**Politechnika Gdańska**

**WYDZIAŁ ELEKTRONIKI TELEKOMUNIKACJI I INFORMATYKI**

# „Badanie bezpieczeństwa sieci Bluetooth”

*Krzysztof Kucharski*

# Spis treści

# 1 STANDARD BLUETOOTH

## 1.1 Czym jest Bluetooth ?

Bluetooth jest otwartą specyfikacją standardu sieci, która umożliwia bezprzewodową, szybką komunikację na małe odległości między różnymi urządzeniami elektronicznymi (np. telefon komórkowy, PDA, piloty telewizyjne, laptopy, urządzenia peryferyjne) . Specyfikacja ta powstała, aby wyeliminować połączenia kablowe pomiędzy urządzeniami. Głównymi zaletami Bluetooth jest otwartość, elastyczność, mały pobór prądu i bardzo niska cena.

Standard Bluetooth od samego początku był tak projektowany, aby można było dzięki niemu przesyłać nie tylko dane, ale również mowę z gwarancją odpowiedniej jakości. Zaimplementowano więc standard QoS (ang. Quality of Service), który miał umożliwić wykorzystanie sieci w kontekście telefonii i konferencji wideo na małą odległość.

Pierwsza specyfikacja standardu komunikacji bezprzewodowej Bluetooth została opublikowana w 1999 roku, a trzy lata później kolejna wersja została zatwierdzona przez IEEE jako standard IEEE 802.15.1. Aktualna wersja specyfikacji ma numer 2.1 i uwzględnia dziesięcioletni zbiór doświadczeń w zakresie projektowania sieci bliskiego zasięgu.

Grupa projektująca sieć standardu Bluetooth zwana SIG (ang. Special Interest Group) już od samego początku postawiła na otwartość standardu. Każdy producent może bez żadnych opłat licencyjnych tworzyć urządzenia zgodne z Bluetooth i po zatwierdzeniu ich zgodności ze specyfikacją przez specjalną komórkę SIG czerpać zyski ze sprzedaży. Jest to jedna z przyczyn szybkiego zdobycia tak wielkiej popularności przez tę sieć. Sama grupa tworząca specyfikacje jest zrzeszeniem non profit i ciągle przyjmuje nowych członków, których według informacji na swojej stronie [23] ma już ponad dziewięć tysięcy.

Obecnie standard ten jest tak popularny, że odbiornik Bluetooth jest wbudowany w prawie każdego nowego laptopa oraz w telefon komórkowy. Można więc powiedzieć, że zdominował on rynek PAN (ang. Personal Area Network) stając się de facto światowym standardem w komunikacji urządzeń elektronicznych na małą odległość i w ciągu dziesięciu lat wyparł inne rozwiązania tego typu. Jedynym standardem, który jest w stanie w obecnej chwili zagrozić Bluetooth jest testowany właśnie Wireless USB, który będzie korzystał z identycznej warstwy radiowej jak nadal opracowywany standard Bluetooth 3.

Ciekawostką jest historia terminu „bluetooth”. Nazwa pochodzi od przydomka duńskiego Króla Harolda Sinozębego (ang. Bluetooth), który zjednoczył Danię i Norwegię i wprowadził chrześcijaństwo w Skandynawii. Logo Bluetooth łączy znaki alfabetu runicznego, będące odpowiednikami liter alfabetu łacińskiego H i B. Logo to przedstawiono na rysunku 1.


Rysunek 1: Logo Bluetooth [23]

## 1.2 Stos protokołów

Analizując standardy sieciowe warto zapoznać się z stosem protokołów, na którym opiera się dana architektura. Standard Bluetooth określa wiele protokołów, pogrupowanych w warstwy. Struktura warstw nie odpowiada żadnemu znanemu modelowi. Nie jest więc zgodna z terminologią Modelu Odniesienia ISO/OSI. Dlatego w niniejszym dokumencie zostanie zachowana terminologia używana w specyfikacji Bluetooth, która może być myląca dla osób przyzwyczajonych do innych modeli.

### 1.2.1 Urządzenia Bluetooth

Specyfikacja protokołu Bluetooth wyróżnia dwa niezależne elementy architektury:

- Bluetooth Controller,
- Bluetooth Host.

***Bluetooth Controller*** jest fizycznym urządzeniem rozszerzającym możliwości hosta o możliwości współpracy z technologią Bluetooth. Może to być zarówno adapter podłączany do któregoś z portu komputera, jak i moduł bezpośrednio wbudowany w urządzenie mobilne taki jak np. telefon komórkowy czy PDA.

***Bluetooth Host*** jest to implementacja stosu protokołów w urządzeniu, do którego podłączony jest adapter.

### 1.2.2 Architektura logiczna

Stos protokołów Bluetooth możemy najogólniej podzielić na trzy grupy logiczne: [4]

- grupę protokołów transportowych:
  - niższe warstwy grupy protokołów transportowych
  - wyższe warstwy grupy protokołów transportowych
- grupę protokołów pośredniczących
- grupę aplikacji

**Grupa protokołów transportowych** została stworzona, aby umożliwić urządzeniom odnajdywanie się oraz tworzenie, konfigurowanie i zarządzanie zarówno fizycznymi jak i logicznymi połączeniami. Grupę tą dzielimy jeszcze na dwie podgrupy: warstwę niższą i warstwę wyższą grupy protokołów transportowych.

Warstwa niższa jest w pełni implementowana w kontrolerze, natomiast warstwa wyższa odpowiedzialna za styk kontroler – host jest implementowana zarówno w kontrolerze jak i na maszynie hosta.

To właśnie ta grupa protokołów zwana jest Core System i jest opisana w pierwszej części specyfikacji Core Specification. Jest to więc grupa protokołów, które muszą być obowiązkowo zaimplementowane w każdym urządzeniu zgodnym z Bluetooth.

**Grupa protokołów pośredniczących** jest wykorzystywana w celu przystosowania architektury Bluetooth do już istniejących standardów i umożliwia w wielu przypadkach zastąpienie przestarzałych łączy kablowych lub bezprzewodowych nową technologią bez jakichkolwiek ingerencji w istniejące już aplikacje. Są to zaadoptowane na potrzeby Bluetooth istniejące już protokoły (np. OBEX – ang. OBject EXchange), jak i specjalnie zaprojektowane przez SIG nowe rozwiązania dopasowane do potrzeb standardu (np. SDP – ang. Service Discovery Protocol).

Ta grupa protokołów nie jest opisana w podstawowej specyfikacji Bluetooth Core

Specification (z pewnymi wyjątkami) co oznacza, że ich implementacja jest nie obowiązkowa w urządzeniach zgodnych ze standardem. W zależności od przeznaczenia urządzenia będą w nim zaimplementowane tylko potrzebne protokoły pośredniczące.

**Grupa aplikacji** to zarówno istniejące już aplikacje nie projektowane dla Bluetooth, jak i nowe specjalnie napisane programy wykorzystujące zalety łączy bezprzewodowych. Aplikacje, które nie były projektowane dla sieci bezprzewodowej, dzięki specjalnie napisanym protokołom emulującym łącze przewodowe mogą współpracować z architekturą Bluetooth.

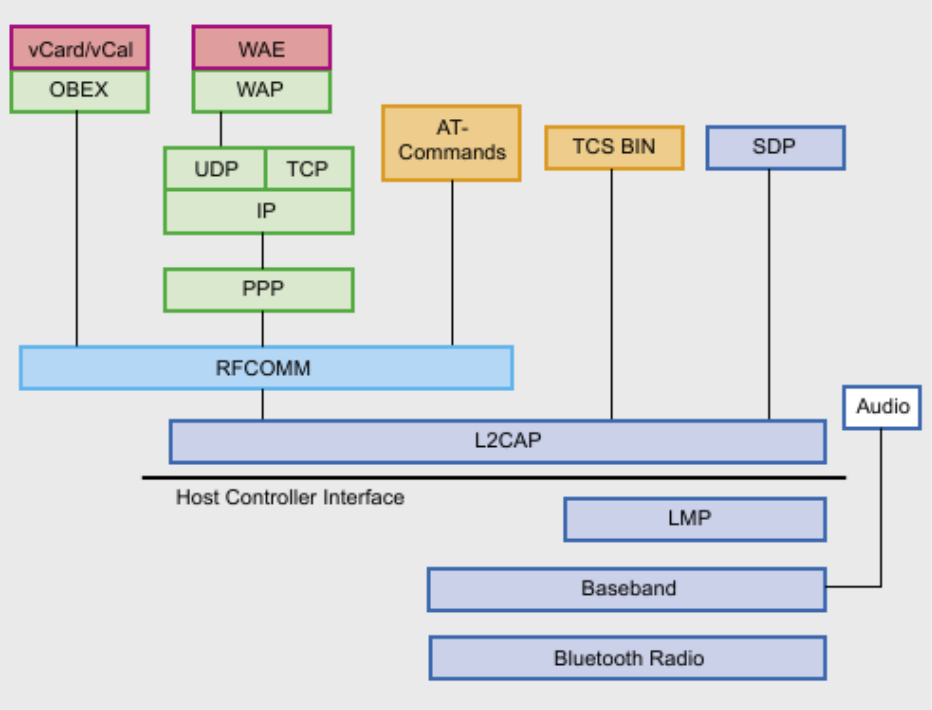
Specyfikacja wspomaga tworzenie aplikacji wykorzystujących sieci bluetooth poprzez stworzenie Profili Aplikacji, które definiują scenariusze wykorzystania standardu.

### 1.2.3 Szczegółowe omówienie stosu protokołów

Wyróżniamy następujące warstwy stosu protokołów: [23]

- niższe warstwy grupy protokołów transportowych:
  - Interfejs Radiowy
  - Kontroler Połączenia i Pasmo Podstawowe
  - Menedżer Połączenia
- wyższe warstwy grupy protokołów transportowych:
  - warstwa L2CAP (ang. Logical Link Control and Adaptation Protocol)
  - interfejs sterujący hosta HCI (ang. Host Control Interface)
- protokoły pośredniczące:
  - protokół emulacji portu szeregowego RFCOMM (ang. Radio Frequency Communication)
  - protokół wyszukiwania usług SDP (ang. Service Discovery Protocol)
- protokoły współpracy z IrDA (ang. Infrared Data Association)
  - protokół zarządzania modemem AT – Commands
- warstwa aplikacji:
  - profile Bluetooth

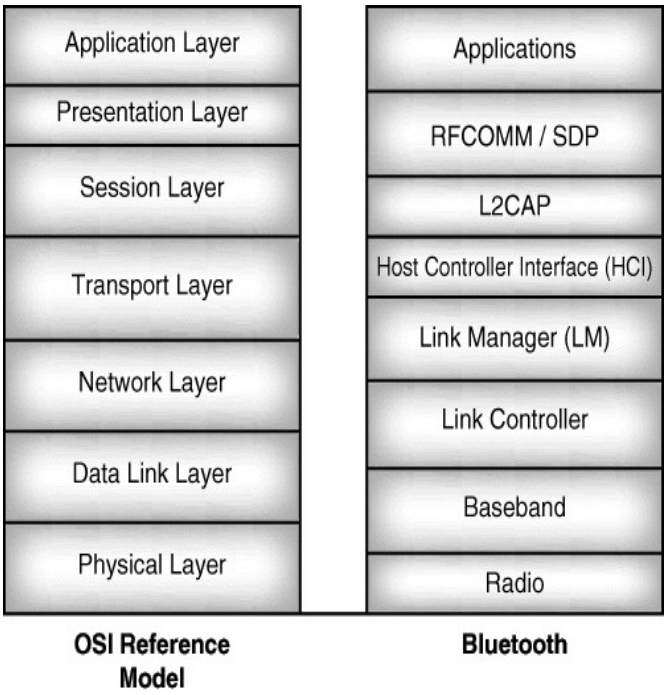
Zależności pomiędzy poszczególnymi warstwami przedstawia rysunek 2.




Rysunek 2: Stos protokołów Bluetooth [3]

Jak widać opisywana architektura jest złożona i niejednorodna. Wiele protokołów zaprojektowanych lub adaptowanych do potrzeb Bluetooth istnieje równolegle i są one przewidziane do pełnienia bardzo wielu różnorodnych funkcji.

Jak wspomniano konstrukcja protokołów Bluetooth nie jest zgodna z Modelem Odniesienia ISO/OSI. Na rysunku 3 pokazano zestawienie obydwu architektur logicznych.




Rysunek 3: Porównanie warstw modelu ISO/OSI i Bluetooth [2]

## 1.3 Warstwa Radiowa

Architektura Bluetooth wykorzystuje pasmo częstotliwości 2,4GHz wolne od opłat licencyjnych w większości krajów Świata i przeznaczone do celów przemysłowych, naukowych i medycznych ISM (ang. Industrial, Scientific, Medical). Pasmo to jest podzielone na 79 kanałów od częstotliwości 2402 MHz do częstotliwości 2480 MHz. Poszczególne kanały są więc wyznaczone co 1MHz.

W przeciwieństwie do wielu innych standardów sieci bezprzewodowej emisja danych nie odbywa się na stale określonej częstotliwości, ale dane są nadawane w postaci pakietów, z których każdy jest transmitowany w innym kanale. Jest to transmisja z widmem rozproszonym uzyskiwanym metodą skakania po częstotliwościach **FHSS** (ang. Frequency Hopping Spread Spectrum). Urządzenie nadzorujące wyznacza sekwencje skoków po częstotliwościach, której używają wszystkie urządzenia komunikujące się z nim w danej podsieci. Technika skoków po częstotliwościach bardzo skutecznie uniemożliwia podsłuch i jednocześnie w pewien sposób umożliwia współistnienie urządzeń innych standardów korzystających z pasma 2.4GHz np. WiFi. Wersja 1.2 specyfikacji wprowadza algorytm **AFH** (ang. Adaptive Frequency Hopping), który pomija kanały zajęte przez inne urządzenia. [1]

Główne funkcje warstwy radiowej to [4] :

- generowanie nośnej,
- modulacja nośnej (dane nad./odb.),
- sterowanie mocą nadawczą,
- pomiar siły sygnału.

W zależności od klasy mocy urządzenie może mieć następujący zasięg:

- klasa 1 - 100 mW (20 dBm) - zasięg do 100 m
- klasa 2 - 2,5 mW (4 dBm) - zasięg do 10 m
- klasa 3 - 1 mW (0 dBm) - zasięg do 1 m.

Szybkość transmisji danych w zależności do wersji specyfikacji są następujące:

- Bluetooth 1.0 - 721 kb/s
- Bluetooth 1.1 - 721 kb/s
- Bluetooth 1.2 - 721 kb/s
- Bluetooth 2.0 - 2,1 Mb/s lub 3,0 Mb/s z EDR

Wraz z wprowadzeniem wersji 2.0 specyfikacji do warstwy radiowej zostało dodane rozszerzenie **EDR** (ang. Enhanced Data Rate). Jest to opcjonalna funkcjonalność, która umożliwia zwiększenie szybkości transmisji danych . Rozszerzenie to wprowadza dodatkowy tryb przesyłania pakietów, w którym kod dostępu i nagłówek pakietu przesyłane są w zwykłym trybie **Basic Rate** z modulacją GFSK i szybkością 1Mbit/s natomiast dane są przesyłane w trybie EDR z modulacją PSK.

