Estudio de las vulnerabilidades de la tecnología Bluetooth

Eduardo Sesé Vega
Máster Universitario en Ingeniería de Telecomunicación UOC – URL
Departamento de Telemática

José López Vicario
Xavi Vilajosona Guillén

7 de enero de 2020

FICHA DEL TRABAJO FINAL

Título del trabajo:	**Estudio de las vulnerabilidades de la tecnología Bluetooth**
Nombre del autor:	*Eduardo Sesé Vega*
Nombre del consultor/a:	*José López Vicario*
Nombre del PRA:	*Xavi Vilajosona Guillén*
Fecha de entrega:	*07/01/2020*
Titulación:	*Máster Universitario en Ingeniería de Telecomunicación UOC-URL*
Área del Trabajo Final:	*Telemática*
Idioma del trabajo:	*Español*
Palabras clave	*Bluetooth, Seguridad, Vulnerabilidades*

Resumen del Trabajo

Bluetooth es una tecnología inalámbrica utilizada en el ámbito de las redes de área personal (WPAN) para comunicaciones de corto alcance en la banda de 2,4GHz. En la actualidad Bluetooth se encuentra integrado en prácticamente todos los aspectos de la sociedad, desde el mercado de consumo, con *smartphones* y demás gadgets que lo implementan, hasta el entorno industrial, pasando por aplicaciones médicas, electrodomésticos, etc.

En este documento se proporciona una descripción general de la tecnología en sus dos implementaciones principales, BR/EDR y LE, y se expone la estructura de seguridad que las componen en todas las versiones publicadas hasta ahora en la especificación, desde la generación de claves, el *bonding*, autenticación, cifrado, etc.

Aunque a lo largo de su andadura, y sobre todo a partir de la versión 2.1, se ha puesto mucho foco en mejorar la seguridad con mecanismos como SSP o *Secure Connections*, todavía quedan aspectos que mejorar o puntos débiles que pueden ser explotados por un atacante malicioso.

Por tanto, también será objeto de este trabajo profundizar en las vulnerabilidades que la aquejan, y las posibles amenazas que dichas vulnerabilidades dejan al descubierto, describiendo desde el punto de vista teórico algunos de los ataques más relevantes que ha sufrido Bluetooth en los últimos años, y recoger una lista de contramedidas o buenas prácticas que los usuarios de Bluetooth deben conocer y aplicar para mitigar el riesgo frente a las amenazas expuestas.

Abstract

Bluetooth is a Personal Area Network (WPAN) wireless technology, used in short-range communications in the 2.4GHz band. Nowadays, Bluetooth is widely used in all aspects of society, starting with the consumer market, with smartphones and other gadgets, and following with the industrial environment, through medical applications, etc.

This document provides a technical overview for its two main implementations, BR / EDR and LE, and exposes its security structure in all versions as published in the SIG specification, like the key generation, bonding, authentication, encryption, etc.

Although recent versions of Bluetooth have been focused on improving security, specially from 2.1 version with SSP, or Secure Connections in 4.1, there are still vulnerabilities and weaknesses that can suppose a threat.

Therefore, this document will also try to collect most recent Bluetooth vulnerabilities and the possible threats they allow. It also identifies the most relevant attacks that Bluetooth has suffered in recent years and collects a list of countermeasures or good practices that Bluetooth users should know and apply to mitigate the risk against exposed threats.

ÍNDICE

Lista de figuras

Lista de Tablas

1. Introducción

1.1. Contexto y justificación del Trabajo

En la actualidad, donde *smartphones* y demás *gadgets* copan el mercado de consumo y donde la automatización y la robotización son una prioridad en el entorno industrial, las tecnologías inalámbricas aplicables en el entorno IoT parecen tener cada día más relevancia. En ese contexto, Bluetooth es una de las tecnologías inalámbricas más extendidas, por lo que su relevancia para fabricantes y consumidores la hace de especial interés también para posibles atacantes que quieran explotar sus vulnerabilidades para fines maliciosos.

Aunque en las últimas versiones se ha puesto mucho foco en reforzar la seguridad de Bluetooth, como en cualquier tecnología aún quedan puntos débiles que pueden suponer una amenaza. Por tanto, en este trabajo se analizan las características de la tecnología Bluetooth y se ahonda en la arquitectura de seguridad que la compone para, una vez expuesto su funcionamiento, centrarnos en las vulnerabilidades que la aquejan, y las posibles amenazas que dichas vulnerabilidades dejan al descubierto.

Además de analizar las vulnerabilidades y amenazas de la tecnología, también se analizan, desde un punto de vista teórico, algunos de los ataques más relevantes que ha sufrido Bluetooth en los últimos años. Por último, se recoge una lista de contramedidas o buenas prácticas que los usuarios de Bluetooth deben conocer y aplicar para mitigar el riesgo frente a las amenazas expuestas, así como la descripción de dos escenarios teóricos para ejemplificar los riesgos reales de una empresa que utilice esta tecnología y las contramedidas que precisan.

1.2. Objetivos del Trabajo

Como se ha comentado, el objetivo principal de este trabajo es aportar una idea clara y detallada de cómo funciona la seguridad en Bluetooth y, en base a este funcionamiento, entender y detallar para todas las versiones de Bluetooth publicadas hasta la fecha de redacción de este documento, qué vulnerabilidades las aquejan y las amenazas que esto supone.

Por tanto, se trata de un trabajo puramente teórico, donde se pretende abarcar:

- Introducción a Bluetooth, para entender su relevancia en la actualidad, su funcionamiento, las distintas versiones publicadas y su arquitectura.
- Descripción de la arquitectura de seguridad de Bluetooth, describiendo el funcionamiento de la generación de claves, *bonding*, autenticación y cifrado para las distintas versiones e implementaciones.
- Recopilación y descripción de las vulnerabilidades de Bluetooth para las distintas implementaciones y versiones.
- Cuáles son las amenazas y los posibles ataques que estas vulnerabilidades hacen posible.

- Descripción detallada del funcionamiento de ataques relevantes, como es el caso de KNOB y BlueBorne.
- Recopilación de contramedidas que permitan mitigar o reducir el riesgo y el impacto de los ataques listados.
- Descripción de escenarios hipotéticos, a fin de aportar un ejemplo de riesgos corporativos para el uso de Bluetooth y medidas a llevar a cabo.

1.3. Enfoque y método seguido

Dado que el trabajo tiene un enfoque únicamente teórico, es fundamental alcanzar un nivel de profundidad detallado en todo lo que se expone, así como incluir aportaciones propias que demuestren comprensión en lo expuesto y reflexión sobre sus implicaciones. Para ello ha sido fundamental contar con bibliografía detallada de organismos oficiales, como la propia especificación de Bluetooth SIG de la versión 5.1.

El trabajo se ha planteado en las siguientes etapas:

- Búsqueda de información y comprensión de conceptos.
- Definición del alcance, planificación y estructuración del trabajo.
- Redacción del documento, con la información relativa a:
    - Tecnología Bluetooth**:** Características técnicas, arquitectura y versiones.
    - Seguridad en Bluetooth: Fases del proceso de establecimiento de conexión.
    - Vulnerabilidades.
    - Amenazas y ataques: recopilación general y detalle de KNOB y BlueBorne.
    - Contramedidas.
    - Escenarios prácticos.

1.4. Planificación del Trabajo

A lo largo del semestre se ha rectificado sobre la planificación inicial del trabajo, estimada durante en la PEC2, principalmente porque el alcance y el enfoque del mismo se vieron rectificados. En un primer momento se planteó como un trabajo con componente práctica, que contaría con simulaciones de ataques realizadas en un entorno de trabajo construido *ad hoc* para este proyecto, pero finalmente se ha optado por un trabajo teórico.

En la Tabla 1 se detallan las tareas realizadas para la consecución de los objetivos de este trabajo, con los entregables que han tenido lugar señalados. Dentro de las tareas contempladas en el entregable de la PEC3 puede observarse en rojo las tareas planificadas inicialmente para la simulación de ataques, y en paralelo la planificación que finalmente se ha seguido, con el análisis teórico de los ataques y amenazas.

Tal y como se aprecia, inicialmente no se contempló un periodo para incluir las correcciones de la PEC2, lo que sin embargo ha resultado en una tarea con

entidad propia. Asimismo, se recogían dentro de esta tarea finalmente desestimada, la investigación de herramientas y el aprendizaje relacionado con el entorno de pruebas a implementar, la construcción de dicho entorno, las propias pruebas de hacking y la documentación del trabajo realizado. Aunque la parte de análisis sí se llevó a cabo, y permitió tomar la decisión de desestimar este aspecto del proyecto, el resto del tiempo se enfocó en analizar los ataques y amenazas más relevantes o recientes, y describirlos desde un punto de vista teórico.

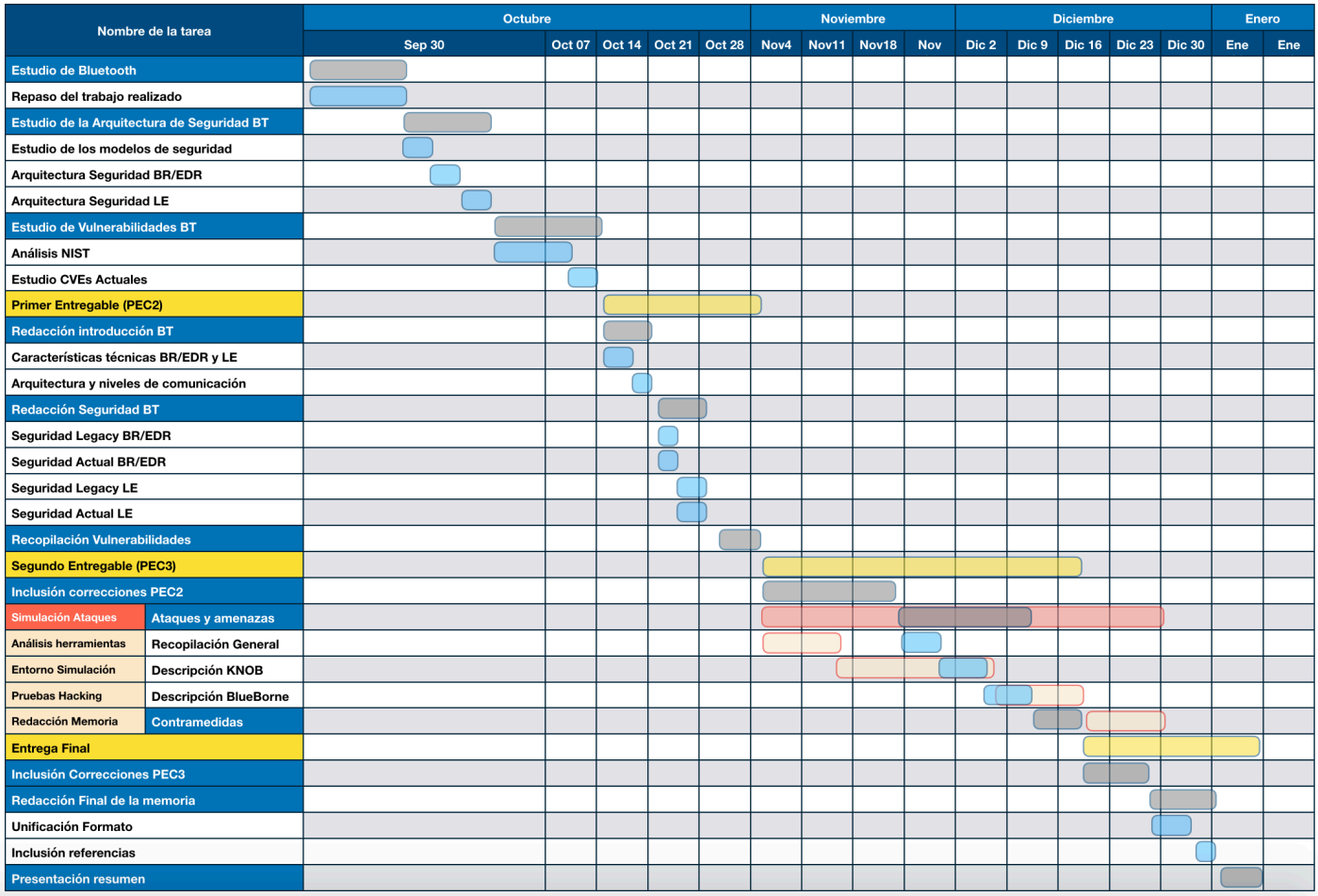

Tabla 1: Planificación del Trabajo

1.5. Estructura de la memoria

En esta memoria se documenta el trabajo realizado durante este semestre, recopilando y sintetizando información de distintas fuentes, así como generando gráficas, tablas e imágenes que ilustren dicha información. Para facilitar el acceso a la misma, se describe brevemente la estructura de la memoria, que consta de 5 grandes bloques:

- En el capítulo 2 se hace una introducción general a Bluetooth, aportando cifras que nos ayuden a entender la relevancia de la misma en la actualidad, y recopilando la arquitectura tanto de BR/EDR como de LE,

analizando las capas de comunicación de ambos y sus diferentes versiones.

- En el capítulo 3 se detalla la estructura de seguridad de Bluetooth, tanto de lo que llamamos seguridad *legacy*, o heredada, de las versiones previas a la 2.1, como de la seguridad evolucionada, que incluye mecanismos más sofisticados en aspectos de emparejamiento, autenticación y cifrado, como son SSP o *Secure Connections*.
- En el capítulo 4 se analizan las vulnerabilidades de Bluetooth, tanto de las versiones actuales como de versiones anteriores, ya que finalmente un dispositivo puede adaptar su modo de comunicarse cuando lo hace con otro que trabaje en una versión anterior. Además, en este apartado se analizan las distintas amenazas que tienen las redes Bluetooth, y se hace una recopilación de los ataques que las explotan. Por último, se recogen las contramedidas o buenas prácticas que los usuarios de Bluetooth deben llevar a cabo para, en la medida de lo posible, protegerse de los atacantes, o mitigar el impacto de sus ataques.
- En el capítulo 5 se tratan algunos de los casos más relevantes de los últimos tiempos con respecto a ataques, como ha sido el caso de BlueBorne y de KNOB. En ambos casos se describen los ataques, y se recogen recomendaciones concretas para evitarlos en la medida de lo posible.
- En el capítulo 6 se recogen dos casos prácticos con hipotéticas empresas reales, detallando sus riesgos y sus recomendaciones de seguridad según sus características de negocio y su uso de tecnología Bluetooth.
- En el capítulo 7 se recogen las conclusiones del estudio, en cuanto a la seguridad de Bluetooth, y en cuanto a la consecución de los objetivos establecidos en este trabajo.
- Por último, se incluyen un glosario con los términos más repetidos en este documento y una bibliografía con las referencias que se han tomado para su elaboración.

2. Tecnología Bluetooth

2.1. Introducción

La tecnología Bluetooth es un estándar de comunicación radio de corto alcance que permite la transmisión de voz y datos entre dispositivos. Nacida gracias al trabajo de los ingenieros de Ericsson Jaap Haartsen y Mattisson Sven, es presentada por primera vez en 1998 por el llamado *Bluetooth Special Interest Group* (SIG), compuesto por IBM, Intel, Toshiba, Nokia y la propia Ericsson. Sus valores principales, robustez, bajo coste y bajo consumo, la han convertido en la especificación más extendida para la comunicación sin cables de corto alcance entre dispositivos.

Localizado actualmente en su versión 5.1, Bluetooth abarca un gran abanico de aplicaciones, tales como la interconexión de dispositivos periféricos (ratones, teclados, etc.), la transmisión inalámbrica de voz y audio (auriculares, altavoces, manos-libres en vehículos) o el envío de datos en dispositivos móviles o *wearables*. Estas aplicaciones y más son posibles gracias al sistema básico de Bluetooth conocido como Bluetooth BR/EDR, que ha ido viendo incrementadas sus prestaciones a lo largo de los años.

Además, con el avance y auge del *Internet of Things* y las propuestas de sensorización en entornos *smart*, desde su versión 4 Bluetooth incluyó dentro de su especificación un nuevo sistema llamado Bluetooth LE, una versión de la tecnología desarrollada específicamente para reducir su consumo de energía, su complejidad y su coste y adaptarse mejor a estos escenarios de operación. Sumado a su estándar de conectividad en malla, *Bluetooth Mesh*, esta tecnología mantiene su posicionamiento para ser una constante en el mundo IoT.

Ilustración 1. Usos de Bluetooth

2.2. Bluetooth en cifras

Desde su llegada, Bluetooth ha tenido una presencia cada vez más extendida en la industria de las comunicaciones electrónicas, y esa tendencia no parece que vaya sino a aumentar. De acuerdo al último informe de mercado lanzado por el SIG en 2019 [1], las perspectivas de futuro de Bluetooth no pueden ser más positivas.

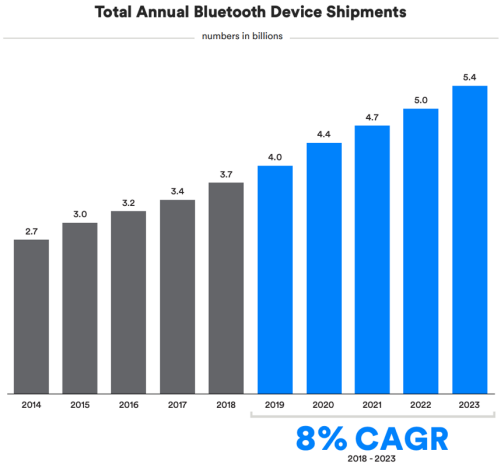

Ilustración 2: Producción de dispositivos Bluetooth [1]

En términos de comunidad, el SIG continúa aumentando su lista de miembros. Desde sus seis fundadores originales, el grupo cuenta con más de 34.000 compañías adscritas en 2019, habiendo experimentado un crecimiento del 70% en los últimos cinco años. Dado este dato, no es extraño pues que la tecnología continúe expandiéndose en distintos mercados y que el número de dispositivos lanzados al mercado con soporte Bluetooth se sitúe en 4 miles de millones de unidades, y con una tendencia de crecimiento para la que no hay indicios de desaceleración. Se espera mantener la tasa de crecimiento anual en un 8% y alcanzar para 2023 un total de 5,4 miles de millones de dispositivos. Cabe destacar la gran aceptación de Bluetooth LE en la industria, y cómo desde su llegada la penetración de esta tecnología ha ido creciendo. No solo se espera que para 2023 los dispositivos LE supongan un tercio de la producción total de sistemas Bluetooth, sino que la tendencia dominante pasa por descartar el desarrollo de soluciones que soporten exclusivamente BR/EDR y producir dispositivos con soporte dual (BR/EDR + LE). Esto arroja una cifra muy clara con respecto al futuro: para 2023 se estima que el 90% de la producción tendrá incorporada la tecnología LE.

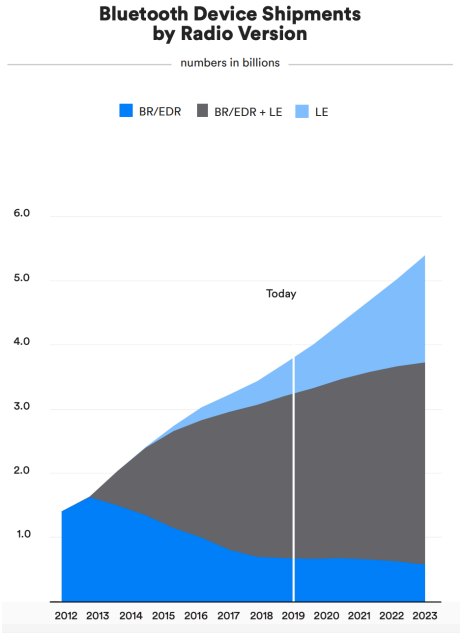

Ilustración 3: Dispositivos Bluetooth por tipo de tecnología [1]

Analizando el mercado con mayor capilaridad, si segregamos por área tecnológica Bluetooth se puede clasificar dentro de cuatro ámbitos: *streaming* de audio, transferencia de datos, redes de dispositivos y servicios de localización.

Audio

La topología punto a punto de Bluetooth BR/EDR está optimizada para la transmisión de audio, lo que lo ha convertido en el estándar principal del mercado de auriculares y altavoces *Wireless*, así como de soluciones manos-libres para

automóviles, con 910 millones de unidades producidas en 2018, y con una tendencia positiva de alrededor de +100 millones al año para el futuro. Una clara muestra de la confianza de las compañías en la tecnología es la reciente tendencia a retirar el conector *jack* de los dispositivos móviles. Asimismo, se estima que 8 de cada 10 altavoces soporten Bluetooth para 2023.

Transferencia de datos

Ya sea con BR/EDR para una gran cantidad de datos como LE para conexión de bajo consumo, Bluetooth es una tecnología perfecta para la conexión de periféricos. Tal es así, que ya en 2018 el 100% de los dispositivos fabricados tenía soporte para Bluetooth en el ámbito de los PCs, *smartphones* y *tablets,* así como en controladores del ámbito de las videoconsolas. Asimismo, en el mercado de los *wearables* se espera alcanzar los 278 millones de unidades para 2023, casi cuadruplicando la producción en el caso concreto de los *smarwatches* y con un crecimiento anual del 40% en el caso de los *wearables* de ámbito médico. En total se espera alcanzar una cifra de 1.350 millones de unidades para el año 2023, lo que supone un crecimiento anual del 14%.

Redes de dispositivos

La aparición de LE permitió a Bluetooth entrar con firmeza en el mercado de las comunicaciones M2M y el sector IoT. Ya sea para sistemas de control, automatización o monitorización, Bluetooth tiene unas grandes expectativas para su expansión dentro de los entornos *smart* en todos sus niveles (*smart home*, *smart building*, *smart industry*, *smart city*). Se espera que para 2023 se hayan alcanzado los 360 millones de dispositivos, creciendo el sector en torno a un 24% anualmente.

Servicios de localización

La localización mediante Bluetooth es una de las mejoras que la tecnología ha adquirido más recientemente. Gracias al uso del AoA (*Angle of Arrival*), los dispositivos LE son capaces de calcular la ubicación del transmisor cuya señal están recibiendo. Esto ha abierto todo un abanico de posibilidades para aprovechar esta capacidad, como el desarrollo de sistemas de navegación *in-door*, gestión y localización eficiente de suministros o instalación de balizas PoI (*Point-of-Interest*) como puntos de información en s*mart cities*. Se espera que este mercado emergente crezca de manera importante en los próximos años, alcanzando los 431 millones de unidades lanzadas para 2023.

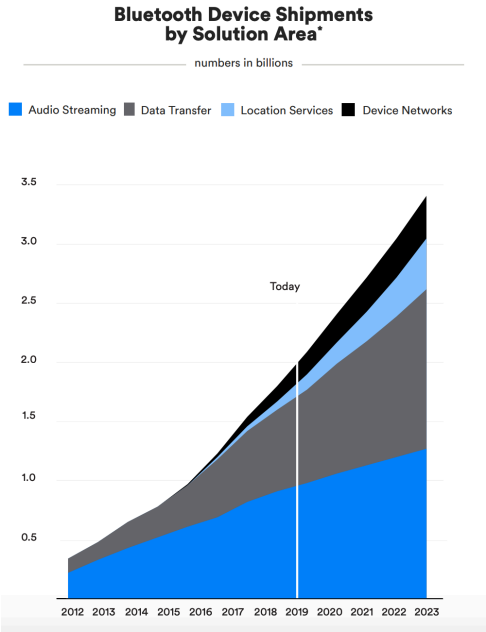

Ilustración 4: Producción de dispositivos Bluetooth por tipo de servicio [1]

2.3. Características técnicas

La tecnología Bluetooth es un estándar de comunicaciones inalámbricas de corto alcance que opera en la banda no licenciada de 2400 a 2483,5 MHz, conocida como banda ISM (*Industrial, Scientific, Medical*), un espacio de frecuencias reservado internacionalmente para usos con dichos propósitos.

Como se ha introducido anteriormente, la tecnología Bluetooth se divide fundamentalmente en dos tipos de sistemas: *Basic Rate* (BR), suplementado por sus extensiones *Enhanced Data Rate* (EDR) y *Alternate Media Access Control and Physical* (AMP), y *Low Energy* (LE). Estos sistemas definen principalmente las características a nivel físico y de enlace de los dispositivos. A continuación se describen en detalle las características técnicas recogidas en la especificación de Bluetooth [2.a].

2.3.1. BR/EDR + AMP

Basic Rate es el sistema original que aparece con la primera versión de Bluetooth. Dado que opera en la banda ISM, las comunicaciones se encuentran generalmente expuestas a constantes interferencias producidas por una gran variedad de sistemas que trabajan en la misma banda, como por ejemplo el WiFi. Para combatir este efecto, Bluetooth BR utiliza FHSS (*Frequency Hopping Spread Spectrum*), esto es, la capacidad de modificar la frecuencia de trabajo dinámicamente de manera pseudoaleatoria. Cuando se establece una conexión entre dispositivos, formando lo que se conoce como *piconet*, uno de ellos asume el rol de maestro y el resto, el de esclavos. El maestro se convierte en la referencia para la sincronización de reloj entre todos los dispositivos, y ejecuta un algoritmo de cambio de frecuencia que permite establecer uno de entre 79 canales de frecuencia para la comunicación y notificárselo a los esclavos. Este algoritmo, que permite incluso descartar ciertos rangos de canales dentro de los 79 en caso de detectar saturación en ellos, hacen posible que Bluetooth coexista de manera eficiente con otras señales. Asimismo, la cadencia de salto entre frecuencias se mide en *slots*, la unidad de tiempo en la que se divide un canal físico. Sobre ellos se transmiten los datos agrupados en paquetes (ocupando uno o más *slots*), recurriendo a un esquema TDD (*Time-Division Duplex*) para permitir la alternancia en el sentido de la comunicación y dar el efecto de una conexión *full-duplex*.

BR transmite utilizando una modulación GFSK, una modulación de frecuencia binaria que permite reducir la complejidad del transceptor y alcanza una velocidad de transmisión de 1 Mb/s. Por su parte, la extensión EDR utiliza para la transmisión de datos una modulación PSK, con la que consigue llegar a velocidades de 2 Mb/s y 3 Mb/s. Adicionalmente, a partir de la versión 3.0 se añadió AMP, una versión opcional de *Controller* compatible con la norma 802.11 con la que incrementar la velocidad de transmisión para aplicaciones más exigentes. A partir de una conexión establecida mediante el *Controller* BR/EDR, si las condiciones son propicias el dispositivo podrá migrar la transmisión de datos a un canal del *Controller* AMP, logrando alcanzar velocidades de hasta 54 Mb/s.

Desde el punto de vista de la potencia de transmisión, los dispositivos BR/EDR se clasifican en tres clases.

Clase de Potencia	Potencia Máxima Salida (P_{max})	Potencia Nominal Salida	Potencia Mínima Salida (P_{min})	Control de Potencia
1	100mW (20dBm)	-	1mW (0dBm)	Pmin<+4dBm a Pmax Opcional: Pmin2 a Pmax
2	2,5mW (4dBm)	1mW (0dBm)	0,25mW (-6dBm)	Opcional: Pmin2 a Pmax
3	1 mW (0 dBm)	-	-	Opcional: Pmin2 a Pmax

Tabla 2: Clases de dispositivos BR/EDR por potencia [2.a]

Transmisores de distinta clase pueden trabajar conjuntamente. Así como la mayor potencia y sensibilidad de un dispositivo clase 1 puede ayudar a ampliar el rango de comunicación con un dispositivo clase 2, un dispositivo clase 1 será capaz de ajustar su potencia cuando se comunique con dispositivos de clase menor a una distancia reducida, a fin de no saturarlos.

2.3.2. LE

Low Energy es la solución de Bluetooth para las comunicaciones de bajo consumo y transferencia de datos reducida, enfocadas al entorno M2M e IoT. Si bien LE opera en la banda ISM, este sistema presenta diferencias sustanciales con respecto a BR/EDR y AMP desde el punto de vista del nivel físico y del nivel de enlace.

