How To Crash A Train (Theoretically)

Eine Sicherheitslücke in ETCS und was sie mit Schweizer Käse zu tun hat

laugengebaeck

20. Juni 2025

ETCS auf der Riedbahn

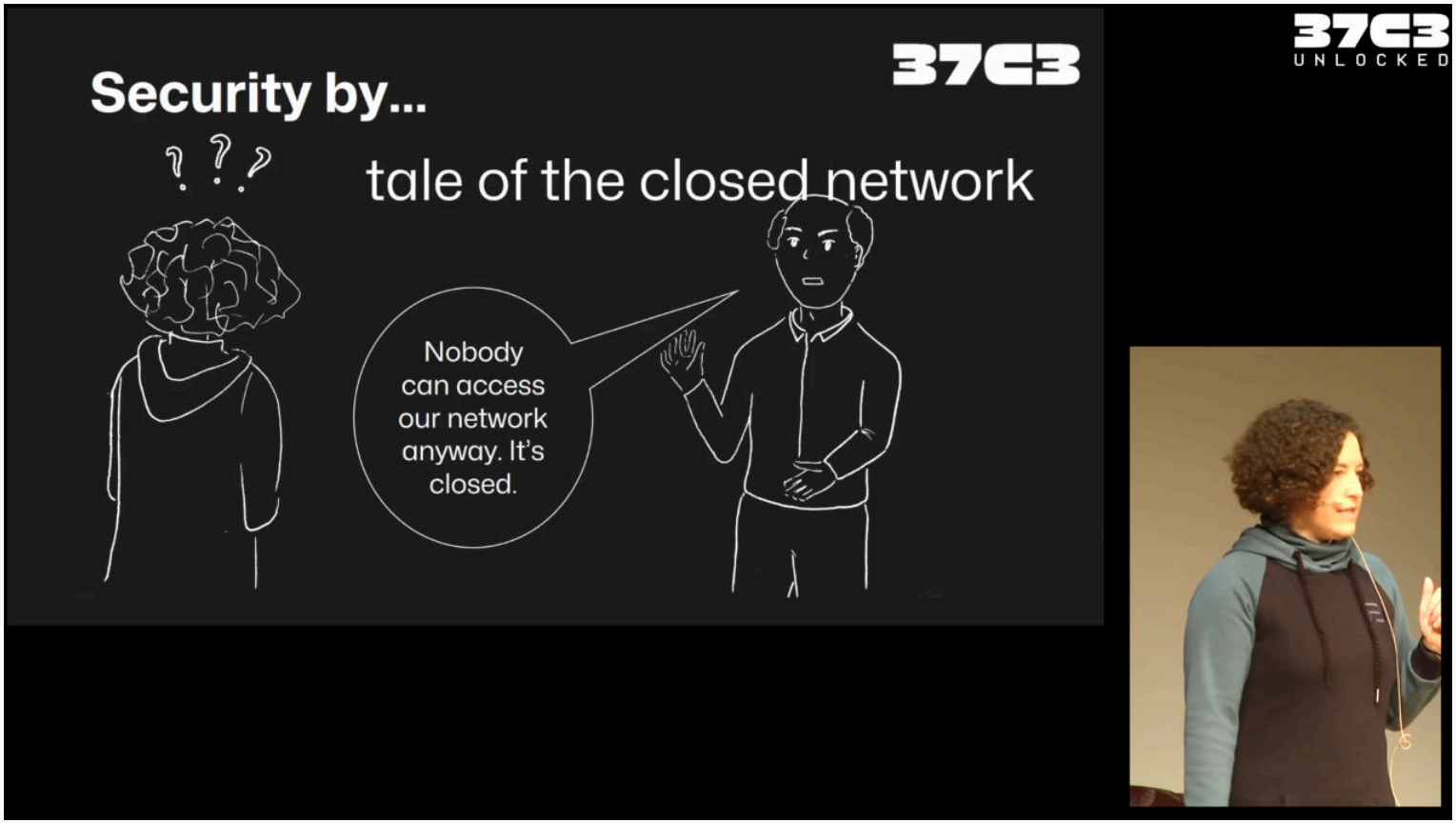
DB AG/Benjamin Kedziora

Vorstellung

laugengebaeck

- auch bekannt als Laura Rost
- Informatikstudium in Potsdam
- parallel Softwareentwicklerin im Kontext *Digitale Stellwerke*
- Forschungsseminar zu Sicherheitslücken in ETCS (mit Katja Assaf, Jörn Sobotta, Ruben Zangerle)

Was bisher geschah...



Why Railway Is Safe But Not Secure (37C3)

