esto ha causado un gran revuelo ya que hoy en día no todos los ordenadores tienen de fabrica un módulo TPM.

Los módulos TPM cuentan con unas características técnicas que están definidas en un estándar abierto definido por el TCG (Trusted Computing Group). Actualmente la versión más reciente del protocolo principal es la 1.2 definida en 2011, revisión 116. La especificación de librerías está en la versión 2.0 con última actualización en 2019.

Estos estándares garantizan varias funcionalidades de seguridad en los módulos TPM. Primeramente, garantiza la integridad de la plataforma. Para ello se asegura de que el proceso de arranque del dispositivo se haga de forma segura. Esto significa que el proceso de arranque utilice hardware y software firmado garantizando así un arranque del sistema correcto. Segundamente, estos módulos contienen algoritmos de encriptación entre los que se puede encontrar SHA-1 y SHA-2 como funciones hash, RSA (Rivest, Shamir y Adleman) y ECC como algoritmos de clave publica, AES como algoritmo de clave privada y HMAC (Hash Message Authentication Code) para la autenticación de dispositivos. El módulo TMP puede generar y manejar claves para funcionar con los algoritmos mencionados.

El módulo TPM está basado en un cripto-procesador que aporta características de seguridad avanzadas. Su principal ventaja frente a los almacenes software es que, permite generar y almacenar claves criptográficas y realizar operaciones con ellas, de una forma en la que las claves nunca abandonan el módulo TPM. Este módulo esta especialmente diseñado contra ataques de tipo tampering. Se pueden integrar a nivel BIOS/UEFI, forma de la que agregan criterios de seguridad en el arranque del sistema. Esto es muy interesante porque permite la autenticación del dispositivo. Un ejemplo de una implementación similar a los módulos TPM es la que se puede encontrar en las tarjetas SIM (Subscriber Identity Module) de los teléfonos móviles. Estas tarjetas permiten la autenticación y el cifrado de las

Figura 19 – Raspberry Pi con el módulo Iridium9670 TPM2.0 LINUX
***Fuente:* [46]**

comunicaciones. Esto se hace mediante el uso del Ki, una clave simétrica de 128 bits almacenada en la tarjeta SIM de la que nunca sale.

Estos módulos TPM son realmente interesantes para una implementación en una placa Raspberry Pi ya que permiten la autenticación del hardware y del software además de contar con todo tipo de algoritmos criptográficos. También cuentan con diseño que proporciona una excepcional resistencia al tampering.

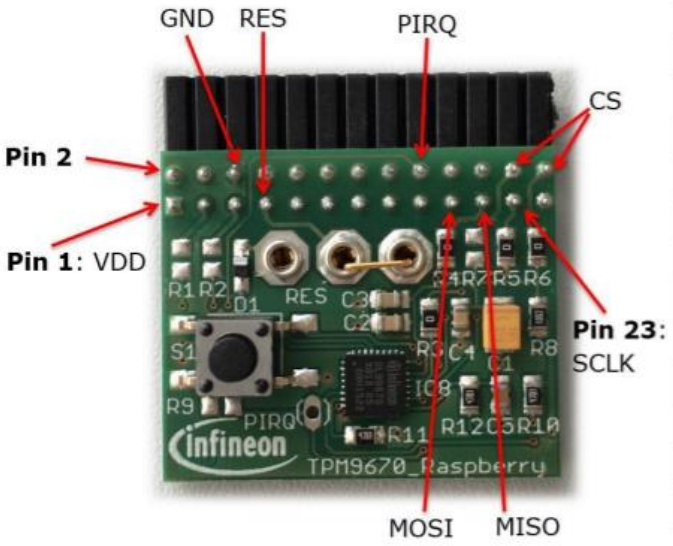
3.3.1 IRIDIUM9670 TPM2.0 LINUX

Este es un módulo de la familia Infineon OPTIGA TPM. Esta es la gama de módulos TPM fabricados por Infineon dedicados a la seguridad de sistemas embebidos. El módulo elegido, Iridium9670 TPM2.0 LINUX está diseñado especialmente para la plataforma Raspberry Pi. Se puede ver el módulo conectado a una placa Raspberry Pi en la *Figura 19*. Es un módulo de tamaño reducido que cuenta con un botón de reseteo.

Se conecta mediante el bus SPI (Serial Peripherial Interface), este bus aporta una serie de ventajas como una comunicación full duplex y una mayor velocidad de transmisión que un bus I2C (Inter Integrated Circuit) entre otras. Para su funcionamiento son necesarias cuatro señales SCLK, MOSI, MISO y CS. En la *Figura 20* se puede ver el mapeo de pines que se hace respecto de la Raspberry Pi. Utiliza el pin 1 para alimentación, el pin 6 para tierra, el pin 7 para la señal de reseteo, pin 18 para las peticiones de interrupciones PCI, pin 19 para recibir datos del maestro, pin 21 para enviar datos al maestro, pin 23 para la señal de reloj y los pines 24 y 26 para la selección del esclavo. La transmisión de bits en este tipo de bus se realiza de manera síncrona con cada pulso de reloj.

Este módulo hace uso del chip OPTIGA TPM SLB9670 2.0. Este chip cuenta con la certificación EAL4+. La certificación EAL (Evaluation Assurance Level) mide la seguridad de los chips en una escala del 1 al 7. Para ello realizan test de penetración. Este módulo está en la escala 4, esta es la escala que se alcanza típicamente en los productos destinados a venderse en el mercado y a producir beneficios, ya que llegar a las escalas más altas conlleva un incremento en los gastos exponencial. En esta escala es donde se encuentran productos como Windows o Mac OS.

Este chip cuenta con todos los mecanismos de seguridad, ya mencionados, cumplimentando con el estándar TPM 2.0. Se pueden resumir estos mecanismos en resistencia al tampering, autenticación software y hardware y algoritmos de encriptación.



Signal	Pin	Pin	Signal
VDD	1	2	-
-	3	4	-
-	5	6	GND
RES	7	8	-
-	9	10	-
-	11	12	-
-	13	14	-
-	15	16	-
-	17	18	PIRQ
MOSI	19	20	-
MISO	21	22	-
SCLK	23	24	CS (optional)
-	25	26	CS (default)

Figura 20 – Mapeo de pines GPIO Fuente: **[46]**

Infineon proporciona unos drivers [14] que cumplen con el estándar TSS (TPM2 Software Stack) de TCG. Este estándar regula que se provea una interfaz estándar para los módulos TPM de forma que las interfaces de distintos fabricantes se puedan comunicar. Estos drivers además contienen una guía de instalación rápida y documentación para poner el módulo en funcionamiento.

Este módulo TPM cuenta con todos los requerimientos de seguridad propuestos para este proyecto e incluso aporta una característica adicional, el arranque seguro del dispositivo.

3.3.2 Conclusión módulos TPM

Los módulos TPM se pueden encontrar en el mercado en un rango de precios medio. Se pueden encontrar módulos desde 40€ hasta más de 70€. El módulo comentado anteriormente cuesta alrededor de 50€ en la página digikey [15]. Estos precios hacen que la solución de utilizar un módulo TPM sea muy realista. Para utilizar este módulo TMP también es necesaria una Raspberry Pi. La Raspberry Pi Zero W, que tiene las características mínimas necesarias para utilizar el módulo TPM tiene un precio de aproximadamente 10 euros más envío en la página kubii [16]. Esto hace un total de aproximadamente 60€ más costes de envío. Esta solución cuenta con un precio razonable, pero a cambio aporta unas características de seguridad muy potentes, difícilmente superables en un rango de precio similar. La dificultad de desplegar este módulo TPM es media o alta ya que requiere conocimientos en el campo de la informática además de leer documentación sobre el módulo TPM y su funcionalidad. Gracias a Infineon el código de funcionamiento básico [14] viene hecho lo que reduce la dificultad y el tiempo de despliegue. Estos módulos actualmente se utilizan, además de en la industria de la informática, en la industria de la automoción y en otros usos industriales.

