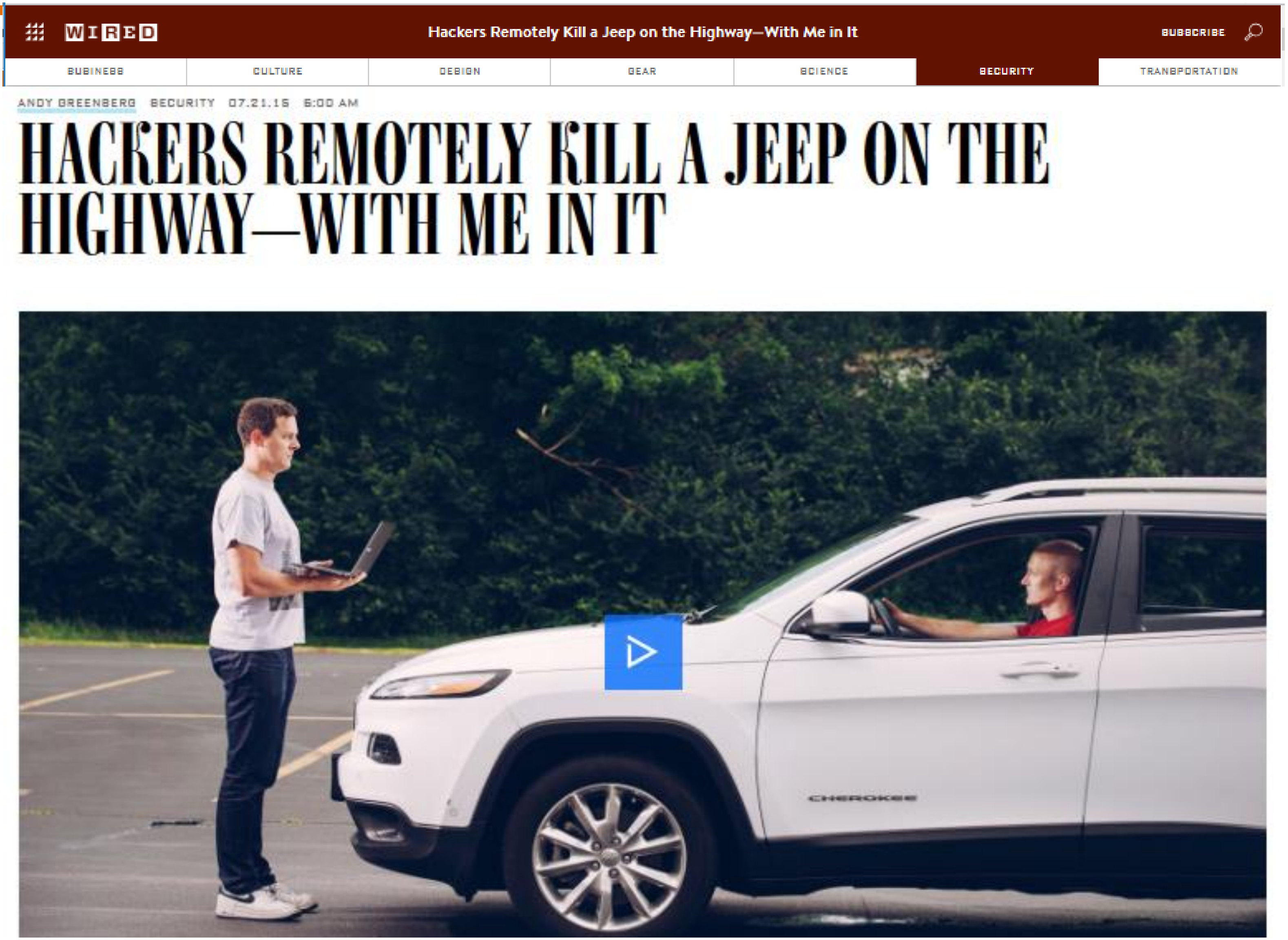
www.crysys.hu

Vehicular Network Security

Levente Buttyán
Tamás Holczer
CrySyS Lab
Budapest University of Technology and Economics
{buttyan,holczer}@crysys.hu

Why vehicular security?



ANDY GREENBERG SECURITY 07.21.15 6:00 AM

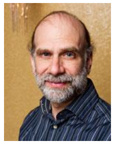
HACKERS REMOTELY KILL A JEEP ON THE HIGHWAY—WITH ME IN IT

Outline

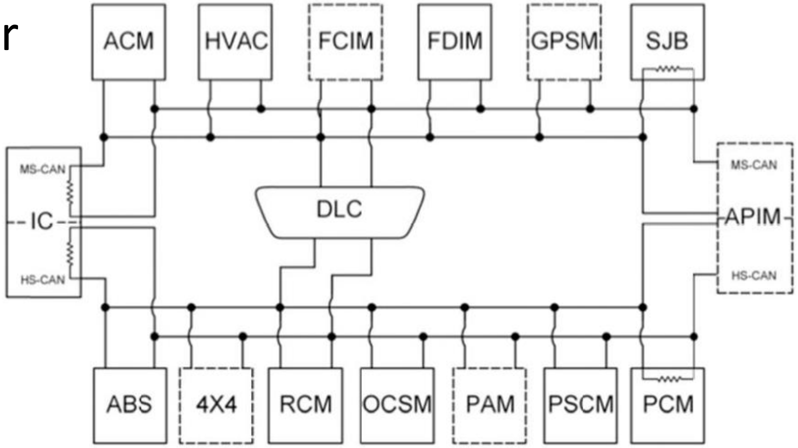
- In vehicle networks
  - Introduction to CAN networks
  - Attacks
  - Countermeasures
- V2X networks

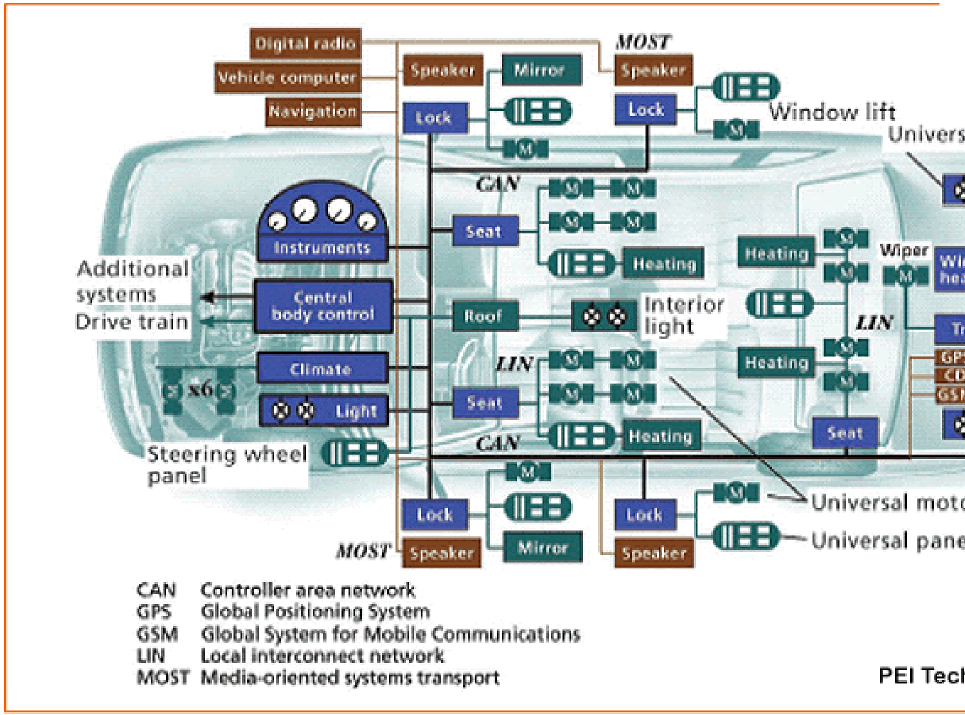
In vehicular networks

”Your car is no longer a mechanical device with some computers inside; it's a computer with four wheels and an engine.”