Average Familiarity

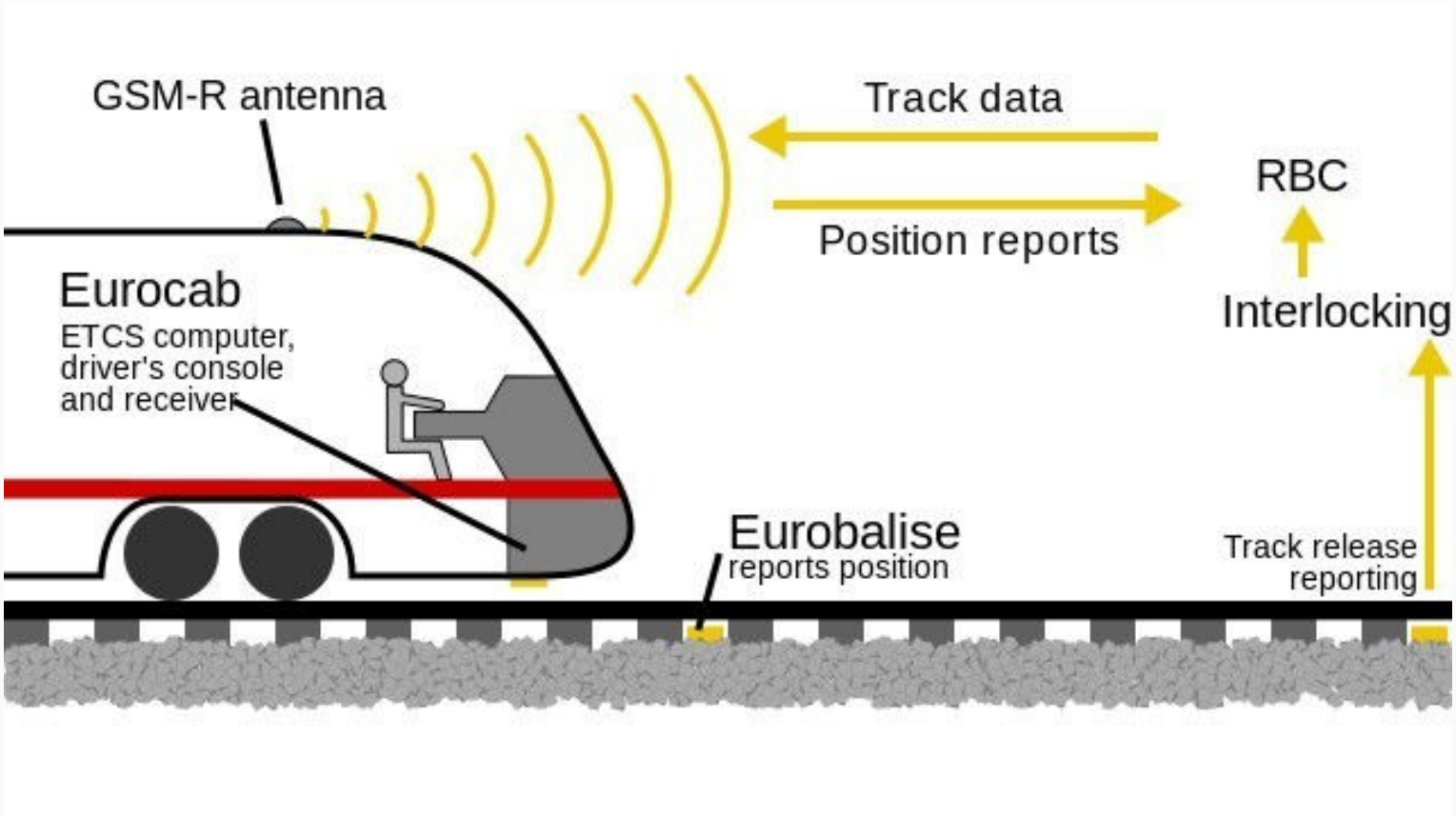
basierend auf https://xkcd.com/2501/

Was ist eigentlich dieses ETCS?

European Train Control System (ETCS)

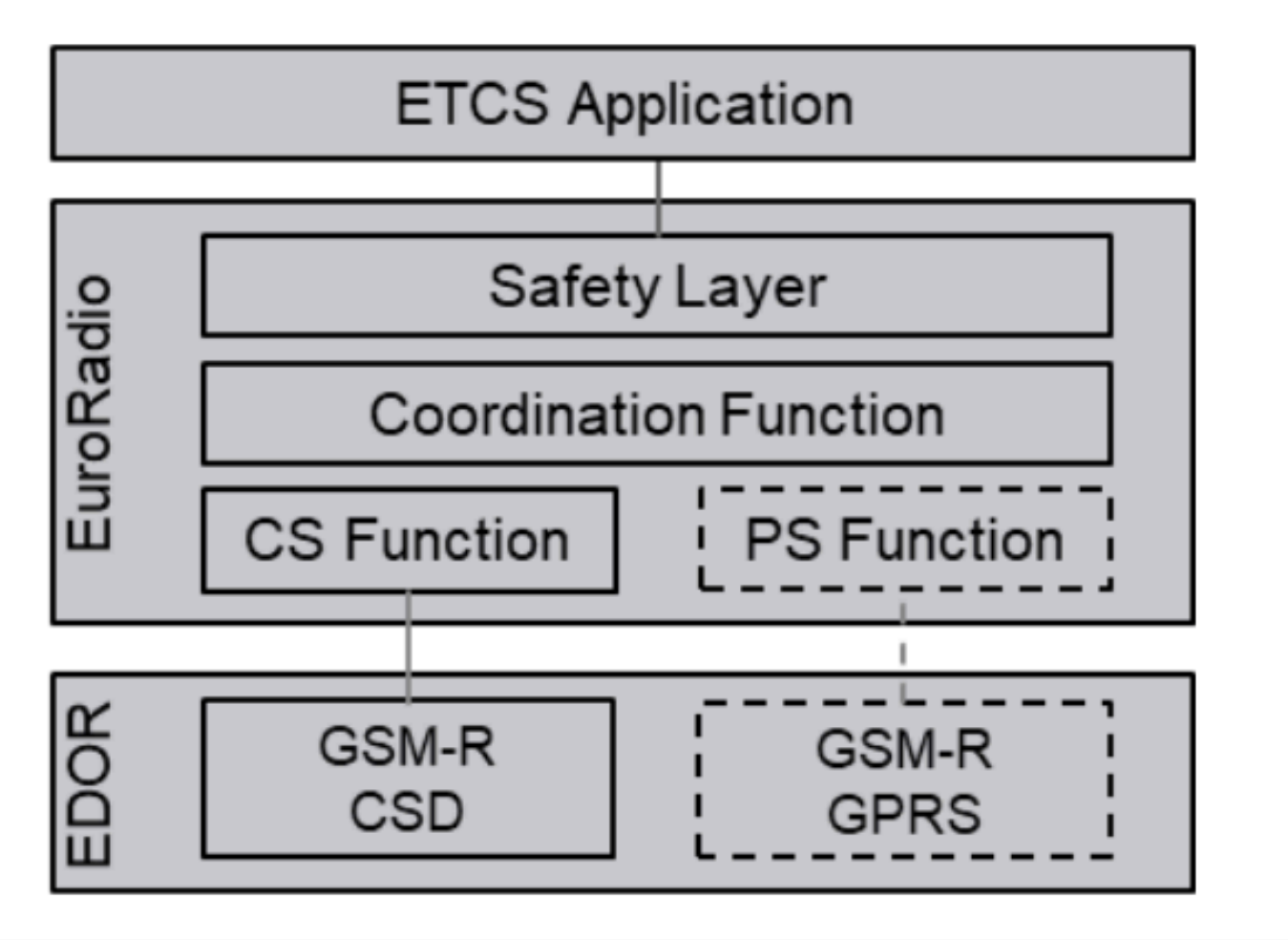
- zusammen mit dem Mobilfunkstandard GSM-R Teil des *European Rail Traffic Management System* (ERTMS)
- Ziel: europaweite Interoperabilität im Eisenbahnverkehr durch Vereinheitlichung von Systemen für Zugsicherung und Signalgebung

