University of BRISTOL

Security Analysis of an Open Car Immobilizer Protocol Stack

4th International Conference on Trusted Systems (InTrust 2012)
December 17-18, Egham, UK

Stefan Tillich and Marcin Wójcik

Security Evaluation

- “Closed”
  - Algorithms & protocols (often proprietary) not disclosed
  - Evaluation by hired experts (usually under NDA)
- “Open”
  - Algorithms & protocols published
  - Feedback by any interested party

What's in a Key (Fob)?

- Several functions using various wireless communication channels
  - **Immobilizer** (LF)
  - Remote Keyless Entry (RF)
  - Passive Entry Go (LF/RF)
  - Remote Start (RF)
  - ...

Immobilizer Principles

- Car checks presence of security token
- Usually short-range LF (near field) communication (125 kHz)
- Car's Engine Control Unit (ECU) conditionally interrupts ignition, starter motor circuit, and/or fuel pump

Open Source Immobilizer Protocol Stack

- Presented at ESCAR 2010
- Goals:
  - Increase interoperability of ICs from different vendors
  - Allow for public review of security
- Implemented in various immobilizer devices

Characteristics

- 125 kHz full-duplex communication
  - Near field -> close range
- Configurable protocol parameters
- Simple command-response structure
  - Reader (usually car) sends command, key fob responds
- 11 commands defined

Command Set

- Read UID
- Read Transponder Error Status
- Start Authentication
- Learn Secret Key1/2
- Initiate/Leave Enhanced Mode
- Repeat Last Response
- Read/Write User Memory
- Write Memory Access Protection

Relay Attack

- Simulate proximity of car and key fob by relaying their communication via a transparent reader and key fob
- General attack
  - Not specific to protocol stack

Prevent Relay Attack

- Car could measure communication delay
- Problem:
  - Transparent key fob could fake CRC error
  - Car then sends “Repeat Last Response”
  - -> More time for attacker

Prevent Relay Attack (cont'd)

- Abandon repeating of responses
  - Instead, repetition of whole command & response sequence after CRC error
- Overhead:
  - Reader: Hardware support for precise timing measurements
  - Both: More costly CRC error handling

Tracking

- "Read UID" command
  - Key fob returns 32-bit unique ID (UID)
  - No reader authentication required
  - UID can be queried by arbitrary readers
- Other commands also usable but less versatile
- Possible read range important factor

See full paper @ eprint.iacr.org for details

Prevent Tracking (I)

- Enhance “Read UID”
  - e.g. Return encrypted UID and nonce: $E_K(\text{nonce} \oplus E_K(\text{UID}))$, nonce
- Response will change for every query
- Overhead:
  - Reader: 2 AES decryptions
  - Key fob: Transmit 128-bit ciphertext & nonce (instead of UID); nonce gen., 1 AES encryption

Prevent Tracking (II)

- New command: “Check UID”
  - Just to check for a specific UID
  - Key fob returns only a part of encrypted UID and the full nonce
  - Reader repeats the same encryptions and checks for equality
- Communication overhead reduced

