Sicherheitsanalyse von DECT ULE

Kolloquium zur Masterarbeit

Paul Schaefer

04. Februar 2019

Übersicht

Motivation und Ziel

- Digitalisierung und stetige Zunahme von vernetzten Geräten

- Besondere Anforderungen
  - geringe Verkabelung
  - hohe Reichweiten
  - lange Laufzeiten

$\Rightarrow$ dedizierte Smart Home Protokolle

Motivation und Ziel

- Digitalisierung und stetige Zunahme von vernetzten Geräten
- Besondere Anforderungen
  - geringe Verkabelung
  - hohe Reichweiten
  - lange Laufzeiten

⇒ dedizierte Smart Home Protokolle

- Sicherheit von Smart Home Protokollen
  - Confidentiality
  - Integrity
  - Availability

⇒ Analyse der Sicherheit notwendig

Verwandte Arbeiten

- deDECTed
  - Reversing von Hardware & Software
  - Angriffe auf DECT Telefone
  - Spoofing von Basestations
  - schwache Zufallszahlengeneratoren
  - DSAA2 & DSC2
- Implementationen für Software Defined Radios
  - ReDECTed zum Speichern von DECT-Rohdaten
  - gr-dect2 zum Abhören von DECT-Telefonaten

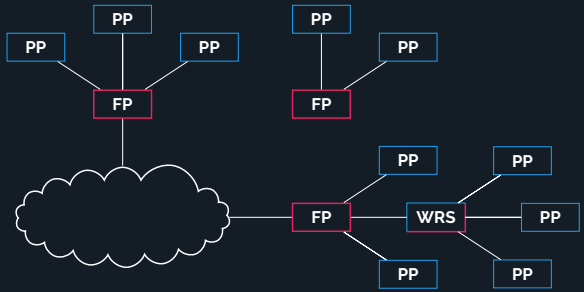
Subprojects

- Reversing DSAA project
- Reversing DSC project
- DSC security analysis project
- DSAA security analysis project
- DSAA implementation project
- DSAA FPGA implementation project
- DECT PRNG analysis project
- COM-ON-AIR windows driver project
- COM-ON-AIR Linux driver project
- DECT Linux kernel stack project
- CON-ON-AIR kismet integration project
- DECT USRP project
- Fritzbox sniffer project

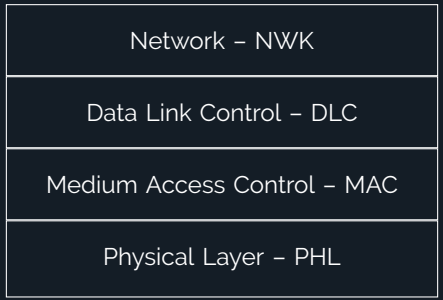
https://www.dedected.org/trac

Topologie

- Zentrale Basisstation:
  ***Fixed Part* (FP)**
- (mobile) Clients:
  ***Portable Part* (PP)**
- Repeater:
  ***Wireless Relay Station* (WRS)**

