Report from ESCAR EU – Nov 2018

Tomas Olovsson
Tomas.Olovsson@chalmers.se
Computer Science and Engineering

AutoSec

CHALMERS

Fast, Furious and Insecure: Passive Keyless Entry and Start Systems in Modern Supercars

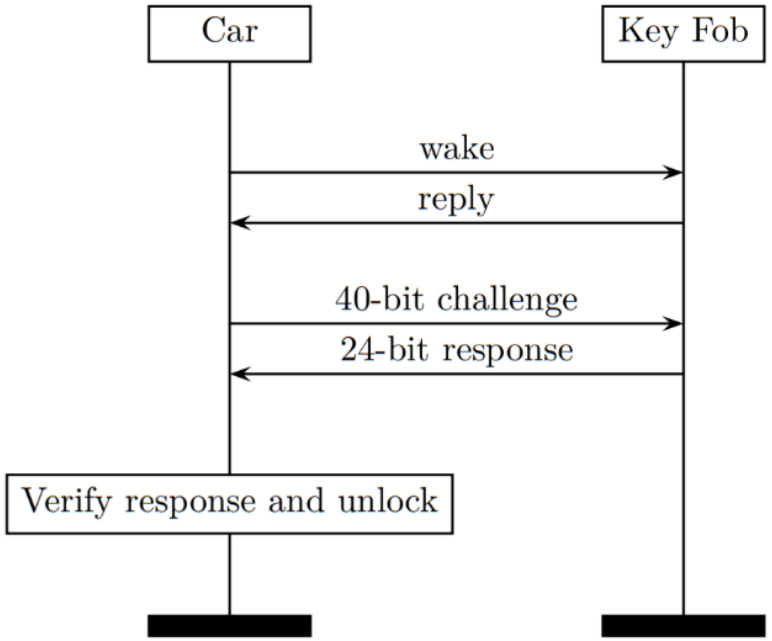
Lennert Wouters: COSIC at KU Leuven

Goal: clone passive keyless cards for Tesla model S

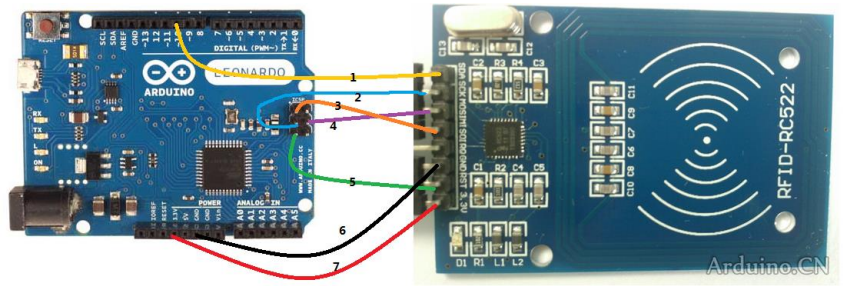
Vehicle periodically broadcasts its ID (2 bytes), and if key nearby, it responds

Reverse engineering revealed algorithms: outdated DST40 which transforms a 40-bit challenge to a 24-bit reply

No authentication between key and vehicle exist, i.e. vehicle cannot know who answers



Cloning the Key card



- The 40-bit challenge 0x636f736963 was selected
- All possible replies were calculated and stored in files
- Due to the 40->24 bit transformation, 65,536 possible keys can generate the same answer
- Another challenge is sent where brute force analysis is done to reveal the key. Takes 2 seconds

Conclusion: Cloning a key card is easy: retrieve 2-byte car ID, get close to the key and send two challenges, record the responses and look up the key! That's it.