Esta es una solución muy buena desde el punto de vista de la seguridad para implementar en una red WSN. Aporta todas las características de seguridad necesarias y más. Sin embargo, el precio se dispara en el caso de implementar una red con un gran número de sensores.

3.4 Solución con chip criptográfico

Como se ha comentado en un apartado anterior, la criptografía permite la encriptación de los mensajes por medio del uso de llaves y de funciones hash. Las operaciones que se realizan en la criptografía son operaciones muy costosas en computo cuando se ejecutan sobre un hardware general. El hardware criptográfico es un hardware especializado en realizar operaciones criptográficas de una forma óptima, lo que hace que las operaciones sean menos costosas en energía y tiempo.

La solución de utilizar hardware criptográfico no es nada nuevo dentro del mundo de la informática, dentro del mundo IoT ya se ha intentado hacer alguna implementación utilizando este tipo de hardware. Hay empresas como Microchip que han creado unos chips criptográficos que son compatibles con dispositivos IoT. Concretamente hay tres chips distintos cada uno con un tipo de encriptación. Se puede ver el chip ATSHA204A, que se comentara más adelante, en la *Figura 21*. Los otros dos chips externamente son idénticos, cambiando la inscripción del nombre.

Estos chips se conectan mediante el bus I2C por lo que son compatibles con dispositivos como Raspberry Pi, Arduino o BeagleBone. Estos chips tienen el formato SOIC (Small Outline Integrated Circuit) de 8 pines, esto los hace difíciles de conectar a una placa. Es recomendable convertir esta conexión a una del tipo DIP (Dual In-line Package) que al ser un poco más grande es más manejable. Para ello se pueden utilizar adaptadores que tienen un precio muy bajo o incluso fabricarlo de forma casera con unos pocos materiales muy básicos. Una vez en formato DIP se puede simplemente conectar con cables a la placa deseada o incluso utilizar una placa protoboard para mayor simplicidad.

Junto con este hardware de Microchip proporciona una documentación detallada y librerías de código abierto a partir de las cuales ya ha habido algunos desarrollos que a su vez son públicos también. Por ejemplo, se puede encontrar la librería de Josh Datko que ha creado una plataforma llamada Cryptotronix basada en uno de estos módulos.

3.4.1 ATSHA204A

El chip ATSHA204A está destinado a la autenticación de dispositivos hardware. Implementa el algoritmo hash SHA-256 con autenticación de mensajes MAC y opciones de autenticación HMAC lo cual hace que el chip pueda autenticar dispositivos. Este chip también cuenta con resistencia a ataques de tipo tampering.

El chip contiene un array de memoria EEPROM (Electrically Erasable Programmable Read-Only Memmory) que se puede utilizar para el almacenamiento de llaves, lectura y escritura de datos diversos, modo solo lectura y guardado de datos secretos entre otros. El acceso a esta memoria puede ser restringido en cada sección además de contar con una sección OTP (One Tieme Programmable). La memoria se divide en tres secciones. La primera es la sección de datos, contiene 512 bytes divididos en 16 registros de propósito general. El acceso de cada uno de estos registros se puede configurar en la segunda zona de memoria. La segunda sección de memoria es la sección de configuración, consta de 88 bytes EEPROM que contienen el número de serie y otra información de identificación del chip. Aquí se

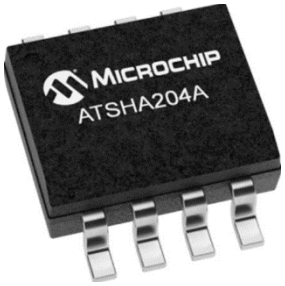


Figura 21 – ATSHA204A
Fuente: [18]

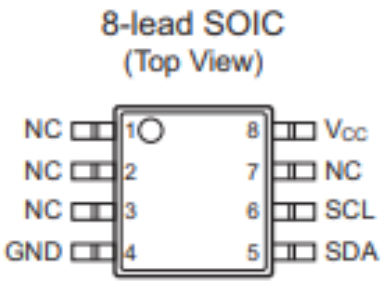


Figura 22 – Esquema de pines I2C
Fuente: [19]

pueden configurar los permisos de acceso a cada sección de memoria además de la política de acceso a la que responde cada sección. Esta información es modificable hasta que se bloquea la configuración. Por último, está la zona OTP de 64 bytes en la que se puede escribir hasta que se bloquean los datos.

En la *Figura 22* se puede ver la disposición de los pines. Solamente se utilizan 4 de los 8 pines disponibles. Estos pines son para el uso del bus I2C y para alimentación, siendo GND tierra, Vcc alimentación, SCL reloj del bus y SDA la línea de datos. Además de por bus I2C se puede comunicar utilizando una interfaz SWI para la cual se ignoraría la señal SCL.

Cada chip ATSHA204A cuenta con un número de serie de 9 bytes el cual se garantiza por la compañía de que es único. Esto permite que, utilizando protocolos criptográficos soportados por el chip, se pueda identificar desde un host si el chip es auténtico o no. También cuenta con un generador de números aleatorios de 32 bytes de gran calidad. Genera los números de forma completamente independiente de los números anteriormente generados. Cuenta con mecanismos de resistencia al tampering como encriptación de la memoria interna o detección de ataques tampering al voltaje o a la temperatura además de protección ante ataques de tampering usando glitches.

Este chip soporta el típico protocolo desafío-respuesta, en el que se utiliza un algoritmo SHA-256 o HMAC/SHA-256. Una implementación básica podría ser el envío desde el host de un desafío a el dispositivo cliente. El dispositivo cliente combina el desafío con una llave secreta usando el comando MAC proporcionado por el sistema y se lo envía de nuevo al host. Se utiliza un algoritmo de hash criptográfico para la combinación. El uso de este algoritmo hash es lo que aporta seguridad ya que, no puede obtener la llave secreta mientras que el host sí que puede verla. Esta operación básica puede ser expandida de distintas formas incluyendo funcionalidades del chip. Se puede incluir desde formas de autenticación de dispositivo hasta modificadores de la llave secreta.

Se puede encontrar la librería disponible en GitHub [17]. El chip se puede comprar en la página de Microchip [18] por menos de 1€. En la documentación se puede encontrar más información [19].

3.4.2 ATAES132A

El chip ATAES132A permite la autenticación, encriptación y desencriptación utilizando llaves guardadas de forma segura dentro del chip. Utiliza un motor hardware criptográfico AES que permite el uso eficiente de llaves AES-CCM de 128 bits que permiten la encriptación y la autenticación de los datos usando llaves privadas. Este chip también es resistente a ataques de tipo tampering.

