Seguridad en redes inalámbricas de alcance personal

Cristina Pérez Solà

PID_00191700

Universitat Oberta de Catalunya
www.uoc.edu

Índice

Introducción 5

Objetivos 6

1. RFID 7
1.1. Descripción de la tecnología RFID 7
1.2. Seguridad en dispositivos RFID 9
1.2.1. Ataques a sistemas RFID 11
1.2.2. Soluciones criptográficas para RFID 13

2. Bluetooth 21
2.1. Descripción de la especificación Bluetooth 21
2.2. Seguridad en dispositivos Bluetooth 22
2.2.1. Modo de seguridad 2: nivel de servicio 22
2.2.2. Modo de seguridad 3: nivel de enlace 24

3. ZigBee 30
3.1. Descripción de la especificación ZigBee 30
3.1.1. Arquitectura 31
3.1.2. Tipos de dispositivos y topologías 32
3.2. Seguridad en dispositivos ZigBee 33
3.2.1. Claves 35
3.2.2. Seguridad en la capa de red 35
3.2.3. Seguridad en la capa de aplicación 38

4. Comparativa y discusión de la seguridad 42

Resumen 44

Actividades 45

Ejercicios de autoevaluación 45

Solucionario 46

Glosario 47

Bibliografía 50

Introducción

Las redes inalámbricas de alcance personal (también conocidas como WPAN por su nombre en inglés, *wireless personal area networks*) son redes formadas por dispositivos, posiblemente heterogéneos, que se encuentran a poca distancia (por lo general, del orden de pocos metros). La magnitud exacta de la distancia a la que se pueden comunicar varios dispositivos, así como las características específicas de la comunicación, estarán determinadas por la tecnología específica que se utilice en cada caso.

En este módulo didáctico veremos a qué problemas de seguridad se enfrentan las WPAN, así como las alternativas que se suelen utilizar habitualmente para solucionarlos. Para ello, nos centraremos en tres de las tecnologías más populares para crear este tipo de redes: RFID, Bluetooth y ZigBee.

En primer lugar, veremos una descripción de la tecnología RFID y de los ataques que pueden experimentar los dispositivos que la utilizan. A continuación, veremos qué se puede hacer para conseguir primitivas criptográficas para dispositivos RFID, haciendo hincapié en los recursos necesarios para implementarlas, y detallaremos algunos de los protocolos que se emplean para proporcionar seguridad en estos sistemas.

En segundo lugar, describiremos el estándar Bluetooth y los mecanismos de seguridad que prevé. Repasaremos los diferentes modos de seguridad de Bluetooth y detallaremos algunos de sus procesos básicos, como los que permiten autenticar dispositivos.

Finalmente, comentaremos el estándar ZigBee y examinaremos los mecanismos de seguridad que incorpora. Expondremos los diferentes servicios de seguridad que ZigBee ofrece para cada una de las capas, analizando algunos de los protocolos del estándar, como el protocolo de establecimiento de claves o el protocolo de autenticación mutua.

Objetivos

En los materiales didácticos asociados a este módulo, el alumnado encontrará las herramientas y los contenidos necesarios para alcanzar los objetivos siguientes:

1. Conocer las tecnologías y los estándares básicos para redes inalámbricas de alcance personal (WPAN).
2. Identificar los problemas de seguridad que surgen en entornos WPAN.
3. Comprender las propiedades que se quieren garantizar cuando se habla del diseño de sistemas para WPAN seguros.
4. Adquirir un conocimiento genérico de las técnicas criptográficas que se utilizan para afrontar los problemas de seguridad de las redes WPAN.
5. Entender las limitaciones de cada una de las tecnologías expuestas, así como lo que implican a la hora de diseñar sistemas seguros.

1. RFID

Comenzaremos el estudio de las tecnologías para redes WPAN analizando la RFID. En primer lugar, describiremos la tecnología RFID y clasificaremos los diferentes tipos de dispositivos RFID según su capacidad de cómputo. Esta clasificación nos será útil más tarde para analizar las soluciones criptográficas que se presentan y su adaptabilidad a los diferentes dispositivos.

A continuación, pasaremos a enumerar los ataques a los que son vulnerables los sistemas RFID, para proponer, después, soluciones que permitan minimizar o evitar completamente los efectos de tales ataques. En este sentido, veremos cómo podemos crear primitivas criptográficas adecuadas a los dispositivos RFID y estudiaremos algunos de los protocolos que permiten garantizar ciertas propiedades de los sistemas seguros.

1.1. Descripción de la tecnología RFID

La RFID* es una tecnología que permite la comunicación inalámbrica a partir de la emisión de ondas de radiofrecuencia. Los principales componentes de un sistema RFID son el lector y el transponedor (o etiqueta). La tecnología RFID se utiliza hoy en día en una gran variedad de escenarios, desde pasaportes hasta sistemas de control de acceso, pasando por sistemas de tiques o mecanismos de protección contra falsificaciones.

*** Del inglés, *radio frequency identification*.**

Los principales elementos de una etiqueta RFID son la antena y el chip. La antena es la encargada de emitir y recibir las ondas de radiofrecuencia que posibilitan la comunicación. Por su parte, el chip incorpora un modulador y un desmodulador que procesan las señales y unos circuitos que implementan la memoria y las herramientas de procesado de la información. Opcionalmente, las etiquetas pueden incorporar otros circuitos con tareas más específicas.

Con el nombre de etiqueta RFID, en realidad se engloba un amplio abanico de dispositivos con características muy diferentes. Mientras que algunas de las etiquetas RFID no disponen de batería y utilizan la señal que les llega del lector para inducir una pequeña corriente eléctrica suficiente para operar (etiquetas pasivas), otras etiquetas sí disponen de una fuente de energía propia y pueden funcionar de manera autónoma (etiquetas activas). A medio camino, encontramos los dispositivos semipasivos, que disponen de una batería que solo se utiliza para computaciones internas pero que necesitan energía externa para hacer posible la comunicación. La fuente de energía no es lo único

que diferencia a las etiquetas RFID. Otros parámetros, como el tamaño, también varían enormemente de un tipo de etiqueta a otro. Dado que el tamaño del área del chip está condicionado por la tecnología que se utiliza, la medida de la capacidad del chip no se hace directamente con el área disponible, sino que se realiza en puertas lógicas equivalentes (o GE*).

*** Del inglés, *gate equivalent*.**

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

La puerta lógica NAND

Las puertas NAND son especialmente importantes en el diseño de circuitos lógicos, ya que cualquier función booleana se puede implementar con solo utilizar una combinación de puertas de este tipo.

Así pues, la capacidad de cómputo de una etiqueta RFID medida en GE es uno de los parámetros que puede variar mucho según el tipo concreto de etiqueta. Estas diferencias hacen que las soluciones de seguridad propuestas para las etiquetas con más recursos puedan no ser viables para las más limitadas y que, por tanto, sea necesario tener muy claro con qué tipo de dispositivo RFID se desea trabajar antes de definir los mecanismos de seguridad que se quieren implementar. En general, podemos clasificar las etiquetas RFID en tres categorías:

- **Gama baja**. Etiquetas de bajo coste con menos de cinco mil GE. Mientras que algunas de estas etiquetas no disponen de ningún tipo de mecanismo de seguridad ni privacidad, otras incorporan funcionalidades de seguridad básicas como sumas de verificación (*checksums*), un comando kill protegido mediante contraseña, una contraseña de acceso o un generador de números seudoaleatorios. Las RFID de gama baja se utilizan, principalmente, para realizar identificación automática.

  Por ejemplo, se encuentran dispositivos de gama baja como etiquetas antirrobo adheridas a los productos en venta de una tienda. Las etiquetas EPC (*electronic product code*) *Class 1 Gen 2* son un ejemplo comercial de esta categoría.

- **Gama media**. Etiquetas de coste moderado con transpondedores que permiten efectuar tanto operaciones de lectura como de escritura. Disponen de memoria de datos no volátil que puede variar desde los cien bytes hasta más de cien kB. Las etiquetas de esta gama suelen implementar protocolos de autenticación mutua y control de acceso a la memoria del transpondedor. Durante la comunicación entre la etiqueta y el lector, se establecen claves de sesión que permiten enviar los datos cifrados y asegurar su integridad.

  Encontramos etiquetas de gama media en inmovilizadores de automóviles (que evitan que el motor se ponga en marcha sin la presencia de la llave correcta), mecanismos de control de acceso a edificios o sistemas de tiques. Entre las etiquetas de esta gama, las más conocidas son las *MiFare Classic*.

El comando kill

El comando kill permite a un lector desactivar de forma permanente una etiqueta RFID.

- **Gama alta**. Etiquetas de coste más elevado que contienen chips de tarjetas inteligentes (*smartcards*) y que están equipadas con sistemas operativos específicos para dichas tarjetas. Las etiquetas de gama alta incorporan mecanismos de seguridad con funciones criptográficas avanzadas, normalmente específicas para cada aplicación. Las etiquetas con más prestaciones de esta gama pueden llegar a incorporar un coprocesador criptográfico que permite efectuar operaciones de criptografía de clave pública, como la creación de firmas digitales.

Con las siglas NFC* se engloba un conjunto de estándares basados en RFID que requieren que la etiqueta y el lector estén muy cerca, del orden de pocos centímetros, para establecer una comunicación entre ellos. El hecho de limitar la distancia existente entre dos dispositivos para que puedan comunicarse podría ser una limitación, pero también resulta deseable a la hora de mejorar la seguridad del sistema.

*** Del inglés, *near field communication*.**

Ejemplos de NFC

Ejemplos de aplicaciones que utilizan NFC son dispositivos de pago con el teléfono móvil, tarjetas de fidelización de líneas aéreas o billetes de metro multiviaje.

Así pues, las características de los diferentes dispositivos RFID estarán determinadas en gran medida por el coste máximo que se pueda asumir a la hora de producirlos. La capacidad de cómputo de dichos dispositivos limitará también el grado de seguridad que puede alcanzar cada uno de ellos.

1.2. Seguridad en dispositivos RFID

Existe un conjunto de propiedades objetivo cuando se intentan crear dispositivos RFID seguros. Mientras que las propiedades más básicas ofrecerán niveles de seguridad elementales, algunas de las más avanzadas garantizarán niveles de seguridad más elevados. El conjunto específico de propiedades que hay que garantizar vendrá determinado por las necesidades de la aplicación que se quiera dar al sistema:

- **Identificación**. La función principal de un lector RFID es, precisamente, identificar un valor único (una ID) que se asigna a cada etiqueta RFID. La propiedad de identificación permite a un lector descubrir la identidad de una etiqueta a partir de la salida de esta. A grandes rasgos, las etiquetas RFID reciben un identificador único cuando están en producción. Este identificador se escribe en la ROM (memoria solo de lectura) de la etiqueta, de manera que resulta muy difícil cambiarlo. Se puede conseguir identificación sin utilizar ningún tipo de técnica criptográfica, aunque esto puede dar como resultado una fuga de datos secretos, lo que podría dar lugar, por ejemplo, a ataques de *tracking* como los que veremos más adelante.

- **Autenticación**. Cuando un lector lee datos de una etiqueta, no puede saber si los datos que está recibiendo son de una etiqueta válida o no a menos

que se añada un sistema de validación. El mismo escenario se presenta en la dirección opuesta, cuando una etiqueta recibe datos de un lector. Para asegurar que las comunicaciones se hacen entre dispositivos válidos, hay que incorporar un mecanismo de autenticación en el sistema que permita garantizar que un lector solo aceptará los datos de una etiqueta y que una etiqueta solo aceptará los datos de un lector si pueden asegurar su validez. Los sistemas de control de acceso a edificios son un ejemplo de RFID que necesitarán implementar autenticación.

- **Privacidad**. El hecho de que la propiedad de identificación sea la básica de un sistema RFID y de que cada etiqueta contenga un identificador único hace que aparezcan problemas de privacidad asociados al uso de dispositivos RFID. Así, por ejemplo, se podrían seguir los movimientos de una persona o de un objeto que lleve una etiqueta RFID adherida. Las necesidades específicas de privacidad dependerán en gran medida de la aplicación concreta que se esté empleando en las etiquetas RFID.

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

- **Seguridad hacia delante***. Se trata de una extensión de las propiedades de autenticidad e indistinguibilidad que garantiza que dichas propiedades se mantienen para transacciones pasadas cuando un atacante es capaz de corromper una etiqueta en un momento determinado. Por ejemplo, es fácil imaginar un escenario en el que se tira a la basura una etiqueta RFID que disponía de mecanismos para garantizar la autenticidad y la indistinguibilidad una vez acabada su vida útil. En ese momento, un atacante puede recuperar la etiqueta, manipularla y obtener los valores secretos que contiene. Si, incluso con esta información, el atacante sigue sin ser capaz de distinguir entre las salidas de dos etiquetas que registró en el pasado (una de las cuales pertenecía a la etiqueta comprometida), entonces decimos que tiene seguridad hacia delante para la propiedad de indistinguibilidad.

  *** En inglés, *forward security*.**

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

- **Prueba de existencia**. La prueba de existencia es una propiedad que permite garantizar la existencia de una etiqueta particular en una localización concreta, en un tiempo determinado y con un conjunto de otras etiquetas particulares. Esta propiedad es necesaria, por ejemplo, en aplicaciones en las que se asignan etiquetas RFID a los diferentes componentes que forman parte de una cadena de suministro, de manera que varios lectores distribui-

dos a lo largo de la cadena puedan controlar su funcionamiento. En este caso, es interesante que el lector sea capaz de detectar que una serie de componentes se encuentran juntos en un espacio en un momento determinado (por ejemplo, si estos componentes se tienen que combinar para formar una sola pieza).