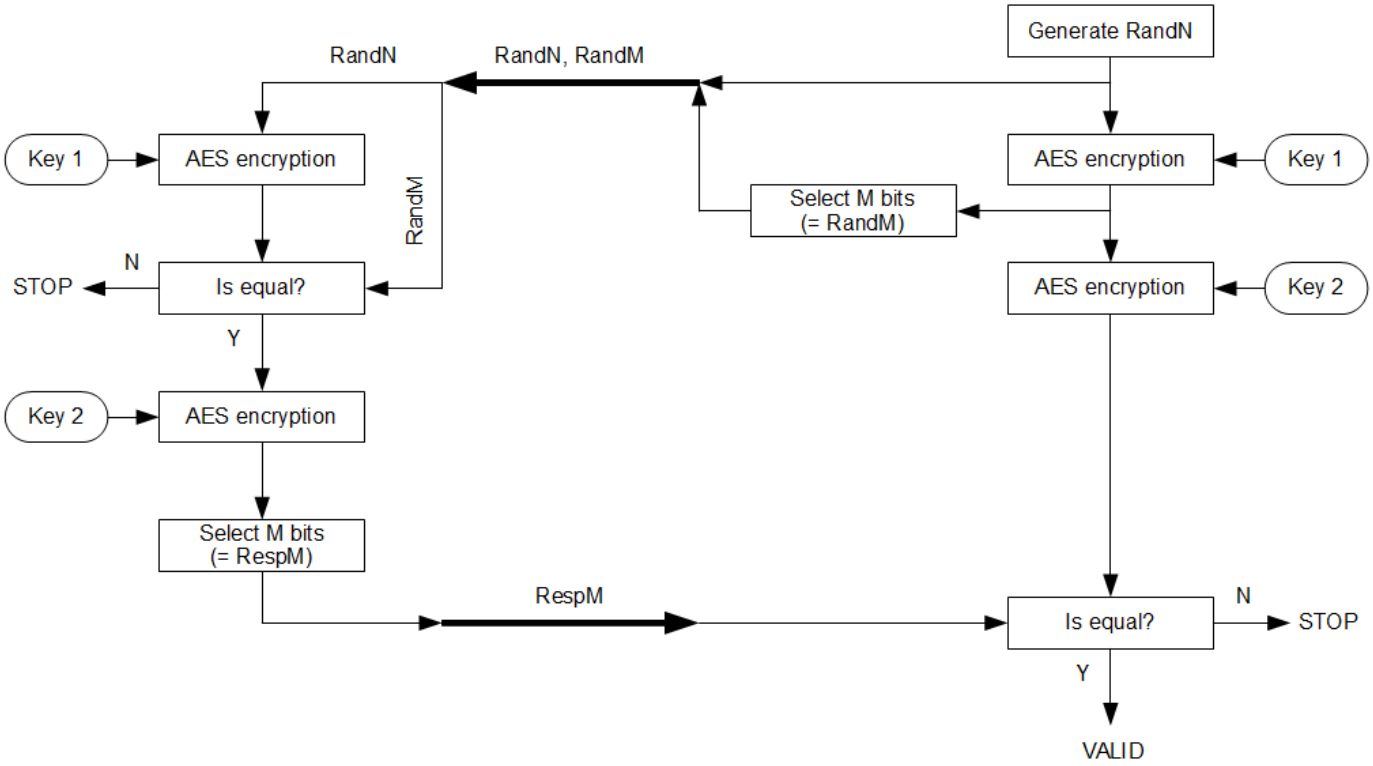
Denial-of-Service

- Make key fob unusable
- “Learn Secret Key1(2)” commands
  - Open mode
    - Any reader can set new key of its choice
  - Secure mode
    - New key encrypted with factory-set default key, but no integrity check
    - Send random 128-bit block to set (unkown) key

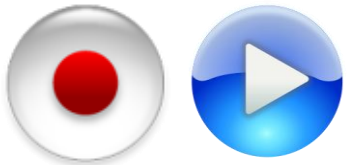
Prevent Denial-of-Service

- Open mode is susceptible by design
- Secure mode
  - Include MAC of encrypted new key
  - Only set new key upon successful verification
  - Overhead:
    - Reader: Generate & transmit MAC
    - Key fob: Verify MAC

Bilateral Authentication



Replay Attack



- Record commands by car and replay to authenticate any reader to key fob
- Gives access to advanced commands like “Read/Write User Memory”

Prevent Replay Attack (I)

- Key fob generates a challenge for reader
- Overhead:
  - Key fob: Generate nonce
  - Both: Extra command-response sequence, where car returns encrypted nonce

Prevent Replay Attack (II)

- Authenticated timestamp/counter value in „Start Authentication“ command from Reader
- Key fob stores timestamp of last successful authentication
  - Authentication aborted if new timestamp not more recent than stored one
- Overhead:
  - Reader: Generate timestamp
  - Key fob: Store timestamp

See full paper @ eprint.iacr.org for details

Spoofing

- EEPROM sections of key fob can be locked via “Write Memory Access Protection” command (without prior authentication)
- Depending on use of these sections, key fob functionality could be impaired

Prevent Spoofing

- Require prior authentication
- Overhead:
  - Both: Extra authentication

Session Hijacking

- After successful authentication with „Start Authentication“ command, privileged commands can be executed
- Malicious reader waits for authentication between car and key fob
  - Then overshadows car's communication to inject privileged commands

Prevent Session Hijacking (I)

- Authenticated key agreement to generate session key
  - Generate MAC for subsequent messages
- Overhead:
  - Reader: MAC generation & transmission
  - Key fob: MAC verification
  - Both: Key agreement

Prevent Session Hijacking (II)

- Make all privileged commands as variants of „Start Authentication“
  - Individual authentication of each privileged command
- Overhead:
  - Reader: Generate timestamp, transmit timestamp & its authentication value
  - Key fob: Transmit authentication response
  - Both: 2x AES encryption

Conclusions

- Described various potential threats on immobilizer protocol stack security
- Proposed possible countermeasures
- Communication range achievable by attacker impacts threat


Q&A