## 1.4 Warstwa kontrolera połączeń i pasmo podstawowe

O ile zadaniem warstwy radiowej jest po prostu fizyczne wysłanie drogą radiową dostarczonych danych, o tyle nie zajmuje się ona tym, jakie dane i gdzie należy przesłać, w jakim trybie, itp. Zadaniem warstwy kontrolera połączeń jest właśnie sterowanie parametrami interfejsu radiowego i przygotowanie danych do nadania. Natomiast warstwy wyższe poza

sterowaniem przesyłają do tej warstwy już przygotowane dane asynchroniczne jak i synchroniczne.

Główne funkcje za które odpowiedzialna jest ta warstwa to: [4]

- synchronizacja,
- zestawienie połączenia (zapytanie i przywołanie),
- wybór częstotliwości przeskoków,
- typ połączenia SCO, ACL,
- sterowanie dostępem do medium: przepytywanie (typ pakietu i ich przetwarzanie),
- tryby oszczędzania energii,
- algorytmy zabezpieczeń.

### 1.4.1 Architektura master - slave

Sieć Bluetooth działa w oparciu o architekturę **master-slave.** Oznacza to, że kiedy dwa urządzenia zestawiają połączenie, jedno z nich przyjmuje rolę urządzenia nadrzędnego (ang. master), a drugie pełni rolę urządzenia podrzędnego (ang. slave). Specyfikacja stwierdza, że dowolne urządzenie może pełnić, którąkolwiek z tych ról. Zazwyczaj jest tak, że rolę urządzenia master przyjmuje urządzenie inicjujące połączenie. W pewnych specyficznych sytuacjach role urządzeń mogą ulec zamianie. Specyfikacja definiuje, że w danej podsieci może być jedno urządzenie typu master pełniące funkcje zarządcy i co najwyżej siedem urządzeń typu slave, które są połączone z danym urządzeniem master.

Transmisja radiowa Bluetooth odbywa się, podobnie jak transmisja USB, na zasadzie odpytywania jednostek slave przez jednostkę master. Oznacza to, że urządzenie w trybie slave nie może samo zainicjować przesyłania danych, tylko musi czekać, na pozwolenie nadawania.

Zadania urządzenia typu master są następujące:

- synchronizowanie komunikacji pomiędzy urządzeniami,
- określenie reguły przeskoków po częstotliwościach ,
- określenie do którego urządzenie slave przesyłane są pakiety i które urządzenie slave może nadawać jako następne.

Topologia typu master-slave ma znaczenie tylko dla protokołów niższych warstw. Dla protokołów warstw wyższych komunikacja przebiega na zasadzie punkt-punkt (ang. point to point) i właściwie zasady panujące w niższych warstwach są przezroczyste.

### 1.4.2 Pikosieci

W danej sieci może być tylko jedno urządzenie zarządzające master i co najwyżej siedem urządzeń aktywnych typu slave. Taką sieć nazywamy **pikosiecią.** Wszystkie urządzenia pikosieci są ze sobą zsynchronizowane i wszystkie jednocześnie zmieniają częstotliwość pracy. Poza aktywnymi urządzeniami typu slave w danej sieci mogą pracować urządzenia synchronizujące się z urządzeniem master, pozostające w trybie parkowania i w zależności od sytuacji mogą stać się urządzeniami aktywnymi. Ogólna liczba wszystkich urządzeń pozostających w stanie synchronizacji nie może przekroczyć 255.

Urządzenia typu slave mogą komunikować się tylko z urządzeniami typu master i nie mogą komunikować się ze sobą nawzajem bez pośrednictwa urządzenia master.

### 1.4.3 Stany pracy urządzenia

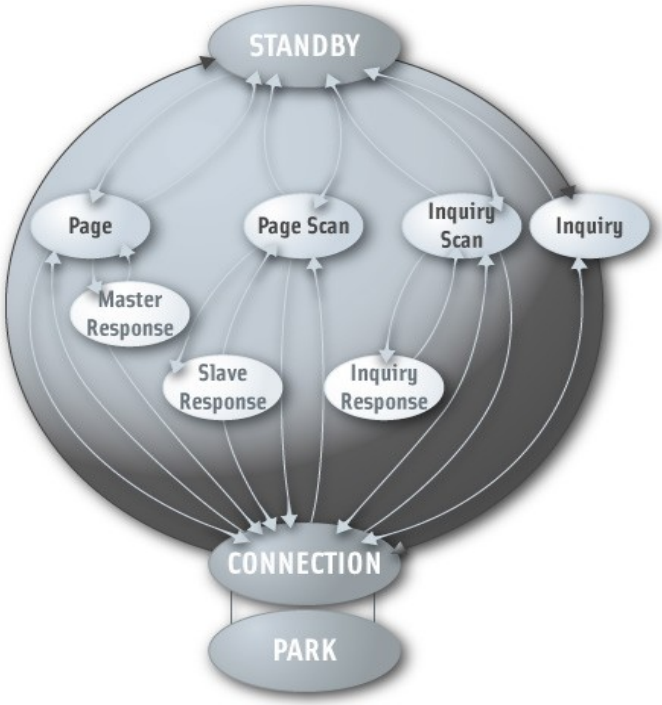
Kontroler połączenia ze względów funkcjonalnych jak i ze względu na ograniczone zasoby energii może wprowadzić urządzenie w następujące stany:

- **CONNECTION** – tryb połączenia, urządzenia w tym trybie mogą wymieniać między sobą dane,
- **STANDBY** – tryb oczekiwania na połączenie,
- **PARK** – tryb parkowania umożliwiający oszczędzanie energii.

Jak można zaobserwować na rysunku 4 poza stanami podstawowymi istnieją również stany pośrednie służące do nawiązania połączenia i umożliwienia odszukania urządzenia. Dodatkowo na rysunku zaznaczone są stany odpowiedzi urządzeń.

Wyróżniamy następujące stany pośrednie:

- **Page** – stan połączenia; urządzenie w tym stanie nawiązuje połączenie z innym urządzeniem.
- **Page Scan** – stan nasłuchiwania na połączenie; urządzenie w tym stanie pozwala, aby inne urządzenia łączyły się z nim. Ten stan umożliwia połączenie tylko tym urządzeniom, które znają adres fizyczny urządzenia będącego w tym stanie.
- **Inquiry** – stan przeszukiwania; urządzenie w tym stanie szuka innych urządzeń, które są widoczne, a których adresów fizycznych nie zna.
- **Inquiry Scan** – nasłuchiwanie zapytań; urządzenia będące w tym stanie pozwalają znajdować się innym urządzeniom.
- **Master Response, Slave Response, Inquiry Response** – odpowiedzi innych urządzeń sieci na przejście urządzenia w określony stan.




Rysunek 4: Graf przejść stanów kontrolera połączeń [23]

### 1.4.4 Adres urządzenia

Każde urządzenie Bluetooth, podobnie jak urządzenie sieciowe innych architektur, ma swój unikatowy adres przypisany przez producenta. Adresy te są nadawane przez niezależną jednostkę zarządzająca adresami IEEE i są one unikalne w skali świata. Adres ten jest 48-bitowy i dzieli się na trzy części:

- Nie oznaczona część adresu **NAP** (ang. Non-significant Address Part),
- Górna część adresu **UAP** (ang. Upper Address Part),
- Dolna część adresu **LAP** (ang. Lower Address Part).

Części UAP i NAP (łącznie 24-bity) stanowią łącznie *unikatowy adres organizacji* OUI (ang. Organizational Unique Identifier). Jest to część adresu przyznana organizacji przez jednostkę zarządzającą adresami. Kolejne 24-bity przydziela organizacja we własnym zakresie trzymając się zasady unikalności adresu. [2]

## 1.5 Warstwa menadżera połączenia

Menadżer połączenia służy do negocjowania z menadżerem innego urządzenia wszystkich parametrów połączenia. Dokładne role jakie pełni ta warstwa pokrywają się z listą komunikatów jakie przesyła do innego urządzenia i najogólniej można je podzielić na kilka grup: [23]

1) Zarządzanie połączeniem:
   - nawiązywanie połączeń,
   - zrywanie połączeń,
   - zarządzanie mocą,
   - zarządzanie algorytmem AFH (ang. Adaptive Frequency Hopping),
   - zarządzanie jakością usług QoS (ang. Quality of Service),
   - zarządzanie rozszerzeniem EDR (ang. Enhanced Data Rate).

2) Zarządzanie zabezpieczeniami:
   - autoryzacja,
   - parowanie urządzeń,
   - wymiana kluczy transmisji,
   - szyfrowanie,
   - bezpieczne parowanie urządzeń (ang. Secure Simply Pairing).

3) Pobieranie informacji:
   - dokładność zegara,
   - różnica zegarów w urządzeniach master i slave,
   - wersja LMP (ang. Link Manager Protocol),
   - wspierane funkcje,
   - nazwa urządzenia,

4) Tryby pracy urządzenia:
   - Sniff Mode - tryb przeszukiwania,
   - Hold Mode - tryb wstrzymania,
   - Park Mode - tryb parkowania.

### 1.5.1 Zarządzanie mocą

Urządzenia Bluetooth pracujące w sieci poza zwykłym trybem pracy mogą pozostawać w jednym z trzech opcjonalnych trybach oszczędzania energii:

- Sniff Mode - tryb przeszukiwania,
- Hold Mode - tryb wstrzymania,
- Park Mode - tryb parkowania.

**Sniff Mode** – polega na zmniejszeniu częstotliwości z jaką urządzenie slave musi nasłuchiwać czy dane, które wysyła urządzenie master nie są kierowane do niego. Komunikacja w tym trybie zachodzi w sposób okresowy, jak zwykła komunikacja z tą różnicą, że okres się wydłuża, kiedy urządzenie jest w tym trybie. Dzięki temu urządzenie dłużej może przebywać w stanie uśpienia pomiędzy kolejnymi trybami nasłuchu.

**Hold Mode** – w tym trybie wymiana danych z urządzeniem master zostaje wstrzymana na określony przez urządzenie czas zwany czasem wstrzymania.

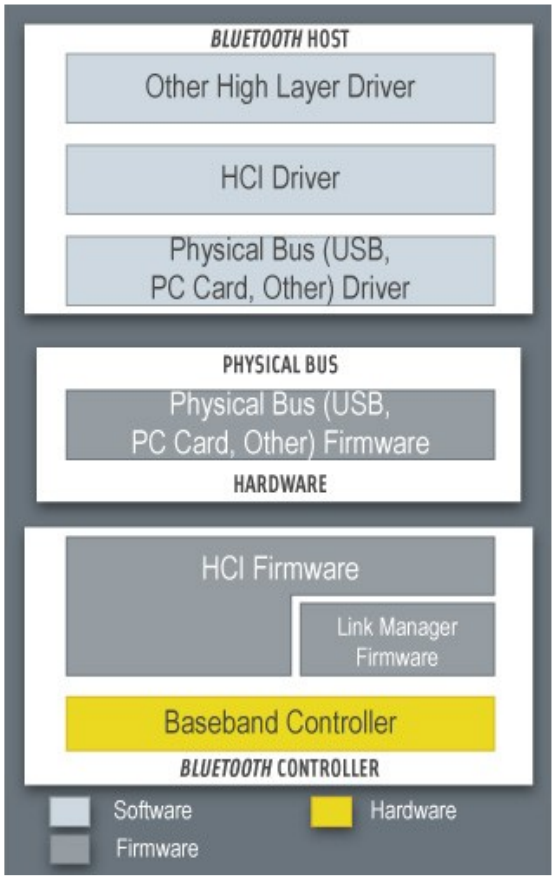
**Park Mode** – urządzenia pozostające w dwóch powyższych trybach pracy są nadal pełnoprawnymi uczestnikami sieci. Oznacza to, że nadal mają przydzielony adres aktywnego elementu pikosieci i są tej sieci częścią. W trybie parkowania natomiast urządzenia przechodzą w tryb nieaktywny w którym przestają być członkiem pikosieci, ale podtrzymują synchronizacje z urządzeniem master. Dzięki temu mogą łatwo stać się znowu członkiem pikosieci bez potrzeby realizowania procedur przeszukiwania i przywołania.

## 1.6 Warstwa HCI

**Host Controller Interface** jest interfejsem komunikującym adapter Bluetooth z komputerem, do którego jest podłączony. Służy do przesyłania danych między nimi i jednolitego zarządzania warstwą menadżera połączeń i warstwą pasma podstawowego z poziomu komputera hosta. Słowo „jednolitym” oznacza, że w przeciwieństwie do innych urządzeń sieciowych i ogólnie urządzeń podłączanych do komputera, urządzenia Bluetooth nie wymagają oddzielnego sterownika dla urządzeń od różnych producentów i dla różnych modeli. Wystarczy jeden sterownik, aby wszystkie urządzenia podłączone do komputera działały. Warstwa HCI jest więc jednolitym interfejsem do komunikacji na poziomie: dowolny adapter – komputer. SIG od samego początku przywiązywała wiele uwagi do tego, aby standard był jak najbardziej elastyczny i otwarty. Należy przyznać, że rozwiązanie problemu sterowników z jakim borykają się użytkownicy innych standardów, bardzo się do tego przyczyniło. Ponieważ istnieją cztery rodzaje połączenia adapterów do komputera, więc istnieją też cztery rodzaje sterowników dla tych urządzeń. Są to następujące porty:

- USB,
- UART,
- SD (ang. Secure Digital),
- trzy-żyłowy UART.

Rysunek 5 ilustruje dokładne umiejscowienie warstwy HCI w stosie protokołów oraz fizyczne miejsce implementacji niższych warstw protokołów. [1]




Rysunek 5: Umiejscowienie warstwy HCI w stosie protokołów [23]

## 1.7 Warstwa L2CAP

Niższe warstwy protokołów transportowych były tak projektowane, aby zapewnić niski pobór energii i jak najbardziej zredukować koszty adaptera Bluetooth odpowiedzialnego za funkcje radiowe, przy jednoczesnym zapewnieniu wymogów bezpieczeństwa.

Takie projektowanie wymagało wprowadzenia warstwy adaptacyjnej zaimplementowanej w komputerze hosta, która zapewniła by wyższy poziom abstrakcji w postaci możliwości przenoszenie pakietów o większym rozmiarze niż ten dostępny w paśmie podstawowym oraz zapewniła by obsługę łączy na poziomie logicznym. Takie zadanie pełni warstwa L2CAP (ang. Logical Link Control and Adaptation Protocol), która ukrywa szczegóły niższych warstw transportowych przed wyższymi warstwami stosu protokołów.

Zadania tej warstwy można podzielić na kilka kategorii:

- kanały i zwielokrotniania protokołów – pozwala współbieżnie koegzystować wielu protokołom warstw wyższych,
- segmentacja i składanie pakietów - pozwala protokołom wyższych warstw przesyłać dane długości nawet do 64 kB,
- wymiana informacji dot. jakości usług QoS.

### 1.7.1 Kanały i zwielokrotnianie protokołów

Istnienie wielu jednoczesnych połączeń wykorzystujących jedno urządzenie Bluetooth wymaga mechanizmu, który pozwoliłby rozróżnić, do którego protokołu warstwy wyższej

należy przesłać dane przychodzące z warstw niższych. Mechanizm taki, stanowiący integralną część warstwy L2CAP nazywano kanałami. Podczas nawiązywania połączenia w warstwie L2CAP negocjowane są numery kanału po obu stronach transmisji i w tak utworzonym kanale przesyłane są dane.