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

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

1.2.1. Ataques a sistemas RFID

Los sistemas RFID son vulnerables a diferentes tipos de ataques, algunos de los cuales aparecen también en muchos otros sistemas de información. Dependiendo del método utilizado para atacar al sistema, podemos clasificar los ataques a dispositivos RFID en varias categorías:

- **Lectura pasiva**. El método de lectura pasiva permite a un atacante escuchar los mensajes transmitidos entre el lector y la etiqueta, normalmente con la intención de descubrir información secreta. Al tratarse de un método pasivo, el atacante solo tiene la habilidad de escuchar los mensajes, sin poder manipularlos de ninguna forma. Lo que exponemos a continuación conforma un ataque que necesita realizar una lectura pasiva:

– Escuchas no autorizadas*. Es el ataque de lectura pasiva en el que el atacante simplemente escucha la comunicación entre una etiqueta y un lector. Dado que las comunicaciones RFID se producen de forma inalámbrica, es relativamente fácil para un atacante interceptar tales comunicaciones, siempre que se pueda situar cerca de los otros dispositivos. Este tipo de ataques puede ser difícil de detectar y puede comprometer la identificación, la autenticación o la privacidad del sistema RFID.

*** En inglés, *eavesdropping*.**

- **Lectura activa**. El método de lectura activa consiste en intentar leer información (de la etiqueta, del lector o del mismo canal inalámbrico) pero en este caso con la capacidad de modificar los mensajes que se intercambian la etiqueta y el lector y de interactuar con las distintas partes. La intención del atacante es descubrir información secreta o atacar al mecanismo de au-

tenticación. Se pueden identificar diferentes ataques que necesitan poder realizar lecturas activas para perpetrarse:

– Ataques de reinyección**. En un ataque de reinyección, el atacante registra la salida de una etiqueta y después envía esta salida hacia el lector. Normalmente, el momento en que el atacante registra la salida y el momento en que este la reproduce se encuentran separados en el tiempo. Los ataques de reinyección permiten romper algunos sistemas de autenticación.

** En inglés, *replay attacks*.

– Ataques de retransmisión*. En un ataque de retransmisión, el atacante lee la salida de una etiqueta, transporta esta salida a otra localización y envía la salida hacia un lector remoto. Este tipo de ataques también puede conseguir romper sistemas de autenticación.

* En inglés, *relay attacks*.

– Ataque de modificación del mensaje. En ataques de modificación del mensaje, el atacante intercepta y modifica la comunicación entre una etiqueta y un lector. Para evitar este tipo de ataques, se añaden mecanismos de control de integridad a los datos que se envían. Los ataques de este tipo pueden romper la identificación, la autenticación o la privacidad de un sistema RFID.

- **Reescritura**. El método de reescritura permite a un atacante reescribir la información almacenada en la etiqueta para obtener información secreta o para burlar al mecanismo de autenticación. Algunos ataques que emplean la reescritura son los siguientes:

– Ataque de reescritura de la etiqueta / del lector. En un ataque de reescritura de la etiqueta, el atacante reescribe los contenidos de la memoria de una etiqueta utilizando un lector falso. Este tipo de ataques se puede prevenir exigiendo a las etiquetas que autentiquen los lectores antes de permitir reescribir contenido o bien desplegando mecanismos de bloqueo de memoria. Del mismo modo, también se pueden considerar ataques de reescritura del lector, donde una etiqueta falsa consigue comprometer la memoria de un lector. Los ataques de este tipo pueden romper la identificación, la autenticación o la privacidad de un sistema RFID.

– Virus / software malicioso. Los ataques de reescritura pueden permitir la transmisión de software malicioso entre dispositivos RFID. Por ejemplo, un lector puede leer código malicioso de una etiqueta contaminada al ejecutar el código y pasar a reescribirlo en otras etiquetas a su alcance, con lo que se propaga el software malicioso.

- **Clonación**. El método de clonación consiste en crear una copia (un clon) de una etiqueta o de un lector. Un ataque de clonación sería el siguiente:

– Ataque de clonación de una etiqueta/lector. En un ataque de clonación, el atacante consigue una copia completa de otra etiqueta o lector. El uso de

mecanismos de autenticación o la creación de etiquetas y lectores a prueba de manipulaciones son mecanismos de defensa contra la clonación. Este tipo de ataques puede romper la identificación y la autenticación de un sistema RFID.

- **Destrucción/denegación de servicio**. Este tipo de ataques inhabilita el uso de los dispositivos RFID, ya sea destruyéndolos físicamente, saturándolos al crear más peticiones de las que se pueden atender o creando interferencias en el canal. Algunos ataques de denegación de servicio son los siguientes:

– Denegación de servicio y destrucción. En este ataque, el adversario intenta destruir físicamente una etiqueta para dejarla inutilizable o bien lleva a cabo un ataque de denegación de servicio que impide el uso del sistema a los usuarios legítimos.

– Ataques de creación de interferencias*. En este tipo de ataques, el adversario intenta bloquear el canal de comunicación inalámbrico creado entre una etiqueta y un lector o interferir sobre él.

* En inglés, *jamming*.

- **Rastreo****. Este método de ataque consiste en escanear una etiqueta, bien para obtener información, bien para detectar sus movimientos. Como ejemplos de ataques de rastreo tenemos los siguientes:

** En inglés, *tracking*.

– Ataques de escaneo. En un ataque de escaneo, el atacante intenta obtener información sobre el objeto que lleva una etiqueta RFID.

– Ataques de rastreo. En un ataque de rastreo, el atacante sigue a una etiqueta (o a la persona u objeto que lleva dicha etiqueta).

- **Canales laterales**. Este método consiste en aprovechar información lateral, como el consumo energético de las etiquetas o el tiempo de respuesta, para tratar de obtener información secreta. El siguiente es un ataque que utiliza el método de canales laterales:

– Ataques de canal lateral***. Dadas las características técnicas y físicas de los dispositivos RFID, un atacante puede observar la fuerza del campo electromagnético que se genera o analizar los tiempos de adquisición y procesado de datos para obtener información secreta almacenada en una etiqueta o un lector RFID.

*** En inglés, *side channel attacks*.

1.2.2. Soluciones criptográficas para RFID

Con el fin de prevenir, detectar o mitigar los ataques a sistemas RFID que acabamos de exponer, se pueden implementar soluciones criptográficas en los dispositivos RFID. Como hemos visto, los sistemas RFID disponen de unos re-

cursos limitados que vendrán determinados por el tipo de etiquetas o lectores que se utilicen. Así pues, la necesidad de conseguir dispositivos de bajo coste limita la complejidad de los algoritmos que se pueden implementar y, por tanto, las soluciones criptográficas aplicables.

El conjunto de técnicas criptográficas diseñadas para dispositivos con recursos limitados que intentan ofrecer un compromiso entre rendimiento, seguridad y coste se conoce como **criptografía ligera***.

* En inglés, *lightweight cryptography*.

Para conseguir implementaciones criptográficas utilizables en dispositivos RFID, normalmente se siguen cuatro enfoques distintos:

- implementar de manera eficiente sistemas de cifra ya existentes;
- utilizar sistemas de cifra ya existentes con parámetros más pequeños;
- diseñar nuevos mecanismos de cifrado, y
- desarrollar soluciones totalmente dedicadas.

En los siguientes subapartados, repasaremos algunas de las soluciones de criptografía ligera que se suelen utilizar en dispositivos RFID, tanto si son adaptaciones de sistemas que originalmente necesitaban muchos recursos como nuevas propuestas que se adaptan a los recursos disponibles en dispositivos RFID. En primer lugar, veremos algunas de las primitivas criptográficas más utilizadas y cómo se pueden utilizar en dispositivos RFID. Después, pasaremos a repasar algunos de los protocolos que permiten ofrecer ciertas garantías de seguridad en RFID.

Generadores de números seudoaleatorios

Generadores seudoaleatorios

Los generadores de números seudoaleatorios se conocen con las siglas PRNG (del inglés, *pseudorandom number generator*) o DRBG (del inglés, *deterministic random bit generator*).

Los generadores de números seudoalteatorios son algoritmos deterministas que permiten generar secuencias de números que parecen aleatorias. Estos algoritmos tienen como parámetro de entrada un valor inicial o semilla que permite inicializarlos y ofrecen una salida con unas propiedades similares a las que se observan en secuencias aleatorias.

Un **generador seudoaleatorio** es una función

$$G : \{0,1\}^s \rightarrow \{0,1\}^n$$

con $n \gg s$. La función G debe ser computable de manera eficiente para un algoritmo determinista y su salida debe ser impredecible.

En el ámbito criptográfico, es importante que las secuencias generadas por un generador seudoaleatorio sean impredecibles, es decir, que dados los k primeros bits de una secuencia, un atacante no sea capaz de predecir el bit $k + 1$ con una probabilidad superior a $1/2 + \epsilon$, para un valor de ϵ no despreciable. Esto implica que la secuencia de salida del generador es indistinguible de una secuencia realmente aleatoria.

Con el fin de decidir si la salida de un generador seudoaleatorio se parece a una secuencia aleatoria o no, se ha creado una serie de tests que las secuencias aleatorias cumplen y que, por tanto, se espera que las seudoaleatorias también satisfagan. Así, por ejemplo, uno de los estándares de EPC Global exige a las implementaciones de PRNG que generen secuencias siguiendo ciertas propiedades, como que la probabilidad de que un número cualquiera de dieciséis bits aparezca en la siguiente salida esté entre $0{,}8 \cdot 2^{-16}$ y $1{,}25 \cdot 2^{-16}$, entre otras.

Uno de los métodos más populares para generar PRNG es a partir de registros de desplazamiento realimentados linealmente, o LFSR.

Un **LFSR** de longitud n es un dispositivo formado por n celdas de memoria (o registros).

Cada ciclo de reloj, el estado del LFSR se actualiza, de manera que el contenido de la celda s_i pasa a ser el que había en la celda s_{i-1}.

El nuevo bit de entrada, S_{n+1} se calcula a partir del estado de los registros y del polinomio de conexiones que definen el LFSR:

$$S_{n+1} = c_1 s_n + \cdots + c_n s_1$$

A pesar de que los LFSR son muy fáciles de implementar por lo que respecta al hardware, utilizados directamente también son muy predecibles, lo que limita su uso como PRNG. Para romper la linealidad de los LFSR (y, por lo tanto, hacerlos menos predecibles), se utilizan varias técnicas, como aplicar una función de filtrado no lineal a los bits que se extraen o combinar la salida de diferentes LFRS mediante una función no lineal.

Existen otras maneras de crear PRNG que se basan en las propiedades estadísticas que presentan los sistemas de cifra y las funciones *hash*. Así, por ejemplo, se pueden construir PRNG haciendo llamadas a algunas funciones *hash* (como SHA-1, SHA-256, SHA-386 o SHA-512) con los valores de un contador como entrada, iterando funciones HMAC utilizando alguna de las funciones

hash ya mencionadas o bien cifrando un contador con ciertos algoritmos de cifra de bloque como el AES o el triple-DES.

Cifras de bloque

Una cifra de bloque sobre el alfabeto binario es una proyección biyectiva $E_k : \{0,1\}^n \rightarrow \{0,1\}^n$ indexada por una clave k. Las cifras de bloque permiten cifrar datos de tamaño arbitrario dividiéndolos en bloques de n bits, añadiendo *padding* si es necesario y aplicando la función E_k. Para utilizar cifras de bloque, se define un conjunto de modos de operación que explicita cómo se tienen que combinar los diferentes bloques, a qué valores se tiene que aplicar la función E_k y cómo se tienen que incorporar vectores de inicialización, *nonces* y contadores a los esquemas.

Entre los sistemas de cifra de bloque más conocidos se encuentran el AES y el DES. Como hemos visto, una de las alternativas más directas para obtener herramientas criptográficas para dispositivos con recursos limitados es la implementación eficiente de sistemas ya existentes. En esta línea, se puede encontrar una implementación de la versión serializada del DES que requiere solo 2.310 GE y que utiliza 144 ciclos de reloj para cifrar un bloque de entrada. Sin embargo, esta versión del DES ya no se considera segura, ya que, por un lado, se han encontrado debilidades en el ámbito teórico y, por otro, se han producido ataques distribuidos que han permitido romper claves DES (de cincuenta y seis bits) en menos de un día.

Una de las variantes del DES creadas con el objetivo de incrementar la complejidad de un ataque por fuerza bruta es el Triple DES:

$$3DES_{k_1,k_2,k_3}(m) = E_{k_1}(E^{-1}_{k_2}(E_{k_3}(m))).$$

Por su construcción, el 3DES ofrece compatibilidad con el DES: si las tres claves utilizadas son la misma, entonces 3DES es equivalente a DES ($3DES_{k,k,k}(m) = E_k(E^{-1}_k(E_k(p))) = E_k(p)$). La longitud de la clave puede ser de hasta 168 bits ($56 * 3$). El 3DES se puede implementar con 4.600 GE, lo que requeriría sesenta y dos ciclos de reloj para la operación de cifrado.

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

$$DESX_{k_1,k_2,k_3}(m) = k_3 \oplus E_{k_1}(k_2 \oplus m).$$

DES

El DES (*data encryption standard*) es un algoritmo de cifra inventado en los años setenta. Las claves del DES están formadas por sesenta y cuatro bits, pero solo cincuenta y seis de estos se utilizan efectivamente en el proceso de cifrado/descifrado. Los otros ocho bits solo se usan para llevar a cabo comprobaciones de paridad.

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

La versión original del DES (y, en consecuencia, la del DESX) prevé la utilización de cajas *S* (cajas de sustitución), cuya implementación necesita una cantidad sustancial de memoria (normalmente se implementan como tablas de *lookup* estáticas). Para obtener una versión del DES con menos requerimientos de espacio que la original, se creó una versión optimizada conocida con el nombre de DESL, que reemplaza las ocho cajas *S* originales por una única caja *S* diseñada de nuevo. De la combinación del DESL con el DESX surgió el sistema de cifra DESXL, que se ha implementado con solo 2.169 GE.

El AES es un algoritmo de cifra que ya fue diseñado con la eficiencia como uno de sus requisitos. El AES (128 bits) en modo de solo cifrado se puede implementar utilizando 3.100 GE y necesita 1.044 ciclos de reloj para realizar el cifrado de un bloque de datos.

