Bluetooth Security

Secorvo White Paper

Sicherheitsmechanismen des Bluetooth Standards (Version 1.1)

Version 1.0
Stand 09. September 2002

Dirk Fox

Secorvo Security Consulting GmbH
Albert-Nestler-Straße 9
D-76131 Karlsruhe

Tel. +49 721 6105-500
Fax +49 721 6105-455

E-Mail info@secorvo.de
Internet http://www.secorvo.de

Inhaltsübersicht

Abkürzungen

ACO	Authenticated Ciphering Offset (96 bit)
AES	Advanced Encryption Standard (NIST-Standard)
BD_ADDR	Bluetooth Device (Endgerät) Adresse (48 bit)
COF	Ciphering Offset Number (96 bit)
DECT	Digital Enhanced Cordless Telekommunication
E_0	Stromchiffre, LFSR Summation Generator (Verschlüsselung)
E_1	Berechnung des Authentikators auf Basis der Blockchiffre SAFER+
E_2 (E_{21}, E_{22})	modifizierte Blockchiffre SAFER+ (Keyed Hash), Generierung des Link Key
E_3	mod. Blockchiffre SAFER+ (Keyed Hash), Generierung des Encryption Key
IEEE	Institute of Electrical and Electronics Engineers
IrDA	Infra-red Data Association
ISDN	Integrated Services Digital Network
ISM	Industrial, Scientific, Medical (2,4 GHz-Band)
L2CAP	Logical Link Control and Adaptation Protocol
LAN	Local Area Network
LFSR	Linear Feedback Shift Register (linear rückgekoppeltes Schieberegister)
LMP	Link Management Protocol
PIN	Personal Identification Number
RFCOMM	Serial Cable Emulation Protocol (ETSI TS 07.10)
RSSI	Received Signal Strength Indication
SAFER	Blockchiffre von Massey [Mass_94]
SIG	Special Interest Group
SRES	Signed Response
TDD	Time-Division Duplex
WLAN	Wireless Local Area Network
XOR	logische Verknüpfung „Exklusiv-Oder“

Historie

Version	Datum	Änderung	Autor
1.0	09.09.02	Erste veröffentlichte Fassung	Dirk Fox

1 Zusammenfassung

Für den Aufbau drahtloser Ad-Hoc-Verbindungen über kurze Distanzen zwischen Geräten unterschiedlichster Art ist der Bluetooth-Standard gedacht, der von inzwischen mehr als 2.000 Herstellern unterstützt wird. Sollte es gelingen, die Bluetooth-Funktionen auf einem einzigen Chip zu einem Preis von weniger als zwei US$ zu realisieren, könnte Bluetooth konkurrierende Kommunikationsstandards wie DECT, WLAN, IrDA, ISDN oder sogar UMTS auf der „letzten Meile“ zu verdrängen. Aber auch bei einer geringeren Marktdurchdringung ist zu erwarten, dass Bluetooth zukünftig im großen Stil auch für die Übertragung sensibler Daten genutzt wird. Damit kommt den Sicherheitsmechanismen des Bluetooth Standards erhebliche Bedeutung zu. Das vorliegende White Paper gibt einen Überblick über die in Version 1.1 der Spezifikation vorgesehenen Mechanismen und bewertet das damit erreichbare Sicherheitsniveau.

2 Einführung

Der Bluetooth Standard verdankt seine Entstehung einer Initiative von Ericsson Mobile Communications. Zusammen mit Nokia, IBM, Intel und Toshiba gründete Ericsson im Mai 1988 die „Special Interest Group“ (SIG) mit dem Ziel, einen herstellerunabhängigen Standard für eine funkbasierte Peer-to-Peer-Datenkommunikation über kurze Distanzen zu schaffen, die sich zu geringen Hardwarekosten realisieren lässt. Dieser Initiative traten mit u. a. Lucent, Microsoft, Motorola und 3Com weitere wichtige Anbieter von Telekommunikations-systemen und -software bei. Inzwischen gehören der SIG mehr als 2.000 Hersteller an.

Der Name „Bluetooth“ ist Programm: Der Standard wurde nach dem Wikinger Harald Bluetooth, König von Dänemark (940-981 n. Chr.) benannt, der die Christianisierung und die Vereinigung von Dänemark und Norwegen bewirkte. In Jelling (Dänemark) errichtete Harald Bluetooth einen Runenstein mit der Inschrift: „König Harald errichtete dieses Monument zu Ehren von Gorm, seinem Vater, und Thyre, seiner Mutter, der (selbe) Harald, der alle Dänen und Norweger gewann und die Dänen zu Christen machte.“ In Analogie wurde im September 1999 am Hauptsitz von Ericsson Mobile Communications in Lund ein Runenstein zu Ehren von Harald Bluetooth errichtet.

Version 1.0 der Bluetooth Spezifikation wurde am 05.07.1999 fertig gestellt und am 26.07.1999 von der SIG verabschiedet. Version 1.0b folgte im Dezember 1999. Eine erheblich überarbeitete und erweiterte Spezifikation (Version 1.1) wurde am 01.12.2000 abgeschlossen und am 22.02.2001 frei gegeben. Sie umfasst neben einem generischen die Spezifikation spezieller Nutzungsprofile, u. a. für kabellose Telefonie, Headsets und LAN-Zugänge.

Ende des Jahres 2000 erfolgten erste Freigaben von Bluetooth Produkten; das erste Bluetooth Netz entstand auf der CeBIT 2001. Inzwischen sind eine große Zahl von Bluetooth Anwendungen auf dem Markt, vom Handy-Zubehör wie z. B. Headsets über kabellose ISDN-Zugänge bis zu kompletten funkbasierten LAN-Lösungen.

Die Bluetooth Protokolle wurden so spezifiziert, dass sie weitest gehend in Hardware realisiert werden können. Ziel ist, sie in einem eigenen „Bluetooth Chip“ zusammen zu fassen, der in großer Auflage so günstig hergestellt werden kann, dass auch die Ausstattung von billiger Massenware mit Bluetooth Kommunikationstechnik wirtschaftlich möglich wird. Im Jahr 2001 lagen die Herstellungskosten für einen Bluetooth Chip bei ca. 20 US$; im Jahr 2002 soll der Preis auf etwa fünf US$ sinken. Angestrebt wird ein Preis unterhalb von zwei US$ bis zum Jahr 2005.

Damit könnte die dem Namen innewohnende Prophezeihung – nomen est omen – aufgehen und Bluetooth die Nachfolge von WLAN-, IrDA- und DECT-Lösungen antreten. In jedem Fall kommt den Sicherheitsmechanismen des Bluetooth Standards zukünftig erhebliche Bedeutung zu.

3 Eigenschaften des Bluetooth Standards

Bluetooth ist ein Standard für die kabellose Funk-Datenübertragung über relativ kurze Distanzen. Anders als die inzwischen weit verbreitete Infrarot-Technologie erfordert Bluetooth keine Sichtverbindung zwischen sendender und empfangender Einheit. Bluetooth wurde entwickelt mit dem Ziel, Datenkommunikationsdienste zwischen Endgeräten auf der Basis einer „Peer-to-Peer"-Spontankommunikation zu ermöglichen. Daher unterstützt der Standard auch den Datenaustausch zwischen mehreren Endgeräten über Mehrpunktverbindungen.

