Bluetooth - The architecture and protocol stack

Tim Sauvageot

Universität Potsdam, Potsdam 14469, Deutschland
sauvageot@uni-potsdam.de

Zusammenfassung. Bluetooth ermöglicht nicht nur Musik hören ohne Kabelsalat, sondern unterstützt auch TCP/IP, Gruppenkommunikation und viele weitere Funktionen. Im Folgenden wird zuerst eine kurze Erläuterung zu der Entstehung von Bluetooth gegeben. Anschließend wird auf Bluetooth Protokolle und die Architektur eingegangen und gezeigt, weshalb Bluetooth eine Vielzahl bekannter Protokolle unterstützen kann. Besonderer Fokus wird dabei auf die Sicherheitsarchitektur gelegt. Abschließend werden neue Funktionen aus den Versionen 5.1 und 5.2 vorgestellt.

Schlüsselwörter: Bluetooth Architektur · Bluetooth Protokolle · IEEE 802.15 · LE · Bluetooth Sicherheit

1 Motivation

Bluetooth entstand aus einer Studie, die 1994 von Ericsson durchgeführt wurde. Ziel der Studie war es, eine günstige und energiesparsame Lösung zu finden, die drahtlose Kommunikation für kleine Entfernungen ermöglicht. Aus dieser Studie entstand 1998 die Bluetooth Special Interest Group (SIG). Diese ist für die Weiterentwicklung der Bluetooth Spezifikation verantwortlich. Prominente Mitglieder dieser Gruppe sind beispielsweise Nokia, Intel und Microsoft.

Bei der Namensgebung war der dänische König Haral Blauzahn (Bluetooth) das Vorbild. Dieser vereinigte zu seiner Zeit Dänemark, indem der Konflikte des Landes löste. Die Bluetooth SIG hatte ähnliche Ziele, denn sie wollten die drahtlose Kommunikation von Geräten vereinheitlichen. Sowohl die Kommunikation, als auch das Zusammenspiel von Anwendungen, konnte harmonisert und im Standard IEEE 802.15 definiert werden [2].

2 Architektur & Protokolle

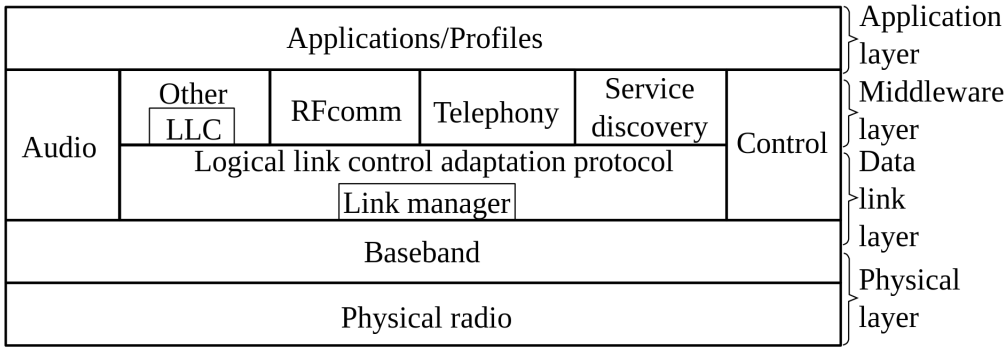

Abbildung 1. Bluetooth Architektur [1]

Bluetooth ist als Schichtenarchitektur entwickelt worden. An Hand von Abbildungen 1 und 2 erkennt man, dass es zu den Architekturkomponenten auch teilweise ein zugehöriges Protokoll gibt. Dies ist beispielsweise bei der Service Discovery und dem Service Discovery Protocol (SDP) als auch dem Link Manager und Link Manager Protocol (LMP) der Fall.

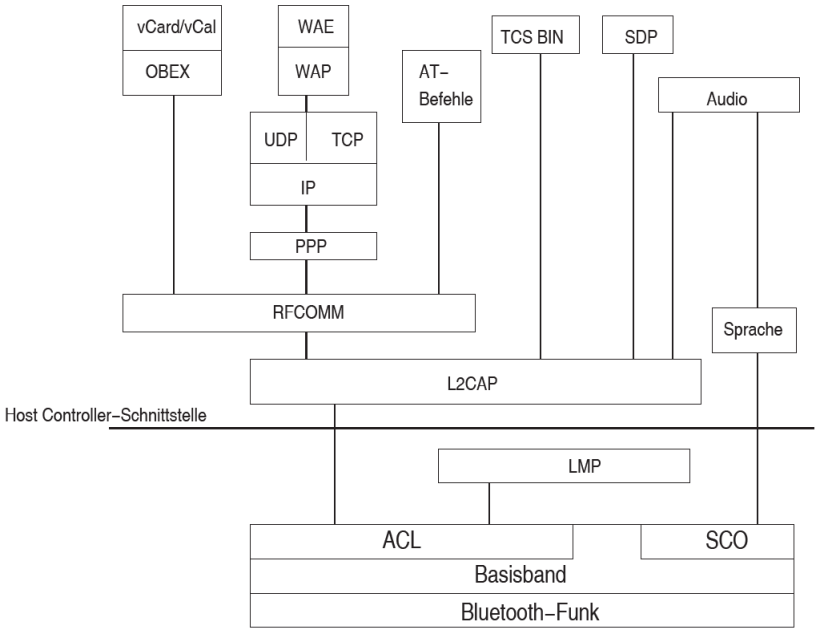

Abbildung 2. Bluetooth Protokollstack[2, p. 941]

2.1 MAC Control

Bluetooth war ursprünglich für Distanzen von maximal 10 Metern ausgelegt. Innerhalb dieser Distanz können sich bereits mehrere Bluetoothfähige Geräte wie Kopfhörer, Tastatur oder Maus befinden. Damit diese Geräte sich nicht gegenseitig stören, war es erforderlich, eine Lösung zu entwickeln, welche die Interferenz auf kleinen Räumen verhindert bzw. abschwächt. MAC Control bei Bluetooth wird mit Hilfe des Spread spektrums umgesetzt. Dabei wird das zu sendende Signal über den gesamten Frequenzbereich (2,401 - 2,479 Ghz) gespreizt. Der Frequenzbereich wird dafür in 79 Kanäle mit jeweils 1 MHz Bandbreite aufgeteilt. Der Master im jeweiligen Netzwerk bestimmt dann die Frequenz, auf der innerhalb des Netzwerks gesendet wird. Diese Frequenz wird dabei 1600 mal pro Sekunde gewechselt. Die Anzahl der Frequenzwechsel wird als Hop Rate bezeichnet. Die Einteilung der Bandbreite und das Wechseln der Kanäle wird als Frequence Hopping bezeichnet [2].

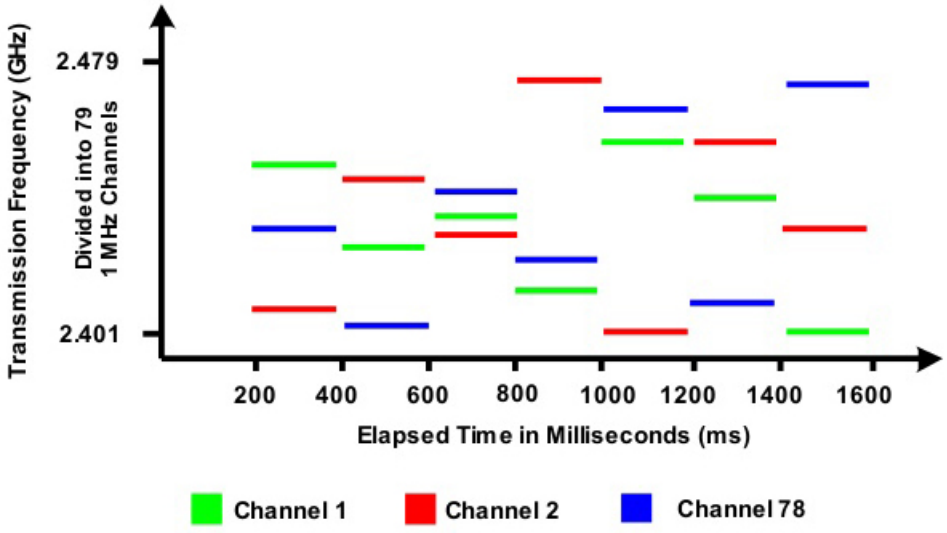

Abbildung 3. Spread spektrum[7]

Durch das schnelle Wechseln der Sendefrequenz wird nicht nur die Interferenz der Signale abgeschwächt, sondern auch eine grundlegende Abhörsicherheit gewährleistet. Angreifer, die nicht Teil des abzuhörenden Netzwerks sind, müssen die jeweils nächste Frequenz erraten, um dauerhaft die Kommunikation mitverfolgen zu können.