Aby wiele różnych protokołów warstw wyższych mogło korzystać z kanałów musi istnieć jeszcze możliwość rozróżniania protokołu, do którego dany kanał będzie się odnosił. Podczas tworzenia kanału podaje się specjalne pole zwane PSM (ang. Protocol and Service Multiplexer), które jest odpowiedzialne za rozróżnienie nie tylko, jak początkowo projektowano, różnych rodzajów protokołów, ale również różnych implementacji konkretnego protokołu. Najogólniej można to porównać do mechanizmu numerów portu protokołu TCP, w którym na różnych portach dostępne są różne usługi i różne implementacje tej samej usługi mogą mieć przypisane różne porty.

Podobnie jak w protokole TCP zakres numerów jakie może przyjmować pole PSM podzielony jest na dwa przedziały adresów:

- wartości od 1 do 1000 - blok zarezerwowanych adresów, opisujący dobrze znane protokoły,
- wartości powyżej 1000 do 65535 - blok swobodnego wykorzystywania – do implementacji jeszcze nie przypisanych nowych protokołów i wielokrotnych implementacji danych protokołów warstw wyższych.

Należy jeszcze zauważyć, że pole PSM może przyjmować tylko wartości nieparzyste.

Przykładowe wartości PSM dla znanych protokołów to: 0x001 – SDP czy 0x003 - RFCOMM

## 1.8 Protokół SDP

SDP (ang. Service Discover Protocol) jest, jak wskazuje nazwa, protokołem wyszukiwania usług, który umożliwia aplikacjom odkrywanie, jakie usługi są dostępne w wykrytym urządzeniu Bluetooth. SDP jest więc rejestrem wszystkich usług, jakie są dostępne w danym urządzeniu i protokołem, który umożliwia elastyczny i prosty dostęp do tego rejestru. Sens istnienia takiego protokołu związany jest z tym, że sieci Bluetooth są sieciami typu ad hoc i ich dynamiczne i spontaniczne tworzenie się wymaga mechanizmu niezależnego od człowieka znajdowania i samokonfigurowania się usług, jakie są dostępne w danej chwili w sieci.

Typowy scenariusz wykorzystania SDP przedstawia się następująco:

- serwer udostępniający usługę w urządzeniu Bluetooth dodaje do rejestru SDP informacje o usłudze,
- klient wyszukujący konkretną usługę przeszukuje wszystkie dostępne urządzenia (lub konkretne urządzenie) i odpytuje serwer SDP, czy dane urządzenie nie udostępnia poszukiwanej usługi,
- jeśli urządzenie udostępnia daną usługę, to kolejne zapytanie do serwera SDP dotyczy szczegółów szukanej usługi.

W rejestrze SDP przechowywane są następujące informacje: [4]

- informacje o klasie usługi (np. audio, faks, wymiana plików),
- informacje o warstwach stosu protokołów, na których bazuje dana usługa,
- informacje czytelne dla użytkownika,
- informacje specyficzne dla danej klasy usług.

Rekordy usług składają się z :

- uniwersalnych atrybutów usług, jest to część charakterystyczna dla wszystkich rodzajów usług,
- indywidualnych atrybutów usług, czyli część charakterystyczna dla określonej klasy usług.

Warto zauważyć, że z protokołu SDP korzystają tylko aplikacje pisane specjalnie dla technologii Bluetooth.

Przykładowy opis usługi wymiany plików zwrócony przez serwer SDP dla telefonu komórkowego zamieszczono na rysunku 6 (po znakach // zamieszczono komentarz) :

```
Service Name: FTP                              // nazwa usługi czytelna dla człowieka

Service RecHandle: 0x10009      // numer usługi w rejestrze SDP (są rumerowane od
0x10000)
Service Class ID List:
  "OBEX File Transfer" (0x1106)
Protocol Descriptor List:        // stos protokołów potrzebny używanych przez usługę
  "L2CAP" (0x0100)
  "RFCOMM" (0x0003)
    Channel: 1      //  konkretny  kanał  protokołu  RFCOMM  na  jakim  dostępna  jest
usługa
  "OBEX" (0x0008)
Language Base Attr List:       // dane specyficzne dla usługi
  code_ISO639: 0x656e
  encoding:    0x6a
  base_offset: 0x100
Profile Descriptor List:
  "OBEX File Transfer" (0x1106)
    Version: 0x0100
```

Rysunek 6: Rejestr usługi OBEX FTP zwrócony przez serwer SDP telefonu komórkowego

Klient pobierający z urządzenia tego typu dane dotyczące określonej usługi nie musi pytać o nie użytkownika i dzięki temu sieci Bluetooth właściwie same się konfigurują i ograniczają interakcje z użytkownikiem do niezbędnego minimum. Jest to bardzo wygodne i elastyczne rozwiązanie, które wpływa w ogromnym stopniu na dużą popularność tej technologii.

## 1.9 Protokół RFCOMM

Celem powstania architektury Bluetooth było zastąpienie kabli plączących się wokół komputera kanałami radiowymi. W czasach powstawania pierwszych wersji specyfikacji najpopularniejszym złączem był port szeregowy RS-232 czyli COMM.

Emulacją portu szeregowego zajmuje się warstwa RFCOMM (ang. Radio Frequency Communication), przypominająca standardowy interfejs szeregowy i pozostająca kompatybilna z RS-232. Sama nazwa RFCOMM jest połączeniem skrótów RF- radiowy i COMM - wirtualny port szeregowy.

Taka kompatybilność wstecz służy temu, aby już istniejące aplikacje korzystające z połączeń kablowych mogły działać w oparciu o sieć Bluetooth bez żadnych zmian w kodzie programu.

Słusznie uważano, że to nowa technologia powinna dostosować się do już istniejących standardów, aby osiągnąć popularność. Wprowadzenie tego protokołu okazało się ogromnym sukcesem, ponieważ stał się on najbardziej popularnym protokołem pośredniczącym, na którym oparte były prawie wszystkie profile w wersji 1.1 specyfikacji. Przyczynił się do tego również brak zaprojektowanego dla Bluetooth protokołu współpracy z sieciami w wersji 1.1 specyfikacji. Dopiero z wersjami 1.2 i 2.0 specyfikacji dodano protokół enkapsulacji sieci LAN: BNEP i protokoły Audio/Video AVCTP i AVDTP, które odciążyły w pewnym stopniu RFCOMM.

RFCOMM opiera się na warstwę L2CAP i może być zarówno bezpośrednio używany przez aplikacje, jak i nadbudowany innymi protokołami specjalistycznymi, które działały wcześniej w porcie szeregowym, jak np. OBEX, AT-commands czy PPP.

### 1.9.1 Kanały RFCOMM

RFCOMM jest wykorzystywany przez wiele profili, pojawia się więc potrzeba podobnie jak w warstwie L2CAP zwielokrotniania łączy tak, aby jedno połączenie RFCOMM mogło być współdzielone przez wiele aplikacji. Mechanizm ten w tej warstwie zwany jest kanałami. Polega na nadawaniu kolejnym profilom korzystającym z protokołu numerów kanału, przez który ich aplikacje mogą się łączyć. Kanały mogą przyjmować wartości od 1 do 30. W listingu w poprzednim podpunkcie widać, że w stosie protokołów usługi wymiany plików nie tylko wymieniony jest protokół RFCOMM, ale również dodana jest informacja, w którym kanale tego protokołu dostępna jest dana usługa :

```
"RFCOMM" (0x0003)
Channel: 1
```

## 1.10 Protokoły współpracy z IrDA (ang. *Infrared Data Association*)

Drugim celem stworzenia sieci Bluetooth, poza wyeliminowaniem zbyt dużej ilości kabli, było zastąpienie przestarzałego i mało elastycznego standardu IrDA. Należało więc poza emulacją portu szeregowego wprowadzić również warstwę, która umożliwiła by działania aplikacji napisanych specjalnie dla sieci IrDA. Ważniejszym jednak powodem adoptowania pewnych rozwiązań z architektury IrDA był fakt, że wiele funkcji, które oferowała ta architektura, miało być od samego początku dostępne w sieciach Bluetooth. Tak więc zamiast opracowywać pewne elementy od nowa adoptowano już dobrze znane i sprawdzone rozwiązania do nowej sieci.

Konkretnie elementy jakie adoptowano to protokół OBEX (ang. OBject Exchange) oraz protokoły, jakie na nim bazują. OBEX to protokół służący do wymiany plików oraz różnych innych obiektów. Protokół ten bazuje bezpośrednio na warstwie RFCOMM.

Przykładami wykorzystania tego protokołu (szerzej omówionymi w podpunkcie 1.11 „Profile Bluetooth”) są: [8]

- Generic Object Exchange Profile,
- Object Push Profile,
- File Transfer Profile,
- Synchronization Profile,
- Basic Imaging Profile,
- Basic Printing Profile.

Wymienione profile zastosowań są profilami dla sieci IrDA i jak się okazało wszystkie zostały

zaadoptowane do sieci Bluetooth.

## 1.11 Profile Bluetooth

Opisywane w tym punkcie profile są scenariuszami wykorzystania architektury Bluetooth. Odpowiadają roli, jakie może spełniać dane urządzenie i funkcjonalność jaką dostarcza. Urządzenia korzystające z sieci Bluetooth mogą pełnić bardzo różne role, dlatego każdy profil może być implementowany niezależnie w zależności od preferencji producenta. Poszczególne profile są opisywane w niezależnie rozwijanych dokumentach, co podkreśla ich odrębność i daje producentom dużą swobodę ich wdrążania. Jest to również związane z tym, że Bluetooth od samego początku projektowano tak, aby była bardzo elastyczna i mogą sprawdzić się w jak największej liczbie sytuacji. Same profile służą zapewnieniu kompatybilności między aplikacjami. Oznacza to, że aby dwa urządzenia mogły się ze sobą porozumieć muszą posiadać przynajmniej jeden wspólny profil.

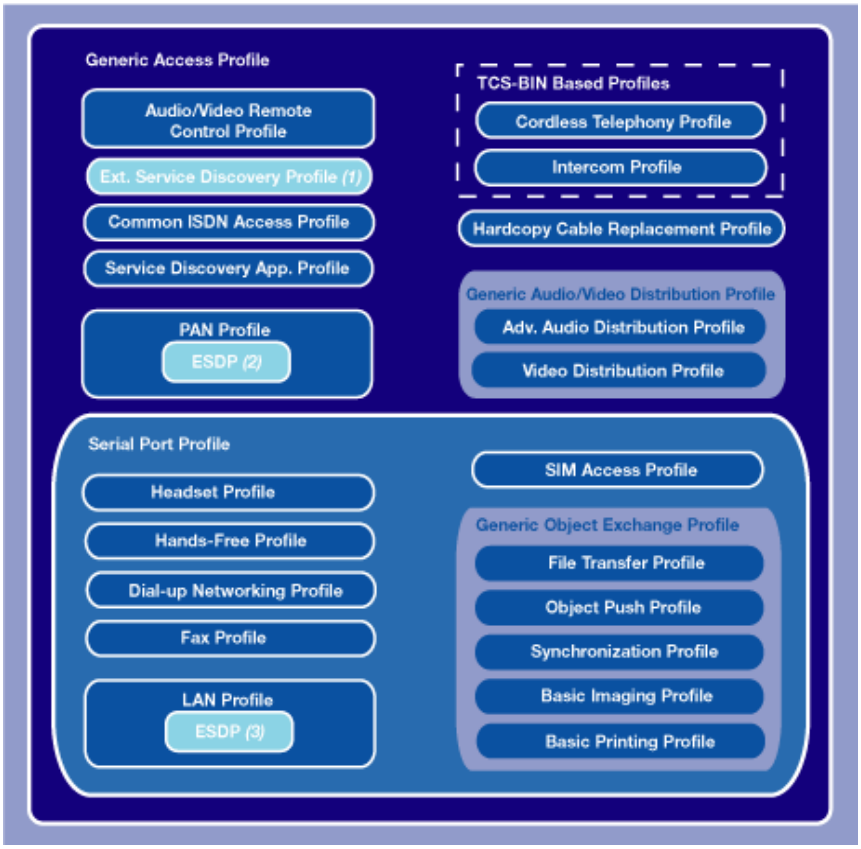
Właściwie jedynymi profilami, które muszą być zaimplementowane w każdym urządzeniu Bluetooth, są profile ogólnego przeznaczenia.

Poniżej zostanie przedstawiona lista dostępnych profili i krótki opis każdego z nich.

Profile według pełnionej funkcji możemy podzielić na:

- profile ogólnego przeznaczenie,
- profile emulacji portu szeregowego,
- profile telefonii,
- profile sieciowe,
- profile audio/video,
- profile wymiany plików i innych obiektów.

Rysunek 7 pokazuje wzajemne zależności między profilami. [22]



Rysunek 7: Wzajemne zależności między profilami [22]

Profile pierwotne dostępne wraz z wersją 1.1 standardu [23][2] to:

- **GAP (ang. Generic Access Profile)** - profil ogólnego dostępu

  Jest to podstawowy profil, na którym opierają się wszystkie inne. Nie odnosi się on do żadnego modelu zastosowań. Opisuje on elementy komunikacji konieczne, aby nawiązać połączenie pomiędzy dwoma urządzeniami Bluetooth.

  Elementy komunikacji to:

  - wyszukiwanie urządzeń i usług,
  - zestawienie i konfiguracja połączeń,
  - zagadnienia bezpieczeństwa.

- **SDAP (ang. Service Discovery Application Profile)** - profil aplikacji wyszukiwania usług

  Jest to profil oparty bezpośrednio na protokole SDP i umożliwiający innym urządzeniom dostęp do informacji o usługach jakie oferuje dane urządzenie.

- **CTP (ang. Cordless Telephony Profile)** - profil telefonii bezprzewodowej

  Profil rozszerzający funkcje zwykłego telefonu komórkowego o możliwość pracy jako radiotelefon bliskiego zasięgu, komunikujący się ze stacją bazową, np. po wejściu do domu przejmuje role bezprzewodowej słuchawki telefonu stacjonarnego.

- **ICP (ang. Intercom Profile)** - profil bezprzewodowej komunikacji

  Profil umożliwiający bezpośrednie połączenie głosowe z innym urządzeniem, czyli pełnienie funkcji tak zwanego „walkie-talkie” lub krótkofalówki.

- **SPP (ang. Serial Port Profile)** - profil wirtualnego portu szeregowego

  Profil emulujący port szeregowy oparty bezpośrednio na protokole RFCOMM, stanowiący podstawę dla wymienionych poniżej profili.

- **HSP (ang. Headset Profile)** - profil bezprzewodowego zestawu słuchawkowego.

  Profil umożliwiający łączność dla bezprzewodowego zestawu słuchawkowego.

- **DUN (ang. Dial-up Networking Profile)** - profil sieci komutowanych (dostęp do modemu).

  Profil bazujący na protokole PPP umożliwiający bezprzewodowy dostęp do sieci poprzez modem.

- **FAX (ang. Fax Profile)** - profil usług telefaksowych, umożliwia bezprzewodowy dostęp do faksu.

- **LAP (ang. LAN Access Profile)** – profil bezprzewodowego dostępu do sieci lokalnej.

  Profil umożliwia bezprzewodowy dostęp do sieci LAN.

- **GOEP (ang. Generic Object Exchange Profile)** - ogólny profil wymiany danych w postaci obiektów.

  Profil ten oparty na protokole OBEX stanowi bazę dla wymienionych poniżej profili wymiany obiektów.

- **OPP (ang. Object Push Profile)** - profil wymiany obiektów.