Drei verschiedene Typen von Bluetooth Kommunikationsverbindungen sind möglich:

- **Punkt-zu-Punkt-Verbindung** zwischen genau zwei Bluetooth Einheiten: Dabei agiert eine Einheit (Bluetooth Device, BD) als Master, die andere als Slave.
- **Piconet**: Kleines Netz von bis zu acht Bluetooth Einheiten; auch hier hat ein BD die Funktion des Masters, alle anderen maximal sieben BDs sind Slaves.
- **Scatternet**: Zusammenschluss von bis zu zehn Piconets; hier übernimmt jeweils ein „Gateway"-Device gegenüber dem eigenen Piconet die Funktion des Masters, reagiert jedoch gegenüber dem Master des Scatternet wie ein Slave.

Für die Datenübertragung wird das Frequenzband von 2.400 bis 2.483,5 MHz (2,4 GHz-Band im Mikrowellenbereich), auch als ISM (Industrial, Scientific, Medical) bekannt, verwendet. Dieser Frequenzbereich darf in fast allen Ländern der Welt genehmigungsfrei und ohne Einschränkungen genutzt werden und ist inzwischen auch für Sprachübertragung zugelassen.

Um Störungen durch Interferenzen und Fading zu minimieren, überträgt Bluetooth die Daten in einzelnen Paketen mit Time-Division Duplex (TDD) unter Verwendung von 0,625 ms langen Zeitschlitzen und nutzt Frequency Hopping mit 1.600 Frequenzwechseln je Sekunde.

Der Standard unterteilt das Frequenzband in 79 Kanäle[1] mit einem Kanalabstand von 1 MHz und einer Kanalbandbreite von entweder 64 kbit/s für synchrone Sprachkanäle (z.B. zur Sprachübertragung), zweier symmetrischer 433,9 kbit/s Bänder oder einer transparenten (asymmetrischen) Übertragung mit einer Bandbreite von 723,2 kbit/s mit einem 57,6 kbit/s Rückkanal.

Hinsichtlich der Sendeleistung und Reichweite werden drei Geräteklassen unterschieden:

- Class 1: Sendeleistung 1-100 mW (0 bis 20 dBm, Reichweite bis ca. 100 m)
- Class 2: Sendeleistung 0,25-2,5 mW (-6 bis 4 dBm, Reichweite um ca. 10 m)
- Class 3: Sendeleistung bis 1 mW (bis 0 dBm, Reichweite ca. 0,1-10 m)

Mit Rücksicht auf die erwartungsgemäß überwiegend mobilen Endgeräte, die ihren Energiebedarf aus Akkus speisen, wurde in Bluetooth Mechanismen zur Senkung des Stromverbrauchs besondere Bedeutung beigemessen:

---

1 In Frankreich ist das Frequenzband auf 2,4465 bis 2,4835 GHz beschränkt und umfasst daher nur 23 Kanäle.

- Die Sendeleistung eines Class 3 Bluetooth Geräts wird automatisch über Empfangssignalmessungen reguliert (Received Signal Strength Indication, RSSI).
- Es wurden drei verschiedene „Ruhe-Modi“ mit geringerem Strombedarf spezifiziert: Sniff Mode, Hold Mode und Park Mode.

Auf diese Weise wurden auch die Reichweite von Bluetooth Signalen auf das Notwendige begrenzt und das Abhören aus größerer Entfernung erschwert. Allerdings sind die Mechanismen zur Steuerung der Signalstärke optional und werden nicht von jedem Bluetooth Gerät unterstützt.

Der Bluetooth Core-Standard spezifiziert neben der physikalischen Übertragungsschicht eine Sicherungsschicht (Link Layer), ein Host Controller Interface (HCI) zwischen Bluetooth Device und Anwendungskomponente, ein Logical Link Control and Adaptation Protocol (L2CAP) sowie das Transportprotokoll RFCOMM (Serial Cable Emulation Protocol) mit einer RS232-Emulation, die ein Bluetooth Device gegenüber einer Anwendung wie ein serieller Port erscheinen lässt.

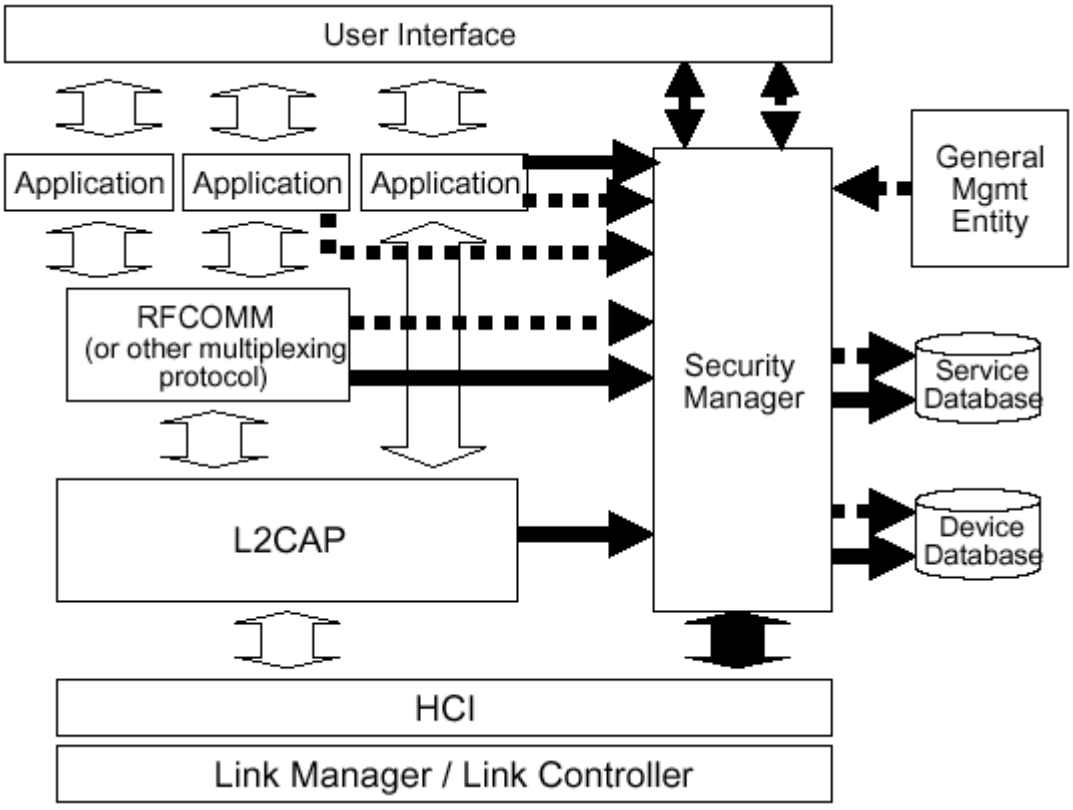

Abbildung 1: Schichtenmodell der Bluetooth Architektur

Die Link-Schicht umfasst nehmen Verfahren zur Fehlerkorrektur auch die Sicherheitsmechanismen des Standards, die in den folgenden Kapiteln ausführlich dargestellt werden. Die Etablierung einer Verbindung auf Link-Ebene wird durch das Link Manager Protokoll (LMP) gesteuert. Bluetooth Datenpakete (auf Link-Ebene) können bis zu 2745 Nutzdaten-Bits (Payload) enthalten. Jedes Paket beginnt mit 72 bit Zugangsdaten (Access Code, fest) und einem 54 bit langen Paketkopf (Header).