Aktuelle Bluetooth Geräte sind auch in der Lage über Distanzen von bis zu 100 Metern zu kommunizieren. Auf Grund der dafür benötigten Signalstärke verbraucht dies jedoch auch mehr Energie. Es ist daher wünschenswert, dass sich Kommunikationspartner in der Nähe befinden, um den Energieverbrauch zu minimieren. Mit Bluetooth 5.2 wurde eine Lösung gefunden, um den Energieverbrauch aktiv zu optimieren.

2.2 Pakete

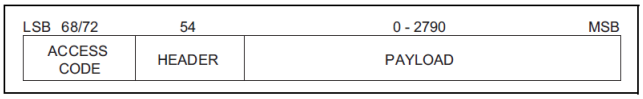

Abbildung 4. Paketformat[3, p. 461]

Der Datenaustausch bei Bluetooth findet über Pakete statt. Diese werden durch das Physical radio über die Luftschnittstelle übertragen. Der grundlegende Auf-

bau eines Pakets ist dabei in Abbildung 4 sichtbar. Der Zugangscode (access code) dient zur Synchronisierung von Kommunikationspartnern innerhalb eines Netzwerks. Die Größe des Zugangscodes beträgt, wenn kein anderer Inhalt des Pakets gesetzt ist, 68 Bit. Andernfalls ist der Zugangscode 72 Bit groß.
Der Header enthält Informationen zur Leitungssteuerung. Um Geräte innerhalb eines Netzwerks zu identifizieren, ist außerdem eine temporäre 3 Bit Addresse im Header gesetzt.
Der Payload dient zur Übertragung der eigentlichen Inhalte. Bei asynchroner Übertragung von Daten (ACL) ist außerdem eine 16 Bit Cyclic Redundancy Check Prüfsumme enthalten.
Die maximale Größe eines Pakets entspricht also 2912 Bit (364 Byte).

Paketvermittlung Bluetooth unterscheidet grundlegend zwischen zwei Paketmodi: Synchronous Connection-Oriented (SCO) und Asynchronous Connection-Less (ACL). Wie der Name schon erkennen lässt, ist SCO für synchrone Anwendungen mit geringer Latenz gedacht. Eine dieser Anwendungen ist beispielsweise die Sprachübertragung bei Kopfhörern. Jeder SCO Kanal hat eine Bandbreite von 64kBit/s.
Im Gegensatz dazu ist ACL dafür ausgelegt, "best effort"Verbindungen zu ermöglichen.

2.3 Link Manager

Der Link Manager setzt auf dem Baseband an und ist dem Data link layer zuzuordnen. Die Funktionen des Link Manages werden durch das Link Manager Protocol umgesetzt,

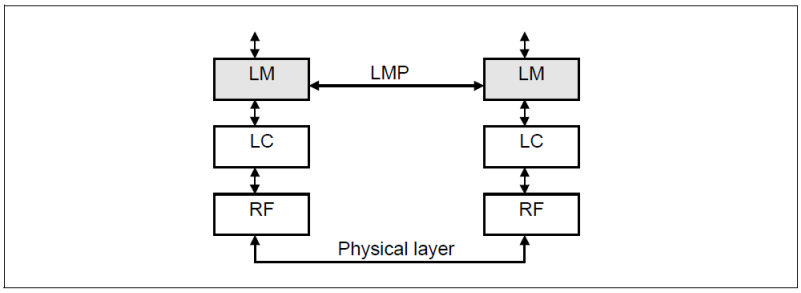

Abbildung 5. LMP Kommunikation[3, p. 572]

Link Manager Protocol Ein integraler Teil des Link Managers ist das Link Manager Protocol. Dieses ist für die Verbindungsverwaltung des Geräts verantwortlich. Dazu gehört beispielsweise die Authentifizierung von anderen Geräten oder die Verschlüsselung von Nachrichten.

Zwei Link Manager kommunizieren mittels Protocol Data Units (PDU). PDUs sind Nachrichtenformate, die standardmäßig durch Bluetooth spezifiziert werden. Allgemein werden die Nachrichten, die Link Manager austauschen, als Link Manager Protocol Nachricht bezeichnet. Diese Nachrichten werden im Payload der Pakete übermittelt. Weiterhin wird das L_CH Feld im Paketheader gesetzt. Über dieses können Link Manager die Nachrichten erkennen und abfangen. LMP Nachrichten werden anschließend nicht auf höhrere Schichten propagiert. Eine weitere Besonderheit vom LMP Nachrichten ist, dass diesen eine höhere Priorität als Benutzerdaten gegeben wird. Konkurrieren also ein Link Manager und eine Anwendung um Zugriff auf einen Kommunikationskanal, wird zuerst die LMP Nachricht gesendet.

Sicherheitsmanager Die zentrale Komponente der Bluetooth Sicherheitsarchitektur ist der Sicherheitsmanager. Dieser verwaltet Sicherheitsanforderungen von Diensten und Authentifizierungsinformationen von Geräten. Über die External Security Control Entity (ESCE) können externe Benutzereingaben, die beim Pairing erforderlich sein können, entgegengenommen werden. Weitere Funktionen sind die Durchführung der Authentifikation, Ver- und Entschlüsselung von Daten und die Initialisierung des Pairings [2].

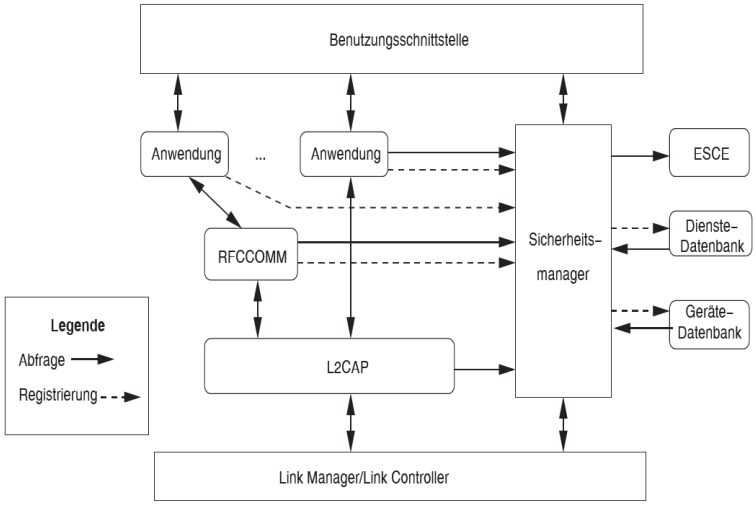

Abbildung 6. Sicherheitsmanager[2, p. 945]

Authentifizierung Um Gerätezugriff zu kontrollieren, ist es erforderlich alle Anfragen zu authentifizieren und authorisieren. Alle Anfrager werden dabei über eine 48 Bit Addresse (BD_ADDR) identifiziert. Dabei ist anzumerken, dass lediglich Geräte und keine einzelnen Benutzer authentifiziert werden. Eine An-

wendung mit benutzerbasierter Authentifizierung ist beispielsweise E-Mail.
Geräte in der Umbegung werden durch periodische inquiry Nachrichten gefunden. Empfängt ein Gerät eine solche Nachricht und hat den Discoverable mode aktiviert, so sendet es eine Nachricht mit dessen eigener BD_ADDR zurück. Durch das anschließende senden einer Page Nachricht wird die Kommunikation gestartet. Die Authentifizierung beider Parteien findet nun statt.
Bluetooth unterscheidet grundlegend zwischen vier unterschiedlichen Sicherheitmodi. Ein Gerät kann sich zu jeder Zeit in nur einem dieser Zustände befinden. Es ist wichtig zu beachten, dass die Namen der einzelnen Modi kein Indiz für die von ihnen bereitgestellte Sicherheit ist. In Abbildung 7 sind die einzelnen Pfade, die bei Authentifizierung und Authorisierung durchlaufen werden, gekennzeichnet.