El chip cuenta con una memoria EEPROM que a la vez proporciona autenticación y almacenamiento no volátil de datos seguro. La memoria se divide en cuatro zonas diferenciadas. En la primera zona se encuentra la memoria de usuario, son 32kb de memoria divididos en 16 zonas de 2kb cada una. Cada una de estas zonas tiene un espacio asignado en la memoria de configuración. Solo puede ser accedida cuando los criterios establecidos en la memoria de configuración lo permiten. La segunda zona es la

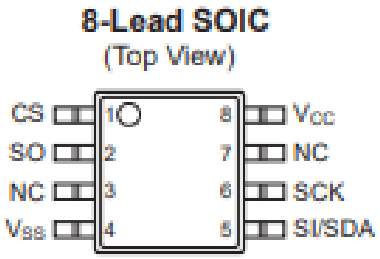


Figura 23 – Esquema de pines I2C + SPI
Fuente: [22]

memoria de llaves, en esta zona se pueden almacenar de forma segura 16 llaves de 128 bits de longitud. Cada llave tiene asociado un espacio en la memoria de configuración que permite establecer criterios de acceso. Esta memoria se puede escribir antes de ser bloqueada. Además, hay una función que permite convertir memoria de usuario en memoria de llaves, de esta forma se podrían almacenar hasta 8 llaves en cada espacio de memoria de usuario. La tercera zona es la zona de configuración, en esta zona se pueden establecer los permisos de acceso a la memoria de usuario, las restricciones del uso de las llaves y las restricciones del uso de los contadores. Es en esta zona donde se almacena el identificador único del chip establecido de fábrica. Por último, esta zona incluye 16 contadores que se pueden utilizar para aumentar la seguridad. La cuarta y última zona de memoria es de tipo SRAM (Static Random Access Memory) por lo que es volátil. Se utiliza para guardar información de estado y datos temporales.

En la *Figura 23* se puede ver el esquema de pines del chip ATAES132A. Este esquema es similar al del chip anteriormente comentado ATSHA204A, pero al permitir conexión mediante bus SPI difiere un poco. En este caso, tiene el pines SDA para la entrada de datos del bus I2C, SCK que es el reloj del bus SPI y del bus I2C, Vcc para la entrada de corriente, Vss que es la tierra (en vez de GND), CS que sirve para la selección de chip y SO que sirve para la salida de datos.

Al igual que el chip ATSHA204A, el chip ATAES132A cuenta con un número de serie de 9 bytes el cual se garantiza por la compañía de que es único. Esto permite que, utilizando protocolos criptográficos soportados por el chip, se pueda identificar desde un host si el chip es auténtico o no.

Cuenta con mecanismos de protección contra ataques de tipo tampering que evitan que el atacante pueda obtener datos internos. Incluye detección de intentos de ataques con voltaje, temperatura, frecuencia y luz además de tener una capa metálica encima de la circuitería que ayuda a la detección de ataques. También cuenta con la memoria encriptada.

Se utilizan llaves de tipo AES-CCM que permiten a la vez encriptación y autenticación por medio del uso de un código MAC con bytes para la autenticación. Se pueden escribir llaves generadas en el host, puede generar él mismo sus propias llaves de forma aleatoria. Puede encriptar y desencriptar hasta 32 bytes de datos de una sola vez. Cuenta con un kit de instrucciones fijo, de modo que no se pueden añadir instrucciones. Pese a ello, la funcionalidad es flexible ya que se pueden habilitar y deshabilitar las funcionalidades ajustándose a la situación.

Se puede encontrar la librería disponible en GitHub [20]. El chip se puede comprar en la página de Microchip [21] por menos de 1€. En la documentación se puede encontrar más información [22].

3.4.3 ATECC608B

El chip ATAES132A combina un almacenamiento de llaves seguro con aceleradores criptográficos hardware que implementan protocolos de autenticación y de encriptación. Entre estos protocolos se pueden encontrar SHA-256, HMAC/SHA256, ECDSA, ECDH y AES. Estos protocolos permiten la

autenticación de los dispositivos y la encriptación de los datos con clave publica y con clave privada. El chip es resistente a ataques de tipo tampering.

El chip cuenta con una memoria EEPROM que a la vez proporciona autenticación y almacenamiento no volátil de datos seguro. La estructura de la memoria se parece a la del chip ATSHA204 ya que contiene tres zonas similares. La primera zona es la de datos, tiene un tamaño de 1208 bytes divididos en 16 registros de propósito general que pueden ser de solo lectura o de escritura/lectura. Pueden ser utilizados para almacenar llaves (públicas o privadas), firmas, certificados, numero del modelo y otra información típicamente relacionada con el dispositivo al que está conectado el chip. La segunda zona de memoria es la de configuración, tiene un tamaño de 128 bytes en donde está contenido el número de serie y otra información de identificación. Aquí es donde se regula el acceso a los registros de la zona de datos por secciones. En la tercera zona de la memoria se encuentra una memoria OTP de 64 bytes. Antes de ser bloqueados estos bytes se pueden escribir de forma normal. Después del bloqueo el acceso es de solo lectura.

El esquema de pines es el que se puede encontrar en la *Figura 22* y su descripción es idéntica a la del chip ATSHA204A ya que solo utiliza el bus I2C.

De fábrica viene con un número de serie único de 72 bits que no se puede cambiar al igual que en los otros chips. Este número de serie se puede utilizar para la autenticación del chip.

Cuenta con mecanismos de protección contra ataques de tipo tampering que evitan que el atacante pueda obtener datos internos. Incluye detección de intentos de ataques con voltaje, temperatura, frecuencia y luz además de tener una capa metálica encima de la circuitería que ayuda a la detección de ataques. También cuenta con la memoria encriptada.

El chip ATECC608B implementa una solución completa en criptografía de clave asimétrica utilizando criptografía de curva elíptica y protocolos de seguridad ECDSA (Elliptic Curve Digital Signature Algorythm). Cuenta con elementos para soportar el ciclo de vida completo de las llaves, desde la generación de llaves de alta calidad pasando por la generación de firmas ECDSA, acuerdo de llaves ECDH (Elliptic-Curve Diffie-Hellman) y verificación de firma de llaves publicas ECDSA. Además de las llaves generadas también puede utilizar llaves externas ECC. Se pueden generar llaves privadas dentro del chip garantizándose que no se pueden obtener desde el exterior de ninguna forma. El chip ATECC608B también implementa AES-128 y SHA256.

Se puede encontrar la librería disponible en GitHub [17]. El chip se puede comprar en la página de Microchip [23] por menos de 1€. En la documentación se puede encontrar más información [24].

3.4.4 Conclusión hardware criptográfico

El hardware criptográfico es la solución calidad-precio optima ya que el precio de un chip criptográfico de los que se ha comentado en este apartado no supera 1€ [23] [21] [18], un adaptador a DIP se puede encontrar fácilmente alrededor de 1€ [25] y la placa deseada, en este caso la Raspberry Pi Zero W con un coste aproximado de 10€. A cambio se recibe una funcionalidad muy amplia, ya que cada módulo cuenta con una gran variedad de funcionalidades entre las que se puede elegir la que mejor se adapte a cada escenario. Incluso se pueden utilizar todos estos chips criptográficos al mismo tiempo para tener el mayor número de funcionalidades a la vez. La dificultad de desplegar un chip criptográfico es media ya que es necesario tener conocimiento en el campo de la informática y leer documentación, pero hay una gran base de software disponible junto con una amplia documentación proporcionada por Microchip que facilita mucho la tarea.

Esta solución es la idónea para el escenario de una red de sensores. Esto se debe a que cumple todos los requisitos propuestos ya que el hardware es resistente al tampering, proporciona métodos de encriptación y métodos de autenticación. Hace todas estas funciones con el mejor precio entre las soluciones propuestas con un amplio margen.

3.5 Comparación de soluciones

Solución	Coste de Materiales	Coste Temporal	Seguridad de la solución	Reusabilidad
Solución Empresarial	Alto	Bajo	Media	Si
Solución con modulo TPM	Medio	Medio	Alta	Si
Solución con chip criptográfico	Bajo	Medio	Media-Alta	No

Tabla 3 – Comparación de soluciones

En el coste de los materiales se ha valorado el precio de los elementos de seguridad encontrado. En el caso de la solución empresarial se ha valorado como alto ya que es un precio muy elevado en comparación con los demás. El coste del módulo TPM es medio ya que se encuentra en un punto entre la solución empresarial y el chip criptográfico siendo un precio razonable. Mientras tanto el chip criptográfico tiene un coste bajo ya que es difícilmente mejorable.