ETCS Level 2



European Union Agency for Railways

ERTMS-Protokollstack



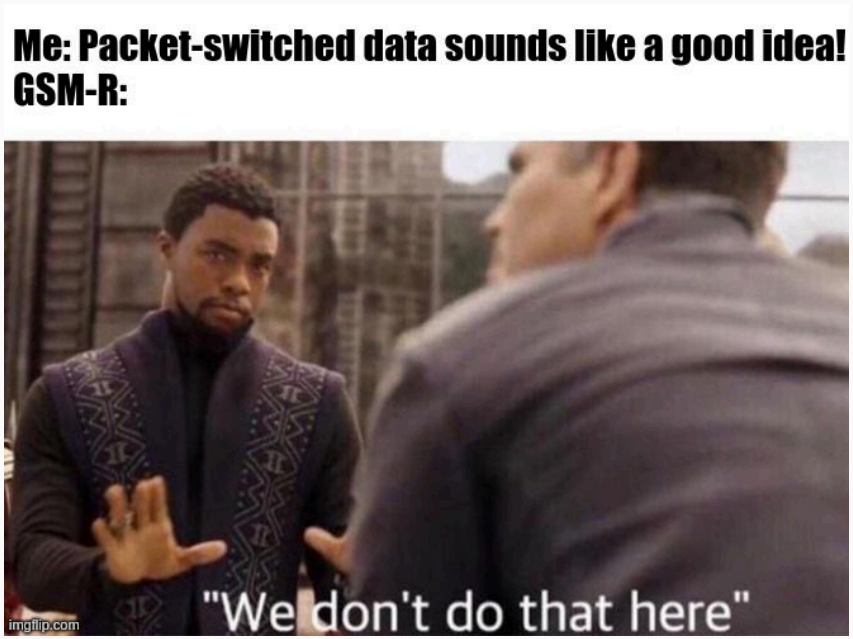
Internationaler Eisenbahnverband (UIC)

GSM-R

- Mobilfunkstandard GSM (= 2G), aber für die Eisenbahn
- auf über 130.000 km Strecke weltweit installiert
- zusätzliche eisenbahnspezifische Funktionalität:
  - Support für Broadcast- und Gruppenanrufe (z. B. für Nothaltaufträge)
  - funktionale und standortunabhängige Adressierung (z. B. Fahrdienstleiter mit der]2[anrufen)
  - Support für ETCS-Datenübertragung

S. Terfloth, CC BY-SA 2.0 DE, via Wikimedia Commons

Paketvermittelt? Brauchen wir nicht!



- ursprünglich nur leitungsvermittelte Kommunikation in GSM-R spezifiziert
  - **Problem**: sehr ineffizient, Kapazitätsprobleme in größeren Bahnhöfen
- seit 2016 auch paketvermittelte Kommunikation (GPRS) erlaubt
  - **neues Problem**: Best Effort ist nicht gut genug
  - wird deshalb nicht tatsächlich eingesetzt (inzwischen aber in Planung)

GSM-R Security

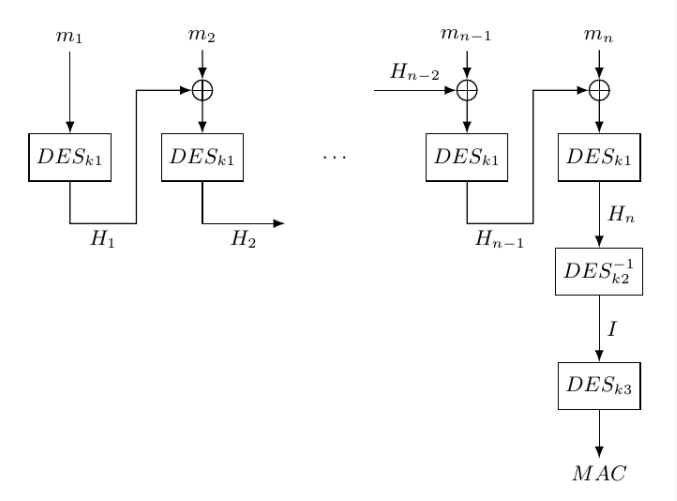
Verschlüsselung mittels symmetrischer Verfahren aus der A5-Ciphersuite:

- **A5/1** (1987)
  - kann inzwischen mithilfe von Rainbow Tables in Echtzeit gebrochen werden
  - absichtlich so schwach auf Wunsch von NATO-Nachrichtendiensten
- **A5/2** (1989)
  - wegen Exportbeschränkungen noch schlechter
- **A5/3** (2000)
  - nutzt von Mitsubishi entwickelten Blockchiffre mit 128 Bit Schlüssellänge
  - leider sind das aber zwei Kopien des gleichen 64-Bit-Keys
  - aufgrund geringer Schlüssellänge anfällig für Brute Force

Authentifizierung: nur Client beim Netzwerk, aber nicht Basisstation beim Client

EuroRadio

- zwei Bestandteile:
  - *Communication Functional Module*: Verbindungsmanagement, ”Transport Layer” von ETCS
  - *Safety Functional Module*: Authentifizierung von über offenes Netzwerk verschickten Nachrichten
- Authentifizierung mittels MAC-Verfahren, welches auf Session Keys basiert
  - verbesserte Variante von ISO 9797-1 Algorithm 3 (Cipher Block Chaining mit DES-Algorithmus)
  - **Problem**: geringe Blockgröße (64 Bit), dadurch hohe™ Kollisionswahrscheinlichkeit



Chothia et al.: ”An Attack...”

ETCS Application Layer

- Spezifikation wird von ERA öffentlich zur Verfügung gestellt
- verschiedene *Messages* mit jeweils einheitlichem Header für Zug $\rightarrow$ RBC bzw. RBC $\rightarrow$ Zug
  - z. B. *Movement Authority*, *Position Report*, *Emergency Stop*
- können *Packets* mit zusätzlichen Informationen enthalten
  - teilweise optional, teilweise verpflichtend

7.4.2.4 Packet Number 15: Level 2 Movement Authority

Description	Transmission of a movement authority for level 2.		
Transmitted by	RBC		
Content	**Variable**	**Length**	**Comment**
	NID_PACKET	8	
	Q_DIR	2	
	L_PACKET	13	
	Q_SCALE	2	
	V_EMA	7	
	T_EMA	10	Can be set to “no time-out”
	N_ITER	5	Set to zero if only one section in the MA
	L_SECTION(k)	15	
	Q_SECTIONTIMER(k)	1	
	T_SECTIONTIMER(k)	10	

ERTMS System Requirements Specification

Und wie kann man damit jetzt einen Zug angreifen?

"'An Attack Against Message Authentication…"

An Attack Against Message Authentication in the ERTMS Train to Trackside Communication Protocols

Tom Chothia
University of Birmingham
Birmingham, UK
t.p.chothia@cs.bham.ac.uk

Mihai Ordean
University of Birmingham
Birmingham, UK
m.ordean@cs.bham.ac.uk

Joeri de Ruiter
Radboud University
Nijmegen, NL
joeri@cs.ru.nl

Richard J. Thomas
University of Birmingham
Birmingham, UK
r.j.thomas@cs.bham.ac.uk

ABSTRACT

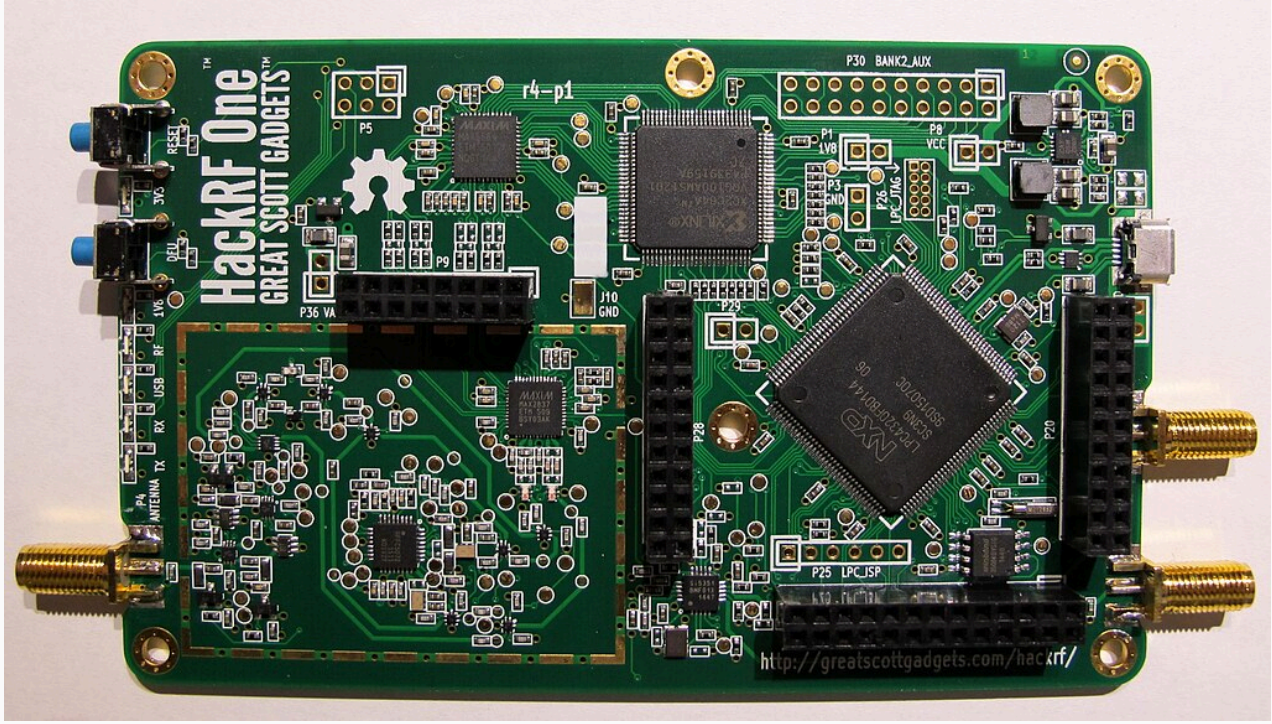
This paper presents the results of a cryptographic analysis of the protocols used by the European Rail Traffic Management System (ERTMS). A stack of three protocols secures the communication between trains and trackside equipment; encrypted radio communication is provided by the GSM-R protocol, on top of this the EuroRadio protocol provides authentication for a train control application-level protocol. We present an attack which exploits weaknesses in all three protocols: GSM-R has the same well known weaknesses as the GSM protocol, and we present a new collision attack against the EuroRadio protocol. Combined with design weaknesses in the application-level protocol, these vulnerabilities allow an attacker, who observes a MAC collision, to forge train control messages. We demonstrate this attack