  Profil umożliwiający przesyłanie różnego typu obiektów takich jak pliki, wizytówki itp.

- **FTP (ang. File Transfer Profile)** - profil przesyłania plików.

Profil pełni identyczną funkcje jak protokół FTP w sieciach TCP/IP, czyli umożliwia operacje na plikach innego urządzenia.

- **SP (ang. Synchronization Profile)** - profil synchronizacji danych.

  Profil umożliwiający synchronizacje danych osobistych między dwoma urządzeniami.

Wraz z wersjami 1.2 i 2.0 specyfikacji pojawiły się zarówno nowe profile, które nie zostały ukończone na czas publikacji wersji 1.1, a także zastąpiono niektóre z pierwotnych profili przez inne lepiej spełniające swoją funkcję wersje.

- **ESDP (ang. Extended Service Discovery Profile)** – rozszerzony profil odkrywania usług

  Profil rozszerza możliwości profilu SDAP o współpracę z Plug and Play

- **PAN (ang. Personal Area Networking Profile)** - profil dostępu do sieci osobistej

  Profil bazujący na protokole BNEP, który zastępuje profil LAP. Profil umożliwia dostęp do sieci LAN, oraz pozwala urządzeniom formować sieci osobiste w trybie ad hoc.

- **HCRP (ang. Hard Copy Cable Replacement Profile)** – profil dostępu do drukarki

  Profil ten służy do zdalnego dostępu do drukarki z wykorzystaniem sterowników. Od opisywanego poniżej profilu prostego drukowania różni się właśnie tym, że zastępuje tylko połączenie kablowe. Za wszystkie operacje drukowania odpowiedzialny jest sterownik zainstalowany w komputerze. Profil ten nadaje się więc do wykorzystanie przez wszelkiego typu urządzenia, na których można zainstalować sterowniki od producenta.

- **CIP (ang. Common ISDN Access Profile)** - profil wspólnego dostępu do sieci ISDN.

  Profil umożliwiający aplikacjom współpracującym z ISDN na bezprzewodowy dostęp bez modyfikacji aplikacji.

- **AVRCP (ang. Audio/Video Remote Control Profile)** - profil zdalnego sterowania urządzeniami audio/wideo.

  Profil docelowo ma służyć do zastąpienia wszelkich pilotów działających na podczerwień przez jeden uniwersalny pilot Bluetooth mogący sterować wszystkimi urządzeniami Audio i Wideo.

- **GAVDP (ang. Generic Audio/Video Distribution Profile)** - profil ogólny dystrybucji audio/wideo.

  Profil definiujący ogólne zasady dystrybucji audio i wideo. Stanowi podstawę dla dwóch opisanych niżej profili.

- **A2DP (ang. Advanced Audio Distribution Profile)** - profil zaawansowanej dystrybucji audio

  Profil zastępujący HSP, który był ograniczony do łącza o przepustowości 64 kb/s i nie nadawał się do użycia bezprzewodowej słuchawki do słuchania muzyki ze względu na zbyt małą przepustowość i brak możliwości przesyłania muzyki w kanałach stereo.

- **VDP (ang. Video Distribution Profile)** – profil dystrybucji strumienia wideo.

  Profil umożliwiający przesyłanie strumieni wideo przez łącze bezprzewodowe.

- **BIP (ang. Basic Imaging Profile)** - profil wymiany obrazów.

  Profil służący do przesyłania obrazów między urządzeniami.

- **BPP (ang. Basic Printing Profile)** - profil prostego drukowania.

  Profil drukowania, bez użycia sterowników do drukarki. Profil zaprojektowano, aby proste urządzenia takie jak telefony, czy PDA mogły bezprzewodowo sterować

drukarkami.

- **HID (ang. Human Interface Device Profile)** - profil komunikacji z urządzeniami HID.

  Profil ten podobnie jak interfejs USB HID definiuje zasady współpracy sieci bezprzewodowej z urządzeniami wejściowymi takimi jak mysz, klawiatura lub inne urządzenia typu HID.

- **HFP (ang. Hands-Free Profile)** - profil „wolnych rąk”

  Profil wykorzystywany przez zestawy hands-free do sterowanie funkcjami określonego urządzenia mobilnego podłączonego przez sieć bezprzewodową. Zazwyczaj jest to sterowanie telefonem komórkowym w samochodzie, a dokładnie takie funkcje jak odbieranie rozmów, sterowanie głośnością itp.

- **SAP (ang. SIM Access Profile)** - profil dostępu do karty SIM.

  Profil umożliwiający zdalny dostęp do kart SIM dostępnych w telefonach komórkowych. Zazwyczaj wykorzystywany w samochodach, aby wbudowane w nie zestawy GSM miały dostęp karty SIM telefonu komórkowego kierowcy.

## 1.12 Wersje specyfikacji Bluetooth i przegląd zmian

Standard Bluetooth ma już dziesięć lat. Na początku 1998 roku powstała grupa Bluetooth SIG będąca pomysłem firmy Ericsson, która już od 1994 badała możliwość stworzenia sieci bezprzewodowej krótkiego zasięgu. Początkowo SIG zrzeszał pięć firm: Ericsson, Nokia, Toshiba, IBM, Intel. Później w miarę rozwoju specyfikacji i jej coraz większej popularności do SIG dołączały kolejne firmy. [3][4]

### 1.12.1 Ewolucja specyfikacji

Specyfikacja techniczna od wersji 1.1 podzielona jest na dwie części. Pierwszą część stanowi opis podstawowy standardu (ang. Core Specification), a drugą część tworzy zbiór profili implementowanych według uznania producenta i przeznaczenia urządzenia.

Historia najważniejszych wersji specyfikacji Bluetooth przedstawia się następująco:

- Core Specification 1.0 wydane w 1999 r.
- Core Specification 1.1 wydane w 2001 r.
- Core Specification 1.2 wydany w 2003 r.
- Core Specification 2.0 + ERD wydany w 2004 r.
- Core Specification 2.1 + ERD wydany w 2007 r.

Od wersji 1.1 wszystkie kolejne wersje do wersji 2.1+ERD są kompatybilne wstecz.

Aktualnie obowiązującą specyfikacją jest Bluetooth Core Specification 2.1+ERD

Obecnie wersja 2.0 protokołu podstawowego jest zaimplementowana w większości urządzeń dostępnych na rynku i nowych dostępnych do zakupu, a wersja 2.1 relatywnie nowa jest jeszcze bardzo mało popularna.

Natomiast w trakcie realizacji jest wersja 3.0 specyfikacji w założeniach oparta o warstwę radiową **UWB (ang. Ultra-Wideband)** cechująca się bardzo małym poborem prądu i szybkością transmisji danych do 470 Mb/s. Na tej właśnie warstwie radiowej jest oparta architektura Wireless USB, która jest właśnie w fazie testów. [1][3]

### 1.12.2 Przegląd zmian wprowadzonych w kolejnych wersjach specyfikacji standardu

Według listy zmian w specyfikacji protokołu [1], wyróżniamy następujące zmiany w podstawowej części specyfikacji:

1) Bluetooth 1.1
   - Pierwsza poprawnie działająca w praktyce wersja specyfikacji, która umożliwiła komunikację urządzeń różnych producentów
   - Standard ratyfikowany jako IEEE Standard 802.15.1-2002
   - Poprawki wielu błędów znalezionych w wersji 1.0B
   - Dodano możliwość komunikacji w kanałach nie kodowanych
   - Dodano wskaźnik mocy odbieranego sygnału radiowego **RSSI (ang. Received Signal Strength Indicator)**
2) Bluetooth 1.2
   - Pierwsza wersja specyfikacji kompatybilna wstecz (z wersja 1.1).
   - Standard ratyfikowany jako IEEE Standard 802.15.1-2005.
   - Dodano algorytm **AFH (ang. Adaptive Frequency Hopping)** – umożliwiający unikanie interferencji radiowych poprzez omijanie zatłoczonych częstotliwości podczas skoków po kanałach. Technika ta w praktyce zwiększa szybkość transmisji.
   - Dodano wsparcie warstwy HCI dla połączeń na kablu trzy żyłowym UART.
   - Dodano połączenia typu **eSCO (ang. Extended Synchronous Connections Oriented links),** czyli tryb transmisji audio wysokiej jakości, wspierający retransmisje uszkodzonych danych audio. Umożliwiło to podniesienie jakości transmisji mowy.
   - Dodano czujnik jakości sygnału.
3) Bluetooth 2.0 + EDR
   - Dodano rozszerzony tryb transmisji danych **EDR (ang. Enhanced Data Rate)**, który umożliwił wprowadzenie niżej opisanych zmian.
   - Zwiększenie szybkości transmisji do 2.1 MB/s (lub 3.0 MB/s z wykorzystaniem EDR)
   - Połączenia w praktyce od trzech do dziesięciu razy szybsze poprzez zwiększenie szybkości transmisji i zmniejszenie opóźnienia.
   - Zmniejszono zużycie mocy poprzez redukcję czasu obowiązkowej aktywności.
   - Dodano wsparcie dla połączeń rozsiewczych (ang. broadcast) i grupowych (ang. multicast).
4) Bluetooth 2.1 + EDR
   - Dodano rozszerzenie **Encryption Pause Resume** umożliwiające odświeżanie kluczy transmisji jeśli czas korzystania z danego klucza przekroczy założoną wartość. Umożliwia to dużo lepsze zabezpieczenie transmisji trwającej długi czas.
   - Dodano rozszerzenie **Extended Inquiry Response** dostarczające więcej informacji podczas procesu zapytań (ang. inquiry), co umożliwia filtrowanie urządzeń jeszcze przed połączeniem się z nimi. Te dodatkowe informacje to nazwa urządzenia, lista usług jakie urządzenie wspiera itp.

- Dodano rozszerzenie **Secure Simple Pairing** radykalnie podnoszące bezpieczeństwo procesu parowania poprzez zastosowanie nowych silniejszych algorytmów zapobiegających pasywnemu i aktywnemu podsłuchowi. Rozszerzenie znacząco upraszcza również proces parowania z punktu widzenia użytkownika poprzez dodanie dodatkowych trybów parowania włącznie z trybami uwzględniającymi urządzenie nie posiadające wyświetlacza czy klawiatury. (mechanizm dokładnie opisano w podpunkcie 1.13.6„Secure Simple Pairing")
- Dodano rozszerzenie **Near Field Communication** (NFC) umożliwiające stworzenie automatycznego połączenia szyfrowanego pomiędzy dwoma urządzeniami bez konieczności parowania ich przez użytkownika z wykorzystaniem kodu PIN. Wystarczy zbliżyć do siebie urządzenia na bardzo małą odległość (kilku centymetrów), aby automatycznie się sparowały. Rozszerzenie to jest częścią **Secure Simple Pairing.** Przykład zasady działania tego mechanizmu można obejrzeć pod adresem: [27].
- Dodano rozszerzenie **Sniff Subrating** redukuje zużycie mocy, kiedy urządzenie znajduje się w trybie nasłuchiwania (ang. sniff), szczególnie dla połączeń asymetrycznych. Poprzednie wersje standardu wymuszały na urządzeniach w tym trybie wysyłanie pakietów podtrzymujących połączenie nawet kilka razy w ciągu sekundy. To rozszerzenie umożliwia urządzeniu negocjowanie tej wartości według przeznaczenia urządzenia i co za tym idzie możliwość pozostania w stanie uśpienia o wiele dłużej. Urządzeniami, które skorzystają najwięcej z tego rozszerzenia są produkty HID (ang. Human Interface Devices).

## 1.13 Mechanizmy bezpieczeństwa sieci Bluetooth

W przeciwieństwie do sieci przewodowych, sieci radiowe o wiele łatwiej jest podsłuchać. Dlatego standard Bluetooth od samego początku projektowany był z dużym naciskiem na aspekty bezpieczeństwa i zapewnienia poufności danych, aby jak najbardziej utrudnić potencjalnym włamywaczom zadanie. [2]

W poszczególnych warstwach stosu protokołów wyróżniamy różne mechanizmy zabezpieczeń.

1) Warstwa radiowa - utrudnienie podsłuchu sygnału radiowego przez wykorzystaną metodę nadawania sygnału: skakania po częstotliwościach (FHSS).

2) Pasmo podstawowe - odpowiedzialne jest za mechanizm szyfrowania i tworzenia kluczy. To właśnie w tym paśmie zaimplementowany jest algorytm szyfrowania SAFER+, wykorzystywany do szyfrowania danych i podczas mechanizmu uwierzytelniania. Sterowaniem procesami szyfrowania zachodzącymi w tym paśmie zajmuje się warstwa menadżera połączenia.

3) Menadżer Połączenia - warstwa odpowiedzialna jest za sterowanie procesami bezpieczeństwa takimi jak:
   a) autoryzacja:
      - parowanie urządzeń.
      - wymiana kluczy transmisji.
   b) szyfrowanie

4) Warstwa HCI - ta warstwa udostępnia użytkownikowi możliwość sterowania mechanizmami bezpieczeństwa poprzez wyznaczenie zbioru zasad zabezpieczeń dla komputera hosta i poprzez konfigurowanie zabezpieczeń kontrolera.

5) Generic Access Profile - ogólny profil dostępu również definiuje wiele mechanizmów bezpieczeństwa na swoim poziomie, z których korzystają inne profile Bluetooth. Dokładnie jest to zdefiniowanie modelu zabezpieczeń i zasad kodowania danych.

### 1.13.1 FHSS – Skakanie po częstotliwościach

Działająca w paśmie 2,4 GHz sieć WiFi dzieli je na określoną liczbę kanałów i w zależności od ustawień pracuje na jednym stałym kanale. Daje to możliwość wszystkim urządzeniom w pobliżu nasłuchiwać na tym kanale i podglądać przepływające dane. Wystarczy, że dowolny użytkownik wprowadzi swoją kartę w tryb uprzywilejowany, aby móc przeglądać cały ruch sieciowy. To czy taki użytkownik będzie miał możliwość odszyfrowania i pełnego podglądu danych zależy tylko od użytego algorytmu kodowania. Co jednak nie przeszkadza podglądać nagłówki pakietów czy ogólny ruch w sieci.

W sieciach Bluetooth zagadnienie podsłuchu wygląda zupełnie inaczej. Jak wspomniano wcześniej, w tej sieci urządzenia nie pracują na żadnej stałej częstotliwości, ale zgodnie z algorytmami Frequency Hopping przeskakują po kanałach z bardzo dużą częstotliwością. Sekwencje przeskoków są zależne od urządzenia master danej pikosieci. Są one niedeterministyczne co oznacza, że urządzenia które nie są członkiem pikosieci właściwie nie mają szans na podsłuchanie transmisji radiowej. Dlatego właśnie zwykłego adaptera Bluetooth nie da się wprowadzić w tryb uprzywilejowany i co za tym idzie bez specjalistycznej aparatury, w domowych warunkach nie ma możliwości podsłuchu transmisji Bluetooth.

### 1.13.2 Uwierzytelnienie

Uwierzytelnienie jest procesem kontrolowanym przez warstwę menadżera połączeń, polegającym na sprawdzeniu czy urządzenia przechowują wspólny tajny klucz.

