Security aspects of RFID-enabled Car Keys

Jos Wetzels

TU/e Technische Universiteit **Eindhoven** University of Technology

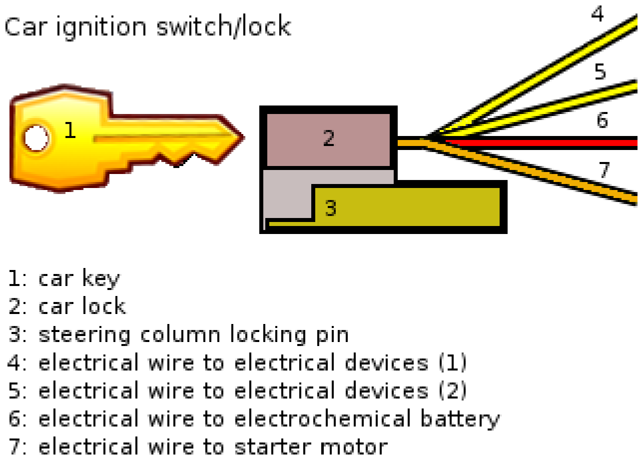
Where innovation starts

Content

- **Introduction**
- **Remote Keyless Entry (RKE)**
  - General attacks
  - Example: **KeeLoq**
- **Ignition Immobilizers**
  - Examples: **DST-40**, **HITAG 2**
- **Conclusion**

Introduction

- **Traditional car security:**
  - Mechanical key to open doors, glove box & start ignition
- **Problems:**
  - Lock-picking, key theft, cloning
  - Hotwiring
- **RFID car keys:**
  - Mitigate old problems, introduce new ones
  - Proprietary, non-standardized nightmare
- **Two functionality 'flavors':**
  - Remote Keyless Entry (RKE)
  - Ignition Immobilizers



Source: *Wikimedia Commons*

RKE Systems

- **Introduced for user-convenience (not security!) in the '80s**
  - Remotely lock & unlock doors
  - RF circuit in key fob
  - Usually operates on 315MHz~433.92MHz

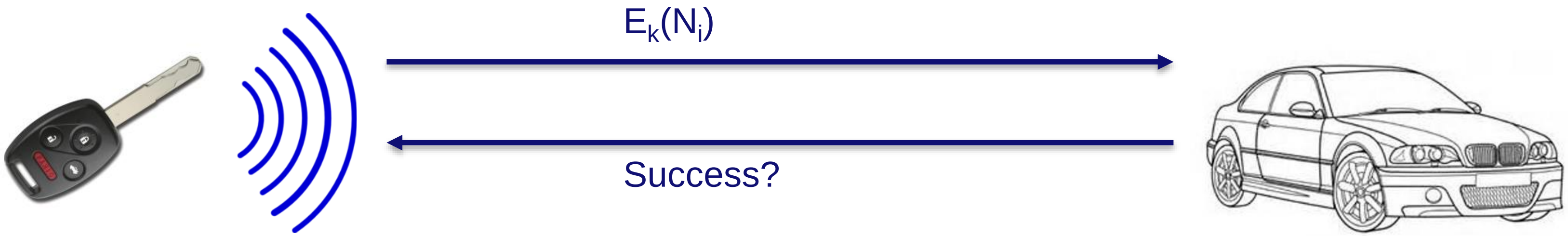
- **Active vs Passive**
  - Active RKE:
    - **Press button** to lock/unlock
    - **Insert mechanical** key to start car
    - < 100m distance

  - Passive RKE (aka 'Smart Keys')
    - **Pull door handle** while close (<2m) to car to open
    - **Insert mechanical key** or press 'start button' while inside car to start

Source: *Google images*

Security – RKE Systems

- **Authentication protocols:**
  - Fixed code
    - Transmit secret key k (shared with vehicle) in the open
  - Rolling code