Cifras de flujo

A diferencia de las cifras de bloque, las cifras de flujo generan una cadena de bits a partir de la clave y realizan la operación de cifrado haciendo una xor del texto claro con la cadena obtenida.

El algoritmo RC4 es una cifra de flujo bastante popular. A pesar de ello, no resulta en absoluto adecuada para sistemas con recursos limitados, ya que contiene una permutación que necesita más de doce mil puertas para ser implementada.

En el caso de las cifras de flujo, la alternativa de diseñar nuevos sistemas que requieran pocas GE es la que ha tenido una mayor aceptación. Así pues, sistemas de cifra como Grain o Trivium se diseñaron específicamente para dispositivos con recursos muy limitados.

El funcionamiento de Trivium se basa en tres LFSR de 93, 84 y 11 bits de longitud. Para generar un bit de salida, se extraen dos bits de cada uno de los LFSR y se realiza una xor de todos ellos. Trivium se puede implementar con 3.090 GE y necesita 176 ciclos de reloj para cifrar 128 bits de datos.

Grain utiliza dos registros de desplazamiento y una función no lineal de salida para operar. Grain puede ser implementado con 3.360 GE, y requiere 104 ciclos de reloj para cifrar 128 bits. Una de las características adicionales de este algoritmo es que permite aumentar su velocidad, aunque hay que pagar un precio por lo que respecta a puertas lógicas necesarias para implementarlo. Así

pues, la implementación más sencilla (y también la más lenta) requiere solo 1.450 GE. El tamaño de la clave es de ochenta bits.

Funciones hash

Una función *hash* H es una función que hace corresponder a un mensaje m de tamaño variable un valor $H(m)$ de tamaño fijo. Hay un conjunto de propiedades que las funciones *hash* deben satisfacer para poder ser consideradas seguras:

1) para cualquier valor y, es difícil encontrar un valor m tal que $H(m) = y$;

2) dado un valor m_1, es difícil encontrar otro valor $m_2 \neq m_1$ tal que $H(m_1) = H(m_2)$;

3) es difícil encontrar dos valores m_1, m_2 con $m_1 \neq m_2$ tales que $H(m_1) = H(m_2)$.

Después de que la popular función de *hash* MD5 se considerase rota, la familia de funciones SHA ha tomado el relevo. Existen implementaciones de SHA-256 con 10.868 GE que requieren 1.128 ciclos de reloj para calcular el *hash* de un bloque de 512 bits. Para SHA-1, las implementaciones requieren algo menos de espacio (8.120 GE). Sin embargo, en ambos casos el número de GE necesarias es muy elevado para usarlas en RFID de gama media-baja.

Criptografía de clave pública

La criptografía de clave pública se basa en el uso de pares de claves que tienen una relación matemática especial. Cada usuario dispone de una clave pública, que es conocida por todo el mundo, y de una clave privada, que solo él conoce. Para cifrar un mensaje m, se aplica una función de cifrado E sobre m utilizando la clave pública del usuario al que va destinado. Para descifrar el mensaje, el usuario destinatario deberá aplicar una función de descifrado D al texto cifrado c valiéndose de su clave privada.

La utilización de criptografía basada en curvas elípticas* para sistemas RFID es un campo en desarrollo. Se estima que una implementación de ECC de 192 bits necesitaría 23.600 GE y más de quinientos mil ciclos de reloj para realizar una operación de multiplicación de puntos.

* En inglés, *eliptic curve cryptography* o ECC.

Protocolos para autenticar

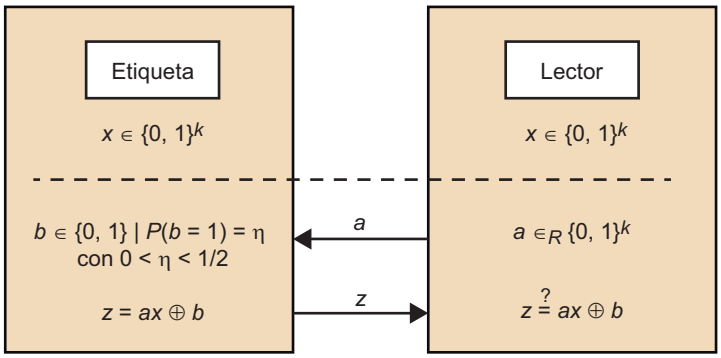
El protocolo de Hopper y Blum (HB) es un esquema que ofrece autenticación sin ofrecer privacidad. El protocolo original, que se conoce como protocolo

HB, solo es seguro ante atacantes pasivos que se limitan a escuchar la comunicación, sin interferir en ella ni modificarla de ninguna forma. La principal ventaja de este protocolo es su gran simplicidad, lo que lo hace más adecuado para dispositivos RFID con pocas GE. Dado un lector y una etiqueta que comparten una tira de k bits secreta, x, el protocolo permite al lector autenticar una etiqueta sin que un atacante que hace una escucha pasiva pueda aprender el valor secreto x. La figura 1 describe los pasos que conforman el protocolo HB.

Figura 1

El protocolo requiere que la etiqueta y el lector compartan un valor secreto x de k bits. El lector (que actúa como verificador) genera un reto aleatorio a también de k bits y lo envía a la etiqueta (el probador). Entonces, la etiqueta calcula el producto escalar entre el valor secreto x y el reto aleatorio a, y cambia el resultado de la operación con una probabilidad η (es decir, selecciona un bit b aleatoriamente, de manera que este sea 1 con probabilidad η y hace una xor del resultado con este bit b). El resultado de esta operación, el valor z, se envía al verificador, que comprobará si es el mismo que el resultado de hacer el producto escalar entre los vectores a y x. Dado que el probador cambiará el resultado con una probabilidad $\eta < 1/2$, este valor será igual con una probabilidad superior a 1/2. Por tanto, el protocolo se repite un número de veces determinado, y la autenticación se da por válida si los valores obtenidos han sido iguales en un número mínimo de repeticiones.

Figura 1. Protocolo HB



Como se puede observar, este protocolo no es seguro ante atacantes activos que pueden hacerse pasar por verificadores y ejecutar el protocolo varias veces, con lo que tendrían en su poder la decisión de los valores de reto a enviados en cada momento. Otra versión de este protocolo, HB+, extiende su funcionalidad para que sea seguro ante atacantes activos que pueden interactuar tanto con la etiqueta como con el lector, aunque no de forma concurrente. Una tercera extensión de este mismo protocolo, llamada $HB^{\#}$, asegura la autenticación correcta ante atacantes activos con acceso concurrente a la etiqueta y al lector.

Protocolos para garantizar la seguridad hacia delante

Como hemos visto, una de las propiedades de seguridad que puede ser interesante garantizar en sistemas RFID es la seguridad hacia delante, que permite evitar que un atacante pueda identificar transacciones pasadas que involucraban a una etiqueta dada si consigue descubrir los datos secretos que contiene

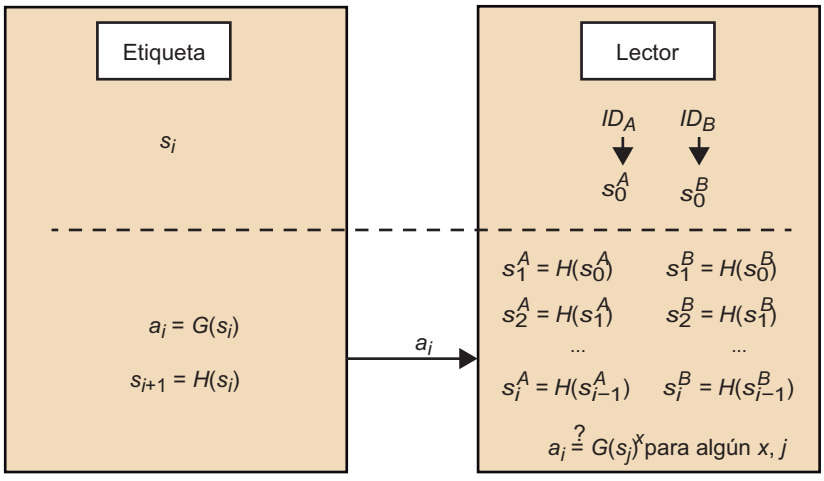
en un instante de tiempo concreto. Una de las alternativas propuestas para conseguir garantizar esta propiedad es el uso de cadenas de funciones *hash*.

El esquema propuesto por Ohkubo, Suzuki y Kinoshita utiliza dos funciones de *hash*, H y G. Cada etiqueta tiene un valor secreto s_i donde i es un contador de transacciones, de manera que el valor secreto cambia en cada nueva transacción. Para identificarse, una etiqueta enviará al lector el valor $a_i = G(s_i)$, que es el resultado de aplicar la función *hash* G al valor secreto que tiene en aquel momento. Entonces, la etiqueta renueva su valor secreto usando la función *hash* H, de manera que $s_{i+1} = H(s_i)$, y a continuación borra el valor secreto s_i. El lector mantiene una base de datos con las correspondencias entre cada identificador de etiqueta y la información secreta inicial s_0 asociada a esta. En el momento en que el lector recibe un valor a_i, comprueba si corresponde a algún valor $G(H^i(s_0))$ para algún s_0 de las etiquetas que conoce, y recupera así el identificador de la etiqueta. Mientras que el número de operaciones que debe realizar la etiqueta es bajo, el lector debe calcular salidas de la función *hash* para cada uno de los valores i y para cada una de las etiquetas que tiene en la base de datos, lo que puede suponer un coste elevado en escenarios determinados.

Ejemplo de autenticación con cadenas *hash*

En el caso que se muestra en la figura 2, el lector solo tiene dos etiquetas en su base de datos. Cuando recibe un valor a_i, el lector puede comprobar a qué etiqueta pertenece calculando todos los valores de la cadena *hash*, utilizando como valor inicial de la cadena el secreto inicial s_i de cada una de las etiquetas que reconoce.

Figura 2. Autenticación con cadenas *hash*



2. Bluetooth

En este apartado describiremos el estándar Bluetooth y los mecanismos de seguridad que prevé. En primer lugar, examinaremos qué clases de dispositivos Bluetooth existen y cuál es la estructura de las redes de dispositivos Bluetooth.

Después repasaremos los diferentes modos de seguridad de Bluetooth y detallaremos algunos de los procesos básicos, como los que permiten autenticar dispositivos; también veremos las diferentes claves que permiten ofrecer seguridad a los dispositivos Bluetooth.

2.1. Descripción de la especificación Bluetooth

Bluetooth es un estándar industrial para comunicaciones inalámbricas baratas de corto alcance. Tiene como principal objetivo sustituir los cables de teclados, ratones y periféricos en general, así como permitir comunicaciones entre dispositivos portátiles. El estándar Bluetooth se desarrolla por medio del Bluetooth Special Interest Group, al que pertenecen compañías del sector de las telecomunicaciones, las redes y la electrónica de consumo.

El origen del nombre Bluetooth

La palabra *Bluetooth* deriva del nombre del rey danés y noruego Harald Blåtand. La traducción directa al inglés de su nombre es Harold Bluetooth. Este rey es conocido por ser un buen comunicador y por unificar a las tribus danesas, noruegas y suecas.

Los dispositivos Bluetooth se clasifican en tres clases, dependiendo de su potencia y de su alcance de comunicación. La tabla 1 muestra las clases disponibles.

Tabla 1. Dispositivos Bluetooth

Clase	Potencia máxima	Rango aproximado
1	100 mW	100 m
2	2,5 mW	10 m
3	1 mW	1 m

El estándar Bluetooth está enfocado a dispositivos que disponen de batería. Así pues, tanto el rendimiento de las aplicaciones que se ejecuten como la seguridad de los dispositivos Bluetooth estarán limitados, por un lado, por el consumo energético y, por otro, por su coste.

Las redes Bluetooth se crean siguiendo una topología de estrella con ocho miembros como máximo, uno de los cuales actúa como maestro. Los demás miembros adoptan el papel de esclavos. Cada una de estas redes se conoce con el nombre de *piconet*. Dada la topología de las *piconets*, las comunicaciones con Bluetooth siempre implican un dispositivo maestro, lo cual permite

que se produzca comunicación tanto desde un maestro hacia un esclavo como desde un esclavo hacia un maestro. Si dos dispositivos esclavos quieren comunicarse entre sí, necesitan que un dispositivo maestro actúe como intermediario. Para optimizar esta comunicación, uno de los dispositivos esclavos implicados puede decidir crear una nueva *piconet* y convertirse en su maestro, ya que así puede comunicarse directamente con el otro dispositivo. Un dispositivo puede pertenecer a más de una *piconet* al mismo tiempo, pero solo puede ser maestro de una como máximo.

2.2. Seguridad en dispositivos Bluetooth

La arquitectura de seguridad de Bluetooth permite obtener las propiedades de confidencialidad, integridad y autenticación. El hecho de que los sistemas de cifrado que se utilizan sean de clave simétrica implica que no se puede conseguir la propiedad de no repudio.

Los dispositivos Bluetooth pueden operar en tres modos de seguridad:

- **Modo de seguridad 1**. Este es el modo más inseguro, ya que no incorpora ningún mecanismo de seguridad. Así permite la conexión entre cualquier dispositivo o aplicación.
- **Modo de seguridad 2** (nivel de servicio). Este es el modo de seguridad que actúa en el ámbito del servicio. El dispositivo deja realizar conexiones pero después aplica la seguridad y restringe el uso de las aplicaciones. En este caso, la política de seguridad se aplica después de la conexión.
- **Modo de seguridad 3** (nivel de enlace). Este es el modo de seguridad que actúa directamente en los enlaces. Por tanto, la política de seguridad se aplica ya antes de hacer la conexión.

Poca seguridad

De hecho, el modo de seguridad 1 no se debería considerar de seguridad, pero las especificaciones de la arquitectura así lo indican.