En el coste temporal se ha valorado el tiempo y la dificultad que conlleva desplegar los elementos de seguridad en una primera instancia. En el caso de la solución empresarial se ha valorado como bajo ya que la solución es Plug & Play. En el caso del módulo TPM y del chip criptográfico se ha valorado como medio ya que requiere un trabajo de investigación además de conocimientos en la materia.

En la seguridad de la solución se ha valorado la robustez ante un ataque y las funcionalidades que ofrece. En el caso de la seguridad de la solución empresarial ofrece una seguridad y unas funcionalidades aceptables. La solución con un módulo TPM es la más completa ya que estos módulos están diseñados específicamente para este propósito, en caso de un ataque físico son más resistentes que los chips criptográficos. Los chips criptográficos son muy buena opción ya que proporcionan un gran nivel de seguridad además de haber varios chips con distintas funcionalidades que en caso de combinarlas pueden ser muy potentes.

En la reusabilidad se valora la capacidad de los elementos de seguridad de moverse de un dispositivo a otro. En el caso de la solución empresarial y del módulo TPM se pueden cambiar sin ningún problema. Sin embargo, en el caso de los chips criptográficos, al tener memoria OTP y una configuración que no se puede cambiar, no son reutilizables. Por el contrario, el adaptador a DIP sí que lo puede ser si el que se compra es reutilizable.

Después de esta comparación de soluciones se puede concluir que la solución óptima para la implementación de una red de sensores es utilizando chips criptográficos. Esto se debe a que proporcionan las características de seguridad necesarias para tener un entorno seguro a la vez que mantienen un coste realmente bajo. Pese al hecho de que no son chips reutilizables, al ser tan baratos, el coste que supone reemplazarlos por unos nuevos es bajo.

4. PROPUESTA DE SOLUCIÓN BASADA EN CHIP CRIPTOGRÁFICO

En este apartado se va a proponer una solución basada en uno de los chips criptográficos comentados en el apartado anterior. Se ha elegido esta opción ya que como se ha visto es la óptima en cuanto a la relación coste-prestaciones.

4.1 Materiales hardware

En este apartado se comenta la selección de materiales justificándola con lo visto anteriormente en el documento. Se debe de elegir el chip criptográfico que se va a utilizar, la plataforma en la que se va a implementar y el sensor que va a utilizar.

4.1.1 Selección de chip

De entre los tres chips comentados anteriormente, se va a elegir el chip ATECC608B debido a que aporta a la vez las características de los chips ATAES132A y ATSHA204A. Puede utilizar al mismo tiempo autenticación por medio de hashes SHA-256, encriptación utilizando clave privada AES o ECDH y clave publica utilizando ECDSA. Además de ser resistente al tampering. Se puede ver en la

Command	Opcode	Description
AES	0x51	Execute the AES-ECB Encrypt or Decrypt functions. Calculate a Galois Field Multiply.
CheckMac	0x28	Verify a MAC calculated on another CryptoAuthentication device.
Counter	0x24	Read or increment one of the monotonic counters
DeriveKey	0x1C	Derive a target key value from the target or parent key.
ECDH	0x43	Generate an ECDH master secret using stored private key and input public key.
GenDig	0x15	Generate a data digest from a random or input seed and a key.
GenKey	0x40	Generate an ECC public key. Optionally generate an ECC private key.
Info	0x30	Return device state information.
KDF	0x56	Implement the PRF or HKDF key derivation functions
Lock	0x17	Prevent further modifications to a zone or slot of the device.
MAC	0x08	Calculate response from key and other internal data using SHA-256.
Nonce	0x16	Generate a 32-byte random number and an internally stored Nonce.
PrivWrite	0x46	Write an ECC private key into a slot in the Data zone.
Random	0x1B	Generate a random number.
Read	0x02	Read four bytes from the device, with or without authentication and encryption.
SecureBoot	0x80	Validate code signature or code digest on power-up
SelfTest	0x77	Test the various internal cryptographic computation elements
Sign	0x41	ECDSA signature calculation.
SHA	0x47	Computes a SHA-256 or HMAC digest for general purpose use by the system.
UpdateExtra	0x20	Update bytes 84 or 85 within the Configuration zone after the Configuration zone is locked.
Verify	0x45	ECDSA verify calculation.
Write	0x12	Write 4 or 32 bytes to the device, with or without authentication and encryption.

Figura 24 – Comandos disponibles ATECC608B Fuente: [24].

Figura 21 el chip ATSHA204A que exteriormente es idéntico a el elegido, a diferencia de la inscripción del chip. El chip se puede comprar en la página de Microchip [21] por menos de 1€

En la *Figura 24* se pueden ver los comandos que ofrece el chip ATECC608B. Se pueden encontrar comandos para la verificación de MAC, encriptación y desencriptación AES, generación de llaves publicas ECC, generación de llaves maestras ECDH y generación de números aleatorios entre otras muchas funciones.

Todos estos comandos proporcionan una gran versatilidad al chip criptográfico ATECC608B que le permiten desempeñar casi todo tipo de tareas criptográficas.

4.1.2 Selección de plataforma

La plataformas consideradas para la implementación son Arduino y Raspberry Pi. Esto se debe a que como se ha comentado en otros capítulos, estas plataformas son ampliamente utilizadas en el entorno de los dispositivos IoT. Ambas plataformas aportan una buena relación funcionalidad-precio. Sin embargo, para esta aplicación se busca una plataforma que sea lo más barata posible mientras que mantenga un mínimo de elementos que faciliten la implementación. La placa elegida debe tener Wifi ya que el objetivo es implementar un dispositivo IoT. Dentro de la gama de productos de Arduino se pude encontrar la placa Arduino uno Wifi. Esta placa tiene un coste de 39€ [26] en la página oficial de Arduino. Sin embargo, se pueden encontrar placas compatibles con el entorno Arduino y que incluyen Wifi como las placas de NodeMCU. Estas placas tienen un coste alrededor de los 2€ [27] y son aptas para el uso como dispositivo IoT. Mientras tanto en la gama de productos Raspberry Pi hay varias opciones disponibles. La opción más interesante sin duda alguna es la de la Raspberry Pi Zero W. Esta placa cuenta con Wifi y tiene un tamaño reducido haciéndola más interesante. Aunque hay una gran diferencia en precio respecto de la placa NodeMCU ya que cuesta más de 10€ [16]. Además, la compra de estas placas está limitada a una por persona lo que hace que esta placa no pueda ser utilizada a gran escala en una red de sensores. A la hora de llevar esta solución a gran escala se podría sustituir la Raspberry Pi Zero W por una Raspberry Pi 3 A+ que cuesta alrededor de 27€ [28].

Después de ver esta comparación la elección es clara, se elegirá una placa NodeMCU debido a la gran diferencia de precio. Concretamente se elegirá la placa NodeMCU v3 que es la versión más reciente, se puede ver la placa en la *Figura 25*. Como se ha comentado previamente, tiene un coste de alrededor de 2€ [27]. Esta placa cuenta con el módulo ESP8266 que proporciona Wifi.

