



NEOCATENA NETWORKS INC.  
*>> Next Generation RFID Security >>*

## RFID Angriffe von Abhören bis Mini-Malware

Karsten Nohl, PhD  
University of Virginia

Lukas Grunwald  
Co-Founder and CTO



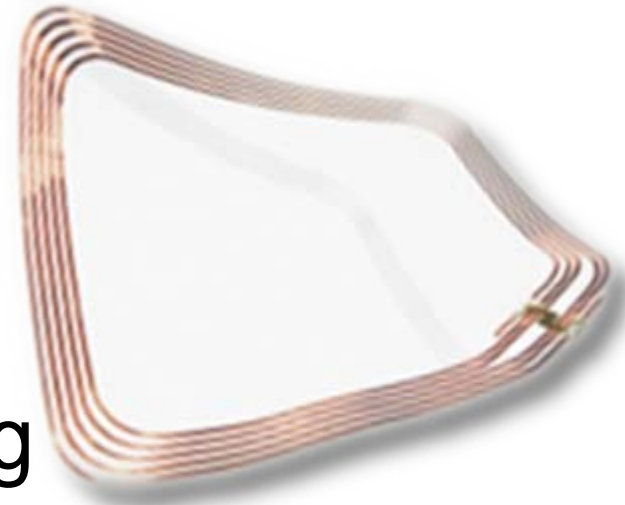
# Agenda

- 
- RFID Basics
  - Mögliche Angriffsvektoren
  - Angriffe auf „sichere“ RFIDs
  - RFID Malware
-



# RFID Tags & Etiketten

- Radio Frequency IDentification
- Winzige Computer Chips
- Keine eigene Energieversorgung





# Verschiedene Transponder Typen

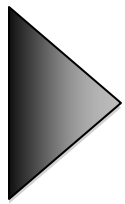
- Read-Only Tags
  - Verwendet zur Identifikation von Produkten (EPC), Autos (Toll collect) und Menschen (pass card)
- Wiederbeschreibbare Tags
  - Einfache Bezahlkarten und Zugangskontrollsysteme
- Contact-less Smart Cards
  - Mikrocontroller mit Radio Interface
  - Bieten starke Kryptographie für Payment und Access





# RFIDs vs. Privatsphäre

- Milliarden von RFIDs sind bereits im Einsatz
  - Zahl wird durch EPC “Barcodes” noch stark steigen
- Jedes einzelne sendet ständig einen elektronischen “Fingerabdruck”
- Schutz der Privatsphäre wurde bisher ignoriert
  - Zur Zeit einziger Schutz: Tags deaktivieren



RFIDs brauchen Schutzmechanismen in Form von starker Kryptographie.

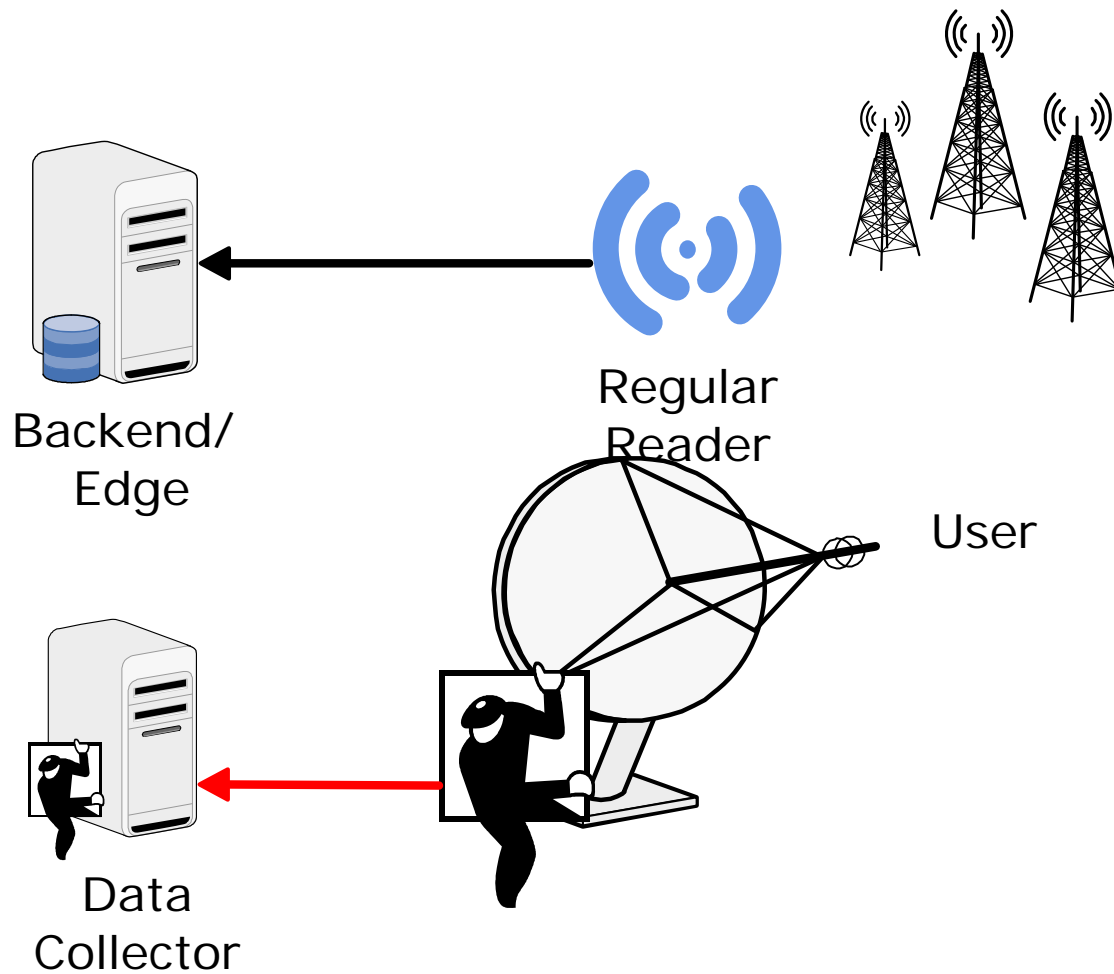


# Agenda

- 
- RFID Basics
  - Mögliche Angriffsvektoren
  - Angriffe auf „sichere“ RFIDs
  - RFID Malware
-

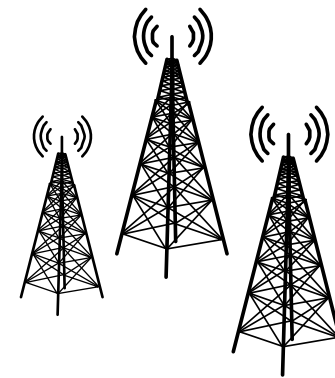
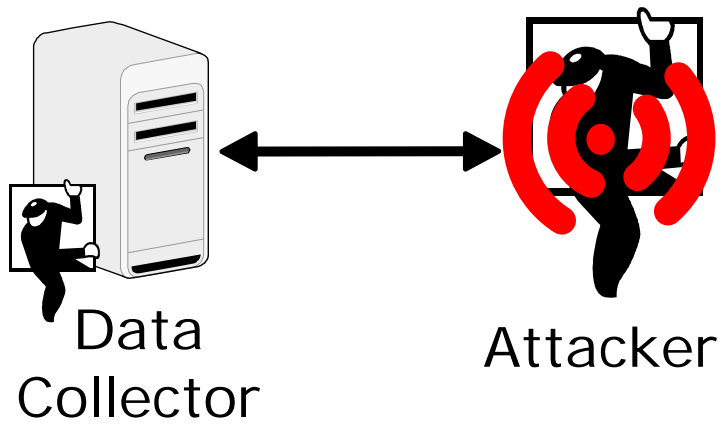