A continuación pasaremos a detallar los diferentes modos de seguridad de la arquitectura Bluetooth. Nos centraremos solo en los modos de seguridad 2 y 3, es decir, el que opera en el ámbito del servicio y el que lo hace en los enlaces, ya que el modo de seguridad 1 no tiene ningún interés porque no ofrece ningún tipo de seguridad.

2.2.1. Modo de seguridad 2: nivel de servicio

El modo de seguridad que actúa en el ámbito de servicio tiene una seguridad más débil que el modo de seguridad 3. Esto se debe al hecho de que las res-

tricciones se aplican cuando la comunicación entre los dispositivos ya ha sido efectuada. La justificación para utilizar este nivel de seguridad en lugar del nivel 3 reside en que si se restringen las conexiones en los enlaces no es posible diseñar aplicaciones más abiertas, como el intercambio de tarjetas de negocio o la consulta de los servicios ofrecidos por un dispositivo.

El componente clave que implementa la política de seguridad es el **gestor de seguridad***.

*** En inglés, *security manager*.**

El **gestor de seguridad** se encarga, entre otras tareas, de:

- almacenar información de seguridad de los servicios;
- almacenar información de seguridad de los dispositivos;
- aceptar o rechazar las peticiones de acceso generadas por los protocolos o las aplicaciones;
- forzar la autenticación o cifrado antes de conectar con la aplicación, y
- pedir el PIN al usuario o a la aplicación correspondiente.

El PIN

Los dispositivos Bluetooth disponen de un PIN para autenticar a los usuarios. Dicho PIN tiene una longitud de entre uno y dieciséis bytes (normalmente, cuatro dígitos) y el usuario puede cambiar su valor cuando desee.

La política de seguridad que interpreta el gestor de seguridad se basa en la información almacenada en dos bases de datos: la base de datos de los dispositivos y la base de datos de los servicios.

La base de datos de los dispositivos mantiene información de los requisitos de seguridad de los dispositivos en función de la confianza que se tiene sobre dichos dispositivos. Así, se especifican dos niveles de confianza:

- **Dispositivos de confianza**. Son aquellos que han sido previamente autenticados, hay una clave de enlace almacenada y en la base de datos están marcados como dispositivos de confianza.

- **Dispositivos *untrusted***. Son aquellos que han sido previament autenticados, hay una clave de enlace almacenada y en la base de datos no están marcados como dispositivos de confianza. Normalmente, se aplica este nivel de seguridad a los dispositivos con los que no se tiene una relación permanente.

La base de datos de los dispositivos puede estar especificada para cualquier servicio o se puede mantener separada para cada servicio o para un conjunto de servicios.

La base de datos de los servicios especifica las necesidades de seguridad de los diferentes servicios. Así, los servicios se dividen en:

- **Servicios abiertos**: aquellos en los que no se restringe el acceso y para los que no se requiere ningún tipo de información.

- **Servicios con autenticación**: aquellos en los que los diferentes dispositivos tienen que autenticarse para acceder a ellos.

- **Servicios con autenticación y autorización**: son los servicios más restringidos que hay y requieren tanto un proceso de autenticación como un proceso de autorización. Es decir, no es suficiente con el hecho de decir quién eres, sino que además tienes que estar autorizado.

Ejemplo

En muchas webs solo hay que registrarse para obtener un servicio. Este sería el caso de servicio con autenticación. En cambio, en otras webs, como por ejemplo las de las entidades bancarias, aparte de identificarte, es preciso estar autorizado para poder operar.

La base de datos de servicios también especifica si un servicio requiere que los datos que se intercambien estén cifrados o sean texto claro.

El gestor de seguridad, situado desde un punto de vista conceptual entre el nivel de enlace y el nivel de aplicación, implementa la política de seguridad por la vía de peticiones y respuestas de los dos niveles y basándose en las informaciones incluidas en las dos bases de datos mencionadas. Además, el gestor de seguridad se encargará también de la petición y comprobación del PIN del usuario para realizar los procesos de autenticación.

2.2.2. Modo de seguridad 3: nivel de enlace

El modo de seguridad en el nivel de enlace es el sistema más seguro especificado en la arquitectura Bluetooth, ya que las restricciones de seguridad se aplican antes de la conexión entre los dispositivos y, de esta forma, se puede minimizar el riesgo de ataques de dispositivos ya conectados. Este modo de seguridad permite obtener las propiedades de autenticación, confidencialidad e integridad.

Para obtener estas propiedades, la arquitectura Bluetooth dispone de las siguientes entidades:

- **Dirección del dispositivo Bluetooth (BD_ADDR)**. Cada dispositivo Bluetooth tiene una dirección única de cuarenta y ocho bits. Esta dirección equivale a las direcciones MAC de las tarjetas de red.

- **Clave de enlace*** (K_e). Para llevar a cabo el proceso de autenticación de los dispositivos, la arquitectura Bluetooth utiliza una clave de 128 bits.

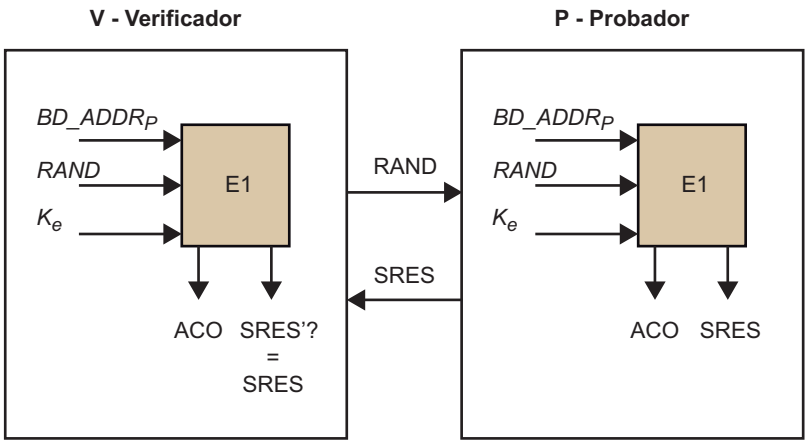
*** En inglés, *link key*.**

- **Clave de cifrado** (K_x). La confidencialidad en la transmisión de los datos se obtiene por medio de una clave de cifrado, diferente de la clave de enlace, que puede tener una longitud de entre 8 y 128 bits. Esta clave también se utiliza para obtener la propiedad de integridad, propiedad que queda incluida cuando se obtiene la confidencialidad.

- **Número aleatorio (RAND)**. Los dispositivos Bluetooth disponen de un generador seudoaleatorio que les permite obtener, cuando es preciso, diferentes valores seudoaleatorios de 128 bits.

Basándose en las entidades que acabamos de describir, el proceso de autenticación de un dispositivo con el modo de seguridad 3 se describe en la figura 3.

Figura 3. Proceso de autenticación



El proceso de autenticación sigue el modelo reto-respuesta. Los pasos que se siguen en el esquema de la figura 3 son los siguientes:

1) El verificador envía al probador un reto en forma de valor aleatorio RAND. Utilizando la función E1 con las entradas RAND, la dirección de P, BD_ADDRP y la clave de enlace K_e, el verificador obtiene SRES'. Además, también obtiene un valor ACO* que se utiliza después para obtener la clave de cifrado.

2) El probador, con el valor RAND y el resto de los valores, calcula una respuesta SRES al mismo tiempo que obtiene el mismo valor ACO que el verificador.

3) El probador envía la respuesta al verificador, que comprueba que SRES' = SRES y, por tanto, valida la identidad del probador.

Nomenclatura

Denominaremos al dispositivo que quiere autenticarse probador, mientras que el dispositivo que valida la autenticación será el verificador.

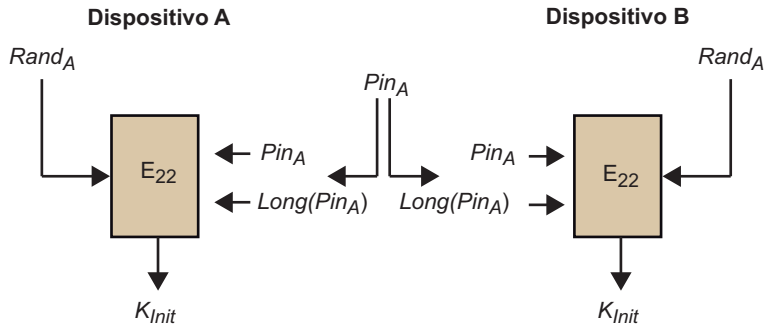
* Del inglés, *authenticated Ciphering Offset*.

Para obtener autenticación mutua, se repite el proceso intercambiando el rol que representan las dos partes en el protocolo anterior.

Como ya hemos mencionado, un proceso de autenticación reto-respuesta como el descrito requiere que los dos interlocutores conozcan un valor antes del proceso de autenticación, en este caso la clave de enlace, K_e. La arquitectura Bluetooth describe diferentes tipos de clave de enlace que se pueden utilizar en función de diferentes supuestos. En concreto, se describen cuatro tipos de claves de enlace:

1) Clave de inicialización K_{Init}. Se utiliza para obtener después la clave de enlace en caso de que los dos dispositivos no se hayan comunicado previamente y, por tanto, no dispongan de ningún valor intercambiado. Esta clave se genera (figura 4) a partir del PIN del dispositivo que se quiere autenticar, y de un valor aleatorio generado por el mismo dispositivo que inicia el proceso de autenticación.

Figura 4. Generación de la clave de inicialización



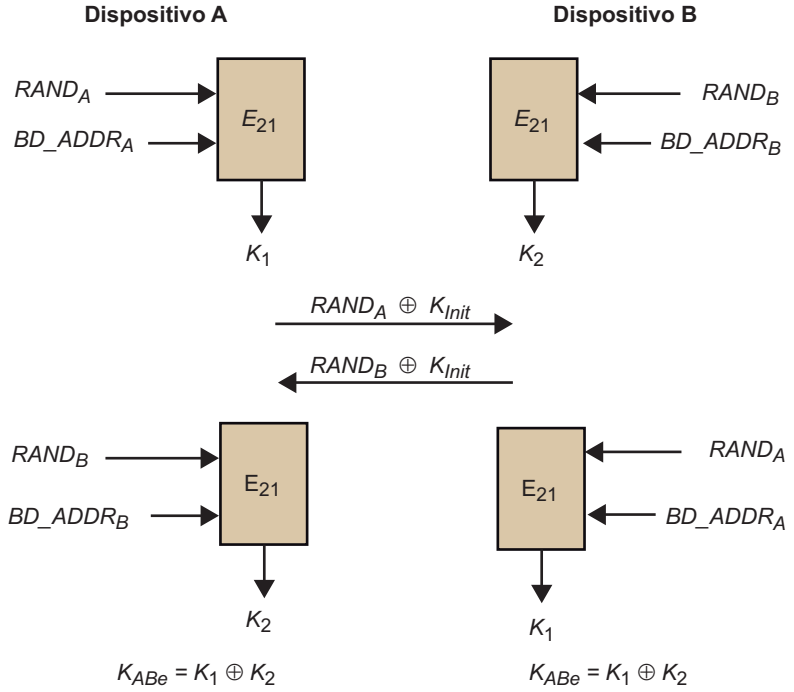
Tal como muestra el gráfico, se utiliza el algoritmo E_2 en modo 2 (denotado por E_{22}) para generar la clave de inicialización a partir de un valor aleatorio, del PIN del dispositivo A (PIN_A) y de la longitud de dicho PIN. Fijaos en que el valor aleatorio RAND lo aporta el dispositivo que se quiere autenticar (y lo envía como texto claro al otro dispositivo) y es necesario que el usuario del dispositivo que se autentica introduzca también su PIN en el otro dispositivo al que se quiere conectar. De este modo, al final del proceso ambos dispositivos disponen de una clave compartida K_{Init}. En principio, dicha clave no se utiliza como clave de enlace, sino que sirve para proteger la información que los dos dispositivos se intercambiarán para fijar la clave de enlace.

2) Clave de dispositivo K_D. La clave de dispositivo se genera con la instalación del dispositivo Bluetooth a partir de la dirección del dispositivo BD_ADDR y un valor aleatorio utilizando el algoritmo E_2 en modo 1 (denotado por E_{21}). Esta clave no se suele cambiar nunca y se guarda en memoria no volátil. A diferencia de la clave de inicialización, la clave de dispositivo sí que se puede emplear como clave de enlace. Se utiliza cuando las restricciones de memoria de alguno de los dispositivos son muy grandes y no se dispone de más espacio para almacenar otra clave de enlace diferente de la de dispositivo. En este caso, si el dispositivo A es el que tiene restricciones de espacio, A enviará su clave de dispositivo K_A al dispositivo B. Para que la transmisión de la clave no pueda ser interceptada, A enviará a B el valor $K_A \oplus K_{Init}$. Cuando B obtiene dicho valor, le volverá a sumar K_{Init}, que ya conoce, y obtendrá el valor K_A.

3) Clave combinación K_{AB}. Se trata de una clave de enlace que se obtiene a partir de la información aportada por los dos dispositivos, A y B, a diferencia de la clave de dispositivo, que cuando se utiliza como clave de enlace solo depende de uno de los dispositivos. Dado que se crea a partir de los dos dispositivos, este tipo de claves se genera para cada par de dispositivos, y ofrece mucha más seguridad que el uso de la clave de dispositivo como clave de enlace. En la figura 5 podéis observar el proceso de generación de la clave combinación.