LE utiliza modulación GFSK y adaptación FHSS para soportar la coexistencia con otras señales dentro de su banda. Sin embargo, utiliza dos esquemas de acceso distintos. Por un lado, se utiliza FDMA (*Frequency Division Multiple Access*) a lo largo de 40 canales separados por 2 MHz, de los cuales 3 se destinan para el tráfico de señalización y control de enlace primario (llamados canales de *advertising*) y los 37 restantes para los canales de datos y de señalización secundarios. Asimismo, un esquema TDMA (*Time Division Multiple Access*) es utilizado en cada canal para permitir la comunicación *dúplex* entre los dispositivos. Para ello, LE divide el tiempo en unidades denominadas eventos (a diferencia de BR/EDR y sus *slots*). Sobre estos eventos se transmiten los paquetes de datos entre dispositivos, pudiendo ser eventos de *advertising*, *extended advertising*, *periodic* o *connection*.

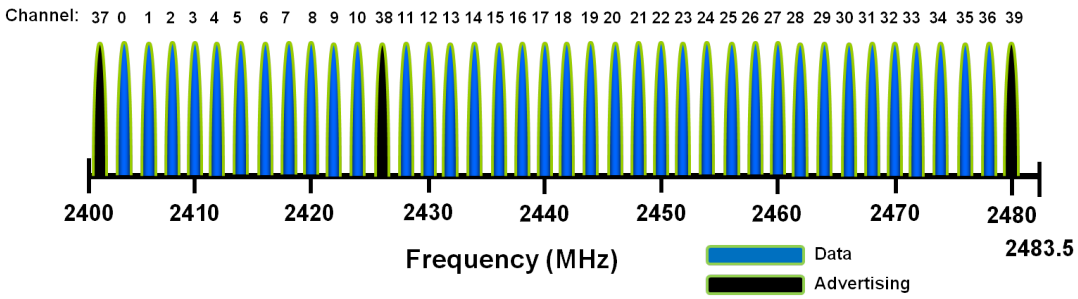

Ilustración 5: Esquema de canales de BLE [3]

Es precisamente el concepto de *advertising* el que diferencia a Bluetooth LE. Se denomina *advertisers* a los dispositivos que envían paquetes de *adveritising*, bien *broadcast* bien *unicast*, por sus canales físicos destinados a tal fin, mientras que se denomina *scanners* a aquellos que reciben estos paquetes sin interés en establecer conexión con los dispositivos que envían la información. Este modelo de comunicación no orientado a conexión, con paquetes de *advertising* o con paquetes de tipo *periodic* en canales de *advertising* secundarios, puede satisfacer las necesidades de una gran cantidad de escenarios. Para el caso de precisar establecer conexión, dispositivos denominados como *initiators* pueden lanzar una solicitud de conexión como respuesta a paquetes de *advertising* que la acepten.

Este sistema tiene soporte para contar con *arrays* de antenas en lugar de una única, lo que además de mejorar sus prestaciones, le otorga la posibilidad de calcular el AoA sobre el que se sustentan las aplicaciones de localización.

LE permite dos velocidades de transmisión, 1 Mb/s y 2Mb/s, conocidas como LE 1M PHY y LE 2M PHY. Además, en el primer caso existe la posibilidad de soportar codificación para la corrección de errores, dando lugar a las tasas de transmisión conocidas como LE *Coded* PHY. Estas tasas pueden ser de 500 Kb/s utilizando un total de S=2 símbolos para representar un bit, o de 125 Kb/s con S=8. LE 2M PHY no admite codificación.

De un modo similar al que se ha mostrado para el caso de BR/EDR, desde el punto de vista de los transmisores los dispositivos LE se clasifican en distintas clases según sus capacidades en términos de potencia.

Clase de Potencia	Potencia Máxima Salida (P_{max})	Potencia Mínima Salida ($P_{mín}$)
1	100 mW (20 dBm)	10 mW (+10dBm)
1.5	10 mW (10 dBm)	0.01 mW (-20 dBm)
2	2.5 mW (4 dBm)	0.01 mW (-20 dBm)
3	1 mW (0 dBm)	0.01 mW (-20 dBm)

Tabla 3: Clases de dispositivos LE por potencia [2.a]

2.4. Arquitectura

La torre de protocolos de la tecnología Bluetooth se divide en tres capas diferenciadas: la capa de *controller*, la capa de *host* y la capa de aplicación. La capa de aplicación es la encargada de comunicar los protocolos Bluetooth con la funcionalidad específica que se desee implementar y es definida por el desarrollador de dicha aplicación, mientras que el host y el *controller*, que constituirían el core del sistema Bluetooth, siguen dentro de cierta flexibilidad las indicaciones dadas por el estándar de la especificación, por lo que nos centraremos en esta última parte.

La capa de *controller* constituye el nivel físico de comunicación, siendo lo que se podría definir como la banda base del dispositivo, mientras que el host se

encarga de la interacción entre la capa física y la capa de aplicación. Estos elementos pueden estar coubicados en un mismo sistema (SoC) o encontrarse separados según el tipo de dispositivo. La comunicación entre ellos se realiza a través del interfaz denominado HCI (*Host Controller Interface*), que según las circunstancias de ubicación mencionadas podrá ser una comunicación interna, a través de protocolos propietarios o de protocolos estándar como USB o UART.

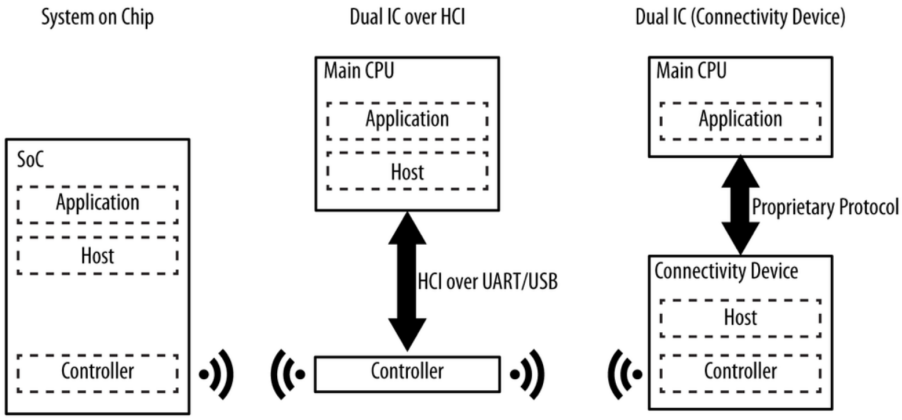

Ilustración 6: Esquema de configuraciones hardware de Bluetooth [4]

El *controller* de un dispositivo Bluetooth determina las capacidades de utilizar un sistema u otro de comunicación. Un dispositivo Bluetooth básico contará siempre con un *host* y, como mínimo, con un *controller* primario, que podrá ser del tipo BR/EDR, LE o dual, y podrá opcionalmente contar con N *controllers* secundarios AMP para los dispositivos que cuenten con soporte BR/EDR.

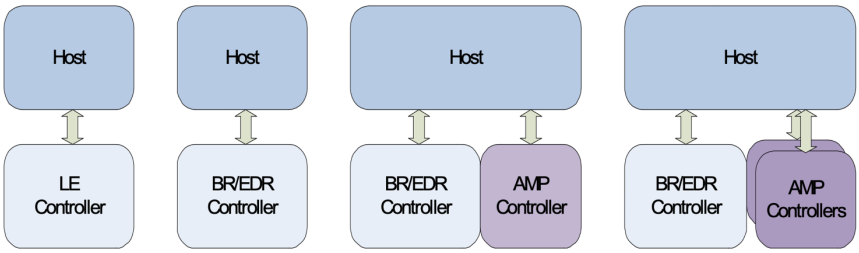

Ilustración 7: Combinaciones host y controller (I) [2.a]

De izquierda a derecha: Solo controller primario LE; Solo controller primario BR/EDR; Controller primario BR/EDR y un controller AMP secundario; y Controller primario BR/EDR y múltiples controllers AMP secundarios

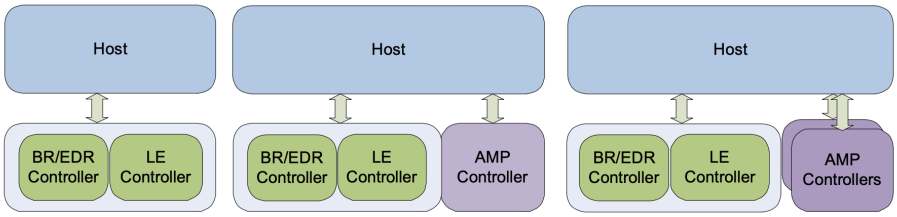

Ilustración 8: Combinaciones host y controller (II) [2.a]

De izquierda a derecha: Controller primario BR/EDR y LE; Controller primario BR/EDR y LE con un controller secundario AMP; y Controller primario BR/EDR y LE con múltiples controllers secundarios AMP

2.5. Niveles de comunicación

Bluetooth establece sus comunicaciones basándose en dos visiones distintas. Por un lado, el transporte de datos se realiza siguiendo una arquitectura vertical de capas desde la radio hasta la llamada capa L2CAP, encargada de la adaptación hacia el nivel de aplicación. Por otro lado, para que los parámetros de configuración de estas distintas capas sea la adecuada para cada aplicación en cuestión, Bluetooth implementa un sistema basado en perfiles que permite ajustar los requisitos entre las interacciones tanto entre dispositivos como entre los distintos niveles del transporte de datos. Partiendo de un perfil genérico denominado GAP (*Generic Access Profile*) establecido por estándar, los desarrolladores podrán adaptarlo en función a las características que precisen.

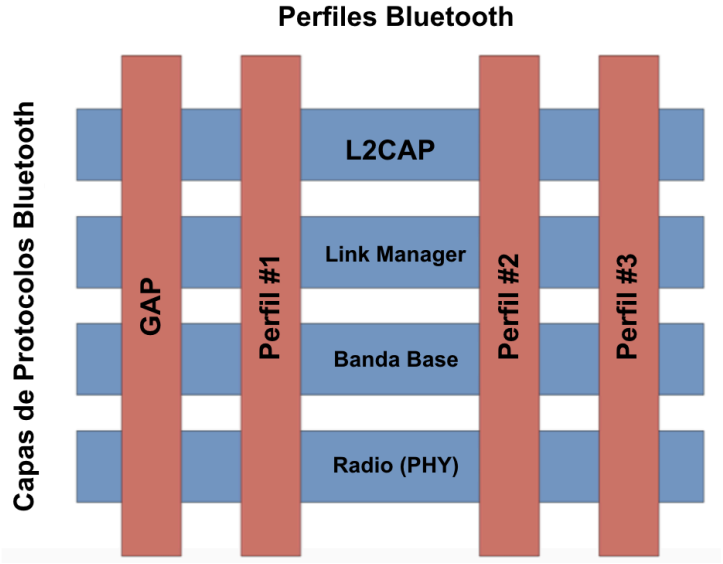

Ilustración 9: Perfiles de Bluetooth [2.a]

El sistema de transporte de datos de Bluetooth sigue en todos sus sistemas una estructura por capas: la capa física, la capa lógica y la capa L2CAP. Las partes que componen cada una de estas capas y los protocolos que soportan difieren en función del sistema Bluetooth empleado.

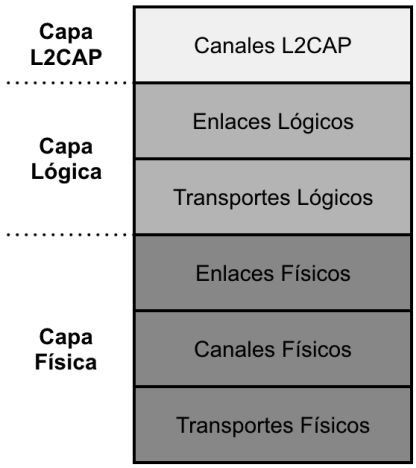

Ilustración 10: Estructura de Capas Bluetooth [2.a]

2.5.1. Capa física

El establecimiento de conexión entre un grupo de dos o más dispositivos, que como hemos definido anteriormente se denomina *piconet*, se basa en la compartición entre estos de un canal físico y su sincronización común con respecto a la frecuencia y al reloj que estipule el dispositivo maestro al resto de esclavos. Este canal físico representa el nivel más bajo de comunicación dentro de la arquitectura Bluetooth.

Dado el limitado número de canales de frecuencia y la gran diversidad de dispositivos presentes en el entorno, es habitual que dos comunicaciones independientes entre distintos dispositivos puedan encontrarse con que comparten frecuencia y sus transmisiones colisionen. Para solucionar esta situación en BR/EDR, los paquetes se encabezan con un código de acceso que permite identificar un canal físico y descartar otros ajenos a la comunicación que comparten canal de frecuencia. En el caso de LE, este código es un valor aleatorio correlado junto a la dirección del transmisor para solventar los casos de solapamiento de códigos. A la transmisión de paquetes a nivel físico (llamado comúnmente transmisión PHY) sin distinción de canales se le denomina transporte físico.

Dentro de la capa física existen distintos tipos de canales físicos según su utilidad y sus necesidades.

Tipo de Canal	Tecnología	Descripción
Basic piconet channel	BR/EDR	Canal estándar de comunicación entre dispositivos conectados.
Adapted piconet channel	BR/EDR	Canal básico con la posibilidad de exclusión de bandas y sin salto de frecuencia entre una transmisión maestro-esclavo y respuesta esclavo-maestro.
Inquiry scan channel	BR/EDR	Canal de recepción de mensajes de descubrimiento.
Page scan channel	BR/EDR	Canal de recepción de mensajes de inicio de conexión.
Synchronization scan channel	BR/EDR	Canal de recepción de mensajes de sincronización.
AMP physical channel	AMP	Canal físico de comunicación entre controladores AMP.
LE piconet physical channel	LE	Canal básico de comunicación entre dispositivos LE conectados.
Advertising physical channel	LE	Canal de descubrimiento y establecimiento de conexiones o comunicación broadcast.
Periodic physical channel	LE	Canal de comunicaciones broadcast periódicas entre dispositivos no conectados.

Tabla 4: Tipología de canales físicos

Un canal físico se divide a su vez en uno o más links físicos. Un link físico, si bien es un concepto virtual sin representación dentro de la estructura de paquetes, representa una conexión en banda base entre un dispositivo maestro y uno o más dispositivos esclavos. No se establecen links entre esclavos. En el caso de BR/EDR, esta comunicación puede ser bien bidireccional en el caso de los links físicos activos (sobre *Basic Piconet Channe*l y *Adapted PCh*), bien unidireccional multidestino en sentido maestro-esclavo en el caso de los links de tipo *conectionless slave broadcast*. Por otro lado, LE tiene un link físico distinto

para operar sobre cada uno de sus canales físicos: link activo punto a punto sobre LE Piconet PCh, link de *advertising* para *broadcast* y link *periodic* para *broadcast*.

2.5.2. Capa lógica

Los links físicos se utilizan para transportar a su vez flujos de datos independientes con distintas funciones, denominados links lógicos. Estos links soportan tráfico unicast tanto síncrono como asíncrono o isócrono, así como tráfico broadcast, y están asociados a su vez a un transporte lógico, que podrá constar de características concretas acorde a las necesidades de transporte del tipo de aplicación, tales como el control de flujos, los mecanismos de ACK y reenvío o las políticas de schedulling. Un dispositivo solo puede tener establecido un link lógico activo en un momento determinado. Para facilitar la capacidad de operar varios links, se utiliza TDMA para permitir permutar entre links activos. De manera general se definen tres tipos de links lógicos: links de control, links L2CAP y links de flujo.

El transporte lógico principal se denomina ACL (*Asynchronous Connection-oriented Logical transport*). Este se crea cuando un dispositivo se une a una piconet, y es el encargado de albergar el link lógico que transmite, además de datos de usuario, la señalización del protocolo de control que opera sobre las capas inferiores de la comunicación. En el caso de BR/EDR, este protocolo se conoce como Link Manager Protocol (LMP), mientras que en LE se denomina Link Layer Protocol (LL). En el caso de los controladores AMP, el equivalente a LMP se denomina PAL (Protocol Adaptation Layer), protocolo destinado al mapeo de los protocolos 802.11 con respecto a la especificación Bluetooth del HCI. La señalización PAL es soportada de igual modo por un link sobre ACL.

Además del ACL y el LE ACL, existen otros tipos de transportes lógicos:

- **SCO (Synchronous Connection-oriented Logical transport)** (BR/EDR) está definido para el transporte de tráfico de datos de usuario síncrono, bi-direccional y punto a punto entre maestro y esclavo.
- **eSCO (extended SCO)** (BR/EDR) es una mejora del transporte SCO capaz de soportar transporte asíncrono y mayor flexibilidad en la selección de periodos de slot y formatos de paquete. Asimismo, puede permitir retransmisión de paquetes.
- **CSB (Connectionless Slave Broadcast**) (BR/EDR) tiene como función transportar el tráfico de broadcast desde el maestro a todos sus potenciales esclavos. Es el único tipo de comunicación que se puede establecer sin necesidad de levantar previamente el ACL entre los dispositivos.
- **ASB (Active Slave Broadcast**) (BR/EDR) está diseñado para el transporte de mensajes broadcast a los esclavos activos de una piconet, ya sea tráfico de control LMP o tráfico de usuario.
- **ADVB (Advertising broadcast**) (LE) se encarga del transporte del tráfico broadcast, tanto de control como de usuario, hacia los scanners del área, sobre el link físico de advertising LE.

- **PADVB (Periodic Advertising Broadcast)** (LE) tiene la misma función que ADVB, pero para el tráfico periódico sobre el link físico periodic.

2.5.3. Capa L2CAP

Logical Link Control and Adaptation Protocol (L2CAP) es el nombre que recibe la capa que sirve como abstracción entre las aplicaciones Bluetooth y los niveles radio y banda base de la comunicación. El protocolo que gestiona la señalización de control de esta capa se denomina con el mismo nombre, L2CAP, y su tráfico está ubicado al igual que el de LMP sobre el transporte ACL. El tráfico de usuario específico de L2CAP puede ser transportado por cualquiera de los links lógicos con soporte para este protocolo.

En Bluetooth BR/EDR existe sobre esta capa un protocolo de uso específico llamado SDP (*Service Discovery Protocol*), cuya función se basa en permitir la comunicación entre dispositivos para solicitar información relevante sobre servicios con los que cuenta el otro terminal.

En el caso de Bluetooth LE además del protocolo L2CAP se utilizan dos protocolos suplementarios: *Security Manager Protocol* (SMP) y *Attribute Protocol* (ATT), que operan sobre un canal L2CAP. SMP es un protocolo específico que implementa las funciones de seguridad entre dispositivos, en lugar de ejecutarlas a nivel de link. ATT tiene la utilidad de permitir la comunicación de pequeñas cantidades de datos para que un dispositivo pueda determinar los servicios que soportan aquellos otros con los que se comunican. Es posible utilizar ATT en BR/EDR.

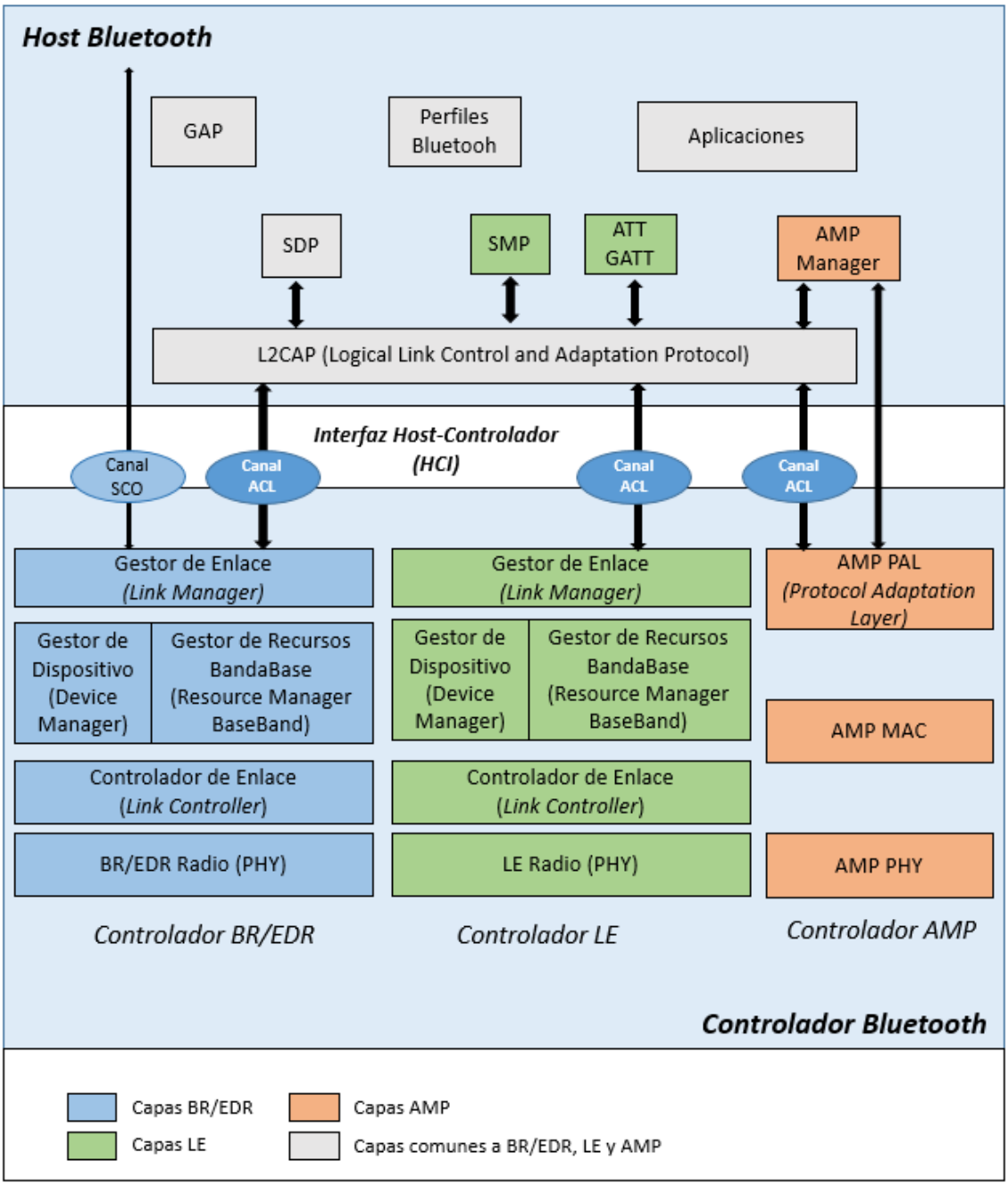

Ilustración 11: Arquitectura Bluetooth [5]

3. Seguridad Bluetooth

Como en tantas otras tecnologías de comunicación, la seguridad en Bluetooth ha pasado de ser una opción a ser un requisito cada vez más crítico debido a la gran expansión de terminales con Bluetooth en todos los aspectos de nuestra vida, desde simples auriculares a sistemas de monitorización *Smart* para ciudades o dispositivos médicos críticos. Es por ello que los mecanismos de seguridad disponibles a la hora de establecer comunicaciones tienen hoy una importancia vital y se encuentran en constante estudio de vulnerabilidades y mejoras para garantizar la mayor fortaleza posible. A continuación, haremos un repaso de la evolución de estos mecanismos a lo largo de la vida de la especificación y detallaremos la manera en la que se implementan, que vienen recogidos en la especificación Core v5.1, en la parte H de los volúmenes 2 y 3. [2.b] [2.c]

3.1. Evolución de la arquitectura de seguridad

Como se ha detallado en la sección 2.2, el asentamiento de la tecnología Bluetooth en las comunicaciones actuales es absoluto, y su campo de trabajo es cada vez más diverso y amplio en los distintos mercados de consumo. Este éxito ha generado una necesidad paralela: mejorar los mecanismos de seguridad de la tecnología a fin de garantizar el cumplimiento de unos requisitos más estrictos, acorde a la mayor criticidad de los servicios soportados sobre Bluetooth, tales como redes de sensorización (tanto públicos como privados), sistemas de biomedicina, etc.

El sistema Bluetooth ofrece un modelo de seguridad que incluye cinco servicios principales:

- ***Pairing***: proceso de generación e intercambio de claves.
- ***Bonding***: almacenamiento de claves con el fin de usarlas en posteriores conexiones.
- **Autenticación**: verificación de que dos dispositivos tienen la misma clave.
- **Encriptación**: establecimiento de confidencialidad entre mensajes.
- **Integridad de mensajes**: protección frente a falsificaciones de los datos.

Desde su creación hasta su versión actual, Bluetooth ha ido evolucionando para mejorar sus capacidades de cara a las garantías de seguridad de sus comunicaciones.

- **Hasta versión 2.0**: Originalmente, en lo que se conoce como seguridad Bluetooth Legacy, el proceso de pairing y el de autenticación utilizaban algoritmos basados en SAFER+ (E21 o E22 para el primero y E1 para el segundo), mientras que la encriptación se ejecutaba mediante un algoritmo E0. En lo que respecta a la integridad de los mensajes, no se implementaba.
- **Versión 2.1**: Con la introducción de EDR se añade Secure Simple Pairing, una serie de funciones de seguridad que fortalecen el proceso de

emparejamiento para defender a las comunicaciones Bluetooth frente a ataques MITM (Man In The Middle), sustituyendo el algoritmo de generación de claves legacy por otros más sofisticados como SHA-256 y ECDH P-192. La autenticación y la encriptación se mantiene igual.

- **Versión 3.0**: La especificación incluye soporte de seguridad para AMP.
- **Versión 4.0:** Se incluye el modelo de seguridad para el recién incorporado protocolo Low Energy.
- **Versión 4.1**: Se incluye a BR/EDR la funcionalidad Secure Connections, mejorando SSP con el algoritmo ECDH P-256 y la autenticación con algoritmos FIPS (HMAC-SHA-256 y AES-CTR). Además, Secure Connections incluye algoritmos para garantizar la integridad de los mensajes (AESCCM).
- **Versión 4.2**: Se mejora el modelo de seguridad de Bluetooth LE introduciéndo Secure Connections, que permite mejorar los algoritmos utilizados en el proceso de pairing. Además, adaptando uno de los modos de trabajo de SSP a Bluetooth LE. El modelo de seguridad BLE previo pasa a conocerse como seguridad BLE Legacy.

3.2. Seguridad en Bluetooth BR/EDR

Los procedimientos de seguridad de Bluetooth BR/EDR se aplican a nivel la capa lógica mediante LMP. Como hemos visto en el repaso de la evolución de la tecnología, según la manera de implementar los mecanismos de generación de claves, autenticación y cifrado la seguridad de Bluetooth BR/EDR se puede dividir en tres clases diferenciadas: Legacy, *Secure Simple Pairing* y *Secure Connections*. En los siguientes apartados se profundizará en cada una de ellas, resumiendo lo expuesto en la parte H del volumen 2 de la especificación Core v5.1[2.b] .

Con cada evolución, los algoritmos de cifrado, autenticación y generación de claves se robustecen. En la siguiente tabla se puede observar un resumen de los mismos.

Mecanismo de Seguridad	Legacy	Secure Simple Pairing (SSP)	Secure Connections
Encriptación	E0	E0	AES-CCM
Autenticación	SAFER+	SAFER+	HMAC-SHA256
Generación Claves	SAFER+	P-192 ECDH HMAC-SHA-256	P-256 ECDH HMAC-SHA-256

Tabla 5: Resumen de algoritmos por mecanismos Seguridad BR/EDR [2.b]

3.2.1. Seguridad Legacy

Los mecanismos de seguridad implementados hasta la versión 2.0 se denominan seguridad *legacy* o seguridad heredada, y son la base del modelo de seguridad empleado por Bluetooth. También denominado como *legacy pairing*, este modelo de seguridad se basa en el establecimiento a través de LMP de una clave secreta denominada clave de link (a la que nos referiremos en adelante como K_L) que los dispositivos comparten para poder establecer comunicaciones

seguras. El proceso de inicialización de una sesión de comunicación segura se divide en las siguientes fases:

- **Generación de clave de inicialización k_{init}**

Al establecerse una nueva comunicación entre dos dispositivos A y B se genera una clave temporal K_{init}, cuya función es ser utilizada para la generación de la clave de link K_L y el envío seguro de esta. Esta clave K_{init} se descartará una vez finalice el proceso de emparejamiento.