railways. At present, the system is being rolled out across Europe and also on high-speed lines around the world.

ERTMS is formed of three core communication layers: GSM-R, EuroRadio and the Application Layer protocol (see Figure 1). The EuroRadio and the Application Layer protocols form ETCS, the European Train Control System. The lowest layer of the stack, GSM-R, is a rail-specific variant of the GSM protocol, used for communications between the train and trackside infrastructure. EuroRadio, the middle layer, provides authentication and integrity of messages sent between the train and track side components using cryptographic MACs. The Application Layer protocol is the highest layer of the stack; this is a stateful protocol that includes timestamps and message acknowledgements to prevent the replay of messages and ensure successful communication.

Abhören von GSM(-R)-Kommunikation

- *Software Defined Radio* = programmierbarer Funkempfänger und -sender
- GNU Radio und `gr-gsm` zur Signalverarbeitung und Ansteuerung



HackRF One (wdwd, CC BY-SA 4.0, via Wikimedia Commons)

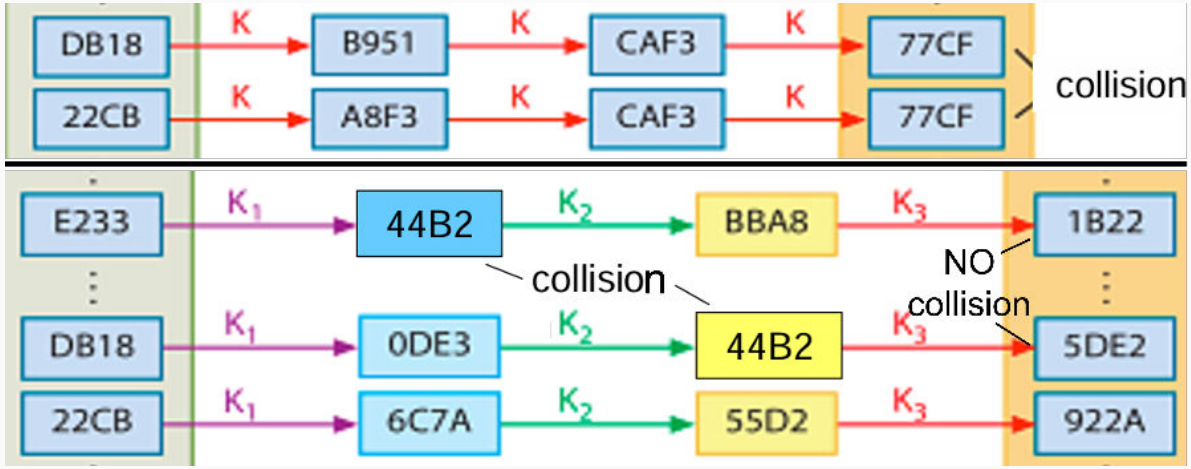
Sie begehen eine Straftat!

Sie begehen eine Straftat!

- verboten nach § 148 TKG
  - Unbefugtes Abhören von Nachrichten
  - Weitergabe des Inhalts von Funkgesprächen
- Hackerparagraph § 202a StGB
- also:
  - Erlaubnis des Senders einholen *oder*
  - Testdaten verwenden

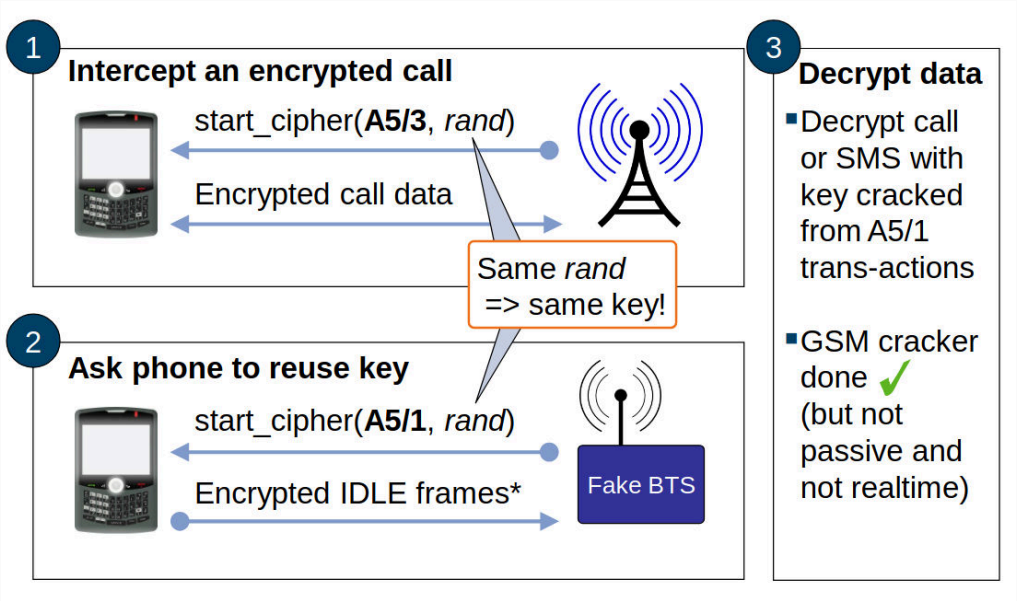
Entschlüsselung von GSM(-R)-Kommunikation