Podczas pierwszego połączenia dwóch urządzeń następuje uwierzytelnienie na podstawie **kodu PIN (ang. Personal Identification Number)**. Kod ten dla urządzenia z możliwością wprowadzania danych jest podawany przez użytkownika, lub jeśli takiej możliwości nie ma zaszyty na stałe w urządzeniu (np. słuchawki bezprzewodowe, które nie mają żadnego interfejsu wejściowego, którym można by wprowadzić kod). Jeśli w obu urządzeniach podany zostanie ten sam kod to następuje uwierzytelnienie. Po uwierzytelnieniu urządzania mogą stworzyć tak zwany wspólny ***klucz połączenia*** (ang. link key) służący do ponownego uwierzytelnienia urządzeń bez pośrednictwa użytkownika. Proces uwierzytelnienia i stworzenia klucza połączenia nazywany jest **parowaniem.**

Cała procedura uwierzytelnienia dwóch urządzeń wygląda więc tak, że najpierw urządzenia sprawdzają czy nie są już sparowane i nie dysponują wspólnym kluczem połączenia. Jeśli tak, to urządzenia uznają się nawzajem za uwierzytelnione. W przeciwnym razie następuje proces weryfikacji z udziałem kodów PIN wprowadzanych przez użytkownika.

Parowanie więc jest jednorazowe i wymaga udziału użytkownika. Sparowane urządzenia ufają sobie nawzajem. Użytkownik jednego z urządzeń może jednak w dowolnym momencie odwołać uwierzytelnienie, co wymusi konieczność ponownego parowania przed połączeniem.

### 1.13.3 Szyfrowanie

O ile uwierzytelnienie urządzeń w większości przypadków jest obowiązkowe, o tyle szyfrowanie transmitowanych przez warstwę radiową danych jest kwestią wyboru. Sam proces szyfrowania danych następuje w warstwie pasma podstawowego, a kontrolę nad tym

procesem sprawuje warstwa menadżera połączeń.

Aby zaszyfrować dane konieczne jest wcześniejsze przeprowadzenie uwierzytelnienia, ponieważ **klucz szyfrujący** jest generowany na podstawie wspólnego klucza połączenia współdzielonego przez urządzenia. Szyfrowanie przeprowadzone jest przez opisany dalej algorytm SAFER+ z max 128 bitowym kluczem szyfrującym. Domyślna wielkość klucza szyfrującego 128 bitów może w niektórych przypadkach być zmniejszona w związku z restrykcjami niektórych krajów dot. eksportu silnych algorytmów szyfrujących.

### 1.13.4 Mechanizmy bezpieczeństwa profilu GAP

Profil GAP wprowadza trzy kategorie reguł zestawiania połączeń [4]:

- tryby podatności na wyszukiwanie,
- tryby podatności na zestawianie połączeń,
- tryby parowania.

Wprowadza również tryby zabezpieczeń z jakimi mogą pracować urządzenia.

**Tryby podatności na wyszukiwanie**

Urządzenia Bluetooth w zależności od konfiguracji mogą być widoczne dla innych urządzeń (odpowiadać na zapytania) lub być niewidoczne (ignorować zapytania). W tym drugim trybie urządzenie nadal utrzymuje aktywne połączenia, ale nie pozwala się wyszukać innym urządzeniom, które nie znają jego adresu.

**Tryby podatności na zestawienie połączeń**

Podobnie jak dla podatności na wyszukiwanie urządzenia mogą pozwalać zestawić ze sobą połączenie lub nie. Zabronienie możliwości ustawienia ze sobą połączeń może mieć sens, jeśli urządzenie filtruje to, z kim się łączy.

Oba wymienione wyżej tryby pracy są od siebie niezależne. Co oznacza, że konfiguracja jednego z nich nie wymusza parametrów drugiego.

Różnica między możliwością wyszukiwania, a możliwością połączenia jest taka, że wyszukujemy urządzeń których adresu nie znamy. A połączyć się może tylko z urządzeniem, którego adres się zna. Więc zanim zostanie ustanowione połączenie z jakimś urządzeniem musi ono wcześniej zostać wyszukane.

Operując tymi dwoma trybami można przyczynić się do zwiększenia poziomu bezpieczeństwa urządzenia.

**Tryb parowania**

Urządzenie może zarówno pozwolić na sparowanie ze sobą czyli na autoryzacje i stworzenie klucza połączenie jak i odmówić parowania i co za tym idzie uniemożliwić szyfrowanie.

**Tryby zabezpieczeń**

Ogólny profil dostępu (GAP) definiuje trzy tryby zabezpieczeń:

- **tryb 1 – brak zabezpieczeń**

  Urządzenia pracujące w tym trybie nie muszą ani przeprowadzać procedury weryfikacji ani szyfrować danych. Co prawda w sieciach Bluetooth obowiązkowe jest uwierzytelnienie, ale urządzenie pracujące w tym trybie nigdy nie wyśle prośby o weryfikacje.

- **tryb 2 – bezpieczeństwo na poziomie usług**

  Ten tryb daje usługom i aplikacjom swobodę w negocjowaniu zabezpieczeń. Reguły w tym trybie są negocjowane podczas tworzenia kanału lub połączenia czyli na poziomie L2CAP lub wyższych.

- **tryb 3 – bezpieczeństwo na poziomie łącza danych:**

  - wymuszone uwierzytelnienie
  - wymuszone uwierzytelnienie i szyfrowanie

  Urządzenia pracujące w tym trybie wymuszają zabezpieczenia na poziomie menadżera połączań. Oznacza po, że zabezpieczenia są negocjowane przed stworzeniem kanału i że są identyczne dla wszystkich usług.

  W tym trybie możliwe są dwie konfiguracje. W pierwszym przypadku urządzenie wymusza uwierzytelnienie, a w drugim wymusza zarówno uwierzytelnienie jak i szyfrowanie.

### 1.13.5 Algorytm kodowania SAFER+

Standard Bluetooth do szyfrowania danych i uwierzytelniania urządzeń wykorzystuje algorytm kodowania SAFER+ (ang. Secure And Fast Encryption Routine). Algorytm ten został zaproponowany jako jeden z kandydatów do amerykańskiego standardu AES (ang. Advanced Encryption Standard), który miał zastąpić wysłużony już standard DES (ang. Data Encryption Standard).

Jako wejście dla tego algorytmu przewidziano następujące dane:

- dane do szyfrowania/deszyfrowania,
- adres urządzenia master,
- wartość zegara urządzenia master,
- tajny klucz wspólny dla obu urządzeń.

Oba komunikujące się urządzenia znają adres urządzenia master. Urządzenie slave zna przesunięcie czasowe względem czasu urządzenia master.

W przypadku kodowania danych jako dane do szyfrowania są po prostu wprowadzane informacje które należy przesłać. Klucz szyfrowania jest tworzony na podstawie klucza połączenia obu urządzeń i dlatego między innymi uwierzytelnienie musi poprzedzać szyfrowanie.

W przypadku procedury uwierzytelnienia sprawdzenie, czy oba urządzenia posiadają wspólny klucz poprzez przesłanie go mija się z celem, ponieważ klucz staje się łatwym łupem. Dlatego w takim przypadku generowane są losowe dane wejściowe, które są wysyłane następnie do maszyny proszącej o uwierzytelnienie, która odsyła te dane zakodowane swoim tajnym kluczem. Jeśli odesłane dane są identyczne z danych uzyskanymi z szyfrowania tych danych w urządzeniu weryfikującym to oznacza, że urządzenia współdzielą taki sam klucz tajny.

Wersja algorytmu adaptowana przez SIG operuje na 128 bitowych kluczach. Taką właśnie wielkość mają nie tylko klucze szyfrujące, ale również klucze połączenia i inne klucze tymczasowe.

Klucze połączenia możemy podzielić na:

- *klucze modułowe* – tworzone na podstawie danych tylko z jednego urządzenia. Ich tworzenie wynika z ograniczeń pamięci danego urządzenie i mogą być dzielone z wieloma urządzeniami. Są potencjalnym zagrożeniem bezpieczeństwa.
- *klucze kombinacyjne* – tworzone na podstawie danych z obu urządzeń. Są oddzielne dla każdej pary urządzeń.

### 1.13.6 Secure Simple Pairing

Secure Simple Pairing jest rozszerzeniem wprowadzonym w specyfikacji Bluetooth 2.1, które radykalnie podnosi bezpieczeństwo procesu parowania poprzez zastosowanie nowych silniejszych algorytmów zapobiegających pasywnemu i aktywnemu podsłuchowi. Rozszerzenie znacząco upraszcza również proces parowania z punktu widzenia użytkownika poprzez dodanie dodatkowych trybów parowania włącznie z trybami uwzględniającymi urządzenie nie posiadające wyświetlacza czy klawiatury. [7]

Jak dotąd, pomimo wielu urządzeń wspierających wersję standardu 2.1 dostępnych na rynku, producenci oprogramowania nie zaktualizowali swoich implementacji stosu protokołów Bluetooth i nie dali użytkownikom możliwości skorzystania z funkcji jakie nowy standard oferuje. [24]

Secure Simple Pairing wprowadza następujące nowe tryby parowania: [7]

- Porównywanie cyfr (ang. Numeric Comparison) jest trybem parowania zaproponowanym dla urządzeń, które mają wyświetlacz zdolny wyświetlić sześć cyfr i umożliwiających wybranie opcji „tak” lub „nie”.

  Użytkownik w tym trybie parowania jest proszony o porównanie sześciu cyfr wyświetlających się na obu urządzeniach i stwierdzenie czy są one zgodne. Jeżeli na obu urządzeniach zostanie wybrane „tak” to parowanie zakończy się sukcesem.

  Opisywany tryb pomimo podobieństwa do parowania poprzez podanie kodu PIN jest o wiele bezpieczniejszy w związku z tym, że wspomniane sześć porównywanych cyfr jest generowane przez bardzo silny algorytm i ich znajomość nie ułatwia w żaden sposób podsłuchu transmisji. Metoda ta zabezpiecza również przed atakiem z pośrednikiem.
- Po prostu działa (ang. Just Works) jest trybem parowania przewidzianym dla pary urządzeń z których przynajmniej jedno nie ma wyświetlacza zdolnego wyświetlić sześć cyfr lub klawiatury, na której te cyfry można wprowadzić. Dobrym przykładem jest parowanie telefonu komórkowego i słuchawek bezprzewodowych. Jest to tryb działający w taki sam sposób jak porównywanie numerów z tą różnicą, że użytkownik nie widzi numerów i może tylko zgodzić się na sparowanie lub nie. Tryb ten zabezpiecza więc bardzo dobrze przed podsłuchem, ale nie potrafi zabezpieczyć przed atakiem z pośrednikiem.
- Wprowadzenie kodu (ang. Passkey Entry) jest trybem parowania, w którym jedno urządzenie ma możliwość wyświetlania sześciu cyfr, a drugie ma możliwość tylko wprowadzania cyfr bez możliwości wyświetlania. Przykładem takiej konfiguracji jest komputer i klawiatura bezprzewodowa. W takim wypadku parowanie następuje poprzez wprowadzenie na jednym urządzeniu cyfr wyświetlonych na wyświetlaczu drugiego.

Parowanie powiedzie się, jeśli wprowadzone cyfry odpowiadają tym wyświetlonym. Mechanizm ten zapobiega nie tylko podsłuchowi, ale również zabezpiecza przed atakiem z pośrednikiem.

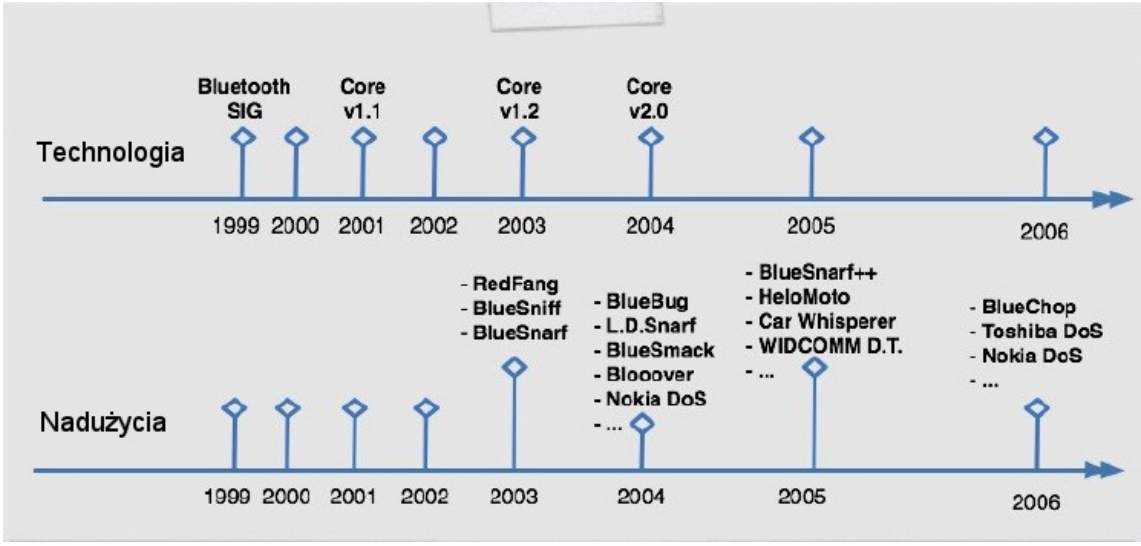
- Poza zakresem (ang. Out Of Band) jest trybem parowania w którym dane potrzebne do tego procesu są przekazywane innym kanałem niż sieć Bluetooth. Przykładem jest mechanizm **Near Field Communication** (NFC) umożliwiający stworzenie automatycznego połączenia szyfrowanego pomiędzy dwoma urządzeniami, zbliżenie do siebie urządzenia na bardzo małą odległość (kilku centymetrów), aby automatycznie się sparowały. Przykład zasady działania tego mechanizmu można obejrzeć pod adresem: [27].

Wszystkie powyższe tryby parowania działają w ten sposób, że urządzenia kiedy się zobaczą same negocjują wszystkie parametry ze sobą, a interakcja użytkownika wymagana jest na końcu w celu potwierdzenia zgodności wyświetlonych na obu urządzeniach cyfr lub po prostu potwierdzenia chęci sparowania. Opisany protokół wykorzystuje podobne mechanizmy jakie są zaimplementowane w standardzie SSL i według specjalistów jest on bezpieczny. [5]

# 2 PODATNOŚĆ SIECI BLUETOOTH NA ATAKI

Celem niniejszego rozdziału jest pokazanie zagrożeń bezpieczeństwa standardu Bluetooth poprzez przegląd najlepiej udokumentowanych luk i sposobów nadużyć odkrytych na przestrzeni lat, od powstania standardu do chwili obecnej. Celem autora nie było zebranie wszystkich znalezionych materiałów w jeden ogromny dokument, tylko wskazanie każdego problemu i odesłanie do odpowiednich dokumentów źródłowych. Na rynku wydawniczym jest bardzo niewiele publikacji książkowych, zawierających tak rozległy przegląd opisywanych zagadnień i nadal aktualnych. Dlatego prawie wszystkie dokumenty źródłowe, z których korzystał autor wskazują na strony internetowe odkrywców luk lub do prezentacji przedstawionych na różnych konferencjach poświęconych sprawom bezpieczeństwa komputerowego.