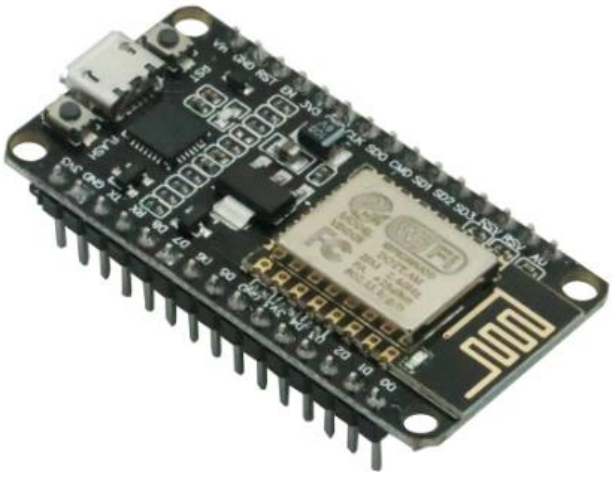


Figura 25 – NodeMCU v3 Fuente: [27].

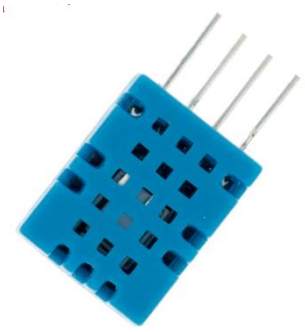


Figura 26 – Sensor DHT11
Fuente: [29]

4.1.3 Selección del sensor

En cuanto al sensor, se ha elegido un sensor de temperatura ya que es un tipo de sensor muy común. Dentro de los sensores de temperatura hay muchas gamas de precio, se ha optado por uno de gama baja con un precio reducido. Concretamente se ha elegido el sensor DHT11 que se puede ver en la *Figura 26*. Este sensor tiene un coste de poco más de 1€ [29]. Es un sensor de una calidad media-baja, permite realizar medidas de temperatura y de humedad proporcionando una salida de datos digital. Puede medir valores de temperatura entre 0 y 50ºC y valores de humedad entre 20 y 90%.

4.1.4 Otros materiales hardware

Además de los materiales hardware principales ya mencionados, es necesario hacer uso de otros materiales hardware para conectar todo. Estos materiales son cables, que tienen un precio de un poco más de 3€ [30] un kit de 10 cables, una placa protoboard para facilitar la conexión, se puede encontrar por alrededor de 5€ [31] y un adaptador de SOIC8 a DIP8 para facilitar la conexión del módulo criptográfico, se puede encontrar fácilmente por alrededor de 1€ [25].

4.2 Materiales software

En este apartado se comentarán la librería software que se van a utilizar para poner en funcionamiento la plataforma NodeMCU, la librería para gestionar el Wifi, la librería para gestionar el sensor de temperatura DHT11 y la librería para gestionar el chip criptográfico.

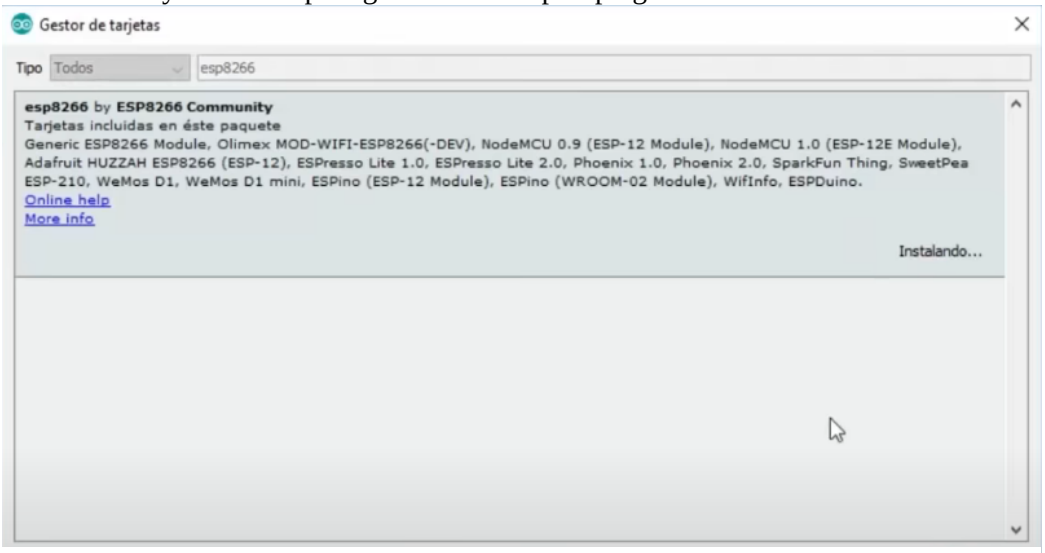


Figura 27 – Instalación de la librería ESP8266 Fuente: Elaboración propia

4.2.1 Librería core Arduino para ESP8266

Para la programación de la placa se utilizará la plataforma Arduino IDE. Para ello primeramente hay que descargar la aplicación de Arduino directamente desde la página oficial [32].

Una vez abierta la aplicación hay que seleccionar la placa que se está utilizando, como esta placa no es oficial de Arduino hay que añadirla. Para ello hay que añadir el link al archivo *.json* de la librería [33] para que sepa de donde descargar la librería. Se entra en herramientas en el gestor de tarjetas y se instala la librería tal y como se muestra en la *Figura 27*. A continuación, ya se puede seleccionar la placa para programar.

4.2.2 Librería del sensor DHT11

Esta librería contiene todas las funciones necesarias para manejar el sensor de temperatura. Las funciones más importantes son la de leer la temperatura y la de leer la humedad. Se puede encontrar la librería en la página oficial de Arduino [34]. Una vez descargada solo hay que añadirla y ya está lista para usar.

4.2.3 Librería de Wifi

La librería para utilizar el Wifi ya viene instalada por defecto con la librería ESP8266, se llama WiFiClient. Hay que configurarla para que se pueda conectar a la red wifi. Para ello hay que declarar variables para establecer el nombre de la red, contraseña y servidor al que se quieren subir los datos.

```
int serialNumber(byte sn[]);
String serialNumber();

long random(long max);
long random(long min, long max);
int random(byte data[], size_t length);

int generatePrivateKey(int slot, byte publicKey[]);
int generatePublicKey(int slot, byte publicKey[]);

int ecdsaVerify(const byte message[], const byte signature[], const byte pubkey[]);
int ecSign(int slot, const byte message[], byte signature[]);

int beginSHA256();
int updateSHA256(const byte data[]); // 64 bytes
int endSHA256(byte result[]);
int endSHA256(const byte data[], int length, byte result[]);

int readSlot(int slot, byte data[], int length);
int writeSlot(int slot, const byte data[], int length);

int locked();
int writeConfiguration(const byte data[]);
int readConfiguration(byte data[]);
int lock();
```

Figura 28 – Funciones de la librería ECCX08 Fuente: Elaboración propia

4.2.4 Librería del chip criptográfico

La librería que permite utilizar el chip criptográfico está hecha por Arduino. Hay que añadirla de la misma forma que se ha añadido la librería core para ESP8266. Esta librería contiene un gran número de funciones que se pueden ver en la *Figura 28*. Entre estas funciones se pueden encontrar funciones para la gestión de llaves públicas y privadas, la autenticación con algoritmos SHA y el uso y verificación de firma digital de curva elíptica.

4.3 Conexión de la placa al sensor y al chip criptográfico

En este apartado se comentará la forma en la que están conectados el sensor de temperatura y el chip criptográfico con la placa.

4.3.1 Conexión chip criptográfico-placa

Esta conexión se realiza a través del bus I2C. Este bus realiza el envío de datos de forma síncrona. Funciona con un esquema de maestro-esclavo. Solo necesita dos líneas de señal y una masa. Estas líneas son el reloj que sincroniza el sistema (SCL) y la línea de datos (SDA).