K_{init} se forma a partir de un valor aleatorio IN_RAND de 128 bits, un código PIN de longitud L octetos (entre 1 y 16) y la dirección de un dispositivo BR_ADDR de 48 bits. El valor del PIN, cuando no alcance los 16 octetos, se concatenará con BR_ADDR completa o parcialmente, en función de no sobrepasar los 16 octetos. Así, denominando PIN' y L' a esta concatenación y a su nueva longitud respectivcamente, los valores se combinan mediante un algoritmo E_{22}:

$$K_{init} = E_{22}(PIN', IN\_RAND, L')$$

El dispositivo que inicia la comunicación (*iniciator*) es el encargado de enviar el valor IN_RAND al otro dispositivo (*responder*). Este se encargará del proceso de generación siempre y cuando no cuente con un PIN prefijado por el sistema. En tal caso, generará un nuevo IN_RAND para el *iniciator* para que lleve a cabo el proceso. En caso de que ambos tengan un PIN prefijado, la comunicación no se podrá establecer.

- **Generación de clave de link K_L**

Una vez fijada la clave K_{init}, los dispositivos pasan a generar la que será la clave de link para la sesión de comunicación. Esta clave puede recibir distintos nombres en función de su proceso de creación.

  - **Clave unitaria**

La clave K_L es generada por uno de los dispositivos. Este genera un nuevo número aleatorio LK_RAND_A que combina con su dirección BR_ADDR_A para crear la clave con el algoritmo E_{21}.

$$K_A=E_{21}(LK\_RAND_A, BR\_ADDR_A)$$

El dispositivo A encripta esta clave mediante la operación $K_A \oplus K_{init}$ y se la envía al dispositivo B, que la desencripta y la guarda como clave de link. El uso de claves unitarias se considera obsoleto, ya que en las primeras versiones Bluetooth las claves unitarias se generan normalmente una vez durante la instalación del dispositivo y no se cambian, lo que supone un grave problema de seguridad al usar la misma en todas sus comunicaciones.

- **Clave combinada**

Cada uno de los dispositivos genera una subclave LK_K_X (donde la X será A o B según el dispositivo) a partir del algoritmo E_{21} e introduciéndole un nuevo número aleatorio LK_RAND_X y su dirección BR_ADDR_X. A continuación, cada uno de los dispositivos envía al otro su número aleatorio LK_RAND_X cifrado mediante K_{init}. De este modo, dado que los dispositivos conocen la dirección del otro, pueden generar la subclave de su contrario. Una vez ambos dispositivos poseen las dos subclaves, las combinan a partir de un XOR para obtener la clave K_{AB}, que se establecerá como clave de link. Esta clave cuenta con la ventaja de involucrar información de ambos dispositivos, mejorando su seguridad. A continuación se describe la comunicación llevada a cabo para este proceso.

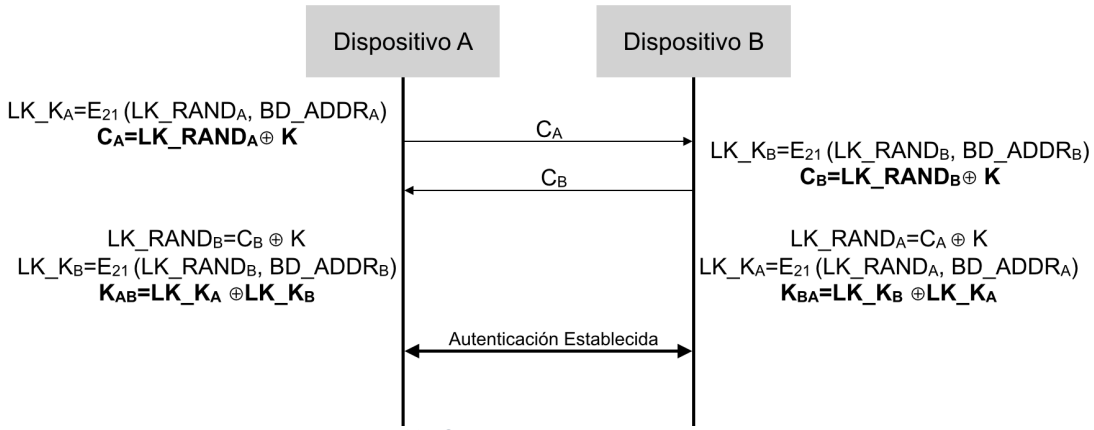

Ilustración 12: Generación clave combinada K_{AB} [2.b]

- **Clave maestra**

En el caso de que el nodo maestro de una piconet necesite enviar información broadcast a todos sus esclavos, realizar estos envíos con una clave distinta para cada uno de ellos derivaría en un uso poco eficiente del ancho de banda y los recursos del maestro. Es por esto que en tal caso el nodo maestro inicia el proceso para generar una nueva clave K_{master} que comparte con todos los nodos esclavos conectados. Esta clave se crea mediante dos números aleatorios generados por el nodo maestro. A continuación, genera un tercer número aleatorio RAND que envía a los esclavos de la piconet. Todos ellos calculan un valor de 128 bits temporal denominado overlay a partir de dicho RAND y la clave previa existente en la comunicación. El nodo maestro utiliza este *overlay* para cifrar K_{master}, que los nodos esclavos pueden recuperar al poseer el mismo valor de *overlay*. A continuación se ilustra la comunicación del proceso.

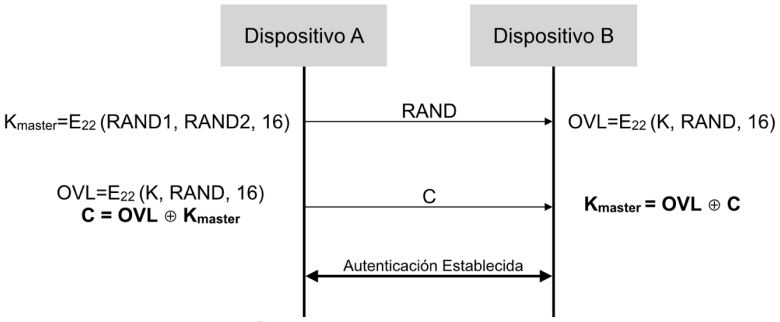

Ilustración 13: Cálculo y distribución de clave maestra [2.b]

El uso de claves maestras se considera menos seguro al tener que utilizar una misma clave para las comunicaciones con distintos dispositivos, lo que implica que cualquiera de ellos podría escuchar las comunicaciones con el resto. Es por esto que se reserva para este tipo de casuísticas de *broadcast*.

- **Autenticación**

Una vez se ha generado y almacenado la clave K_L para la sesión, es posible realizar la autenticación entre dispositivos para garantizar que la comunicación se realiza entre dos dispositivos conocidos. Para ello, el dispositivo A que inicia el proceso, denominado *verifier*, envía un valor aleatorio AU_RAND_A al dispositivo B, denominado *claimant*. Este mensaje se conoce como reto de autenticación (*challenge*), y con la combinación de este valor, su dirección BR_ADDR_B y la clave K_L, el dispositivo B obtiene mediante el algoritmo E_1 una salida SRES (Signed Response) de 32 bits que envía al dispositivo A. Este, que ha realizado el mismo proceso paralelamente, compara su salida SRES con la recibida y en caso de coincidencia, da por válida la autenticación. Cabe destacar que el algoritmo E_1 genera una segunda salida de 96 bits denominada ACO (*Authentication Ciphering Offset*) que en caso de autenticación válida ambos dispositivos almacenan para utilizar en el proceso de encriptación.

Para evitar conseguir autenticar un dispositivo fraudulentamente a partir de la fuerza bruta, la especificación introduce un retardo entre intentos consecutivos que va aumentando exponencialmente según se vayan repitiendo los fallos de un dispositivo.

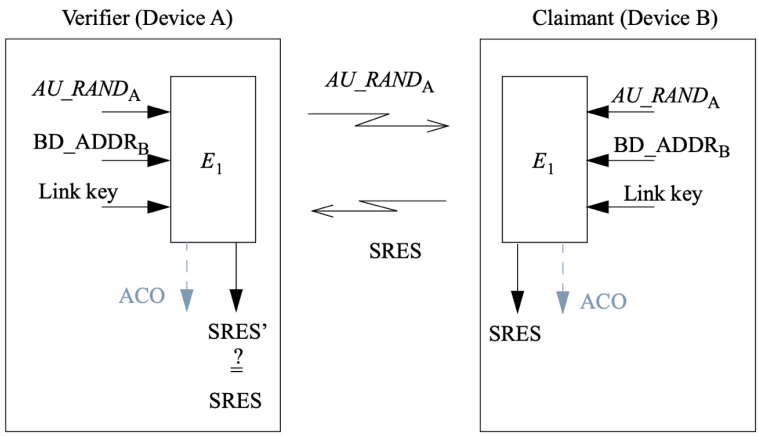

Ilustración 14: Esquema challenge-response de Bluetooth. [2.b]

- **Encriptación**

En el caso de querer llevar a cabo encriptación de las comunicaciones, el primer paso es que los dispositivos implicados negocien la longitud de la clave de cifrado KC que van a utilizar, que podrá estar entre los 8 y los 128 bits de longitud. Es posible que a nivel de aplicación se imponga una longitud mínima para garantizar la seguridad del cifrado. La negociación comienza con el dispositivo A ofreciendo al dispositivo B el uso de una clave de longitud máxima. Este deberá responder afirmativamente si acepta o, en caso de que no lo soporte, responder proponiendo la máxima longitud que su sistema sea capaz de manejar. El dispositivo A entonces hará la misma comprobación y confirmará la longitud u ofrecerá una nueva, hasta que se alcance un acuerdo o se sobrepase el límite inferior que marque la aplicación y se cancele la solicitud de encriptación.

Una vez decidida la longitud L, el dispositivo iniciador envía un número aleatorio EN_RAND y ambos generan la clave KC mediante el algoritmo E3 a partir de la clave KL, EN_RAND y un registro COF (Cipher Offset). Este valor será la salida ACO guardada durante la autenticación en el caso de trabajar con claves combinadas o unitarias, o la concatenación de la dirección BD_ADDR del dispositivo maestro en caso de trabajar con claves maestras para comunicación broadcast.

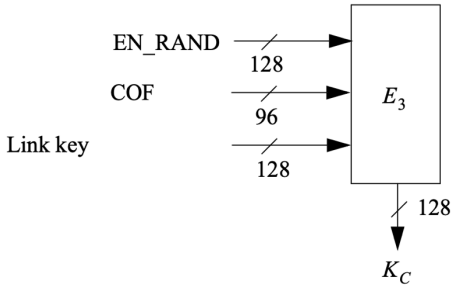

Ilustración 15: Generación clave de cifrado mediante algoritmo E_3 [2.b]

Esta clave se adapta al tamaño negociado y se utiliza para generar, mediante el algoritmo E_0, un stream de encriptación sincronizado con el reloj del sistema que se suma en módulo 2 tanto con los datos salientes (encriptación) como con los datos entrantes (desencriptación).

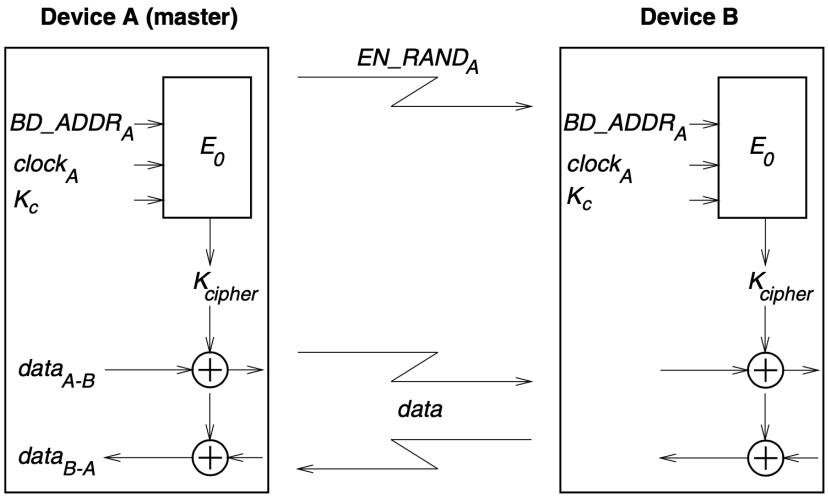

Ilustración 16: Descripción funcional del proceso de cifrado con E_0 [2.b]

3.2.2. Secure Simple Pairing

A partir de la versión 2.1 se incluye la funcionalidad *Secure Simple Pairing* (SSP) para mejorar la protección frente a escucha pasiva o activa (MITM). Para ello, se mejora el procedimiento de *pairing* de cara a obtener mayor seguridad en la generación de claves dependiendo de valores con más entropía que el PIN *legacy* cuando sea posible en base a las capacidades de los dispositivos.

Secure Simple Pairing se compone de cinco fases:

- Intercambio de claves públicas
- Autenticación fase 1
- Autenticación fase 2
- Cálculo de clave de link
- Autenticación y encriptación LMP

A continuación pasamos a describirlas.

3.2.2.1. Fase 1: Intercambio de claves públicas

El primer paso que llevan a cabo los dispositivos es generarse una pareja de claves pública-privada basada en Curvas Elípticas Diffie-Hellman (ECDH) P-192. El dispositivo que inicia la comunicación comienza enviando su clave pública al receptor, el cual procede de igual manera al recibirla.

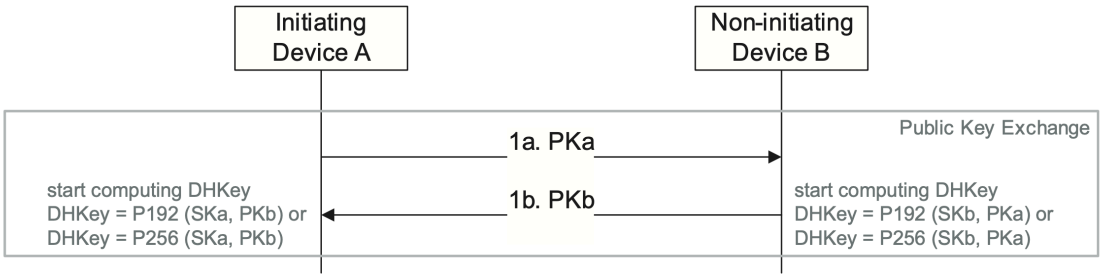

Ilustración 17: Detalle del intercambio de claves públicas de SSP [2.b]

Asimismo, mediante estos datos cada dispositivo pasa a calcular el secreto compartido DHKEY, que sale como resultado de operar sobre la clave privada propia y la clave pública del otro dispositivo. Gracias a sus características, estos cálculos cruzados deben dar el mismo resultado. Dada la complejidad de la operación, es común que el cálculo se comience en este punto y se deje operando mientras se ejecuta la fase siguiente.

3.2.2.2. Fase 2: Autenticación fase 1

La segunda etapa de SSP puede llevarse a cabo mediante cuatro protocolos distintos o modelos de asociación, que se eligen en función de las capacidades I/O de los dispositivos implicados, capacidades de las que los dispositivos informan al establecer un intento de comunicación. Estos modelos son:

- **Numeric Comparison**

Este modelo de asociación se utiliza cuando ambos dispositivos cuentan con una pantalla capaz de mostrar a los usuarios un PIN de 6 dígitos y con una entrada que permita al usuario introducir un “ok” o un “no ok”. El modelo más típico es el de un PC y un teléfono móvil.

Este modelo comienza con la generación de un número aleatorio N_X de 128 bits por parte de cada dispositivo. A continuación, el *responder* utiliza su valor aleatorio y las claves públicas de ambos dispositivos para generar mediante una función criptográfica f1 un nuevo valor aleatorio llamado *commitment*, C_B. Una vez generado, el *responder* envía C_B al dispositivo A, y es entonces cuando ambos dispositivos comparten su valor aleatorio N_X. El dispositivo A calcula entonces el valor C_B utilizando los mismos campos que usó el dispositivo B y verifica que su resultado coincide con el que recibió previamente. Este envío previo de C_B y posterior recálculo por parte del dispositivo B se basa en aumentar la protección para ataques MITM, ya que para infiltrarse en la comunicación, un atacante deberá enviar al dispositivo A un valor C_B antes de averiguar el número aleatorio NB que se envía posteriormente. Esto hace que a efectos prácticos la ejecución de un ataque MITM sea virtualmente imposible, al tener una probabilidad de éxito de 0,000001.

A continuación, ambos dispositivos generan un valor VX mediante sus claves públicas y sus números aleatorios NX, que será el PIN de 6 dígitos que deberán mostrar al usuario y que deberá ser igual en ambos casos. La fase finaliza una

vez los dispositivos reciben la confirmación del usuario en los dispositivos tras comprobar que se muestra el mismo PIN.

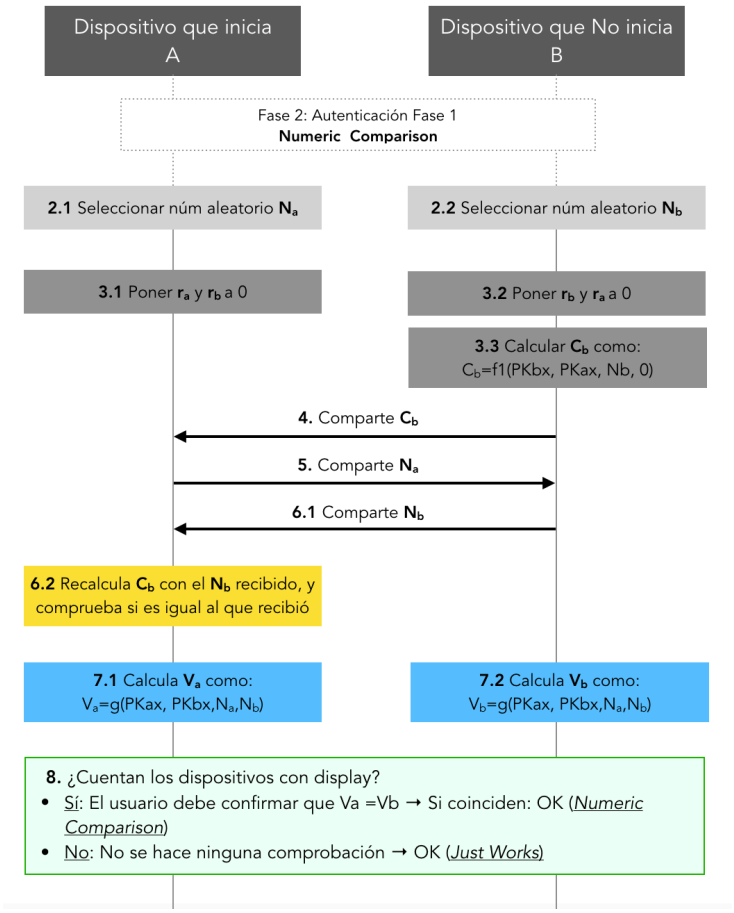

Ilustración 18: Detalle Numeric Comparison y Just Works [2.b]

- **Just Works**

Este modelo se considera una variación de *Numeric Comparison* para comunicaciones en las que un dispositivo no cuenta con pantalla ni con teclado de entrada, como puede observarse en la ilustración 18. Esto tiene como consecuencia que el paso de confirmación por parte del usuario que hemos visto no se puede realizar en dicho dispositivo, y que en ningún momento el usuario conoce el PIN de 6 dígitos, lo que conlleva que este modelo no proteja de ataques MITM. Este modelo es el típico en escenarios de auriculares o dispositivos de sonido conectados a reproductores.

- **Passkey Entry**

Este modelo se utiliza cuando el escenario cuenta con un dispositivo con teclado para introducir datos (pero no pantalla) y otro dispositivo con, al menos, pantalla. Un caso habitual es el de la conexión de teclados Bluetooth con PCs.

En esta fase los dispositivos comienzan o bien solicitando que se introduzca el mismo PIN de 6 dígitos en ambos, o bien generándolo en aquel con pantalla para que el usuario lo introduzca en el otro. Esto da lugar a que ambos dispositivos posea un valor de 20 bits denominado r (r_a=r_b). A continuación, se inicia un

proceso similar al visto en *Numeric Comparison*, generando cada dispositivo un número aleatorio N_x con el que cada uno, junto a sus claves públicas y el primer bit del PIN r, generan un *commitment* C_x. Igual que antes, C_x se envía al dispositivo contrario para, a continuación, enviarle el número N_x y que el dispositivo receptor calcule por su cuenta el *commitment* del contrario y verifique que coinciden (en este caso, al contrario que en *Numeric Comparison*, este proceso se ejecuta en ambos lados).

Todo este proceso descrito desde la introducción del PIN r se debe ejecutar un total de 20 veces, haciendo que el bit de r usado en la función que genera el *commitment* vaya avanzando de modo que se use toda la palabra r.

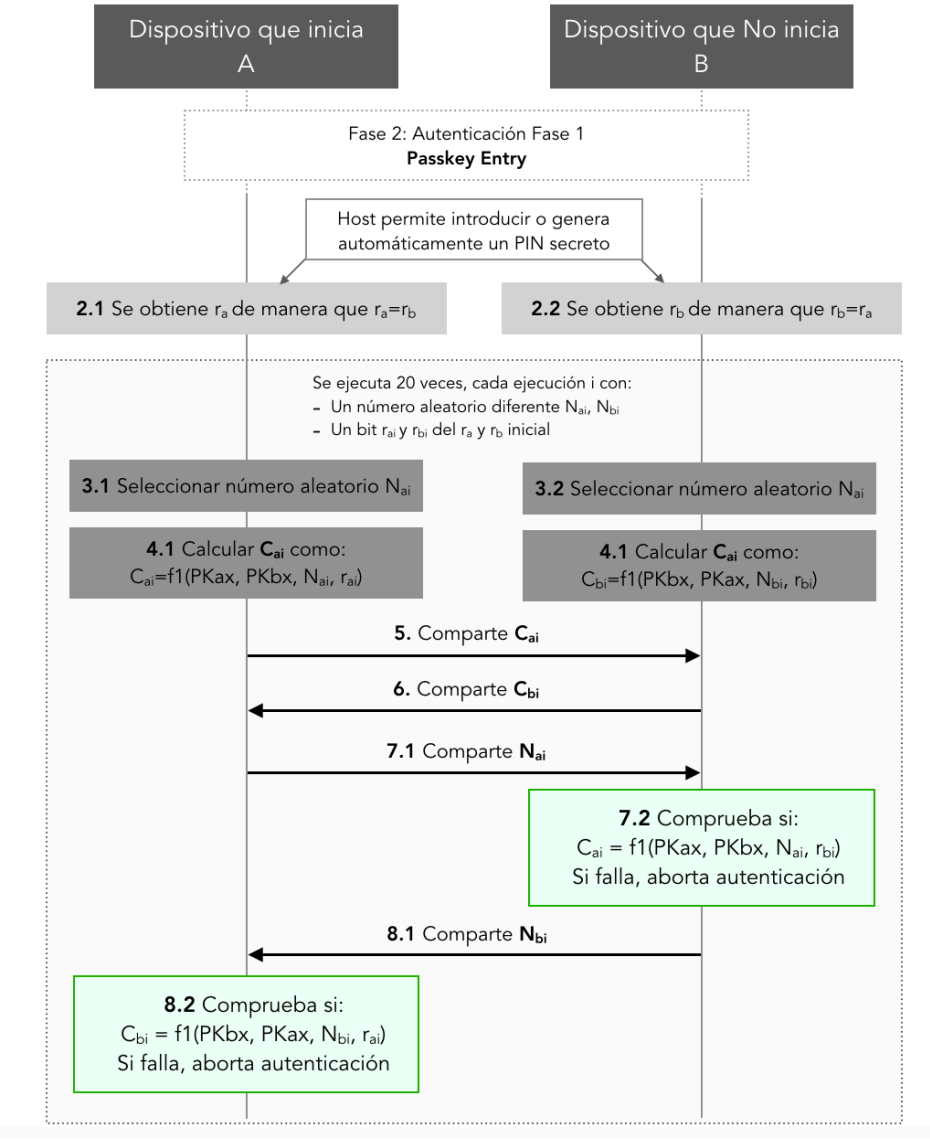

Ilustración 19: Fase 1 de Autenticación: Detalle Passkey Entry [2.b]

- **Out of Band (OOB)**

Este último modelo se basa en recurrir a tecnologías alternativas a Bluetooth para realizar el proceso de pairing, tales como NFC. Se utiliza cuando al menos uno de los dispositivos ha indicado que tiene capacidad para utilizarla, a través

del parámetro *OOB Authentication Data Present* en la secuencia I/O de LMP intercambiada.

En caso de que ambos dispositivos puedan transmitir/recibir datos a través de un canal OOB, la autenticación mutua se hará mediante la comparación de *commitments* calculados y enviados a través del canal OOB, C_a y C_b. En caso de ser posible solo en una dirección, la autenticación del dispositivo que recibe la comunicación OOB se basará en un número aleatorio *r* recibido a través de ese canal OOB. Dicho número *r* debe ser secreto y generado de nuevo cada vez que se intente autenticar un nuevo dispositivo. Si *r* no es enviado por un dispositivo, se supone que es 0 por el dispositivo que recibe la información OOB en los pasos 4.1 y 4.2 de la ilustración 20.

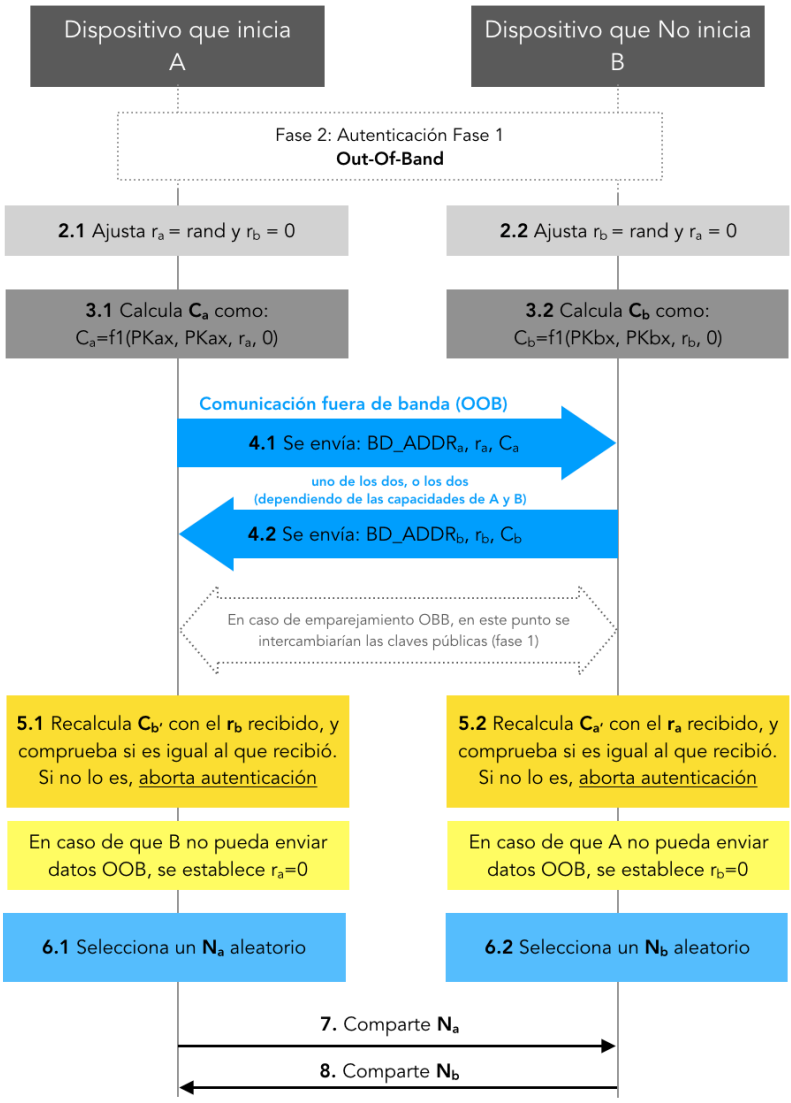

Ilustración 20: Fase 1 de Autenticación: Detalle Out-Of.Band [2.b]

Como se ve en el diagrama, los pasos 2, 3 y 4 se realizan después del intercambio de claves públicas realizado en comunicación dentro de banda (fase 1), pero en el caso de que el emparejamiento se inicie directamente con una

comunicación OOB, este intercambio de claves se realizará entre el punto 4 y el punto 5.

El valor *r* que se comprueba en caso de que uno de los dos dispositivos no tenga capacidades OOB se establece inicialmente por defecto a cero para el otro dispositivo, para que en caso de que la comunicación OOB no sea recibida se iguale a cero el del propio dispositivo, y siendo ambos iguales para A y B, de manera que $r_a = r_b$ la autenticación pueda progresar.