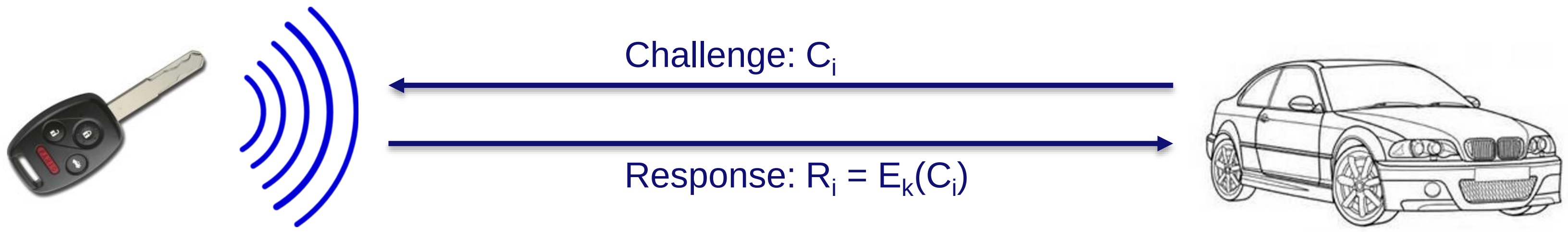


- Authentication:
  - Key fob pick sequence counter N_i
  - Encrypt using shared secret key k
  - Key fob updates to N_{i+1}
  - Vehicle receives, checks if difference(N_i, M_j) < threshold
  - Success? Perform action & update to M_{j+1}
  - Diffcheck is to prevent desync from accidental keypress

Technische Universiteit Eindhoven University of Technology

Security – RKE Systems

- **Authentication protocols:**
  - Challenge-Response



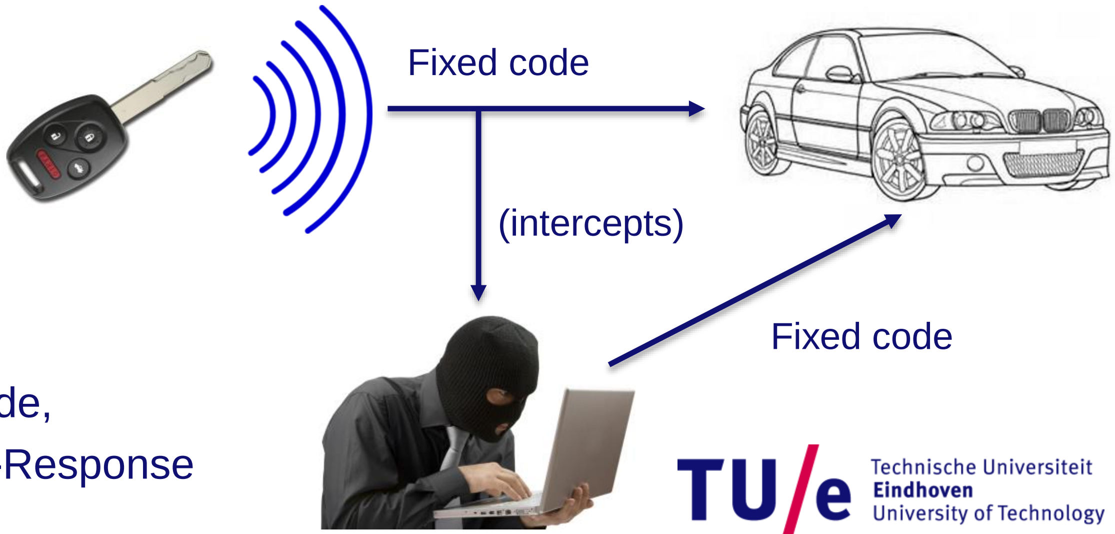
- Authentication:
  - Vehicle generates random challenge C_i
  - Vehicle computes $R'_i = E_k(C_i)$ using shared secret key k
  - Send C_i to keyfob

  - Keyfob computes $R_i = E_k(C_i)$, sends to vehicle
  - Success iff $R'_i = R_i$

Security – RKE Systems

- **Attacks:**
  - **Brute-force**
    - Exhaustive: every possibility in key space
    - Scanning: respond with constant (intercepted) response against challenge

    - *Mitigation*: Strong cipher, strong key



  - **Replay**
    - Intercept + Resend

    - *Mitigation*: Rolling Code,
      Challenge-Response

Security – RKE Systems

- **Man-in-the-Middle (MITM)**
  - *Focus*: Challenge-Response (since authentication is vehicle-initiated)

  - Trigger authentication (eg. pulling door handle)
  - Intercept challenge, transmit to second attacker close to key fob
  - Key fob responds to attacker, relays to vehicle

  - After relay attack against RKE to enter vehicle, relay can be used against immobilizer too