# Passives Scannen





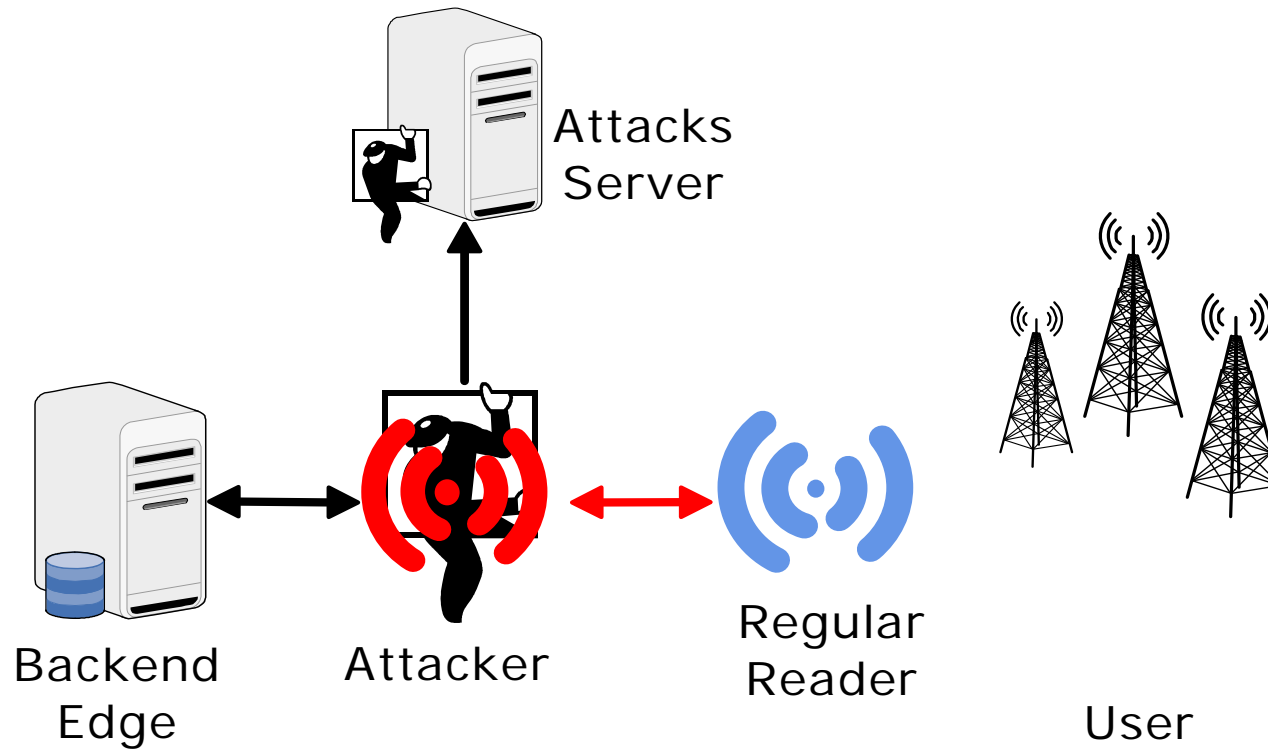
# Aktives Scannen



User



# Man in the Middle



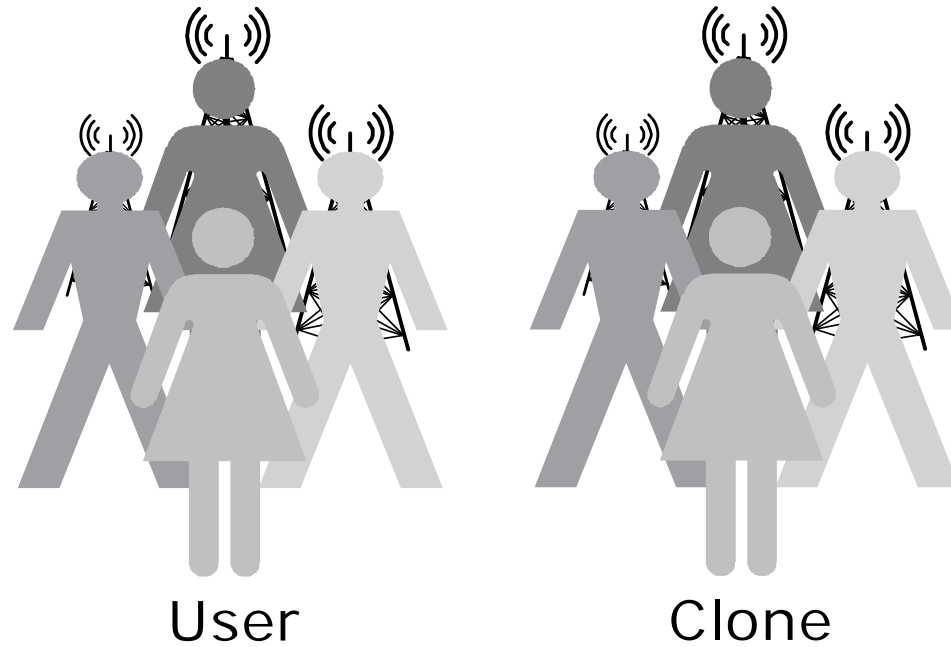


# Manipulation von Daten



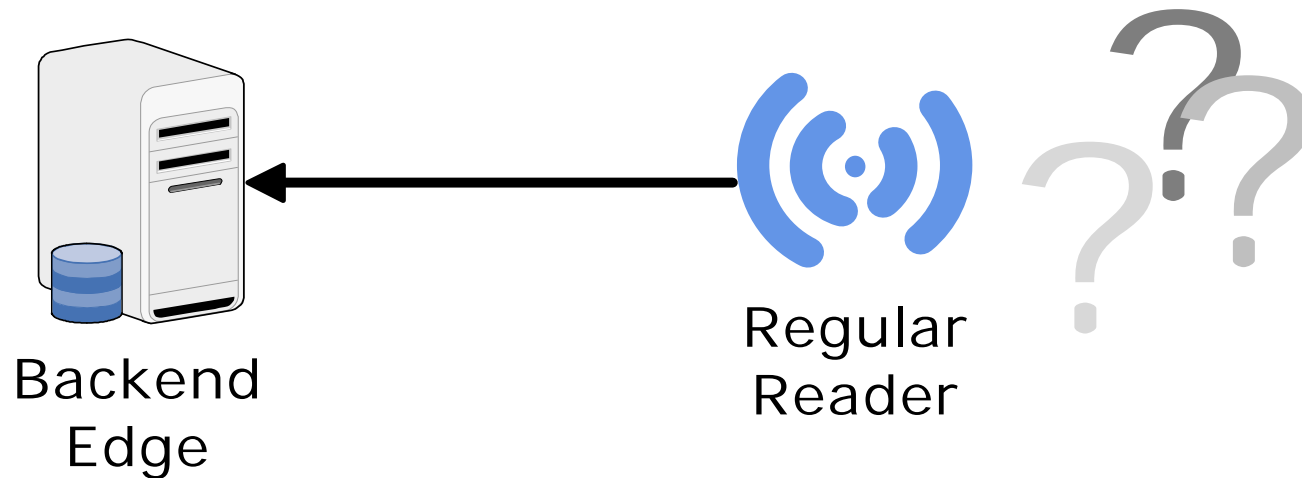


# Cloning





# Deaktivieren von Tags / DoS





# Agenda

- 
- RFID Basics
  - Mögliche Angriffsvektoren
  - Angriffe auf „sichere“ RFIDs
  - RFID Malware
-



# Es gibt starke Tags: Mifare SmartMX & DESFire

- Mikrocontroller mit Krypto-Beschleuniger
  - 3DES, AES, RSA, ECC, ...
- Kompatibel mit ISO 14443 Standard
- Deutschlandweit im Nahverkehr eingesetzt (VDV Kernapplikation)





# Und es gibt schwache Tags: Mifare Classic

- 2 Milliarden Karten im Umlauf
  - Sehr beliebt im Nahverkehr
    - ~ 85% Marktanteil
    - Rio de Janeiro, São Paulo, Madrid, Valencia, Oslo, Sydney, Hamilton, Delhi, Nanjing, Shanghai, Taipei, Kuala Lumpur, Atlanta, St. Paul, Houston, Los Angeles, Bangkok, Netherlands, London, Boston, ...
  - Auch vielfach in Zugangskontrolle eingesetzt