4 Sicherheit von Bluetooth

Die Sicherheitsarchitektur des Bluetooth Standards ist vor allem auf zwei zentrale Bedrohungen ausgerichtet: die unberechtigte Teilnahme eines Bluetooth Endgeräts an einer Bluetooth Verbindung und das unberechtigte Abhören von auf der Luftschnittstelle übertragenen Daten.

Der Schwerpunkt der Sicherheitsarchitektur liegt auf der sicheren Authentifikation der an einem Bluetooth Kommunikationsnetz beteiligten Endgeräte. In den Sicherheits-Betriebsarten 2 und 3 ist zusätzlich die Verschlüsselung der Nutzdaten als optionaler weiterer Sicherheitsdienst vorgesehen.

Der Anwendungsebene ist die Bereitstellung und Nutzung von auf den Benutzer bezogenen Sicherheitsdiensten vorbehalten, wie der Nicht-Abstreitbarkeit oder der Verwendung digitaler Signaturen. Diese Dienste wurden im Standard nicht spezifiziert.

Maßnahmen gegen Angriffe auf die Verfügbarkeit, wie z. B. zum Schutz vor Störsignalen oder einer Denial of Service Attacke auf die Energiereserven eines mobilen Bluetooth Geräts, beispielsweise durch ein gezieltes „Hochregeln“ der Sendeleistung über RSSI, wurden im Bluetooth Standard nicht berücksichtigt.

4.1 Sicherheitsarchitektur

In der Sicherheitsarchitektur werden drei Sicherheits-Betriebsarten unterschieden:

- **Non-Secure Mode** (Sicherheitsmodus 1): In dieser Betriebsart werden keine speziellen Sicherheitsmechanismen genutzt. Eine Authentifikation von Endgeräten findet nicht statt. Das Abhören der Kommunikation wird lediglich durch Frequency Hopping mit 1.600 Frequenzwechseln pro Sekunde zwischen allen 79 Kanälen erschwert.
- **Service-Level Enforced Security** (Sicherheitsmodus 2): Wird die Sicherheit auf die Anwendungsebene (Application Layer) verlagert, ist diese für die Auswahl und die Nutzung der Bluetooth-Sicherheitsmechanismen zuständig.
- **Link-Level Enforced Security** (Sicherheitsmodus 3): Auf der Verbindungsschicht (Link Layer, Schicht 2) bietet der Bluetooth Standard zwei Sicherheitsdienste: eine kryptografische Authentifikation sowie die Verschlüsselung der übertragenen Nutzdaten. In diesem Sicherheitsmodus ist die Authentifikation Bestandteil des Verbindungsaufbaus; die Verschlüsselung der übertragenen Daten ist optional.

Die spezifizierten Sicherheitsmechanismen wurden – abgesehen von Frequency Hopping – in der Verbindungsschicht (Link Layer) realisiert. Hier wurden die für die Sicherheitsdienste Authentizität und Vertraulichkeit erforderlichen Protokolle und Algorithmen implementiert. Sie werden gesteuert durch das Link Management Protocol (LMP).

Der Bluetooth Standard verwendet die folgenden vier primären Sicherheitsparameter:

- die Bluetooth Device Address (**BD_ADDR**), eine weltweit eindeutige IEEE 48-bit-Adresse, die für jedes Bluetooth Gerät vergeben wird;
- 128 bit lange Zufallswerte (**RAND**), die von einem (Pseudo-) Zufallszahlengenerator in einem Bluetooth Device erzeugt werden;[2]

[2] Die Güte dieser, für die Qualität der im weiteren erzeugten Schlüssel wichtigen Zufallszahlengeneratoren ist implementierungsabhängig und kann daher sehr unterschiedlich sein.

- ein **Unit Key** (128 bit), der üblicherweise einmalig bei der Erzeugung einer Bluetooth Einheit erzeugt wird und anschließend nur in Ausnahmefällen geändert wird, sowie
- eine konfigurierbare, bis 128 bit lange geheime Endgeräte-Kennung (**PIN**); üblich ist eine Länge von vier Octets (32 bit).[3]

Aus diesen vier primären Sicherheitsparametern werden alle weiteren Sicherheitsparameter wie der Initialisierungs-, der Authentifikations- (Link Key) und der Verschlüsselungsschlüssel abgeleitet.

4.2 Authentifikation

Ist der Sicherheitsmodus 3 (Link-Level Enforced Security) gewählt, erfolgt automatisch beim Verbindungsaufbau auf Link-Ebene eine gegenseitige Authentifikation der beteiligten Bluetooth Geräte.[4] Dieser Authentifikationsprozess setzt sich aus zwei Phasen zusammen:

- einer Initialisierungsphase, in der der Link Key gewählt und vereinbart wird, sowie
- einem Authentifikationsprotokoll zwischen zwei oder mehreren Endgeräten mit der Möglichkeit, anschließend einen Encryption Key zu vereinbaren (siehe Abschnitt 4.4.3).

4.2.1 Initialisierungsphase

Die Initialisierungsphase umfasst drei Schritte: Die Generierung eines Initialisierungsschlüssels, die Generierung bzw. Auswahl eines Link Keys und die Vereinbarung des Link Keys mit dem (oder den) beteiligten Bluetooth Device(s).

Der 128 bit lange Initialisierungsschlüssel (K_{init}) wird für den Schutz der Datenkommunikation zwischen den beteiligten Bluetooth Endgeräten während der Initialisierungsphase benötigt. Das Verfahren zur Erzeugung von K_{init} ist in Abschnitt 4.4.3 beschrieben; die Generierung wird vom Master (Verifier) durch das Kommando LMP_in_rand initiiert, die den Zufallswert IN_RAND an den Slave (Claimant) übermittelt.