3.2.2.3. Fase 3: Autenticación fase 2

En este punto se espera que los dispositivos hayan terminado de calcular el secreto DHKEY que se inició en la fase 1. Los dispositivos proceden entonces a verificar que comparten el mismo valor a partir de operarlo junto a distintos campos compartidos en las fases previas.

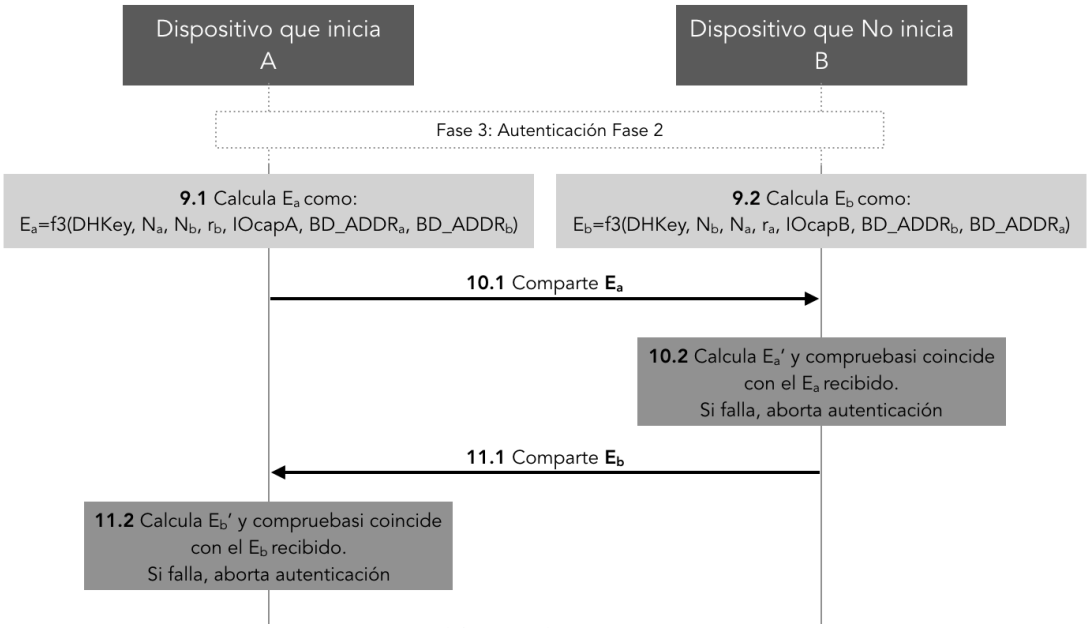

Ilustración 21: Fase 2 de Autenticación [2.b]

Estos campos son (donde X puede ser A o B según el dispositivo):

- $IOcap_X$: 3 octetos con información de capacidades I/O del dispositivo, compartido al inicio de la conexión.
- N_X: número aleatorio generado durante la fase 2.
- r_x: número aleatorio generado durante la fase 2 para *Passkey Entry* y OOB. Para *Numeric Comparison* y *Just Works* se establece a cero.
- BD_ADDR_x: dirección Bluetooth del dispositivo.

Cada dispositivo genera un registro E_X como salida, que comparte con el contrario. Ambos deben recalcularlo con los datos del otro dispositivo y verificar que el DHKEY compartido es igual. En caso negativo se aborta la conexión.

3.2.2.4. Fase 4: Cálculo de clave de link

A continuación, se pasa a generar la clave de link para la sesión. Para ello se ejecuta una función f2 pasando como parámetros el secreto DHKEY, el *string* "btlk", las direcciones Bluetooth de los dispositivos y los números aleatorios compartidos en las primeras fases. Es importante destacar que, a diferencia de la clave *legacy*, la clave de link de SSP no utiliza el PIN para su generación, lo que permite mejorar su entropía sin necesidad de complicar el proceso de emparejamiento a ojos del usuario.

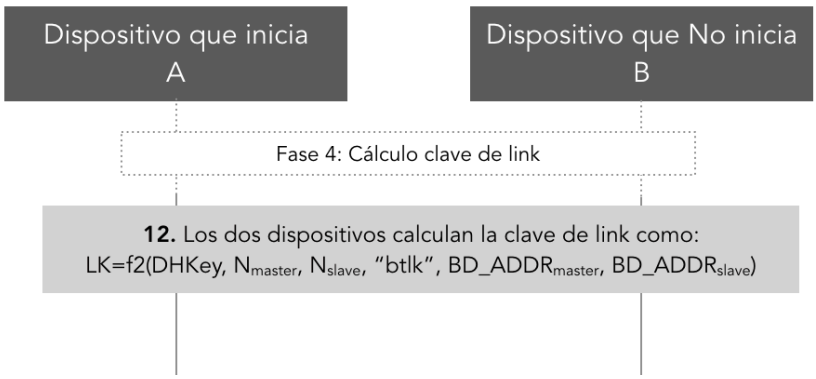

Ilustración 22: Cálculo clave de link [2.b]

3.2.2.5. Fase 5: Autenticación y encriptación LMP

La última fase de SSP se compone de los procesos de autenticación y encriptación siguiendo los procedimientos que ya hemos estudiado en Seguridad *Legacy*.

3.2.3. Secure Connections

A partir de la versión 4.1 BR/EDR pasa a implementar *Secure Connections* para mejorar la seguridad de sus procesos. *Secure Connections* es en esencia un *upgrade* de SSP sustituyendo sus funciones criptográficas por otras de mayor capacidad en todas sus etapas (generación de claves, autenticación y emparejamiento). Estas mejoras son:

- P-256 ECDH para la generación de claves.
- HMAC-SHA-256 para la autenticación.
- AES-CCM para la encriptación, que además añade integridad de mensajes.

3.2.4. Modos de seguridad

Según la versión de Bluetooth del dispositivo se soportan distintos modos de seguridad a la hora de establecer comunicaciones. Estos modos son:

- **Modo 1**: no activa ningún mecanismo de seguridad.

- **Modo 2**: los mecanismos de seguridad se inician tras el establecimiento del enlace y antes del establecimiento del canal lógico L2CAP (*service level-enforced*). En este modo un gestor de seguridad controla el acceso según el servicio al que se quiere llegar.
- **Modo 3**: los mecanismos de seguridad se aplican antes del establecimiento del enlace, ejecutándose los procedimientos descritos en el apartado Seguridad *Legacy*.
- **Modo 4**: se introduce a partir de la versión 2.1. Al igual que en el modo 2, los mecanismos de seguridad se aplican tras establecerse el enlace (*service-level enforced*), pero se utiliza el recién añadido a la especificación SSP para llevar a cabo la generación de claves, autenticación y encriptación.

Cualquier dispositivo de versión 2.1 o posterior está obligado a implementar el modo 4 de seguridad. En caso de ser anteriores, se permite utilizar el modo 3 por compatibilidad, considerándose los modos 1 y 2 obsoletos.

A partir de la versión 4.1, con la inclusión de *Secure Connections*, se añade un nuevo modo de seguridad llamado ***Secure Connections Only Mode***. Este modo de funcionamiento obliga a que los dispositivos solo establezcan comunicación con otros que puedan ejecutar este nivel de seguridad. Este modo se define cuando el nivel de seguridad en un entorno se considera prioritario a establecer comunicación. También se le denomina FIPS *Mode*, por utilizar algoritmos aprobados por la FIPS (*Federal Information Processing Standard*).

3.3. Seguridad en Bluetooth LE

La especificación de seguridad de Bluetooth LE se introduce a la vez que la propia versión, con la versión 4.0. BLE se caracteriza por implementar los mecanismos de seguridad en el host en lugar de en el *controller*, como sucede en BR/EDR, a través del protocolo SMP (*Secure Manager Protocol*). Asimismo, BLE introduce dos nuevas funcionalidades a su modelo de seguridad, privacidad LE y firmado de datos.

Por el tipo de escenarios para los que está orientado BLE, un dispositivo en fase de descubrimiento transmite mensajes de *advertising* que dejan al descubierto su dirección Bluetooth, generando un riesgo mayor que en los clásicos escenarios BR/EDR, ya que su ubicación es probable que sea accesible y no cambie en el tiempo, y permitiendo que se haga un seguimiento de un nodo. Para paliar esta situación, BLE introduce el concepto de privacidad LE, que se basa en que los dispositivos modifiquen su dirección cada cierto tiempo mediante el uso de una clave llamada IRK (*Identity Resolving Key*).

El firmado de datos se basa en el uso de una clave denominada CSRK (*Connection Signature Resolving Key*) compartida durante el proceso de *pairing* y que se utiliza para firmar los datos y que el receptor pueda verificar su autenticidad en escenarios en los que el modo de seguridad activado no permita el cifrado.

Existen actualmente dos clases de seguridad en BLE: *LE Legacy Pairing* y *LE Secure Connections*, introducido este último con la versión 4.2. A grandes rasgos, estos dos modelos de seguridad son una adaptación de BR/EDR a LE de SSP y de *Secure Connections* respectivamente. El uso de uno u otro modelo determina los mecanismos para la generación de claves. Dado que los procesos llevados a cabo por SMP son análogos para ambos modelos, describiremos todo el proceso de manera general, matizando las diferencias para cada modelo.

El procedimiento de seguridad de SMP se divide en tres fases:

- Fase 1: Intercambio de parámetros de emparejamiento
- Fase 2: Generación de claves
- Fase 3: Distribución de claves

3.3.1. Fase 1: Intercambio de parámetros de emparejamiento

Cuando un dispositivo trata de iniciar un proceso de emparejamiento, el primer paso es conocer las características técnicas y los requisitos impuestos por el otro dispositivo. Esta comunicación se inicia con un mensaje SMP de *Pairing_Request* o *Security_Request* (según inicie la solicitud el maestro o el esclavo respectivamente), que, al ser respondido, permite que ambos dispositivos conozcan los siguientes parámetros:

- Capacidades I/O del dispositivo
- Requerimientos de autenticación (con *flag* de necesidad de MITM)
- Capacidad OOB del dispositivo
- Tamaño de clave de encriptación
- Claves específicas a distribuir

Los tres primeros parámetros son los utilizados para seleccionar el modelo de asociación que se utilizará posteriormente en el emparejamiento. Estos modelos de asociación son similares a los vistos en BR/EDR, con las opciones *Just Works*, OOB y *Passkey Entry* para *LE Legacy Pairing*. *LE Secure Connections* añade además la opción de usar el modelo *Numeric Comparison*.

3.3.2. Fase 2: Generación de claves

La fase de generación de claves difiere según el modelo de seguridad que estén utilizando los dispositivos.

3.3.2.1. LE Legacy Pairing

Para establecer cifrado durante el proceso de emparejamiento, *Legacy Pairing* proporciona una clave de 128 bits denominada STK (*Short Term Key*). Esta clave requiere a su vez de otra clave temporal, también de 128 bits, denominada TK para su cálculo. La clave TK es básicamente un número aleatorio que se comparte entre ambos dispositivos gracias al modelo de asociación seleccionado. Para elegir el modelo de asociación se sigue el siguiente razonamiento:

- Si ambos dispositivos soportan capacidades fuera de banda, se ignora el resto de parámetros y se usa el modelo OOB.
- En caso contrario, se verifica el *flag* de protección MITM. Si su valor es 0 para ambos, se utiliza el modelo *Just Works*.
- En caso contrario, se revisan las capacidades I/O y se decide en base a ello el uso de *Just Works* o de *Passkey Entry*.

		Iniciator			
		OOB Sí	OOB No	MITM SÍ	MITM No
Responder	OOB Sí	Uso de OOB	Comprobación MITM	-	-
	OOB NO	Comprobación MTM		-	-
	MITM Sí	-	-	Uso Capacidades IO	
	MITM NO	--		Uso Capacidades IO	Uso Just Works

Tabla 6: Reglas del uso de flags OOB y MITM en pairing Legacy [2.c]

La compartición de la clave TK se da del siguiente modo:

- *Just Works*: TK se establece a 0 en ambos dispositivos, siendo por tanto una clave no autenticada, además de no segura.
- *PassKey Entry*: se sigue el procedimiento visto en el punto 3.2.2.2., solicitando al usuario introducir un PIN de 6 dígitos idéntico en ambos dispositivos o bien mostrándolo en uno para ser introducido en el otro. Mediante este PIN se genera la clave TK.
- OOB: la distribución de la clave TK se realiza por este medio fuera de banda.

Una vez los dispositivos han compartido la clave TK, el dispositivo que ha iniciado la comunicación genera un número aleatorio Mrand y un valor Mconfirm resultado de aplicar la función c1 sobre Mrand y TK. A su vez, el dispositivo que responde hace lo propio, siendo sus valores designados como Srand y Sconfirm. A continuación se envían sus valores M/Sconfirm primero y M/Srand después (siguiendo el mismo procedimiento de envío de *commitments* visto en SSP *Numeric Comparison*) y recalculan el valor de *confirm* del contrario, a fin de verificar que la TK que comparten es idéntica. Tras confirmarse, ambos generan la clave STK a partir de la clave TK y los números aleatorios Mrand y Srand.

La clave STK será utilizada para cifrar las comunicaciones y realizar la distribución de claves.

3.3.2.2. LE Secure Connections

La diferencia fundamental de *LE Secure Connections* con *Legacy Pairing* se basa en que únicamente genera una clave durante el proceso de emparejamiento, denominada como LTK (*Long Term Key*). Asimismo, *Secure Connections* incluye *Numeric Comparison* como modelo de asociación extra sobre los presentes en *Legacy Pairing*. La elección del modelo de asociación

sigue un razonamiento similar al de *Legacy Pairing* con la excepción de que se añade *Numeric Comparison* entre las opciones según las capacidades I/O y que OOB es seleccionado aunque solo un dispositivo lo soporte.

		Iniciator			
		OOB Sí	OOB No	MITM SÍ	MITM No
Responder	OOB Sí	Uso de OOB		-	-
	OOB NO	Uso de OOB	Comprobación MTM	-	-
	MITM Sí	-	-	Uso Capacidades IO	
	MITM NO	--		Uso Capacidades IO	Uso Just Works

Tabla 7: Reglas del uso de flags OOB y MITM en LE SC [2.c]

El procedimiento de emparejamiento con *Secure Connections* se compone de tres etapas, similares a las estudiadas en SSP de BR/EDR.

- **Intercambio de claves públicas**

Los dispositivos generan una pareja de claves pública-privada a partir de una curva ECDH P-256 y se envían mutuamente las claves públicas. A continuación pasan a generar a partir de estas un secreto compartido DHKEY.

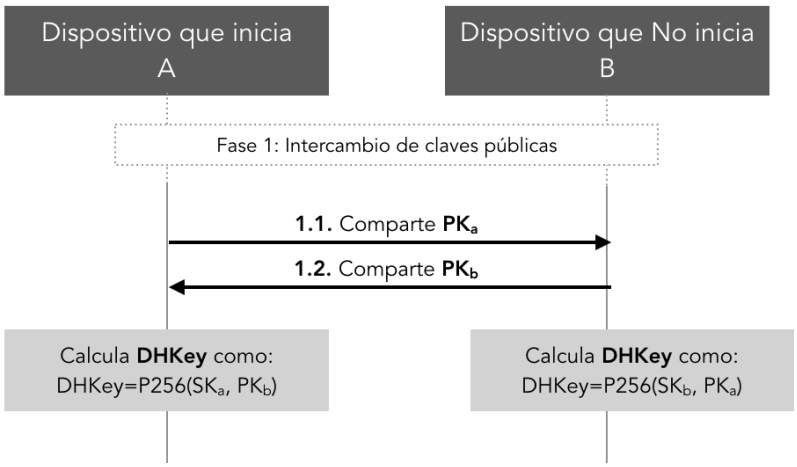

Ilustración 23: Intercambio claves públicas LE[2.c]

- **Autenticación fase 1**

En función del modelo de asociación seleccionado previamente, se realiza la primera fase de la autenticación de los dispositivos. El procedimiento concreto utilizado en cada modelo es idéntico al descrito en el punto 3.2.2.2., por lo que no se detallará de nuevo.

- **Autenticación fase 2**

En esta fase se calcula la clave LTK de 128 bits que se utilizará para cifrar las comunicaciones. Para ello, introducen en una función f5 los valores aleatorios

compartidos durante la autenticación fase 1, las direcciones Bluetooth de los dispositivos y el secreto DHKEY que se comenzó a calcular tras el intercambio de claves públicas. La función f5 tiene como salida un registro de 256 bits, de los cuales se extrae la clave LTK seleccionando los 128 bits menos significativos y la denominada MacKey seleccionando los 128 bits más significativos. Esta MacKey es la utilizada para realizar un nuevo cómputo (en esta ocasión mediante una función f6) que permita que los dispositivos verifiquen si el proceso ha terminado en éxito.

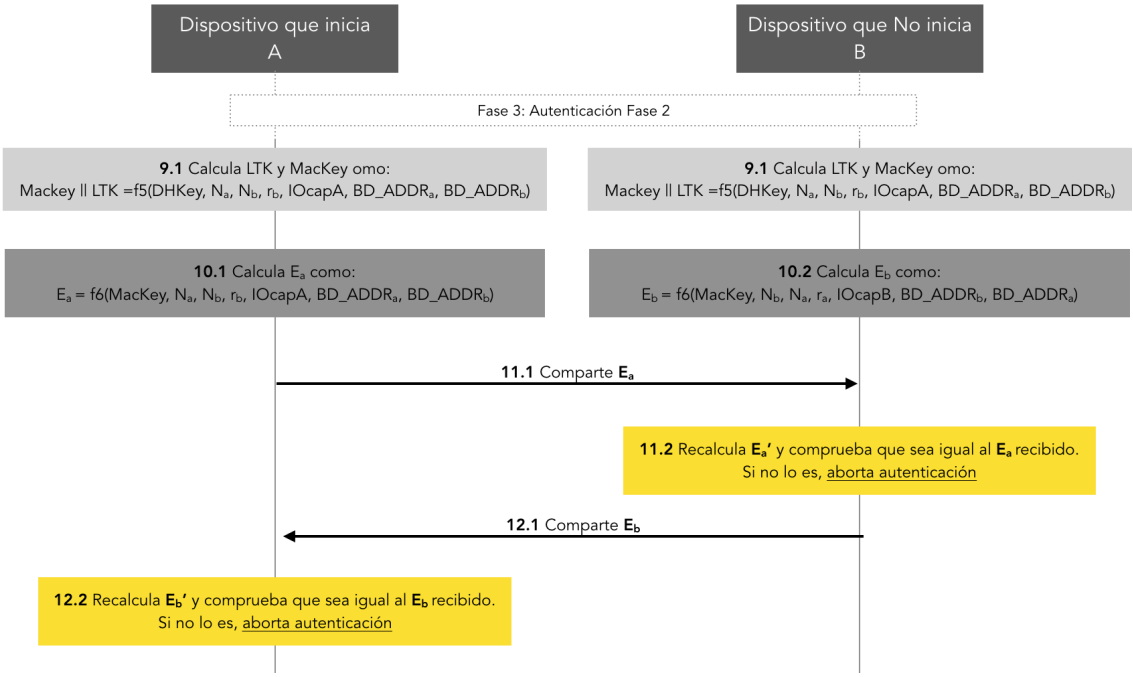

Ilustración 24: Fase 2 de Autenticación LE Secure Connections [2.c]

3.3.3. Fase 3: Distribución de claves

Una vez se han cifrado las comunicaciones, los dispositivos comienzan la distribución del resto de claves adicionales según la negociación llevada a cabo durante el intercambio de parámetros de emparejamiento.

En *LE Legacy Pairing*, a través de la conexión cifrada mediante la clave STK, se distribuyen las siguientes claves:

- LTK: clave de 128 bits de mayor durabilidad que los dispositivos guardan para nuevas conexiones entre ellos.
- EDIV y RAND: estos registros de 16 y 64 bits respectivamente son complementarios a la clave LTK, y se utilizan para controlar el tiempo de vida de esta y saber cuándo es necesario generar una nueva LTK.
- IRK: clave utilizada para la privacidad LE.
- CSRK: clave utilizada para el firmado de datos LE.

En el caso de *Secure Connections*, dado que la clave establecida durante la fase 2 es la LTK, desaparece la necesidad de generar los registros EDIV y RAND, por lo que en la fase de distribución de claves solo se comparten las claves IRK y CSRK.

3.3.4. Cifrado e integridad

Bluetooth LE, tanto en modo *Legacy Pairing* como en modo *Secure Connections*, utiliza el algoritmo AES-CCM para los servicios de cifrado y de integridad de mensajes.

3.3.5. Modos de seguridad

Bluetooth LE cuenta con dos modos de seguridad que marcan los requisitos de la comunicación establecida con otros dispositivos, clasificados asimismo en distintos niveles. Estos modos son:

- Modo 1, dividido en:
  - Nivel 1: sin seguridad establecida.
  - Nivel 2: emparejamiento sin autenticación con cifrado.
  - Nivel 3: emparejamiento con autenticación y con cifrado.
  - Nivel 4: emparejamiento con autenticación *LE Secure Connections* y con cifrado (añadido con la versión 4.2). Se considera el modo más seguro de Bluetooth LE.

- Modo 2, dividido en:
  - Nivel 1: emparejamiento sin autenticación con firma de datos.
  - Nivel 2: emparejamiento con autenticación y con firma de datos.

3.4. Resumen y conclusiones

Como se ha podido observar a lo largo de este punto, la especificación Bluetooth ha sufrido una intensa evolución en lo que respecta a sus protocolos de seguridad en los últimos años. Partiendo de un modelo *legacy* con medidas limitadas y algoritmos de cifrado y generación de claves que a día de hoy se encuentran obsoletos y que se pueden vulnerar fácilmente con equipamiento básico, Bluetooth ha ido incorporando mejoras que han permitido fortalecer sus defensas frente a ataques y accesos no autorizados.

La incorporación de *Secure Simple Pairing* con la versión 2.1 fue el primer paso para esta evolución, fortaleciendo la generación de claves de sesión entre terminales con el uso de curvas elípticas de 192 bits y un esquema de clave pública-privada. Asimismo, cuando las capacidades I/O de los terminales así lo permitan, los nuevos modos de emparejamiento añadidos ofrecen la posibilidad de establecer enlaces con protección contra escucha pasiva y ataques MITM en el mejor de los casos posibles. Queda claro en este punto que el uso de *Just Works* como modo de emparejamiento es sin lugar a dudas el más inseguro de todos, y que su uso debe ser restringido a escenarios en los que el riesgo sea asumible y no quede más remedio por el tipo de dispositivo con el que se trata. Un buen ejemplo de esta situación es el de los dispositivos de audio (auriculares especialmente), que debido a sus características están limitados a *Just Works*. Es razonable considerar, sin embargo, que el tipo de uso que a priori se les da (generalmente en movimiento por la zona urbana) hacen francamente difícil llegar a la situación de poder ejecutar un ataque MITM, no digamos ya

mantenerlo en el tiempo salvo que se siga a la víctima, lo que podemos definir como una casuística improbable.

Con la incorporación de *Secure Connections*, Bluetooth termina de fortalecer el resto de algoritmos utilizados en sus distintas fases de seguridad. Además de mejorar de nuevo la generación de claves ampliando la curva elíptica hasta los 256 bits, con la versión 4.1 se descartan los algoritmos E1 y E0 para pasar a usar HMAC-SHA-256 y AES-CCM en autenticación y en encriptación respectivamente. Esto no solo aumenta la entropía de las claves utilizadas, sino que, en el caso de la encriptación, AES-CCM ofrece también integridad de mensaje, dificultando en mayor medida la ejecución de ataques MITM. Asimismo, *Secure Connections* se introduce como un modo con el que activar el máximo nivel de seguridad, negando la conexión en caso de que el dispositivo destino no comparta este nivel de protección.

	Bluetooth BR/EDR	
	Secure Simple Pairing (SSP)	**Secure Connections (SC)**
Versiones	**2.1 a 4.0**	**A partir de 4.1**
Generación de claves (pairing)	- 4 Métodos Pairing (Numeric Comparison, Passkey Entry, Out Of Band, Just Works) - ECDH Acuerdo de clave (Algoritmo P-192 Elliptic Curve) - LK Clave de enlace compartida - Algoritmos de generación de claves basados en HMAC-SHA-256 - Función E3 para generación de la clave de cifrado Kc a partir de LK	- 4 Métodos Pairing (Numeric Comparison, Passkey Entry, Out Of Band, Just Works) - ECDH Acuerdo de clave (Algoritmo P-256 Elliptic Curve) - LK Clave de enlace compartida - Algoritmos de generación de claves basados en HMAC-SHA-256 - Función h3 (HMAC-SHA-256) para generación de la clave de cifrado AES a partir de LK
Autenticación	Autenticación *Legacy* - Unidireccional por defecto - Mutua Opcional - Algoritmo E1 basado en SAFER+	Secure *Authentication* - Autenticación Mutua Obligatoria - HMAC-SHA-256
Cifrado	Algoritmo E0	AES-CCM
Integridad	No	AES-CCM
Protección Eavesdropping	Sí, independientemente del modelo de pairing utilizado, aunque el tráfico de pairing y de autenticación sea capturado a través de escuchas ilícitas, debe resolverse el problema criptográfico de clave pública (ECDH) para resolver la clave LK.	

Tabla 8: Resumen capacidades de seguridad BR/EDR [5]

Por su parte, la llegada de BLE con la versión 4.0 fue, acorde con la madurez del estándar, acompañada por unos mecanismos de seguridad más avanzados, en comparación con los primeros pasos de BR. BLE cuenta de inicio con modos de emparejamientos más seguros, semejantes a los diseñados para SSP, y de hecho los algoritmos implementados en autenticación y encriptación son ya basados en AES-CCM, adelantándose a las mejoras que BR/EDR no verá hasta la versión 4.1. Sin embargo, al recurrir a una clave temporal para la generación de la clave de encriptación (fácilmente interceptable), no se considera que esté protegido frente a escuchas pasivas. Por otro lado, las nuevas problemáticas introducidas por los distintos escenarios y casos de uso para los que se plantea la incorporación de BLE comienzan a tenerse en cuenta con la incorporación de la privacidad LE, que permite ocultar la dirección de los dispositivos y protegerse frente a amenazas de vigilancia.

Con la versión 4.2 llegaría a BLE su propia versión de *Secure Connections*. Implica la necesidad de compartir dicho nivel de protección para poder establecer conexión, además de aumentar la protección frente a escuchas pasivas al utilizar en esta ocasión la clave DHKEY para la generación de la clave de encriptación, que no se distribuye y que por tanto dificulta un ataque de este tipo. Asimismo, la generación de claves pasa a implementarse mediante curvas ECDH de 256 bits, poniéndose al nivel de protección de BR/EDR en modo SC.

	Bluetooth LE	
	Legacy Pairing	**Secure Connections (SC)**
Versiones	**4.0 y 4.1**	**A partir de 4.2**
Generación de claves (pairing)	- Generación clave de cifrado STK - Uso de TK, clave temporal - 3 Métodos Pairing (Passkey Entry, Out Of Band, Just Works) - Algoritmo de generación de claves AES-128	- Generación clave de cifrado LTK. - 4 Métodos Pairing (Numeric Comparison, Passkey Entry, Out Of Band, Just Works) - ECDH Acuerdo de clave (P-256 Elliptic Curve). - Algoritmo de generación de claves AES-CMAC
Autenticación	AES-CCM	AES-CCM
Cifrado	AES-CCM	AES-CCM
Integridad	AES-CCM	AES-CCM
Protección Eavesdropping	No. No usa ECDH para la generación de la clave de cifrado STK, sino que se genera usando una clave temporal TK que puede ser capturada al ser intercambiada en un enlace no cifrado. Solo estaría protegido en caso de pairing OOB, si se configura el canal	Sí. Utiliza algoritmos ECDH (P-256) para la generación de la clave DHKey compartida que se utiliza también para generar la clave de cifrado LTK. Dado que la DHKey no es distribuida, es muy difícil averiguar la clave LTK.