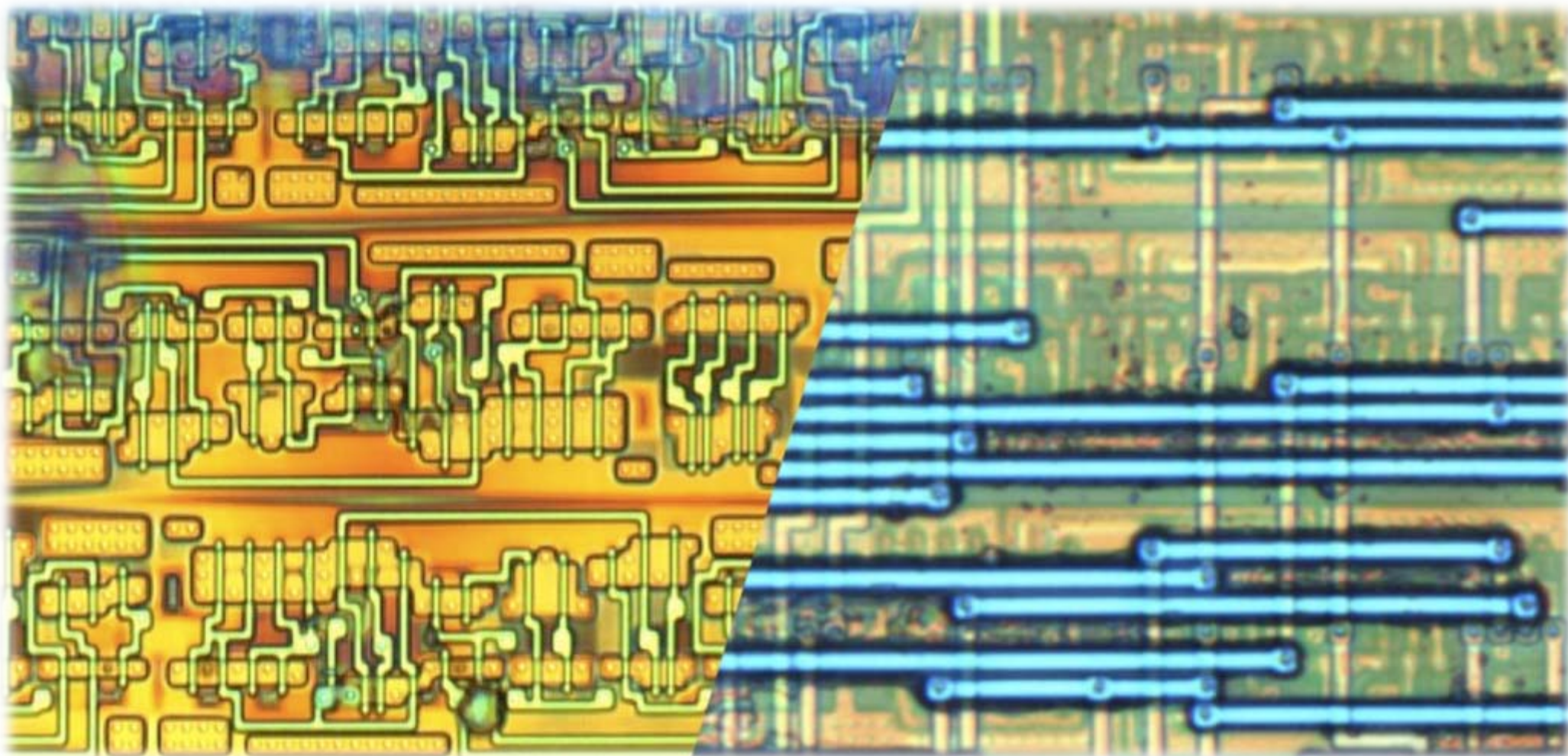
# Mifare Chip Analyse



- Chips mit Schmirgelpapier öffnen
- Chip-Schichten fotografieren
- Strukturen automatisiert in Schaltungen zurückführen