-- Bruce Schneier, security expert

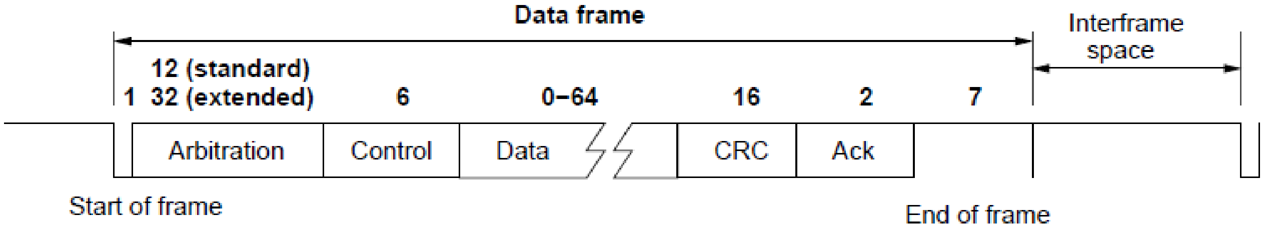




CAN Controller area network
GPS Global Positioning System
GSM Global System for Mobile Communications
LIN Local interconnect network
MOST Media-oriented systems transport

PEI Technologies

CAN Networks



- Identifier (11 or 29 bits from the arbitration field above)
  - determines the set of potential receivers of the frame
- Data (0-8 bytes)
- No source address
- No cryptographic authentication or integrity protection
- CRC is only against faults, not against attackers
- Messages can be injected
- Channel can be overloaded = Denial of Service attack

Adventures in Automotive Networks and Control Units

By Dr. Charlie Miller & Chris Valasek

DIRECT LOCAL ATTACKS

Attacks – Injecting CAN data

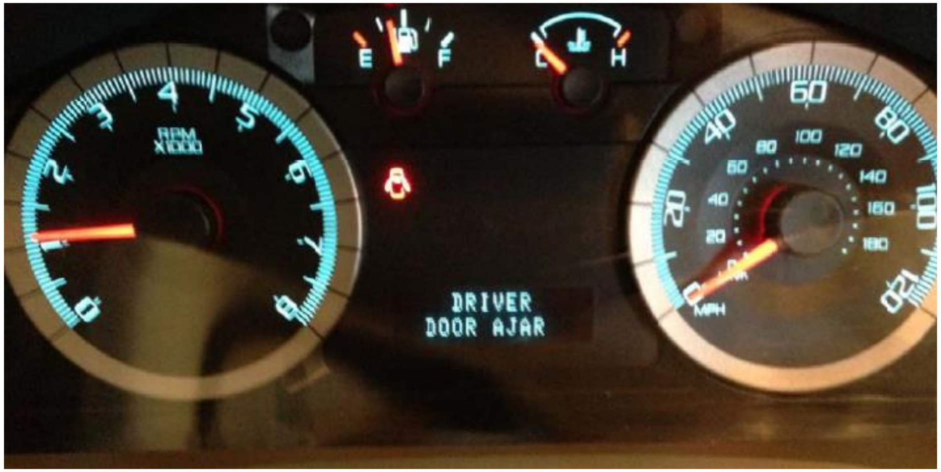
- simple example: door ajar indication on the Ford Escape
  - periodic CAN message sent every 2 seconds
    - when no door is ajar:

```
ID: 3B1, Len: 08, Data: 00 00 00 00 00 00 00 00
```

    - when the driver's door is ajar:

```
ID: 3B1, Len: 08, Data: 80 00 00 00 00 00 00 00
```

  - injecting the second CAN message results in door ajar indication even when the doors are closed:



Pitfalls with injecting CAN data

- not everything can be controlled via the CAN bus
  - e.g., on the Ford Escape, only the cruise control module can control acceleration of the car without the driver pressing the gas pedal, but the controls are wired directly into the PCM (i.e., it is not controlled via CAN)
  - however, the more ECUs the car has, the more likely is that they communicate via the CAN bus (and not direct wiring)

- some CAN packets contain only informative data, and no action will be taken by the car if you change those
  - e.g., in the Ford Escape, a CAN packet with ID 200 can be observed that has a byte indicating how much the accelerator is depressed
  - however, this packet is intended for the ABS and changing the value in it will not accelerate the car
  - it takes a lot of reverse engineering to locate specific packets that are requests from one ECU to another ECU to take action

Attacks on the SecurityAccess protocol

- before performing most diagnostic operations against an ECU, one needs to authenticate himself to the ECU by responding to the challenge received in the SecurityAccess protocol

- sometimes the challenge is static (always the same) ☺
  - sufficient to eavesdrop the correct response once and replay it later

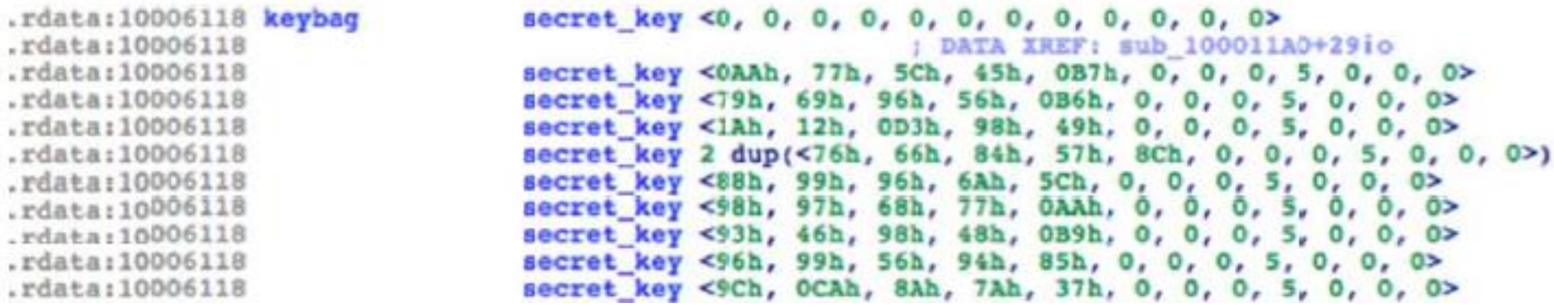
- mostly, however, there’s a new challenge every time you try
  - the attacker needs
    - the algorithm which computes the response
    - the secret key
  - both must be there in the ECU software and in external diagnostic applications that can actually perform those protected functions
  - it is usually ”easier” to reverse engineer the external application, because it is readily available and written for better known (for the typical attacker) platforms (e.g., x86)

Attacks on the SecurityAccess protocol

- bonus: as diagnostic tools can talk to all ECUs and can perform protected functions on them, by reverse engineering them, one gets an entire ”bag” of keys

example:

– a key bag in memory



```
.rdata:10006118 keybag          secret_key <0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0>
.rdata:10006118                                         ; DATA XREF: sub_100011A0+29io
.rdata:10006118                 secret_key <0AAh, 77h, 5Ch, 45h, 0B7h, 0, 0, 0, 5, 0, 0, 0>
.rdata:10006118                 secret_key <79h, 69h, 96h, 56h, 0B6h, 0, 0, 0, 5, 0, 0, 0>
.rdata:10006118                 secret_key <1Ah, 12h, 0D3h, 98h, 49h, 0, 0, 0, 5, 0, 0, 0>
.rdata:10006118                 secret_key 2 dup(<76h, 66h, 84h, 57h, 8Ch, 0, 0, 0, 5, 0, 0, 0>)
.rdata:10006118                 secret_key <88h, 99h, 96h, 6Ah, 5Ch, 0, 0, 0, 5, 0, 0, 0>
.rdata:10006118                 secret_key <98h, 97h, 68h, 77h, 0AAh, 0, 0, 0, 5, 0, 0, 0>
.rdata:10006118                 secret_key <93h, 46h, 98h, 48h, 0B9h, 0, 0, 0, 5, 0, 0, 0>
.rdata:10006118                 secret_key <96h, 99h, 56h, 94h, 85h, 0, 0, 0, 5, 0, 0, 0>
.rdata:10006118                 secret_key <9Ch, 0CAh, 8Ah, 7Ah, 37h, 0, 0, 0, 5, 0, 0, 0>
```

– some of the keys are readable ASCII strings:

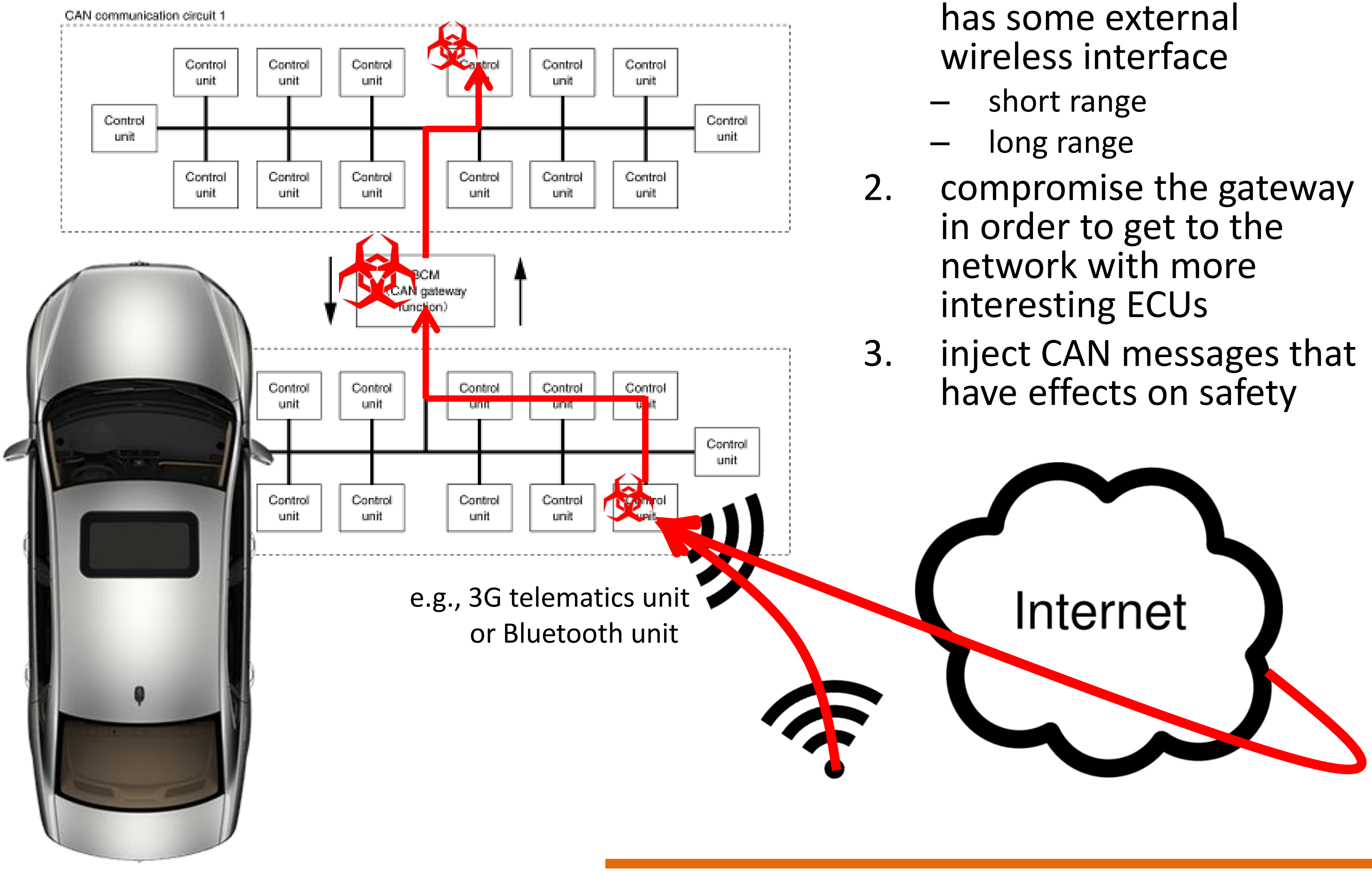
JAMES, MAZDA, MazdA, mAZDa, PANDA, Bosch, a_bad, 12345, Jesus, ...

A Survey of Remote Automotive Attack Surfaces

by Charlie Miller and Chris Valasek

REMOTE ATTACKS

Anatomy of a remote attack



e.g., 3G telematics unit
or Bluetooth unit

1. compromise an ECU that has some external wireless interface
   - short range
   - long range
2. compromise the gateway in order to get to the network with more interesting ECUs
3. inject CAN messages that have effects on safety

Potential attack targets – today

- Park Assist
  - a dedicated ECU takes sensor data and calculates how the steering wheel should be turned to park in a spot
  - the ECU communicates the desired steering wheel position with the steering wheel ECU that then turns the wheel
  - under some conditions (at slow speed), the steering can be turned by sending messages over the automotive network

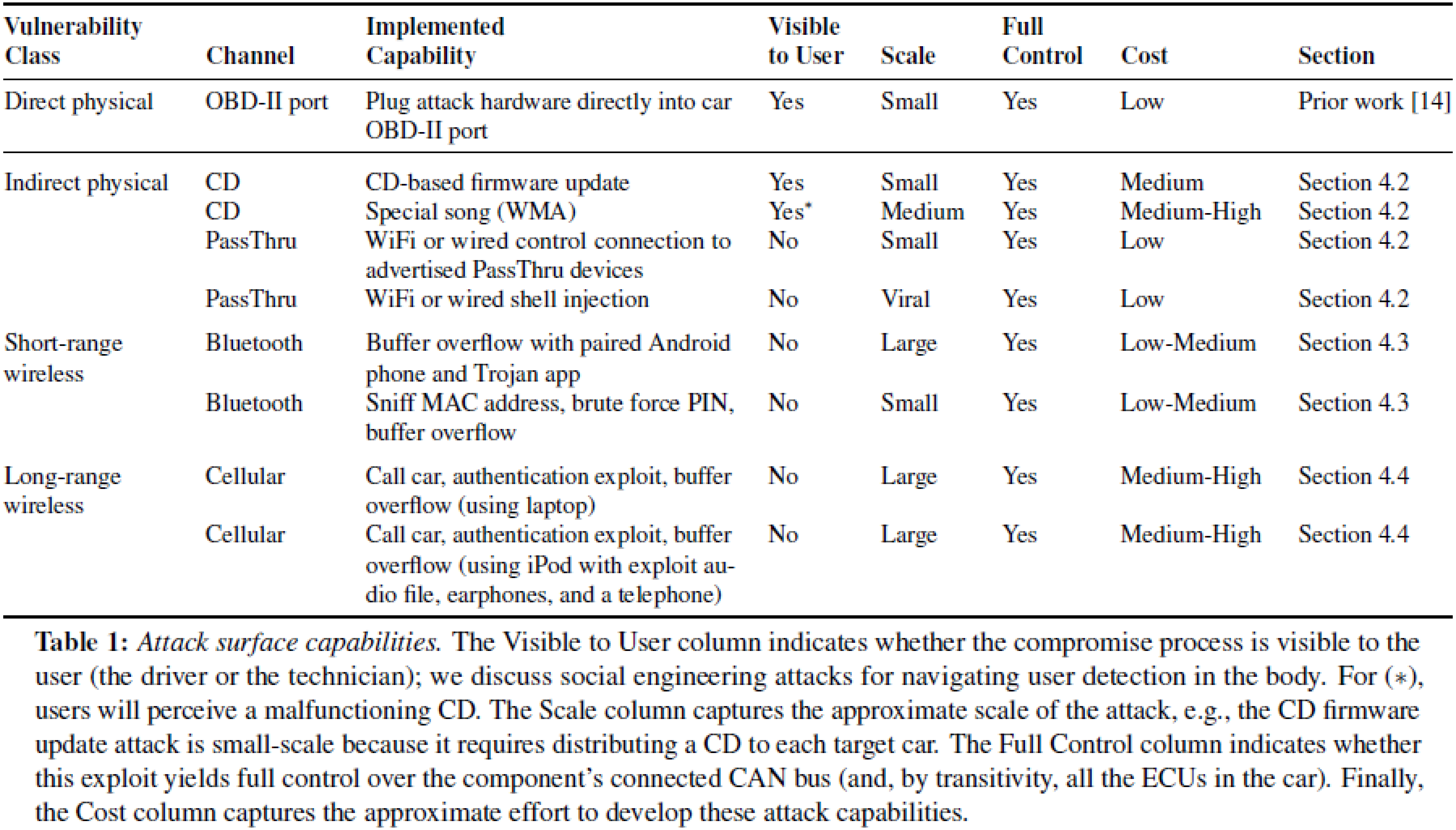
- Adaptive Cruise Control
  - tries to maintain the desired speed of the vehicle even in the presence of other vehicles
    - as the vehicle approaches a slower car, it will apply the brakes to slow down
    - when the slower car gets out of the way, the car will speed up again
  - portions of the vehicle control are performed over the internal vehicle network
  - **designed to work at speed**