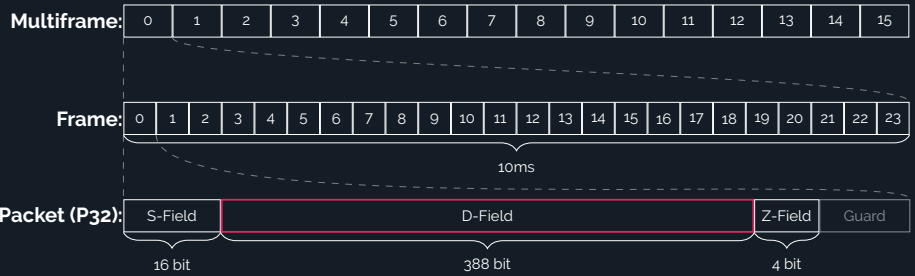


Netzwerkschichten



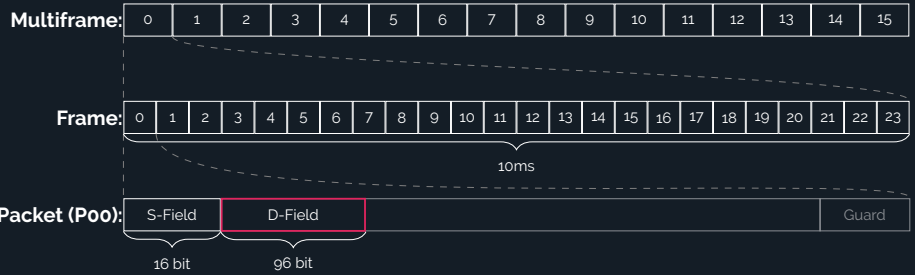
Physical Layer

- 10 Kanäle im Frequenzbereich 1880 MHz - 1980 MHz
- Gaussian Frequency Shift Keying (GFSK), 1 bit pro Symbol
- Time Division Multiple Access (TDMA)



Physical Layer

- 10 Kanäle im Frequenzbereich 1880 MHz - 1980 MHz
- Gaussian Frequency Shift Keying (GFSK), 1 bit pro Symbol
- Time Division Multiple Access (TDMA)

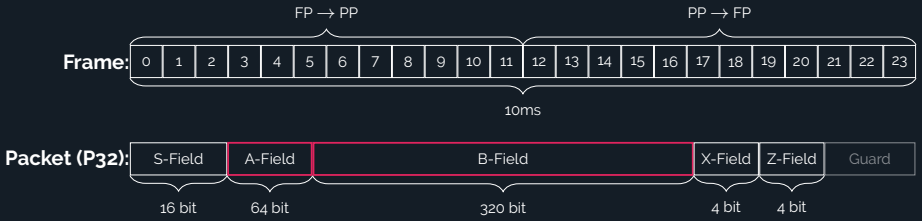


Medium Access Control Layer

- D-Field besteht aus A-Field und B-Field
- Downstream in Slots 0-11, Upstream in Slots 12-23
- Bearer (simplex und duplex)
- Simplex Bearer für *System Information Broadcast*

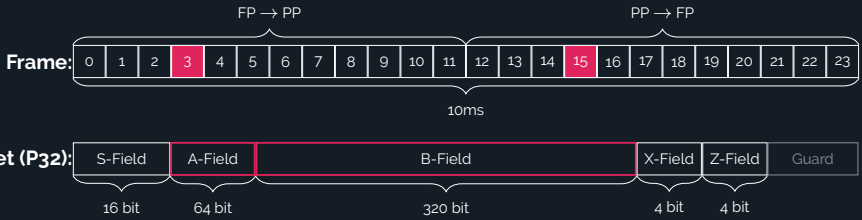
Medium Access Control Layer

- D-Field besteht aus A-Field und B-Field
- Downstream in Slots 0-11, Upstream in Slots 12-23
- Bearer (simplex und duplex)
- Simplex Bearer für *System Information Broadcast*

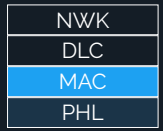


Medium Access Control Layer

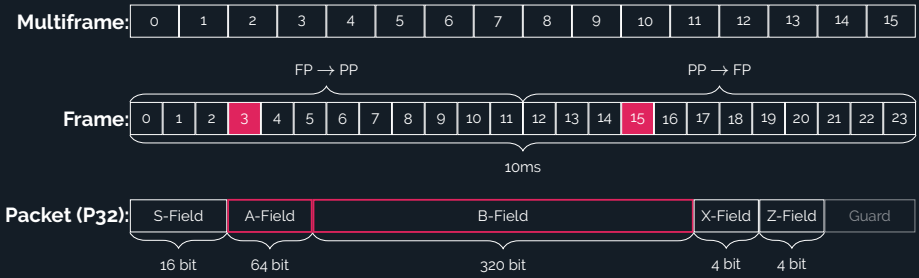
- D-Field besteht aus A-Field und B-Field
- Downstream in Slots 0-11, Upstream in Slots 12-23
- Bearer (simplex und duplex)
- Simplex Bearer für *System Information Broadcast*