Je nach Anwendung und Möglichkeiten der beteiligten Bluetooth Endgeräte wird dann einer der folgenden 128 bit langen Schlüssel als Link Key gewählt:

- Der **Unit Key**: Jedes Bluetooth Device besitzt einen festen, 128 bit langen Schlüssel, der bei der erstmaligen Verwendung aus einem Zufallswert und der Bluetooth Device-Adresse erzeugt wird (siehe Abschnitt 4.4.3). Der Unit Key wird praktisch nie geändert und sollte daher nur in Ausnahmefällen als Link Key verwendet werden. Die Verwendung kann sinnvoll sein, wenn eines der beteiligten Endgeräte über zu wenig temporären Speicher für weitere Kommunikationsschlüssel verfügt.
- Der **Combination Key**: Ein Combination Key hängt von beiden an einer Bluetooth Verbindung beteiligten Geräten ab und wird für jede Session neu erzeugt. Er besteht aus der XOR-Verknüpfung zweier Teilschlüssel, die die beiden Bluetooth Geräte zunächst unabhängig voneinander erzeugen. Die Teilschlüssel werden wie ein „temporärer Unit Key“ aus der eigenen Bluetooth Device-Adresse und einem Zufallswert berechnet (siehe Abschnitt 4.4.3). Die dabei verwendeten Zufallswerte werden von

---

3 Nicht jedes Bluetooth Device verfügt über eine PIN; dieser Sicherheitsparameter kann also – abhängig vom Endgerät – wertlos sein. Gelegentlich verwenden Bluetooth Geräte auch feste, vorausgenerierte PINs.

4 Im Sicherheitsmodus 2 muss die Authentifikation von der Anwendung initiiert werden.

beiden Geräten mit dem Initialisierungsschlüssel K_{init} verschlüsselt (XOR-verknüpft) und an das jeweils andere Bluetooth Device übertragen. Damit können beide Geräte den jeweils anderen Teilschlüssel erzeugen und durch XOR-Verknüpfung beider Teilschlüssel den gemeinsamen Combination Key bestimmen.

- Der **Master Key**: Um eine aufwändige geschützte Punkt-zu-Mehrpunkt-Kommunikation (Broadcasts) in Piconets mit unterschiedlichen Verbindungsschlüsseln zu vermeiden, wurde im Standard die Möglichkeit der temporären Nutzung eines gemeinsamen Authentifikations- (und damit auch Verschlüsselungs-) Schlüssels geschaffen. Ein solcher Master Key ersetzt die bestehenden Link Keys für einen begrenzten Zeitraum. Die Erzeugung eines Master Keys wird in Abschnitt 4.4.3 erläutert. Der Austausch des Master Keys erfolgt durch eine verschlüsselte Übertragung (XOR-Verknüpfung mit einem temporären „Overlay"-Schlüssel) vom Master an den Slave.

Der vereinbarte Link Key übernimmt anschließend die Funktion des Authentifikationsschlüssels. Nach Abschluss der Authentifikation kann der Link Key jederzeit geändert werden. Dabei übernimmt der aktuelle Link Key die Funktion des Initialisierungsschlüssels.

4.2.2 Authentifikationsprotokoll

Bluetooth verwendet als Authentifikationsprotokoll ein zweischrittiges Challenge Response Protokoll mit einem symmetrischen Schlüssel (Link Key). Das Protokoll setzt bei beiden beteiligten Endsystemen die Kenntnis des gemeinsamen Link Keys voraus, der in der vorausgegangenen Initialisierungsphase erzeugt und vereinbart wurde.

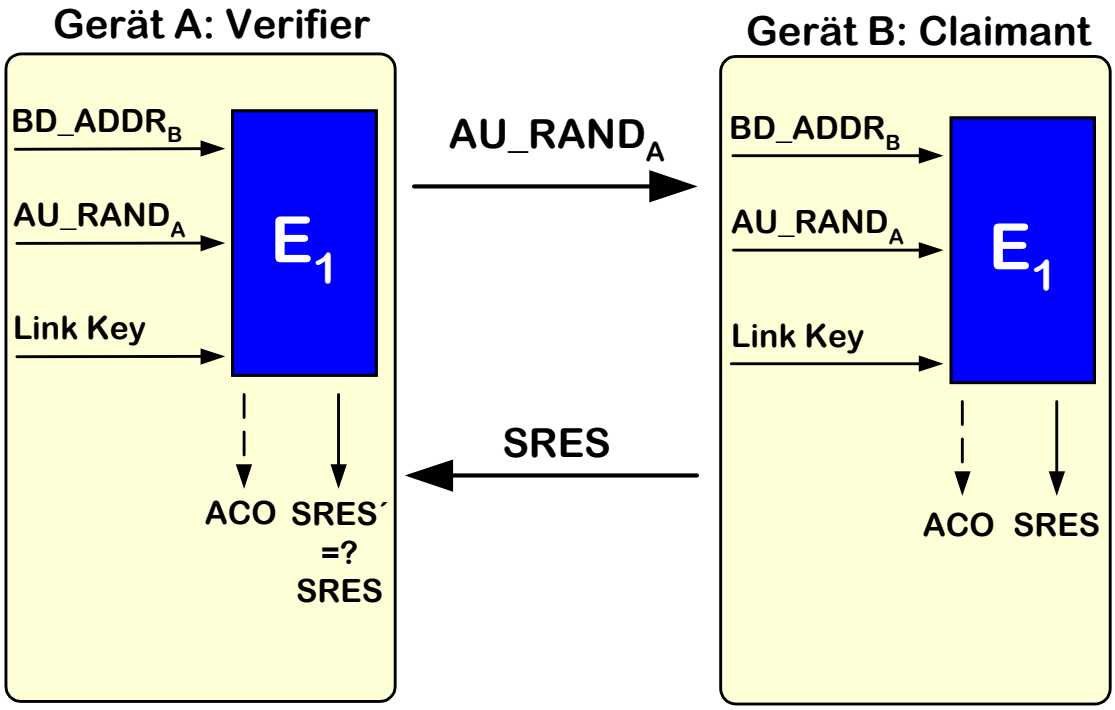

Abbildung 2: Challenge-Response-Authentifikationsprotokoll zweier Bluetooth Geräte

Die Authentifikation wird – üblicherweise vom Master-Device[5] – durch das Kommando LMP_au_rand initiiert. Dabei wird eine zuvor beim Master (Verifier) erzeugte Zufallszahl (AU_RAND) als Challenge an den Slave (Claimant) übertragen.

Aus dem empfangenen Zufallswert AU_RAND, der eigenen Bluetooth Device-Adresse BD_ADDR und dem Link Key berechnet der Slave (Claimant) mit dem Algorithmus E_1 eine 32 bit lange „Signed Response“ (SRES). Dieser Wert wird mit dem Kommando LMP_sres an den Master (Verifier) zurückgesandt.

Der Verifier vergleicht die Antwort des Slaves mit dem Ergebnis seiner eigenen Berechnung (siehe Abbildung 2). Für eine gegenseitige Authentifikation wird das Protokoll ein weiteres Mal vom Slave initiiert.

Um Brute-Force- und Denial-of-Service-Angriffe zu erschweren, kann eine Authentifikation bei einem Fehlschlag erst nach einem gewissen Zeitintervall wiederholt werden. Mit jedem weiteren Authentifikationsversuch des selben Bluetooth Devices vergrößert sich das Zeitintervall exponentiell.