Tabla 9: Resumen capacidades de seguridad BLE [5]

Se aprecia pues, a raíz de este estudio cómo el nivel de seguridad de la comunicación ha ido aumentando satisfactoriamente con cada nueva versión, pero que por cuestiones de retrocompatibilidad viene fuertemente determinada por los eslabones más débiles de la comunicación, esto es, los terminales más antiguos, que puedan forzar a otros dispositivos más modernos a trabajar con mecanismos desfasados, así como por las capacidades de I/O de los terminales conectados, que hagan imposible ejecutar el máximo nivel de protección. En el próximo punto abordaremos las distintas vulnerabilidades conocidas, así como los ataques más sufridos en el entorno Bluetooth.

4. Vulnerabilidades y amenazas

Como se ha descrito en detalle a lo largo del apartado 3, la tecnología Bluetooth ha evolucionado y mejorado en cuanto a seguridad se refiere a lo largo de los años. Desde unas primeras versiones donde prácticamente no se la consideraba en absoluto, hasta la reciente versión 5.1, gran parte de los esfuerzos en el desarrollo de esta tecnología han ido orientados a mejorar su robustez para hacerla más segura en dos aspectos claves: a nivel computacional y a nivel funcional.

A nivel computacional se han sustituido algoritmos de generación de claves y de cifrado por otros más robustos. En BR/EDR, como puede observarse en la tabla 5, funciones como E1 basadas en SAFER+ han dado paso a la utilización de algoritmos HMAC-SHA256 y al uso de ECDH, que a su vez ha pasado de P-192 en SSP a P-256 en SC. En el caso de LE pasan de utilizarse algoritmos de generación de claves AES-128 a utilizarse algoritmos basados en AES-CMAC.

En cuanto a nivel funcional, los mecanismos de emparejamiento se sofistican para evitar ataques de suplantación y de espionaje. Se descarta por completo la autenticación unidireccional, siendo obligatoria la autenticación mutua, y se utilizan parejas de claves pública-privada, de manera que no se compartan en el canal claves que pueden ser interceptadas. Además, *Secure Connections* incluye mecanismos de integridad con AES-CCM, algo que no había sido incluido hasta la versión 4.1. Sin embargo, a pesar de todo, continúa teniendo aspectos vulnerables que suponen una brecha para posibles ataques maliciosos. Por tanto, procede analizar los puntos débiles de la seguridad de Bluetooth, entendiendo que existen amenazas que pueden materializarse gracias a ataques que explotan vulnerabilidades concretas. En este apartado discutiremos cuáles son estas vulnerabilidades y cuáles son los ataques más conocidos que permiten explotarlas, así como una lista de contramedidas o guía de buenas prácticas que permitan mitigar las posibilidades del atacante y el impacto de sus ataques.

4.1. Vulnerabilidades

A continuación, se recogen las vulnerabilidades conocidas hasta el momento para Bluetooth, en el siguiente formato:

Nº Id	**Título Descriptivo de la vulnerabilidad**
Sistemas a los que afecta (BR/EDR y/o LE)	Versiones a las que afecta

Proceden tanto del último informe del *National Institute of Standars and Technology* (NIST) [6], como de aquellas registradas en CVE (*Common Vulnerabilities and Exposures*) [7] y de los últimos ataques que se han descubierto hasta la fecha.

1	El modo de seguridad 1 carece de mecanismos de seguridad
BR/EDR y LE	1.0, 1.1, 1.2, 2.0

Los dispositivos BR/EDR que usan el modo de seguridad 1 son inherentemente inseguros. Para dispositivos 2.0 y anteriores, se recomienda encarecidamente el Modo de seguridad 3 (seguridad de nivel de enlace). LE sí tiene un nivel 4 dentro del Modo 1 que incluye autenticación, emparejamiento y cifrado, por lo que se recomienda su uso.

2	Las claves de link basadas en claves unitarias son estáticas y se reutilizan
BR/EDR	1.0, 1.1

En las primeras versiones de Bluetooth BR/EDR se permitía el uso de claves unitarias para generar las claves de link. Estas eran estáticas, generadas habitualmente durante la instalación, por lo que el dispositivo siempre utilizaba la misma clave de link. Esto supone un problema grave de seguridad, dando lugar a espionaje y suplantación, aunque las comunicaciones se cifren (dado que la clave está comprometida).

Ocurre de manera similar con claves maestras propagadas dentro de una piconet, donde además es una clave compartida que permite espiar y suplantar los enlaces con todos los esclavos de la red.

3	Los PINS suelen ser cortos y fáciles de adivinar
BR/EDR	1.0, 1.1, 1.2, 2.0

En los modos de seguridad *Legacy* previos a la versión 2.1, se utiliza un código PIN para la generación de la clave de inicialización a raíz de la cual se generará la clave de enlace. Las personas tienden a seleccionar PINs cortos y débiles, que al ser utilizados en la generación de claves durante el emparejamiento comprometen la seguridad al ser fáciles de adivinar (p.ej: 0000)

4	Las claves de cifrado se repiten cada 23,3 horas
BR/EDR	1.0, 1.1, 1.2, 2.0

Como puede verse en la Ilustración 16, la clave de encriptación en los modos de seguridad Legacy se genera en base a los parámetros BD_ADDR, EN_RAND, Kc y una referencia de reloj del dispositivo master. Esta referencia se genera con un registro de 28 bits que cambia cada 312,4 μs, por lo que se repite cada 23,3 horas. Esto presenta una vulnerabilidad criptográfica grave, ya si una conexión se mantiene por más de 23,3 horas la repetición una secuencia clave permitiría descifrar el mensaje original.

5	El modelo *Just Works* no proporciona protección MITM
BR/EDR y LE	2.1, 3.0, 4.0, 4.1, 4.2

Aunque a partir de la versión 2.1 ya se implementa SSP, durante la primera fase de autenticación utilizando Just Works no hay confirmación por el usuario (al contrario que con *Numeric Comparison*), por lo que se pueden producir ataques MITM.

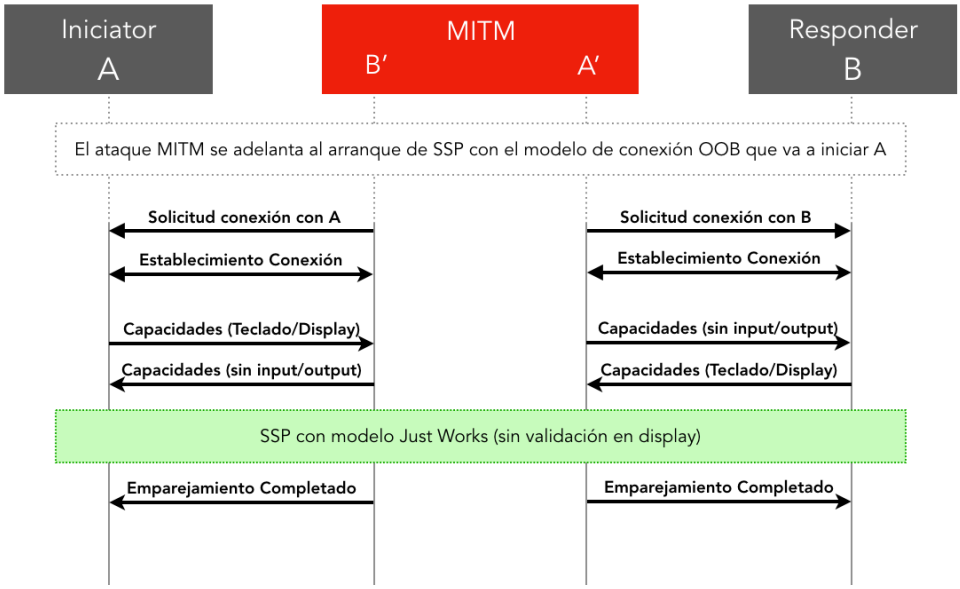

Ilustración 25 Ataque MITM usando Just Works

Para implementar este tipo de ataques, el atacante debe contar con dos dispositivos: uno para responder al *iniciator* (A) y otro para iniciar la comunicación con el *responder* (B).

6	Las claves de ECDH en SSP pueden ser débiles (CVE-2018-5383)
BR/EDR y LE	2.1, 3.0, 4.0, 4.1, 4.2

SSP utiliza el protocolo de claves de curva elíptica Diffie-Hellman (ECDH) para generar el par de claves pública-privada con las que se calcula la clave de enlace que permite la comunicación cifrada entre los dispositivos. Además de compartir las claves públicas, los dispositivos deben estar de acuerdo con los parámetros de curva elíptica que se utilizan.

Sin embargo, en 2018 se identificó un problema con los parámetros de ECDH. Si no están validados por la implementación del algoritmo criptográfico, cuando se calcula la clave compartida se reduce considerablemente el esfuerzo del atacante para obtener la clave privada del dispositivo, inyectando una clave pública no válida para determinar la clave de sesión con alta probabilidad, si se encuentra dentro del alcance inalámbrico. El atacante podría entonces interceptar y descifrar pasivamente todos los mensajes del dispositivo y falsificar o inyectar mensajes maliciosos. [8]

7	El uso de passkeys estáticas facilita ataques MITM
BR/EDR y LE	2.1, 3.0, 4.0, 4.1, 4.2

Aunque utilizando *passkeys* en SSP se da protección frente a ataques MITM, solventando por ejemplo los problemas de vulnerabilidad de *Just Works*, si las claves utilizadas son estáticas puede suponer un problema para garantizar la seguridad. Es necesario que los códigos numéricos utilizados como PIN durante el emparejamiento sean aleatorios y únicos, o pueden precisamente facilitar un ataque de estas características. [10]

8	Retrocompatibilidad en modos de seguridad
BR/EDR y LE	2.1, 3.0, 4.0, 4.1, 4.2

Aunque un dispositivo soporte el Modo 4 (es decir, versión 2.1 o posterior) pueden retroceder a cualquier otro modo de seguridad anterior cuando se conecta con un dispositivo antiguo que no sea compatible (versión 2.0 y anteriores). Es decir, podría incluso retroceder al modo 1, que carece por completo de mecanismos de seguridad.

<table>
<tr><th rowspan="2">Versión</th><th colspan="2">Modo más seguro para emparejarse con</th></tr>
<tr><th>2.0 o inferior</th><th>2.1 o superior</th></tr>
<tr><td>4.2</td><td rowspan="9">Modo 3</td><td rowspan="5">Modo 4
(Obligatorio)</td></tr>
<tr><td>4.1</td></tr>
<tr><td>4.0</td></tr>
<tr><td>3.0</td></tr>
<tr><td>2.1</td></tr>
<tr><td>2.0</td><td rowspan="4">Modo 3</td></tr>
<tr><td>1.2</td></tr>
<tr><td>1.1</td></tr>
<tr><td>1.0</td></tr>
</table>

Tabla 10 Recomendación uso versiones para emparejamiento [5]

9	Intentos de autenticación ilimitados
BR/EDR y LE	Todas

Como mecanismo de seguridad, Bluetooth introduce un intervalo de espera exponencialmente creciente entre intentos de autenticación sucesivos. Sin embargo, esto no aplica a los *challenges*, por lo que un atacante podría recopilar un gran número de respuestas a estas solicitudes (que están cifradas con la clave de enlace), y extraer información de las mismas.

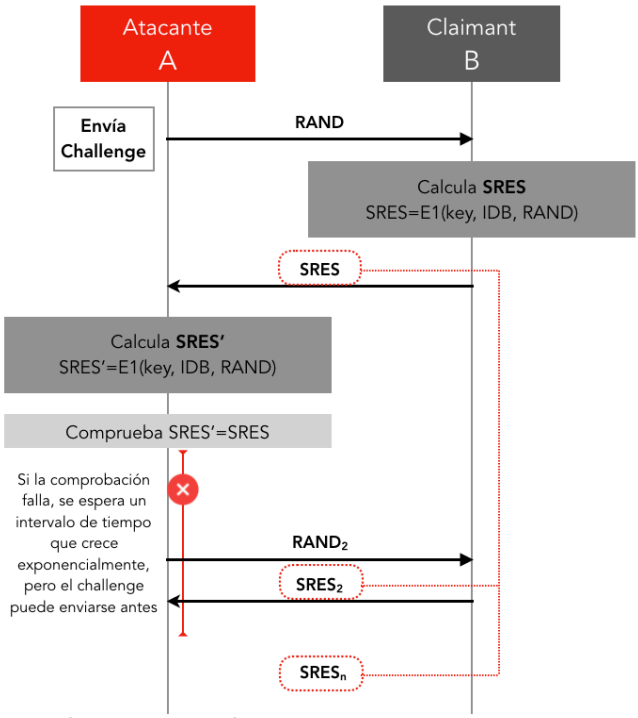

Ilustración 26 Obtención de información con intentos reiterados de challenge

Se debe incluir un mecanismo para evitar solicitudes de *challenges* ilimitadas o para cambiar la clave privada después de varios intentos.

10	La clave maestra utilizada para el cifrado de broadcast se comparte entre todos los dispositivos de la piconet
BR/EDR y LE	1.0, 1.1, 1.2, 2.0, 2.1, 3.0

En los modelos en los que se forma una piconet compuesta por distintos dispositivos en formato broadcast, las claves secretas compartidas entre el maestro y los esclavos facilitan los ataques de suplantación, ya que la clave utilizada es la misma y la posibilidad de introducirse en la comunicación crece con el número de partes implicadas.

11	El algoritmo de cifrado E0 es relativamente débil
BR/EDR	1.0, 1.1, 1.2, 2.0, 2.1, 3.0, 4.0

Se ha demostrado que el algoritmo de cifrado E0 presenta vulnerabilidades claras en cuanto a fuerza algorítmica. De hecho, no está considerado un estándar FIPS. Desde 1999, diversos experimentos y estudios han demostrado su vulnerabilidad y han puesto de manifiesto la posibilidad de romper este algoritmo. El último de ellos fue publicado por Lu, Meien y Vaudenay en 2005, y consistía en un ataque de correlación condicional, donde se necesitaron únicamente los primeros 24 bits de $2^{23.8}$ tramas y 2^{38} operaciones para descifrar la clave.

No aplicaría a Bluetooth LE, ya que el algoritmo de cifrado utilizado en ese caso es AES-CCM. [11]

12	La privacidad en BR/EDR se puede ver comprometida si se asocia la BD_ADDR a su usuario
BR/EDR	1.0, 1.1, 1.2, 2.0, 2.1, 3.0

La dirección BD_ADDR es la clave que identifica a cada dispositivo en Bluetooth, y se trata de un identificador único extendido de 48 bits (EUI-48) creado de acuerdo con el estándar IEEE 802-2014. Está formada por: una parte NAP, utilizada para la sincronización de los saltos de frecuencia; una parte UAP, que sirva como semilla de los RNG; y una parte LAP, asignada por el vendedor para identificar las comunicaciones del dispositivo.

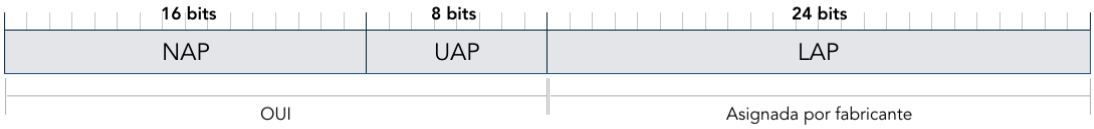

Ilustración 27: Estructura BD_ADDR

Al tratarse de una clave estática que es utilizada en partes del proceso de emparejamiento, autenticación y encriptación, conocerla para un dispositivo concreto hace que sus conexiones y actividades puedan verse comprometidas.

En LE se implementan políticas de privacidad específicas para reducir este riesgo y hacer que el dispositivo no sea tan fácilmente rastreable. Este sistema cambia la dirección del dispositivo cada cierto periodo de tiempo, y mediante un algoritmo de resolución de identidad (IRK) el dispositivo remoto resuelve que la dirección recibida pertenece a un dispositivo de confianza.

13	La autenticación simple es vulnerable a ataques MITM
BR/EDR y LE	1.0, 1.1, 1.2, 2.0, 2.1, 3.0

Los métodos de autenticación simple, donde se realiza *one-way challenge/response*, están especialmente dirigidos a cuando un dispositivo es de solo lectura, y el otro es de lectura/escritura (como podría ser el caso de un *smartphone* que se sincroniza con unos auriculares, por ejemplo). Al hacerse la comprobación en un solo sentido, son vulnerables a ataques de suplantación.

Es recomendable, por tanto, utilizar métodos de autenticación mutua, donde se verifique que ambos dispositivos son legítimos, realizando primero el procedimiento de autenticación de A hacia B, y posteriormente realizándolo en la dirección opuesta.

14	Los métodos de emparejamiento *legacy* de LE no protegen frente a ataques de escucha
LE	4.0, 4.1

Si el atacante consigue capturar las claves (como LTK, CSRK o IRK) durante el proceso de emparejamiento, no hay forma de ser consciente ni detener la

escucha de la comunicación.

15	Las claves de enlace se pueden almacenar de manera poco segura
BR/EDR y LE	Todas

Cuando dos dispositivos se emparejan, almacenan sus claves para conexiones futuras. Si no se securiza también el almacenamiento de las claves, un atacante podría leerlas y/o modificarlas. Se recomienda el uso de códigos de acceso a las mismas.

16	Generadores de números aleatorios (RNG) débiles
BR/EDR y LE	Todas

El RNG puede producir número estáticos, o periódicos, que reducirían la efectividad de los mecanismos de seguridad. Deberían utilizarse RNG fuertes basados en estándares NIST.

17	La longitud de la clave de cifrado es negociable
BR/EDR y LE	1.0, 1.1, 1.2, 2.0, 2.1, 3.0

La versión 3.0 y anteriores permiten a los dispositivos negociar el tamaño de las claves de cifrado, hasta un byte. LE requiere un tamaño de clave mínimo de 7 bytes. NIST recomienda encarecidamente utilizar el modo *Secure Connection Only*, que requiere claves de 7 bytes (AES-CCM) tanto para BR/EDR como para LE.

18	No existe autenticación a nivel de usuario
BR/EDR y LE	Todas

La autenticación se concibe solo a nivel de dispositivo, por lo que para discriminar por usuarios es necesario incluir seguridad a nivel de aplicación en el desarrollo de la misma.

19	No existe seguridad extremo a extremo
BR/EDR y LE	Todas

Solo se securizan los enlaces individuales, por lo que los datos son descifrados en los puntos intermedios de la conexión. La seguridad extremo a extremo debe proporcionarse con mecanismos de control adicionales, implementados ad hoc y fuera de la estructura de protocolos de Bluetooth.

20	Los servicios de seguridad son limitados
BR/EDR y LE	Todas

Al igual que con la seguridad extremo a extremo, aspectos como la auditoría, no repudio, etc, no están contemplados dentro de Bluetooth y tienen que implementarse al margen.

21	La visibilidad activada por defecto hace vulnerables los dispositivos
BR/EDR y LE	Todas

Los dispositivos en modo visible son más propensos a ser atacados. Cualquier dispositivo BR/EDR/LE que debe entrar en modo detectable o conectable para emparejarse o conectarse solo debe hacerlo durante un tiempo mínimo. Un dispositivo no debe estar en modo detectable o conectable todo el tiempo.

22	En dispositivos emparejados, la autenticación mutua no siempre funciona en modos 3 o 4
BR/EDR y LE	1.0, 1.1, 1.2, 2.0, 2.1, 3.0

Dos dispositivos que ya hayan sido emparejados relajan la autenticación posterior. Por ejemplo, si el dispositivo A inicia la autenticación con B, la configuración del cifrado comenzará después de esa autenticación inicial. Si la configuración de cifrado es exitosa y es suficiente para satisfacer a B, entonces B puede no molestarse en autenticar A.

Los huecos de seguridad que dejan estas vulnerabilidades son explotables por gran variedad de ataques que se especificarán en el apartado siguiente, los cuales suponen una amenaza. Algunas son insalvables, mientras que en otras su riesgo se puede reducir considerablemente con buenas prácticas o contramedidas específicas.

Un aspecto importante que puede observarse es que la comunicación bluetooth ha evolucionado, pero los dispositivos que la implementan rara vez son actualizables. Es decir, aunque en los *smartphones* de última generación se incluyan versiones modernas del protocolo, muchas de las conexiones que establecen lo hacen con dispositivos antiguos, que no soportan aspectos como *Secure Conections*, o incluso SSP. Por tanto, es la retrocompatibilidad finalmente lo que más penaliza la seguridad en Bluetooth.

4.2. Amenazas y ataques

Cualquier sistema está expuesto a una serie de amenazas durante su funcionamiento, amenazas que se pueden ejecutar mediante ataques gracias a las vulnerabilidades que presente dicho sistema. Como cualquier tecnología de comunicación *wireless*, Bluetooth presenta una gran cantidad de amenazas posibles derivadas de que el medio de transmisión de la información sea la radio, accesible por cualquiera que cuente con una antena para el canal específico. A continuación, realizaremos un repaso de los ataques más extendidos que se realizan sobre el sistema Bluetooth de los dispositivos, clasificándolos según el tipo de amenaza que suponen para el usuario y relacionándolo con las vulnerabilidades implicadas cuando así proceda. [12]

4.2.1. Vigilancia

El primer paso para poder vulnerar la seguridad de otros dispositivos es detectarlos y obtener toda la información posible sobre ellos, tales como la dirección, el fabricante, los servicios Bluetooth ofrecidos, etc.

La herramienta más básica para la vigilancia y en general, para el inicio de cualquier ataque Bluetooth, es el paquete de Linux **hcitool**[13]. Este comando nos permite realizar un escaneo general desde nuestro terminal de ataque y detectar los dispositivos emitiendo en el rango de alcance de nuestra antena, obteniendo sus direcciones y nombres, y que da la posibilidad de ampliar dicha información mediante el envío de comandos HCI_inq. Esta información se suele complementar con la herramienta **sdptool**, que permite generar consultas dentro del protocolo SDP y obtener toda la información posible sobre los servicios que soporta el dispositivo atacado.

Ilustración 28: Detalle hcitool

Existen herramientas más sofisticadas basadas en estas utilidades, tales como BlueScanner, BluePrinting, RedFang, BT Audit, War-Nibling, Bluefish o BNAP.

4.2.2. Ofuscación

[14] Con el fin de poder ejecutar otros ataques más comprometedores, es habitual recurrir a técnicas de ofuscación con el fin de proteger la identidad del atacante en caso de que su actividad sea detectada.

A la hora de modificar los datos básicos para identificar a un dispositivo, es posible recurrir a los comandos básicos de Linux **hciconfig** (una versión Bluetooth análoga al comando de configuración de red **ifconfig**) y el comando **bdaddr**. Mientras que el primero permite modificar los campos de *device name* y *device class*, el segundo permite cambiar la dirección Bluetooth que utiliza el dispositivo atacante.

Sin embargo, una de las herramientas más usadas que permite automatizar la ocultación es **SpoofTooph**. Este comando permite automatizar hasta el nivel deseado las funciones de ofuscación y definir para el atacante un perfil Bluetooth (dirección, *device name*, *device class*) falso, proporcionando cinco modos de funcionamiento distinto:

- Escaneo de dispositivos cercanos y almacenamiento de la lista de ellos, permitiendo la selección de uno para clonar su perfil Bluetooth.
- Generación de un perfil aleatorio basado en dispositivos comunes en la industria.
- Definición manual de perfil específico.

- Clonación de perfil a partir de lista de terminales escaneados en el pasado.
- Modo incógnito: escaneo y clonación del primer perfil encontrado, ejecutado periódicamente según el parámetro de tiempo introducido.

4.2.3. Sniffing

[15] Se denomina *sniffing* o escucha pasiva a la captura del tráfico que está siendo transmitido por los dispositivos para su posterior análisis. Es evidente que la capacidad de un atacante de capturar el tráfico que emitimos por el canal radio es imposible de evitar o detectar, por lo que lo realmente determinante de este tipo de amenaza se basa en el nivel de seguridad que tienen nuestras comunicaciones en cuanto a encriptación y la confidencialidad que es capaz de proporcionarnos con los mecanismos de los que cuenta.

Realizar escuchas pasivas de comunicaciones Bluetooth no es una cuestión trivial debido al uso de FHSS para modificar el canal de transmisión de manera dinámica durante una comunicación. Asimismo, los *dongles* básicos usados para aportar conectividad Bluetooth a PCs no están diseñados para ser capaces de escuchar tráfico en modo promiscuo. Así, si bien es posible utilizar herramientas para tracear tráfico tales como hcidump, que captura los mensajes transmitidos entre el *host* y el *controller* del equipo atacante, es habitual recurrir a dispositivos especializados para esta clase de actividades, denominados *sniffers*.

Existe una gran variedad de *sniffers* en el mercado con un gran rango de precios, desde dispositivos económicos por menos de 100 euros accesibles para cualquier usuario, como los kits de desarrollo nRF5 de Nordic, a equipamientos de altas prestaciones para la investigación y precio muy elevado, del orden de decenas de miles de euros, como el Ellysis Bluetooth Explorer. El dispositivo más popular es el **Ubertooth One**, que presenta unas prestaciones muy eficientes por un precio de alrededor de 120 euros, convirtiéndolo en una herramienta de ataque muy accesible para cualquier *hacker*.

Ilustración 29: Dispositivo Ubertooth One

4.2.4. MITM

Un ataque Man-In-The-Middle se basa en que el atacante consiga interponerse entre dos dispositivos que tratan de comunicarse entre sí y emparejarse con cada uno de ellos. De este modo, el atacante sería capaz de engañar a los dispositivos involucrados haciéndoles creer que se encuentran emparejados de manera segura, pero ejerciendo como *proxy* entre ellos.

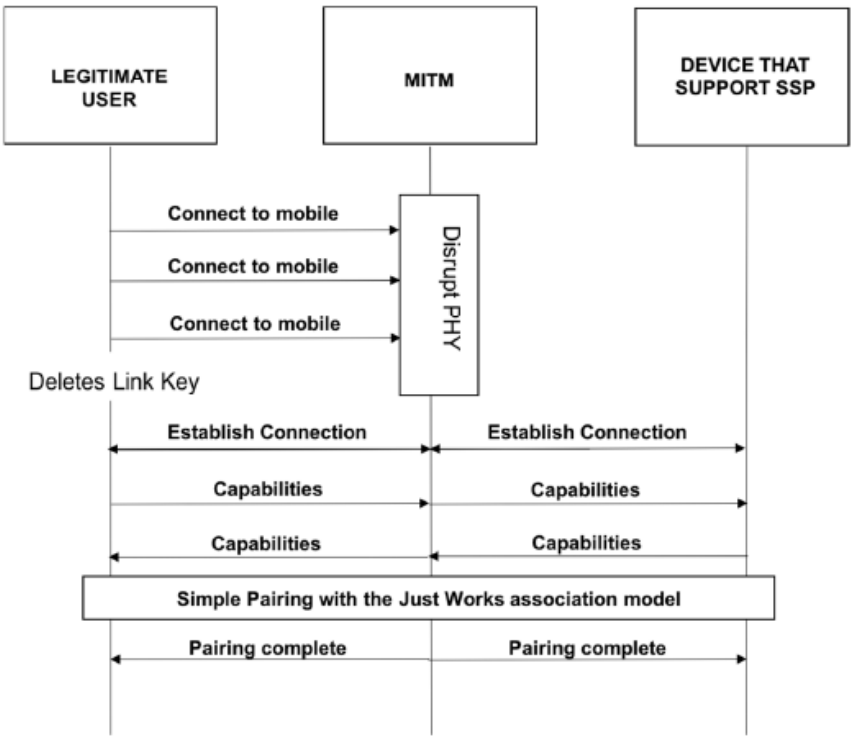

Ilustración 30: Detalle ejecución ataque MITM [12]

Si un atacante consigue ejecutar con éxito este emparejamiento fraudulento pasa a tener acceso total al tráfico que comparten los dos dispositivos, tráfico que además se encontrará en claro al ser el atacante el propio responsable de negociar la encriptación con cada una de las víctimas. Asimismo, el atacante tendría la posibilidad de modificar los mensajes que los terminales se envían y obligarles a ejecutar acciones no deseadas, en lo que se denominaría como un ataque MITM activo.

Este tipo de ataque precisa de ejecutarse durante el proceso de *pairing* entre dos dispositivos para poder llevarse a cabo. Es por eso que es habitual que venga precedido de un ataque de disrupción del medio físico que corte la comunicación entre las víctimas el suficiente tiempo como para que su sesión caduque y se vean forzados a realizar el proceso de *pairing* de nuevo.