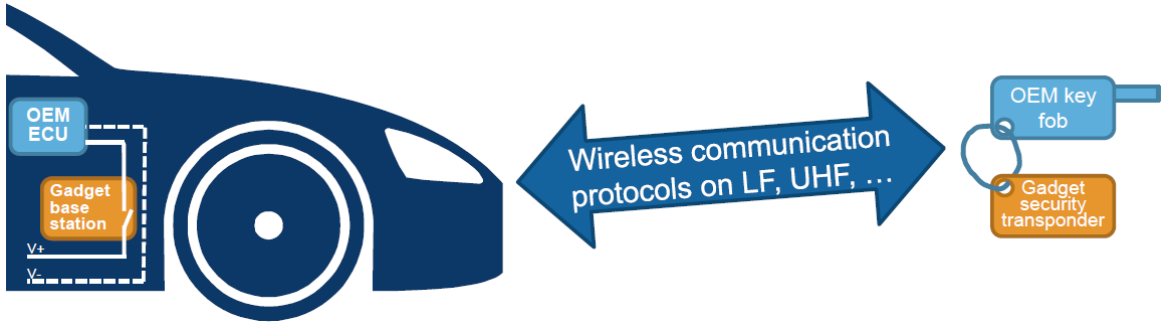
Automotive anti-theft retrofit kits and their Safety and Security implications

Tobias Hoppe: IAV automotive engineering

A new gadget is installed to prevent against theft

After gadget authorizes access, power is turned on to original key fob receiver/ECU

Software defined radio is used to interact with the system, easy and cheap. A DVB USB-stick works well!



AutoSec

Security analysis

The system uses a 20-bit challenge, but turns out only 12 bits are random
Every 2.8 second, a new challenge is issued

- But after 16 messages (45 seconds), the code sequence repeats, so replays are trivial to do
- Relay attacks also possible
- Brute force attacks, i.e. to transmit 2^{12} = 4096 messages takes max 2 minutes and was demonstrated

Takeaways:
Replay protection needed (longer keys, rolling codes or challenge response)
Relay protection also needed but more complicated – not yet state of the art

Real-World Adversarial Attacks on Traffic Signs

Alexander Kreines: Harman

Neural networks are now becoming ubiquitous and used in sign recognition, object detection from sensors, traffic lane detection, etc.

Problem: we don't know how robust they are

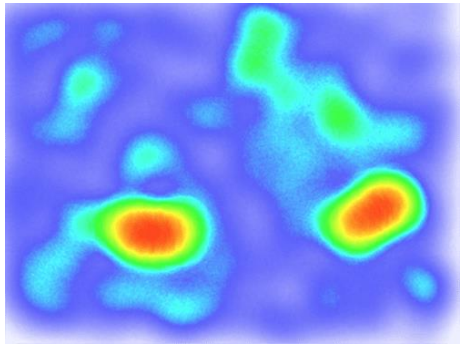
New attacks (neural network spoofing) possible:

- Hard to notice
- Impossible to detect with classical methods
- Easy to implement

AutoSec

Analyzing neural network performance

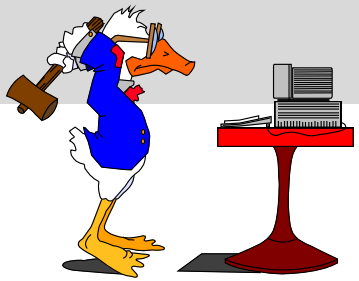
They have a test lab where they build heat maps:



- A dot is inserted in an image to see how sensitive the system is in this area
- Possible to visualize how the system works when analyzing an image (compare with a brain scan)
- Possible to know accuracy and robustness of detections
- Can be used to train system in sensitive areas

There Will Be Glitches: Extracting and Analyzing Automotive Firmware Efficiently

Niek Timmers - Riscure



Inject faults to bypass security: if (authenticated) then ….
by running ECUs outside specified voltages or frequencies, create glitches, …

UDS (Unified diagnostic services), ISO 14229 also useful target

- Often used for “secure” firmware updates
- Demonstrated it is possible to bypass secure boot [Blackhat demo]

Easy to extract firmware – cannot be protected, just a matter of time

- Reverse engineering hard,
- “Tainting” works well!

Takeaway: Use fault injection in your own testing

- And write code that require two or more glitches to fail!