4.3 Verschlüsselung

Zum Schutz der Vertraulichkeit der übertragenen Daten bietet der Bluetooth Standard eine Verschlüsselung auf Link-Ebene an. Diese Verschlüsselung kann in den Sicherheitsmodi 2 und 3 genutzt werden und setzt eine vorangegangene erfolgreiche Authentifikation voraus. Sie verwendet eine Stromchiffre mit einem bis zu 128 bit langen Verschlüsselungsschlüssel (siehe Abschnitt 4.4.3).

Die Vereinbarung des Veschlüsselungsmodes erfolgt über das LMP-Kommando LMP_encryption_mode_req (encryption mode = 1 oder 2) durch den Master. Dieses Kommando muss vom Slave mit LMP_accepted bestätigt werden. Anschließend wird die Länge des Verschlüsselungsschlüssels zwischen den beiden Einheiten ausgehandelt. Dafür legt die Applikation eine Mindestschlüssellänge (L_{min}) fest. Jedes Bluetooth Device wiederum besitzt eine feste Einstellung der maximal unterstützten Schlüssellänge (L_{max}). Beide Längen können einen Wert von 1 bis 16 (in Octets) annehmen.

Als gemeinsame Schlüssellänge wird der größte Wert der Schnittmenge

$$\{L_{min}, ..., 16\} \cap \{1, ..., L_{max}\}.$$

gewählt. Zur Aushandlung der Schlüssellänge sendet das Master-Device mit dem LMP-Kommando LMP_encryption_key_size_req die maximale Schlüssellänge an den (oder die) Slave(s). Schickt der Slave ein LMP_accepted, ist diese Schlüssellänge vereinbart. Antwortet der Slave mit LMP_not_accepted, schickt der Master ein LMP_encryption_key_size_req mit der nächst kleineren Schlüssellänge – sofern die Mindestschlüssellänge noch nicht erreicht ist.

Akzeptiert der Slave auch die Mindestschlüssellänge nicht, kommt die gewünschte sichere Verbindung nicht zu Stande. Dadurch wird verhindert, dass ein Bluetooth Device, das eine extrem kleine maximale Schlüssellänge angibt, die Etablierung einer unsicheren Verbindung bewirken kann.

---

5 Wird das Bluetooth Device im Sicherheitsmode 2 betrieben, legt die Anwendung fest, welche Einheit sich gegenüber welcher anderen authentifizieren muss. Letztere initiiert dann das Authentifikationsprotokoll.

Nach Vereinbarung der Schlüssellänge wird die Verschlüsselung vom Master mit dem Kommando LMP_start_encryption_req gestartet und der Verschlüsselungsschlüssel K_c mit dem Algorithmus E_3 aus dem Link Key und einer Zufallszahl EN_RAND erzeugt (genauer siehe Abschnitt 4.4.3). Die 128 bit lange Ausgabe der Schlüsselgenerierung wird auf die ausgehandelte Schlüssellänge (8-128 bit) reduziert.

Bei jedem Aufruf der Verschlüsselungsfunktion (LMP_start_encryption_req) wird ein neuer Verschlüsselungsschlüssel gewählt.

4.4 Sicherheitsmechanismen

Für die technische Umsetzung der Authentifikations- und Vertraulichkeitsmechanismen auf Verbindungsebene (Link Level) sind im Bluetooth Standard neben den vier primären Sicherheitsparametern die folgenden drei Schlüssel spezifiziert[6]:

- ein geheimer, 128 bit langer **Initialisierungsschlüssel** (**K_{init}**): temporärer Schlüssel, der nur zu Beginn der Initialisierungsphase verwendet und anschließend verworfen wird.
- ein geheimer, 128 bit langer **Authentifikationsschlüssel** (**Link Key**): wird während der Initialisierungsphase erzeugt und ist entweder semi-permanent (Verwendung in mehreren Sessions) oder wird für jede Session neu gewählt.
- ein geheimer **Verschlüsselungsschlüssel** (**Encryption Key**): mit Rücksicht auf existierende Exportrestriktionen und Kryptoregulierungen in einzelnen Ländern ist er in der Länge konfigurierbar auf 1-16 Octets; er wird für jede Session neu aus dem Link Key abgeleitet.

Diese sieben Sicherheitsparameter werden von den folgenden fünf kryptografischen Algorithmen genutzt bzw. erzeugt:

- einem **Zufallszahlengenerator**,
- einem **Authentifikator-Algorithmus** zur Berechnung der Signed Response SRES (E_1),
- zwei Algorithmen zur **Schlüsselerzeugung** (E_2 für den Authentifikations- und E_3 für den Verschlüsselungsschlüssel), und
- einem **Verschlüsselungsalgorithmus** (E_0).

Die jeweils verwendeten Algorithmen und ihre Sicherheitseigenschaften werden im Folgenden dargestellt.

4.4.1 Zufallszahlengenerator

Jedes Bluetooth Device verfügt über einen kryptografischen Zufallszahlengenerator. Das kann entweder ein echter, physikalischer Zufallsprozess oder auch ein in Software implementierter Pseudozufallsalgorithmus sein. Von dem verwendeten (Pseudo-) Zufallsverfahren fordert der Standard, dass die erzeugten Zufallsbits sich während der Lebenszeit des Authentication Keys nicht wiederholen und zufällig, d. h. nicht vorhersagbar generiert werden. Genauer ausgedrückt: Eine L bit lange Zufallsfolge darf nicht mit einer Erfolgswahrscheinlichkeit größer $1/2^L$ geraten werden können.

---

6 Die Sicherheitsmechanismen des *Bluetooth* Standards sind Gegenstand von Kapitel 14 der *Bluetooth* Spezifikation Version 1.1 vom 22.02.2001 [Blue_01].

Die Güte der Zufallszahlengeneratoren (und damit auch die der mit Zufallswerten bestimmten Schlüssel) hängt allerdings von der Qualität der jeweiligen Implementierung ab und kann daher stark variieren.

4.4.2 Berechnung des Authentikators

Zur Berechnung des Authentikators wird der Authentifikationsalgorithmus E_1 verwendet. E_1 besteht aus der Hintereinanderausführung zweier Blockchiffren: SAFER+ und einer geringfügigen Modifikation von SAFER+ (Algorithmus E_2, siehe Abschnitt 4.4.3).

SAFER+ ist eine Variante der ursprünglich von Massey entwickelten 64 bit Blockchiffre SAFER-SK 128 (Schlüssellänge 128 bit) [Mass_94], die frei verfügbar ist und ohne Lizenzgebühren genutzt werden kann. Eine detaillierte Darstellung von SAFER+ findet sich in [Blue_01].

Wie in Abschnitt 4.2.2 beschrieben hat E_1 drei Eingabeparameter:

- den Link Key (128 bit),
- einen Zufallswert AU_RAND (128 bit) und
- die Device-Adresse BD_ADDR (48 bit) des Masters.

Der Link Key dient als Schlüssel für SAFER+ und E_2. Der 128 bit lange Zufallswert AU_RAND wird mit SAFER+ verschlüsselt und mit dem Ergebnis XOR-verknüpft. Das Resultat wird mit der auf 128 bit expandierten Device-Adresse des Masters UND-verknüpft und dann mit E_2 verschlüsselt.