A partir de un valor aleatorio, $RAND_A$, y su dirección, BD_ADDR_AA, el dispositivo A utiliza el algoritmo E_2 en modo 1 (denotado por E_{21}) para generar el

Figura 5. Generación de la clave combinación K_{AB}



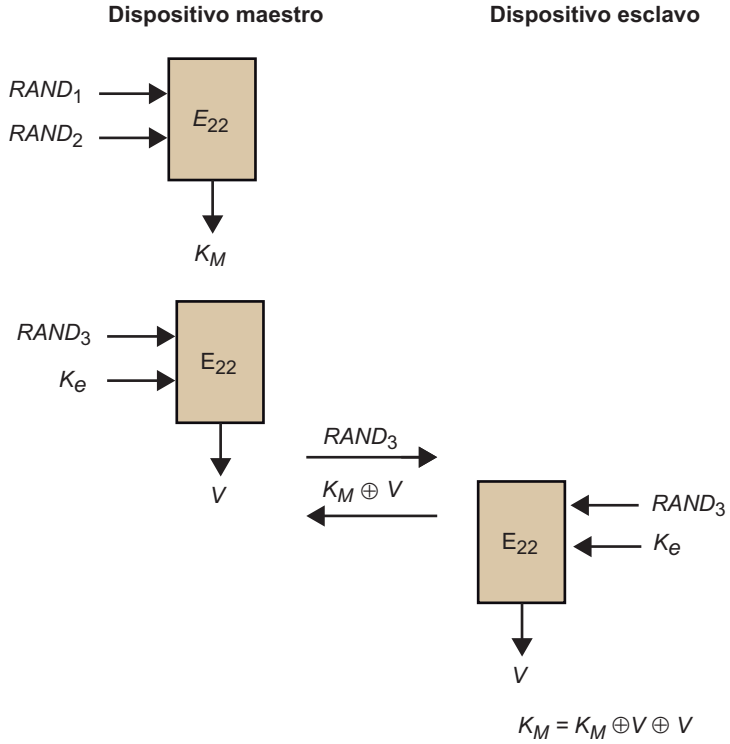
valor K_1. El dispositivo B sigue el mismo proceso para obtener K_2. Ambos dispositivos se intercambian los valores aleatorios RAND_A y RAND_B protegidos por la clave de inicialización K_{Init}. En este punto, cada dispositivo está en condiciones de poder generar el valor K_i del otro dispositivo. De este modo, los dos dispositivos pueden establecer una clave de enlace compartida que será el valor $K_1 \oplus K_2$.

4) Clave maestra K_M. La clave maestra es una clave temporal que se utiliza en una red Bluetooth con más de dos dispositivos conectados cuando el dispositivo maestro quiere transmitir de forma simultánea a los demás dispositivos. Esta clave maestra sustituye cada una de las claves de enlace K_e que el maestro compartía con cada uno de los dispositivos. El proceso de generación de esta clave maestra se describe en la figura 6.

El dispositivo maestro utiliza dos valores aleatorios y el algoritmo E_2 en modo 2 (denotado por E_{22}) para generar la clave maestra. Además, utiliza otro valor aleatorio y la clave de enlace para obtener un valor V que servirá para proteger la transmisión de la clave maestra K_M. El dispositivo maestro envía RAND_3 y $K_M \oplus V$ al dispositivo esclavo. Este, por medio del valor RAND_3 y la clave de enlace, que compartía con el dispositivo maestro, puede generar el valor V que le permitirá obtener el valor de la clave K_M.

Hasta este punto hemos descrito el proceso de autenticación que se realiza a partir de la clave de enlace, que, como hemos visto, puede ser de diferentes tipos. Pasemos ahora a ver cómo el modo de seguridad 3 de la arquitectura Bluetooth nos ofrece la propiedad de confidencialidad.

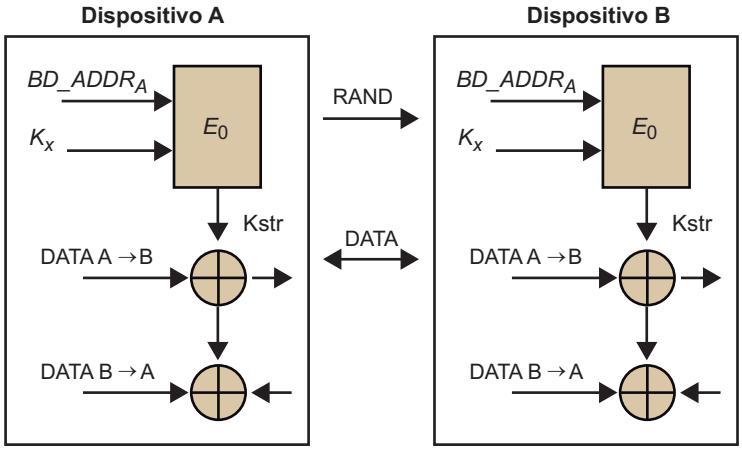
Figura 6. Generación de la clave maestra K_M



Para obtener confidencialidad en la comunicación entre dos dispositivos, es preciso cifrar los datos. Para hacerlo, Bluetooth utiliza un criptosistema de cifrado en flujo llamado E_0.

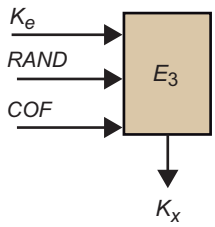
Tal como se muestra en la figura 7, a partir de la dirección de A, BD_ADDR$_A$, y la clave de cifrado, K_x, el criptosistema genera una secuencia aleatoria, denotada en la figura por Kstr, que se utiliza para cifrar o para descifrar la información que circula entre los dos dispositivos. De este modo, a través de los datos que se envían cifrados, se obtienen las propiedades de integridad y confidencialidad.

Figura 7. Generación de la clave de cifrado K_x



Como vemos, la clave de cifrado es el único elemento secreto que comparten los dos dispositivos. Este valor se genera a partir de un valor aleatorio, la clave de enlace, y un tercer valor llamado *ciphering offset number* (COF), tal como se muestra en la figura 8.

Figura 8. Generación del COF



El valor COF es, en general, el valor ACO intercambiado durante el proceso de autenticación. La longitud de la clave de cifrado obtenida varía de 8 a 128 bits, y en cada caso se fija por medio de una negociación entre los dos dispositivos. En cada dispositivo hay un parámetro que define la máxima longitud de clave permitida. Cada aplicación tiene también un mínimo aceptable de longitud de clave.

3. ZigBee

En este apartado describiremos algunas de las funcionalidades que ofrece el estándar ZigBee. En primer lugar, expondremos las principales características de las redes ZigBee y su arquitectura. A continuación, repasaremos los dispositivos que reconoce la especificación de ZigBee y las posibles configuraciones que permiten agruparlos.

Después de la descripción de la especificación de ZigBee, nos centraremos en los aspectos de seguridad que prevé: analizaremos los diferentes modos de seguridad, estudiaremos las diferentes claves que utilizan los dispositivos ZigBee y su función, y detallaremos los diferentes servicios de seguridad que ZigBee especifica para cada una de las capas.

3.1. Descripción de la especificación ZigBee

El origen del nombre ZigBee

Se dice que el nombre *ZigBee* viene de un paralelismo con la forma en que se comunican las abejas, realizando una especie de danza para comunicar información importante a otros miembros de la colmena. Esta danza es la que los diseñadores de ZigBee intentan emular con este protocolo, al permitir a un conjunto de dispositivos sencillos comunicarse y trabajar juntos para llevar a cabo tareas complejas.

ZigBee es el nombre con el que se conoce la especificación de un conjunto de protocolos para redes WPAN basadas en el estándar IEEE 802. ZigBee está diseñado para ser utilizado en dispositivos de radiofrecuencia de corto alcance que disponen de baja potencia y que requieren una tasa de transmisión de datos baja. ZigBee se diferencia así de otros protocolos para WPAN, como por ejemplo Bluetooth, que permiten alcanzar tasas de transmisión de datos mucho más elevadas pero que tienen un consumo energético también mucho más alto.

Las principales características de ZigBee son:

- **Bajo consumo**. Los dispositivos ZigBee pueden estar funcionando durante años con el mismo par de baterías AA. Esto es posible porque no es preciso que los dispositivos de una red ZigBee estén constantemente enviando mensajes (pueden estar dormidos).

- **Estándar abierto**. La especificación de ZigBee es un estándar abierto, lo que permite asegurar, por un lado, la interoperabilidad de dispositivos y, por otro, el acceso libre a la especificación. A pesar de todo, la licencia de ZigBee permite utilizar el estándar libremente solo para aplicaciones no comerciales, lo que supone conflictos con ciertas licencias como la GPL.

- **Seguridad**. ZigBee permite añadir seguridad a las comunicaciones mediante una serie de servicios que permiten la gestión de claves criptográficas, el

cifrado de los mensajes transmitidos y la autenticación de los dispositivos, entre otras características que veremos a lo largo de este apartado.

- **Bajo coste de implementación**. Manteniendo una especificación sencilla, ZigBee intenta minimizar el coste de crear nuevas aplicaciones que la utilicen.

- **Baja velocidad de transmisión**. Para poder ofrecer dispositivos de bajo consumo y de bajo coste, ZigBee no permite alcanzar velocidades de transmisión de datos elevadas, y limita su uso a aplicaciones poco exigentes en este ámbito.

ZigBee opera en las bandas de frecuencia de 868 MHz (en Europa), 915 MHz (en América) y 2,4 GHz (en el ámbito global). La tasa de transmisión de datos máxima que puede alcanzar es de 250 kb/s cuando opera a 2,4 GHz (dieciséis canales), 40 kb/s a 915 MHz (diez canales) y 20 kb/s a 868 MHz (un canal). La distancia máxima a la que dos dispositivos ZigBee pueden comunicarse es muy variable y depende de la potencia de salida y de las condiciones ambientales, pudiendo variar desde diez metros hasta mil seiscientos.

La especificación de ZigBee está desarrollada por la ZigBee Alliance, una asociación de empresas, universidades y gobiernos fundada en el 2002 con el objetivo de desarrollar estándares y productos para redes inalámbricas de bajo consumo. La ZigBee Alliance se encarga tanto de mantener y actualizar la especificación de ZigBee como de promocionar y fomentar su uso.

Dado que ZigBee es una especificación de protocolos de alto nivel, se puede utilizar en multitud de escenarios diferentes. Así, por ejemplo, podemos encontrar dispositivos ZigBee utilizados en domótica y automatización, en aplicaciones comerciales, en dispositivos de *fitness* o en electrónica de consumo, por poner varios ejemplos.

Evolución de la especificación ZigBee

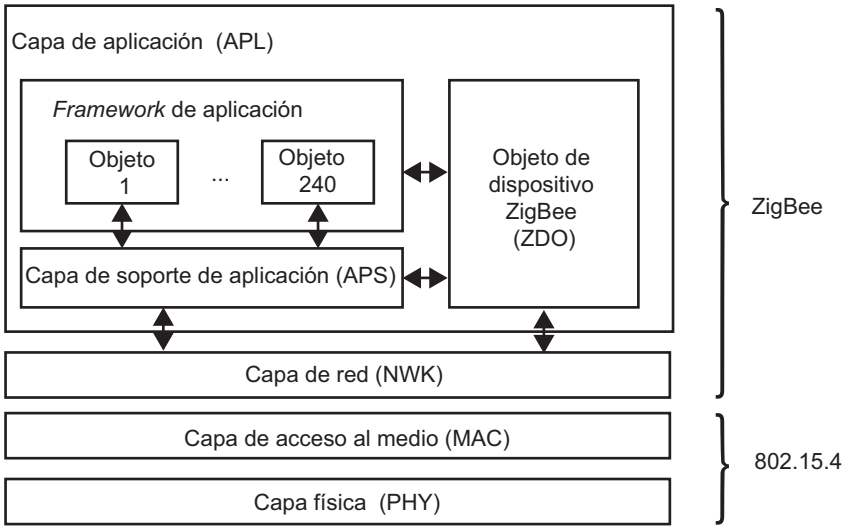
La especificación de ZigBee ha ido cambiando con el tiempo, implementando mejoras en cada nueva versión. La primera versión de la especificación fue aprobada en diciembre del 2004. Esta primera versión se conoce como ZigBee v1.0 o como ZigBee 2004, y actualmente se considera obsoleta. Dos años después, en diciembre del 2006, se publicó una nueva versión (ZigBee 2006). En octubre del 2007, se anunció otra revisión. Esta última especificación contiene dos perfiles de pila diferentes, ZigBee 2007 y Pro, que presentan algunas diferencias en relación con las características disponibles y, en consecuencia, con la memoria necesaria para ejecutarlas.

3.1.1. Arquitectura

ZigBee adopta la especificación de la capa física y de la subcapa de control de acceso al medio (MAC) del estándar IEEE 802.15.4. De este modo, ZigBee proporciona una especificación para la capa de red (NWK) y un *framework* para la capa de aplicación (APL), y delega en el estándar IEEE 802.15.4 toda la especificación de bajo nivel.

Como se puede apreciar en la figura 9, por lo que respecta a la aplicación (APL), ZigBee ofrece una subcapa de soporte a las aplicaciones (APS), unos objetos de dispositivo ZigBee (ZDO) y un *framework* de aplicación.

Figura 9. Modelo de capas ZigBee



La subcapa APS es una interfaz entre la capa de red (NWK) y la capa de aplicación (APL) que ofrece una serie de servicios disponibles tanto para las aplicaciones como para el ZDO. Estos servicios incluyen, entre otros, sistemas de emparejamiento de dispositivos, fragmentación de mensajes, generación de PDU de aplicación y servicios de seguridad.

Ved también

Analizaremos los servicios de seguridad con más detalle en el subapartado 3.2.3.

Los objetos ZDO proporcionan una interfaz entre los objetos de aplicación, el perfil del dispositivo y la subcapa APS. El ZDO es el encargado de inicializar la subcapa de aplicación (APS) y la capa de red (NWK). Además, también ofrece servicios de gestión de la red como el descubrimiento de otros dispositivos o el descubrimiento de servicios.

PDU

PDU, o *protocol data unit*, es el nombre que recibe el conjunto de la información de control del protocolo y de los datos de usuario en una capa concreta.

El *framework* de aplicación puede contener hasta doscientos cuarenta objetos de aplicación, es decir, de módulos definidos por el usuario que forman parte de la aplicación ZigBee.