El proceso de comunicación en el bus I2C es siempre iniciado por un maestro. Para iniciar la comunicación el bus tiene que estar libre, esto ocurre cuando la señal SDA y la señal SCL están en estado lógico alto. Entonces, se manda una señal start que consiste en poner en estado bajo la señal SDA. A continuación, se envía una cadena de 8 bits que indica el dispositivo que se quiere seleccionar (7 bits) y si se requiere para una operación de lectura o escritura (1 bit). Si el dispositivo requerido está conectado al bus responde con un ACK. A partir de aquí se inicia la transmisión de datos en un sentido u otro dependiendo del valor del bit de lectura/escritura.

En la *Figura 29* se puede ver a la izquierda como está conectado el chip criptográfico a la placa. La alimentación se conecta a un puerto de alimentación cualquiera ya que todos son de 3V, la tierra también a una tierra cualquiera. Mientras tanto la señal SCL se conecta al pin D1 y la señal SCK se conecta al pin D2.

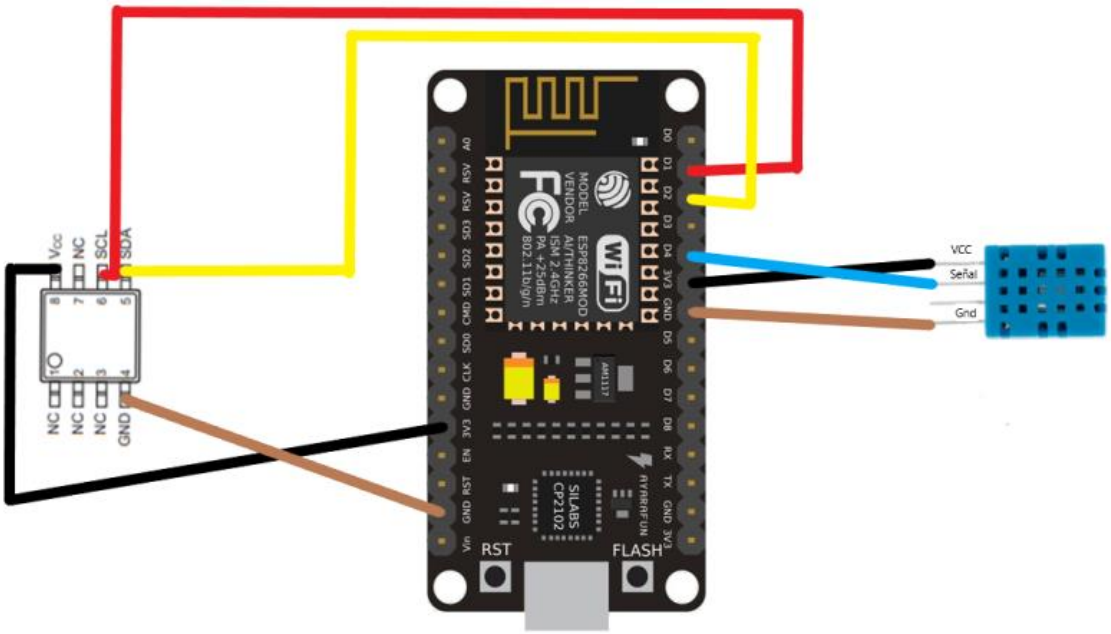


Figura 29 – Esquema de conexiones Fuente: Elaboración propia

4.3.2 Conexión sensor de temperatura-placa

Este sensor utiliza una comunicación de un solo cable. El proceso de comunicación se divide en tres pasos. La comunicación la inicia la placa mandado una petición al sensor para que le envíe datos. Este responde con una cadena de 40 bits o 5 bytes. Los 2 primeros bytes contienen la humedad, los bytes 3 y 4 contienen la temperatura y el byte 5 es un checksum.

4.4 Funcionamiento

Para poner en funcionamiento un sensor es necesario tener todas las conexiones hechas tal y como se ha explicado en el anterior apartado. A continuación, se instalan las librerías mencionadas anteriormente. Finalmente se tiene que personalizar el chip criptográfico. Aquí principalmente hay cuatro pasos, el primero consiste en personalizar la zona de configuración, a continuación, se bloquea esta zona. Si no se bloquea la zona no se puede tener acceso a las secciones de memoria de datos ni OTP. El tercer paso consiste en personalizar las zonas de datos y OTP. Finalmente se bloquean estas zonas. Se puede encontrar información más detallada al respecto en la documentación [35]. Con esto ya estaría listo el entorno para la creación de un programa.

El programa que se debe desarrollar hará uso de las directivas de lectura de datos del sensor de temperatura. Seguidamente, utilizando la librería del Wifi se enviarán los datos a el servidor. De esta forma se completará el envío de información, esto se puede hacer con un bucle que realice las mediciones y envíos cada cierto tiempo.

No se debe de olvidar que hay que securizar el dispositivo. Para ese debe encriptar la información que se envía a través de Wifi, además de autenticar el dispositivo.

Para autenticar el dispositivo, al comenzar la conexión con el nodo administrador, se le puede enviar el número de serie de la placa que es único. Este número se puede conseguir utilizando el método *SerialNumber()* proporcionado en la librería. Otra forma de autenticación es utilizando un certificado autofirmado. La librería viene con un código que permite generar un certificado autofirmado, para ello primero hay que crear una llave privada utilizando la función *generatePrivateKey(int slot, byte publicKey[])*. A continuación, se ejecuta el código proporcionado por la librería [36] el cual pide los datos necesario para generar el certificado autofirmado y finalmente lo genera. También se pueden verificar certificados utilizando el comando *ecdsaVerify(const byte message[], const byte signature[], const byte pubkey[])*. Con este conjunto se puede realizar la verificación de nodos.

Para la encriptación de mensajes se puede utilizar el algoritmo SHA-256. Primero se hashean los datos a encriptar utilizando la función. Para ello se debe de utilizar la función *beginSHA256()* que inicializa en la memoria el contexto de SHA. A continuación, se ejecuta el comando *updateSHA256(cosnt byte data[])*, a este comando se le pasa como parámetro el mensaje a hashear en el parámetro *data*. Finalmente se utiliza el comando *endSHA256(byte result[])* que devuelve el digest. Se utiliza este digest generado por el algoritmo para realizar una operación XOR con los datos a encriptar. Una vez encriptados estos datos, se pueden enviar de forma segura. Para desencriptar estos datos en el nodo administrador se tiene que realizar la operación inversa.

Otra forma de encriptar la comunicación es implementando algo similar a TLS/SSL. Para ello el nodo administrador debe tener un certificado, una clave privada y una clave publica, los dos primeros puede generarlos de la forma que se ha comentado anteriormente. Mientras tanto la clave publica la puede generar a partir de la clave privada utilizando otro código proporcionado por la librería [37]. Este código preguntará que llave privada quieres utilizar o si por el contrario se prefiere generar una nueva. Para realizar el handshake primeramente el nodo sensor envía una petición al nodo administrador. Este nodo

le responde enviando su certificado. El nodo sensor verifica el certificado, de la forma anteriormente vista. Para realizar el cambio de llaves privadas, el nodo sensor genera una llave privada, de la forma vista anteriormente, la encripta utilizando la llave publica y se lo envía al nodo administrador. Para esta encriptación se tiene que utilizar el comando AES del chip criptográfico el cual hay que implementar utilizando la información incluida en la documentación [24] en el apartado 11.1. Finalmente, el nodo administrador desencripta el mensaje con la llave privada pareja de la llave publica usada para encriptar el mensaje. Ahora ya tienen los dos extremos un par de llaves privadas intercambiadas de forma segura.

Utilizando estos mecanismos se puede alcanzar la securización de la comunicación del sistema y la autenticación de los sensores.

4.5 Presupuesto

En el presupuesto solo se incluye el coste de los materiales. Los costes del desarrollo y las horas de trabajo de la implementación no se incluyen ya que se está haciendo una propuesta de solución. Se tendrán en cuenta los gastos de comprar el hardware en tiendas normales, buscando los precios más bajos. No se tendrán en cuenta los gastos de envío.