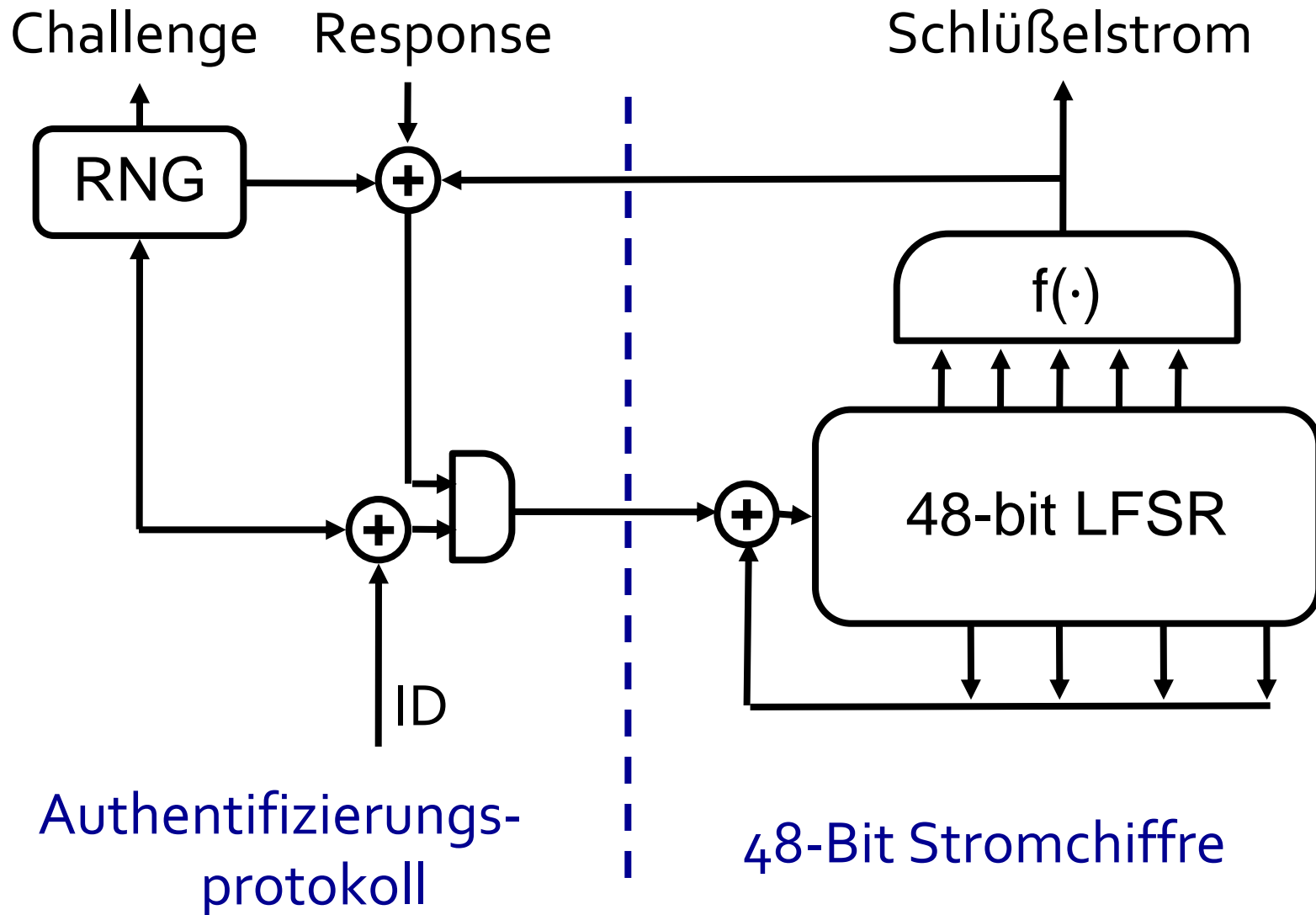


# Chip Reverse-engineered durch Abschleifen & Mikroskopie



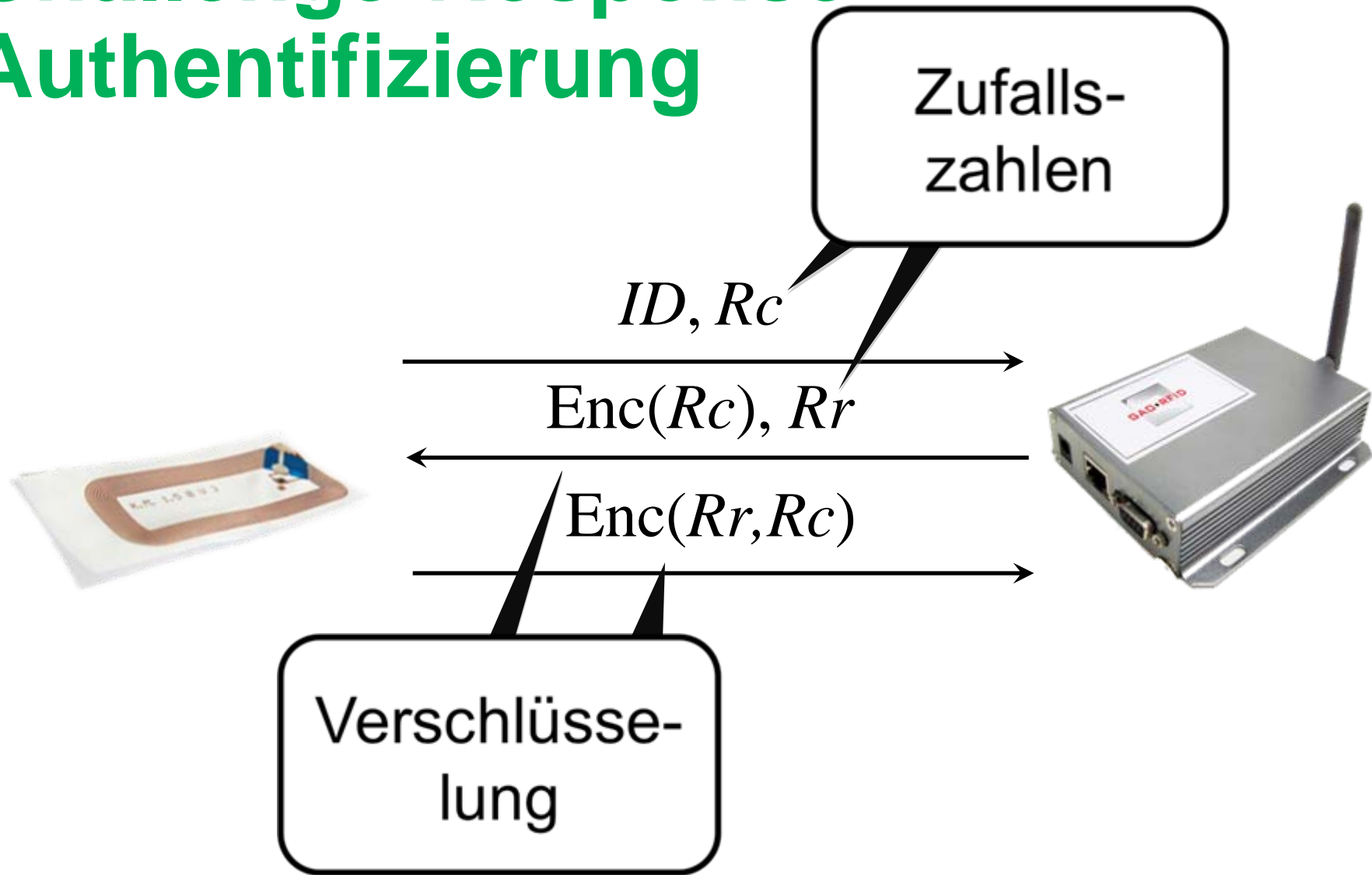


# Mifare Crypto-1



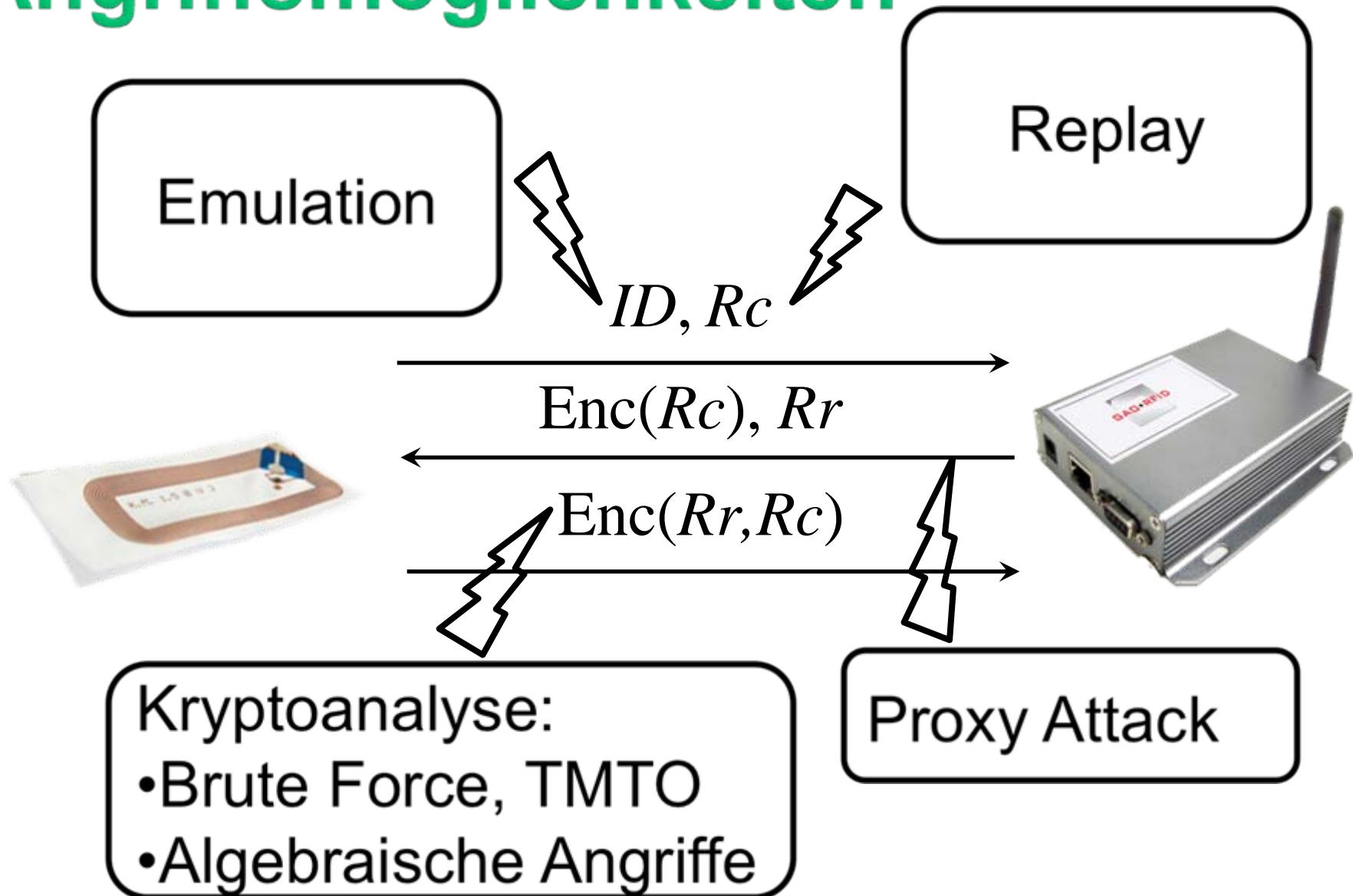


# Challenge-Response Authentifizierung





# Angriffsmöglichkeiten





# Emulation

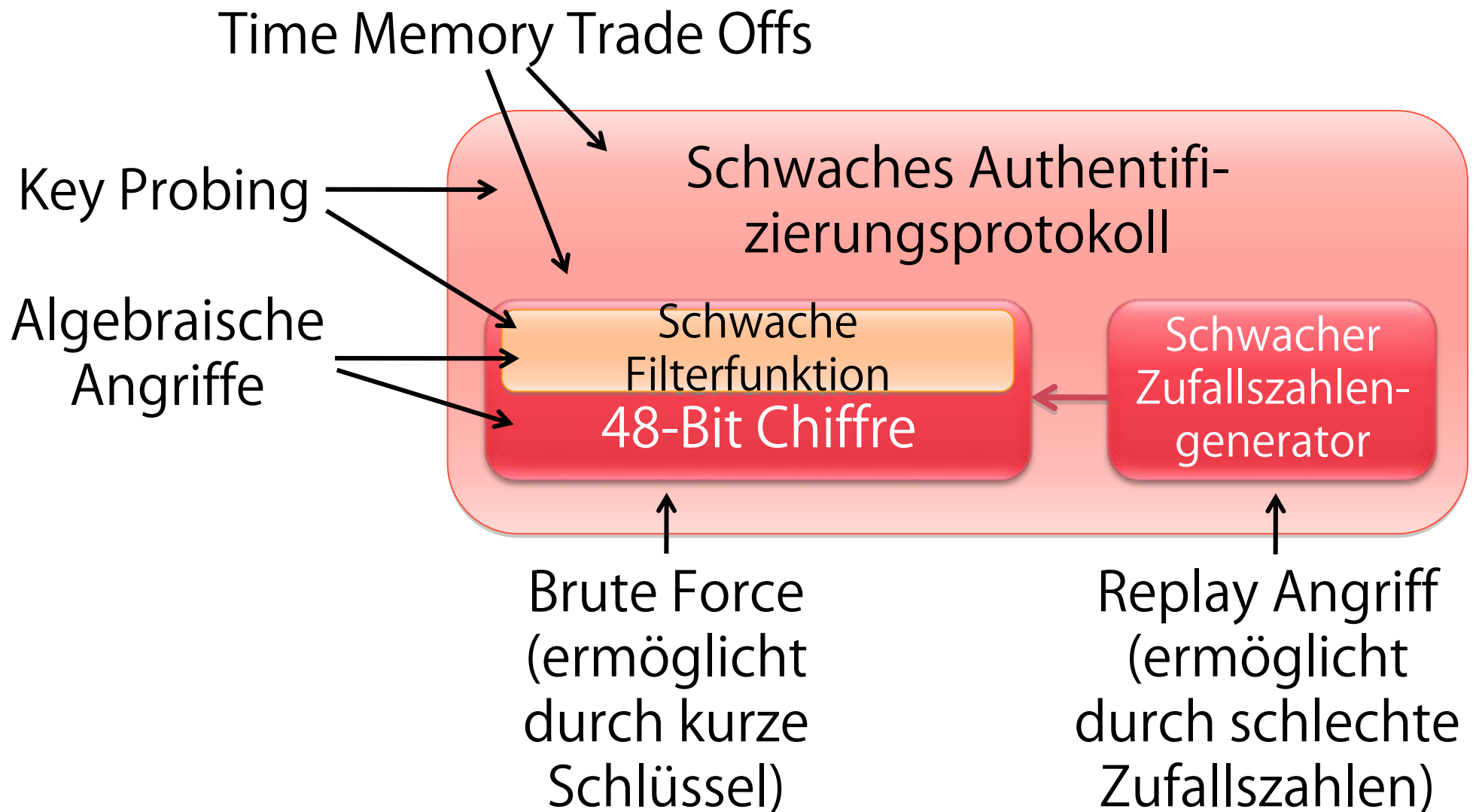
- Aufzeichnen und Abspielen der UID
- Möglich mit Emulator-Hardware (Proxmark) oder frei programmierbaren RFIDs (Soft-Tag)
- Emulation bildet die Basis für andere Angriffsvektoren