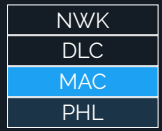
Medium Access Control Layer



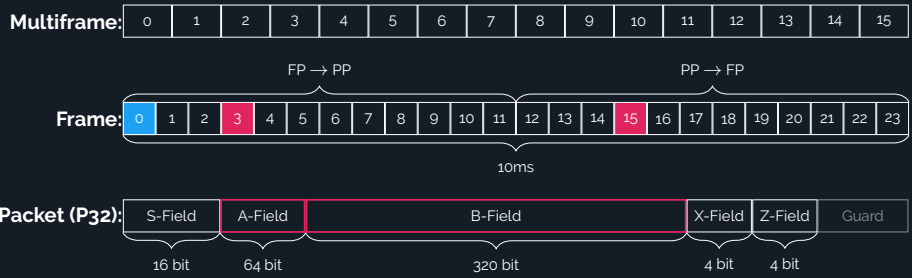
- D-Field besteht aus A-Field und B-Field
- Downstream in Slots 0-11, Upstream in Slots 12-23
- Bearer (simplex und duplex)
- Simplex Bearer für *System Information Broadcast*



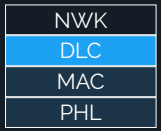
Medium Access Control Layer



- D-Field besteht aus A-Field und B-Field
- Downstream in Slots 0-11, Upstream in Slots 12-23
- Bearer (simplex und duplex)
- Simplex Bearer für *System Information Broadcast*



Data Link Control Layer



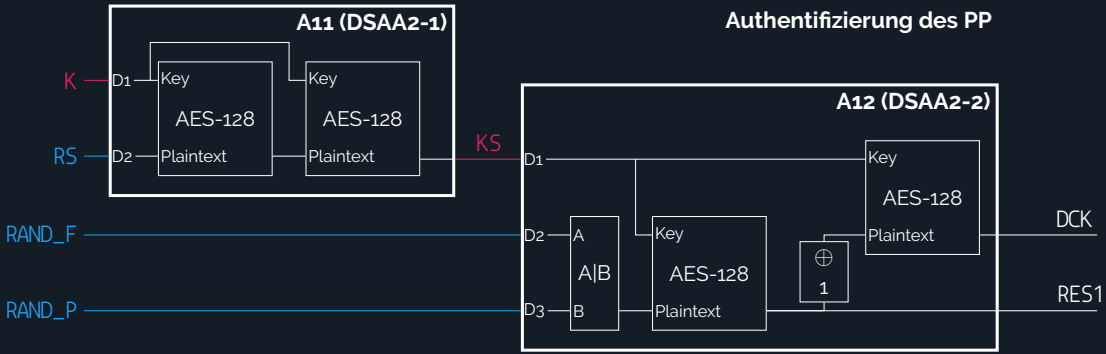
- Bereitstellung rudimentärer Protokollfeatures
  - Fragmentation & Reassembling
  - Retransmission bei Fehlern
  - Verschlüsselung
  - …

- **LU14**: *Enhanced Frame RELay service with CCM (EFREL-CCM)*
  Verschlüsselung mit AES-CCM, Versand über *LU10*
- **LU10**: *LU10 Enhanced Frame RELay (EFREL) service*
  Fragmentation & Reassembling, Retransmission bei Fehlern
- **LU13**: *LU13 Enhanced Frame RELay service with CRC (EFREL-CRC)*
  Checksum mit 16 oder 32 bit langer CRC