Die 128 bit lange Ausgabe wird in eine 32 bit lange Signed Response (SRES) und einen 96 bit langen Authenticated Ciphering Offset (ACO) aufgeteilt. Der ACO wird für die mögliche spätere Generierung eines Verschlüsselungsschlüssels aufbewahrt.

4.4.3 Schlüsselgenerierung

4.4.3.1 Erzeugung des Initialisierungsschlüssels K_{init}

Für die Erzeugung des Initialisierungsschlüssels K_{init} wird der Algorithmus E_2, eine leicht modifizierte Variante der Blockchiffre SAFER+ verwendet. E_2 unterscheidet sich von SAFER+ in einer zusätzlichen Additionsoperation, die den Input der ersten zum Input der dritten Runde addiert. Dadurch ist E_2 im Unterschied zu SAFER+ nicht invertierbar und kann somit nicht zur Verschlüsselung genutzt werden.

Der Initialisierungsschlüssel K_{init} wird mit dem Algorithmus E_2 in Mode 2 (kurz: E_{22}, siehe Abbildung 3) bestimmt aus

- der eindeutigen Bluetooth Device-Adresse BD_ADDR (48 bit),
- einem im Bluetooth Device erzeugten 128 bit langen Zufallswert IN_RAND,
- einem zwischen den beiden Bluetooth Endgeräten „out of band“ vereinbarten PIN-Code (1 bis 16 Octets) und der Länge dieser PIN (in Octets).[7]

[7] Bluetooth Geräte können nach dem Standard auch eine fest voreingestellte oder sogar keine PIN besitzen. Im letzten Fall sind PIN-Länge und PIN gleich „0“.

Ausgabe von E_2 ist ein 128 bit langer Schlüssel (K_{init}). Dieser wird nur während der Initialisierungsphase des Verbindungsaufbaus verwendet und verfällt spätestens nach dem Ende der Session, d. h. der zu einem anderen Bluetooth Gerät aufgebauten Verbindung bzw. der „Teilnahme“ an einem Piconet. Nach der Erzeugung des Schlüssels wird außerdem der Wert der PIN um BD_ADDR erhöht, um einen zweiten Eingabeparameter des Generierungsprozesses zu ändern.[8]

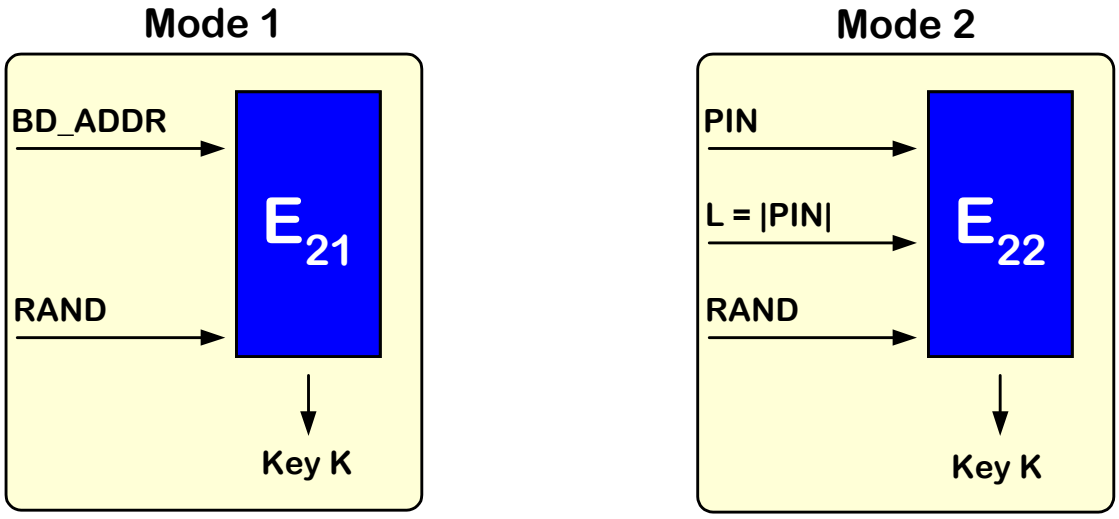

Abbildung 3: Erzeugung des Unit Keys und des Initialisierungsschlüssels mit E_{21} bzw. E_{22}

4.4.3.2 Erzeugung von Unit Keys und Combination Keys

Auch für die Generierung des Unit Keys wird der Algorithmus E_2 verwendet, allerdings in einer anderen Betriebsart (Mode 1, kurz: E_{21}). Die Erzeugung des Unit Keys erfolgt einmalig bei der erstmaligen Nutzung eines Bluetooth Geräts. Der 128 bit lange Schlüssel wird aus der Device-Adresse BD_ADDR des Bluetooth Geräts und einem 128 bit langen Zufallswert (RAND) gewonnen (siehe Abbildung 3). Der erzeugte Unit Key wird im nicht-flüchtigen Speicher des Bluetooth Geräts abgelegt und üblicherweise nie mehr geändert.

Auf die selbe Weise werden die Teilschlüssel eines Combination Keys mit E_{21} aus der Device-Adresse und einer selbst generierten Zufallszahl erzeugt.

4.4.3.3 Erzeugung eines Master Keys

Ein Master Key wird mit dem Algorithmus E_{22} aus zwei Zufallszahlen des Masters (RAND = RAND1, PIN = RAND2) und mit L = 16 erzeugt.[9]

Anschließend generiert der Master einen weiteren Zufallswert RAND und überträgt diesen offen mit dem Kommando LMP_temp_rand zum Slave. Mit E_{22} leiten Master und Slave aus dem aktuellen Link Key, dem übertragenen Zufallswert RAND und L = 16 daraufhin einen temporären Verschlüsselungsschlüssel ab („Overlay“, OVL). Der Master verschlüsselt den

---

8 Verwendet die Einheit eine feste PIN, wird die PIN der anderen Einheit entsprechend erhöht. Arbeiten beide *Bluetooth* Einheiten mit einer festen PIN, ist keine Authentifikation möglich.

9 Als Master Key wird keine Zufallszahl verwendet, um zu verhindern, dass ein schwacher Zufallszahlengenerator des Masters zu einer Kompromittierung des Master Keys führt.

Master Key durch eine XOR-Verknüpfung mit OVL und überträgt das Ergebnis mit dem Kommando LMP_temp_key an den Slave.

4.4.3.4 Erzeugung des Verschlüsselungsschlüssels

Die Erzeugung eines Verschlüsselungsschlüssels K_C erfolgt mit dem Algorithmus E_3 aus den folgenden drei Parametern (siehe Abbildung 4):

- dem 128 bit langen Authentifikationsschlüssel (Link Key),
- einem ebenfalls 128 bit langen, vom Master erzeugten und mit dem LMP-Kommando LMP_start_encryption_req übertragenen Zufallswert (EN_RAND) und
- einem 96 bit Ciphering Offset (COF).