Security Modus - 1 Im ersten Sicherheitsmodus werden keine Sicherheitsfunktionen bereitgestellt. Jede Anfrage wird sowohl authentifiziert als auch authorisiert. Anfragende Geräte haben also Zugriff auf alle Funktionen des Kommunikationspartners. Der Modus sollte deshalb nur für Geräte verwendet werden, die keine sicherheitsrelevanten Funktionen bereitstellen, da auch der Zugriff auf diese uneingeschränkt ist. Ein Anwendungsfall ist der Visitenkartenaustausch.

Security Modus - 2 und 4 Der zweite Sicherheitsmodus ermöglicht es anwendungsspezifische Sicherheitsmaßnahmen zu definieren. Diese werden im Service Layer umgesetzt. Wie in Abbildung 7 erkennbar ist, werden Anfragen in diesem Modus nicht authentifiziert sondern nur authorisiert. Um in diesem Modus zu kommunizieren muss außerdem eine L2CAP Verbindung aufgebaut werden.
Der Modus 4 unterscheidet sich von Modus 2 nur in einem Punkt. Im vierten Modus kann die Variante des Secure Simple Pairings definiert werden. Das Pairing spielt eine wichtige Rolle beim Generieren von Schlüsseln.
Eine weitere Funktion dieser Modi ist die Kategorisierung von Geräten an Hand von trust leveln. Die drei Kategorien sind: trustworty, not trustworthy und unknown. Wird ein Gerät als trustworthy eingestuft, so gilt es als authentifiziert und hat uneingeschränkten Zugriff auf alle Funktion des angefragten Geräts. Not trustworthy und unknown werden die gleichen Zugriffsrechte gewährt. Bei beiden wird der Zugriff eingeschränkt. Anwendungen und Dienste können definieren, mit welchem trust level auf sie zugegriffen werden darf. Diese Definitionen werden als Einträge in der Dienste-Datenbank des Sicherheitsmanagers gespeichert. Trust level, BD_ADDR, Gerätename und Link Key werden als Einträge in der Geräten-Datenbank des Sicherheitsmanagers gespeichert. Der Gerätename ist dabei ein friendly name wie beispielsweise Max Mustermanns Kopfhörer".

Modus 3 Im dritten Modus, welcher im Link Layers angesiedelt ist, wird ein grundlegender Schutz für alle Anwendungen gewährleistet. Alle Anfragen werden authentifiziert, jedoch nicht authorisiert. Um auf einen Dienst zugreifen zu können, muss vorher eine Link Manager Protocol Verbindung aufgebaut werden.

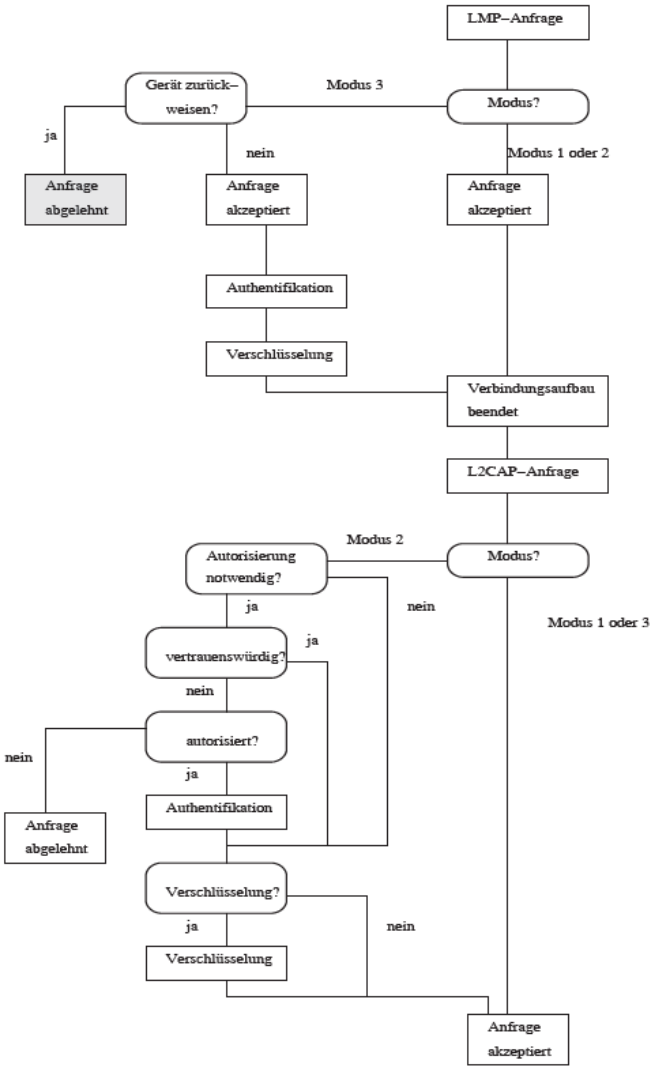

Abbildung 7. Sicherheitsmodi[2, p. 946]

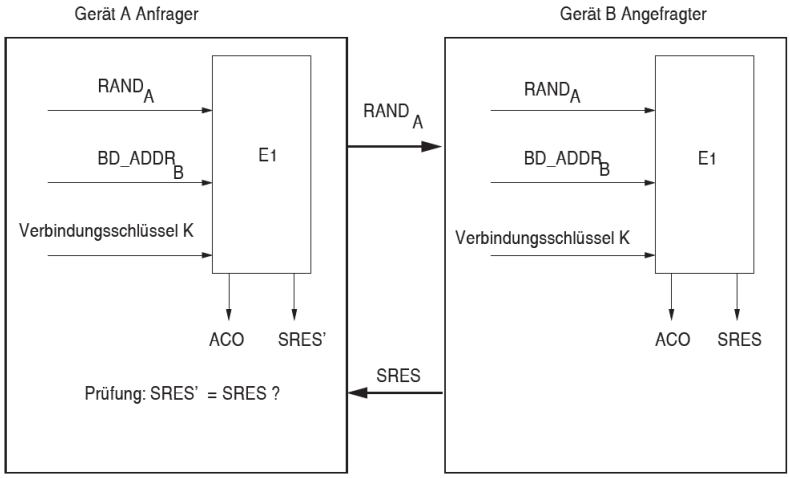

Abbildung 8. Ablauf Authentifizierungsprotokoll[2, p. 954]

Authentifizierungsprotokoll Bei der Authentifizierung sendet der Anfrager A zuerst eine 128 Bit Zufallszahl RAND an den Kommunikationspartner B. Beide berechnen anschließend eine Antwort. Diese wird mit dem E21 Algorithmus berechnet. Als Eingaben dienen die Zufallszahl RAND, die Geräteaddresse von B und ein Verbindungsschlüssel K. Haben sich die Geräte vorher bereits authentifiziert, so verfügen sie über einen gemeinsamen Verbindungsschlüssel, der beispielsweise in der Geräte-Datenbank des Sicherheitsmanagers abgelegt ist. Ist ein solcher Schlüssel nicht vorhanden, wird das Pairing initialisiert. Falls der Schlüssel vorhanden ist und die berechnete Antwort beider Geräte nicht übereinstimmt, schlägt die Authentifizierung fehl. Der Anfrager wird für eine bestimmte Zeit blockiert. Die Dauer der Sperrung steigt mit jedem weiteren fehlgeschlagenen Authentifizierungsversuch. Dadurch sollen Brute-Force Attacken verhindert werden.
Wie die beiden Geräte sich auf einen Verbindungsschlüssel einigen und diesen geheim halten, wird im folgenden Abschnitt erläutert.