Przeszukując zasoby Internetu autor przekonał się, że to właśnie konferencje nt. bezpieczeństwa komputerowego i zjazdy hakerów, a nie prasa czy książki są miejscem publikacji znalezionych nowych luk i błędów w oprogramowaniu i sprzęcie. Odkrywcami tych luk są zazwyczaj pracownicy firm zajmujących się tym zagadnieniem w praktyce. Pewna część osób opisanych w tym rozdziale okazała się nawet szefami tych firm i jednocześnie „hakerami" prowadzącymi własne badania. Głównym więc źródłem informacji o nowych, nie publikowanych wcześniej lukach bezpieczeństwa są prezentacje w formie elektronicznej opublikowane przez autorów wystąpień ze wspomnianych konferencji. Coraz częściej, kilka dni po zakończeniu takich spotkań dostępne są również materiały filmowe, które pozwalają dokładnie zapoznać się nie tylko z treścią wystąpienia, ale i z prezentowanymi przykładami. Przeglądając wspomniane materiały można również w miarę chronologicznie odtworzyć odkrycia poszczególnych sposobów ataku, jak i poznać ich autorów. Często wiedza ta jest bardzo pomocna podczas analizy zagadnienia, dlatego w niniejszym rozdziale podano nie tylko opisy ataków, ale również w większości wypadków autorów i datę znalezienia lub publikacji luki. Na ile również było to możliwe zachowano pewien porządek chronologiczny opisanych zagadnień, aby można było prześledzić kolejność odkryć. Rysunek 8 przedstawia historię najważniejszych nadużycia na tle dat publikacji kolejnych wersji standardu. [21]



Rysunek 8: Historia rozwoju standardu Bluetooth i związanych z nim ataków [21]

Znacząca część prezentowanej w tym rozdziale treści wskazuje grupę trifinity.org jako źródło informacji. Grupa ta założona przez Martina Herfurta, odkrywcę luki BlueBug pierwszej poważnej luki bezpieczeństwa Bluetooth, na przestrzeni lat skupiła wokół siebie prawie wszystkich najbardziej aktywnych hakerów zajmujących się zagadnieniem łamania bezpieczeństwa standardu Bluetooth. Jej członkowie to między innymi Marcel Holtmann

obecnie opiekun i główny programista implementacji BlueZ, będącej oficjalnym stosem protokołów standardu Bluetooth dla Linuxa. Adam Laurie, dyrektor The Bunker Secure Hosting Ltd. i bardzo znany haker, twórca między innymi modułu Apache-SSL, który stał się światowym standardem. Wszystkie wspomniane osoby są dodatkowo członkami grupy eksperckiej Bluetooth SIG do spraw bezpieczeństwa tego standardu (Bluetooth SIG Security Experts Group). Grupa trifinity.org do dnia dzisiejszego jest wskazywana na pierwszym miejscu jako źródło informacji na temat luk w sieciach Bluetooth przez niemal wszystkie publikacje i dokumenty poświęcone temu zagadnieniu. Warto również wspomnieć, że według informacji na stronie grupy i ilości dostępnych tam prezentacji jej członkowie występowali aż na kilkunastu konferencjach poświęconych bezpieczeństwu komputerowemu prezentując coraz to nowe znalezione luki i zadziwiając uczestników praktycznymi przykładami włamywania się do urządzeń Bluetooth wykonywanymi na żywo. Grupa ta miała również pokaz w Warszawie na konferencji IT UNDERGROUND w 2005 roku.

## 2.1 Podsłuchiwanie w sieci Bluetooth

### Lokalne podglądanie pakietów w warstwie HCI – hcitools

Podglądanie lokalnych pakietów transportowanych pomiędzy adapterem Bluetooth i komputerem jest relatywnie proste i sprowadza się do podglądania komunikatów protokołu HCI. W systemie Linux można wykorzystać do tego narzędzie hcidump będące częścią pakietu BlueZ. Program ten do działania potrzebuje praw administratora i potrafi dodatkowo dekodować pakiety wszystkich wyższych warstw będących częścią specyfikacji Bluetooth. Ponieważ dane wysyłane do innych urządzeń kodowane i dekodowane są już w adapterze, więc wykorzystując hcidump widać tylko nie zaszyfrowane dane. Program ten umożliwia przychwytywanie kodów PIN i innych poufnych informacji. Metoda ta pozwala oczywiście tylko podglądać dane wymieniane pomiędzy lokalnym komputerem, a innymi urządzeniami w sieci. Nie umożliwia jednak podsłuchu danych wymienianych przez inne urządzenia między sobą.

### Ocena zagrożeń płynących z ataku

Udane przeprowadzenie opisywanych ataków podsłuchu umożliwia dostęp do wszystkich informacji przesyłanych z użyciem sieci Bluetooth włącznie z przesyłanymi plikami jak i możliwość podsłuchu rozmów. Umożliwia również odkrywanie urządzeń ukrytych. Jest to więc bardzo niebezpieczny typ ataków.

## 2.2 Bluejacking

### Opis ataku

Bluejacking nie jest formą ataku, ale sposobem komunikowania się użytkowników telefonów przy pomocy sieci Bluetooth poprzez wysyłanie sobie wiadomości. Chodzi mianowicie o takie przekazanie drugiej osobie wiadomości, aby mogła ją odebrać na urządzeniu nie mającym żadnego przeznaczonego do tego celu oprogramowania. Początkowo wiadomości przekazywano przez pole z nazwą urządzenia Bluetooth. Potem zaczęto używać elektronicznych wizytówek (ang. vCard) przygotowanych w taki sposób, że w nagłówku znajduje się wiadomość. Kolejną zmianą tej metody związaną z rozwojem telefonów komórkowych jest możliwość wysyłania obrazów i plików muzycznych. Forma takiej aktywności uprawiana jest przez użytkowników telefonów komórkowych, którzy przeszukują

dostępne w pobliżu inne widoczne urządzenia Bluetooth. Celem jest zazwyczaj zabawa lub żart.

Terminem Bluejacking często jednak określa się socjotechnikę polegającą na takiej modyfikacji nazwy urządzenia Bluetooth, aby następnie inicjując dostęp do zdalnego urządzenia wymagający parowania, wprowadzić użytkownika w błąd. Na przykład niczego nie podejrzewający użytkownik widzący jako nazwę zdalnego urządzenia przed parowaniem tekst „Wystąpił błąd, wpisz 1234, aby kontynuować" może z dużym prawdopodobieństwem po prostu podążyć za komunikatem. Zwłaszcza, kiedy komunikat powtórzy się kilka razy, albo będzie powtarzał się do czasu uzyskania efektu. Sam tekst przygotowany przez biegłego w sztuce manipulacji hakera może być tak dobrany, że użytkownik nawet nie będzie się zastanawiał co się właściwie dzieje i po prostu pozwoli sparować urządzenie dając hakerowi pełny dostęp do swoich danych.

Kolejną czarną stroną Bluejackingu może być zalewanie użytkowników tak dużą ilością wiadomości (vCards, zdjęcia, muzyka), aby można było mówić o ataku typu DoS czyli uniemożliwieniu normalnego korzystania z zaatakowanego urządzenia.

Plagą dużych miast stają się kioski Bluetooth wykrywające urządzenia przechodniów i wysyłające im spam w postaci materiałów reklamowych i innych tego typu informacji. [30] Sam autor zetknął się jakiś czas temu z wypowiedziami jednego z lokalnych polityków, który po wygranych wyborach chwalił się wykorzystaniem nowoczesnych mediów w „nakłanianiu" obywateli, aby głosowali na niego w wyborach. Jednym ze sposobów wysyłania wiadomości miał być właśnie spot Bluetooth wysyłający do wszystkich znalezionych w pobliżu urządzań materiały promujące spamera.

Warto zauważyć, że firmy korzystające z tego typu praktyk reklamowych opracowały specjalne słowo na określenie tego procederu: "bluecasting". Użytkownicy będący ofiarami takiego postępowania znaleźli inne adekwatne słowo określające opisane proceder: „Bluespamming". [31]

Kolejnym sposobem ataku jest namówienie osoby, której ufa ofiara, aby to ona sparowała urządzenie ofiary i napastnika.

**Podatność na atak**

Na opisywany typ ataków podatni są ludzie nie urządzenia. Jest to więc zbiór nadużyć nie zależnych od konkretnej specyfikacji czy standardu. Ponieważ zwykli użytkownicy zazwyczaj nie przykładają większej uwagi do zagadnień bezpieczeństwa więc statystyczna podatność na opisywane techniki jest bardzo duża.

**Ocena zagrożeń płynących z ataku**

Duża jest również skala zagrożeń płynących z udanego ataku tego typu. Mogą to być zarówno zwykła dezinformacja i wprowadzenie użytkownika w błąd po pełne sparowanie urządzeń i dostęp do wszystkich funkcji jakie udostępnia siec Bluetooth.

**Metody przeciwdziałania**

Zabezpieczenie się przed tego typu atakiem jest relatywnie proste dla użytkownika znającego zagadnienia bezpieczeństwa sieciowego i sprowadza się do obrony przed inteligentną manipulacją i wyrażaniem zgody na parowanie urządzeń tylko zaufanym użytkownikom. Oczywiście pozostawanie w trybie ukrytym utrudnia większość opisanych powyżej nadużyć.

Wiele nowszych urządzeń implementujących stos protokołów Bluetooth, np. telefony komórkowe firmy Nokia, są w obecnej chwili tak skonfigurowane, aby domyślnie, pomimo sparowania urządzeń, pytać użytkownika o pozwolenie na każdą pojedynczą operację, o którą prosi zewnętrzne urządzenie. Użytkownik ma więc możliwość pełnej kontroli tego co się dzieje w jego urządzeniu. Funkcjonalność ta jest jednak na tyle uciążliwa, że potrafi pytać użytkownika o zgodę na każdą zmianę katalogu przez zewnętrzne urządzenia przeglądające zdalnie strukturę plików. Okazuje się, że niektóre firmy potraktowały aspekty bezpieczeństwa tak poważnie, że do pewnego stopnia stają się one tak restrykcyjne, że uniemożliwiają normalne użytkowanie sprzętu. Potwierdzanie takie można oczywiście wyłączyć dla konkretnego urządzenia i wymaga to zmiany parametrów polityki bezpieczeństwa dla każdego sparowanego urządzenia z osobna. Z drugiej strony trzeba mieć świadomość, że wyłączenie tej funkcji daje zewnętrznemu sparowanemu użytkownikowi pełne prawa do robienia wszystkiego z danym urządzeniem.

## 2.3 BlueSnarf

### Historia

BlueSnarf jest pierwszą znalezioną i opisaną dużą luką bezpieczeństwa sieci Bluetooth. Została ona odkryta niezależnie przez Marcela Holtmanna w sierpniu 2003 roku i Adama Laurie w listopadzie 2003 roku. Pełna prezentacja sposobu przeprowadzenia tego ataku została zademonstrowana na konferencji 21st Chaos Communication Congress w Berlinie pod koniec 2004 roku przez odkrywców luki, członków grupy trifinite.org. Według nich okres trzynastu miesięcy jaki upłynął od znalezienia luki do opublikowania informacji o jej istnieniu powinien wystarczyć producentom sprzętu na jej załatanie. I zgodnie ze słowami hakerów śledzących stanowisko producentów sprzętu stwierdzili oni, że ich produkty są już odporne na opublikowany na konferencji sposób przeprowadzenia ataku. [12]

O pochodzeniu nazwy tego ataku można dowiedzieć się więcej ze strony [9].

### Opis ataku

BlueSnarf wykorzystuje błędy w implementacji profilu OPP (ang. OBEX Push Profile). Normalnie profil ten służy do tego, aby przesyłać między urządzeniami pojedyncze obiekty takie jak pliki, wizytówki, dane kalendarza, itp. Ponieważ przesyłane są tylko wybrane obiekty bezpośrednio wskazane przez użytkownika, więc z założenia profil ten nie wymaga uwierzytelnienia. Od strony implementacji można powiedzieć, że profil ten jest okrojoną wersją profilu OBEX FTP (ang. File Transfer Profile), ponieważ implementuje tylko funkcje PUSH brak w nim natomiast funkcji GET i możliwości przeglądania struktury katalogów.

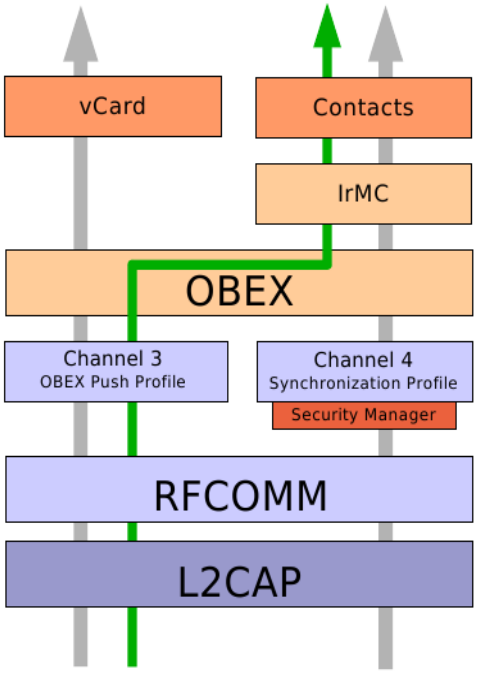
Okazuje się jednak, że pewne implementacje tego profilu dopuszczają funkcje GET i możliwość pobierania plików z danego urządzenia bez autoryzacji. Luka ta nie umożliwia przeglądania struktury katalogów i listy plików, a tylko pobieranie plików, których lokalizacja jest znana lub możliwa do odgadnięcia. Ponieważ w większości telefonów popularnych firm struktura katalogów jest znana, więc znajomość producenta urządzenia i ew. modelu umożliwia sprawny, nieautoryzowany dostęp do danych użytkownika.[34][17]

Luka ta jest o tyle ciekawa, że dostęp do struktury katalogów poprzez OBEX umożliwia, zgodnie z założeniami profilu IrMC Specification [29], również dostęp do danych użytkownika takich jak książka adresowa, kalendarz czy SMSy. Dane te, mimo że nie będące częścią struktury katalogów, są dostępne poprzez OBEX przez wywołanie funkcji GET z nazwą nie istniejącego, ale określonego pliku. Nazwami takich plików są np.

- 'telecom/pb.vcf' - książka adresowa
- 'telecom/cal.vcs' - kalendarza
- 'telecom/rtc.txt' - zegar
- 'telecom/devinfo.txt' – informacje o urządzeniu!

Atak ten umożliwia więc dostęp do plików, ale nie pozwala ich modyfikować i co za tym idzie usuwać. Koncepcje przeprowadzenia ataku przedstawia rysunek 9. Demonstruje on jak unikając parowania mieć dostęp do zabezpieczonego profilu synchronizacyjnego poprzez skorzystanie z profilu OBEX Push.

Warto również zauważyć, że określenia „bluesnarfing” używa się często, aby określić jakiekolwiek praktyki prowadzące do pozyskania danych z urządzenia bez konieczności parowania.