# Schwächen der Mifare Classic





# RFIDs mit schwacher Verschlüsselung

- Mifare Classic, Hitag2
  - Zahlkarten, Gebäudeschutz, Autos
- Legic (ältere)
  - Zugangskontrolle (Europa)
- HID (ältere)
  - Zugangskontrolle (USA)
- Atmel
  - CryptoRF—Zugangskontrolle
  - CryptoMemory—Schlüsselspeicher



Source: [hidglobal.com](http://hidglobal.com)



# Agenda

- 
- RFID Basics
  - Mögliche Angriffsvektoren
  - Angriffe auf „sichere“ RFIDs
  - RFID Malware
-

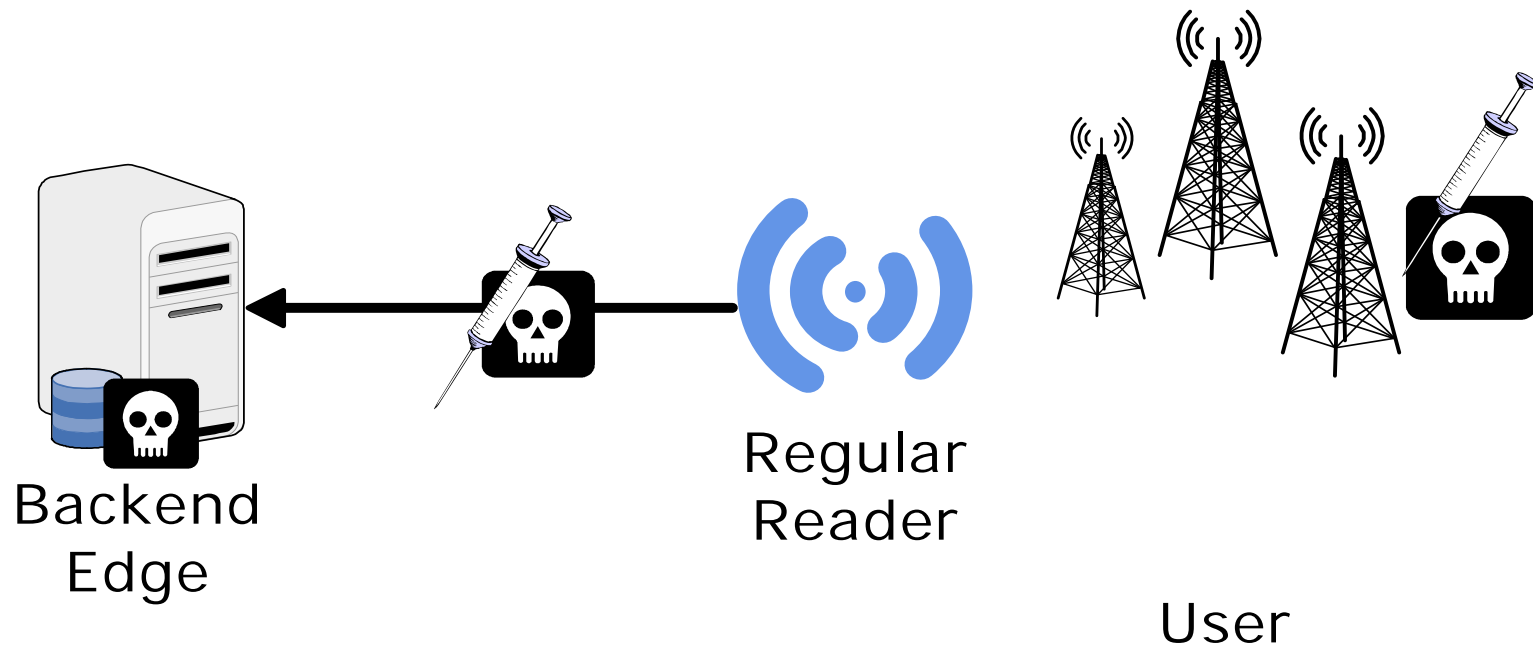


# Angriffe auf das Backend

- ISO 15693 Tag verhalten sich wie normale EEPROMS
- RFDump (Black Hat 2004) kann benutzt werden um die Daten auf dem Tag zu verändern
- Einschleusen von SQL-Injections und anderer Malware möglich



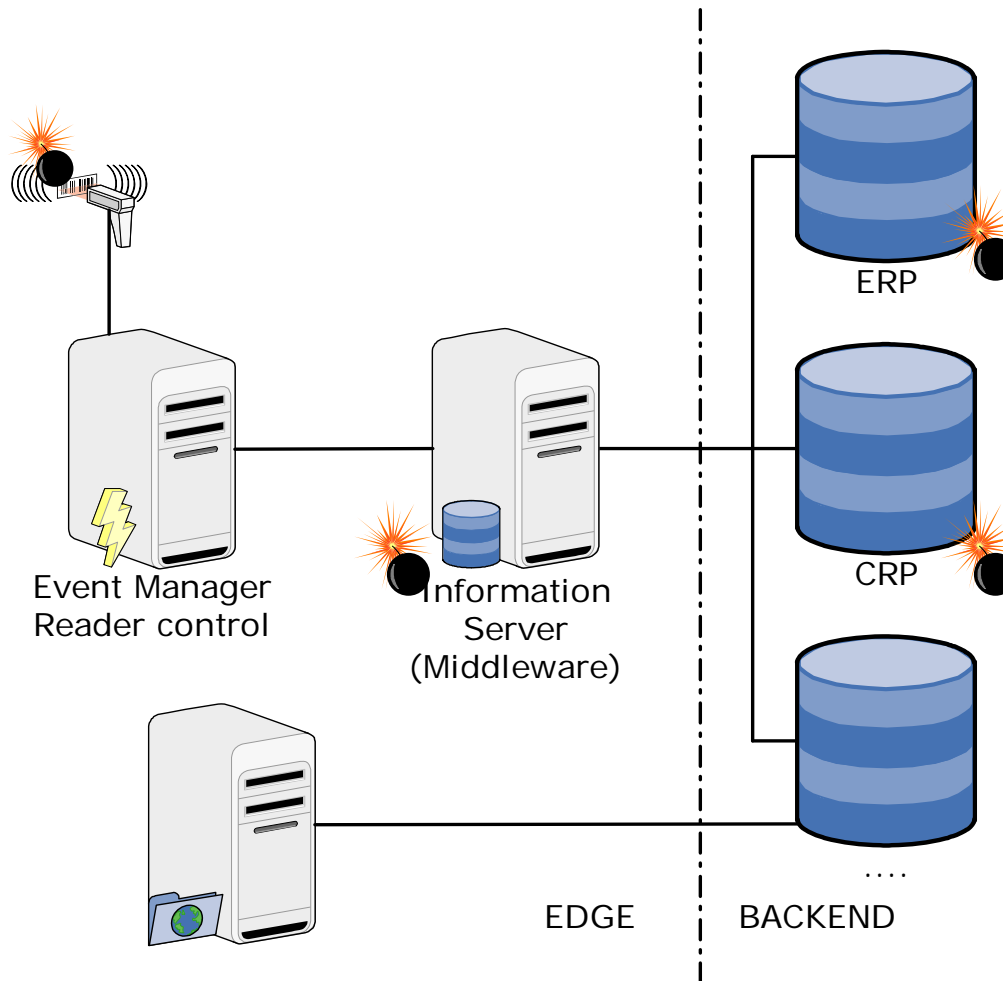
# Malware Injection



- Einschleusen selbstverbreitender Malware
  - Datenbankwürmer, RFID Viren, DoS, ...