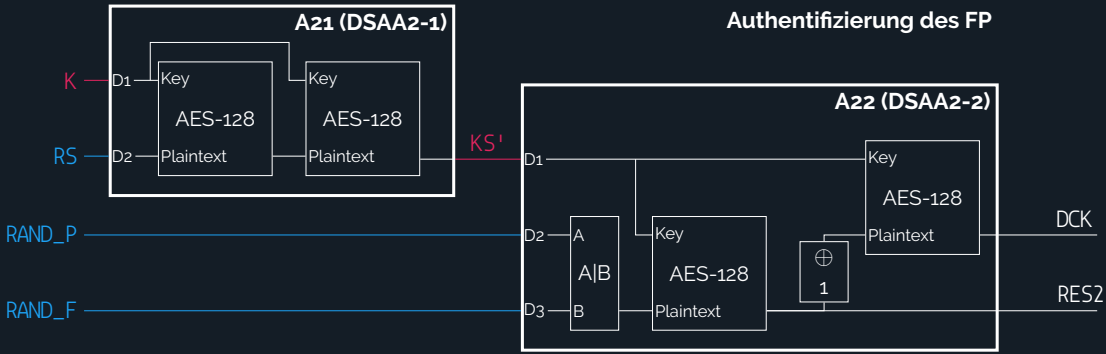
Network Layer

- Mobility Management
  - Schlüsselverwaltung
  - Key Exchange
  - Authentifizierung
  - …
- Call Control
  - Abweichend zu DECT (vereinfacht)
  - Verbindungen aufbauen und abbauen

Authentifizierung I



Authentifizierung II



Verschlüsselung

- Derived Cipher Key *DCK* als Ergebnis der Authentifizierung des PP

- MAC Verschlüsselung: Stream Cipher
  - **DECT Standard Cipher**, 64 bit Schlüssellänge
  - **DECT Standard Cipher #2**, 128 bit Schlüssellänge

  $\Rightarrow$ keine Authentifizierung von Nachrichten (*privacy mechanism*)

- DLC Verschlüsselung
  - AES-128 im CCM Modus, Authenticated Encryption

Verschlüsselung

- Derived Cipher Key *DCK* als Ergebnis der Authentifizierung des PP

- MAC Verschlüsselung: Stream Cipher
  - **DECT Standard Cipher**, 64 bit Schlüssellänge
  - **DECT Standard Cipher #2**, 128 bit Schlüssellänge

  $\Rightarrow$ keine Authentifizierung von Nachrichten (*privacy mechanism*)

- DLC Verschlüsselung
  - AES-128 im CCM Modus, Authenticated Encryption

Bedrohungsmodell

- Betrachtungsgegenstand: Funkschnittstelle
- kein physischer Zugriff auf ULE Geräte im Netzwerk
- Eingriffe in den Funkverkehr
  - Verändern
  - Löschen
  - Einfügen
- Geräte im Netzwerk vertrauen untereinander
- Geräte im Netzwerk haben alle Rechte

Vorgehensweise

Für jedes Protokollszenario:

1. Protokollszenario beschreiben
2. STRIDE-per-interaction[1]
   - Spoofing
   - Tampering
   - Repudiation
   - Information Disclosure
   - Denial of Service
   - Elevation of Privileges
3. Schutzmechanismen identifizieren und beurteilen

[1] Adam Shostack. Threat Modelling – Designing for Security. Wiley, 2014.

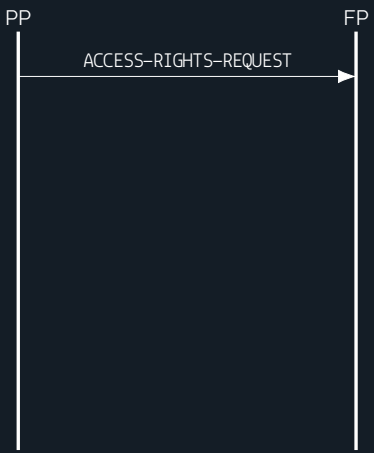
Vorgehensweise

Für jedes Protokollszenario:

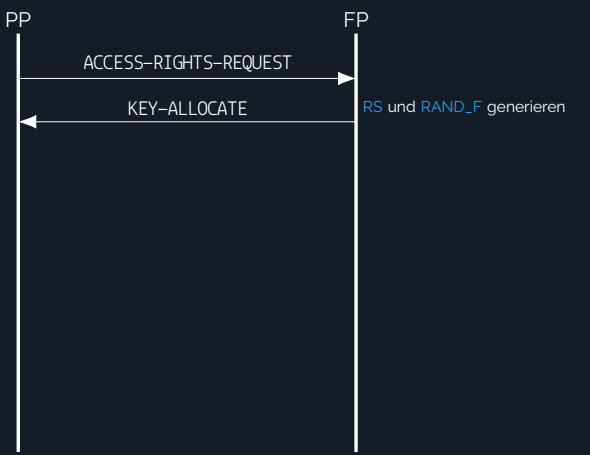
1. Protokollszenario beschreiben
2. STRIDE-per-interaction[1]
   - **Spoofing**
   - **Tampering**
   - Repudiation
   - **Information Disclosure**
   - **Denial of Service**
   - Elevation of Privileges
3. Schutzmechanismen identifizieren und beurteilen

[1] Adam Shostack. Threat Modelling – Designing for Security. Wiley, 2014.

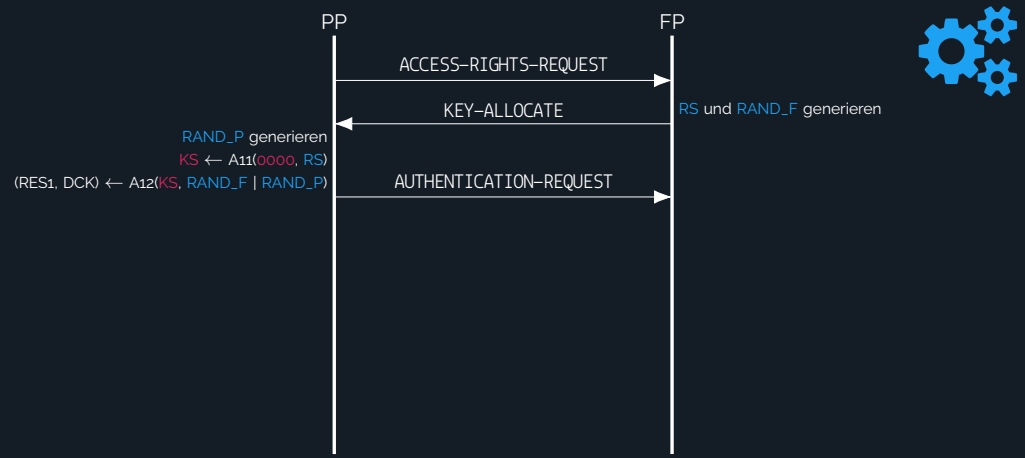
Beispiel: Easy Pairing – Protokollszenario beschreiben



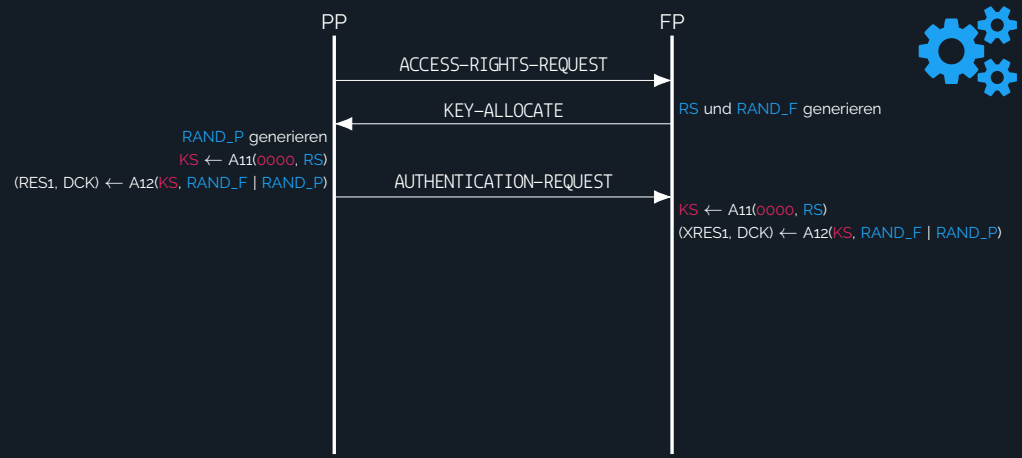
Beispiel: Easy Pairing – Protokollszenario beschreiben