3.1.2. Tipos de dispositivos y topologías

Como hemos visto, ZigBee adopta la especificación del IEEE 802.15.4 para la capa de control de acceso al medio. Esta especificación define dos tipos de dispositivos:

- **Dispositivos con funcionalidad completa**. Pueden realizar cualquier papel dentro de la red, es decir, pueden funcionar como coordinadores, direccionadores o dispositivos finales. Además, pueden comunicarse con cualquier otro dispositivo de la red.

- **Dispositivos con funcionalidad reducida**. Solo pueden actuar como dispositivos finales y solo pueden interaccionar con un único dispositivo FFD.

ZigBee identifica, por lo que respecta a la red, tres tipos de dispositivos diferentes:

- **Dispositivo final**. Se trata del dispositivo más sencillo, normalmente conectado a sensores, que corresponde o bien a un RFD o bien a un FFD que actúa como dispositivo simple.
- **Direccionador**. Es un FFD con capacidades de direccionamiento de red.
- **Coordinador ZigBee**. Es un FFD que coordina toda la red.

FFD y RFD

En inglés, llamamos *full-function devices* o FFD a los dispositivos de funcionalidad completa y *reduced-function devices* o RFD a los dispositivos de funcionalidad reducida.

ZE, ZR y ZC

En inglés, llamamos *ZigBee end device* o ZE al dispositivo final, *ZigBee router* o ZR al dispositivo direccionador y *ZigBee coordinator* o ZC al dispositivo coordinador.

Teniendo en cuenta los papeles de cada dispositivo en la red y sus interconexiones, ZigBee permite estructurar los nodos desplegados en tres topologías diferentes:

- **Estrella**. Se trata de la topología más sencilla, formada por un único dispositivo coordinador conectado a múltiples dispositivos finales. Con esta topología, los dispositivos finales no pueden comunicarse directamente. El coordinador es el responsable de gestionar todas las comunicaciones, así como de inicializar y mantener todos los dispositivos de la red.
- **Árbol**. Es una topología jerárquica, en la que cada dispositivo tiene un único padre (excepto el nodo raíz, que no tiene ninguno). El nodo raíz es el dispositivo coordinador, que se encarga de la inicialización de la red. La red se puede extender añadiendo dispositivos direccionadores, que se encargan de transportar los datos y los mensajes de control utilizando una estrategia jerárquica.
- **Malla**. Se trata de una topología no jerárquica, en la que cada dispositivo puede intentar comunicarse con cualquier otro dispositivo de la red, ya sea directamente, ya sea por medio de algún dispositivo direccionador. Las rutas entre nodos se crean bajo demanda y pueden ser modificadas dinámicamente, lo que permite a esta topología adaptarse a los cambios de la red.

3.2. Seguridad en dispositivos ZigBee

La seguridad de ZigBee se basa en un modelo de confianza abierta en el que las diferentes aplicaciones que corren en un mismo dispositivo, así como las dife-

rentes capas de la pila de comunicaciones, confían unas en otras. Este modelo de confianza permite que tanto las diferentes aplicaciones como las diferentes capas compartan las claves, con lo que se consigue ahorrar recursos, normalmente escasos en redes de sensores. El uso de este modelo implica que solo existe protección en el ámbito criptográfico entre los diferentes dispositivos de la red.

Otras decisiones de diseño de la arquitectura de ZigBee también afectan a la seguridad, como por ejemplo el principio de que cada capa es responsable de la seguridad de los mensajes que se originan en ella.

La especificación de ZigBee proporciona herramientas para garantizar la autenticación, la confidencialidad y la integridad de los datos transmitidos. Además, también ofrece herramientas para garantizar el frescor*, es decir, para asegurar que un atacante no podrá reutilizar paquetes capturados durante una comunicación válida.

*** En inglés, *freshness*.**

La especificación de ZigBee define la existencia de un dispositivo especial en cada red llamado centro de confianza*, que goza de la confianza de todos los dispositivos de aquella red.

***En inglés, *trust center*.**

El centro de confianza se puede utilizar para la distribución de claves y puede operar en dos modos diferentes:

- **Modo de alta seguridad**. Este modo está diseñado para aplicaciones comerciales que necesiten un alto nivel de seguridad. Cuando el centro de confianza se encuentra configurado en este modo, debe mantener una lista con todos los dispositivos y todas las claves necesarias para asegurar que se cumplen las políticas de renovación de claves y de admisión en la red. Además, cuando se está operando en este modo, se verifica el frescor de todas las tramas entrantes, y se asegura así que no son tramas duplicadas.

- **Modo de seguridad estándar**. Este modo está diseñado para aplicaciones residenciales que necesiten un nivel de seguridad bajo. Cuando el centro de confianza se encuentra configurado en este modo, no es necesario que mantenga todas las claves (solo la clave de red, que describiremos más adelante), aunque sí que debe controlar las políticas de admisión de la red. De este modo, se consigue que la memoria necesaria para que opere el centro de confianza no crezca con el número de dispositivos de la red, como sucedía en el modo de alta seguridad.

3.2.1. Claves

ZigBee utiliza claves simétricas para establecer comunicaciones seguras. La seguridad de las comunicaciones depende de la correcta inicialización e instalación de dichas claves. La arquitectura de seguridad se vale de tres claves diferentes para ofrecer seguridad:

- **Clave de enlace**. Se trata de una clave de 128 bits solo compartida por dos dispositivos que se utiliza para asegurar la comunicación *unicast* entre dos entidades APL. Un dispositivo puede adquirir una clave de enlace utilizando los servicios de la subcapa APS de transporte de clave o de establecimiento de clave, o bien por medio de la preinstalación de la clave (por ejemplo, en la fábrica). La clave de enlace también se utiliza para generar claves derivadas para diferentes servicios de la red usando funciones de un solo sentido. De este modo, se consigue utilizar claves independientes para ejecutar diferentes protocolos de seguridad, y se evitan interacciones no deseadas.

- **Clave maestra**. La clave maestra se emplea en el protocolo de establecimiento de claves simétricas (por ejemplo, para generar claves de enlace). Un dispositivo puede adquirir la clave maestra por medio del servicio de transporte de clave, mediante preinstalación o bien a partir de alguna información que proporciona el usuario (como una contraseña, por ejemplo).

- **Clave de red**. Es una clave de 128 bits compartida entre todos los dispositivos de la red que se utiliza tanto para enviar mensajes de *broadcast* desde la subcapa de aplicación (APS) como para enviar mensajes desde la capa de red (NWK). Un dispositivo debe adquirir una clave de red o bien mediante el servicio de transporte de la clave o bien mediante preinstalación.

Los diferentes métodos que permiten obtener cada una de las claves son servicios ofrecidos por la capa de aplicación de ZigBee y, por tanto, se encuentran descritos en el subapartado 3.2.3.

La clave de enlace y la clave maestra solo son accesibles desde la subcapa APS, mientras que la clave de red se encuentra disponible tanto desde la capa APL como desde la capa NTW.

3.2.2. Seguridad en la capa de red

Los mensajes ZigBee se protegen criptográficamente en la capa de red cuando se originan en esta capa (según el principio de que cada capa es responsable de la seguridad de los mensajes que se originan en ella) o bien cuando se originan

en una capa superior y se especifica explícitamente que se tienen que proteger en el ámbito de la red.

La tabla 2 muestra los campos que contiene una trama ZigBee de la capa de red. Como se puede apreciar, aparte de las cabeceras propias de cada capa, la capa de red añade una cabecera auxiliar y un campo de integridad, que permiten incluir la información necesaria para gestionar la seguridad del contenido.

Tabla 2. Trama de red

SYNC	PHY HDR	MAC HDR	NWK HDR	Auxiliary	ENC NWK Payload	MIC

La cabecera auxiliar contiene:

- **Campo de control de seguridad**:

– Nivel de seguridad. Indica los parámetros de seguridad que se han utilizado en aquella trama. La tabla 3 muestra todos los niveles de seguridad ofrecidos, así como las características de cada uno de ellos.

Tabla 3. Niveles de seguridad disponibles en las capas de red y aplicación

Identificador nivel	Seguridad	Cifrado	Integridad
0	Ninguno	No	No
1	MIC-32	No	Sí
2	MIC-64	No	Sí
3	MIC-128	No	Sí
4	ENC	Sí	No
5	ENC-MIC-32	Sí	Sí
6	ENC-MIC-64	Sí	Sí
7	ENC-MIC-128	Sí	Sí

Tabla 3

Fijaos en que los niveles 0 y 4 no aseguran de ninguna manera la integridad de los mensajes enviados y que los niveles de 0 a 3 no proporcionan confidencialidad.

– Identificador de la clave. Contiene dos bits que identifican el tipo de clave que se ha utilizado.

– *Nonce* extendida. Se trata de un bit que indica si se incluye el campo de dirección de origen o si se omite.

- **Contador de tramas**. Permite, por un lado, asegurar el frescor de la trama y, por otro, evitar que se procesen tramas por duplicado.

- **Dirección de origen**. Si se ha especificado anteriormente que se incluiría la dirección de origen (en el campo *extended nonce*), entonces contiene la dirección del dispositivo responsable de añadir seguridad a la trama.

- **Número de secuencia de la clave**. Si se ha especificado una clave de red (en el campo identificador de la clave), contiene el número de secuencia de la clave de red.

La especificación de ZigBee exige el uso de AES (*advanced encryption standard*) como algortimo de cifrado. Dado que AES es un algoritmo de cifrado en bloque, la especificación también fija cómo hay que utilizarlo para proteger los mensajes, es decir, el modo de operación. ZigBee utiliza AES en modo CCM* (*counter with cipher block chaining message authentication code*), un modo que combina cifrado y autenticación, de manera que tanto el mensaje cifrado como el valor del campo de integridad son el resultado de aplicar AES-CCM* sobre la carga de la trama de red. El modo CCM* también permite operar en modo solo de cifrado (sin autenticación) o en modo solo de autenticación (sin cifrado).

AES

El NIST aprobó el algoritmo de cifrado Rijndael como AES el 26 de mayo del 2002. Rijndael fue creado por dos criptólogos, Joan Daemen y Vincent Rijmen.

Si el nivel de seguridad exige cifrado, la carga de la trama se cifra con AES en CCM*. De este modo, si un atacante observa el tráfico entre dispositivos, no será capaz de leer el contenido, y así se garantizará la confidencialidad de la información transmitida.

El modo CCM*

El modo CCM* coincide con la especificación de CCM para mensajes que requieran autenticación y, posiblemente, cifrado. CCM* añade la alternativa de permitir solo cifrar, sin garantizar la autenticidad del mensaje.

Si el nivel de seguridad exige integridad, el campo MIC* contiene una etiqueta calculada a partir de la carga de la trama y de la clave, de una manera conocida tanto por el emisor como por el receptor. Cuando el receptor recibe la trama, calculará también el valor MIC a partir del contenido recibido y de la clave que comparte con el emisor. Si durante el transporte del mensaje el contenido de la trama ha sido alterado, entonces el MIC calculado por el receptor será diferente del MIC contenido en la trama, y así se puede detectar que ha habido una modificación del mensaje. Dado que es necesario conocer la clave para calcular el MIC, el atacante que modifique el mensaje tampoco podrá modificar el MIC de forma adecuada, ya que desconoce la clave utilizada. El nivel de seguridad marca la longitud del campo de MIC (0, 32, 64 o 128 bits), que determina la probabilidad de que un valor elegido al azar coincida con el valor correcto de MIC.

*** Del inglés, *message integrity code*.**

MIC

El MIC también se conoce con el nombre de MAC (*message authentication code*). En la especificación de ZigBee, se utiliza MIC en lugar de MAC para evitar la confusión que podría provocar el hecho de interpretar MAC como *medium access control*, una subcapa de la capa de enlace especificada por el modelo OSI.

Aparte de los datos y de la clave, el modo CCM* requiere el uso de un valor de *nonce* para operar. Dada una misma clave, el valor de *nonce* será único para cada mensaje que se envíe. El valor de *nonce* que se utiliza en CCM* se construye concatenando los valores del campo de control de seguridad, el contador de trama y la dirección de origen. Fijaos en que el valor del *nonce* cambia para cada nuevo mensaje mientras se utiliza una misma clave, ya que el contador de tramas se va incrementando. De esta forma, si un atacante captura un paquete y trata de volver a utilitzarlo pasado un tiempo, el receptor será capaz de detectarlo, y así garantizará el frescor de los mensajes. Si el atacante no modifica el contador de la trama, entonces el receptor detectará que se trata de una trama antigua. Si, por el contrario, el atacante modifica el contador de la trama, entonces la verificación del MIC no será correcta, y también detectará el ataque. El uso de un *nonce* también garantiza que mensajes con exactamente el mismo contenido en claro sean cifrados como textos diferentes.

***Nonce* deriva de la expresión en inglés *number used once*.**

Contador de trama

ZigBee utiliza contadores de 32 bits: dos dispositivos intercambiando un mensaje cada segundo no generarán un contador de trama duplicado hasta transcurridos más de 136 años de interacción.

3.2.3. Seguridad en la capa de aplicación

La subcapa APS se encarga de la seguridad de los mensajes originados en la capa de aplicación, utilizando o bien la clave de enlace (para mensajes *unicast*) o bien la clave de red (para mensajes *broadcast*).

La tabla 4 muestra los campos que contiene una trama ZigBee de la capa de aplicación. Como se puede apreciar, aparte de las cabeceras propias de cada capa, la capa de aplicación añade una cabecera auxiliar y un campo de integridad, que permiten incluir la información necesaria para gestionar la seguridad del contenido. La creación de la cabecera auxiliar, del campo de integridad y de la carga cifrada (si es el caso) sigue el mismo formato que el especificado para la capa de red en el subapartado 3.2.2. Sin embargo, en este caso, el valor del campo de *nonce* extendido siempre será 0. La capa de aplicación también utiliza AES en modo CCM* para cifrar y ofrecer integridad y autenticidad a los mensajes.