Rysunek 9: Sposób na uniknięcie parowania [0]

**Atak w praktyce**

Dnia 6 kwietnia 2006 Konstantina Sapronova z laboratorium ds. analizy wirusów Kaspersky Lab opublikował wyniki swojego testu przeprowadzonego w kilku centrach handlowych w Rosji polegającego na wyszukaniu urządzeń Bluetooth dostępnych w pobliżu i sprawdzeniu ich podatności na ataki. Testy wykazały, że z 194 wykrytych urządzeń Bluetooth aż 48% jest podatne na atak BlueSnarf i aż 25% użytkowników przyjęło wysłane im pliki! Badania pokazały również, że spośród wszystkich znalezionych urządzeń Bluetooth 33% stanowiły urządzenia firmy Nokia, a 32% firmy Sony Ericsson.[35]

Rok później 20 marca 2007 roku Sapronov opublikował wyniki kolejnego testu, który przeprowadził, aby przekonać się jak zmieniła się sytuacja. Przedstawił on następujące

wnioski:

„Jeśli porównamy tegoroczne dane z danymi z zeszłego roku, zauważymy dwie rzeczy: pierwsza to taka, że Nokia, która w zeszłym roku znalazła się na pierwszym miejscu, nadal pozostaje na prowadzeniu. Jednak, na drugim miejscu jest teraz Samsung, który zepchnął Sony Ericssona na trzecią pozycję. W tym roku wykryliśmy znaczną liczbę zestawów słuchawkowych w technologii Bluetooth, łącznie z zestawami dla samochodów. Zwiększyła się również liczba urządzeń PDA i laptopów.

Pod względem bezpieczeństwa istotne znaczenie ma liczba urządzeń podatnych na snarfing. Snarf jest prawdopodobnie najbardziej znanym rodzajem ataku Bluetooth. Skuteczny atak typu snarf pozwala zdalnemu szkodliwemu użytkownikowi uzyskać każdy plik z urządzenia ofiary, łącznie z książką adresową, notatnikiem, zdjęciami itd.

W 2007 roku liczba urządzeń podatnych na snarfing była znacznie mniejsza niż w 2006 roku i ograniczyła się do 25% wszystkich badanych urządzeń (poprzednio było to 48% urządzeń podatnych na atak).”[36]

**Podatność na atak**

Na opisany atak podatne są telefony prawie wszystkich producentów produkowane do 2005 roku. Odkrywcy luki podają jako przykład następujące modele [12]:

- Ericsson R520m, T39m, T68
- Sony Ericsson T68i, T610, Z1010
- Nokia 6310, 6310i, 8910, 8910i

Według producentów nowe urządzenia pojawiające się na rynku już od dawna nie są podatne na tę lukę. Jednak jak wspomniano powyżej praktyczne badania udowadniają, że w użyciu jest jeszcze wiele urządzeń podatnych na to nadużycie.

**Ocena zagrożeń płynących z ataku**

Udane przeprowadzenie opisywanego ataku daje hakerowi dostęp do plików i prywatnych danych użytkowników. Ponieważ jednak należy z góry znać dokładną lokalizację tych plików i można je tylko odczytać bez możliwości zmiany czy usuwania nie jest to więc zbyt niebezpieczny atak.

**Metody przeciwdziałania**

Zabezpieczenie się przed tego typu atakiem sprowadza się do aktualizacji oprogramowania sterującego telefonem komórkowym (jeśli aktualizacja została opublikowana) lub wymiany telefonu na nowszy model, odporny na atak. Rozwiązaniem połowicznym może być ustawienie telefonu w tryb niewidoczności lub włączanie Bluetooth tylko kiedy to jest absolutnie konieczne.

## 2.4 BlueBug

**Historia**

Odkrycie i upublicznienie przez Adama Laurie luki BlueSnarf wraz z informacją o możliwościach tego ataku bez wyjaśnień w ówczesnym czasie, jak dokładnie ten atak przeprowadzić skłoniło Martina Herfurta do własnych badań. Odkrył on lukę bezpieczeństwa, która umożliwiała podobne nadużycia. Kiedy obaj panowie spotkali się w 2004 roku okazało się, że znalezione przez nich luki, pomimo że udostępniają podobne możliwości, są całkowicie innymi sposobami ataku. Przez przypadek została więc odkryta kolejna bardzo poważna jak na owe czasy luka, która umożliwiła prawie całkowite przejęcie przez hakera kontroli nad atakowanym telefonem. Pierwsze opublikowanie informacji o BlueBug nastąpiło na targach CeBIT w Hanowerze w 2004 roku. Jako ciekawostkę można podać, że Martin Herfurt przebywając na tych targach przeprowadził eksperyment z którego wynikało, że na około 1300 unikalnych urządzeń Bluetooth około 50 telefonów komórkowych było podatne na ten atak. Eksperyment ten został przez niego opisany dokładnie w dokumencie [14]. Dokument ma w nazwie słowo BlueSnarf, ponieważ jak wspomniano wcześniej, w trakcie tworzenia raportu jego autor był przekonany, że odkryta przez niego luka to atak BlueSnarf.

Pełna prezentacja sposobu przeprowadzenia tego ataku, jak i innych odkrytych przez grupę trifinite.org ataków, została przeprowadzona na konferencji 21st Chaos Communication Congress w Berlinie pod koniec 2004 roku przez odkrywców luki, członków grupy trifinite.org. [15]

**Opis ataku**

BlueBug podobnie jak BlueSnarf bazuje na błędnej implementacji profilów aplikacji Bluetooth. W przeciwieństwie jednak do BlueSnarf nie bazuje na profilu OPP i protokole OBEX, ale na protokole RFCOMM i poleceniach sterujących modemem (ang. AT commands). A dokładniej na nie upublicznionych w rejestrze SDP serwerach bazujących na protokole RFCOMM. Do tych serwerów można się podłączyć emulatorem terminala i bez autoryzacji wydawać polecenia sterujące telefonem. Polecenia te, działające jako standard już w modemach podłączanych do komputerów, dzisiaj są standardem zaimplementowanym w każdym telefonie komórkowym i umożliwiają sterowanie niemal wszystkimi funkcjami telefonu. [12][34] Przykładowo są to:

- sterowanie połączeniami (tworzenie połączeń, zawieszanie połączeń itp.),
- wysyłanie, czytanie i usuwanie SMSów,
- czytanie i zapisywanie wpisów książki adresowej,
- ustawianie przekierowania rozmów,
- konfiguracja telefonu,
- dostęp do plików.

**Podatność na atak**

Na opisywany atak podatne są telefony szczególnie firm Nokia i Sony Ericsson produkowane do 2005 roku. Odkrywcy luki podają jako przykład następujące modele [12]:

- Nokia (6310, 6310i, 8910 8910i,...)
- Sony Ericsson T86i, T610, ...

Według producentów nowe urządzenia pojawiające się na rynku już od dawna nie są podatne na tę lukę. Jednak praktyczne badania udowadniają, że w użyciu jest jeszcze wiele urządzeń podatnych na to nadużycie.

**Ocena zagrożeń płynących z ataku**

Udane przeprowadzenie tego ataku daje niemal nieograniczoną kontrolę nad funkcjami telefonu i jest wyjątkowo niebezpieczne.

**Metody przeciwdziałania**

Zabezpieczenie się przed tego typu atakiem sprowadza się do aktualizacji oprogramowania sterującego telefonem komórkowym (jeśli aktualizacja została opublikowana) lub wymiany telefonu na nowszy model, odporny na atak. Rozwiązaniem połowicznym może być ustawienie telefonu w tryb niewidoczności lub włączanie Bluetooth tylko kiedy to jest absolutnie konieczne.

**Narzędzia służące do przeprowadzenia ataku**

- Bluebugger: [33]
- BloooverII: [13]
- Bluediving: [6]

## 2.5 Skanowanie w sieci Bluetooth

Obecność w urządzeniach Bluetooth ukrytych przez producentów implementacji pewnych usług, które mogły być podatne na ataki dała początek programom skanującym urządzenia w poszukiwaniu tego typu usług. Pracami nad programem zajął się Collin Mulliner, a ich owocem był pakiet skanerów „BT Audit".

Skanowanie jest metodą rozpoznania usług dostępnych w zdalnym urządzeniu. O ile w normalnej sieci TCP/IP sprowadza się ono do sprawdzenia otwartych portów protokołu TCP, o tyle w standardzie Bluetooth, ze względu na mnogość protokołów pośredniczących, sprawa jest o wiele bardziej skomplikowana.

**Rejestry SDP**

Najprostszą metodą rozpoznania usług w zdalnym urządzeniu jest po prostu zapytanie o nie tego urządzenia. W przeciwieństwie do sieci TCP/IP w sieciach Bluetooth istnieje rejestr usług udostępnianych przez urządzenie dostępny dla innych odpytujących członków sieci. Można by więc pomyśleć, że skanowanie w tej sieci mija się z celem. Okazuje się, że jednak nie.

Bluebug jako jedna z najbardziej znanych luk w systemie bezpieczeństwa telefonów wspierających Bluetooth opiera się na fakcie istnienia nieujętego w rejestrze SDP kanału RFCOMM. Okazuję się więc, że często warto przeskanować wszystkie protokoły na obecność otwartych portów nie ujętych w rejestrach.

Pozostając jeszcze przy zagadnieniu rejestru SDP okazuje się, że nawet on może pozostać ukryty lub całkowicie źle zaimplementowany. Doświadczenia autora z urządzeniem firmy HTC pracującym pod kontrolą systemu Windows Mobile 5 pokazują, że standardowe zapytanie o wpisy SDP w rejestrze urządzenia (polecenie w Linuksie: 'sdptool browse') nie zwraca żadnych danych! Użycie natomiast polecenia, które po kolei żąda podania szczegółów wpisu o określonym numerze w rejestrze (polecenie w Linuksie: 'sdptool records') skutkuje

dopiero możliwością obejrzenia całego rejestru. Odpytanie urządzenia o obecność konkretnej usługi również zwraca poprawne odpowiedzi.

### Protokół L2CAP

Podstawowym protokołem na którym opierają się wszystkie protokoły pośredniczące Bluetooth jest L2CAP. Jak wiadomo protokół ten, z racji istnienia wielu protokołów pośredniczących wyższych warstw, posiada specjalne pole PSM (ang. Protocol and Service Multiplexer), umożliwiające ich rozróżnianie, jak i istnienie wielu różnych ich instancji. Zostało to opisane w podpunkcie 1.7 „Warstwa L2CAP”:

„Podobnie jak w protokole TCP zakres numerów jakie może przyjmować pole PSM podzielone jest na dwa przedziały adresów:

- wartości od 1 do 1000 - blok zarezerwowanych adresów opisujący dobrze znane protokoły,
- wartości powyżej 1000 do 65535 - blok swobodnego wykorzystywania – do implementacji jeszcze nie przypisanych nowych protokołów i wielokrotnych implementacji danych protokołów warstw wyższych.

Należy jeszcze zauważyć, że pole PSM może przyjmować tylko wartości nieparzyste. Przykładowe wartości PSM dla znanych protokołów to: 0x001 – SDP czy 0x003 - RFCOMM ”

Jeśli producent urządzenia chciałby ukryć pewną funkcjonalność przed wścibskimi oczami użytkowników, to mógłby wykorzystać wartość pola PSM powyżej 1000. Warto więc badając nowe urządzenie zeskanować wszystkie wartości PSM pod kontem nie standardowych usług.

### Protokół RFCOMM

Kolejnym sposobem dającym idealne możliwości ukrycia pewnych usług jest protokół RFCOMM. Jest on wykorzystywany przez wiele profili. Pojawia się więc potrzeba, podobnie jak w warstwie L2CAP, zwielokrotnienia łączy tak, aby jedno połączenie RFCOMM mogło być współdzielone przez wiele aplikacji. W tej warstwie stosowany jest mechanizm tzw. kanałów. Polega on na nadawaniu kolejnym profilom korzystającym z protokołu numerów kanału, na który ich aplikacje mogą się łączyć. Kanały mogą przyjmować wartości od 1 do 30.

Atak Bluebug został wykryty w telefonach firmy Nokia na ukrytym kanale 17.

Inne protokoły ze względu na swoje bardziej specyficzne przeznaczenie nie są przydatne do skanowania.

Więcej informacji na ten temat można znaleźć pod adresami: [18][20]

### Podatność na atak

Skanowanie nie jest luką zabezpieczeń, ale zwykłym sposobem analizy udostępnianych przez urządzenia informacji koniecznych do działania sieci. Na skanowanie podatne są więc wszystkie urządzenia pracujące w sieci Bluetooth nie zależnie od producenta czy wersji standardu.

### Ocena zagrożeń płynących z ataku

Jak wspomniano opisana metoda nie jest atakiem, więc nie niesie ze sobą

bezpośrednich zagrożeń bezpieczeństwa. Jest to jednak świetny sposób na analizę podatności urządzenia na inne ataki i luki bezpieczeństwa.

**Metody przeciwdziałania**

Standard Bluetooth nie przewiduje możliwości ukrywania usług udostępnianych przez urządzenie. Coraz częściej pojawiają się jednak urządzenia umożliwiające wybranie użytkownikowi, które profile mają być włączone i udostępniane przez urządzenie, a które mają pozostać wyłączone. Przykładem są telefony z MS Windows Mobile 2005, które umożliwiają wyłączenia dla przykłady profilu OBEX FTP co skutecznie zapobiega możliwości zdalnego przeglądania plików w urządzeniu.

**Narzędzia służące do przeprowadzenia ataku:**

Narzędziem umożliwiającym skanowanie urządzeń pod kątem wykorzystywanych przez nie numerów PSM i kanałów RFCOMM jest pakiet „BT Audit" dostępny pod adresem: [18]

## 2.6 BlueSmack

**Opis ataku**

BlueSmack jest najprostszą wersją ataku DoS dla sieci Bluetooth. Pierwowzorem tego ataku jest doskonale znany z systemu Windows „Ping of Death". Atak ten powodował zawieszenie się systemu Windows 95 poprzez wysłanie zbyt dużego pakietu ping. Pakiet taki przepełniał bufor systemu i w konsekwencji zawieszał cały system. BlueSmack jest dokładnie takim samym atakiem skierowanym w warstwę L2CAP standardu Bluetooth.

Atak ten jest o tyle ciekawy, że dotyczy on implementacji niskiej warstwy stosu protokołów Bluetooth. Powoduje to, że na lukę mogą być podatne właściwie wszystkie urządzenia Bluetooth, ponieważ, o ile różne urządzenia implementują określone profile aplikacji, o tyle każde z nich bez wyjątku musi mieć zaimplementowaną warstwę L2CAP.

**Sposób przeprowadzenia ataku:**

Do przeprowadzenie opisywanego ataku wystarczy standardowe narzędzie, będące częścią pakietu BlueZ w systemie Linux. Narzędziem tym jest polecenie „l2ping", które jak łatwo się domyślić służy do sprawdzania obecności zdalnego urządzenia w sieci. Jedną z opcji tego narzędzia jest możliwość wyspecyfikowania wielkości przesyłanego do urządzenia pakietu echo request. Tą wielkości podaje się z parametrem '-s'. Większość źródeł podaje wartość 600 lub większą jako odpowiednią do przepełnienia bufora i zawieszenia zdalnego urządzenia podatnego na tego typu atak.

Polecenie wygląda więc następująco: 'l2ping -s 600 bdaddr ' gdzie bdaddres jest adresem atakowanego urządzenia. Sukces ataku można poznać po braku odpowiedzi echo response od atakowanego urządzenia. [16]

**Podatność na atak**

Opisany atak ma znaczenie historyczne. Według odkrywców luki szczególnie podatnymi urządzeniami na ten atak były palmtopy Compaq iPaq.

### Ocena zagrożeń płynących z ataku

Udane przeprowadzenie ataku BlueSmack powoduje zawieszenie atakowanego urządzenia i umożliwia dzięki temu skuteczne uniemożliwienie korzystania z urządzenia.