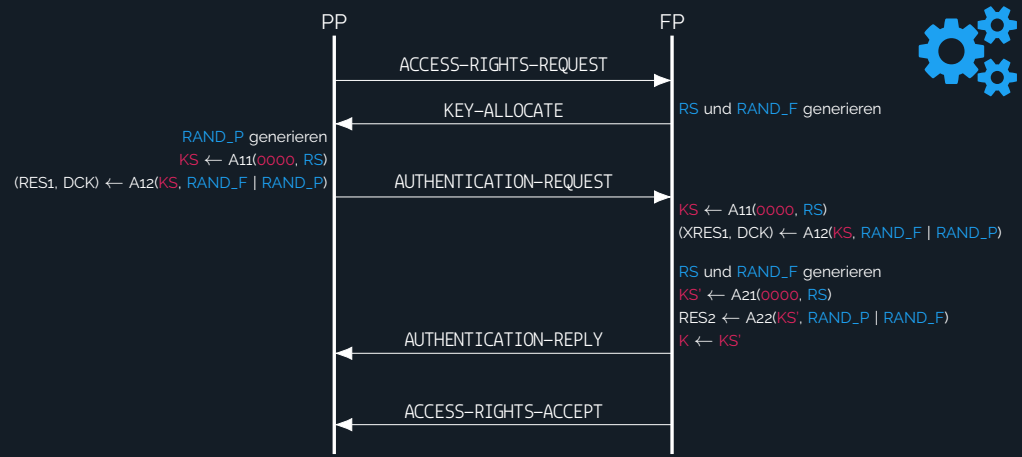
Beispiel: Easy Pairing – Protokollszenario beschreiben



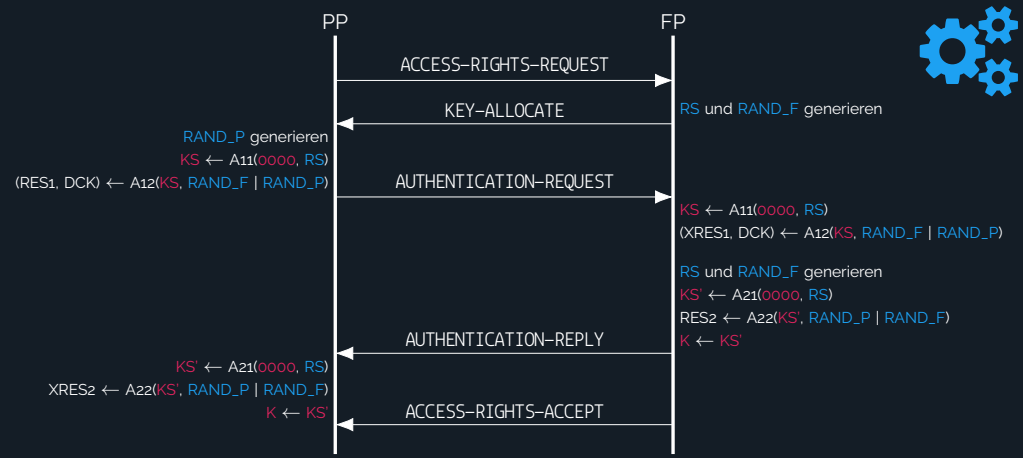
Beispiel: Easy Pairing – Protokollszenario beschreiben



Beispiel: Easy Pairing – Protokollszenario beschreiben



Beispiel: Easy Pairing – Protokollszenario beschreiben



Beispiel: Easy Pairing – STRIDE-per-interaction

Sichtweise:	FP
Protokollszenario:	Easy Pairing

Spoofing:

Tampering:

Information Disclosure:

Denial of Service:

Beispiel: Easy Pairing – STRIDE-per-interaction

Sichtweise:	FP
Protokollszenario:	Easy Pairing
Spoofing:	• Angreifer tritt dem Netzwerk bei
Tampering:	
Information Disclosure:	
Denial of Service:	