Tabla 4. Trama de la capa de aplicación

SYNC	PHY HDR	MAC HDR	NWK HDR	APS HDR	Aux HDR	ENC APS Payload	MIC

La subcapa APS de la capa de aplicación también es la responsable de ofrecer servicios de seguridad a las aplicaciones y al ZDO. En los siguientes subapartados, veremos qué servicios de seguridad ofrece la subcapa APS y cómo funcionan.

Gestión de claves

La subcapa APS ofrece cuatro servicios básicos de gestión de claves: el establecimiento de clave, el transporte de clave, la petición de una clave y el cambio de clave.

El servicio de **establecimiento de clave** permite que dos dispositivos ZigBee puedan derivar una clave secreta compartida (una clave de enlace) a partir de una información secreta compartida previamente (la clave maestra). El protocolo de establecimiento de claves se lleva a cabo entre dos dispositivos, uno que inicia el protocolo y otro que responde a la petición de establecimiento de claves. Antes, los dos dispositivos deben compartir algún secreto, que se puede obtener a través del centro de confianza. Así pues, para establecer una clave, se siguen los pasos que exponemos a continuación:

1) establecer una relación de confianza;

2) intercambiar datos efímeros;

3) utilizar los datos efímeros para derivar una clave de enlace, y

4) confirmar que las claves se han calculado de forma correcta.

Otra alternativa para obtener una clave es el envío de dicha clave a través de un canal, preferiblemente seguro. El servicio de **transporte de clave** del APS permite que un dispositivo envíe una clave a otros dispositivos. El servicio puede operar de manera segura, protegiendo criptográficamente las claves enviadas, o bien de manera no segura, sin ofrecer ningún tipo de protección sobre el contenido enviado. En este último caso, se entiende que la seguridad del transporte de la clave se garantizará por algún otro medio (no criptográfico).

Cuando el transporte de clave se hace en modo seguro, entonces se utilizan claves específicas para cifrar los mensajes que transportan las claves. Si la clave que se transporta es una clave maestra, se utiliza la clave para carga de clave (*key-load key*), mientras que para cualquier otra clave se utiliza la clave para transporte de clave (*key-transport key*). Tanto la clave para carga como la de transporte de claves son claves derivadas de la clave de enlace utilizando HMAC (*hash-based message authentication code*, observad la figura 10) con la función de *hash* Matyas-Meyer-Oseas.

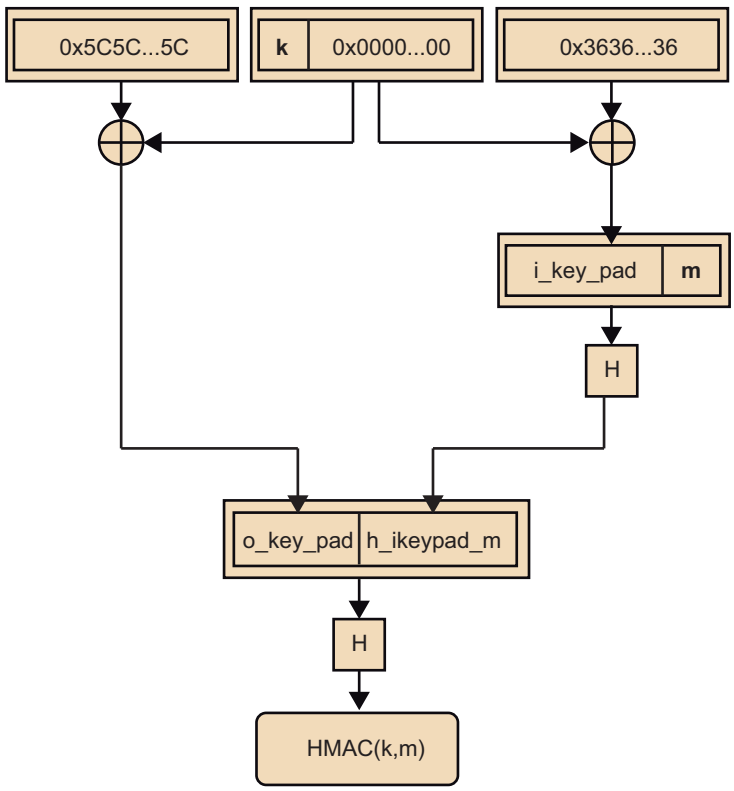
Clave de transporte y clave de carga

La clave de transporte de claves se obtiene utilizando como entrada el valor 0x00 y como clave la clave de enlace. En cambio, la clave de carga de clave utiliza como entrada el valor 0x02 (y también la clave de enlace como clave).

Figura 10

Dada una función de *hash* criptográfica H, una clave k y un mensaje m, el esquema de la figura muestra cómo calcular su código de autenticación HMAC. Como se puede apreciar, HMAC utiliza dos cadenas constantes (0x5C . . . 5C y 0x36 . . . 36) junto con la aplicación de una función *hash* intermedia para enmascarar los valores de entrada de la función *hash* que devuelve el valor HMAC.

Figura 10. Esquema HMAC



La subcapa APS también proporciona servicios para solicitar una clave o bien para informar a otro dispositivo de que debería cambiar de clave. El servicio de **solicitud de clave** permite a un dispositivo pedir una clave (o bien la clave de red activa o bien una clave maestra) a otro dispositivo de manera segura. El servicio de **cambio de clave** permite a un dispositivo informar a otro de que debería cambiar a una clave de red activa diferente de manera segura.

Modificación de la red

La capa APL también dispone de servicios para eliminar un dispositivo de la red de manera segura y para informar de tales cambios, así como de cualesquiera otros cambios de estado, a otros dispositivos de la red.

El servicio para eliminar un dispositivo permite a un dispositivo como el centro de confianza informar a otro dispositivo, por ejemplo al direccionador, de que uno de sus hijos debería ser eliminado de la red. Este servicio puede resultar útil, por ejemplo, si se quiere eliminar de la red un dispositivo que no cumple los requisitos de seguridad que establece el centro de confianza.

El servicio de actualización permite a un dispositivo informar a un segundo de que un tercero ha cambiado su estado. De esta forma, el centro de confianza puede mantener una lista actualizada de los dispositivos activos en la red.

Autenticación

La subcapa de soporte de aplicación también ofrece el servicio de autenticación de entidades, lo cual permite a los dispositivos sincronizar información y asegurar al mismo tiempo la autenticidad de los dispositivos implicados. Opcionalmente, también permite autenticar no solo los dispositivos, sino también los datos que se están transmitiendo. La autenticación se lleva a cabo a partir de un secreto compartido previamente, en este caso, una clave compartida entre los dispositivos.

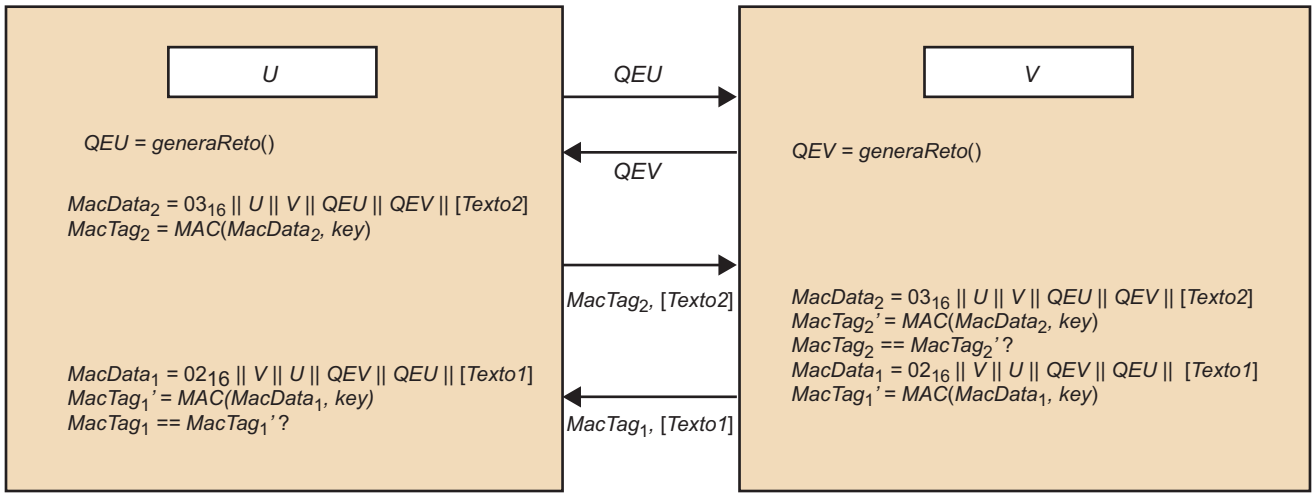
La figura 11 describe el funcionamiento del protocolo de autenticación mutua de entidades. Una vez finalizado el protocolo, los dos dispositivos saben que se están comunicando con el dispositivo con el que compartían previamente la clave utilizada durante el protocolo. El protocolo asegura que una escucha pasiva no permita a un atacante descubrir la clave que comparten los dispositivos.

Figura 11

El esquema se inicia con un intercambio de retos: el dispositivo que inicia el protocolo, *U*, envía un reto al dispositivo *V* y *V* responde con otro reto. El dispositivo iniciador calcula entonces el valor MacTag2 aplicando una función MAC a una cadena que contiene un

valor fijo 03_{16}, los dos retos intercambiados, los identificadores de los dispositivos y, opcionalmente, un texto por transmitir. Para calcular esta MAC, *U* utiliza la clave *key* que los dos dispositivos, *U* y *V* compartían previamente. *U* envía el valor calculado a *V*. Para comprobar que se está comunicando con el dispositivo con el que compartía la clave, *V* calculará también el mismo valor MacTag2 y lo comparará con el valor recibido. Si son iguales, entonces *V* sabe que, efectivamente, se está comunicando con *U*, dispositivo con el que comparte la clave *key*. El protocolo sigue ahora en sentido contrario, y permite a *U* comprobar la autenticidad de *V*.

Figura 11. Protocolo de autenticación mutua



4. Comparativa y discusión de la seguridad

Como hemos visto, la tecnología RFID se utiliza en dispositivos muy diversos, desde etiquetas pasivas que no disponen de ningún tipo de batería para operar de manera autónoma y con unos recursos de cómputo muy limitados, hasta dispositivos activos, que disponen de batería y de una capacidad de cómputo mucho más elevada. Toda esta gama de dispositivos RFID responde a diferentes exigencias del mercado, donde el coste de cada etiqueta es un factor clave. El área de la etiqueta, medida en puertas lógicas equivalentes, determinará las capacidades de cómputo del dispositivo y, por tanto, limitará los mecanismos de seguridad que se puedan desplegar. Así pues, el nivel de seguridad que se pueda conseguir depende en gran medida del dispositivo RFID concreto que se esté utilizando.

Como hemos visto, se está trabajando en desarrollar protocolos y primitivas criptográficas que requieran pocos recursos de cálculo para ser implementados. De este modo, se puede dotar incluso a los dispositivos más sencillos de algún nivel de seguridad. A pesar de ello, las necesidades de recursos de ciertos mecanismos, como por ejemplo la criptografía de clave pública, todavía están muy lejos de los que se encuentran disponibles en la mayoría de los dispositivos RFID.

Los dispositivos Bluetooth no se encuentran tan limitados por los recursos disponibles. A diferencia de la RFID, Bluetooth tiene una especificación de seguridad que describe sus modos de uso y los mecanismos de seguridad que se tienen que utilizar con cada configuración. A pesar de ello, algunos puntos del diseño de la seguridad hacen disminuir su robustez.

Desde el punto de vista del cifrado, el criptosistema de flujo E_0 no es lo bastante robusto, ya que se ha demostrado que se puede romper bajo ciertas circunstancias.

Sin embargo, la problemática más importante proviene de los mecanismos de autenticación. Como hemos visto, la clave de enlace es muy importante y puede ser de diferentes tipos. En el caso de no existir una conexión previa entre los dispositivos, la clave de enlace se obtiene por medio de la clave de inicialización. Esta clave se genera, básicamente, a partir del PIN, ya que el valor aleatorio se transmite como texto claro de un dispositivo a otro. Teniendo en cuenta que los PIN están formados por cuatro dígitos, el número de posibilidades es bastante bajo. Por otro lado, se produce un caso peor cuando se utiliza la clave de dispositivo como clave de enlace. Ya hemos comentado que

la clave de dispositivo se genera al inicializarlo y raramente se cambia. Este hecho hace que, una vez que un dispositivo A ha utilizado su clave de dispositivo como clave de enlace para autenticarse ante B, el siguiente proceso de autenticación de A ante un tercer dispositivo con su clave de dispositivo como clave de enlace no puede ser fiable, ya que el dispositivo B podría hacerse pasar por A (dado que conoce su clave de dispositivo).

Finalmente, hay que destacar que los esquemas de seguridad que incorpora la arquitectura Bluetooth autentican dispositivos pero no a usuarios. Este hecho, junto con la poca longitud del PIN, hace que no resulte adecuado en ciertas aplicaciones.

Al igual que Bluetooth, ZigBee es una especificación de un conjunto de protocolos de comunicaciones. La especificación de seguridad de ZigBee también prevé el uso de diferentes modos de seguridad que permiten ajustar las necesidades de seguridad según la aplicación. Como hemos visto, la seguridad de ZigBee se basa en el uso de dos claves de 128 bits: la clave de enlace y la clave de red. Esta última, compartida entre todos los dispositivos de una misma red, es uno de los puntos por los que se puede atacar fácilmente a una red ZigBee. Un atacante puede obtenerla, por ejemplo, interceptando su transmisión por el canal fuera de banda o consiguiendo acceder a uno de los dispositivos de la red y extrayendo la información que contiene. Conociento esta clave, un atacante puede descifrar todas las comunicaciones *broadcast* de la red.

ZigBee también propone el uso de un centro de confianza en el que confían todos los nodos de la red. El centro de confianza es responsable del control de admisión de los nodos y de la distribución de claves. Mientras que disponer de un centro de confianza permite mantener un control centralizado sobre la seguridad de la red, también supone un único punto de error que un atacante puede aprovechar.