Dangerous Intelligence: Attacks against Machine Learning Systems

Konrad Rieck, TU Braunschweig

Keynote talk about AI and machine learning and how to attack machine learning systems.

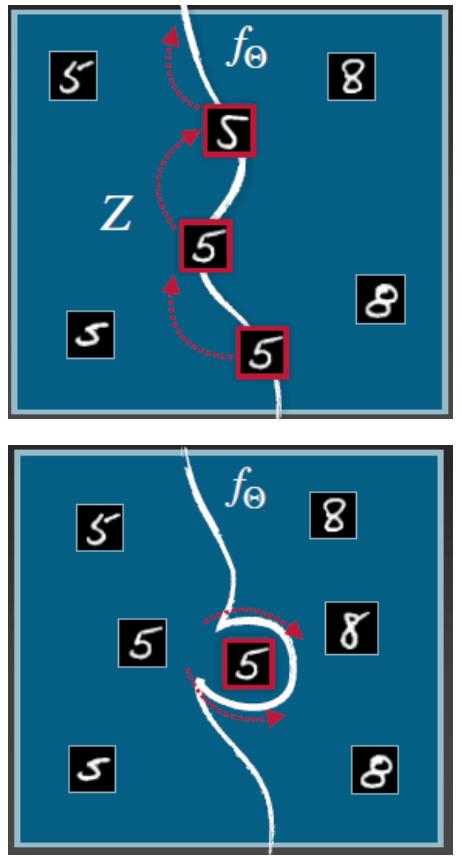
Problem: ML algorithms are good at average-case problems, not worst case problems.

An attacker can manipulate training data: remove or add input, bias data in some direction.

The attacker is interested in the minimum required input change to cause misclassification

Attacking MLS systems

1. By testing different input, the decision border can be reconstructed
2. By poisoning data, the learning process itself is manipulated



Backdoors and unwanted behavior can be the result. Showed an example with an artificial road sign that triggers strong steering to the right.

Defense is hard. Security during training needed and protection against manipulation of training data.

Takeaways

- Machine learning is insecure!
- Algorithms are not smart, learned models not same as human perception and understanding.
- Security research is urgently needed in the machine learning world!

Woodpecker, a Software-only True Random Generator for the CAN Bus

Tsvika Dagan: Tel Aviv University

Proposed a way to use the CAN bus to collect entropy for pseudo random number generation

Inter-arrival times for frames were measured, and results were promising when analyzed