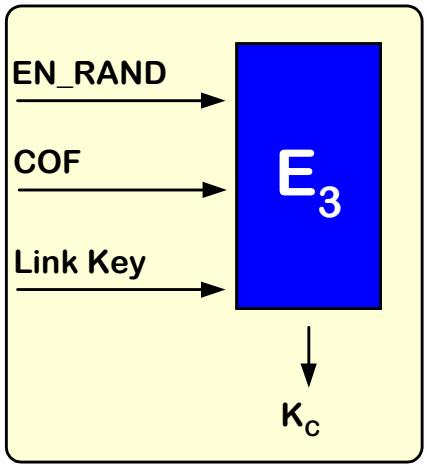

Abbildung 4: Bestimmung eines Verschlüsselungsschlüssels K_C mit E_3

Als COF wird der bei der Authentifikation bestimmte, neben SRES verbleibende 96 bit lange Authenticated Ciphering Offset (ACO) verwendet.[10]

4.4.4 Verschlüsselungsverfahren

Für die Verschlüsselung wird der Algorithmus E_0 verwendet. Die Stromchiffre basiert auf einem von Massey und Rueppel entwickelten Summengenerator aus vier linear rückgekoppelten Schieberegistern (LFSR) [MaRu_84, Ruep_86, Ruep_92]. Die LFSR werden aus den folgenden primitiven Rückkoppelungspolynomen mit Hamming-Gewicht fünf[11] und einer gesamten Registerlänge von 128 gebildet:

- $L_1 = 25$, $f_1(t) = t^{25} + t^{20} + t^{12} + t^8 + 1$
- $L_2 = 31$, $f_2(t) = t^{31} + t^{24} + t^{16} + t^{12} + 1$
- $L_3 = 33$, $f_3(t) = t^{33} + t^{28} + t^{24} + t^4 + 1$

---

10 Ausnahme: Wird ein Master Key als (temporärer) Link Key verwendet, bilden die beiden 48 bit langen Bluetooth Device-Adressen der beteiligten Endgeräte den COF.

11 Je kleiner das Hamming-Gewicht, desto geringer die Zahl der benötigten XOR-Gatter und zugleich auch Chipgröße und Kosten. Wird das Hamming-Gewicht zu klein, sinkt die Qualität der statistischen Eigenschaften – und damit die kryptografische Güte des LFSR.

- $L_4 = 39$, $f_4(t) = t^{39} + t^{36} + t^{28} + t^4 + 1$

Der vom Summengenerator, einer Zustandsmaschine mit insgesamt 16 verschiedenen Zuständen erzeugte Schlüsselstrom K_{cipher} hängt dabei von vier Parametern ab:

- dem vereinbarten geheimen Verschlüsselungsschlüssel K_C (Länge: 8-128 bit),
- einer (offen übertragenen) 128 bit langen Zufallszahl EN_RAND, um Wiederholungen des Schlüsselstroms vernachlässigbar unwahrscheinlich zu machen,
- der eindeutigen, 48 bit langen Device-Adresse (BD_ADDR) sowie
- 26 Bit des Clock-Signals (CLK) des Masters.

$$K_{cipher} = E_0(K_C, BD\_ADDR, CLK, EN\_RAND)$$

Abbildung 5 zeigt schematisch den Ablauf einer verschlüsselten Übertragung zwischen zwei Bluetooth Einheiten. Der Master (im Bild Gerät A) initiiert die Verschlüsselung mit dem Kommando LMP_start_encryption. Dabei gibt er dem Slave (Gerät B) die für die Erzeugung des Schlüsselstroms zu verwendende Zufallszahl EN_RAND vor.

Aus den vier Eingabeparametern werden der effektive Verschlüsselungsschlüssel bestimmt und die vier LFSR initialisiert. Dazu werden die Datenbits in eine geeignete Reihenfolge gebracht und als Startwerte in die insgesamt 128 Register eingetragen. Mit diesen Werten werden 239 bit des Schlüsselstroms erzeugt, von denen 128 bit wiederum als Initialwerte in die 128 Register geladen werden.

Ab dem 240sten Takt wird der von dem Summengenerator erzeugte Schlüsselstrom auf Seder- und Empfängerseite mit dem Datenstrom bitweise XOR-verknüpft.

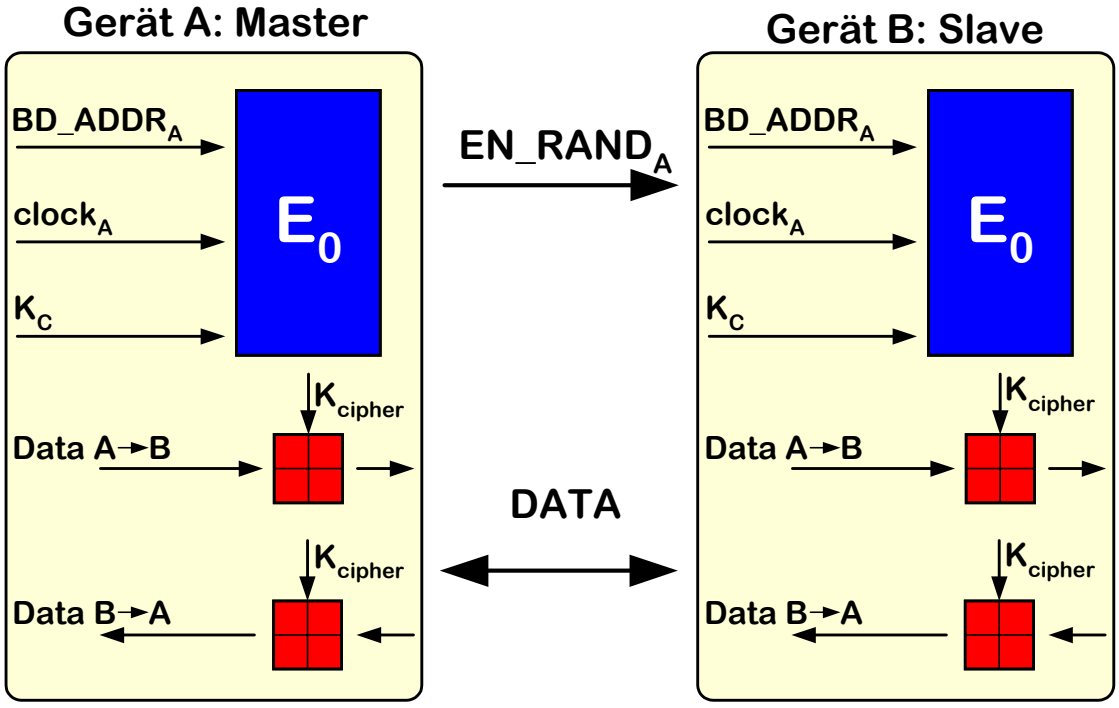

Abbildung 5: Ablauf der Verschlüsselung mit der Stromchiffre E_0

Da Summengeneratoren aus linear rückgekoppelten Schieberegistern bekanntermaßen anfällig gegen Korrelationsangriffe sind, erfolgt mit jedem Datenpaket eine Resynchronisation.