Schlüsselmanagement Die Sicherheit bei Bluetooth basiert auf einer Reihe von Schlüsseln. Diese werden unterteilt in Verbindungs- und Kommunikationsschlüssel. Die Verbindungsschlüssel (link key) werden zur Berechnung des Kommunikationsschlüssels (communication key) verwendet. Mit dem Kommunikationsschlüssel wird dann der Payload der Pakete verschlüsselt.

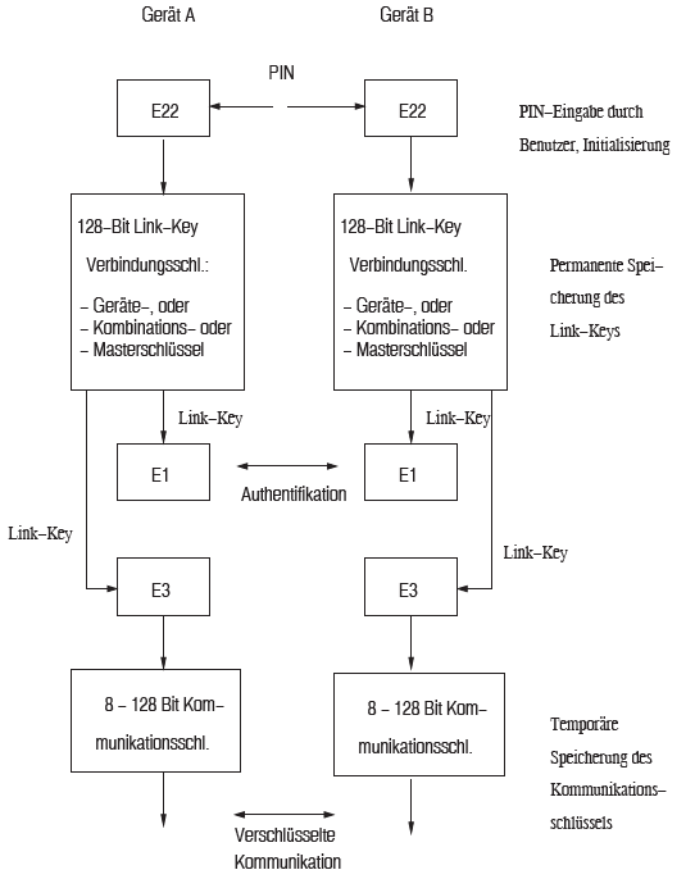

Abbildung 9. Übersicht Schlüsselmanagement[2, p. 949]

Pairing Wie bereits im Security Modus 4 erwähnt, spielt Pairing eine wichtige Rolle. Das Pairing wird beim Erstkontakt von zwei Geräten initialisiert. Dieses dient dem sicheren Austausch eines gemeinsamen Schlüssels. Um dieses durchzuführen ist es erforderlich, dass beide Geräte über ein gemeinsames Geheimnis, eine PIN, verfügen. Die Länge der PIN kann zwischen 1 und 16 Byte variieren. Folgende Möglichkeiten stehen dabei zur Verfügung:

- Auf beiden Geräten wird mittels der ESCE Komponente des Sicherheitsmanagers die gleiche PIN eingegeben.
- Eins der beiden Geräten besitzt eine feste, bekannte PIN. Diese wird auf dem zweiten Gerät eingegeben.
- Die Generierung der PIN wird auf Anwendungsebene durchgeführt. Ein Algorithmus der dafür verwendet wird ist beispielsweise Diffie-Hellman.

Initialisierungsschlüssel Nach Eingabe der PIN wird der 128 Bit Initialisierungschlüssel (IK) berechnet. Dieser wird während der weiteren Kommunikation zur Verschlüsselung der Nachrichten verwendet. Das anfragende Gerät A generiert zuerst eine Zufallszahl RAND und sendet diese an den Kommunikationspartner B. Beide Geräte berechnen anschließend mit Hilfe es E22 Algorithmus den IK. Die Eingaben bei E22 sind die PIN, PIN-Länge und die BD_ADDR von B. Eines der Geräte schickt anschließend eine Challenge in Form einer Zufallszahl an den anderen. Unter Einbeziehung der Zufallszahl, Geräteadresse und IK wird dann eine Antwort berechnet. Stimmen die Ergebnisse beider Parteien überein, wird der Vorgang fortgesetzt. Bei keiner Übereinstimmung wird der Vorgang abgebrochen und das Pairing muss erneut initialisiert werden.

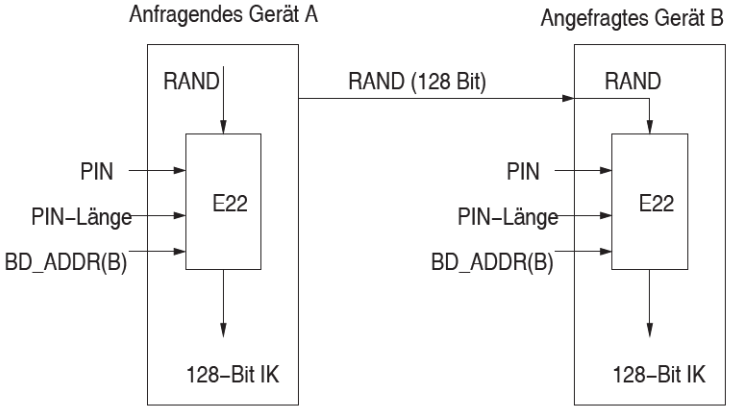

Abbildung 10. Berechnung IK[2, p. 951]

Verbindungsschlüssel Nach erfolgreicher Berechnung des IK muss der Verbindungsschlüssel (Link-key) gewählt werden. Dabei kann zwischen Geräte-, Kombinations- und Masterschlüssel gewählt werden. Jeder dieser Schlüssel ist 128 Bit lang.
Der Geräteschlüssel ist gerätespezifisch und wird bei der ersten Nutzung eines Geräts erstellt. Unter Einbeziehung der Geräteadresse und einer 128 Bit Zufallszahl berechnet der E21 Algorithmus den Schlüssel. Dieser wird anschließend im nicht flüchtigen Speicher des Geräts abgelegt.
Da Bluetooth Geräte die Verbindungsschlüssel von authentifizierten Geräten speichern, führt das bei geringem Speicher zu Problemen. Betroffene Geräte verwenden deshalb den Masterschlüssel als Verbindungsschlüssel. Dafür wird dieser exklusiv-oder mit dem IK verknüpft. Der Kommunikationspartner erhält durch erneutes exklusiv-oder verknüpfen mit dem IK den Masterschlüssel.

Masterschlüssel Der Masterschlüssel dient bei Punkt-zu-Multipunktverbindungen als Verbindungsschlüssel. Dieser ersetzt temporär den eigentlichen Verbindungsschlüssel. Der Master im Netzwerk berechnet dafür mittels des E22 Algorithmus und zwei Zufallszahlen den 128 Bit Masterschlüssel. Alle Slaves verwenden dann den Masterschlüssel zur Berechnung des Kommunikationsschlüssels. Der Master kann im weiteren Verlauf die Verwendung des ursprünglichen Verbindungsschlüssels auffordern.