Potential attack targets – today

- Collision Prevention
  - a.k.a. crash mitigation, automatic braking
  - designed to prevent or lessen crashes by applying the brakes when a crash is eminent
    - calculations are typically performed by an ECU and messages are sent to the brakes to tell them to engage
  - **designed to work at speed**

- Lane Keep Assist
  - designed to prevent cars from leaving their lane on accident
    - a camera detects the lines of the lane and an ECU computes if the car is about to leave the lane
    - by either sending messages to the steering or brakes, the car is able to adjust its location within the lane
  - **designed to work at speed**

Summary of results*



Vulnerability Class	Channel	Implemented Capability	Visible to User	Scale	Full Control	Cost	Section
Direct physical	OBD-II port	Plug attack hardware directly into car OBD-II port	Yes	Small	Yes	Low	Prior work [14]
Indirect physical	CD	CD-based firmware update	Yes	Small	Yes	Medium	Section 4.2
	CD	Special song (WMA)	Yes*	Medium	Yes	Medium-High	Section 4.2
	PassThru	WiFi or wired control connection to advertised PassThru devices	No	Small	Yes	Low	Section 4.2
	PassThru	WiFi or wired shell injection	No	Viral	Yes	Low	Section 4.2
Short-range wireless	Bluetooth	Buffer overflow with paired Android phone and Trojan app	No	Large	Yes	Low-Medium	Section 4.3
	Bluetooth	Sniff MAC address, brute force PIN, buffer overflow	No	Small	Yes	Low-Medium	Section 4.3
Long-range wireless	Cellular	Call car, authentication exploit, buffer overflow (using laptop)	No	Large	Yes	Medium-High	Section 4.4
	Cellular	Call car, authentication exploit, buffer overflow (using iPod with exploit audio file, earphones, and a telephone)	No	Large	Yes	Medium-High	Section 4.4

Table 1: *Attack surface capabilities.* The Visible to User column indicates whether the compromise process is visible to the user (the driver or the technician); we discuss social engineering attacks for navigating user detection in the body. For (∗), users will perceive a malfunctioning CD. The Scale column captures the approximate scale of the attack, e.g., the CD firmware update attack is small-scale because it requires distributing a CD to each target car. The Full Control column indicates whether this exploit yields full control over the component's connected CAN bus (and, by transitivity, all the ECUs in the car). Finally, the Cost column captures the approximate effort to develop these attack capabilities.

*** Checkoway *et al.*, Comprehensive Experimental Analyses of Automotive Attack Surfaces, Usenix Security Symposium, 2011.**

Countermeasures

- Integrity protection of CAN messages
- Firewalls
- Anomaly detection

CAN security extensions

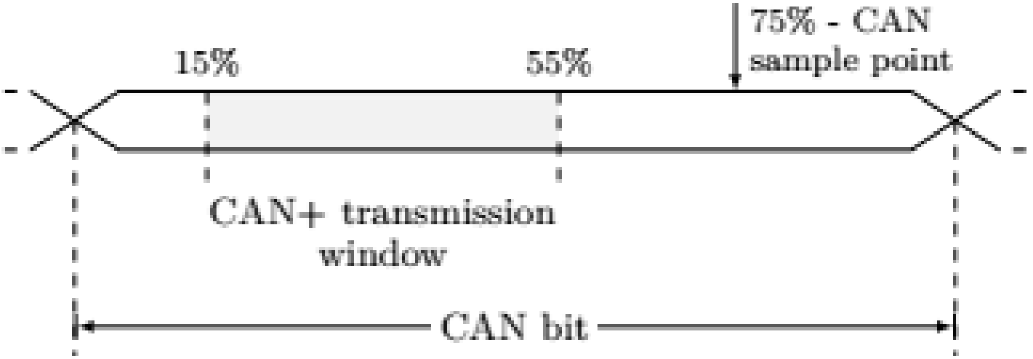
- CAN messages are not authenticated now
- Anyone can inject CAN messages
- Create new version of CAN
- Idea: use cryptography to avoid this!

If you think cryptography is the answer to your problem, then you don't know what your problem is. —Peter G. Neumann

- Requirements
  - Authenticate messages
  - Simple key management
  - Legacy systems should work without modification
  - Should not be expensive
  - Gradual introduction should be possible
  - Should be based on well known crypto

CANAuth*

- Uses CAN+
  - Bit stuffing inside bits before sampling point
  - Requires faster sampler and decoder (15 extra byte / normal byte @ 300 MHz FPGA)
  - Old controllers not affected (backward compatible)

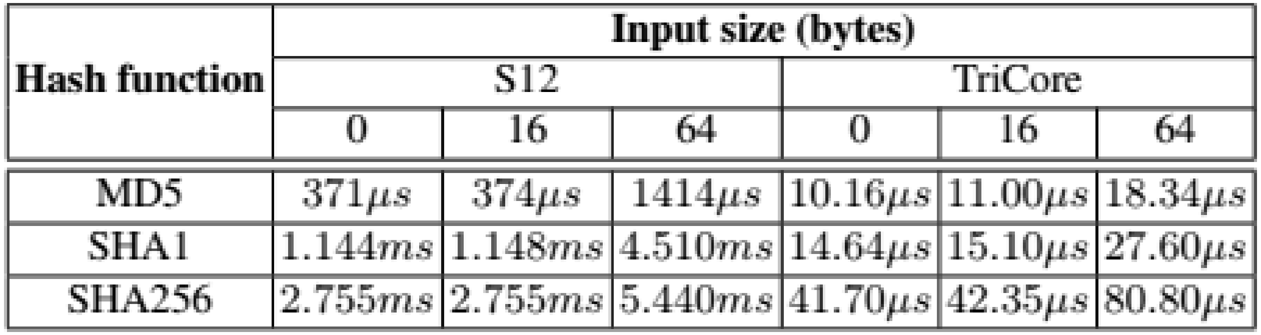


- Symmetric key based message authentication
- 80 bit message authentication code with 32 bit counter
  - Reasonable for short living messages

*Van Herrewege, Anthony, Dave Singelee, and Ingrid Verbauwhede. "CANAuth-a simple, backward compatible broadcast authentication protocol for CAN bus." *ECRYPT* 2011.

LiBrA-CAN*

- Similar to CANAuth
- Based on CAN+ as well (backward compatible)
- Centralized solution, one controller computes more
- Mix several message authentication code into one
- Assumes only minority is compromised
- Can be efficient if controller is strong enough



Hash function	Input size (bytes)					
	S12			TriCore		
	0	16	64	0	16	64
MD5	$371\mu s$	$374\mu s$	$1414\mu s$	$10.16\mu s$	$11.00\mu s$	$18.34\mu s$
SHA1	$1.144ms$	$1.148ms$	$4.510ms$	$14.64\mu s$	$15.10\mu s$	$27.60\mu s$
SHA256	$2.755ms$	$2.755ms$	$5.440ms$	$41.70\mu s$	$42.35\mu s$	$80.80\mu s$

*Groza, Bogdan, et al. "Libra-can: a lightweight broadcast authentication protocol for controller area networks." *ICCNS*, 2012.