Beispiel: Easy Pairing – STRIDE-per-interaction

Sichtweise:	FP
Protokollszenario:	Easy Pairing
Spoofing:	• Angreifer tritt dem Netzwerk bei • Man-in-the-Middle
Tampering:	
Information Disclosure:	
Denial of Service:	

Beispiel: Easy Pairing – STRIDE-per-interaction

Sichtweise:	FP
Protokollszenario:	Easy Pairing
Spoofing:	• Angreifer tritt dem Netzwerk bei • Man-in-the-Middle
Tampering:	—
Information Disclosure:	
Denial of Service:	

Beispiel: Easy Pairing – STRIDE-per-interaction

Sichtweise:	FP
Protokollszenario:	Easy Pairing
Spoofing:	• Angreifer tritt dem Netzwerk bei • Man-in-the-Middle
Tampering:	—
Information Disclosure:	• Angreifer berechnet den geheimen abgeleiteten K
Denial of Service:	

Beispiel: Easy Pairing – STRIDE-per-interaction

Sichtweise:	FP
Protokollszenario:	Easy Pairing
Spoofing:	• Angreifer tritt dem Netzwerk bei • Man-in-the-Middle
Tampering:	—
Information Disclosure:	• Angreifer berechnet den geheimen abgeleiteten K
Denial of Service:	• Angreifer manipuliert Werte (RANDF, RANDP, *RS*, RES1, RES2, …)

Beispiel: Easy Pairing – STRIDE-per-interaction

Sichtweise: **Protokollszenario:**	FP Easy Pairing
Spoofing:	• Angreifer tritt dem Netzwerk bei • Man-in-the-Middle
Tampering:	—
Information Disclosure:	• Angreifer berechnet den geheimen abgeleiteten K
Denial of Service:	• Angreifer manipuliert Werte (RANDF, RANDP, *RS*, RES1, RES2, …) • Jamming

Beispiel: Easy Pairing – Schutzmechanismen identifizieren und beurteilen

Angreifer tritt dem Netzwerk bei

Man-in-the-Middle

Angreifer berechnet K

Angreifer manipuliert Werte

Jamming

Beispiel: Easy Pairing – Schutzmechanismen identifizieren und beurteilen

Angreifer tritt dem Netzwerk bei Keine Authentifizierung

Man-in-the-Middle

Angreifer berechnet K

Angreifer manipuliert Werte

Jamming

Beispiel: Easy Pairing – Schutzmechanismen identifizieren und beurteilen

Angreifer tritt dem Netzwerk bei	Keine Authentifizierung	✖
Man-in-the-Middle	Keine Authentifizierung	✖
Angreifer berechnet K		
Angreifer manipuliert Werte		
Jamming		

Beispiel: Easy Pairing – Schutzmechanismen identifizieren und beurteilen

Angreifer tritt dem Netzwerk bei	Keine Authentifizierung	✖
Man-in-the-Middle	Keine Authentifizierung	✖
Angreifer berechnet K	Kein Geheimnis zur Verschlüsselung verwendet	✖
Angreifer manipuliert Werte		
Jamming		

Beispiel: Easy Pairing – Schutzmechanismen identifizieren und beurteilen

Angreifer tritt dem Netzwerk bei	Keine Authentifizierung	✗
Man-in-the-Middle	Keine Authentifizierung	✗
Angreifer berechnet K	Kein Geheimnis zur Verschlüsselung verwendet	✗
Angreifer manipuliert Werte	Easy Pairing scheitert: berechnete RES1, RES2 oder K stimmen nicht überein	✓
Jamming		

Beispiel: Easy Pairing – Schutzmechanismen identifizieren und beurteilen

Angreifer tritt dem Netzwerk bei	Keine Authentifizierung	✗
Man-in-the-Middle	Keine Authentifizierung	✗
Angreifer berechnet K	Kein Geheimnis zur Verschlüsselung verwendet	✗
Angreifer manipuliert Werte	Easy Pairing scheitert: berechnete RES1, RES2 oder K stimmen nicht überein	✓
Jamming	Easy Pairing scheitert: • Antworten bleiben aus • spätere Authentifizierung scheitert	✓