Cantidad	Objeto	Precio Unidad
1	Chip ATECC608B	0,68€ [21]
1	NodeMCU v3	2,08€ [27]
1	Sensor DHT11	1,20€ [29]
1	Kit de 10 cables	3,35€ [30]
1	Placa protoboard	4,99€ [31]
1	Adaptador SOIC8 a DIP8	1,04€ [25]
	TOTAL	13,34€

Tabla 4 – Presupuesto hardware

El precio total de los componentes hardware es de 13,34 esto lo hace un presupuesto realmente bajo para implementar un sensor. En caso de querer implementar una red de sensores con una gran cantidad de ellos, los costes se abaratarían al comprar componentes en mayor cantidad.

5. CONCLUSIÓN Y LÍNEAS FUTURAS

Este documento ha tenido como objetivos principales el análisis de la seguridad de los dispositivos IoT en las redes de sensores y la búsqueda de una solución basada en hardware criptográfico. La idea inicial era implementar la solución propuesta. Sin embargo, debido a una falta de tiempo causada por las asignaturas del último curso, juntándose con la fecha límite para la matriculación de un master que quiero cursar el curso que viene no ha sido posible la implementación física de la propuesta. A cambio, se ha realizado un análisis más en profundidad de las vulnerabilidades de las redes de sensores, se han buscado otras soluciones alternativas que proponen distintas formas de solucionar el problema de la seguridad y se ha propuesto una solución de forma teórica que aporta una muy buena base en caso de quererse implementar de forma práctica.

En cuanto a las conclusiones, se pueden resumir en las siguientes ideas.

La seguridad en el mundo IoT es imprescindible para que estos dispositivos puedan seguir con su desarrollo sin poner en peligro a los usuarios. Es imprescindible que se tomen medidas y que se haga de la forma más rápida posible, ya que cuanto más tiempo se tarde en adoptar medidas de seguridad, más difícil se volverá la securización de los dispositivos IoT.

La solución propuesta para poner resolver los problemas de seguridad de los sensores IoT basada en un chip criptográfico es realista. Esto se debe a que aporta unas características de seguridad mas que suficientes para securizar un sensor IoT. Utiliza una placa basada en Arduino que es una plataforma muy extendida, lo que hace que la solución pueda ser útil para muchos usuarios en muchos campos distintos, incluso fuera del mundo de los sensores. Los costes de los materiales para la implementación son muy bajos lo que hace que no sean un impedimento a la hora de tomar la decisión de securizar un sensor IoT.

En el futuro seria interesante implementar de forma física la propuesta realizada en este documento. Al hacer un despliegue real de este sensor, se puede ver su desempeño en una situación real con ataques reales. Sería recomendable probar ataques contra el sensor para buscar posibles agujeros en la seguridad, parchearlos y repetir el proceso hasta que no se encuentren más vulnerabilidades para terminar de garantizar la seguridad del sensor. Puede ser interesante también ver los consumos de batería que tiene este sensor en un funcionamiento normal, y compararlos con los costes energéticos cuando utiliza el módulo criptográfico ya que la batería es un bien muy preciado en los sensores.

En el futuro, el mundo IoT hará con los dispositivos criptográficos lo mismo que se ha hecho con los módulos wifi, integrarlos en el diseño. Hace unos años, los dispositivos venían únicamente con puerto ethernet y si se quería utilizar de forma inalámbrica era necesario utilizar un módulo wifi. Hoy en día se puede elegir comprar placas que vienen con los módulos integrados de fábrica. Con los módulos criptográficos en los dispositivos IoT va a ocurrir lo mismo, dentro de unos años existirán placas que incluyan módulos criptográficos instalados de fábrica. El hecho de incluir los módulos criptográficos en las fases de diseño de los dispositivos puede hacer que se integren los módulos dentro del chip principal ahorrando espacio. Al venir de fabrica también serán más baratos que comprados por separado y se pueden implementar mejor métodos de ahorro de. Cuando las placas traigan módulos criptográficos de fabrica las situaciones de hackeos masivos se verán reducidas

BIBLIOGRAFÍA

[1] Knud Lasse Lueth, «IoT Analytics,» 2020. [En línea]. Available: https://iot-analytics.com/state-of-the-iot-2020-12-billion-iot-connections-surpassing-non-iot-for-the-first-time/. [Último acceso: 2021].

[2] Arne Holst, «Statista,» 2021. [En línea]. Available: https://www.statista.com/statistics/1194709/iot-revenue-worldwide/. [Último acceso: 2021].

[3] Alex Scroxton, «computerweekly.com,» 2021. [En línea]. Available: https://www.computerweekly.com/news/252497593/Attack-on-surveillance-cameras-a-warning-over-security-ethics?_gl=1*545obv*_ga*MTY0ODkzOTUyOC4xNjIzMDYyNDY5*_ga_TQKE4GS5P9*MTYyNDg4NjQ3NC44LjAuMTYyNDg4NjQ3NC4w&_ga=2.227763677.1745827258.1624830843-1648939528.. [Último acceso: 2021].

[4] G. I. o. T. Manos Antonakakis, A. Tim April, U. o. I. U.-C. Michael Bailey, U. o. M. A. A. Matt Bernhard, G. Elie Bursztein, C. Jaime Cochran y Z. D. a. J. Al, «usenix.org,» 2017. [En línea]. Available: https://www.usenix.org/system/files/conference/usenixsecurity17/sec17-antonakakis.pdf. [Último acceso: 2021].

[5] iot-analytics, «iot-analytics,» [En línea]. Available: https://iot-analytics.com/state-of-the-iot-2020-12-billion-iot-connections-surpassing-non-iot-for-the-first-time/. [Último acceso: 2021].

[6] E. Edwards, «thomasnet.com,» 2021. [En línea]. Available: https://www.thomasnet.com/articles/instruments-controls/types-of-internet-of-things-iot-sensors/. [Último acceso: 2021].

[7] M. Leonard, «supplychaindive,» 2019. [En línea]. Available: https://www.supplychaindive.com/news/declining-price-iot-sensors-manufacturing/564980/. [Último acceso: 2021].

[8] F. Zhang, W. He y X. Liu, «ieeexplore.ieee.org,» 2011. [En línea]. Available: https://ieeexplore.ieee.org/document/5961736. [Último acceso: 2021].

[9] C. V. Wright, S. E. Coull y F. Monrose, «ndss-symposium.org,» 2017. [En línea]. Available: https://www.ndss-symposium.org/wp-content/uploads/2017/09/wright.pdf. [Último acceso: 2021].

[10] M. A. Hamid, M. Mamun-Or-Rashid y C. S. Hong, «networking.khu.ac.kr,» 2006. [En línea]. Available: http://networking.khu.ac.kr/gallery/entry/layouts/net/publications/data/Routing%20Security%20in%20Sensor%20Network%20HELLO%20Flood%20Attack%20and%20Defense.pdf. [Último acceso: 2021].

[11] V. P. Illiano y E. C. Lupu, «core.ac.uk,» 2015. [En línea]. Available: https://core.ac.uk/download/pdf/77002559.pdf. [Último acceso: 2021].

[12] cdw, «cdw,» [En línea]. Available: https://www.cdw.com/product/extreme-defender-for-iot-defender-adapter-sa201-security-appliance/5414609. [Último acceso: 2021].

[13] Amazon, «Amazon,» 2021. [En línea]. Available: https://www.amazon.es/DELL-Aerohive-AP150W-1300Mbit-Blanco/dp/B07BFSKFYH. [Último acceso: 2021].