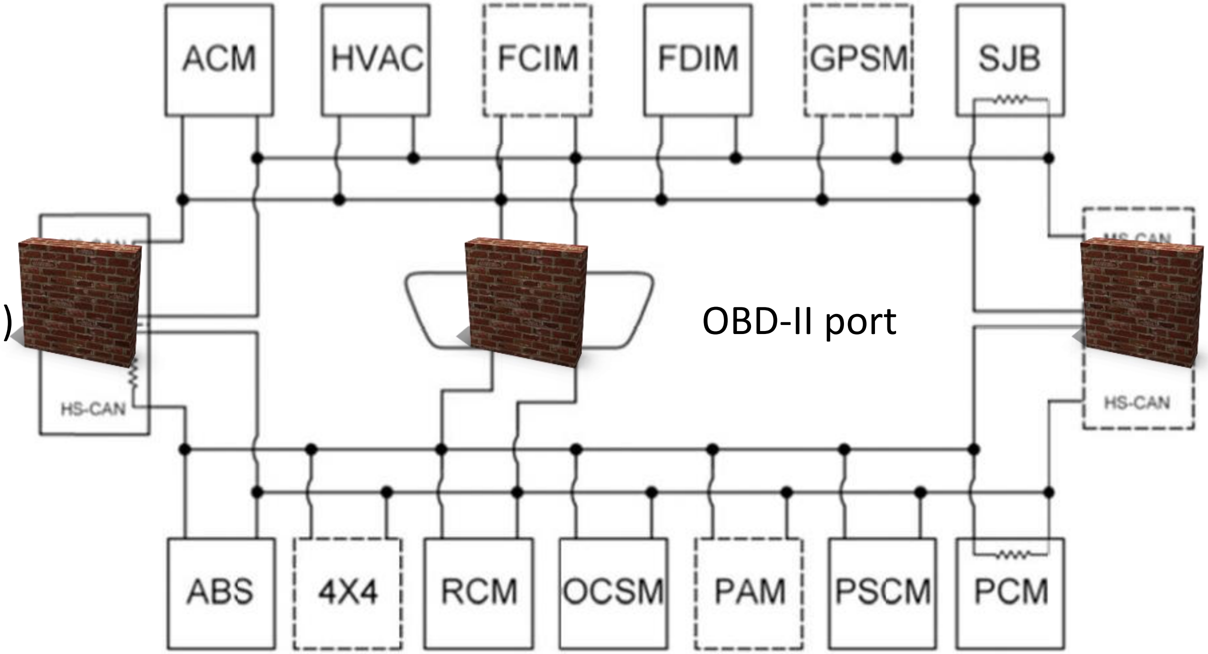
vatiCAN

- Backward compatible
- Based on message authentication codes (MAC)
- Less than 4 ms latency by message checking (acceptable?)
- MAC is sent in separate CAN message with different ID
  - New ID should be close to original ID
- Only critical messages are protected
- Hashed MAC (HMAC) based on SHA-3
- Counters are used to avoid replay attacks
- h=HMAC(ID, message, counter)

Nürnberger, Stefan, and Christian Rossow. "–vatiCAN–Vetted, Authenticated CAN Bus.", 2016.

Where to put the firewall

- CAN bus is broadcast channel → not on the bus itself
- Between segments on bridges or gateways
- Between the internal interface and the bus
  - Integrated into the OBD interface
  - Integrated into ECU



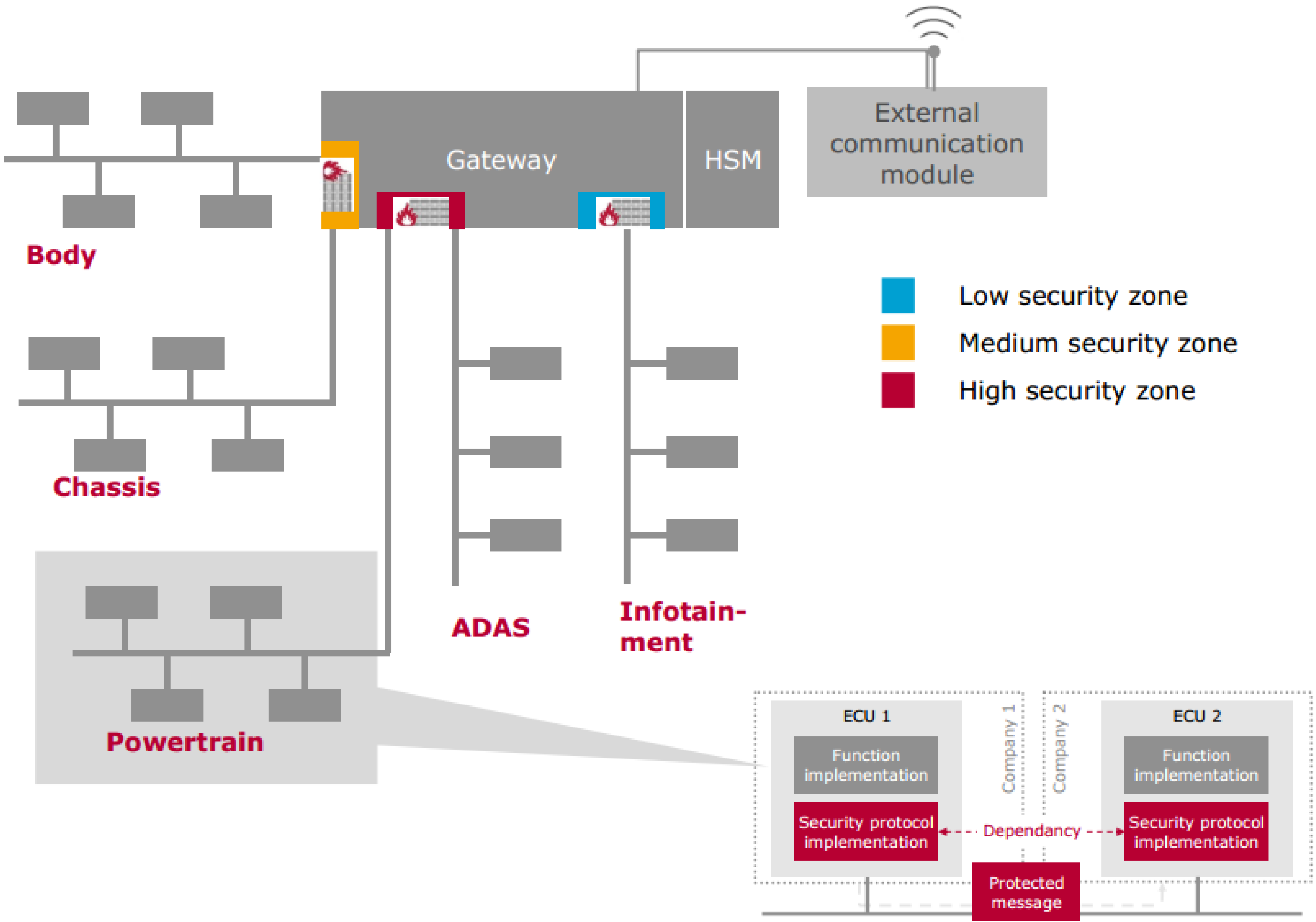
(bridge)

OBD-II port

(bridge)

Where to put the firewall 2.*

Network Strategies



* Armin Happel, Christof Ebert: Security in Vehicle Networks, Vector, 2015

CAN Bus firewall*

- Rule based firewall
  - Whitelist
  - Rate limit
  - Authentication
- No vehicle redesign needed
  - Device with two CAN connector
  - Can be integrated into ECU
- Protection
  - Rate limit against DoS
  - Rule based filter
  - Secure software
- Problems
  - Who writes the rules?
  - How the rules are updated?
  - Who is responsible if important benign packet is dropped causing an accident?

* Arilou: Feasible car cyber defense, ESCAR 2010

Potential rules, some examples

- More critical segment can send to less critical segment
- Less critical segment cannot send to more critical segment
- Filtering based on CAN ID
- Filtering based on CAN Data
- Filtering based on previous messages (statefull filtering)
- Filtering based on message counts per ID (avoid DoS)
- …
- Examples:
  - Id=ID(RPM) AND source segment=infotainment DROP
  - Id=ID(RPM) AND source segment=engine ACCEPT

Challenges in vehicular IDS

- How to update the system regularly?
- Many data sources → complicated decision
  - Sensors
  - ECU internal data
  - Networks
- Detection should be simple to keep the price low
- Central or distributed implementation?
- Real time performance required
  - CAN messages cannot be delayed
  - Security events must be detected **before** an accident occurs
- How to interact with the driver when driving?

- Summary: lot of efforts and questions, no reliable solution yet

Vehicular anomaly sensor types*

- Formality sensor
  - Formal correctness of messages
  - E.g.: header, filed sizes, checksum, delimiters etc.
- Location sensor
  - Is the message allowed on that segment
  - RPM coming from infotainments system?
- Range sensor
  - Payload is in expected range
  - Speed > 300 km/h?
- Frequency sensor
  - Timing behaviour of message is correct?