  - *Mitigation*: Key pouch (RF shielding), distance bounding

Security – RKE Systems

- **Forward Prediction**
  - *Focus*: Challenge-Response

  - Collect series of challenges $\{C_i,..,C_n\}$
  - Generated with weak PRNG? => Predict subsequent challenge C_{n+1}
  - Send C_{n+1} to key fob, collect R_{n+1}
  - Trigger authentication, vehicle sends C_{n+1}
  - Respond with R_{n+1}

  - *Mitigation*: CSPRNG, delay after certain # of unanswered challenges

Security – RKE Systems

- **Dictionary attack**
  - *Focus*: Challenge-Response

  - Generate series of random challs, send to key fob
  - Key fob replies with responses
  - Repeated until dictionary with high % success
  - Trigger authentication until challenge in dictionary
  - Look up challenge and reply with response

  - *Mitigation*: Sender verification in key fob, delay after # of unanswered challenges, key shielding

Security – RKE Systems

- **Jamming attack**
  - *Focus*: rolling-code

  - *Simple scenario:* Jam lock signal so car doesn't get locked
  - Problem: car gives confirmation (lights + horn)

  - *Better scenario:* Jam first transmission, record code
  - Jam second transmission, record code, send first to lock car
  - At later moment, send second code to unlock

  - *Mitigation*: Introduce nonces, eg. rolling code $C_i = E_k(N_i + \text{nonce})$

Technische Universiteit **Eindhoven** University of Technology

Security – RKE Systems

- **Cryptographic & Side-Channel attacks**
  - Design varies from vendor to vendor, proprietary
  - Example: **KeeLoq**

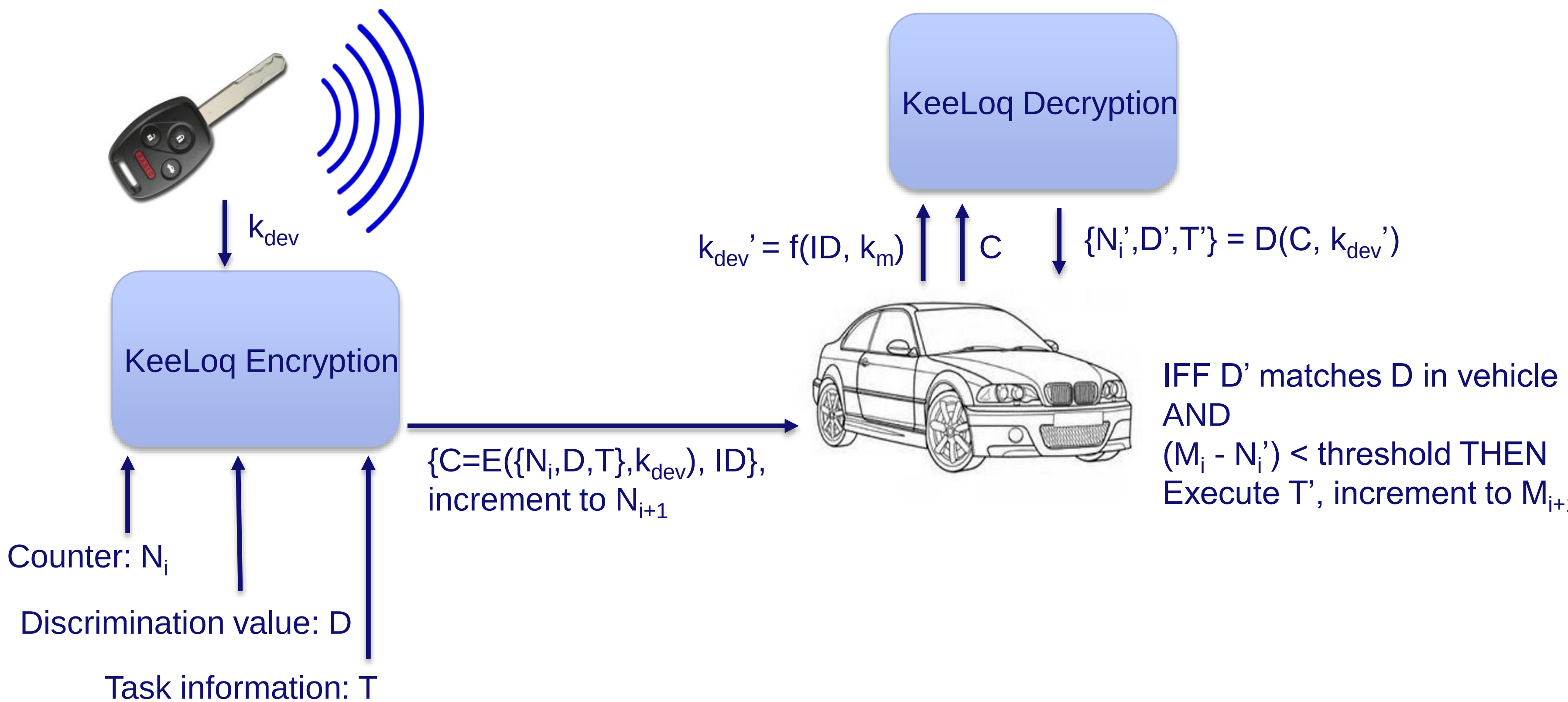
  - Rolling code mode: RKE systems
  - Challenge-Response mode: immobilizers