# Einbruch in das System





# Sichere Ansätze

- RFID Systeme werden sicher durch:
  - KISS - “Keep It Stupid and Simple”
  - Vertraue keiner Eingabe
  - Prüfe alle Input Daten
    - Sind die Daten plausibel?
  - Filtere alle ungültigen Daten (PDUs)
  - Dokumentiere Anomalien und werte diese regelmäßig aus





# Take Aways

- Viele Anwendungen brauchen sichere RFIDs
  - Fälschungssicherheit und Datenschutz sind nur mit starker Kryptographie möglich
  - Proprietäre Verschlüsselung ist nicht sicher
- Alle Anwendungen brauchen eine sichere Infrastruktur
  - (Kunden-)daten sind eine wertvolle Ressource, die Hacker-Begehrlichkeiten weckt
  - Das Equivalent zu Firewalls und Virensclannern fehlt noch in fast allen RFID Systemen

# Fragen?



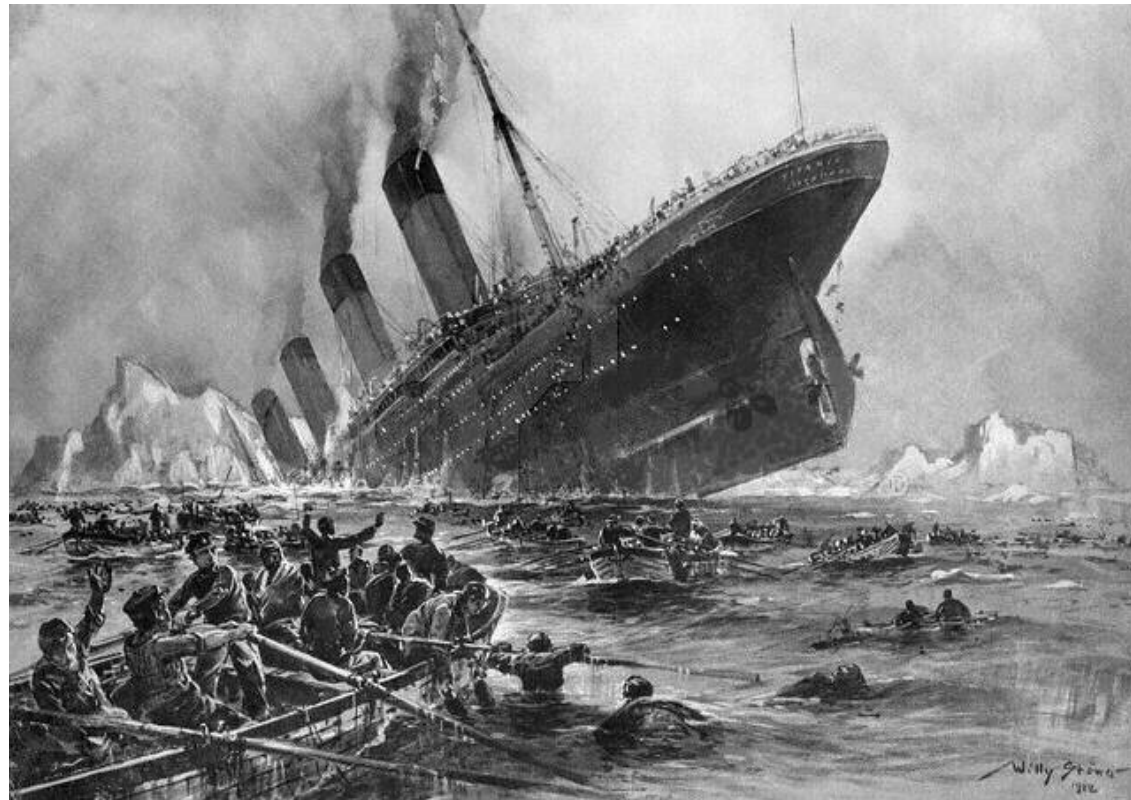
NEOCATENA NETWORKS INC.  
*>> Next Generation RFID Security >>*

**Lukas Grunwald**  
**lukas@neocatena.com**

**Karsten Nohl**  
**nohl@virginia.edu**



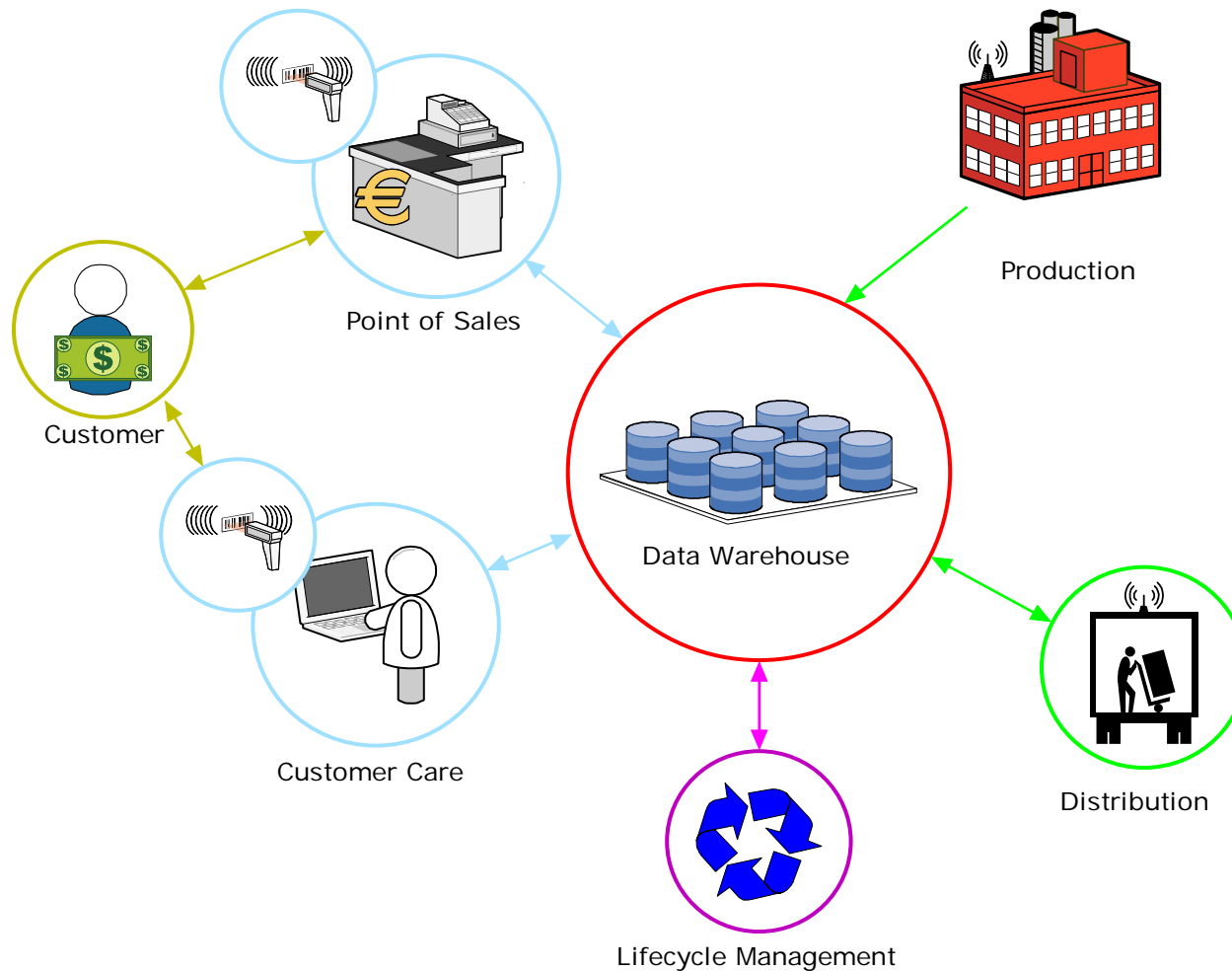
# Fragen?