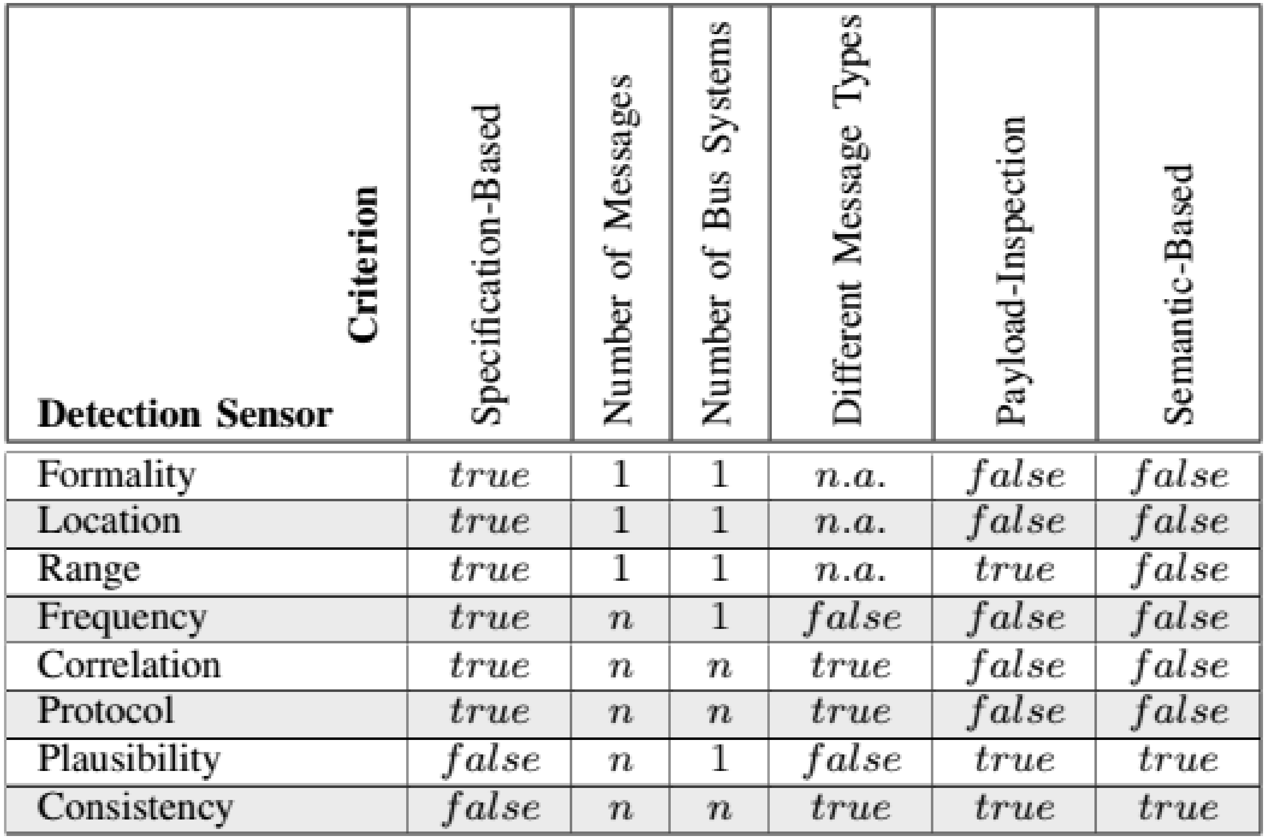
* Mütter, Groll, Freilling: A structured approach to anomaly detection for in-vehicle networks

Vehicular anomaly sensor types*

- Correlation sensor
  - Correlation of messages on different segments is correct?
  - Can detect attackers who has access only to one segment
- Protocol sensor
  - Correct order, start time, challenge response etc.
- Plausibility sensor
  - Checks if payload is plausible
  - E.g. speed changes from 20 km/h to 200 km/h within a second
- Consistency sensor
  - Data from redundant sources is consistent
  - E.g. tire rotation sensor shows standing vehicle while GPS indicates movement

* Mütter, Groll, Freilling: A structured approach to anomaly detection for in-vehicle networks

Properties of different sensors*



Criterion Detection Sensor	Specification-Based	Number of Messages	Number of Bus Systems	Different Message Types	Payload-Inspection	Semantic-Based
Formality	*true*	1	1	*n.a.*	*false*	*false*
Location	*true*	1	1	*n.a.*	*false*	*false*
Range	*true*	1	1	*n.a.*	*true*	*false*
Frequency	*true*	*n*	1	*false*	*false*	*false*
Correlation	*true*	*n*	*n*	*true*	*false*	*false*
Protocol	*true*	*n*	*n*	*true*	*false*	*false*
Plausibility	*false*	*n*	1	*false*	*true*	*true*
Consistency	*false*	*n*	*n*	*true*	*true*	*true*

- Specification based: comes from formal spec. (CAN-Matrix)

* Mütter, Groll, Freilling: A structured approach to anomaly detection for in-vehicle networks

Entropy based anomaly detection*

- Basic assumption: CAN traffic is regular not random
- Measure the randomness of traffic
- Randomness is measured by entropy in information theory

$$H(X) = \sum_{x \in C_X} P(x) log \frac{1}{P(x)}$$

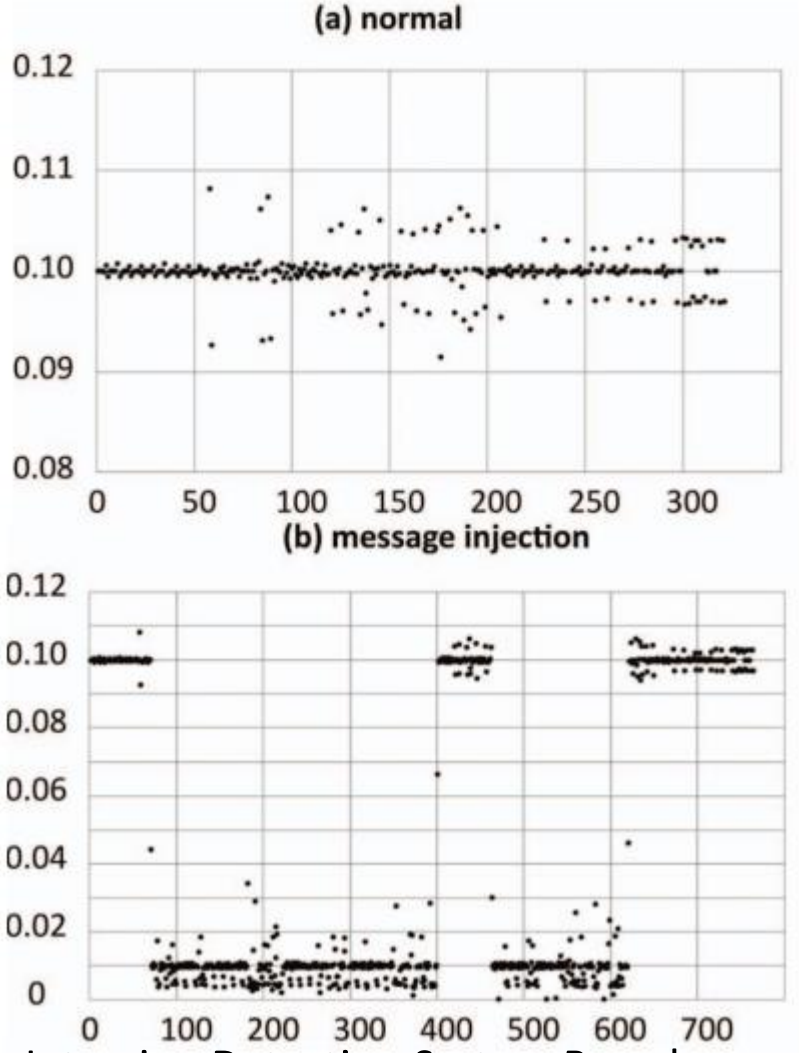
- X is event that ID=X was detected
- P(X) is probability of X (in practice in a measurement window)
- Attack changes H(x)
  - Random injection increases
  - DoS decreases
- Conditional self-information and relative entropy based measures are also defined

* Michael Müter, Naim Asaj: Entropy-Based Anomaly Detection for In-Vehicle Networks, IVS 2011

Time interval based anomaly detection*

- Basic assumption: CAN traffic is regular not random
- Measure the time intervals of each ID
- Can be good against message injections