  - OEM assigned manufacturer key k_m , stored in all receivers
  - New transponder gets device key $k_{dev} = f(ID, k_m)$
  - Key derivation function *f*:
    - Weak XOR function, or
    - KeeLoq encryption of ID using k_m as key
    - In some modes, ID combined random (32,48 of 60 bit) shared seed

Technische Universiteit
Eindhoven
University of Technology

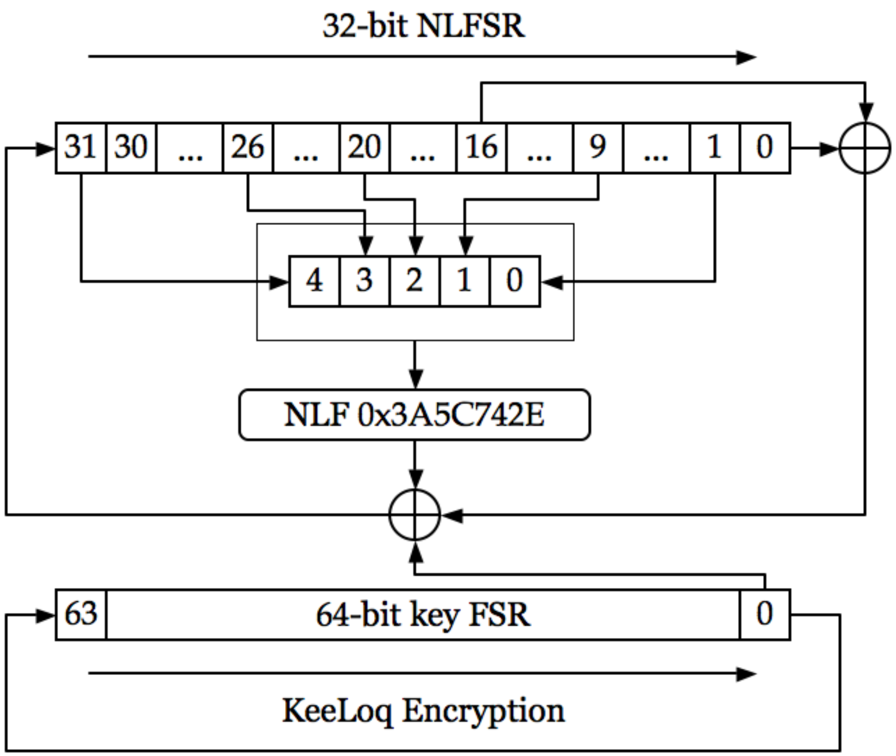
Security – RKE Systems

- **KeeLoq Rolling code scheme**



Security – RKE Systems

- **Keeloq Cipher**
  - NLFSR-based block cipher
    - 32-bit blocks
    - 64-bit key
    - 528 rounds
  - Operation:
    - Key to key register
    - Plaintext to state register
    - Each clock cycle:
      - Key register rotated right
      - State register shifted right
      - Fresh bit from XOR part of state
    - After 528 cycles state register holds ciphertext