5 Bewertung

Wie in vielen Kommunikationsstandards sind auch bei Bluetooth die spezifizierten Sicherheitsmechanismen optional. Daher ist zu erwarten, dass – wie bei WLANs – mit hoher Wahrscheinlichkeit der überwiegende Teil der Anwendungen Bluetooth gänzlich ohne Sicherheitsmechanismen nutzen wird.

Aber auch die verwendeten primären Sicherheitsparameter leiden unter sicherheitskritischen Kompromissen:

- Die Qualität der für die Güte der erzeugten Schlüssel wesentlichen Zufallszahlengeneratoren ist implementierungsabhängig, da keine Algorithmen vorgeschrieben oder spezifiziert wurden. Zweifellos werden daher nicht nur Geräte mit geringer Rechenleistung kryptografisch schwache Verfahren verwenden.
- Die Nutzung des Unit Keys, der üblicherweise nur einmalig erzeugt und nie mehr geändert wird, als fester Link Key führt dazu, dass dieser Eingabeparameter von E_1 (zur Berechnung des Authentikators) und E_3 (zur Berechnung des Verschlüsselungsschlüssels) nicht mehr variiert.
- Die PIN kann ein unzuverlässiger Sicherheitsparameter sein, da es Bluetooth Geräte geben wird, die eine feste oder sogar gar keine PIN verwenden. Aber auch eine PIN, die vom Benutzer gewählt, jedoch auf die Länge von einem Octet begrenzt werden kann, ist kein zuverlässiger Sicherheitsparameter. Schließlich kann auch eine lange PIN schlecht gewählt oder ohne die Beachtung von Sicherheitsmaßnahmen vereinbart werden, so dass ein Angreifer Kenntnis von der PIN erlangt.

Besonders schwer wiegt jedoch, dass der Initialisierungsschlüssel K_{init} von der PIN als einzigem geheimen Parameter abhängt, denn BD_ADDR ist bekannt und IN_RAND wird unverschlüsselt vom Master an den Slave übermittelt. Ein Angreifer, der die PIN kennt (oder erraten kann), kennt damit auch den Initialisierungsschlüssel und kann anschließend alle mit diesem verschlüsselt übertragenen Daten mitlesen. Dadurch erfährt er den Link Key und gewinnt auch jeden daraus abgeleiteten Verschlüsselungsschlüssel [JaWe_01].

Schließlich sind die verwendeten kryptografischen Algorithmen nicht mehr „State of the Art“: Auf den verwendeten LFSR-Summengenerator E_0 wurden inzwischen Angriffe mit einer kryptoanalytischen Komplexität von 2^{66} (gegenüber einer Komplexität von 2^{127} einer Brute-Force-Attacke) veröffentlicht [Goli_97]. Und auch zur Blockchiffre SAFER+ – einem der 15 AES-Kandidaten – wurden im Rahmen des AES-Auswahlprozesses Schwächen bekannt [KeSW_99]; SAFER+ gehörte daher nicht zu den fünf Algorithmen der Endauswahl.

Als Gegenmaßnahmen ist dreierlei zu empfehlen:

- Die Mindestanforderungen an die Zufallszahlengeneratoren sollten z.B. durch die Spezifikation geeigneter Algorithmen angehoben werden.
- An die PIN müssen höhere Anforderungen gestellt werden. Ein kryptografisch ausreichendes Sicherheitsniveau wird – bei zufälliger Wahl der PIN – erst bei einer Mindestlänge von 10 Octets (80 bit) erreicht.
- Die verwendete Stromchiffre E_0 sowie die den Algorithmen E1, E2 und E3 zu Gunde liegende Blockchiffre SAFER+ sollte durch die in Hard- und Software sehr effizient implementierbare symmetrische Blockchiffre AES ersetzt werden [NIST_01].

6 Literatur

[Blue_01] Bluetooth: *Bluetooth Specification v1.1* (22.02.2001), http://www.bluetooth.com/developer/specification/specification.asp

[Daid_01] Mc Daid, Cathal: *Bluetooth Security*, Feb. 2001 http://www.palowireless.com/bluearticles/cc1_security1.asp

[Goli_97] Golić, J.: *Cryptanalysis of Alleged A5 Stream Cipher*. In: Fumy, W. (Hrsg.): Proceedings of Eurocrypt '97, LNCS 1233, Springer 1997, S. 239-255.

[JaWe_01] Jakobsson, M.; Wetzel, Susanne: *Security Weaknesses in Bluetooth* (19.02.2001), RSA Security Conference 2001 http://www.bell-labs.com/user/markusj/bluetooth.pdf

[KaOw_02] Karygiannis, Tom; Owens, Les: *Wireless Network Security (Draft)*. National Institute of Standards and Technology (NIST), Special Publication 800-48, 24.07.2002. http://csrc.nist.gov/publications/drafts/draft-sp800-48.pdf

[KeSW_99] Kelsey, J.; Schneier, Bruce; Wagner, D.: *Key schedule weaknesses in SAFER+,* The Second AES Conference, March 22-23, 1999, S. 155-167.

[MaRu_84] Massey, James L.; Rueppel, Rainer A.: *Linear Ciphers and Random Sequence Generators with Multiple Clocks*. In: Beth, T.; Cot, N.; Ingemarsson, I. (Hrsg.): Proceedings of Eurocrypt 84, LNCS 209, Springer 1984, S. 74-87.

[Mass_94] Massey, James L.: *SAFER K-64: A byte-oriented block-ciphering Algorithm*. In: Anderson, R. (Hrsg.): Fast Software Encryption Workshop, LNCS 809, Springer 1994, S. 1-17.

[Müll_99] Müller, Thomas: *Bluetooth Security Architecture* (15.07.1999), http://www.bluetooth.com/developer/whitepaper/whitepaper.asp

[NIST_01] National Institute of Standards and Technology (NIST): *Advanced Encryption Standard (AES)*. Federal Information Processing Standards Publication 197 (FIPS-PUB), 26.11.2001.

[Pohl_01] Pohl, Winfried: *Bluetooth: Technik und Einsatzgebiete*, KES 1/2001, S. 43-49

[Ruep_86] Rueppel, Rainer: *Analysis and Design of Stream Ciphers*. Springer-Verlag, Berlin, 1986.

[Ruep_92] Rueppel, Rainer A.: *Stream Ciphers*. In: Simmons, G.J. (Hrsg.): Contemporary Cryptology: The Science of Information Integrity. IEEE Press, New York 1992, S. 65-134.

[Stie_02] Stiegler, Leonhard: *Datensicherheit in Bluetooth und Wireless-LAN-Funknetzen*. Unterrichtsblätter, Nr. 7/2002, S. 332-341.

[Vain_00] Vainio, Juha T.: *Bluetooth Security.* 25.05.2000, http://www.niksula.cs.hut.fi/~jiitv/bluesec.html