Kombinationsschlüssel Bluetooth erfordert bei jeder Verbindung eine erneute Authentifizierung. Dazu gehört auch das Pairing, welches eventuell Benutzereingaben erfordert und dementsprechend Zeit kostet. Um dieses Problem zu umgehen, kann ein Kombinationsschlüssel berechnet und auf beiden Geräten persistent gespeichert werden. Der Ablauf der Schlüsselerzeugung ist in Abbildung 11 sichtbar.
Beide Geräte berechnen mittels des E21 Algorithmus, ihrer eigenen Geräteaddresse und einer Zufallszahl einen Schlüssel. Anschließend wird die Zufallszahl exklusiv-oder mit dem vorher berechneten IK verknüpft und an den jeweils anderen gesendet. Da beide Parteien über den gleichen IK verfügen, können sie durch erneutes eklusiv-oder verknüpfen der erhaltenen Nachricht die Zufallszahl des jeweils anderen erhalten. In der Abbildung 11 erkennt man nun, dass die Geräte in Schritt 3 jeweils die Berechnung des anderen in Schritt 1 durchführen. Beide erhalten damit den Schlüssel des Anderen. Durch exklusiv-oder Verknüpfung beider Schlüssel erhalten sie dann den Kombinationsschlüssel.

Nach der Wahl und Berechnung des Verbindungschlüssels kann nun das im vorherigen Abschnitt beschriebene Authentifizierungsprotokoll erfolgreich durchgeführt werden. Nach erfolgreicher Authentifizierung muss ein Kommunikationsschlüssel, der zur Verschlüsselung der Nachrichten dient, berechnet werden.

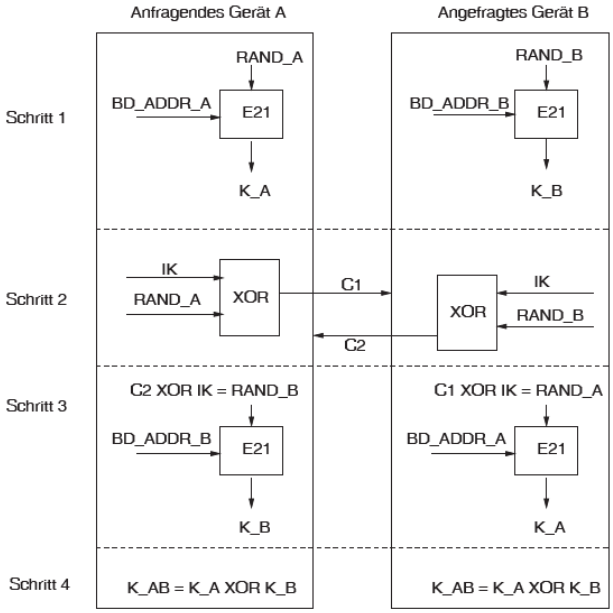

Abbildung 11. Berechnung Kombinationsschlüssel[2, p. 952]

Kommunikationsschlüssel Der Kommunikationsschlüssel wird mittels des E3 Algorithmus berechnet. Als Eingabe werden der vorher berechnete Verbindungsschlüssel, eine 128 Bit Zufallszahl und eine 96 Bit Ciphering Offset Number (COF) verwendet. Die COF wird aus dem ACO Wert des Authentifizierungsprotokolls abgeleitet. Die Länge des berechneten Schlüssels variiert zwischen 8 und 128 Bit. Dadurch kann Bluetooth auch in Ländern verwendet werden, welche die erlaubte Schlüssellänge einschränken. Jedes Gerät definiert deshalb eine maximale und minimale Länge des Kommunikationsschlüssels. Bei der Aushandlung der Schlüssellänge startet der Master immer mit der maximalen Länge. Die Länge wird dann runtergehandelt bis eine Einigung gefunden wurde. Im worst case bedeutet dies einen 8 Bit Schlüssel. Auf Anwendungsebene kann jedoch definiert werden, wie groß die minimale Schlüssellänge sein muss. Die Aushandlung wird abgebrochen, wenn diese Größe unterschritten wird.

Verschlüsselung Der Payload jeder Nachricht wird mit dem E0 Algorithmus, einer symmetrischen Stromchiffre, verschlüsselt. Als Schlüssel wird der zuvor berechnete Kommunikationsschlüssel verwendet. Durch Kryptoanalyse wurde gezeigt, dass Angriffe auf E0 mit einem Aufwand von mindestens $\mathcal{O}(2^{66})$ möglich sind. Auf Grund dieser Komplexität sind Angriffe nicht praxisrelevant.

Schwächen & Probleme

Geräteauthentifizierung Die gerätebasierte Authentifzierzung und Authorisierung ist problematisch, wenn ein Gerät entwendet wird. Da auf erneute Authentifizierung mit PIN Eingabe häufig verzichtet wird, kann der Angreifer das Gerät weiterhin verwenden, um auf alle bereits authentifizierten Geräte zuzugreifen.

Geräteschlüssel als Verbindungsschlüssel Die Verwendung des Geräteschlüssels als Verbindungsschlüssel bei Geräten mit wenig Speichern führt zu Problemen. Jedes Gerät, welches sich einmal mit einem speicherarmen Gerät authentifiziert hat, verfügt über dessen Geräteschlüssel. Da dieser als Grundlage für den Kommunikationsschlüssel dient, können alle Nachrichten entschlüsselt werden. Um diese abzuhören muss jedoch auch die richtige Frequenz gewählt werden.

Masterschlüssel Bei der Verwendung des Masterschlüssels als Verbindungsschlüssel verfügen alle Geräte in einer Multipunktverbindung über den selben Kommunikationsschlüssel. Dadurch ist es ihnen möglich, auch Nachrichten, die nicht für sie bestimmt sind, zu entschlüsseln.

PIN basiert Die Berechnung der Verbindungsschlüssel basiert auf dem Initialisierungsschlüssel, welcher wiederum auf der PIN basiert. Jegliche Schwächen der PIN beeinträchtigen dementsprechend die gesamte Sicherheit der Verschlüsselung. Da die PIN Eingabe Zeit kostet und lästig ist, wird häufig entweder eine kurze PIN vom Benutzer verwendet, oder die standardmäßig gesetzte PIN 0000 verwendet. Dies macht es einfach, die PIN zu knacken.
Eine weitere Schwäche liegt in der Bluetooth Spezifikation, welche die Übertragung der PIN über die Luftschnittstelle vorsieht. Dies macht es für Angreifer einfach, diese abzufangen. Erneut ist zu beachten, dass der Angreifer die jeweils richtige Frequenz abhören muss. Das BTCrackTool ermöglicht es aus abgefangenen Pairing-Daten und aus diesen den Verbindungsschlüssel herzuleiten.

Man-in-the-Middle Nachdem der Angreifer im Besitz des Verbindungsschlüssels ist, kann dieser ein Man-in-the-Middle Angriff durchführen. Dafür nimmt er Kontakt mit zwei Geräten auf und gibt sich als der jeweils andere aus. Damit dieser Angriff nicht erkannt wird, muss auf unterschiedlichen Frequenzen gesendet werden. Deshalb fordert der Angreifer die Ziele dazu auf, zum Netzwerkmaster zu werden. Diese können die Anfrage jedoch ablehnen, wodurch der Angriff abgewehrt wird. Geschieht dies nicht, kann der Angreifer die Geräte zur Aushandlung neuer Verbindungsschlüssel auffordern und sich daraufhin bei beiden Geräten als den jeweils anderen authentifizieren.

Secure Simple Pairing Das Secure Simple Pairing (SSP) ist eine Erweiterung des Pairings und soll Eavesdropping als auch MIM Angriffe während des Pairings verhindern. Dieses wurde bereits beim Sicherheitsmodus 4 kurz erwähnt. Das SSP ist in fünf Phasen eingeteilt:

1. Public-Key Exchange basierend auf Diffie-Hellman
2. Authentifizierung

3. Challenge
4. Berechnung Verbindungsschlüssel
5. Berechnung Kommunikationsschlüssel

Auf nährere Details der Phasen wird hier nicht weiter eingegangen. Der Hauptaspekt ist, dass die Authentifizierung des Key-Exchanges für zusätzliche Sicherheit sorgt. Die dafür zur Verfügung stehenden Optionen sind: numerischer Vergleich, Just Works, Out-of Band und Passkey Entry. Welcher dieser Varianten genutzt wird, kann im Sicherheits Modus 4 definiert werden [2].