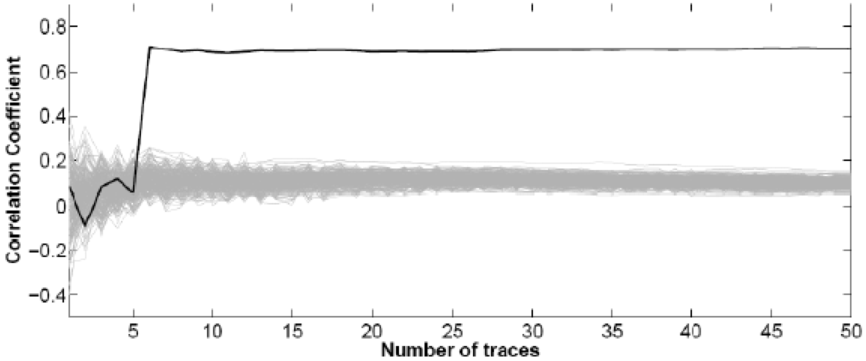
Source: *Wikimedia Commons*

Security – RKE Systems

- **Attacks on KeeLoq**
  - Best cryptanalytic attack targets C-R mode
    - Slide/Meet-in-the-Middle (Indesteege et al.)
      - Possible because multiple rounds of the same function F (vuln. to known-plaintext attack)
    - Challenges not authenticated so chosen plaintext is feasible
    - 2^{16} chosen plaintexts (65 minutes gathering)
    - 3.4 days cracking on 64 CPU cores
    - *Problem*:
      - weak key derivation => recover key, otherwise only clone transponder
      - C-R not dominant application of KeeLoq
      - *Better*: Brute-force using COPACOBANA recovers keys derived from 48-bit seed in under 6 hours
    - *Mitigation*: Strong, scrutinized cryptography (eg. AES)

Security – RKE Systems

- **Attacks on KeeLoq**
  - Differential Power Analysis (DPA)
    - Variations in power consumption
      - Related to state register
    - Only 2 intercepted messages
    - Transponders implement cipher in hardware
      - Extract device key from only 5~30 power traces
      - Clone any transponder within minutes
    - Receivers implement cipher in software
      - Extract manufacturer key from ~10.000 power traces
      - Generate new/clone any transponder, *but*:
        harder to execute, takes hours



Source: *Breaking KeeLoq with Power Analysis,* Eisenbarth et al.

Technische Universiteit Eindhoven University of Technology

Security – RKE Systems

- **Attacks on KeeLoq**
  - Simple Power Analysis (SPA)
    - Source-code of software implementation became available

      - Implementation leaks key dependent information
      - Constant cycle consumption except lookup table to build NLF
      - Execution time varies for different ciphertexts -> SPA vulnerable

    - Allows for extraction of manufacturer key from receiver from a single power trace

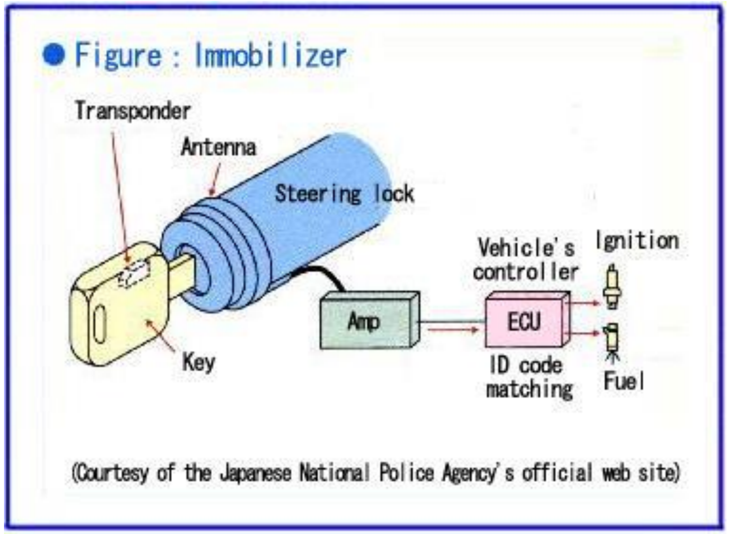
    - *Mitigation*: Problematic, eg. constant run times open up to timing attacks

Technische Universiteit Eindhoven University of Technology

Ignition Immobilizers

- **Introduced as an anti-theft measure**
  - Standard since 1995, often (insurance) mandatory
  - Unauthorized ignition attempts sometimes log data for investigation