- - -

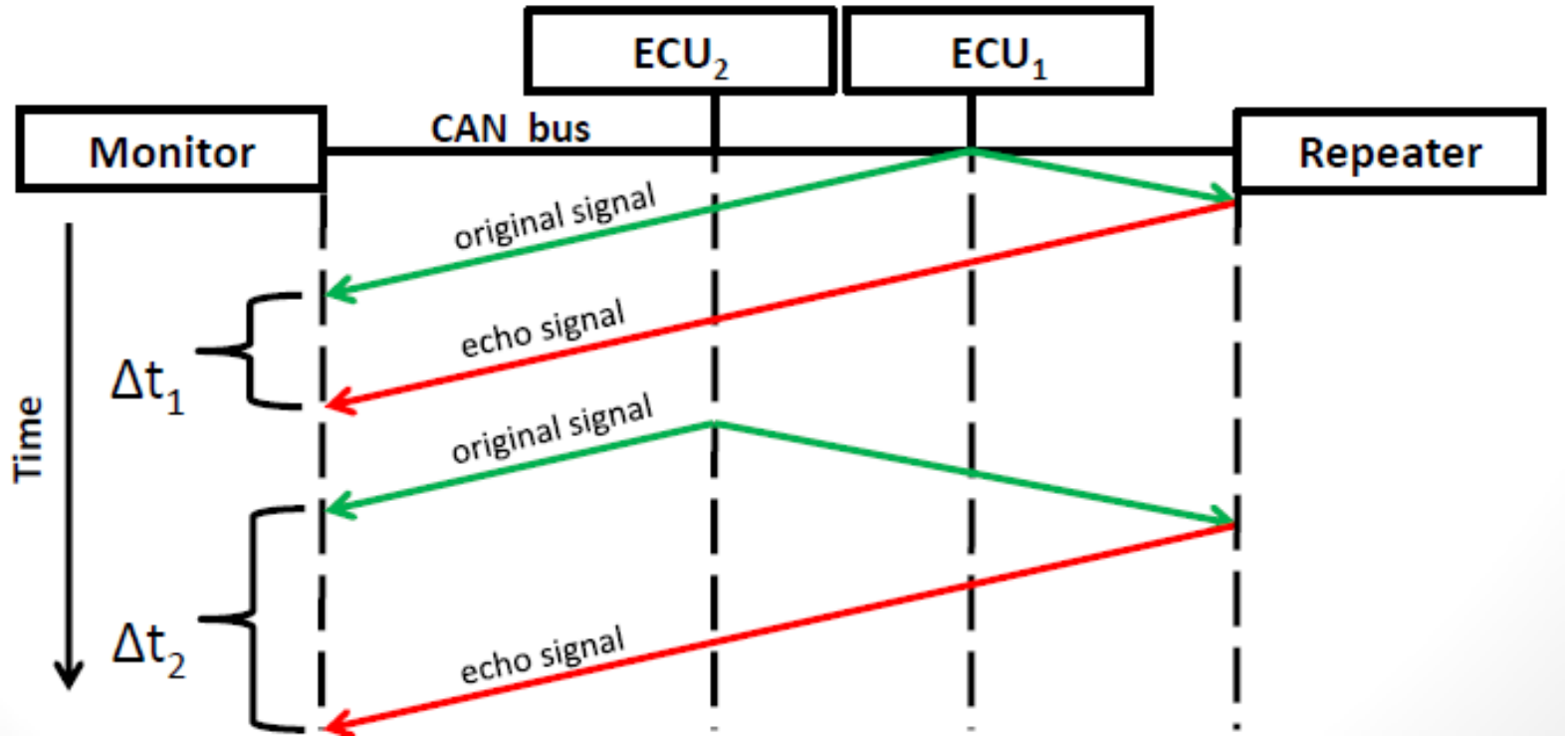
A question from the audience:
wouldn’t a compromised ECU have similar knowledge of the random data and therefore sessions and keys be predictable?

TCAN: Authentication Without Cryptography on a CAN Bus Based on Nodes Location on the Bus

Eli Gavril: Technion, computer science dept., Israel

Goal: knowing the true sender of a message and verifying it has not been tampered with

- Message arrival time – depends on the location of the ECU on the bus
- By measuring arrival time difference, we know the location of the sender
- There is also a weak echo signal when the signal reaches the end of the bus
- A monitor is used which contains an authentication table with legal pairs of location and message type
- If the message pair is illegal, it invalidates the message by transmitting an error frame (overwrites the message)
- They propose an echo signal implementation has a voltage difference which is higher than a regular dominant signal. The monitor can measure the voltage, regular ECUs can't.



ECU-1: msg 1482
ECU-2: msg 1350
msg 1351
ECU-3: …

The authentication table can either be created by the OEM or learnt at the startup of the vehicle, possibly using cryptographic authentication.

TCAN is patent pending. Not yet implemented in a real vehicle.

Shift Left: Fuzzing Earlier in the Automotive Software Development Lifecycle using HIL Systems

Dennis Oka: Synopsys, Ryo Kurach, Nagoya University Japan

Security testing and fuzzing can be integrated into vehicle development

Approach: integrate fuzz testing into the functional testing workflow and do it earlier in the development process

They have built a system with remote-controlled toy cars where acceleration, steering, lights, etc. can be controlled

Uses Autosar and 4 controllers/ECUs control the vehicle using CAN communication

After a 1 hour session with 100,000 fuzzed CAN messages, they found hang-up and freezes and incorrect configurations of the Autosar OS