2.4 Logical Link Control and Adaptation Protocol (L2CAP)

Genau wie das LMP befindet sich auch L2CAP im Data link layer. Dieses setzt, wie in Abbildung 2 sichtbar ist, auf ACL auf. Bluetooth ermöglicht durch das "KabelersatzprotokollRFCOMM auch die Nutzung von Protokollen, die einen seriellen Port erfodern. Dazu gehört unter anderem TCP/IP. Eine Übersicht der unterstützen Protkolle ist in der Abbildung Protokollstack sichtbar. Durch Paketfragmentierung ist es L2CAP möglich, höheren Protokollen das senden von bis zu 64KB großen Paketen zu erlauben. Das Basisband kann lediglich 341 Byte große Pakete versenden, weshalb die Fragmentierung und anschließende Zusammenfügung durch L2CAP erforderlich ist. Da das Basisbandprotokoll nicht in der Lage ist, Protokoll-Multiplexing durchzuführen, wird dies auch von L2CAP übernommen.
L2CAP stellt außerdem Kommunikationskanäle (ATT bearers) bereit, die vom Attribute Protocol verwendet werden [2].

2.5 Service Discovery Protocol

Im Abschnitt zu Sicherheitsmodi wurde bereits erwähnt, dass Sicherheitsfunktionen durch die Dienstebene bereitgestellt und umgesetzt werden. Was genau Dienste (services) sind und diese in Bluetooth funktionieren, wird in diesem Abschnitt erklärt.

Dienst Dienste sind dafür verantwortlich Informationen bereitzustellen oder Daten zu manipulieren. Sie werden in Primär- und Sekundärdienste aufgeteilt. Primärdienste stellen dabei die eigentliche Funktion bereit, während Sekundärdienste Hilfsfunktionalitäten bereitstellen. Am Beispiel eines Thermometerdienstes kann man sich das so vorstellen, dass der Primärdienst die Temperatur abliest und ein Sekundärdienst die Temperatur anschließend von Grad Celcius in Kelvin umwandelt.
Dienste können entweder als Software, Hardware oder als Kombination von beiden implementiert werden. Eine Einschränkung ist jedoch, dass backwards compability gewährleistet werden muss. Es können also zusätzliche Variablen (Characteristics) und Funktionalitäten hinzugefügt werden, die ursprüngliche

Funktion muss aber erhalten bleiben. Dies kann man sich ähnlich wie in der objektorientiertren Programmierung vorstellen. In diesem Fall wäre ein Dienst ein Interface und die Implementierung eine Klasse. Man kann zwar innerhalb der Klasse die Implementierungsdetails ändern, jedoch keine Änderungen an der Schnittstelle vornehmen.

Dienste definieren außerdem Characteristics, die man sich wie Klassenvariablen vorstellen kann. Diese werden durch 16 oder 128 Bit UUIDs identifiziert. Bei der Entwicklung eigener Dienste sollte man also darauf achten, dass man nicht bereits reservierte IDs für seinen Dienst verwendet. In der Abbildung Characteristics sieht man einen Ausschnitt von standardmäßig definierten Characteristics. Die Characteristics eines Dienstes kann man sich wie einen Key-Value Store vorstellen. Der Key ist dabei ein 16 Bit unsigned integer der innerhalb der Service Klasse eindeutig sein muss. Die Länge und Inhalte der Werte sind jeweils variabel.

Name	Uniform Type Identifier	Assigned Number	Specification
Aerobic Heart Rate Lower Limit	org.bluetooth.characteristic.aerobic_heart_rate_lower_limit	0x2A7E	GSS
Aerobic Heart Rate Upper Limit	org.bluetooth.characteristic.aerobic_heart_rate_upper_limit	0x2A84	GSS
Aerobic Threshold	org.bluetooth.characteristic.aerobic_threshold	0x2A7F	GSS
Age	org.bluetooth.characteristic.age	0x2A80	GSS
Aggregate	org.bluetooth.characteristic.aggregate	0x2A5A	GSS
Alert Category ID	org.bluetooth.characteristic.alert_category_id	0x2A43	GSS
Alert Category ID Bit Mask	org.bluetooth.characteristic.alert_category_id_bit_mask	0x2A42	GSS
Alert Level	org.bluetooth.characteristic.alert_level	0x2A06	GSS

Abbildung 12. Service Characteristics Beispiele [6]

Durch Service Klassen werden die Attribute eines Dienstes definiert. Jede Service Klassen muss eine eindeutige ID besitzen. Wie in der objectorientierten Programmierung, können Dienste auch von anderen Diensten erben. In diesem Fall werden alle Attribute der Superklasse geerbt. Ein Beispiel für Vererbung wäre ein Musikdienst, der das grundlegende Abspielen von Musik definiert. Eine weitere Service Klasse könnte dann von dieser erben und Musikstreaming über

TCP/IP ermöglichen.
Ein großer Vorteil von Diensten ist die Wiederverwendbarkeit. Dienste können andere Dienste einbinden. Der Grad-Celcius-Kelvin Dienst könnte also auch von anderen Diensten eingebunden und genutzt werden. Bei der beschriebenen Komposition gibt es keine Einschränkung bezüglich der maximal erlaubten Anzahl eingebundener Dienste.

Um Dienste auf einem Gerät ausfindig zu machen wird das Service Discovery Protocol (SDP) verwendet. Eine vereinfachte Kommunikation ist dabei in der Abbildung 13 sichtbar. Der SDP Client schickt eine Anfrage an den SDP Server und erhält anschließend Informationen über den angegebenen Dienst. Dienstinformationen werden auf dem SDP Server in einer SDP Datenbank aufbewahrt. Zu jedem Dienst gibt es einen Eintrag, der als Service Record bezeichnet wird.

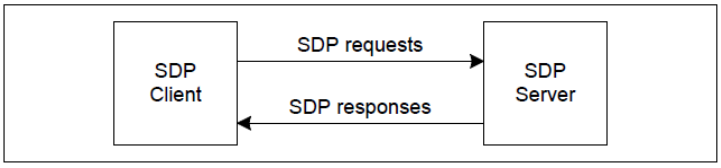

Abbildung 13. SDP Kommunikation [3, p. 1213]

Ein Service Record wird eindeutig durch eine 32 Bit ID identifiziert. Dabei darf die ID auf dem SDP Server nur einmal vergeben werden. Es ist nicht festgelegt, welcher Dienst eine jeweilige ID erhält. Der gleiche Dienst kann also auf zwei Servern unterschiedliche IDs besitzen. Die ID 0x00000000 ist teilweise für den SDP Server selbst reserviert und enthält Informationen über unterstützte Protkolle.
Eine weitere Möglichkeit Dienste aufzufinden ist über die Search Service Transaction. Dabei sendet der SDP Client eine List von Attributwerten. Der SDP Server guckt dann in den Service Records nach Einträgen in denen die gesendeten Werte als Attributwerte gesetzt sind. Sind alle Werte in dem Service Record enthalten, wird die ID des Records in der Antwort des Servers zurückgegeben. Anschließend kann der Client eine SDP Request mit einer Record ID stellen um so die vollständigen Informationen des Dienstes zu erhalten [3].

2.6 Bluetooth Profile

Ziel von Bluetooth Profilen ist die Interoperabilität von Anwendungen zu gewährleisten. Aus diesem Grund beschreibt ein Profil die Interaktion von mehreren Komponenten. Dazu gehört die Interaktion der einzelnen Schichten, das Verhalten von Anwendungen, die Schichteninteraktion von zwei Geräten, die ausgetauschten Da-

tenformate als auch die Service Discovery Anforderungen. Stimmen zwei Anwendungen in den genannten Punkten überein, so können diese Zusammenarbeiten. Beispiele für Profile sind das Advanced Audio Distribution Profile (A2DP) für audio streaming und das Video Distribution Profile (VDP) für Videoübertragung [8].