**High-tech  $\neq$  High-security**

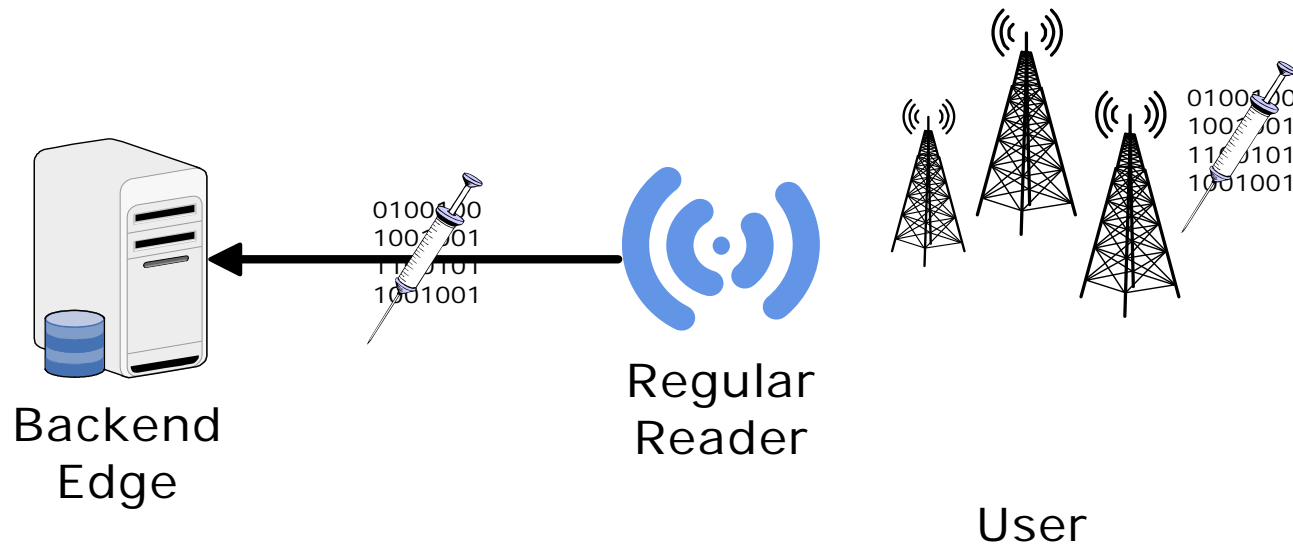


# Die Digitale Supply Chain





# Code Injection





# Abbildung für das Backend

- Tag sieht nach Endlosspeicherband aus
  - E.g. RFDump benutzt eine Tag Datenbank um das Überlesen zu verhindern
- Normalerweise ist das Lesen ereignisgesteuert
  - Lesen bis zum EOF (fehlende Markierung)
  - In den meisten Anwendungen wird die Eingabe ungeprüft weiterverarbeitet



# Code Injection

- Einschleusen von ausführbaren Programmteilen in das RFID Tag
  - SQL injection
  - Shell-Code
  - String format attack
  - Buffer overrun
- Angriff auf die Edge-Server, Middleware und Backends mit manipulierten RFID Tags
- Nicht selbst verbreitender Angriff



# Problem Speichergrösse

| Adr | Memory  |
|-----|---|
| 0x1 | 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 |
| 0x2 | 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 |
| 0x3 | 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 |
| 0x4 | 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 |
| 0x5 | 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 |
| 0x6 | 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 |
| 0x7 | 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 |
| 0x8 | 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 |
| 0x9 | 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 |
| 0xa | 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 |
| 0xb | 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 |
| 0xc | 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 |
| 0xd | 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 |
| 0xe | 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 |
| 0xf | 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 |

Page 0x76  
Byte 6



# Proposed Mitigations

- Countermeasures for Mifare Classic include:
- Signing:
  - Strongly authenticate data to prove authenticity
  - “Valid states” can be tied to cards and times iff emulation is detectable
- Radio fingerprinting:
  - Measure and verify physical properties of tags
  - Potential to detect emulation
  - (see Day1 talk “RFID fingerprinting” by cryptocrat, Boris Danev)



# DoS Angriff mit C-Strings

End of String

| Adr | Memory  |
|-----|---|
| 0x1 | 68547369 69202073 6e616520 6178706d 656c6f20 20662061 616d696e 75706100 |
| 0x2 | FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF |
| 0x3 | FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF |
| 0x4 | FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF |
| 0x5 | FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF |
| 0x6 | FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF |
| 0x7 | FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF |
| 0x8 | FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF |
| 0x9 | FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF |
| 0xa | FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF |
| 0xb | FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF |
| 0xc | FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF |
| 0xd | FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF |
| 0xe | FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF |
| 0xf | FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF |



# Tag DoS mit XML

Mass reading

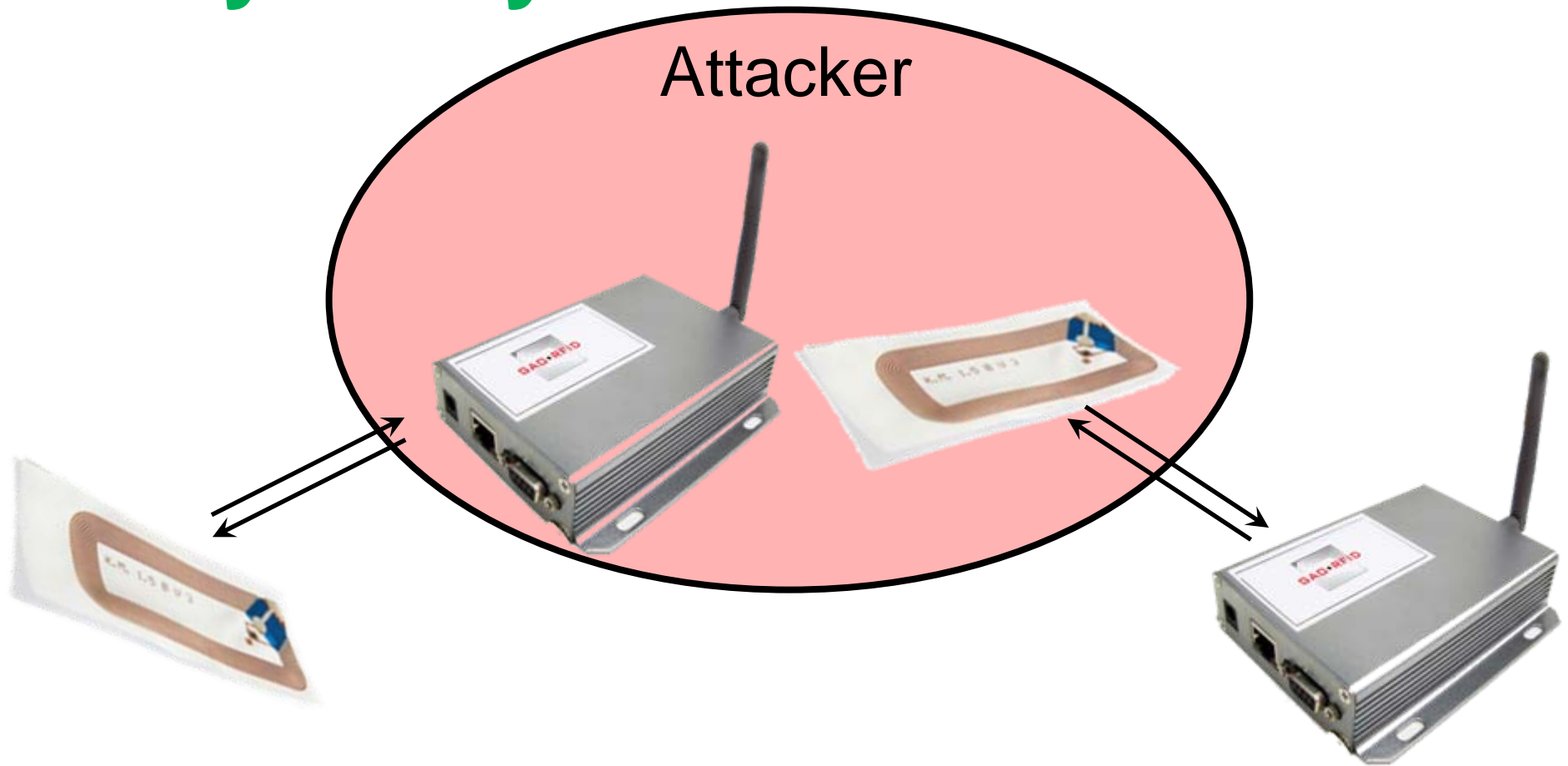
| Addr | Memory in ASCII                              |
|------|--|
| 0x1  | <rfiduid:ID>urn:epc:1:4.16.36</rfiduid:ID>   |
| 0x2  | <rfidcore:Observation><rfidcore:DateTime>    |
| 0x3  | <rfidcore:DateTime>2002-11-06T13:04:34-06:00 |
| 0x4  | </pmlcore:DateTime>                          |
| 0x5  |  |
| 0x6  |  |