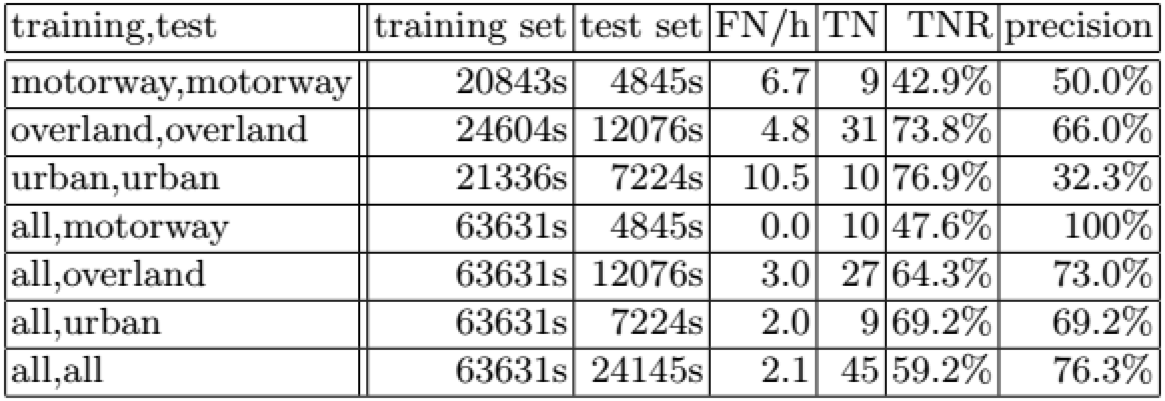
- Caveat: we tried, not every ID is so regular in real life, many false alerts while driving (hope we were not under attack ☺)



* Hyun Min Song, Ha Rang Kim and Huy Kang Kim: Intrusion Detection System Based on the Analysis of Time Intervals of CAN Messages for In-Vehicle Network, 2016

SVM based anomaly detection*

- Learn normal behaviour
- Measure actual distance from normal
- Basic idea: find a minimal sphere which contains all normal events
  - Find c centre and R radius
  - Where R^2 is minimal
  - ||xi-a||^2<=R^2, x1..xn are the events
- Use some tricks (multivariate, non spherical boundary etc.)
- Lot of FN=false alert
- High missed rate



training,test	training set	test set	FN/h	TN	TNR	precision
motorway,motorway	20843s	4845s	6.7	9	42.9%	50.0%
overland,overland	24604s	12076s	4.8	31	73.8%	66.0%
urban,urban	21336s	7224s	10.5	10	76.9%	32.3%
all,motorway	63631s	4845s	0.0	10	47.6%	100%
all,overland	63631s	12076s	3.0	27	64.3%	73.0%
all,urban	63631s	7224s	2.0	9	69.2%	69.2%
all,all	63631s	24145s	2.1	45	59.2%	76.3%

* Andreas Theissler: Anomaly detection in recordings from in-vehicle networks, BIGDAP 2014

Neural network based anomaly detetion

- Neural networks mimic human brain
- Consist of network of simple deciders
- Must learn normal and abnormal behaviour
- Very preliminary phase, but interesting and promising approach

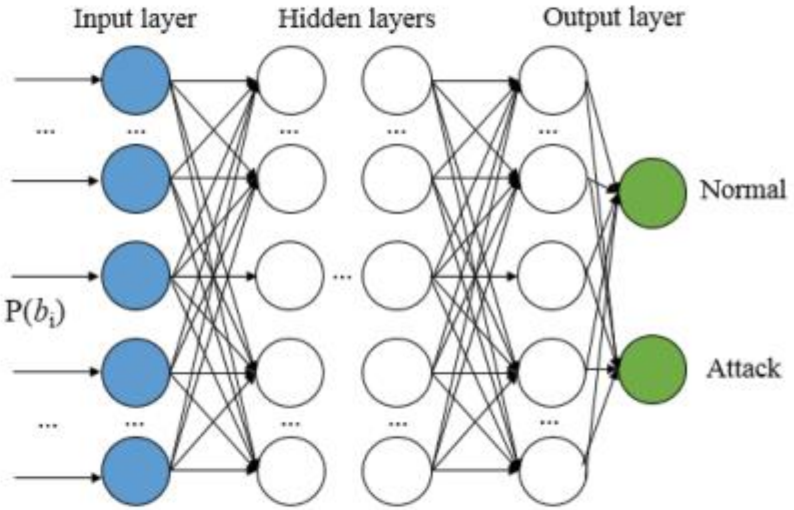
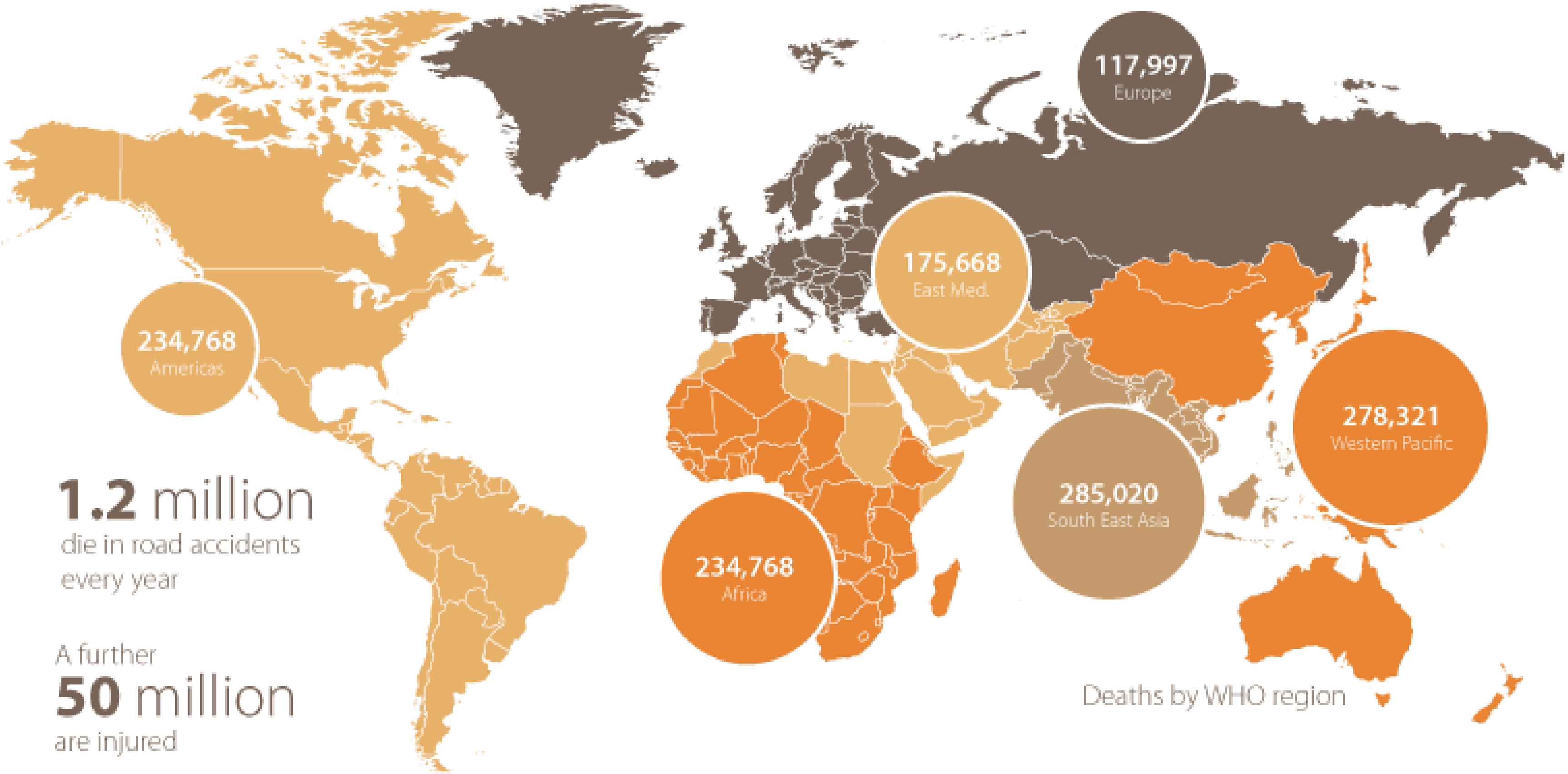


Figure: Min-Joo Kang, Je-Won Kang Intrusion Detection System Using Deep Neural Network for In-Vehicle Network Security

Motivation for V2X communications

Road Traffic Accidents: The Modern Killer

The Global Status Report released by WHO this year, confirms that road traffic injuries are still a big global health and development problem