- Rainbow Tables
  - Mapping von verschlüsseltem Output auf Key ermöglichen
  - Aufbau: $V_1 \xrightarrow{A5/1} O_1 \xrightarrow{Red.} V_2 \xrightarrow{A5/1} O_2 \xrightarrow{Red.} \ldots \xrightarrow{Red.} V_n$
- Optimierungen:
  - *Distinguished Points:* nur Kettenglieder, die ein bestimmtes Kriterium erfüllen, speichern
  - *Unterschiedliche Reduktionsfunktionen:* Kollisionen verschiedener Ketten vermeiden



Entschlüsselung von GSM(-R)-Kommunikation

- Softwaresammlung *Kraken* automatisiert alle nötigen Schritte
  - 40 Rainbow Tables à 40 GB (durch Community berechnet)
  - 30 bis 60 Sekunden pro Key
- A5/3 ist verwundbar durch Downgrade-Angriffe

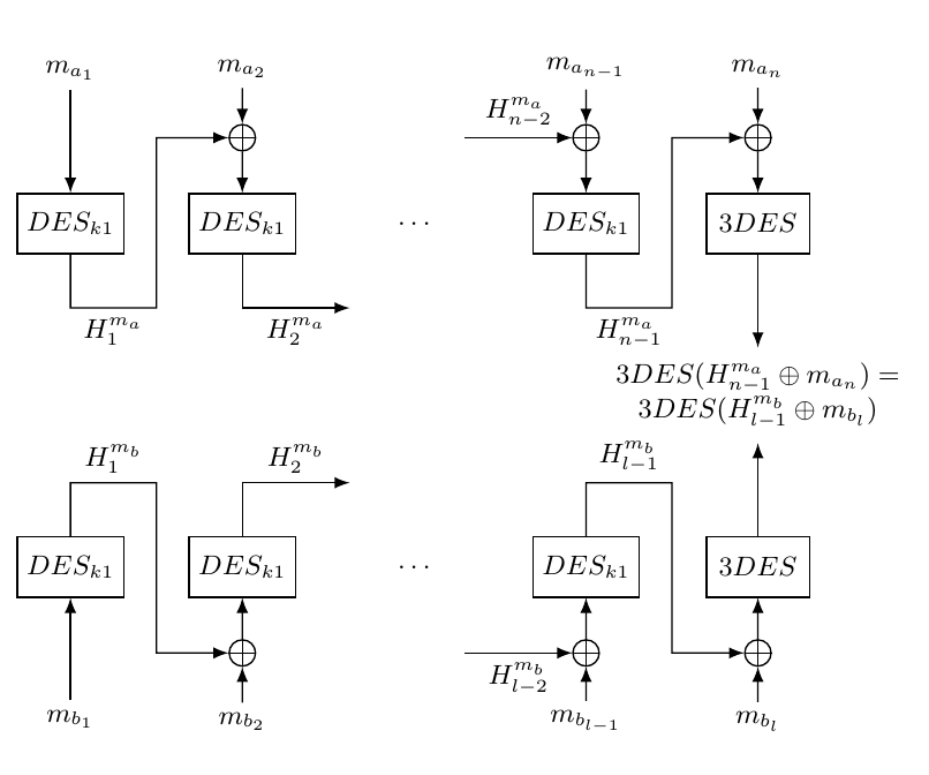


Ganz so einfach ist es nicht...

- Angriff benötigt einzelne Pakete auch als Plaintext
  - bei GSM: System Information Type 5 & 6
  - bei GSM-R: werden laut Spezifikation nicht verwendet
- Gibt es alternative Pakete?
- Sind andere Angriffe erfolgversprechender?