Una de las herramientas más extendidas para la ejecución de ataques MITM es **Btlejuice** [16], un sofisticado software con interfaz gráfica que permite tanto tracear el tráfico transmitido entre las víctimas como enviar comandos a ejecutar mediante la réplica de órdenes interceptadas. Asimismo, Btlejuice permite realizar modificación de datos *on-the-fly*, definiendo patrones de datos que

deban ser sustituidos por los deseados. Otra herramienta común para la ejecución de ataques MITM es **Btproxy**.

4.2.5. Denegación de servicio

Toda actividad que pueda obstaculizar el funcionamiento del sistema de comunicación se encuadra dentro de los ataques de denegación de servicio. Existe un gran abanico de ataques de denegación de servicio con diversas capacidades e implicaciones sobre el usuario.

Uno de los ataques más conocidos es el denominado **bluejacking** [17]. Este se basa en el envío de mensajes a los dispositivos que se encuentren activados en modo visible a través de la funcionalidad vCard, que permite el envío de contactos sobre el protocolo OBEX. Si bien suele usarse con fines de promoción y *spam* y se considera un tipo de ataque no dañino, bluejacking es una técnica que puede resultar incómoda para el usuario si se usa en un modo continuo. Otras técnicas similares son **vCardBlaster** o **BlueSpam**. [14]

Con un mayor grado de gravedad, **BlueSmack** [18] se basa en el concepto del "ping de la muerte" conocido en el entorno IP. A partir del comando l2ping, que permite solicitar mensajes *echo* a otros dispositivos sobre el protocolo L2CAP, el atacante puede tratar de bloquear a víctimas con Bluetooth en modo visible especificando un tamaño de paquete generalmente de órdenes superiores a 600 bytes.

Asimismo, existe la posibilidad de realizar un ataque de denegación de servicio a nivel físico (lo que se conoce como ***sygnal jamming***) mediante el uso de un inhibidor de frecuencias diseñado para atacar el ancho de banda ISM. Sin embargo, hay que tener en cuenta que este tipo de ataque es menos habitual, no solo por el equipamiento necesario, sino porque la afectación que tiene (no solo todas las comunicaciones Bluetooth de la zona, sino también las comunicaciones WiFi) genera un impacto muy grande, con el consiguiente disparo de alerta de las víctimas. Además, siendo un ataque basado en la emisión de potencia, es fácilmente localizable el foco de esta y, por tanto, del atacante.

4.2.6. Acceso no autorizado a datos

Es posible aprovechar ciertas vulnerabilidades específicas para acceder de manera no deseada a información confidencial de un usuario a través de su sistema Bluetooth. **Bluesnarf** [19], el ataque más extendido y conocido dentro de esta categoría, se basa en explotar una vulnerabilidad del protocolo OBEX de transferencia de archivos para poder acceder a los ficheros de un dispositivo móvil. El atacante se aprovecha del servicio OBEX Push Profile (OPP), un servicio típicamente sin autenticación, para ejecutar consultas tipo GET pasando como parámetro el nombre del fichero. De este modo podría acceder a los ficheros del terminal cuyo nombre sea capaz de adivinar, así como los ficheros del sistema conocidos, siendo algunos objetivos habituales el calendario (/telecom/cal.vcs) o los contactos (/terminal/pb.vcf).

Un caso muy similar es el de **BlueBugging** [12], que se basa en aprovechar una vulnerabilidad descubierta en 2004 que permite explotar el protocolo RFCOMM, un protocolo de comunicaciones serie sobre L2CAP, para conectarse al dispositivo de la víctima sin su conocimiento y ejecutar comandos sobre el sistema. Esto le permite al atacante extraer información del dispositivo y, por ejemplo, extraerla mediante el envío de SMS.

Existe una gran variedad de ataques de este tipo, entre los que se encuentran algunos como BTCrack, Car Whisperer, HID Attack, HeloMoto, Bloover o Btaptap.

Un ataque de este tipo de especial gravedad es el denominado como **MouseJack** [21]. Este ataque se basa en controlar un equipo a partir de órdenes enviadas a través del receptor USB que suelen utilizarse para conectar ratones y teclados inalámbricos con ordenadores, ya sea portátiles o fijos. Si bien el estándar y la propia lógica dictan que una comunicación de este estilo debería ir encriptada, un estudio descubrió que una gran cantidad de fabricantes estaban sacando al mercado estos periféricos con comunicación en claro y sin autenticación. Esto permite que un atacante pueda hacerse pasar por el ratón y enviar órdenes al equipo víctima y tomar el control como si el atacante se hiciera con el propio equipo. Para mayor gravedad, la implementación habitual de los protocolos HID no distingue entre ratones y teclado, por lo que un atacante puede ser capaz de enviar órdenes de teclado a través del dongle de un ratón y básicamente ejecutar cualquier comando en el equipo de la víctima a través de la consola cmd. Esto podría permitir a un hacker llevar a cabo cualquier acción, como por ejemplo conectarse al equipo del atacante para poder sacar documentos confidenciales del equipo, o bien trasladar software malicioso a la víctima para poder instalarlo.

4.2.7. Fuzzing

[12] Los ataques de *fuzzing* se basan en el estudio del comportamiento de los dispositivos con el fin de conocer vulnerabilidades. Mediante el uso de paquete malformados y formatos de datos no reconocidos por el estándar, el atacante trata de provocar un mal funcionamiento de la víctima, con el fin de, en caso de alcanzar el éxito con su ataque, deducir posibles vulnerabilidades del sistema objetivo y sus protocolos. Algunos ataques basados en este concepto son BlueStab, HCIDump Crash, o Bluetooth Stack Smasher.

4.2.8. Malware

Se define como *malware* al software introducido en un terminal con el fin de manipular su comportamiento, robar información del usuario o extorsionarlo para recuperar sus datos, así como tratar de extenderse replicándose desde dispositivos infectados a nuevas víctimas (lo que se suele denominar como gusanos).

El primer gusano desarrollado para expandirse a través de Bluetooth fue **Caribe** [20], un *malware* desarrollado por el grupo de hackers españoles 29A para el sistema operativo Symbian. Caribe requiere que el usuario víctima acepte la

conexión Bluetooth y ejecute el fichero enviado por el atacante. En caso de que logre ser engañado para hacerlo, Caribe instala una aplicación /system/apps/caribe/caribe.app junto a una serie de ficheros complementarios que permiten convertir al terminal móvil en un nuevo atacante que escanee otras conexiones Bluetooth cercanas a las que tratar de infectar. Si bien su comportamiento afecta especialmente al consumo de la batería de los terminales infectados por su uso continuo del escaneo Bluetooth, Caribe no ejecutaba ninguna actividad más que pudiera generar mayor perjuicio, como robo de información o control del terminal. Un caso similar fue el de **CommWarrior**, un gusano similar que, si bien solía recurrir a los MMS como entrada hacia otros terminales, tenía también la capacidad de replicarse mediante el interfaz Bluetooth.

Es evidente que los ejemplos anteriores son ataques que requieren la colaboración de la víctima, previo engaño del atacante, para instalar el *malware* en cuestión. Esta es por tanto una situación que se puede corregir mediante la concienciación y la formación de los usuarios con respecto a las medidas de seguridad que se deben tener al manejar dispositivos. Sin embargo, existen *malwares* cuya ejecución se basa únicamente en defectos del sistema, lo que permite a los atacantes entrar en los terminales sin necesidad de que la víctima sea consciente ni participe de la operativa. Es el caso de **Blueborne**, sin duda el *malware* más extendido y de mayor gravedad de los últimos años. Dada su importancia le dedicaremos un punto aparte en el presente documento.

4.3. Contramedidas y buenas prácticas

Al igual que en el resto de redes, especialmente en el caso de las inalámbricas, tanto los usuarios que utilizan Bluetooth como las organizaciones que lo implementan en parte de sus sistemas y/o actividades, deben considerar seriamente la seguridad y adoptar aquellas medidas que ayuden a reforzarla.

Si bien es cierto que Bluetooth tiene con la proximidad una barrera grande frente a atacantes maliciosos, esto no siempre es suficiente, y ya hemos comprobado en el apartado 4.1 que Bluetooth tiene vulnerabilidades tanto a nivel de funcionamiento como a nivel estructural, por lo que será cuestión de usuarios y organizaciones mitigar los riesgos aplicando contramedidas contra vulnerabilidades específicas.

Sin embargo, ni siquiera de esta manera se garantiza un entorno Bluetooth seguro, ya que nunca se pueden asegurar otras penetraciones adversas. Además, la seguridad siempre tiene un precio, que va desde los gastos directos asociados a equipamiento adicional requerido hasta inconvenientes e incomodidades en la operación, mantenimiento, ralentización de procesos, etc. Por tanto, se trata de un complejo análisis de riesgos e inversión-coste necesario, que permitan tomar decisiones que optimicen el compromiso entre seguridad-coste para proteger los activos del usuario u organización.

En el FIPS 199, el estándar sobre seguridad en los sistemas de información publicado por la organización FIPS, se establecen tres objetivos principales para

la seguridad en los sistemas TI: **confidencialidad**, que preserve las restricciones de acceso a la información a aquellos usuarios autorizados; **integridad,** que proteja los datos de ser eliminados o alterados, manteniendo también la autoría de los mismos; y por último la **disponibilidad**, que asegure el acceso a los mismos. A continuación, se describen una lista de buenas prácticas o recomendaciones que permitan maximizar en la medida de lo posible estos tres objetivos en las redes Bluetooth.

4.3.1. Recomendaciones de Gestión de Riesgos

Estas recomendaciones están orientadas especialmente a organizaciones que utilicen Bluetooth como tecnología inalámbrica para conectar dispositivos que manejen información sensible, y que necesiten implementar una política de seguridad exhaustiva sobre el tema. Muchas de ellas podrían ser equivalentes a las de cualquier otra tecnología inalámbrica.

1. Desarrollar política de seguridad. Elaborar una política detallada de seguridad Bluetooth, que incluya a su vez todas las recomendaciones que van a exponerse a continuación, y que pueda ser consultada por los usuarios, así como actualizada frecuentemente. Esta política deberá incluir al menos:
    a. Lista de dispositivos Bluetooth autorizados por la organización.
    b. Uso que debe hacerse de Bluetooth y las medidas disciplinarias a aplicar en caso de que no se cumpla. Esto incluye una descripción detallada de a qué servicios, recursos o información se puede acceder a través de Bluetooth.
    c. Configuración de seguridad mínima de los dispositivos y medidas de protección a tomar, dependiendo del tipo de dispositivo del que se trate (por ejemplo, los portátiles tienen más posibilidades de ser robados o perdidos).
    d. Política de almacenamiento de información sensible en los dispositivos Bluetooth.

2. Concienciación de los usuarios. Es importante que los usuarios sean conscientes de los riesgos que corren y de sus responsabilidades con la seguridad. Además del repositorio anteriormente citado pueden promoverse actividades formativas, buzones de consulta, canales de comunicación en redes sociales corporativas, etc.
3. Evaluaciones de seguridad. Realizar un seguimiento con cierta periodicidad tanto del nivel de conocimiento de la misma por parte de los usuarios como de identificación de dispositivos candidatos.
4. Identificar la arquitectura de red de la organización y posicionar Bluetooth en ella. Dado que Bluetooth rara vez se encuentra presente en exclusividad, sino que trabaja generalmente en dispositivos que implementan varias tecnologías e interfaces de red para conexiones de área local y de área extensa, los expertos de la organización deben comprender muy bien la conectividad de los dispositivos para identificar las vulnerabilidades y riesgos de manera global y no solo aplicadas a Bluetooth.

5. Tener un inventario actualizado de dispositivos Bluetooth y sus direcciones (BD_ADDR). De esta manera puede tenerse controlada la lista de usuarios que tienen acceso a según qué información y qué uso de ella, lo que permite identificar fácilmente dispositivos no autorizados con un proceso de auditoría.
6. Designar a un experto en seguridad Bluetooth. Su labor será rastrear el progreso de los productos y están dares de seguridad de Bluetooth (a través del Bluetooth SIG) y las amenazas y vulnerabilidades con la tecnología, para adaptar la tecnología de la organización a las últimas novedades que permitan minimizar los riesgos del uso de Bluetooth.

4.3.2. Recomendaciones Técnicas y Operativas

Estas recomendaciones aplican al funcionamiento en sí de Bluetooth, por lo que no son tan generalistas como las de gestión, sino que aplican a vulnerabilidades concretas de la tecnología y su operación. Posteriormente se relacionarán con las vulnerabilidades de apartado 4.1 a las que aplican.

- **Protección de Equipos**

7. Medidas para proteger el robo de los dispositivos de mano: Dado que los dispositivos Bluetooth suelen ser de pequeño tamaño y portátiles, el robo físico del dispositivo puede suponer en sí mismo la pérdida de la información que contienen, por lo que dar consejos y recomendaciones para evitar que esos dispositivos sean sustraídos también irá en pos de la seguridad de Bluetooth.
8. Proteger con una clave de acceso los dispositivos portátiles con interfaces Bluetooth: Dado que se trata de dispositivos que por su naturaleza pueden ser fácilmente robados o perdidos, protegerlos con una contraseña puede evitar el robo de información o la suplantación en comunicaciones Bluetooth.

- **Configuración de Seguridad**

9. Adaptar la configuración predeterminada: Dado que generalmente la configuración por defecto del dispositivo Bluetooth no es segura, es necesario adaptar esta configuración a lo que la organización haya dispuesto en su política de seguridad mencionada anteriormente. Por ejemplo, es una buena práctica **cambiar el nombre del dispositivo que viene por defecto**, ya que suele ser demasiado descriptivo (por ejemplo, “iPhone de Eduardo”), y aportar datos de la marca, modelo, propietario u organización del dispositivo puede facilitar ataques dirigidos.
10. Reducir nivel de potencia: Reduciendo la potencia máxima de los dispositivos a aquella potencia mínima que permita la conexión dentro del perímetro de la organización estamos evitando que posibles atacantes que se encuentren fuera de él se conecten con los dispositivos interiores. Se debe evitar, por tanto, el uso de dispositivos de Clase 1 que son los que tienen mayor nivel de potencia de salida, así como amplificadores externos o antenas de alta ganancia debido a su rango extendido.
11. Desactivar perfiles innecesarios: Bluetooth está diseñado para soportar multitud de perfiles, pero estos deben ajustarse a la mínima funcionalidad

necesaria, de manera que aquellos que no sean imprescindibles deberían estar desactivados, si el dispositivo lo permite.

12. Configurar el dispositivo por defecto en modo no visible (*undiscoverable*): De esta manera se dificulta que el dispositivo sea detectado cuando no hay una voluntariedad expresa de emparejamiento por parte del usuario. Aunque sí es posible detectar dispositivos no visibles mediante el uso de un sniffer o técnicas de fuerza bruta con la dirección del dispositivo (BD_ADDR), en modo no visible se complica considerablemente las posibilidades del atacante.

13. Deshabilitar Bluetooth cuando no esté en uso: Las capacidades Bluetooth del dispositivo deben permanecer deshabilitadas excepto cuando vaya a hacerse uso de ellas. En el caso de dispositivos que no permitan deshabilitarla, como puede ser el caso de auriculares inalámbricos, estos deberán permanecer apagados mientras no estén en uso. De esta forma se minimiza la exposición a posibles ataques, aunque también resulta una incomodidad para los usuarios, inviable en algunos casos donde debe existir comunicación constante, por ejemplo, con una pulsera de actividad o un *smartwatch*.

14. Mantener actualizado el dispositivo Bluetooth: Constantemente se descubren y resuelven nuevas disponibilidades en los dispositivos Bluetooth, que se incluyen en el repositorio de CVE, por lo que es necesario mantener el dispositivo al día en cuanto a parches que el fabricante implemente para evitar *exploits* maliciosos.

15. Instalar antivirus en los dispositivos que lo permitan: Proteger con un antivirus los dispositivos que lo permitan, como por ejemplo ordenadores, supondrá también una protección frente a *malware* desconocido en redes y dispositivos Bluetooth.

- **Protección de las Comunicaciones**

16. Elegir código PIN aleatorio, largo y privado: En muchos casos se eligen PIN estáticos muy sencillos, como todo ceros, o con algún significado que los hace sencillos de adivinar. Para dispositivos Bluetooth con versión 2.0 o anteriores se debe usar un PIN alfanumérico de ocho caracteres, de ser posible.
17. Evitar claves unitarias para generar la clave de link: El uso de claves unitarias, generadas por único dispositivo y compartidas por el canal, puede dar lugar a ataques de suplantación y MITM. De hecho, quedó obsoleto a partir de la versión 1.2, usándose en su lugar claves combinadas.
18. Evitar el uso de *Just Works* en SSP: A partir de la versión 2.1, aunque se utiliza SSP como método de emparejamiento avanzado, si se utiliza en modelo *Just Works*, donde no hay validación por parte de los usuarios, no se proporciona protección frente a ataques MITM. Por tanto, los dispositivos que solo admiten este modelo, por no tener capacidades de entrada/salida, no deben adquirirse si hay disponibles dispositivos

similares con capacidad para implementar OOB, *Numeric Comparison* o *Passkey Entry*.

19. Uso de claves aleatorias y únicas en modelo *passkey entry*: Los dispositivos de versión 2.1 o posterior, que ya admiten el uso de SSP, deben utilizar claves aleatorias en el uso del modelo *passkey entry*, que sean únicas y se renueven con cada intento de emparejamiento. Si se utiliza una clave estática para múltiples emparejamientos, la protección MITM proporcionada por el modelo de asociación passkey entry se reduce.
20. Emplear los modos de seguridad más robustos: Es recomendable utilizar las implementaciones de Bluetooth con modos de seguridad más robustos:
    a. En BR/EDR la recomendación es utilizar *Secure Connections* con Modo 4 y Nivel 4, que proporciona una clave de enlace autenticada y cifrado AES-CCM, lo que supone utilizar dispositivos con versión 4.1 o posteriores.
    b. En LE se recomienda el Modo 1 Nivel 4, que proporciona *pairing* con autenticación LE *Secure Connections* y cifrado AES-CCM, lo que supone utilizar dispositivos con versión 4.2 o posterior.
    c. En cualquier caso, se desaconseja encarecidamente utilizar dispositivos con versiones anteriores a la 2.1.
21. Utilizar algoritmos acreditados por los organismos pertinentes: Algunos algoritmos utilizados en Bluetooth, como el E0, con el avance de la tecnología y la capacidad computacional resultan débiles para cifrar las comunicaciones. Por tanto, debe optarse por algoritmos aceptados por el FIPS, organismo de EEUU que recoge estos estándares, o por el Centro Criptológico Nacional (CCN), en el caso de España. En la siguiente tabla se recoge un resumen de los algoritmos utilizados por Bluetooth:

	Bluetooth BR/EDR			**Bluetooth LE**	
	v2.1 y anteriores	v2.1 a 4.0	v4.1 y superiores	v4.0 y 4.1	v4.2 y posteriores
Pairing	E21/E22 (SAFER+)	P-192 ECDH HMAC-SHA-256	P-256 ECDH HMAC-SHA-256	AES-128	P-256 ECDH AES-CMAC
Cifrado	E0 (Basado en SAFER+)		AES-CCM	AES-CCM	AES-CCM
Integridad	-		AES-CCM	AES-CCM	AES-CCM

Tabla 11: Resumen algoritmos por tecnología [5]

22. Cifrar los datos transmitidos en todas las comunicaciones: Si se utilizan comunicaciones no cifradas se corre el riesgo de sufrir ataques de *eavesdropping*. Es importante asegurar el cifrado en comunicaciones inalámbricas multisalto, ya que un salto no cifrado en la cadena compromete la cadena completa.
23. Asegurar autenticación mutua: No se recomienda permitir la autenticación unidireccional. Todas las tecnologías Bluetooth proporcionan mecanismos de autenticación mutua de dispositivos.

24. Configurar los tamaños de clave de cifrado al máximo: Incrementar el tamaño de la clave de cifrado protege exponencialmente frente a los ataques de fuerza bruta. La longitud de la clave de cifrado es un parámetro que se acuerda entre los dispositivos Bluetooth en la fase de negociación del *pairing* y puede tener una longitud entre 7 y 16 octetos. Se recomienda que la clave tenga como mínimo 14 octetos (112 bits).
25. Solicitar autorización para conectar: El dispositivo Bluetooth debe solicitar al usuario que autorice todas las solicitudes de conexión entrantes antes de permitir que el dispositivo se empareje con otro. Los usuarios no deberían aceptar conexiones o archivos de fuentes desconocidas o no fiables.
26. Utilizar autenticación y cifrado a nivel de aplicación: Dado que Bluetooth como tal no tiene seguridad a nivel de aplicación, se recomienda el uso de aplicaciones que incorporen vía software una capa adicional de autenticación y cifrado sobre la pila de protocolos Bluetooth. Esto haría posible, por ejemplo, que se evitaran emparejamientos automáticos con dispositivos almacenados en la memoria de emparejamientos previos, que se incluyeran otros mecanismos de autenticación más sofisticados, como pruebas biométricas, o que se incluyera una capa de cifrado adicional sobre el cifrado nativo del protocolo para proteger aún más los datos en tránsito.
27. Utilización de Privacidad LE: Dado que existen escenarios donde se puede hacer seguimiento de un dispositivo y conocer su dirección BD_ADDR, lo cual puede comprometer sus comunicaciones, LE incorpora un algoritmo de resolución de identidad (IRK) que asegura la privacidad de la dirección del dispositivo y por ello la seguridad en sus comunicaciones

- **Concienciación**

28. Minimizar los emparejamientos: Es recomendable emparejar el dispositivo con el menor número de dispositivos posible, y siempre en un entorno seguro, preferentemente no público e interior, donde sea complicado para un atacante visualizar las claves introducidas o interceptar los mensajes de emparejamiento. Los usuarios no deben responder a ninguna solicitud de PIN si no están seguros de que la solicitud viene de un dispositivo conocido y confiable.
29. Borrar dispositivo perdido de otros dispositivos previamente emparejados: En caso de que un dispositivo se pierda, se recomienda al usuario borrarlo (u olvidarlo) de todos los demás dispositivos de que disponga, de manera que un atacante que lo encuentre no pueda obtener información de ellos, suplantando su identidad.

4.4. Resumen Recomendaciones y Vulnerabilidades

Se recogen en la siguiente tabla en formato resumen las recomendaciones descritas en el apartado anterior, con las vulnerabilidades identificadas en el apartado 4.1 a las que hacen referencia. Algunas de ellas tienen carácter general y no aplican sobre ninguna vulnerabilidad en concreto.

Id	Recomendación	Vulnerabilidades
Gestión Riesgos		
1	Desarrollar política de seguridad:	
2	Concienciación de los usuarios	
3	Evaluaciones de Seguridad	
4	Identificar la arquitectura de red de la organización, y posicionar Bluetooth en ella	
5	Tener un inventario actualizado de dispositivos Bluetooth y sus direcciones (BD_ADDR)	
6	Designar a un experto en Seguridad Bluetooth:	
Protección de Equipos		
7	Medidas para proteger el robo de los dispositivos de mano	
8	Proteger con una clave de acceso los dispositivos portátiles con interfaces Bluetooth	15. Las claves de enlace se pueden almacenar de manera poco segura
Configuración de Seguridad		
9	Adaptar la configuración predeterminada	
10	Reducir nivel de potencia	
11	Desactivar perfiles innecesarios	
12	Configurar el dispositivo por defecto en modo no visible (*undiscoverable*)	21. La visibilidad activada por defecto hace vulnerables los dispositivos
13	Deshabilitar Bluetooth cuando no esté en uso	
14	Mantenga actualizado el dispositivo Bluetooth	
15	Instalar antivirus en los dispositivos que lo permitan	
Protección Comunicaciones		
16	Elegir código PIN aleatorio, largo y privado	3. PINS cortos y fáciles de adivinar
17	Evitar claves unitarias para generar la clave de link	2. Claves de link basadas en claves unitarias son estáticas y se reautilizan
18	Evitar el uso de *Just Works* en SSP	5. Modelo Just Works no proporciona protección MITM
19	Uso de claves aleatorias y únicas en modelo *passkey entry*	7. El uso de passkeys estáticas facilita ataques MITM
20	Emplear los modos de seguridad más robustos	1. El modo de seguridad 1 carece de mecanismos de Seguridad. 4. Las claves de cifrado se repiten cada 23,3h 8. Retrocompatibilidad en modos de seguridad 10. Clave maestra broadcast se comparte entre los dispositivos de la piconet
21	Utilizar algoritmos acreditados por los organismos pertinentes	6. Las claves ECDH en SSP pueden ser débiles 11. El algoritmo de cifrado E0 es relativamente débil 16. Generadores de números aleatorios (RNG) débiles
22	Cifrar los datos transmitidos en todas las comunicaciones	14. Los métodos de emparejamiento Legacy no protegen frente a ataques de escucha.
23	Asegurar autenticación mutua	13. La autenticación simple es vulnerable a ataques MITM
24	Configurar los tamaños de clave de cifrado al máximo	17. La longitud de la clave de cifrado es negociable
25	Solicitar autorización para conectar	
26	Utilizar autenticación y cifrado a nivel de aplicación	18. No existe autenticación a nivel de usuario 19. No existe seguridad extremo a extremo 20. Los servicios de seguridad son limitados
27	Utilización de Privacidad LE	12. Privacidad BR/EDR puede verse comprometida si se asocia la BD_ADDR al dispositivo
Concienciación		
28	Minimizar los emparejamientos	22. Los dispositivos emparejados la autenticación mutua no siempre funciona en modos 3 y 4
29	Borrar dispositivo perdido de otros dispositivos previamente emparejados	

Tabla 12: Resumen recomendaciones y relación con vulnerabilidades

5. Casos de relevancia

En los últimos años se ha descubierto dos tipos de ataques de especial gravedad para los dispositivos Bluetooth. Dada su importancia, les dedicamos un apartado particular a cada uno de ellos para revisarlos en detalle.

5.1. BlueBorne

[22] BlueBorne es el nombre con el que fue bautizado un nuevo vector de ataque descubierto en 2017 por la empresa de seguridad Armis, relacionado con una serie de vulnerabilidades identificadas en los principales sistemas operativos del mercado.

5.1.1. Descripción e impacto

A la hora de hablar de BlueBorne no hablamos de un ataque individual, sino de un grupo de 8 vulnerabilidades descubiertas que afectan a los sistemas operativos más extendidos entre los usuarios, esto es, Android, iOS, Linux y Windows, lo que lo convierte en un problema de seguridad crítico, afectando a millones de dispositivos en todo el mundo. La base de estas vulnerabilidades reside en el modo que tienen dichos fabricantes de implementar el protocolo Bluetooth dentro de sus sistemas. A continuación pasamos a describir cada una de estas vulnerabilidades clasificándolas por sistema operativo.

5.1.1.1. Linux

Fuga de información (CVE-2017-1000250)

Como se ha mencionado en el apartado 2, el protocolo SDP tiene la función de permitir entre dispositivos la consulta de información sobre los servicios que soportan. Este protocolo permite el uso de fragmentación de paquetes para los casos en los que la información a enviar del servidor al cliente que la ha solicitado

sea mayor que la MTU permitida. En estos casos, el servidor responde a la SDP_request con una SDP_response que consta entre toda su información del llamado *continuation state*.

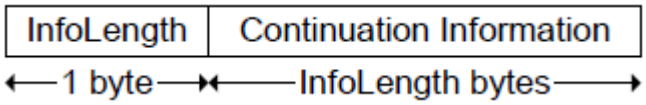

Ilustración 31: Formato del parámetro continuation state [22]

Este parámetro constará de un primer byte para indicar su longitud (ya que no tiene un tamaño determinado) seguido de un dato denominado *continuation information*. Una vez el cliente reciba esta respuesta deberá enviar una nueva SDP_request incluyendo el parámetro *continuation state* recibido previamente para que el servidor pueda saber a partir del dato *continuation information* por dónde debe continuar.

La implementación de esta lógica en Linux se basa en indicar en este parámetro la dirección del buffer que el servidor debe seguir leyendo para responder a la consulta del cliente, y es debido a una mala implementación de los controles del parámetro *continuation state* recibido que un atacante en el rol de cliente (que puede manipular el dato a su antojo) puede forzar al dispositivo al que consulta a leer información sensible del sistema. Estos datos podrían ser utilizados para elaborar de manera más eficiente ataques más precisos aprovechando otras de las vulnerabilidades que veremos.