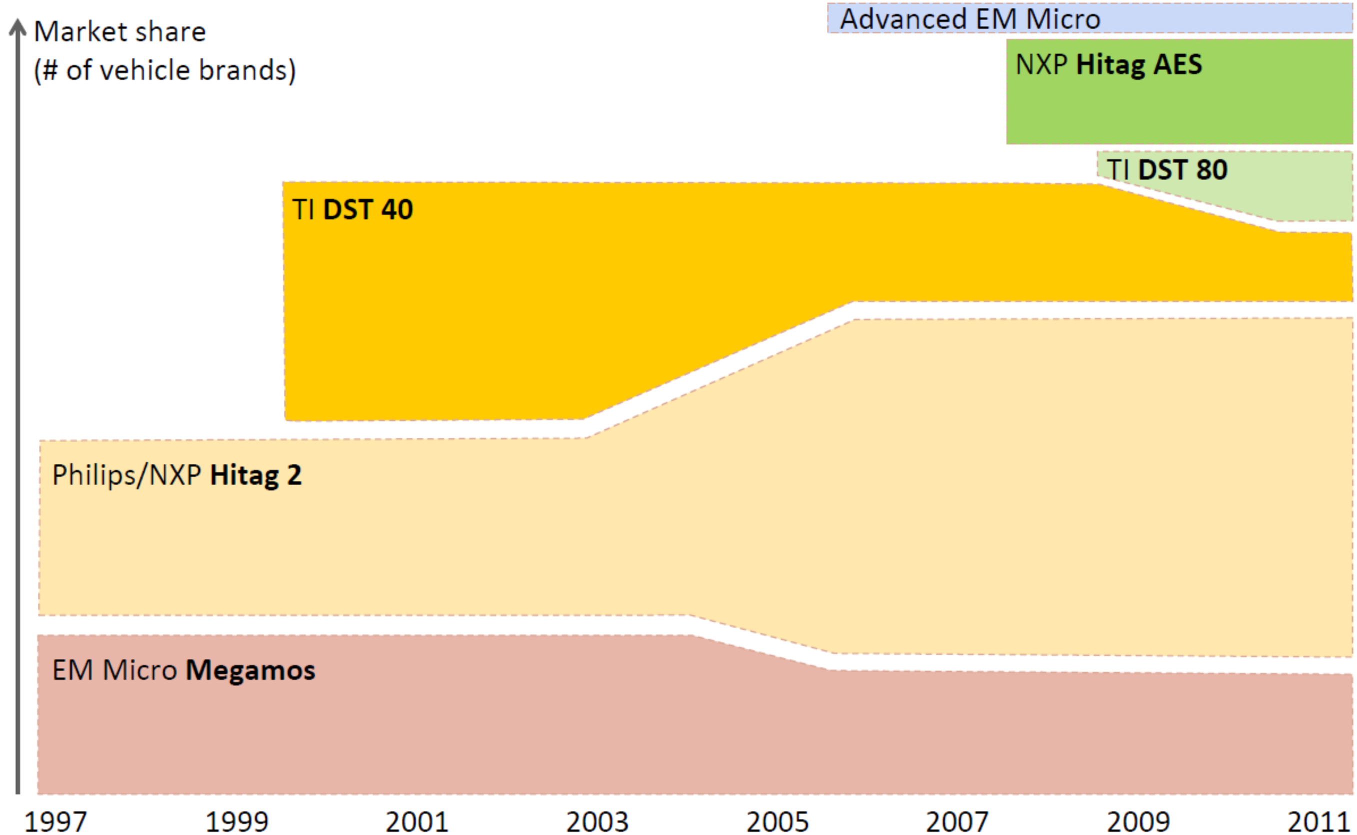
- **How does it work?**
  - Insert mechanical key/press start button
  - Vehicle initiates authentication with key fob
  - Success? Car is started
  - Passive RFID, close proximity (~cm)
  - RKE & Immobilizer in one key fob