### Metody zabezpieczeń

Zabezpieczenie się przed tego typu atakiem sprowadza się do aktualizacji oprogramowania sterującego urządzeniem (jeśli aktualizacja została opublikowana) lub wymiany urządzenia na nowszy model, odporny na atak. Rozwiązaniem połowicznym może być przejście w tryb ukryty lub włączanie Bluetooth tylko kiedy to jest absolutnie konieczne.

## 2.7 Nadużycie poprzez wymuszenie uwierzytelnienia (ang. Mode3 Abuse)

### Historia

Opisane nadużycie (bo trudno je nazwać atakiem) zostało po raz pierwszy opisane przez grupę trifinity.org na konferencji LayerOne odbywającej się od 24 do 25 kwietnia 2005 roku w Pasadena, w USA. [10]

### Opis ataku

Mode3 Abuse jest to zwykłe nadużycie zaufania ofiary, polegające na wymuszeniu sparowania urządzeń pod błahym pretekstem, poprzez ustawienie na urządzeniu atakującego wymuszania uwierzytelnienia (tryb zabezpieczeń 3). Jeśli ofiara zgodzi się sparować urządzenia, napastnik wykorzystuje okazane mu zaufanie i łączy się z innymi interesującymi usługami już bez konieczności uwierzytelnienia.

Jak widać początkowe etapy tego nadużycia są identyczne jak dla opisanego powyżej ataku BlueBump. Różni się on tylko tym, że ofiara nie jest proszona o usunięcie klucza, którego pozostawienie jest podstawą nadużycia. [10]

### Podatność na atak

Na opisany atak podatni są ludzie, nie urządzenia. Jest więc on ciągle aktualny i stanowi prawdopodobnie najprostszy i najmniej wyrafinowany sposób na obejście zabezpieczeń. Jeśli pretekst którego użyje atakujący jest wystarczająco przekonujący, ofiara po prostu pozwoli sparować urządzenie i bardzo często zdarza się, że takie sparowanie nigdy nie jest cofane.

### Ocena zagrożeń płynących z ataków

Udane przeprowadzenie tego ataku skutkujące nieautoryzowanym sparowaniem urządzeń umożliwia dostęp do wszystkich usług atakowanego urządzenia, daje więc nieograniczony dostęp atakującemu. Jest to więc wyjątkowo niebezpieczny atak.

### Metody zabezpieczeń

Metoda zabezpieczenia się przed tego typu atakiem jest bardzo prosta i polega na autoryzowaniu użytkowników, którym bezwzględnie ufamy lub cofnięciu autoryzacji tuż po wykonaniu postulowanej przez drugą osobę operacji.

## 2.8 Wirusy w sieci Bluetooth

Bluetooth jako samo konfigurująca się sieć ad hoc jest idealnym sposobem rozprzestrzeniania się wszelkiego typu złośliwego oprogramowania, szczególnie wirusów. Pojawienie się wirusów przenoszących się z pomocą tej sieci było tylko kwestią czasu.

### Caribe

Pierwszym wirusem, zaprojektowanym dla smartfonów i korzystającym z sieci Bluetooth w celu przenoszenia się, był wirus nazwany przez jego autorów Caribe znany szerzej jako Cabir wykryty w połowie 2004 roku. Wirus ten został napisany w charakterze proof-of-concept, aby udowodnić, że sieć Bluetooth jest podatna na tego typu ataki. Autorami jest grupa hakerów z Czech i Słowacji zwana 29a, która tuż po jego stworzeniu wysłała wirusa do wielu firm, zajmujących się bezpieczeństwem komputerowym, aby uczulić je na tego typu zagrożenie. Wirus działa na platformie Symbian Series 60 i rozsyła się jako plik z rozszerzeniem „.sis". Aby infekcja wirusa zakończyła się sukcesem konieczne jest podwójne wyrażenie przez użytkownika zgody. Najpierw zgody na kontynuowanie instalacji oprogramowania nie podpisanego cyfrowo (niektóre telefony), a potem na właściwą instalację. Tak więc dla użytkownika, który w minimalnym stopniu jest świadomy zasad bezpieczeństwa komputerowego wirus nie stanowi dużego zagrożenia. Wystarczy odpowiedzieć negatywnie, kiedy w telefonie pojawi się pytanie czy przyjąć nowy plik. Zarażony telefon komórkowy będzie jednak próbował wysłać wirusa tak długo, aż użytkownik wyrazi zgodę. Wirus nie jest niebezpieczny, ponieważ nie ma żadnych przewidzianych przez autorów funkcji destrukcyjnych. Zasada jego działania, polegająca na częstym, okresowym wyszukiwaniu urządzeń Bluetooth będących w pobliżu i wysyłaniu własnego kodu do tych urządzeń powoduje o wiele szybsze rozładowywanie się baterii i wolniejsze działanie telefonu i często jego mniej stabilną pracę.[19][11]

Niedługo po pojawieniu się tego pierwszego wirusa okazało się, że jego kod źródłowy został przez twórców opublikowany pod koniec 2004 roku i do dzisiaj jest dostępny wraz z ósmym wydaniem magazynu grupy 29a. [25] Prawdopodobnie wirus wyciekł jeszcze zanim został oficjalnie udostępniony, ponieważ jego zmodyfikowane wersje zaczęły się pojawiać w wielu krajach jeszcze przed oficjalną publikacją. [32] Z pewnością jednak znacząca większość dostępnych dzisiaj wirusów i prawie wszystkie dostępne na platformę Symbian są modyfikacją tej pierwszej wersji wirusa Caribe.

### Kolejne wirusy

Najciekawsze modyfikacje tego wirusa to:

a) 11 stycznia 2005 Worm.SymbOS.Lasco.a\Worm/SymbOS.Cabir.f

„Jest to robak infekujący komputery przenośne PocketPC oraz telefony komórkowe działające pod kontrolą systemu operacyjnego Symbian. Szkodnik jako pierwszy w historii infekuje pliki wykonywalne (archiwa SIS). Szkodnik został napisany przez twórcę robaka Worm.SymbOS.Cabir.a i bazuje na jego kodzie. Lasco.a rozprzestrzenia się poprzez port BlueTooth.

Poza wykorzystywaniem sieci BlueTooth Lasco.a infekuje także pliki. Podczas uruchamiania robak szuka pliku SIS i umieszcza w nich własny kod.

Lasco.a jest zarówno aplikacją dla platformy Win32, która infekuje pliki SIS, jak również aplikacją dla platformy Symbian.” [37]

b) 10 marca 2005 Worm.SymbOS.Comwar.a

„Robak rozprzestrzenia się poprzez sieć Bluetooth oraz wiadomości MMS. (...) Jest to pierwszy wirus dla telefonów komórkowych, który może rozprzestrzeniać się za pośrednictwem wiadomości MMS. Infekuje telefony działające pod kontrolą systemu operacyjnego Symbian Series 60. Szkodnik ma postać archiwum systemu Symbian (SIS). Jego rozmiar to około 27 - 30 KB. Nazwa zainfekowanego pliku może być różna - jeżeli robak rozprzestrzenia się poprzez Bluetooth nazwa jest losowa i składa się z ośmiu znaków, przykładowo bg82o_s1.sis.” [38]

c) 11 kwietnia 2005 Worm.SymbOS.Cabir.k\SymbOS.Mabir.A

„... Dodatkowo, w przeciwieństwie do poprzednich wersji, Cabir.k może rozprzestrzeniać się za pośrednictwem wiadomości MMS. Robak automatycznie odpowiada na każdą wiadomość MMS wysyłając do nadawcy własną kopię.” [39]

**Wirusy infekujące system OSX firmy Apple**

W lutym 2006 pojawił się pierwszy wirus wykorzystujący sieci Bluetooth w celu atakowania systemów OSX firmy Apple nazwany Inqtana (Worm.OSX.Inqtana.a).

„Jest to robak działający w systemie operacyjnym Mac OSX. Szkodnik rozprzestrzenia się poprzez technologię Bluetooth - wysyła do potencjalnej ofiary żądanie transmisji danych Object Exchange (OBEX) Push. Jeżeli użytkownik zaakceptuje żądanie robak wykorzystuje lukę Bluetooth File and Object Exchange Directory Traversal w celu uzyskania dostępu do zasobów znajdujących się poza ścieżką systemową Bluetooth File and Object Exchange.

Robak umieszcza w folderze LaunchAgents dwa pliki: com.openbundle.plist oraz com.pwned.plist, co zapewnia mu uruchamianie wraz z każdym startem komputera. W folderze /Users/ umieszczany jest plik w0rm-support.tgz zawierający główny kod robaka.

Po ponownym uruchomieniu systemu moduł com.pwned.plist wypakowuje składniki robaka, natomiast com.pwned.plist uruchamia jego główną bibliotekę. Następnie robak podejmuje próbę powielania się poprzez skanowanie w poszukiwaniu urządzeń z włączoną usługą Bluetooth. Po znalezieniu takich urządzeń szkodnik wysyła własne kopie poprzez żądania Object Exchange (OBEX) Push.” [40]

Opisane wirusy to tylko najpopularniejsze wersje wszystkich wirusów korzystających ze standardu Bluetooth, jednak jak się okazuje sieci Bluetooth raczej nie zostaną zalane kolejnymi falami nowych wirusów. Według informacji opublikowanych przez firmę G DATA Software powstaje coraz mniej wirusów na telefony komórkowe, ponieważ hakerzy już udowodnili, że jest to możliwe, a przestępcy mają bardzo ograniczone możliwości zarobku na tego typu procederze. [26]

Również policja nie próżnuje i osoby rozpowszechniające złośliwe oprogramowanie na telefony komórkowe nie są jak się okazuje bezkarne. Zdarzają się przypadki aresztowania twórców tego typu oprogramowania. Jeden z takich faktów opisał serwis Heise Security dokumentując ujęcie przez policję w Hiszpanii osoby podejrzanej o rozpowszechnianie ponad dwudziestu różnych wersji wirusów Carib i Commwarrior. [28]

**Podatność na atak:**

Złośliwe oprogramowanie jest cały czas zmorą użytkowników komputerów i za sprawą sieci Bluetooth zaczyna być również uciążliwe dla posiadaczy telefonów komórkowych. Jednak podatne są na nie zazwyczaj urządzenia typu smartfone, które mają bardziej zaawansowane systemy operacyjne taki jak Windows Mobile czy Symbian, umożliwiający instalacje dodatkowego oprogramowania. Użytkownicy prostszych telefonów są mniej narażeni na atak.

**Ocena zagrożeń płynących z ataków**

Zainfekowane przez wirusy urządzenie jest właściwie skazane na pomysłowość autorów wirusów, co oznacza że zagrożenie można właściwie szacować dla danego typu wirusa. Infekcja może zakończyć się tylko zmniejszoną żywotnością baterii i mniejszą stabilnością urządzenia dla wirusów, których zadaniem jest tylko przenoszenia się dalej. Natomiast jeśli autor wirusa zaimplementował w nim funkcję destrukcyjne takie jak np. usuwanie plików i prywatnych danych użytkowników czy przesyłanie się wirusa w postaci wiadomości SMS lub MMS to konsekwencją może być nie tylko utrata ważnych danych ale również duże koszty finansowe.

**Metody przeciwdziałania**

Dla świadomego zagrożenia użytkownika obrona przed złośliwym oprogramowaniem jest bardzo prosta i sprowadza się do wyłączenie sieci Bluetooth zawsze kiedy to możliwe, pracy urządzeń w trybie ukrytym i odrzucanie wszelkiego typu oprogramowania, które zostało nam przysłane z nie znanych źródeł.

Producenci oprogramowania antywirusowego dostarczają już programy przeznaczone dla smartfonów, które zabezpieczają użytkowników przed wszelkiego typu złośliwym oprogramowaniem i często również przed atakami z zewnątrz. Przykładem takiego oprogramowania jest Kaspersky Anti-Virus Mobile dostępny dla platform Symbian i Windows Mobile.

# BIBLIOGRAFIA


[1]: Bluetooth SIG, "Bluetooth Core Specification v2.1 + EDR", , 2007

[2]: Bray Jennifer, Struman F Charles, "Bluetooth: connect without cables", Prentice Hall PTR, 2001

[3]: Lopez Ugo, "Hakowanie Bluetooth", Software, 2007

[4]: Miller A. Brent, Bisdikian Chatschik, "Bluetooth", Helion, 2003

[5]: http://advice.cio.com/avishai_wool/new_bluetooth_2_1_spec_to_fix_security_flaws

[6]: http://bluediving.sourceforge.net/

[7]: http://bluetooth.com/NR/rdonlyres/0A0B3F36-D15F-4470-85A6-F2CCFA26F70F/0/SimplePairing_WP_V10r00.pdf

[8]: http://en.wikipedia.org/wiki/OBEX

[9]: http://en.wikipedia.org/wiki/Snarfing

[10]: http://layerone.info/archives/2005/trifinite_bluetooth_presentation_layerone.pdf

[11]: http://searchmobilecomputing.techtarget.com/sDefinition/0,,sid40_gci970552,00.html

[12]: http://trifinite.org/Downloads/21c3_Bluetooth_Hacking.pdf

[13]: http://trifinite.org/Downloads/Blooover.jar

[14]: http://trifinite.org/Downloads/BlueSnarf_CeBIT2004.pdf

[0]: http://trifinite.org/Downloads/trifinite.presentation_blackhat.pdf

[15]: http://trifinite.org/trifinite_stuff_bluebug.html

[16]: http://trifinite.org/trifinite_stuff_bluesmack.html

[17]: http://trifinite.org/trifinite_stuff_bluesnarf.html

[18]: http://trifinite.org/trifinite_stuff_btaudit.html

[19]: http://vil.nai.com/vil/content/v_126245.htm

[20]: http://www.betaversion.net/btdsd/

[21]: http://www.blackhat.com/presentations/bh-usa-06/bh-us-06-Merloni-v1.2.pdf

[22]: http://www.bluetomorrow.com/content/section/96/154/

[23]: http://www.bluetooth.com/

[24]: http://www.bluez.org/bluetooth-21-devices/

[25]: http://www.flavioweb.it/ezines/29a-8.zip

[26]: http://www.gdata.pl/portal/PL/content/view/244/9/

[27]: http://www.gearlive.com/news/article/161-bluetooth-21-edr/

[28]: http://www.heise-online.co.uk/security/Cell-phone-virus-suspect-arrested-in-Spain--/news/91674

[29]: http://www.irda.org

[30]: http://www.lockergnome.com/net/2005/07/20/bluetooth-spam-assaults-moviegoers-say-hello-to-bluetooth-promotional-kiosks/

[31]: http://www.lockergnome.com/net/2005/08/05/a-rose-by-any-other-name-bluespamming-cast-as-bluecasting/

[32]: http://www.mobiletracker.net/archives/2004/12/28/cabir_source_co.php

[33]: http://www.nruns.com/_downloads/23C3-Berlin-Bluetooth-Hacking-Revisited-Thierry-Zoller.pdf

[34]: http://www.remote-exploit.org/codes_bluebugger.html

[35]: http://www.thebunker.net/security/bluetooth.htm

[36]: http://www.viruslist.pl/analysis.html?newsid=185

[37]: http://www.viruslist.pl/analysis.html?newsid=417

[38]: http://www.viruslist.pl/encyclopedia.html?cat=13&uid=4274

[39]: http://www.viruslist.pl/encyclopedia.html?cat=13&uid=4307

[40]: http://www.viruslist.pl/encyclopedia.html?cat=13&uid=4313

[41]: http://www.viruslist.pl/encyclopedia.html?cat=13&uid=4513