Ejecución de código malicioso (CVE-2017-1000251)

Según hemos visto, L2CAP es el protocolo de nivel más bajo en el lado de *host* en un sistema Bluetooth. A la hora de establecer una conexión, dos terminales inician el llamado proceso de configuración mutua, que consiste en iniciar un diálogo basado en mensajes L2CAP_ConfigReq y L2CAP_ConfigRsp a partir de los cuales los dispositivos proponen y aceptan o renegocian los parámetros pertinentes para levantar la conexión.

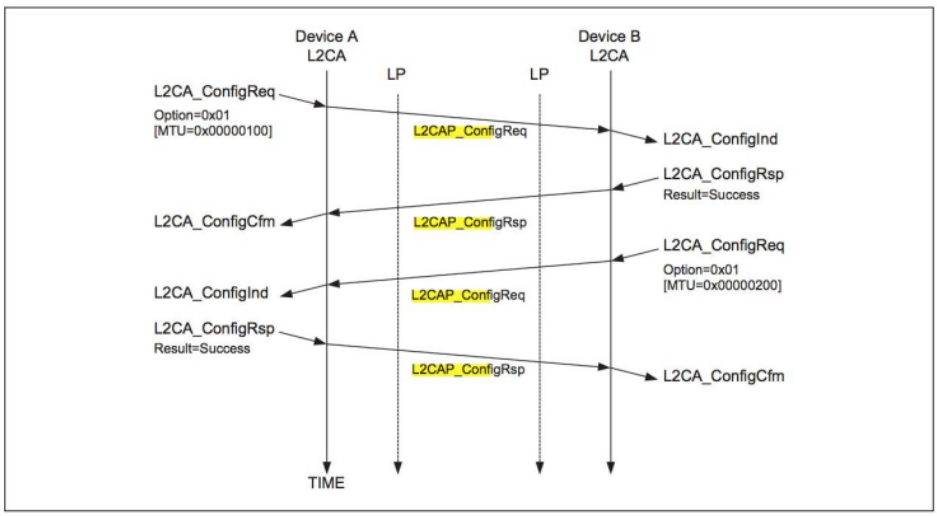

Ilustración 32: Configuración mutua L2CAP [22]

L2CAP cuenta con una opción complementaria denominada *Extended Flow Specification*, la cual permite establecer un formato más preciso de parámetros para garantizar cierta calidad de servicio global. Este modelo implica compartir la colección de parámetros completa, que debe ser revisada por los terminales y mantenida en estado "pendiente" hasta ser dada por buena y aprobar todo el formato en conjunto.

La implementación de esta lógica en el *stack* de Linux BlueZ contiene una falla a nivel de control de parámetros que genera la posibilidad de llevar a cabo un ataque de *buffer overflow* a partir del cual ejecutar código malicioso en el dispositivo atacado.

5.1.1.2. Android

Fuga de información (CVE-2017-0785)

La implementación del protocolo SDP y su política de fragmentación de paquetes está afectada por la misma vulnerabilidad que hemos visto para Linux en la CVE-2017-1000250

Ejecución de código malicioso (CVE-2017-0718/0782)

Como se ha visto en el apartado 3, dentro de las maneras de llevar a cabo la autenticación fase 1 durante un proceso de emparejamiento, en el caso de no contar con las capacidades I/O adecuadas, los dispositivos se ven obligados a seleccionar Just Works, un modelo que implica que el usuario no puede comprobar el PIN compartido y tan solo pueda confirmar la conexión sin la certeza de que el emparejamiento es limpio. La implementación de la autenticación Just Works en el sistema Android posee una debilidad que deriva en que un terminal pueda autenticarse con otro mientras cumpla ciertos requisitos, esto es, pretender establecer una conexión temporal y aceptar solo Just Works, sin que la víctima tenga siquiera que aceptar la conexión. Esto permite a un atacante autenticarse y poder acceder a ciertos permisos sobre servicios Bluetooth en los que, si se diera el caso, explotar sus vulnerabildiades. Este es el caso del protocolo BNEP.

BNEP (*Bluetooth Network Encapsulation Protocol*) es uno de los servicios que puede ofrecer Bluetooth, y cuya función es permitir encapsular el tráfico Ethernet para poder transportarlo sobre L2CAP cuando se está utilizando un dispositivo en modo *tethering* sirviendo a otros terminales como punto WiFi. Este protocolo se basa a efectos prácticos en traducir la cabecera Ethernet del tráfico en una nueva cabecera BNEP adaptada.

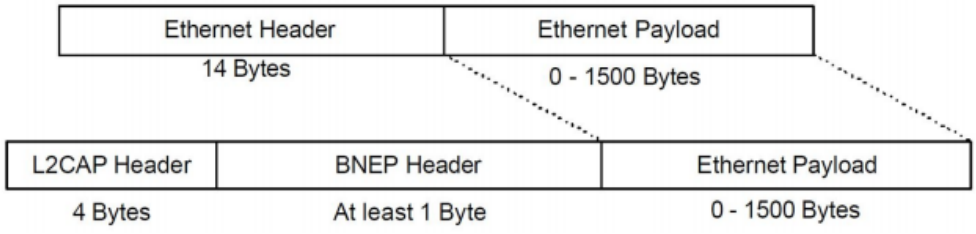

Ilustración 33: Encapsulación BNEP [22]

Además, BNEP permite la transmisión de mensajes de control, los cuales se requieren para establecer una conexión PAN (*Private Access Network*), que es básicamente el nivel de red por encima de BNEP en el *stack* Bluetooth.

La implementación de la lógica de control de estos mensajes de control BNEP en Android poseen dos vulnerabilidades a la hora de validar parámetros que dan lugar a la posibilidad de que un atacante explote ataques de desbordamiento de pila mediante dos formatos de estos mensajes de control. La conjunción de estas vulnerabilidades con la referente a la implementación de SDP vista anteriormente puede permitir a un atacante ejecutar código malicioso en el terminal y tomar control absoluto del sistema.

Bluetooth Pineapple MITM (CVE-2017-0783)

La misma vulnerabilidad vista en el punto anterior que permite autenticarse temporalmente en modo Just Works da la posibilidad a un atacante a explotar otro tipo de vulnerabilidad relacionada con el perfil PAN (que funciona sobre BNEP) para llevar a cabo un ataque Man-In-The-Middle sobre un sistema Android.

Esta vulnerabilidad se basa en que, aprovechando la condición de dispositivo autenticado, un dispositivo atacante pueda establecer una conexión exitosa con el servicio BNEP e iniciar el perfil PAN (*Private Access Network*).

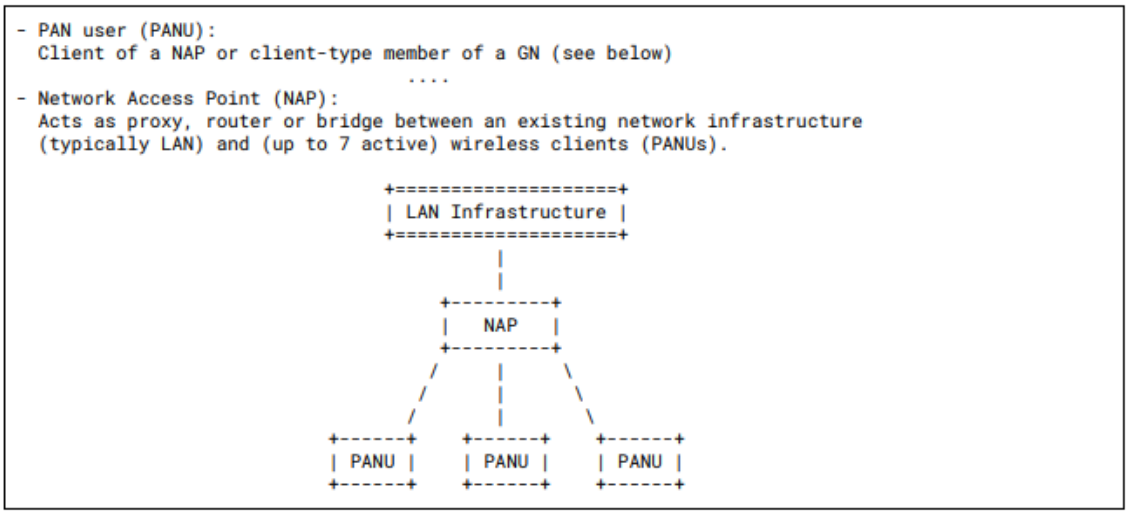

Ilustración 34: Jerarquía PAN [22]

Mediante el uso de este perfil el atacante es capaz de iniciar una sesión de *tethering* en la que se establezca como el NAP (*Network Access Point*) del dispositivo atacado. Así, el atacante pasa a convertirse en la puerta de entrada y salida del tráfico de internet que procesa la víctima, permitiéndolo acceso total a la información que intercambia e incluso, contando con las herramientas adecuadas, modificar la información que recibe, ya sea de manera dinámica sobre el tráfico entrante o definiendo para la víctima una dirección de DNS fraudulenta que permita atacar mediante DNS *spoofing*. Este ataque es muy similar al conocido WiFi Pineapple.

5.1.1.3. Windows

Bluetooth Pineapple MITM (CVE-2017-8628)

Como se ha visto en Android, la implementación de Bluetooth en Windows permite la autenticación en modo *Just Works* de un atacante sin necesidad de confirmación de la víctima. Esto deriva en que tenga la capacidad de explotar la misma vulnerabilidad que hemos visto en el punto anterior relacionada con el perfil PAN utilizado en la funcionalidad de *tethering* para ejecutar un ataque Man-In-The-Middle, pudiendo tanto espiar como manipular el tráfico del dispositivo objetivo.

5.1.1.4. iOS

Ejecución de código malicioso (CVE-2017-14315)

A pesar de aportar todo un repertorio de servicios y protocolos estándar, algunos desarrollados prefieren implementar sus protocolos propios a fin de optimizarlos para sus sistemas. Es el caso de Apple, que a la hora de transmitir audio entre sus dispositivos recurre al protocolo LEAP (*Low Energy Audio Protocol*), diseñado ad hoc por la compañía. Este protocolo ha resultado poseer una vulnerabilidad que deriva en la posibilidad de ejecutar un ataque de desbordamiento de pila que puede permitir ejecutar código malicioso dentro del terminal objetivo.

5.1.2. Resumen

Amenaza	Origen vulnerabilidad	Código	Sistema afectado
Acceso no autorizado a datos	SDP	CVE-2017-0785	Android
		CVE-2017-1000250	Linux
MITM	PAN profile	CVE-2017-0783	Android
		CVE-2017-8628	Windows
Malware	BNEP	CVE-2017-0781	Android
		CVE-2017-0782	Android
	L2CAP EFS	CVE-2017-1000251	Linux
	LEAP	CVE-2017-14315	iOS

Tabla 13: Resumen vulnerabilidades BlueBorne

5.1.3. Contramedidas

Queda patente que la especificación Bluetooth es una especificación complicada. Véase que mientras que la de la tecnología WiFi ocupa del orden de 400 páginas, la que nos ocupa tiene una extensión de 3000, aumentando la complejidad a la hora de comprenderla y ejecutarla correctamente. Además, la alargada pila de protocolos de la que consta multiplica el número de niveles de

comunicación en los que poder encontrar vulnerabilidades que puedan haber pasado por alto para los fabricantes de dispositivos.

Como se ha podido observar a lo largo del repaso de todas las vulnerabilidades implicadas en el vector BlueBorne, el componente fundamental para abrir la puerta a cualquiera de estos ataques es el error de diseño a nivel de código en la solución. Hablamos prácticamente en su totalidad de ataques de muy bajo nivel de corrupción de memoria, tan difíciles de detectar como graves de sufrir. De este modo, no queda más remedio de cara a los fabricantes que dedicar una atención especial a su depuración, así como mantener una política de actualización de software muy estricta de cara a los usuarios. El punto positivo de las vulnerabilidades que hemos visto en este apartado es que son del tipo de problemas fáciles de corregir con un parche de rápida publicación por parte de los fabricantes, y es nuestro deber como usuarios estar al tanto de cualquier actualización de cara a estar protegidos.

Por otro lado, es de vital importancia que a la hora de implementar la especificación Bluetooth por parte de los fabricantes se ponga especial atención a limitar al máximo los permisos para realizar operaciones y las facilidades para emparejarse con otros dispositivos. Llama la atención lo sencillo que sería neutralizar las vulnerabilidades vistas en el protocolo BNEP y en el perfil PAN de llevar a cabo una política más restrictiva y adecuada con respecto a los requisitos para poder autenticar en Just Works y a los permisos que este emparejamiento puede proporcionar, sabiendo que es el más inseguro. Buen ejemplo de ello es el caso de iOS, que a diferencia de Android y Windows impide cualquier tipo de autenticación sin la intervención del usuario, y por tanto queda a salvo de las vulnerabilidades de protocolos superiores para los que se requiere cierto nivel de permisos. Es obvio que, si bien se debe esperar que p.ej. BNEP se implemente correctamente, proporcionar a los atacantes la posibilidad de llegar hasta él es del todo absurda.

5.2. KNOB

[23] El ataque *Key Negociation Of Bluetooth* (KNOB) fue publicado en el año 2019 mediante la vulnerabilidad CVE-2019-9506 y ha sido uno de los ataques de mayor alcance de los últimos tiempos, afectando a todos los fabricantes con los que se ha experimentado hasta la fecha.

5.2.1. Descripción

KNOB se basa en explotar una vulnerabilidad de Bluetooth relacionada con la política de negociación de la clave de encriptación de las comunicaciones que determina la especificación, de ahí su gran alcance. Como hemos visto en el punto 3.2.1, el proceso de generación de la clave de encriptación K_C permite por cuestiones de adaptabilidad a los distintos entornos definir como parámetro a nivel de banda base la máxima longitud para la clave de encriptación que podrá soportar, dentro de los límites de 1 y 16 bytes. Así, al iniciar una conexión encriptada, los dispositivos involucrados deberán previamente negociar la longitud de su clave en base a estos parámetros, comunicando su máxima

longitud aceptada y esperando la aceptación del otro dispositivo o su nueva propuesta de longitud en caso de que esta no satisfaga su configuración.

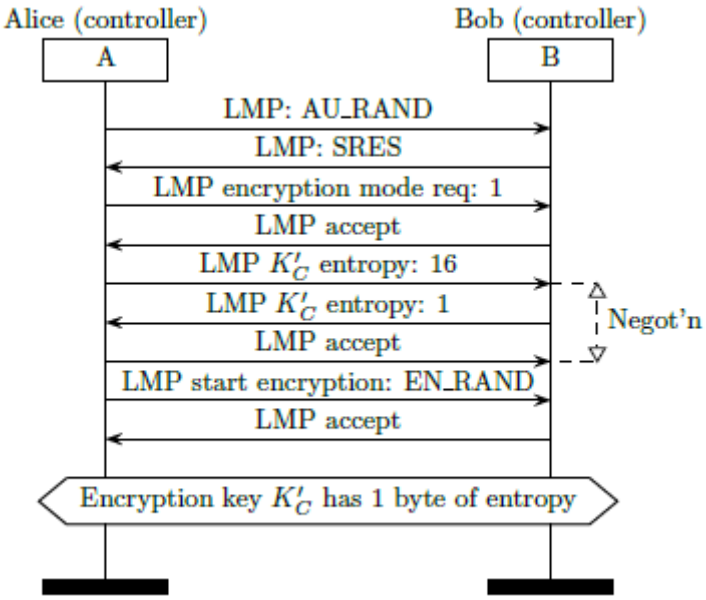

Ilustración 35: Proceso establecimiento de encriptación [23]

La base del ataque KNOB consiste en interferir en las comunicaciones entre los dos dispositivos durante el intercambio de mensajes involucrados en la negociación de la longitud de la clave, transmitidos sobre el protocolo LMP. Aunque previamente los dispositivos han realizado la autenticación y el establecimiento de la clave de link, el proceso de negociación se realiza sin encriptación (pues no se cuenta aún con clave) y sin control de integridad de mensajes. Esto permite que, tras permitir que los dispositivos A y B se autentiquen satisfactoriamente, un atacante interfiera en el momento de la negociación de longitud de clave, denotada en la figura previa como “negot’n”. En la siguiente figura podemos ver el cronograma concreto.

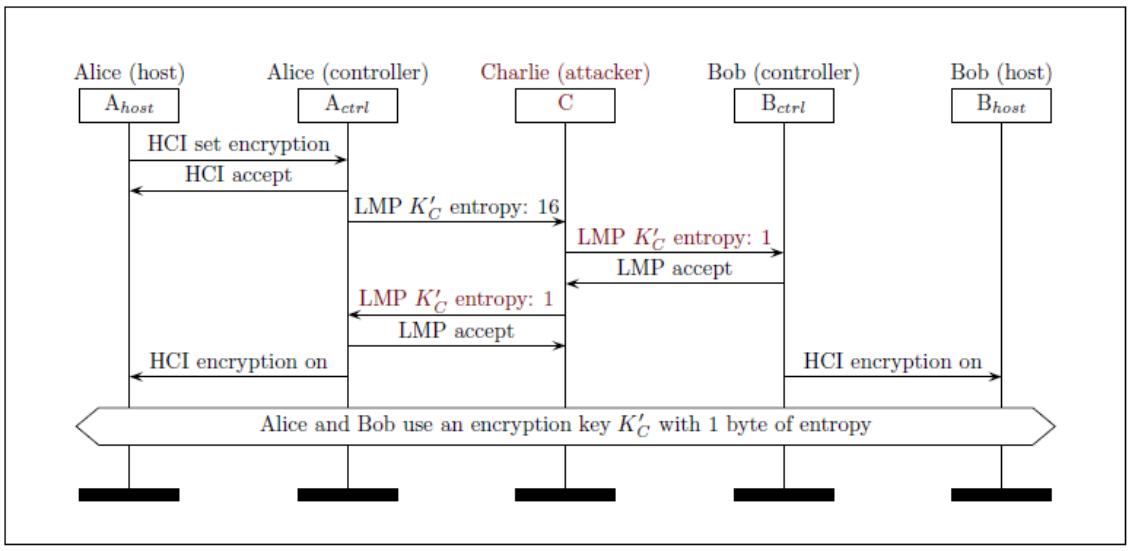

Ilustración 36: Ataque MITM KNOB [23]

Tras establecer la necesidad de encriptar los datos a partir de un mensaje LMP_encryption_mode_req=1, el atacante interfiere en las comunicaciones y suplanta al contrario de cara a ambos dispositivos. El atacante intercepta el mensaje LMP_encryption_key_size_req con valor 16 bytes (el máximo posible del estándar) enviado por el dispositivo A y lo modifica para hacer llegar al dispositivo B una solicitud de 1 byte únicamente. Este, considerándolo la propuesta inicial del dispositivo A y estando el valor dentro de lo aceptado como L_{min}, transmite un LMP_accept. El atacante entonces se encarga de modificar esa transmisión de vuelta, sustituyéndola de nuevo por un LMP_encryption_key_size_req con valor 1, ya que para el dispositivo A la longitud negociada hasta ese momento era de 16 bytes. Tras aceptar la restricción, el atacante corta el envío del LMP_accept, ya que el dispositivo B ha dado por finalizado el proceso de negociación y no lo espera. De este modo, ambos cierran la negociación de manera satisfactoria con el mínimo nivel de encriptación permitido.

5.2.2. Impacto

La problemática de esta vulnerabilidad se debe a que reside en la propia especificación Bluetooth en el lado del *controller*, a nivel de link. Esto implica que, perteneciendo a un nivel tan bajo de la especificación, los fabricantes acatan lo que establece el SIG, por lo que una vulnerabilidad a este nivel afectaría teóricamente a todos los dispositivos Bluetooth del mercado. Además, los desarrolladores de aplicaciones Bluetooth no tienen el proceso de encriptación en cuenta, ya que se confía en la fiabilidad del estándar, y por tanto dentro de los procesos realizados por el *host* no existe ninguna comprobación de las condiciones en las que se está ejecutando.

Dados los datos expuestos, podríamos clasificar a KNOB como un ataque de tipo *sniffing* que sin embargo requiere de una compontente MITM para poder ejecutarse. El estudio que sacó a la luz este tipo de ataque ha demostrado que, debido a estas circunstancias, es efectivo para dispositivos basados en todos los chips conocidos, y que es posible desencriptar en tiempo real las comunicaciones cifradas con claves de únicamente 1 byte mediante equipamiento genérico (p.e. un portátil de gama básica). Esto supone una grave vulnerabilidad que puede explotarse para cualquier tipo de aplicación Bluetooth utilizada, pero que se aprecia especialmente grave para algunos tipos de perfiles básicos:

- **HID**: cualquier dispositivo que utiliza un perfil HID (*Human Interface Device*), tales como ratones o teclados, estarían expuestos a ver capturado su tráfico con el dispositivo maestro y extraído en claro. Esto podría permitir a un atacante capturar información valiosa introducida a través de teclados, tales como direcciones de correo, contraseñas o datos bancarios.
- ***Tethering***: el uso de dispositivos móviles como puntos de conexión a internet auxiliares a modo de *hotspots* utiliza la tecnología Bluetooth para la transferencia del tráfico entre el dispositivo final y el dispositivo operando como *hotspot*, encapsulado mediante el protocolo BNEP. Esto

significa que la intercepción y desencriptación de dicho tráfico permitiría al atacante realizar *sniffing* de los datos de navegación de la víctima.

- **A2DP**: gracias a un ataque KNOB un atacante podría recurrir al perfil estándar para la transmisión de audio sobre *Bluetooth, Advanced Audio Distribution Profile* (A2DP), para poder capturar conversaciones que la víctima esté recibiendo en su terminal móvil y atendiendo a través de auriculares o dispositivos manos-libres. Esta casuística se ve especialmente agravada por la tendencia actual a la desaparición de los conectores tipo Jack, que ha provocado una penetración cada vez mayor de los auriculares Bluetooth.

5.2.3. Contramedidas

La base de la vulnerabilidad que desencadena un ataque KNOB reside en la posibilidad de seleccionar la longitud de la clave de encriptación. Si bien en la especificación se alude a la adaptabilidad de las políticas de encriptación a distintos entornos, la realidad es que actualmente prácticamente todas las implementaciones de Bluetooth tratan de establecer una longitud de clave de 16 bytes. Teniendo en cuenta además que a la hora de generar el flujo de bits encriptado se normaliza la clave elegida hasta alcanzar los 16 bytes, y que por lo tanto no hay diferencias en cuanto a nivel de consumo o de performance, no parece sensato mantener en el estándar unos parámetros de encriptación que puedan llegar a ser inocuos. Se proponen por tanto dos tipos de contramedidas para solventar la vulnerabilidad:

- ***Legacy complaint***: a fin de evitar que se pueda producir un ataque de estas características, una posible solución sería modificar a nivel de *controller* el valor de L_{min} establecido. Sin embargo, esto requiere una actualización sobre el firmware, por lo que otra posible solución sería introducir un procedimiento a nivel de *host* que revise la longitud de clave que se ha seleccionado para la encriptación, posible mediante el comando Read Encryption Key Size del interfaz HCI, y que en caso de no cumplir con un valor que garantice la fortaleza de la encriptación, aborte la conexión. Por otro lado, esta segunda opción sería negativa en términos de eficiencia, ya que actuaría sobre una conexión ya establecida en lugar de durante el proceso de establecimiento de esta. Una tercera opción sería que los desarrolladores implementen encriptación a nivel de aplicación como medida de seguridad extra.

- ***No legacy complaint***: en términos de solucionar la vulnerabilidad a largo plazo, la propuesta más razonable es modificar la especificación en una futura versión, modificando los valores de rango de longitud de clave de encriptación L_{min} y fijándolos al máximo de 16 o, preferiblemente, directamente eliminar del estándar el proceso de negociación de longitud de clave, utilizando por defecto el valor máximo.

6. Escenarios prácticos

A fin de aportar una visión más precisa de las implicaciones de seguridad derivadas del uso de comunicaciones Bluetooth, a continuación se describen una serie de escenarios distintos, tanto por uso como por riesgos, en los que se utilizan dispositivos con esta tecnología, las amenazas a las que están expuestos y las medidas que se deberían afrontar con el fin de alcanzar el mayor nivel de protección posible.

6.1. Escenario 1: empresa A

6.1.1. Descripción

Supongamos a la empresa A, una empresa de mensajería a nivel nacional. Para agilizar la operativa de sus trabajadores durante las labores de reparto, esta compañía los equipa con impresoras portátiles de etiquetado y escáneres de códigos de barras para el control de pedidos e inventarios, ambos con conexión Bluetooth a sus terminales móviles corporativos, encargados de la comunicación por red móvil a los sistemas centrales.

Por otro lado, los centros de logística de la empresa A cuentan con un gran número de dispositivos de domótica industrial para el control de diversas métricas y automatizaciones, tales como luces y temperatura, además de detecciones para controlar las cintas transportadoras y puertas de entrada para los camiones de reparto.

6.1.2. Riesgos

El uso de comunicación Bluetooth para fines corporativos tiene como consecuencia un aumento considerable del alcance que puede tener un ataque al ser ejecutado contra los empleados de una compañía. En nuestro caso, si bien las posibilidades de sufrirlo por parte de un repartidor se reducen por el hecho de encontrarse en constante movimiento, existe cierto riesgo en caso de ataque dirigido, además de los momentos de estacionamiento largos y de parada en los almacenes. Por otro lado, para los dispositivos de domótica industrial ubicados en los centros de logística contamos con el escenario contrario al de los repartidores: la posición fija de estos sensores permite una mayor exposición de estos a ataques, pero su ubicación en el entorno privado de la empresa dificultará en mayor medida la proximidad de un atacante a una posición desde la que pueda ejecutar estos ataques.

Las comunicaciones Bluetooth de estos dispositivos podrían sufrir ataques como:

- *Sniffing*
- KNOB (*sniffing* con componente activa)
- BlueBorne (inyección de *malware* por desbordamiento de pila)
- Btlejuice (ataques MITM)

En primer lugar, una falta de aplicación de mecanismos de encriptación seguros podría derivar en la captura de información relevante intercambiada entre los terminales que usan los repartidores durante su trabajo o entre los sensores de los centros. En este mismo hilo de vulnerabilidades, tal y como hemos visto en la descripción de KNOB, una posible ejecución de este ataque que estableciera el nivel de encriptación al mínimo de 8 bits permitiría a cualquier atacante acceder fácilmente a la información intercambiada por Bluetooth. ¿Son útiles estos datos para un atacante? Es posible que no, pero nunca se sabe cuáles pueden ser los objetivos de un atacante y, por tanto, una información que nos pueda parecer poco relevante podría aportarle una valiosa información a un atacante (por ejemplo, tener conocimiento del estado de las puertas de entrada a los centros para planear un robo). Asimismo, si la gestión de la autenticación de las comunicaciones no es adecuada se abriría la puerta a que un atacante pudiera enviar órdenes maliciosas a los sensores de los centros, permitiendo así boicotear su operativa.

Como se ha podido observar en el estudio de los ataques BlueBorne, la posibilidad de que un dispositivo pueda estar afectado es muy alta, tanto por sistema operativo (Linux, Windows y Android están afectados en distinta medida) como por cercanía del descubrimiento del ataque. Por ejemplo, suponiendo que el terminal móvil que los repartidores utilizan para sus labores son Android, un atacante podría explotar las vulnerabilidades relacionadas con el protocolo BNEP para tomar el control del terminal y extraer información valiosa o expandir algún tipo de malware por la red corporativa a la que está conectado el *smartphone*. Cualquier otro ataque de este tipo sobre los sensores de los centros de logística sería posible en función del sistema operativo que se utilice para la solución ideada por la empresa (p.ej., se entiende que habrá un servidor central que recoja los datos de los sensores y que será con una gran probabilidad Linux o Windows, afectados por BlueBorne).

6.1.3. Contramedidas

Las comunicaciones Bluetooth corporativas deben tener un nivel de protección alto que evite la posibilidad de que se puedan vulnerar la integridad o la operativa de la compañía, especialmente si se utilizan en un rango de distancias accesibles para los viandantes. En términos generales, la empresa A deberá llevar a cabo una serie de medidas de seguridad para prevenir posibles ataques a través de Bluetooth:

- Utilizar dispositivos y terminales con soporte al menos hasta la versión 4.2 para garantizar el uso de *Secure Connections* y configurarlos para implementar por obligación este nivel de protección. De este modo se garantiza el uso de los algoritmos más eficaces presentes en el estándar.
- Utilizar encriptación de como mínimo 114 bits en todas las comunicaciones para protegerse frente a escuchas pasivas. Implementar dentro de sus aplicaciones corporativas por Bluetooth la comprobación de la longitud de clave de encriptación a nivel de aplicación. De este modo la empresa estará protegida frente a KNOB hasta que el SIG presente una actualización de especificación que corrija esta situación.