Más allá de los mecanismos de seguridad que implementan, ZigBee y Bluetooth presentan algunas similitudes, aunque también numerosas diferencias. Mientras que una red ZigBee puede tener hasta 65.535 nodos (en subredes de 255), una *piconet* Bluetooth solo puede tener ocho. El consumo de los dispositivos también es un punto diferenciador: mientras que los dispositivos ZigBee consumen menos de 1 mW, los dispositivos Bluetooth pueden consumir hasta 100 mW. En relación con la velocidad de transmisión, Bluetooth ofrece hasta 3 Mb/s (velocidad nominal en la versión 2.2), mientras que con ZigBee solo se llega a los 250 kb/s. Tales diferencias hacen que Bluetooth se utilice normalmente en teléfonos móviles o dispositivos portátiles y que el uso de ZigBee se encuentre más extendido en otros ámbitos, como la domótica.

Resumen

En este módulo didáctico hemos tratado los problemas de seguridad que afrontan las redes inalámbricas de alcance personal y hemos dado algunas pinceladas a los esquemas que ponen en juego diferentes tecnologías para afrontarlos.

La tecnología RFID se encuentra limitada en gran medida por el coste que se puede asumir en la producción de cada dispositivo. Hemos visto que estas limitaciones afectan al nivel de seguridad que se puede llegar a implementar en estos dispositivos. Para poder implementar algoritmos criptográficos en dispositivos RFID, hemos observado que se tienen que adaptar para reducir su complejidad, bien mediante el rediseño de los sistemas, bien por medio de la reducción del tamaño de los parámetros que utilizan. Aparte de adaptar algoritmos existentes a las capacidades de los RFID, otra alternativa es diseñar esquemas nuevos teniendo en cuenta desde el primer momento los recursos disponibles.

La tecnología Bluetooth puede trabajar en diferentes modos de seguridad, de los que hemos destacado los dos más importantes: el modo de seguridad en el ámbito de enlace y el modo de seguridad en el ámbito de servicio. En el primero, la política de seguridad se aplica antes de la conexión, mientras que en el segundo, se aplica en el ámbito de aplicación (una vez que la conexión entre los dispositivos ya ha tenido lugar). Los problemas básicos de los mecanismos de seguridad se desprenden de la gestión de las claves de enlace y su reutilización en diferentes procesos de autenticación.

La especificación de ZigBee también prevé el uso de diferentes modos de seguridad para sus redes. A lo largo del apartado, hemos repasado diferentes aspectos de la seguridad en sistemas ZigBee, desde la gestión de claves hasta los mecanismos de seguridad que se pueden aplicar a las diferentes capas.

Finalmente, hemos revisado los aspectos de seguridad comentados para las tres tecnologías desde un punto de vista más crítico y hemos elaborado una pequeña comparativa que resume las diferencias esenciales entre los dispositivos RFID, Bluetooth y ZigBee.

Actividades

1. Buscad información sobre aplicaciones reales en las que se utilice RFID y tratad de averiguar qué mecanismos de seguridad implementan.

2. Para decidir si una secuencia seudoaleatoria se parece a una secuencia verdaderamente aleatoria o no, se utiliza una serie de tests estadísticos. Buscad información sobre cuáles son estos tests.

3. Los términos *bluejacking*, *bluesnarfing* y *bluebugging* se emplean para definir ataques a dispositivos móviles utilizando su módulo Bluetooth. Buscad información sobre estos ataques.

4. ZigBee describe un conjunto de perfiles de alto nivel para favorecer la interoperabilidad entre dispositivos. Buscad información sobre los perfiles permitidos y su utilización.

5. UWB es otra especificación para comunicaciones inalámbricas. Buscad información sobre la seguridad que implementa UWB.

Ejercicios de autoevaluación

1. Hemos dicho que el protocolo HB solo es seguro ante un atacante pasivo. ¿Qué podría hacer un atacante activo, con capacidad para interactuar con la etiqueta, para descubrir el valor secreto x?

2. ¿Qué limitaciones presentan el modelo de autenticación reto-respuesta y el uso de la criptografía de clave compartida en las comunicaciones sin hilos?

3. ¿Por qué se describe un modo de seguridad 2 en el estándar Bluetooth si el modo de seguridad 3 incorpora un nivel más elevado de seguridad?

4. ¿Por qué se describe un modo de seguridad estándar en ZigBee si el modo de alta seguridad incorpora un nivel más elevado de seguridad?

Solucionario

1. Un atacante activo tiene la capacidad de ejecutar el protocolo con la etiqueta tantas veces como quiera, y tiene la habilidad de elegir el valor *a* en cada ejecución del protocolo. Por tanto, en primer lugar, el atacante puede ejecutar varias veces el protocolo con un mismo valor *a*, de manera que puede eliminar el ruido introducido por el bit *b*. Una vez que asumimos que el atacante conoce el resultado de *ax*, solo hay que recoger los valores *ax* para diferentes *a* y construir un sistema de ecuaciones lineal.

2. La principal limitación que presentan el modelo de autenticación reto-respuesta y el uso de la criptografía de clave compartida es que en ambos casos las partes que se comunican tienen que compartir cierta información. Según el entorno (por ejemplo, en la telefonía móvil), este hecho no presenta ningún problema, pero para aplicaciones abiertas, el intercambio de esta información puede convertirse en un problema difícil de gestionar.

3. El modo de seguridad 3 es mucho más restrictivo que el modo 2, ya que el control se hace directamente en el ámbito de enlace, mientras que en el modo 2 se hace en la aplicación. La ventaja de disponer de un nivel de seguridad 2 es que permite la conexión de cualquier dispositivo, lo que facilita la existencia de aplicaciones más abiertas, como la consulta de los servicios disponibles de un dispositivo.

4. Al igual que con Bluetooth, los modos de seguridad más elevados son más restrictivos, y limitan el acceso a la red a nodos que cumplen todos los requisitos de seguridad fijados. Además, en modo de seguridad alta, los recursos necesarios para mantener el centro de confianza son mucho más elevados y crecen a medida que la red aumenta de tamaño.

Glosario

AES *m advanced encryption standard*.

autenticidad *f* En relación con la información, propiedad que presenta de encontrarse en el mismo estado en que fue producida, sin modificaciones no autorizadas.
en authenticity

centro de confianza *m* En una red ZigBee, dispositivo especial que goza de la confianza de todos los demás dispositivos de la red.
en trust center

cifra *f* Véase **criptosistema**.

clave *f* Parámetro, normalmente secreto, que controla los procesos de cifrado o descifrado.

clave combinación *f* Clave generada conjuntamente por dos dispositivos Bluetooth y utilizada como clave de enlace.

clave de dispositivo *f* Clave específica de cada dispositivo Bluetooth a menudo utilizada como clave de enlace.

clave de enlace *f* En la tecnología Bluetooth, clave utilizada para llevar a cabo el proceso de autenticación entre dispositivos. En ZigBee, clave que comparten dos dispositivos ZigBee utilizada en la comunicación *unicast*.

clave de inicialización *f* Clave utilizada en la tecnología Bluetooth para proteger el intercambio de la clave de enlace.

clave de red *f* Clave que comparten todos los dispositivos ZigBee en una misma red. Se utiliza para la comunicación *broadcast* en el ámbito de la aplicación o para la comunicación en el ámbito de red.

clave maestra *f* En Bluetooth, clave temporal que se utiliza en una red Bluetooth con más de dos dispositivos conectados cuando el dispositivo maestro quiere transmitir simultáneamente a los otros dispositivos. En ZigBee, clave que comparten dispositivos ZigBee que sirve como secreto compartido inicial para derivar nuevas claves.

confidencialidad *f* Propiedad que asegura que solo los que están autorizados tendrán acceso a la información.
en secrecy

criptografía *f* Ciencia que estudia las técnicas matemáticas utilizadas para la protección de la información.

criptografía de clave compartida *f* Grupo de criptosistemas que basan su seguridad en una sola clave, que emisor y receptor utilizan tanto para cifrar como para descifrar.

criptografía ligera *f* Conjunto de técnicas criptográficas diseñadas para dispositivos con recursos limitados que intentan ofrecer un compromiso entre rendimiento, seguridad y coste.
en lightweight cryptography

criptosistema *m* Método que permite cifrar un texto claro para obtener un texto cifrado ininteligible.
sin. **Cifra**

criptosistema de flujo *m* Criptosistema que basa su funcionamiento en un generador seudoaleatorio que, por medio de una clave como valor de entrada, genera una secuencia de cifrado.

denegación de servicio *f* Ataque que consiste en conseguir que el servicio no esté disponible para los usuarios legítimos o bien que el servicio que se dé se retrase o se interrumpa.
en denial of service

denial of service *m* Véase **denegación de servicio**.
sigla **DoS**

DES *m data encryption standard*.

deterministic random bit generator *m* Véase **generador seudoaleatorio**.
sigla **DRBG**

dispositivo con funcionalidades completas *m* En la especificación ZigBee, dispositivo que puede realizar cualquier papel dentro de la red.
en full-function device

dispositivo con funcionalidades reducidas *m* En la especificación ZigBee, dispositivo que solo puede actuar como dispositivo final y que solo puede interaccionar con un único dispositivo con funcionalidades completas.
en reduced-function device

DoS *m* Véase **denegación de servicio**.

DRBG *m* Véase **generador seudoaleatorio**.

FFD *m* Véase **dispositivo con funcionalidades completas**.

forward security *f* Véase **seguridad hacia delante**.

full-function device *m* Véase **dispositivo con funcionalidades completas**.
sigla **FFD**

gate equivalents *f* Puertas equivalentes a una NAND de dos entradas.
sigla **GE**

GE *f* Véase **gate equivalents**.

generador seudoaleatorio *m* Proceso determinista capaz de generar una secuencia seudoaleatoria.
en deterministic random bit generator

gestor de seguridad *m* Entidad de la tecnología Bluetooth que implementa la política de seguridad especificada en el modo de seguridad 2.

indistinguibilidad *f* En RFID, propiedad que impide que un adversario pueda distinguir, es decir, diferenciar, dos etiquetas distintas solo observando sus salidas.

integridad *f* Propiedad que asegura la no alteración de la información.

jamming *m* Ataque que consiste en atenuar la señal de radio para provocar interferencias en el servicio.

LFSR *m* Véase **registro de desplazamiento realimentado linealmente**.

lightweight cryptography *f* Véase **criptografía ligera**.

NLFSR *m* Registro de desplazamiento realimentado no linealmente.

nonce *m* Véase **number used once**.

number used once *m* Número arbitrario que se utiliza una única vez y que permite, por ejemplo, evitar ataques de *replay*.

privacidad *f* Véase **confidencialidad**.

PRNG *m* Véase **generador seudoaleatorio**.

pseudorandom number generator *m* Véase **generador seudoaleatorio**.
sigla **PRNG**

reduced-function devices *m* Véase **dispositivo con funcionalidades reducidas**.
sigla **RFD**

registro de desplazamiento realimentado linealmente *m* Dispositivo físico o lógico formado por *n* celdas de memoria y una función de alimentación lineal.
sigla **LFSR**

reto-respuesta *m* Sistema de autenticación por el cual dos partes se pueden autenticar de forma remota. Este sistema de autenticación requiere que las dos partes se hayan intercambiado cierta información antes del proceso de autenticación.

RFD *m* Véase **dispositivo con funcionalidades reducidas**.

RFID *f* Identificación mediante radiofrecuencia.

seguridad hacia delante *f* Extensión de las propiedades de autenticidad y de indistinguibilidad que garantiza que dichas propiedades se mantienen para transacciones pasadas cuando un atacante es capaz de corromper una etiqueta en un momento determinado.
en forward security

trust center *m* Véase **centro de confianza**.

Bibliografía

Baronti, P.; Pillai, P.; Chook, V.; Chessa, S.; Gotta, A.; Hu, Y. (2007). *Wireless Sensor Networks: a Survey on the State of the Art and the 802.15.4 and ZigBee Standards*. Amsterdam: Elsevier Science Publishers B. V.

Domingo, J.; Herrera, J. (1999). *Criptografia*. Barcelona: UOC.

Farahani, S. (2008). *ZigBee Wireless Networks and Transceivers*. Newton, MA, EE. UU.: Newnes.

Finkenzeller, K. (2003). *RFID Handbook: Fundamentals and Applications in Contactless Smart Cards and Identification*. John Wiley & Sons.

Gehrmann, C.; Persson, J; Smeets, B. (2001). *Bluetooth security*. Artech House Publishers.

Gislason, D. (2008). *ZigBee Wireless Networking*. Newton, MA, EE. UU.: Newnes.

Jakobsson, M.; Wetzel, S. (2001). "Security weaknesses in Bluetooth". En: *Proceedings of RSA 2001* (LNCS 2020, págs. 176-191). Springer Verlag.

Knospe1, H.; Lemke-Rust, K. (2010). "Towards Secure and Privacy-Enhanced RFID Systems". En: *RFID Systems - Research Trends and Challenges* (cap. 16). John Wiley & Sons.

Lee, J. S.; Su, Y. W.; Shen, C. C. (2007). *A Comparative Study of Wireless Protocols: Bluetooth, UWB, ZigBee, and Wi-Fi*. Taiwan: Proceedings of the 33rd Annual Conference of the IEEE Industrial Electronics Society.

Maimut, D.; Ouafi, K. (2012). "Lightweight Cryptography for RFID Tags". *IEEE Security and Privacy*. IEEE Computer Society.

Menezes, A.; Oorschot, P.; Vanstone, S. A. (2001). *Handbook of Applied Cryptography (5a ed.)*. CRC-Press.

Vainio, J. T. (2000). "Bluetooth security". En: *Proceedings of Helsinki University of Technology, Telecommunications Software and Multimedia Laboratory*. Helsinki.

ZigBee Alliance (2007). *ZigBee Specification*. ZigBee Document 053474r17.