[14] J. A. e. al., «github,» 26 abril 2021. [En línea]. Available: https://github.com/tpm2-software/tpm2-tss. [Último acceso: 2021].

[15] digikey, «digikey,» 2015. [En línea]. Available: https://www.digikey.es/product-detail/en/IRID9670TPM20LINUXTOBO1/IRID9670TPM20LINUXTOBO1-ND/9922568?utm_campaign=buynow&utm_medium=aggregator&curr=eur&utm_source=octopart. [Último acceso: 2021].

[16] kubii, «kubii,» 2017. [En línea]. Available: https://www.kubii.es/raspberry-pi-3-2-b/1851-raspberry-pi-zero-w-kubii-3272496006997.html. [Último acceso: 2021].

[17] JuergenReppSIT y A. e. al., «GitHub,» mayo 2017. [En línea]. Available: https://github.com/tpm2-software/tpm2-tss. [Último acceso: 2021].

[18] Microchip, «Microchip,» [En línea]. Available: https://www.microchip.com/wwwproducts/en/atsha204a. [Último acceso: 2021].

[19] Microchip, «MIcrochip,» [En línea]. Available: https://ww1.microchip.com/downloads/en/DeviceDoc/ATSHA204A-Data-Sheet-40002025A.pdf. [Último acceso: 2021].

[20] diogorac, eckelj y dogusural, «GitHub,» Noviembre 2019. [En línea]. Available: https://github.com/RiddleAndCode/ATAES132. [Último acceso: 2021].

[21] Microchip, «Microchip,» [En línea]. Available: https://www.microchip.com/wwwproducts/en/ATECC608B. [Último acceso: 2021].

[22] MIcrochip, «Microchip,» [En línea]. Available: https://ww1.microchip.com/downloads/en/DeviceDoc/ATAES132A-Data-Sheet-40002023A.pdf. [Último acceso: 2021].

[23] MIcrochip, «Microchip,» [En línea]. Available: https://www.microchip.com/wwwproducts/en/ataes132a#additional-features. [Último acceso: 2021].

[24] MIcrochip, «GitHub,» [En línea]. Available: https://atecc608a.github.io/ATECC608A.pdf. [Último acceso: 2021].

[25] Aliexpress, «Aliexpress,» [En línea]. Available: https://es.aliexpress.com/item/32830792620.html?spm=a2g0o.search0304.0.0.e0c77cfe5MAK6X&algo_pvid=a381968e-1407-4600-b173-0963aeb304a4&algo_exp_id=a381968e-1407-4600-b173-0963aeb304a4-0. [Último acceso: 2021].

[26] Arduino, «Arduino,» [En línea]. Available: https://store.arduino.cc/arduino-uno-wifi-rev2. [Último acceso: 2021].

[27] Aliexpress, «Aliexpress,» [En línea]. Available: https://es.aliexpress.com/item/33045361273.html?acnt=439-079-4345&aff_platform=aaf&ds_e_device=c&albcp=10191226958&ds_e_product_id=es330453612

73&ds_url_v=2&ds_dest_url=https%3A%2F%2Fs.click.aliexpress.com%2Fdeep_link.htm%3Faff_short_key%3DUneMJZVf&ds_e_pr. [Último acceso: 2021].

[28] kubii, «kubii,» [En línea]. Available: https://www.kubii.es/raspberry-pi-3-2-b/2334-raspberry-pi-3-modelo-a-kubii-652508442181.html?search_query=raspberry+pi+3&results=198. [Último acceso: 2021].

[29] Aliexpress, «Aliexpress,» [En línea]. Available: https://es.aliexpress.com/item/1005001621864387.html?spm=a2g0o.search0304.0.0.541543b3y0b2tS&aem_p4p_detail=2021071420342413000876370719200249459 59. [Último acceso: 2021].

[30] RS-online, «RS-online,» [En línea]. Available: https://es.rs-online.com/web/p/kits-de-cable-conector-para-placas-de-prueba/7916463/. [Último acceso: 2021].

[31] Amazon, «Amazon,» [En línea]. Available: https://www.amazon.es/AZDelivery-Breadboard-pruebas-contactos-Arduino/dp/B07K8PQ4B5/ref=sr_1_5?adgrpid=63021088064&dchild=1&gclid=CjwKCAjwlrqHBhByEiwAnLmYUISvvTu-w2NYMx7UI1GUwlhfjAo42A5GRG5ziOEFbuOEkKwv-lSV3BoCQLcQAvD_BwE&hvadid=275343623474&hvdev=c&hvloc. [Último acceso: 2021].

[32] Arduino, «Arduino,» [En línea]. Available: https://www.arduino.cc/en/software. [Último acceso: 2021].

[33] Arduino, «Github,» [En línea]. Available: https://github.com/esp8266/Arduino. [Último acceso: 2021].

[34] Arduino, «Arduino,» [En línea]. Available: https://www.arduino.cc/reference/en/libraries/dht-sensor-library/. [Último acceso: 2021].

[35] Atmel, «microchip,» [En línea]. Available: http://ww1.microchip.com/downloads/en/Appnotes/Atmel-8845-CryptoAuth-ATSHA204A-ATECC508A-Personalization-Guide-ApplicationNote.pdf. [Último acceso: 2021].

[36] sandeepmistry, «GitHub,» [En línea]. Available: https://github.com/arduino-libraries/ArduinoECCX08/blob/master/examples/Tools/ECCX08SelfSignedCert/ECCX08SelfSignedCert.ino. [Último acceso: 2021].

[37] sandeepmistry, «GitHub,» [En línea]. Available: https://github.com/arduino-libraries/ArduinoECCX08/blob/master/examples/Tools/ECCX08JWSPublicKey/ECCX08JWSPublicKey.ino. [Último acceso: 2021].

[38] iot5, «iot5,» [En línea]. Available: https://iot5.net/5-things-you-should-know-about-iot-devices-market/. [Último acceso: 2021].

[39] C. Garcia, «researchgate,» 2010. [En línea]. Available: https://www.researchgate.net/publication/47640073_Impacto_de_la_seguridad_en_redes_inalambricas_de_sensores_IEEE_802154. [Último acceso: 2021].

[40] R. Alonso, «Hardzone,» [En línea]. Available: https://hardzone.es/reportajes/comparativas/raspberry-pi-vs-arduino/. [Último acceso: 2021].

[41] E. Williams, «medium,» [En línea]. Available: https://medium.com/@emilywilliams_43022/cryptography-101-symmetric-encryption-444aac6bb7a3. [Último acceso: 2021].

[42] J. Campos, «javiercampos,» [En línea]. Available: https://javiercampos.es/blog/2011/07/22/el-algoritmo-de-diffie-hellman/. [Último acceso: 2021].

[43] extremenetworks, «extremenetworks,» [En línea]. Available: https://www.extremenetworks.com/product/extreme-defender-for-iot/. [Último acceso: 2021].

[44] E. networks, «youtube,» [En línea]. Available: https://www.youtube.com/watch?v=4F5KzLijJ7I. [Último acceso: 2021].

[45] Aerohive, «aerohive,» [En línea]. Available: https://docs.aerohive.com/330000/docs/help/english/ng/Content/hardware/ap/ap150w.htm. [Último acceso: 2021].

[46] infineon, «infineon,» [En línea]. Available: https://www.infineon.com/dgdl/Infineon-Iridium_1-0_9670_HD-AdditionalTechnicalInformation-v01_01-EN.pdf?fileId=5546d46271bf4f920171ef70667e51b4. [Último acceso: 2021].

[47] Microchip, «microchip,» [En línea]. Available: https://www.microchip.com/wwwproducts/en/atsha204a. [Último acceso: 2021].