Angriff auf DES in EuroRadio

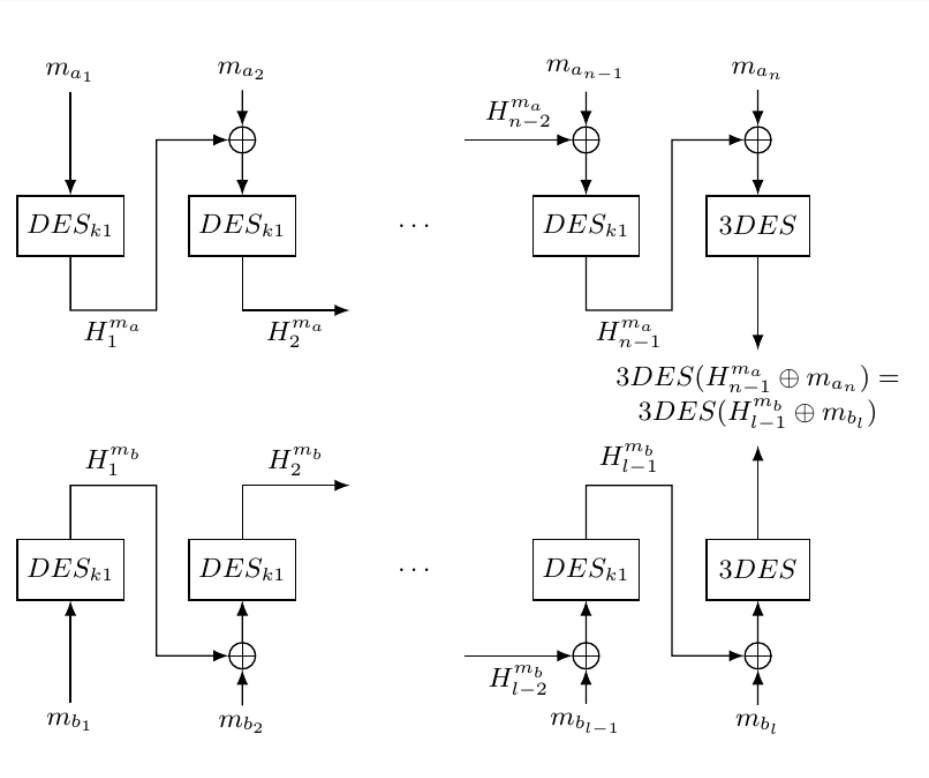
- Wir haben eine MAC-Kollision gefunden
  - … also muss die Eingabe von 3DES identisch sein



Chothia et al.: ”An Attack…”

Angriff auf DES in EuroRadio

- Alle anderen DES-Runden benutzen nur k_1
  - Bruteforcen!
  - 56 Bit heutzutage in annehmbarer Zeit möglich



Chothia et al.: "An Attack..."

Aktueller Zustand

juicy_fish on Flaticon

juicy_fish on Flaticon

juicy_fish on Flaticon

Hilft uns das?

ETCS-Nachrichten fälschen

- Wir wissen, wie wir den gleichen MAC wie bei einer vorgegebenen Nachricht produzieren
  - Eingabe von 3DES muss gleich sein!
  - nicht bei jeder Nachricht so einfach möglich
- ETCS erlaubt es, 255 Zeichen lange Textnachrichten für den Triebfahrzeugführer als optionales Datenpaket zu senden
  - können beliebige Blöcke zur Nachricht hinzufügen, um den gewünschten MAC zu produzieren
  - Anzeigebedingungen so gewählt, dass unsichtbar für den Triebfahrzeugführer
- Wir können beliebige ETCS-Nachrichten fälschen!

Aber wie realistisch ist so ein Angriff?

Mögliche Auswirkungen des Angriffs

- sicherheitsrelevant:
  - Entgleisungen (durch überhöhte Geschwindigkeit, nicht gesicherten Fahrweg, ...)
  - Kollisionen (durch noch besetzten Fahrweg)
- ”nur” nervig: Emergency Stops
- RBC und Triebfahrzeugführer als Kontrollinstanzen
  - ggf. auch PZB-Doppelausrüstung

Bundespolizeidirektion Berlin

Erfolgswahrscheinlichkeit

Variante 1: Einzelner Zug

- für 1 % Kollisionswahrscheinlichkeit: 600 Millionen Nachrichten ($\approx 19,5\ GB$)
- Übertragung benötigt über GSM-R 23 Tage (bei 10 Kbps)
- Session Keys werden aber nach RBC-Wechsel oder Neustart des Zuges neu ausgehandelt

Erfolgswahrscheinlichkeit

Variante 2: Gesamtes Eisenbahnnetz

- momentan: ca. 500 Sessions am Tag (vor allem auf SFS)
  - 1 % Erfolgswahrscheinlichkeit nach 375 Tagen
- Ziel: ETCS Level 2 auf allen Eisenbahnstrecken in Deutschland
  - Ausbau kommt bekanntlich sehr gut voran
  - $\geq$ 50000 Zugfahrten pro Tag
  - 1 % Erfolgswahrscheinlichkeit nach maximal 4 Tagen
- quasi nur für Inside Attacker möglich