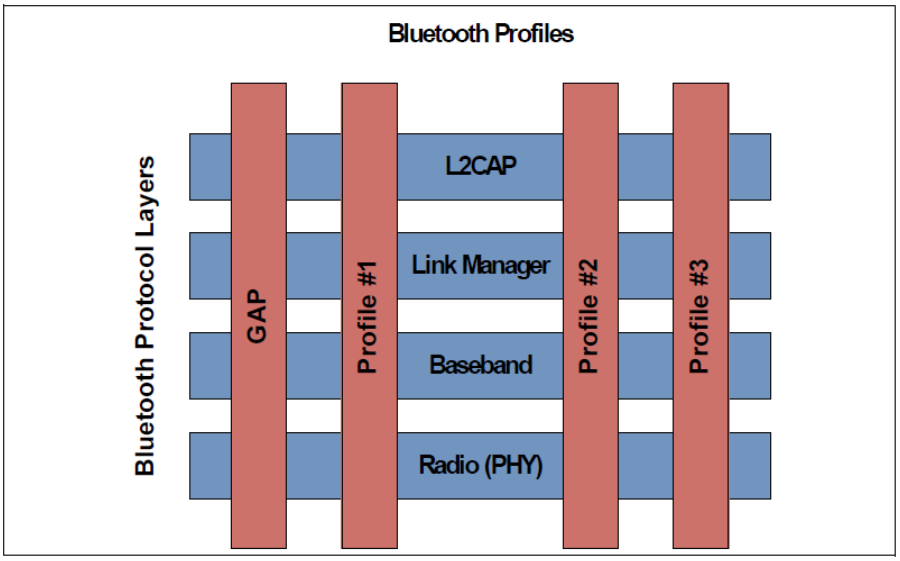

Abbildung 14. Profil [3, p. 281]

Generic Access Profile Das Generic Access Profile (GAP) wird von jedem Bluetooth Gerät implementiert und definiert grundlegende Anforderungen. Abhängig von der grundlegenden Technologie unterscheidet sich das GAP. Es ist an dieser Stelle deshalb wichtig, diese kurz zu erwähnen.

Bluetooth Technologien Bluetooth unterscheidet sich grundlegend in zwei Technologien: Basic Rate / Enhanced Data Rate (BR/EDR) und Low Enery (LE). BR/EDR ist vorallem für Anwendungsfälle mit hoher Datenrate und kurzen Entfernungen ausgelegt. Darunter fällt beispielsweise Musikstreaming. Bluetooth LE kennzeichnet sich, wie der Name schon sagt, durch einen geringeren Energieverbrauch aus. Zeitgleich verfügt es jedoch über den gleichen Funktionsumfang wie BR/EDR. Bluetooth LE wurde mit Version 4.0 eingeführt und ist nicht backwards kompatibel.
Bei BR/EDR umfasst GAP Definitionen zum Radio, Baseband, Link Manager, L2CAP und dem SDP. Bei Bluetooth LE hingegen zum Physical und Link Layer, L2CAP, dem Security Manager, Attribute Protocol und GAT.
Basierend auf dem Generic Access Profile können weitere Profile angelegt wer-

den. In diese können dann weitere Anforderungen definiert werden. Diese Profile werden als Generic Profile bezeichnet. Profile, die Anwendungsinteroperabilität beschreiben heißen Application Profile. In der Abbildung 16 ist die Hierarchie der Profile nochmal dargestellt. Man erkennt in diesem auch deutlich, dass Profile alle Spezifikation der unteren Profilschicht umfassen und entsprechend implementieren müssen.

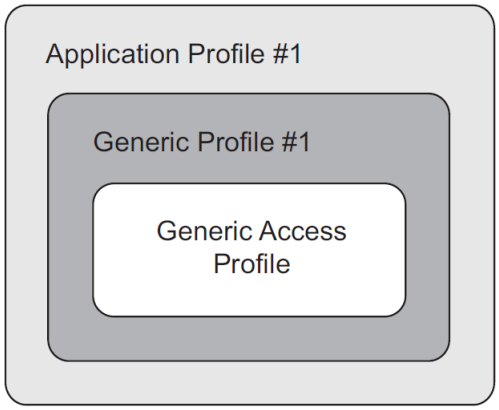

Abbildung 15. Profilhierarchie [3, p. 283]

Attribute Protocol Das Attribute Protocol (ATT) ermöglicht Clients das auslesen und manipulieren von Serverwerten. Jeder Wert ist dabei ein Attribut, dessen Datentyp durch eine UUID identifiziert wird. Das Protokoll verwendet L2CAP Kanäle zur Datenübertragung.

Generic Attribute Architecture Aufbauend auf ATT gibt es auch die Generic Attribute Architecture (GATT). Diese ist Teil der Service Discovery in Bluetooth LE. Die Implementierung in BR/EDR ist optional. Das GATT Profil definiert jeweils wie Anwendunge Daten austauschen [4].

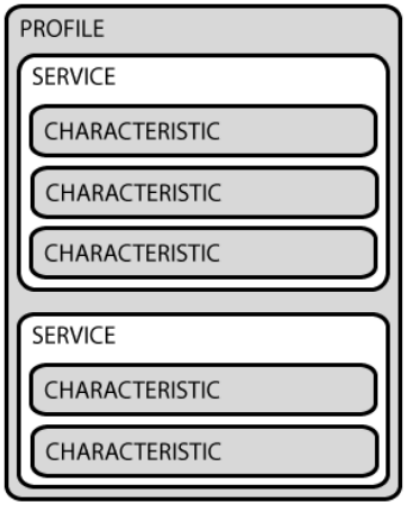

Abbildung 16. GATT Profil [2]

3 Neue Änderungen

3.1 Version 5.1

GATT Caching Bisher hat der Server den Client informiert, wenn sich seine Attribute Table geändert hat. Der Client hat dann mit einer Bestätigung geantwortet und per Service Discovery die aktuellen Werte heruntergeladen. Dafür musste der Server Informationen über jeden verbundenen Client und deren zuletzt geladenen Daten pflegen. Mit dem neuen GATT Caching ist dies nicht mehr nötig.
Der GATT Server verwaltet nun eine Liste mit Clients und dem Daten-Hash zum Zeitpunkt an dem diesen zuletzt die Daten geladen haben. Clients können dann anfragen, ob sich die Daten geändert haben. Der Server vergleicht dann den letzten Clienthash mit dem aktuellen Datenhash. Falls die Daten sich geändert haben stimmen die Hashwerte nicht mehr überein und der Client kann die neuen Daten per Service Discovery herunterladen.

Direction finding Bisherige Bluetooth Ortungssysteme benutzen die Signalstärke (RSSI) um Entfernungen abzuschätzen. Dieser Lösungsansatz wurde auch in der deutschen Corona Warn App [9] gewählt. Die neue Direction finding Funktionen ermöglichen die Bestimmung des Angle of arrival (AoA) und Angle of departure(AoD). Ein sendendes Gerät kann jeweils den AoD berechnen während der Empfänger den AoA berechnet. Um diese Werte zu berechnen sind jedoch mehrere Antennen erforderlich. In Abbildung 17 ist die Funktionsweise dargestellt.