- **Security?**
  - Vulnerable to many of same attack types as RKE

Ignition Immobilizers

- **Market dominated by three solutions**



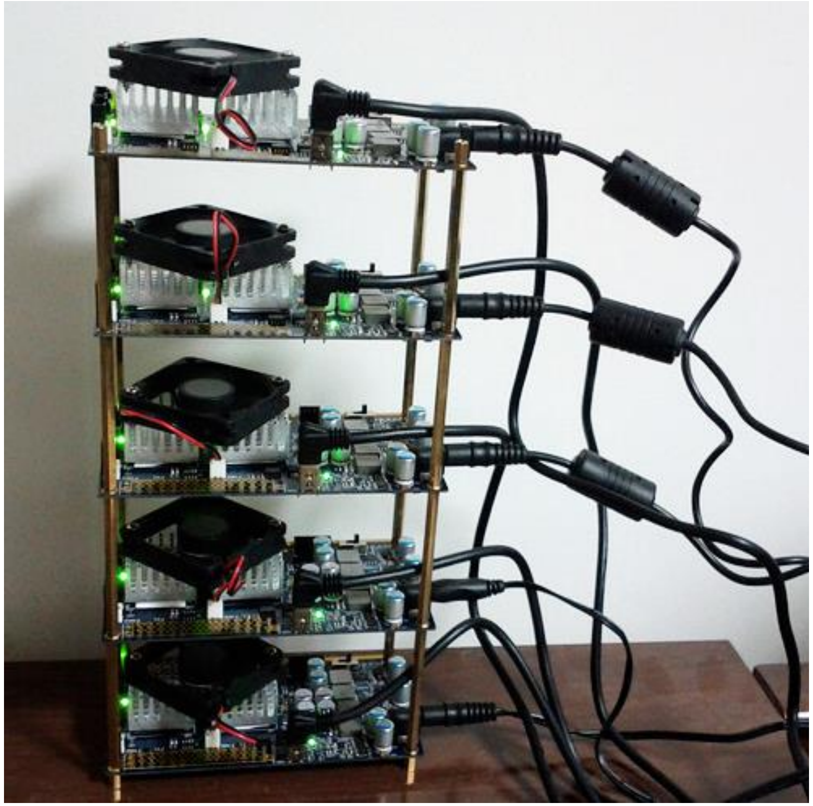
Source: *Car immobilizer hacking,* Karsten Nohl

Technische Universiteit **Eindhoven** University of Technology

Security – Ignition Immobilizers

- **DST-40**
  - CR-scheme
    - Block cipher
    - 40-bit key
    - 200 rounds
    - LFSR key schedule w. 3-round period
      - Some weak keys (eg. zero key)

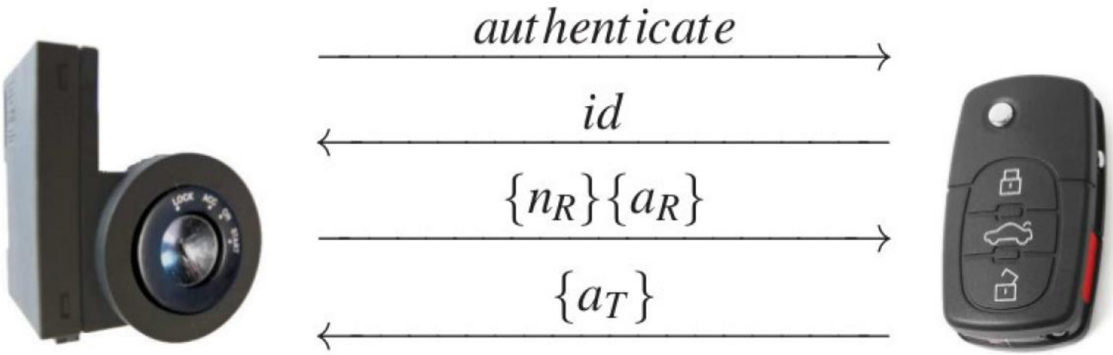
  - Bruteforce with 2 intercepted pairs
    - 1 FPGA: 11 hrs
    - 16 FPGAs: 1 hrs
    - FPGA-generated lookup table: seconds



Source: *Car immobilizer hacking,* Karsten Nohl

Security – Ignition Immobilizers

- **HITAG 2**
  - CR-scheme
    - Stream cipher, 48-bit key
    - nR = Nonce (IV), aR = authenticator (chall), both encrypted
    - aT = encrypted transponder password (fallback in case of broken crypto)



Source: *Gone in 360 seconds,* Verdult et al.

  - <u>Bruteforce</u> with 1 intercepted pair
    - 1 CPU: 4 yrs
    - GPU-based cluster: 11 hrs (~$5k)
    - COPACOBANA: 2 hrs (expensive!)