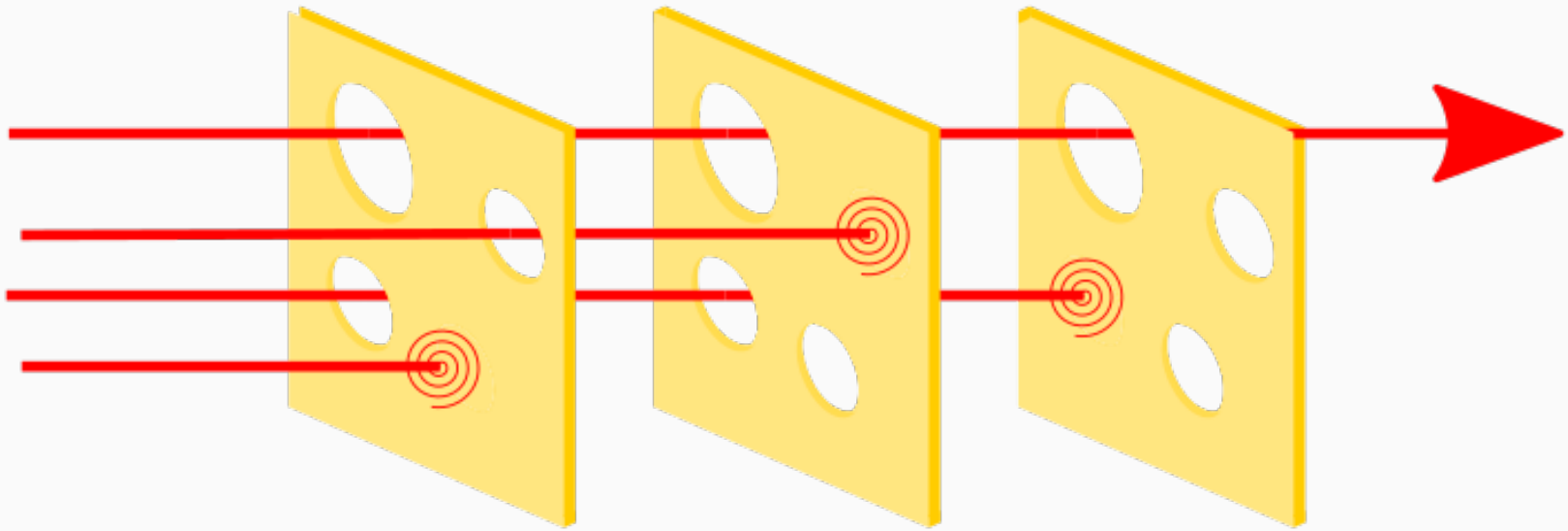
DB InfraGO AG / Infrastrukturregister

Was kann man dagegen tun?

Gegenmaßnahmen

1. **GSM-R:** Stärkere Verschlüsselungsverfahren
   - sind Teil von FRMCS als Nachfolger von GSM-R
2. **EuroRadio:** Wahrscheinlichkeit für MAC-Kollisionen begrenzen
   - Session Key regelmäßig neu aushandeln (z. B. nach 4500 Nachrichten für $p < 10^{-6}$)
   - nicht in Applikationen mit hoher Datenrate verwenden
3. **EuroRadio**: Kollisionsresistentere MAC-Algorithmen verwenden
   - z. B. AES- oder HMAC-basiert
   - Deployment sehr aufwändig
4. nicht in Panik und Aktionismus verfallen, sondern Plan zur Verbesserung der Sicherheit entwickeln

Learnings



BenAveling, CC BY-SA 4.0, via Wikimedia Commons

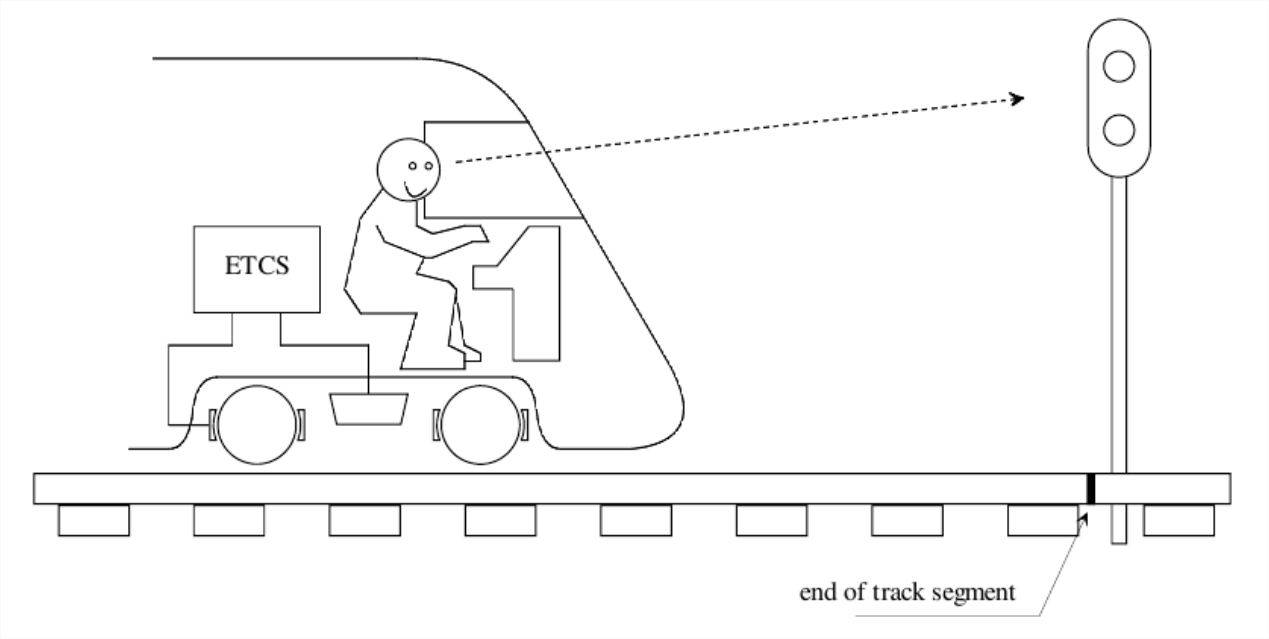
Weiterführende Talks

- *Why Railway Is Safe But Not Secure* (Katja Assaf, 37C3): `https://media.ccc.de/v/37c3-11717-why_railway_is_safe_but_not_secure`
- *Can trains be hacked?* (Stefan Katzenbeisser, 28C3): `https://media.ccc.de/v/28c3-4799-de-can_trains_be_hacked`
- *GSM: SRSLY?* (Karsten Nohl & Chris Paget, 26C3): `https://media.ccc.de/v/26c3-3654-en-gsm_srsly`
- *Wie die Eisenbahn aus Fehlern lernt und dabei IT-Sicherheit erhält* (Oliver Knapp, FrOSCon 2022): `https://media.ccc.de/v/froscon2022-2758-wie_die_eisenbahn_aus_fehlern_lernt_und_dabei_it-sicherheit_erhalt`

Fragen

Für Fragen und Anregungen erreicht ihr mich auch nach der GPN

- im Fediverse unter `@laugengebaeck@zug.network`
- sowie per E-Mail unter `gpn [at] laurarost.de`.



ERTMS System Requirements Specification