Inf. Items in one Tag

| Addr | Memory in ASCII  |
|------|--|
| 0x1  | <rfiduid:ID><rfiduid:ID><rfiduid:ID><rfiduid:ID><rfiduid:ID><rfiduid:ID> |
| 0x2  | <rfiduid:ID><rfiduid:ID><rfiduid:ID><rfiduid:ID><rfiduid:ID><rfiduid:ID> |
| 0x3  | <rfiduid:ID><rfiduid:ID><rfiduid:ID><rfiduid:ID><rfiduid:ID><rfiduid:ID> |
| 0x4  | <rfiduid:ID><rfiduid:ID><rfiduid:ID><rfiduid:ID><rfiduid:ID><rfiduid:ID> |
| 0x5  | <rfiduid:ID><rfiduid:ID><rfiduid:ID><rfiduid:ID><rfiduid:ID><rfiduid:ID> |
| 0x6  | <rfiduid:ID><rfiduid:ID><rfiduid:ID><rfiduid:ID><rfiduid:ID><rfiduid:ID> |



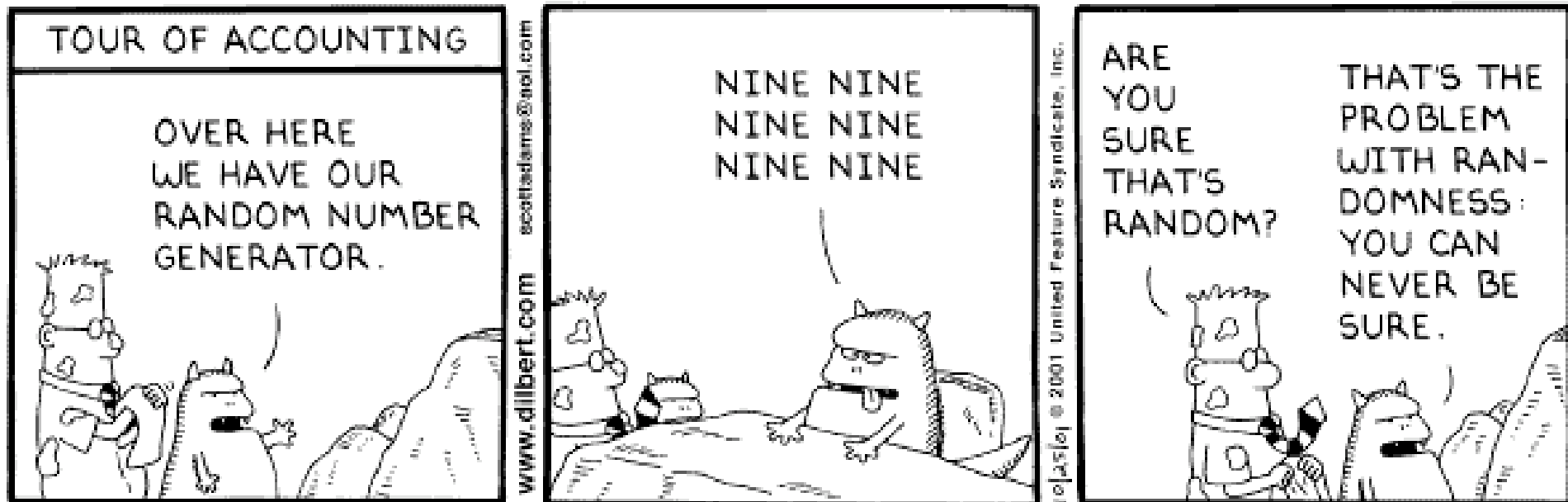
# Proxy/Relay Attack





# Replay

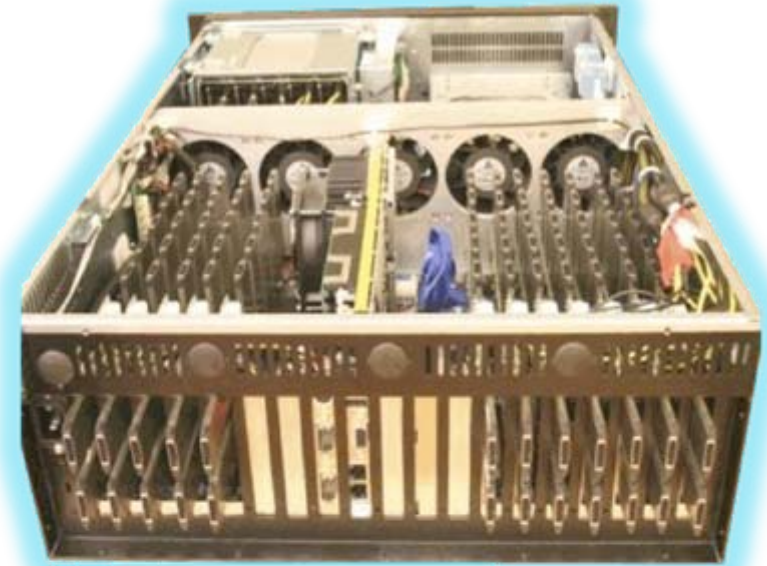
1. Overhear legitimate authentication
  2. Force same challenge, answer with same response
- Requires predictable “random” numbers





# Brute Force Key Search

- “Try all keys”
- Only possible for small keys
- Mifare easy target:
  - Cipher complexity low, enables efficient FPGA implementation
  - FPGA cluster finds key in 50 minutes!



Source: Pico Comp.



# Time-Memory Trade-Offs

- Basic idea: Pre-compute and compress code book
- Corner cases:
  - Brute Force:  $O(N)$  time
  - Full code book:  $O(N)$  space
- Trade offs exists between:
  - Time – space – data/success
- Countermeasure: use IVs

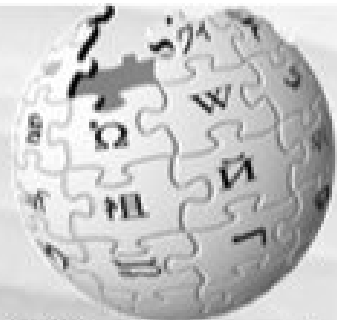
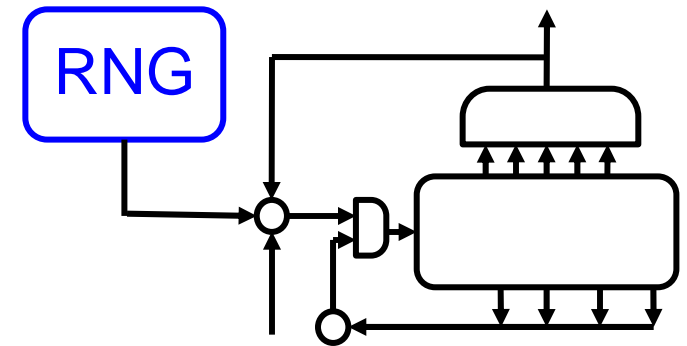


German article "Kunterbuntes Schlüsselraten" on heise.de



# Replay — Mifare

- Mifare random numbers are completely predictable and well documented



WIKIPEDIA  
 The Free Encyclopedia

article

discussion

edit this page

history

## Linear feedback shift register

From Wikipedia, the free encyclopedia  
 (Redirected from [LFSR](#))

A **linear feedback shift register** (LFSR) is a [shift register](#) whose input bit is a [linear](#) function of its previous. The only linear functions of single bits are xor and inverse-xor; thus it is a shift register whose input bit is driv

The tap sequence of an LFSR can be represented as a [polynomial mod 2](#). This means that the coefficients of polynomial. For example, if the taps are at the 16th, 14th, 13th and 11th bits (as below), the resulting LFSR p

$$x^{16} + x^{14} + x^{13} + x^{11} + 1$$

### navigation

- [Main Page](#)
- [Contents](#)
- [Featured content](#)
- [Current events](#)
- [Random article](#)



# Algebraic Attacks

- Attacks that exploit simple feedback structure and statistical weaknesses:
- Describe weak parts of cipher as system of equations
- Brute-Force through complex parts:  
Guess-and-Determine attack.
- Solve system of equations:  
MiniSAT is our friend