Technische Universiteit **Eindhoven** University of Technology

Security – Ignition Immobilizers

- SAT-solver with 2 intercepted pairs
  - Reduce cipher to series of multivariate quadratic equations
  - Possible because cipher is very linear (eg. output not filtered back into state, etc.) & not too many rounds
  - *Attack time*: 2 days

Source: *Car immobilizer hacking,* Karsten Nohl

Security – Ignition Immobilizers

- <u>Malleability attack</u>
  - *Protocol flaw*: No transponder nonce (since no PRNG) => replay any {nR},{aR}
  - Extend len of any command with multiple of 5 bits (redundancy msg)
  - Replay + variable len => key stream oracle

  - Intercept 1 valid session
  - Use oracle to recover 42 keystream bits
  - Recover all memory blocks

  - *Attack time*: 1s
    - *However*: secret key only if not well configured (read protection)

Try all 32 possibilities, only answers when correct

$$
\begin{array}{rl}
\text{read (block3)} = & 11011\ \ 00100\ \ 11011 \\
\text{keystream} = & 01010\ \ 01101\ \ \ldots\ldots \oplus \\
\hline
 & 10001\ \ 01001\ \ \ldots\ldots
\end{array}
$$

Source: *Gone in 360 seconds,* Verdult et al.

Security – Ignition Immobilizers

- TMTO attack
  - Build cipher-state/key stream table with 2^{37} entries (~1.2TB)
  - Emulate transponder, get {nR},{aR} from car
  - Replay to transponder, use key stream oracle to get 256 key stream bytes

  - i = 0
    - Is $ks_i, .., ks_{i+47}$ in table? => candidate state, otherwise i++

  - Use rest of ks to confirm valid internal state
  - Use rollback to recover key

  - *Attack time*: 1 min

Technische Universiteit **Eindhoven** University of Technology

Security – Ignition Immobilizers

- <u>Cryptanalytic attack</u>
  - *Cipher design flaws*:
    1. Session dependency:
       - After cipher initialized => only produces key streams
       - Only further randomized by 32-bit nonces
       - 16 persistent bits constant among sessions
    2. Low det. of filter func:
       - 1/4 chance output det. only by first 34 bits of internal state

  - Emulate transponder to get 136 auths from car
  - Guess first 34 bits. Using flaw (2), test first bit of aR. Using flaw (1) repreat many times (136/4 = 34)
  - For each candidate key that passes (~2-3):
    - Exhaustive search for remaining 14 bits

  - *Attack time*: 360 seconds

Real world?

Thieves placed bugs and hacked onboard computers of luxury cars

The leader of a gang that hacked into the onboard computers of luxury cars and bugged them with GPS tracking devices before stealing them is facing jail.

The GPS tracking devices allowed the gang to work out the easiest time and place to steal the car Photo: REX FEATURES

By **Telegraph reporters**
11:53AM BST 02 Jul 2012
Follow 351K followers

Print this article
Share
Facebook
Twitter
Email
LinkedIn 0

Alan Watkins, 42, created false identities for over 150 stolen cars worth up to £3.5m to sell them on in Cyprus. He particularly targeted models of BMWs, Audis and Range Rovers.

Watkins had details of over 500 vehicles and had all the required documentation to create false registrations for over 300 stolen luxury cars - a practice known as 'ringing'.

Thieves Are Using "Mystery Gadgets" To Electronically Unlock Cars And Steal What Is Inside

Michael Snyder
American Dream
January 2, 2014

All over America, criminals are using improvised electronic devices to electronically unlock vehicles and steal whatever they find inside. These "mystery gadgets" reportedly recreate the same signals that the key fobs that so many of us carry around send out.

Image: Key Fobs (Wikimedia Commons).

As you will see below, footage is popping up nationwide of thieves using these "mystery gadgets" to remotely unlock car doors and disable alarm systems. Once a car has been unlocked, it takes these thieves just a few moments to take what they want before leaving without a trace. This is now happening all over the country, and authorities do not know any way to prevent it from happening. For now, the most common piece of advice that police are giving to people is to not leave any valuables inside your vehicle at all.

High tech car theft: 3 minutes to steal keyless BMWs

If you owned a very expensive BMW, how upset would you be to learn that car could be stolen in less than three minutes? Car thieves are exploiting 'features,' then using a BMW on-board diagnostics (OBD) port to clone a key and steal a car.

By Ms. Smith on Sun, 07/08/12 - 4:23pm.

TU/e Technische Universiteit **Eindhoven** University of Technology

The big picture

- **Three major problems:**
  - Legacy
    - Phasing out of broken systems takes years (DST-40: 5 yrs, Hitag2: 4 yrs)
  - Weak, proprietary design
    - Poor cipher design
    - Poor protocol design
    - Proprietary jungle
  - Implementation faults
- **No public research into privacy aspects!**

Questions

Questions?