- No permitir el uso de *Just Works* como modo de emparejamiento en sus sistemas. En caso de utilizar *Passkey Entry*, garantizar que las claves sean aleatorias y renovadas en cada uso.
- No establecer nombres de dispositivo que puedan dar pistas sobre sistema operativo, compañía o dueño del terminal.
- Establecer el mínimo nivel de potencia para reducir el rango de alcance de un atacante.

Se asume que las soluciones corporativas que utilizan Bluetooth son proyectos desarrollados por un departamento especializado o por una consultora encargada de diseñar la solución, por lo que todos estos parámetros y requisitos a nivel de configuración deben ser perfectamente ejecutables.

Asimismo, a nivel de política de seguridad administrativa, se recomiendan las siguientes medidas:

- Desarrollar un plan de formación para empleados para concienciar sobre los riesgos de las comunicaciones Bluetooth e instruir con conductas seguras. P.ej., los repartidores deberán saber que en caso de precisar emparejar sus terminales en medio de su jornada deberán tratar de hacerlo en un entorno lo más seguro posible que le permita su situación, y siempre deberán deshabilitar sus interfaces Bluetooth al finalizar su jornada.
- Mantener un inventario de los terminales utilizados por la empresa y realizar auditorías periódicas.
- En caso de pérdida, robo o avería de un dispositivo, deberá eliminarse inmediatamente de las listas de dispositivos conocidos del resto de terminales.
- Mantener los dispositivos actualizados a las últimas versiones posibles tras las pruebas de certificación necesarias. Hay que tener en cuenta que algunas de las vulnerabilidades estudiadas (como BlueBorne) son fácilmente neutralizables mediante actualizaciones de los fabricantes que solventen vulnerabilidades a nivel de código.
- Contar con seguridad en las instalaciones de la empresa que controlen posibles actividades sospechosas. Esto es especialmente importante en el caso de que los edificios donde se encuentran operando las comunicaciones Bluetooth corporativas como la domótica industrial mencionada no cuenten con un perímetro alejado de la vía pública.

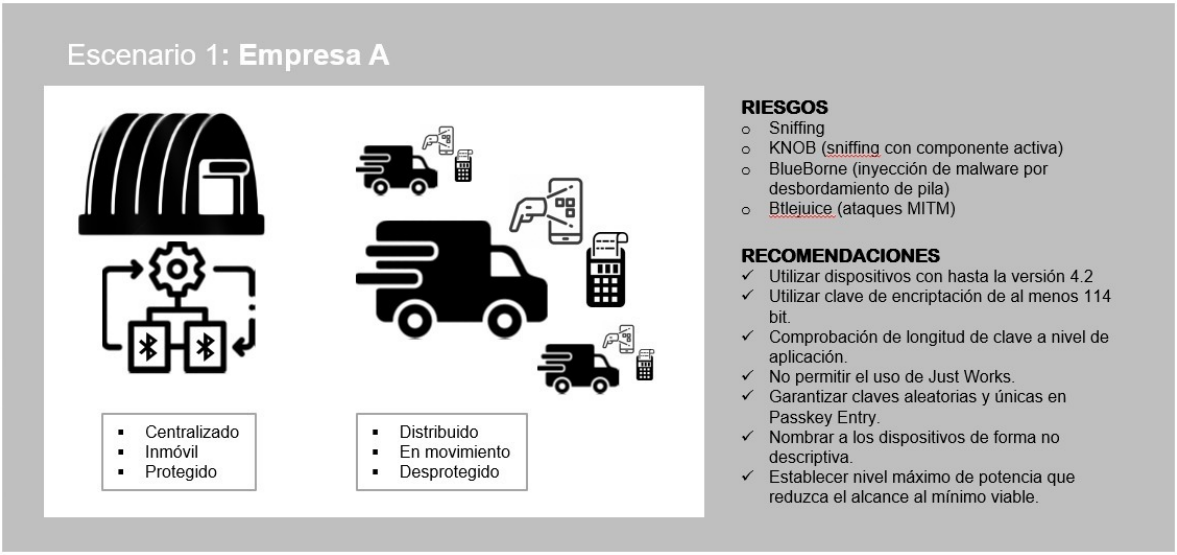

Ilustración 37: Resumen escenario 1

6.2. Escenario 2: empresa B

6.2.1. Descripción

Supongamos una gran empresa B con un elevado número de empleados dedicada al sector bancario. Esta empresa cuenta con una gran sede formada por una ciudad empresarial, con seguridad privada y control de acceso para empleados, así como de numerosas oficinas localizadas en locales comerciales para la atención de los clientes.

6.2.2. Riesgos

Las grandes empresas son el objetivo principal de los cibercriminales al contar con información confidencial de gran valor. En el caso de la empresa B esta confidencialidad es especialmente grave, ya que hablamos de los datos bancarios de los clientes y del acceso a los sistemas de la entidad, que en última instancia y poniéndonos en el peor caso podría llegar a implicar el robo de divisa electrónica. El uso de dispositivos con interfaz Bluetooth podría abrir la puerta a que un atacante ejecutara ataques como:

- *Sniffing*
- KNOB (*sniffing* con componente activa)
- MouseJack (acceso no autorizado a datos)
- BlueBorne (inyección de malware por desbordamiento de pila)
- Btlejuice (ataques MITM)

Hay que tener en cuenta que, tal y como se ha visto en el punto 4.2 y el punto 5, el equipamiento necesario para ejecutar estos ataques es muy accesible, pudiendo servir un portátil de gama media-alta con *dongles* de no más de 100 euros.

Un atacante podría por tanto vulnerar los ratones y teclados de los empleados de la empresa B mediante el uso de MouseJack para extraer documentos de ordenadores personales o para instalar *malware* más específico, como los célebres *ransomware*, que se pueda replicar por la red corporativa. Asimismo, cualquier terminal que no haya sido parcheado adecuadamente podrá ser atacado mediante las vulnerabilidades explotadas por BlueBorne, que como hemos visto puede permitir tomar el control de terminales Android y ejecutar ataques Bluetooth Pineapple en sistemas Windows, sin duda los dos sistemas operativos más extendidos tanto en el entorno doméstico como en el corporativo.

Es necesario remarcar que en este escenario contemplamos dos situaciones distintas que modifican el riesgo de un ataque. Por un lado, la sede central, estando ubicada en unas instalaciones de gran tamaño y con riguroso control de acceso exclusivo para empleados, hay que valorar que las posibilidades de que un atacante pueda establecer contacto con los dispositivos de los empleados es baja, dada la necesidad de encontrarse cerca. Edificios de oficinas en plantas altas con control de acceso en el hall o incluso previo a la llegada a los edificios se encontrarían ampliamente protegidos frente a ello. Sin embargo, este no sería el caso de las oficinas de atención al público, ya que no solo suelen ubicarse en locales comerciales a pie de calle, sino que cualquier atacante tendría acceso a la propia oficina como cliente. Es por ello que el nivel de riesgo de estas oficinas, al contrario de la sede central, sería alto.

6.2.3. Contramedidas

En primer lugar, es necesario indicar que, dada la inseguridad inherente a las comunicaciones inalámbricas, ninguna compañía que se encuentre altamente expuesta y tenga unos requisitos de seguridad altos debe utilizar comunicaciones Bluetooth con fines corporativos. Por este motivo se recomienda efusivamente que los ordenadores de los empleados tengan deshabilitadas las capacidades Bluetooth con el fin de cerrar esa puerta a los atacantes. Asimismo, en las oficinas comerciales se debe prohibir el uso de ratones y teclados inalámbricos, ya que suponen una posible entrada a los equipos corporativos mediante ataques MouseJack. Su uso en sede tendría menos riesgo por las características de protección geográfica comentadas en el punto anterior, pero en este estudio no se recomienda ya que el valor añadido que aportan a la operativa de los empleados no justifica el riesgo que puede suponer en caso de presencia de cibercriminales y aparición de nuevas vulnerabilidades.

Siendo pues los terminales móviles personales de los empleados los puntos Bluetooth accesibles en este entorno, la empresa deberá llevar a cabo un plan de formación y concienciación para los empleados, que los haga conscientes de los riesgos e implicaciones que tiene el uso de comunicaciones y dispositivos inseguros en un entorno corporativo de alta confidencialidad. Los empleados deberán:

- Mantener sus terminales actualizados siempre a la última versión disponible de su sistema operativo. Muchos de los ataques estudiados en el presente documento se solventan mediante los parches de los desarrolladores (p.ej. BlueBorne).

- Mantener el interfaz Bluetooth de sus terminales deshabilitado cuando no esté en uso. No hay mayor protección que cerrar esa puerta cuando no sea necesaria.
- Evitar realizar el proceso de *pairing* en entornos no seguros (oficina comercial, lugares públicos, etc.). El proceso de *pairing* es el momento más vulnerable a la hora de atacar un enlace Bluetooth (p.ej. KNOB).
- No utilizar el *smartphone* como punto de conexión WiFi para equipos corporativos (p.ej. un portátil de empresa). La implementación del protocolo BNEP sobre Android es una de las puertas de entrada para realizar un ataque BlueBorne.
- Evitar manejar y almacenar documentación confidencial con el terminal personal. De este modo un atacante que accediera a él se vería aún aislado de la información comprometedora de la compañía.

Es posible que estos terminales personales sean corporativos, ya que es habitual que una empresa de grandes dimensiones facilite terminales móviles a sus empleados. En este caso la empresa tiene la posibilidad de bastionar adecuadamente los *smartphones* entregados a sus trabajadores para velar por su máximo nivel de seguridad frente a atacantes aplicando un nivel de detalle que un usuario básico no alcanzará. Entre estas medidas de bastionado de terminales corporativos se incluyen:

- Configuración del nivel de potencia del interfaz Bluetooth al mínimo para reducir el rango de ataque.
- Facilitar a los empleados únicamente terminales con Bluetooth versión 4.2 o posterior, a fin de poder configurar por defecto únicamente el uso de *Secure Connections* en sus comunicaciones.
- Configuración de un nombre de terminal que no dé información sobre el usuario o su sistema, a fin de dar la mínima información posible a un atacante.
- Instalación de antivirus específicos cuando sea posible.

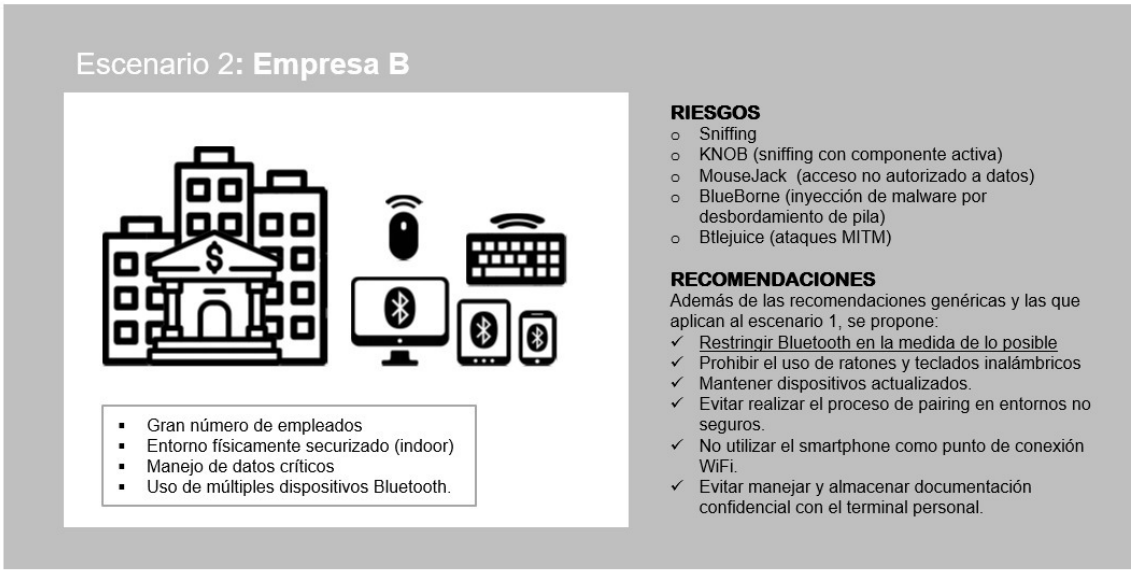

Ilustración 38: Resumen escenario 2

7. Conclusiones

El presente estudio ha pretendido ofrecer una visión general del estado de la seguridad del estándar Bluetooth en su actual versión 5.1, así como dar una visión pormenorizada de las distintas amenazas a las que se encuentran expuestos los dispositivos que utilizan esta tecnología según su antigüedad, su versión y su configuración.

A lo largo de este documento se ha podido observar que la seguridad ha sido un aspecto de Bluetooth al que sus desarrolladores han ido dando una mayor importancia desde sus primeros pasos hasta hoy, con esta tecnología extremadamente asentada en el mercado internacional. Así, la especificación Bluetooth ha ido desde mecanismos de seguridad laxos, o en algunos casos nulos, hasta modos de configuración altamente protegidos. Los fabricantes son conscientes de que en un mundo hiperconectado en el que el número de terminales con interfaz Bluetooth por persona es ampliamente mayor que uno y en el que Bluetooth no deja de aumentar su margen operativo con soluciones de domótica, salud o industria, las comunicaciones inalámbricas en la banda ISM son las más accesibles para ser atacadas, y que es de vital importancia ofrecer las máximas garantías para salvaguardar la integridad de los usuarios sin aumentar la dificultad de uso de sus terminales.

Un análisis pormenorizado de las últimas vulnerabilidades conocidas de la especificación y de los tipos de ataques y herramientas más comunes que se pueden sufrir cuando se utiliza esta tecnología, ha permitido constatar en primer lugar lo mencionado en el punto anterior: los primeros pasos de Bluetooth no contaron con todas las garantías de seguridad que se puedan esperar de un estándar de primera necesidad. Así, algunos de los ataques vistos son fruto de malas políticas de seguridad fácilmente solucionables, tal y como se ha podido ver por ejemplo en el caso de Mousejack, cuya raíz principal es la negligencia de algunos fabricantes de periféricos. De este estudio se deduce que este tipo de problemáticas se verán con poca frecuencia en el futuro, siendo hoy en día el estudio y la mejora de la seguridad de los sistemas de comunicaciones uno de los hitos de mayor importancia en el sector TIC en general, y que habrá que enfrentarse a amenazas tales como las dos publicadas más recientemente y recogidas en este documento, BlueBorne y KNOB, cuya naturaleza se basa en vulnerabilidades mucho más precisas y difíciles de detectar que, en principio, deberían ir resolviéndose mediante actualizaciones de los fabricantes tan pronto como sean descubiertas por los expertos.

Es por ello que la conclusión más importante que se saca de este estudio es que es de vital importancia que las compañías contemplen Bluetooth dentro de sus diagramas de riesgos, se pongan requisitos de configuración a los sistemas que garanticen el máximo nivel de protección y que se forme adecuadamente a los empleados no solo para llevar a cabo una serie de buenas prácticas con los dispositivos Bluetooth corporativos, sino con sus propios terminales personales, que a día de hoy en muchos casos se utilizan indistintamente para fines laborales o personales. Como suele verse en todos los aspectos de la ciberseguridad, el 100% de protección nunca se podrá alcanzar, pero una política adecuada y unas

prácticas concienciadas pueden neutralizar una gran mayoría de las amenazas a las que nos encontramos expuestos.

Inicialmente se pretendía que este trabajo tuviera menor alcance en cuanto al estudio de ataques y ofrecer en último lugar una serie de pruebas reales en las que poder reproducir los ataques mencionados y demostrar la vulnerabilidad de Bluetooth en función de los distintos niveles de protección configurados sobre los dispositivos empleados. Sin embargo, la curva de aprendizaje de entrada a la hora de poder desarrollar el escenario de pruebas necesario y las herramientas implicadas no ha sido compatible con el tiempo disponible para la finalización del trabajo como habría sido satisfactorio. Es por ello que se consideró en el ecuador de la planificación modificar el alcance del trabajo y descartar las pruebas en entorno real para ofrecer en su lugar un estudio más detallado de las amenazas presentes en Bluetooth, así como dedicar un espacio específico a una descripción en profundidad de los dos ataques más recientes e importantes descubiertos, BlueBorne y KNOB, cuyo conocimiento se ha considerado de una importancia especial para cualquier persona que quiera informarse técnicamente sobre el estado de la seguridad en Bluetooth debido a la juventud de estos ataques. Asimismo, a fin de enriquecer el conocimiento aportado por este trabajo, se ha decidido incluir dos ejemplos de escenario corporativos reales con distintos niveles de seguridad y de uso de tecnología Bluetooth para detallar sus riesgos y las medidas que dichas empresas deberían llevar a cabo para protegerse al máximo frente a ataques.

En definitiva, se ha pretendido que este documento otorgue a su lector unas bases lo suficientemente amplias sobre el funcionamiento general del estándar Bluetooth, los mecanismos de seguridad implementados dentro de su especificación y un repaso de las distintas vulnerabilidades y amenazas a las que ha estado y está expuesta esta tecnología, así como una serie de contramedidas que deben permitir reforzar la protección de cualquier usuario a la hora de usarla.

Queda pendiente como posible continuación de este trabajo llevar a cabo pruebas reales para simular los distintos ataques que se han enunciado en este documento, poniendo especial hincapié en los más recientes y graves, como han sido KNOB y BlueBorne. Habría sido deseable haber podido incluir este apartado práctico dentro de este trabajo para hacerlo más completo aún, pero tanto por tiempo como por extensión se plantea quizás como mejor opción dedicarle un trabajo aparte a su implementación.

8. Glosario

A continuación, se listan los términos y acrónimos acuñados en este documento que puedan necesitar aclaración sobre su significado, por orden alfabético:

- **ACL** *Asynchronous Connection-oriented Logical transport.*
- **ADVB** *Advertising Broadcast*, uno de los tipos de transportes lógicos.
- **AES-CCM** *Advanced Encryption Standard Counter* con CBC-MAC.
- **AMP** *Alternate Media Access Control and Physical*.
- ***AoA*** *Angle of Arrival.*
- **ASB** *Active Slave Broadcast*, uno de los tipos de transportes lógicos.
- **BR/EDR** *Bluetooth Basic Rate/Enhanced Data Rate*.
- **Broadcast** Difusión amplia, a más de un receptor.
- ***Claimant*** Dispositivo Bluetooth que intenta probar su identidad al dispositivo *Verifier* durante el proceso de conexión.
- **CSB** *Connectionless Slave Broadcast*, uno de los tipus de transportes lógicos.
- **CSRK** *Connection Signature Resolving Key*, clave utilizada para el firmado de datos LE.
- **CVE** *Common Vulnerabilities and Exposures*, lista de información registrada sobre vulnerabilidades de seguridad conocidas
- **DHKEY** Clave utilitzada para generar clave de cifrado.
- **Eavesdropping** Escucha (espionaje), una de las amenazas clave en redes inalámbricas.
- **ECDH** *Elliptic-curve* Diffie–Hellman, protocolo de generación de claves pública-privada
- **eSCO** *Extended* SCO, uno de los tipos de transportes lógicos.
- **FDMA** *Frequency Division Multiple Access*, acceso múltiple por división de frecuencia
- **FHSS** *Frequency Hopping Spread Spectrum*, espectro ensanchado por salto de frecuencia, técnica de modulación donde se emite saltandoa diferentes frecuencias, aparentemente aleatorias.
- **FIPS** *Federal Information Processing Standards*, lista de estándares recomendados por el gobierno de EEUU.
- **Fuzzing** Técnica de prueba de software o ataque, donde se envían datos inesperados o inválidos a un dispositivo para analizar su comportamiento o forzar el error, respectivamente.
- **GAP** *Generic Access Profile*, perfil genérico de Bluetooth del que parten los demàs.
- **GFSK** *Gaussian-Shift Keying*, variante de la modulación FSK donde la información se pasa por un filtro gaussiano antes de ser modulada.
- **Inquiry** Investigación, procedimiento por el cual un dispositivo transmite y escucha mensajes.
- **IoT** *Internet of Things*
- **IRK** Identity Resolving Key, clave utilitzada para la privacidad en LE.
- **ISM** *Industrial, Scientific, Medical,* banda no licenciada entre 2.400 y 2.483,5 MHz en que trabaja Bluetooth.
- **L2CAP** *Logical Link Control and Adaptation Protocol*, uno de los protocolos básicos de Bluetooth.
- **LAP** *Lower Address Part,* parte de la dirección Bluetooth de un dispositivo que es assignada por el fabricante.
- **LE** *Low Energy*.
- **LMP** *Link Manager Protocol.*

- **LTK** *Long Term Key*, clave de emparejamiento de Secure Connections.
- **M2M** *Machine to Machine*.
- **Master** Maestro, el dispositivo que domina la comunicación en Bluetooth.
- **MITM** *Man in The Middle*, un tipo de ataque de suplantación.
- **NAP** *Non-significant Address Part*, parte de la dirección Bluetooth utilizada para la sincronización en el salto de frecuencia.
- **NFC** *Near Field Communication*, protocolo de comunicación en campo cercano.
- **NIST** *National Institute of Standards and Technology*, Instituto Nacional de Estándares y Tecnología de EEUU.
- **OOB** *Out of Band*, método de autenticación donde se realiza una comunicación fuera de la banda de Bluetooth.
- **PADVB** *Periodic Advertising Broadcast*, uno de los tipos de transportes lógicos.
- **PAL** *Protocol Adaptation Layer*.
- **PC** *Personal Computer*.
- **PEC** Prueba de Evaluación Contínua.
- **PHY** *Physical Layer*, capa física.
- **PIN** *Personal Identification Number*.
- **PSK** *Phase Shift Keying*, modulación en fase.
- **RNG** *Random Number Generator*, generador de números aleatorios.
- **SAFER+** Nombre de una familia de algoritmos de cifrado por bloques que se utiliza en Bluetooth como algoritmo de autentificación y para la generación de claves.
- **SC** *Secure Connections*.
- **SCO** *Synchronous Connection-oriented Logical transport*, uno de los tipos de transportes lógicos.
- **SHA-256** *Secure Hash Algorithm-256*, función hash de 256 bits.
- **SIG** Bluetooth Special Interest Group, grupo de empresas que desarrolló (y desarrolla) el estándar Bluetooth, compuesto originariamente por IBM, Intel, Toshiba, Nokia y Ericsson.
- **Slave** Esclavo.
- **SMP** *Security Manager Protocol.*
- **Sniffer** Dispositivo que captura las tramas que se transmiten por una red.
- **SSP** *Secure Simple Pairing*, mecanismo de seguridad mediante el cual dos dispositivos Bluetooth pueden establecer una comunicación segura.
- **TDD** *Time-Division Duplex*, esquema de multiplexación en el tiempo
- **TDMA** *Time Division Multiple Access*, acceso múltiple por división en el tiempo.
- **UAP** *Upper Address Part*, parte de la dirección Bluetooth de un dispositivo.
- **UART** *Universal Asynchronous Receiver-Transmitter*, Transmisor-Receptor Asíncrono Universal, es el dispositivo que controla los puertos y dispositivos sèrie.
- **USB** *Universal Serial Bus*, bus de comunicaciones estándar.
- **Verifier** Dispositivo que verifica la identidad de un *claimant*.
- **WPAN** *Wireless Personal Area Network*, área personal inalámbrica.

9. Bibliografía

A continuación, se listan las referencias utilizadas para la realización de este documento.

1. Bluetooth Market Update (2019), Bluetooth SIG, Inc.
   - Url: https://3pl46c46ctx02p7rzdsvsg21-wpengine.netdna-ssl.com/wp-content/uploads/2018/04/2019-Bluetooth-Market-Update.pdf
2. Bluetooth Core Specification v5.1, Bluetooth SIG, Inc., 21 de enero de 2019 (Consultado durante toda la redacción del trabajo)
   - Volumen 1 parte A
   - Volumen 2 parte H
   - Volumen 3 parte H
   - Url Descarga: https://www.bluetooth.com/specifications/bluetooth-core-specification/
3. Bluetooth® Low Energy Channels
   - Url: https://microchipdeveloper.com/wireless:ble-link-layer-channels
4. Robert Davidson, Akiba, Carles Cufí, Kevin Townsend (O'Reilly Media, Inc., 2014), Getting Started with Bluetooth Low Energy
5. Guía de Seguridad de las TIC CCN-STIC 837 (Febrero 2019), Centro Criptológico Nacional).
   - Url: https://www.ccn-cert.cni.es/series-ccn-stic/800-guia-esquema-nacional-de-seguridad/2707-ccn-stic-837-ens-seguridad-en-bluetooth/file.html
6. Guide to Bluetooth Security, Special Publication 800-12, Revision 2 (Mayo 2017), National Institute of Standards and Technology, NIST
   - Url: https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-121r2.pdf
7. CVE Database
   - Url: https://cve.mitre.org/cgi-bin/cvekey.cgi?keyword=bluetooth
8. Vulnerability Note VU#304725, NVD
   - Url: https://www.kb.cert.org/vuls/id/304725/
9. Dr. Ghalib A. Shah, Bluetooth/Wireless Personal Area Network (WPAN),
   - Url: https://slideplayer.com/slide/10633636/
10. Da-Zhi Sun, Yi Mu y Willy Susilo, (2017), Man-in-the-middle attacks on Secure Simple Pairing in Bluetooth
    - Url: https://ro.uow.edu.au/cgi/viewcontent.cgi?article=1706&context=ei spapers1
11. Yi Lu, Willi Meier y Serge Vaudenay, The Conditional Correlation Attack: A Practical Attack on Bluetooth Encryption
    - Url: https://www.iacr.org/archive/crypto2005/36210097/36210097.pdf
12. Angela M. Lonzetta, Peter Cope, Joseph Campbell, Bassam J. Mohd y Thaier Hayajneh, (2018), Security Vulnerabilities in Bluetooth Technology as Used in IoT.

    - Url: https://www.mdpi.com/2224-2708/7/3/28/pdf
13. Hcitool Man Page
    - Url: https://www.systutorials.com/docs/linux/man/1-hcitool/
14. JP Dunning, Virginia Tech (2010), Breaking Bluetooth by being bored, DefCon 2010.
    - Url: https://www.defcon.org/images/defcon-18/dc-18-presentations/Dunning/DEFCON-18-Dunning-Breaking-Bluetooth.pdf
15. Bluetooth Hacking Tools Comparison
    - Url: https://duo.com/decipher/bluetooth-hacking-tools-comparison
16. Btlejuice
    - Url: https://github.com/DigitalSecurity/btlejuice
17. Kaviarasu.S and Muthupandian.P (2016), Bluejacking Technology: A Review, IT Department, Sri Krishna Arts and Science College, Coimbatore, TamilNadu, India
    - Url: https://www.researchgate.net/publication/314233155_Bluejacking_Technology_A_Review
18. BlueSmack
    - Url: https://trifinite.org/trifinite_stuff_bluesmack.html
19. BlueSnarf
    - Url: https://www.finjanmobile.com/what-is-bluesnarfing/
20. Caribe
    - Url: http://virus.wikidot.com/caribe
21. Mousejack
    - Url: https://github.com/BastilleResearch/mousejack/blob/master/doc/pdf/MouseJack-whitepaper-v1.1.pdf
22. Ben Seri & Gregory Vishnepolsky (2017), BlueBorne. The dangers of Bluetooth implementations: Unveiling zero day vulnerabilities and security flaws in modern Bluetooth stacks, Armis.
    - Url: https://go.armis.com/hubfs/BlueBorne%20Technical%20White%20Paper-1.pdf
23. Daniele Antonioli, SUTD; Nils Ole Tippenhauer, CISPA; Kasper B. Rasmussen, University of Oxford (2019), The KNOB is Broken: Exploiting Low Entropy in the Encryption Key Negotiation Of Bluetooth BR/EDR, 28th USENIX Security Symposium.
    - Url: https://www.usenix.org/system/files/sec19-antonioli.pdf