Ergebnisse

- Easy Pairing nutzt statischen Schlüssel
- DECT Key Allocation mit bis zu 32 bit → offline Brute Force durchführbar
- Downgrade auf DSAA
- MAC Verschlüsselung nicht authentifiziert
- Jamming kann nicht verhindert werden

Fragestellung

- Hürde für Ausnutzbarkeit ermitteln
- Hardwareanforderungen
  - keine Spezialhardware
  - Software Defined Radio
  - möglichst gering
- Alle DECT Channels gleichzeitig
- Freie Software
- Fehlende Softwareteile sollen implementiert werden

Umsetzung

- Software Defined Radio
  - GNURadio
  - gr-dect2
  - ReDECTed
  - gr-dectule
- Parsing und Analyse
  - Wireshark
  - rule

Umsetzung – gr-dectule

1. DECT Frequenzbereich aufnehmen
   - Speicherung der Rohdaten auf Festplatte
   - DECT Frequenzbereich: 1880 MHz - 1900 MHz

   ⇒ Frequenz: 1890 MHz, Bandbreite 20 MHz

Umsetzung – gr-dectule

1. DECT Frequenzbereich aufnehmen
   - Speicherung der Rohdaten auf Festplatte
   - DECT Frequenzbereich: 1880 MHz - 1900 MHz

   ⇒ Frequenz: 1890 MHz, Bandbreite 20 MHz
2. Rohdaten offline verarbeiten
   - Channels dekodieren
   - Signale verschieben (*mischen*) und filtern
   - Dekodierung der Pakete durch Logik von gr-dect2

Umsetzung – gr-dectule

1. DECT Frequenzbereich aufnehmen
   - Speicherung der Rohdaten auf Festplatte
   - DECT Frequenzbereich: 1880 MHz - 1900 MHz

   ⇒ Frequenz: 1890 MHz, Bandbreite 20 MHz
2. Rohdaten offline verarbeiten
   - Channels dekodieren
   - Signale verschieben (*mischen*) und filtern
   - Dekodierung der Pakete durch Logik von gr-dect2
3. Speicherung der dekodierten Pakete in pcap-File
   - Analog zu ReDECTed
   - Versand der dekodierten Pakete an dummy-Interface
   - Aufnahme des Traffics auf dummy-Interface durch Wireshark

Umsetzung – rule

- Verarbeitungsgeschwindigkeit für zehn Channels: Rust
- Verarbeitung von pcap-Files
- Berechnung der Framelänge, um Bearer zuzuordnen
- interpretierte Filtersprache
- Status
  - Physical Layer
  - Medium Access Layer
  - Data Link Control Layer
  - Network Layer

Bewertung

- Lesen von Nachrichten
- Injizieren von Nachrichten
- Jamming von Nachrichten
- Manipulation von Nachrichten

Bewertung

- Lesen von Nachrichten
- Injizieren von Nachrichten
- Jamming von Nachrichten
- Manipulation von Nachrichten

Bewertung

- Lesen von Nachrichten
- Injizieren von Nachrichten
- Jamming von Nachrichten
- Manipulation von Nachrichten

- Signalverarbeitung verbesserungswürdig
- DECT umfangreich ⇒ hoher Implementationsaufwand

Fazit

- Pairing unsicher
  - ⇒ Abhören des Funkverkehrs problemlos möglich
  - ⇒ Statischer Schlüssel trivial zu brechen
  - ⇒ Brute-Force für 32 bit Schlüssel offline durchführbar
- Unauthentifizierte Verschlüsselung auf MAC Layer
- Downgrading möglich
  - ⇒ Manipulation von Nachrichten komplex
- Jamming möglich
  - ⇒ Timing benötigt verbesserte Signalverarbeitung

Ausblick

1. Kein Easy Pairing verwenden
2. Key Allocation nur mit starken Schlüsseln durchführen
3. DSAA2 sollte nicht mehr implementiert werden
4. DLC Verschlüsselung vs. MAC Verschlüsselung

Fragen