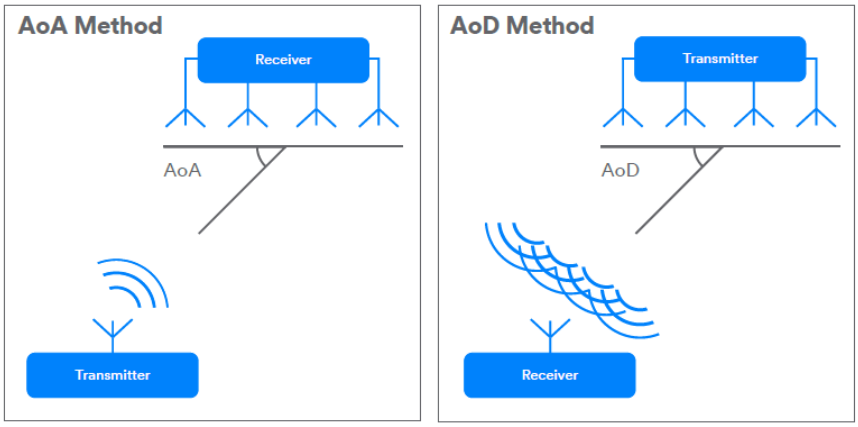

Abbildung 17. Direction finding [5]

Advertising Verbesserungen Advertising wird verwendet, um Informationen von sich selbst an Geräte in der Umgebung zu senden. Ein Problem dabei war, dass Geräte den Adversiting Channel in einer festgelegten Reihenfolge gewählt haben. Durch die zufällige Wahl eines Channels wird nun die Wahrscheinlichkeit, dass Advertising Pakete kollidieren, verringert.
Durch Periodic Advertising Sync Transfer (PAST) ist es Geräten nun möglich, dass ein Gerät Synchronisierungsdetail aus Advertisements mit anderen Geräten teilt. Dadurch können Geräte, die momentan selbst beschäftigt sind, von einem anderen Gerät aktuelle Informationen erhalten und müssen keinen eigenen Aufwand betreiben, um an diese Daten zu gelangen. Ein Beispiel dafür ist in der Abbildung 18 sichtbar [5].

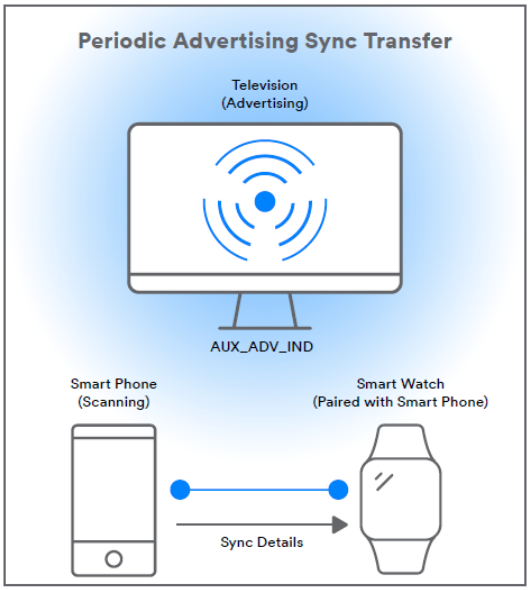

Abbildung 18. Anwendungsfall PAST [5]

3.2 Version 5.2

LE Isochronous Channel Die neuen LE Isochronous Channel ermöglichen die synchronisierte Übertragung von Daten an mehrere Kommunikationspartner. Eine Anwendung der Channel ist Musikstreaming von einem zentralen Bluetooth Gerät an mehrere Geräte in der Nähe. Die Kanäle dienen außerdem als Grundlage für LE Audio, die neue Generation der Bluetooth Audioübertragung.

Enhanced Attribute Protocol Das Enhanced Attribute Protocol (EATT) ist eine Erweiterung des Attribute Protocol. EATT erlaubt im Gegensatz zu ATT die Durchführung paralleler Transaktionen. Dadurch wird die Latenz bei Geräten, auf denen mehrere Bluetooth LE Anwendungen laufen, reduziert. EATT führte außerdem einen neuen L2CAP Kanal mit Flow control ein. Datentransporte über diese Kanäle sind also deutlich zuverlässiger. Ein Sicherheitsvorteil von EATT ist, dass dieses nur über eine verschlüsselte Verbindung genutzt werden kann.

LE Power Control LE Power Control ermöglicht es Kommunikationspartnern die Signalstärke, die zur Übermittlung verwendet wird, dynamisch anzupassen. Dabei wird an Hand der von Empfängern wahrgenommenen Signalstärke eine Verstärkung oder Abschwächung der Signalstärke angefordert. Durch die Anpassung der Signalstärke ist es möglich Energie zu sparen. Die Anpassung der Signalstärke verbessert außerdem die Zuverlässigkeit, da sichergestellt wird,

dass diese in einem optimalen Bereich liegt. Bei geringerer Signalstärke werden Geräte, die sich in der Nähe befinden weniger beeinflusst, da die Reichweite, in der Interferenzeffekte auftreten können, reduziert wird. Dies hat Vorteile für alle Geräte, welche den 2,4 GHz Frequenzbereich verwenden. Besonders auf kleinen Räumen mit vielen Bluetoothgeräten wird dies für Verbesserungen der Übertragungsqualität sorgen [4].

4 Fazit

Die ursprünglichen Ziele eine günstige und stromsparende Lösung für kurze Distanzen zu finden wurden erreicht. Durch die Bluetooth LE Technologie und neue Funktionen wie LE Power Control wird der Energieverbrauch weiter reduziert. Durch die Verwendung von Profilen und Diensten wurde es nicht nur geschafft die drahtlose Kommunikation zu vereinheitlichen sondern auch die Anwendungen, die auf dieser aufbauen. Die Wiederverwendbarkeit und Komposition von Diensten macht es außerdem für Entwickler einfacher neuer Anwendung zu entwickeln.
Die Sicherheitsprobleme bei Bluetooth entstehen vorallem durch die gerätebasierte Authentifizierung und die Abhängigkeit vom PIN. Mit Secure Simple Pairing wurde jedoch eine Lösung gefunden, um den PIN Austausch sicherer zu gestalten und das Risiko von Eavesdropping als auch dadurch möglichen Man-in-the-middle Angriffe zu reduzieren.
Ein weiterer Negativpunkt ist, dass Bluetooth standardmäßig keine Ende-zu-Ende Sicherheit sondern nur Punkt-zu-Punkt Sicherheit bietet. Ist E2E Sicherheit gewünscht, muss diese durch Protokolle wie SSL/TLS zusätzlich hinzugefügt werden.
Zwar bietet Bluetooth Sicherheitsmodi an, die einen grundlegenden Schutz und Verschlüsselung anbieten, standardmäßig sind Modus 2 und 3 jedoch deaktiviert, wodurch der Endbenutzer keinen Vorteil daraus zieht.
Auf Grund dieser Bedenken sollte Bluetooth nur für sicherheitskritische Anwendungen verwendet werden, wenn die Anwendung selbst zusätzliche Sicherheitskonzepte umsetzt oder diese durch Einbindung anderer Dienste hinzufügt.

Literatur

1. Bluetooth Architektur, Wikipedia https://upload.wikimedia.org/wikipedia/commons/thumb/9/9f/Bluetooth_protokoly.svg/1920px-Bluetooth_protokoly.svg.png
2. Eckert, C.: IT-Sicherheit: Bluetooth. Walter de Gruyter(2013)
3. Bluetooth Core Specification Version 5.2, Version 1, https://www.bluetooth.com/de/specifications/bluetooth-core-specification/
4. Bluetooth Core Specification Version 5.2 Feature Overview, Version 1, https://www.bluetooth.com/wp-content/uploads/2020/01/Bluetooth_5.2_Feature_Overview.pdf

5. Bluetooth Core Specification Version 5.1 Feature Overview, Version 1, https://www.bluetooth.com/bluetooth-resources/bluetooth-core-specification-v5-1-feature-overview/
6. https://www.bluetooth.com/de/specifications/gatt/characteristics/, abgerufen: 22.06.2020
7. https://image.slidesharecdn.com/lecture3spreadspectrumtechnologies-151115073210-lva1-app6892/95/spread-spectrum-technologies-14-638.jpg?cb=1447572747, abgerufen: 22.06.2020
8. https://de.wikipedia.org/wiki/Bluetooth-Profile, abgerufen: 25.05.2020
9. https://github.com/corona-warn-app/cwa-documentation, abgerufen: 22.06.2020