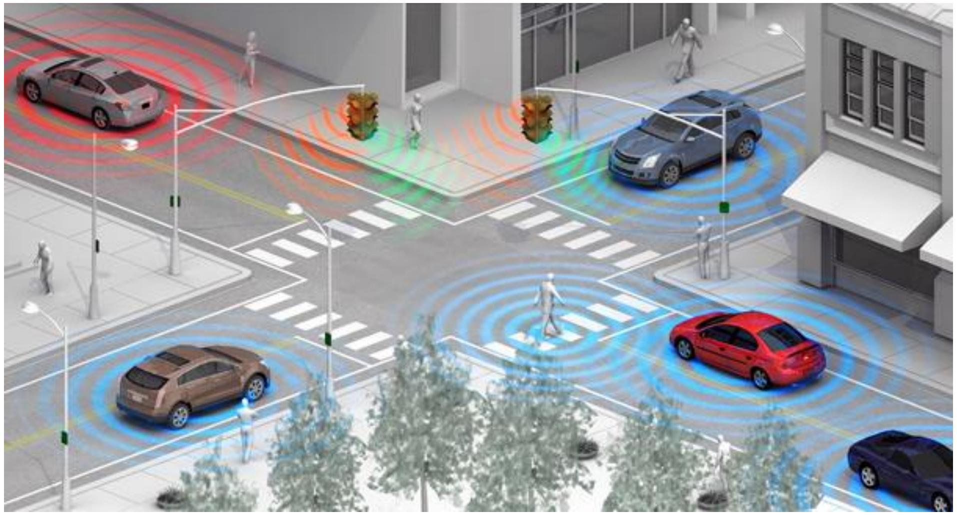
1.2 million
die in road accidents
every year

A further
50 million
are injured

Deaths by WHO region

Motivation for V2X communications

V2X communications promise safer and more efficient driving via ensuring that the right information is available at the right time at the right place



http://www.pcworld.com/article/261623/the_latest_and_best_in_car_tech.html

http://www.haptic.ro/greater-safety-comfort-thanks-v2x-communication-haptic-feedback-display/

V2X communications

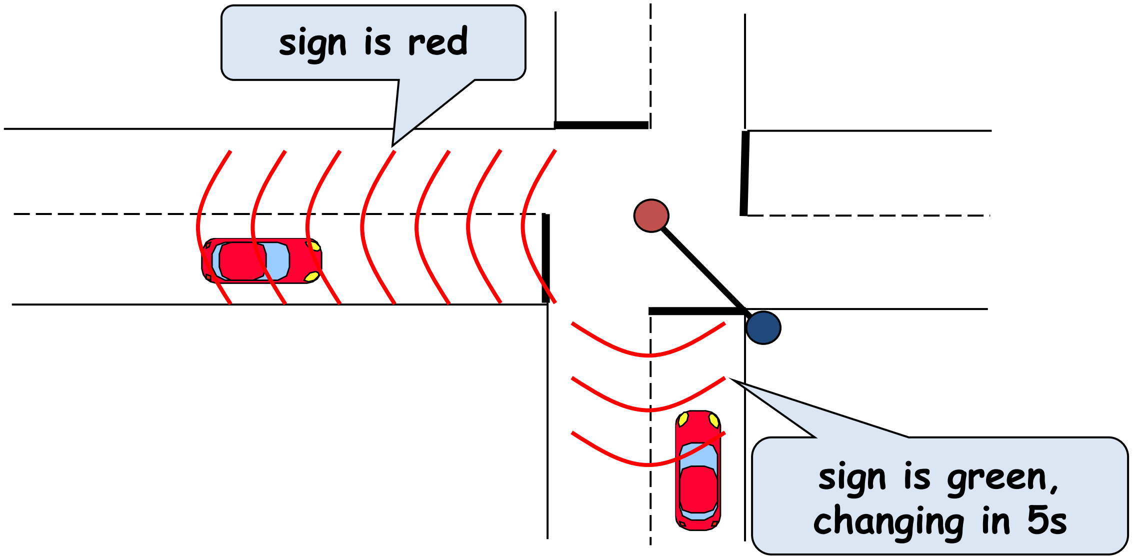
- allow vehicles to communicate with each other and with infrastructure elements and form vehicular networks and connect to the Internet infrastructure

- network nodes
  - road side units (RSU): network nodes embedded in road side infrastructure (e.g., traffic lights, road signs)
  - on-board units (OBU): network nodes embedded in vehicles

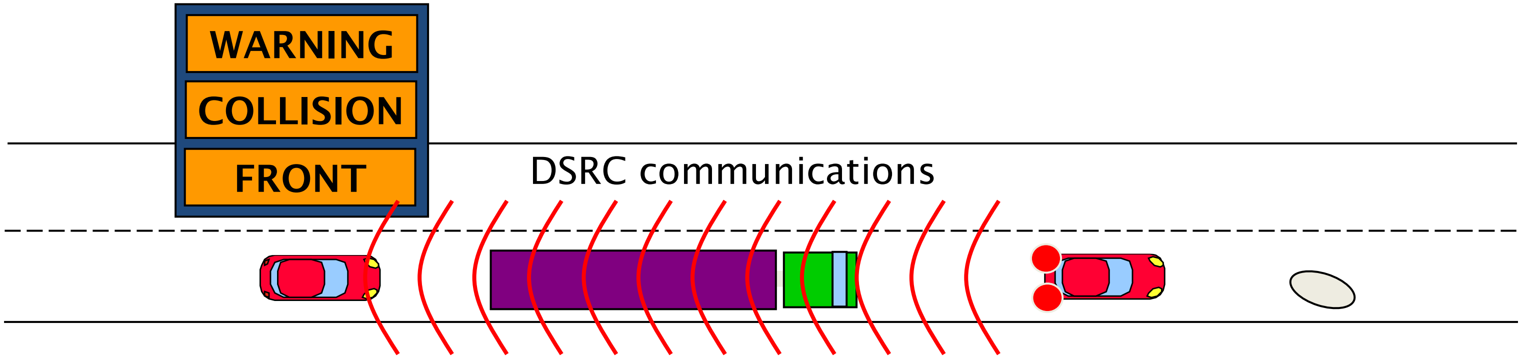
- both types of network nodes consist of
  - general purpose processor and associated memory
  - a radio transmitter and receiver
  - interfaces to sensors as required
  - a GPS receiver (for non-stationary units)

Example applications

- traffic signal violation warning



- extended break lights



Security and privacy requirements

- **correctness of message content**
  - the receiver of a message should obtain an accurate picture of the state of the world, as far as the transmitter knew it
- **message integrity and origin authentication**
  - in order to protect against spoofing and modification attacks
- **privacy**
  - owners of personal vehicles have the right for privacy
  - messages and security services should not leak personal, identifying, or linkable information
  - e.g., it should be difficult to track the location of vehicles
- **robustness**
  - we must assume that some units will be compromised; how to revoke them?
  - denial of service by jamming wireless communications will always be possible...

Challenges

- **real-time requirements**
  - safety applications are usually time-critical
  - the processing and bandwidth overhead due to security must be kept to a minimum in order to improve responsiveness and decrease the likelihood of packet loss
- **scalability**
  - many applications should be available to all vehicles on the road
  - security mechanisms must be as flexible and scalable as possible
- **authenticity vs. privacy**
  - authentication of messages is crucial, as otherwise anybody can inject fake information into the system
  - authentication = proving an identity (e.g., that of the message origin)
  - privacy often requires hiding identities
  - there seems to be a conflict here!

A first attempt for securing V2X comm

- let’s assume that there is a globally shared symmetric key in each unit

- features
  - message integrity and origin authentication can be based on a symmetric key MAC computation
    - only group membership is authenticated
    - RSUs and public safety OBUs cannot be differentiated from regular OBUs
  - perfect privacy
    - no one really knows who sent a message
  - no robustness
    - entire system can be compromised by breaking a single unit
  - no correctness
    - compromised units can send false information
    - compromised units cannot be reliably identified and revoked

A better security architecture

- system should be based on public key cryptography
- general message structure:

  [header | payload | timestamp | position | key ID | signature]

  where key ID is a certificate or a key index
- design questions
  - What PKI structure to use?
  - Which signature algorithm to use?
- privacy requirements
  - regular OBUs need privacy protection
  - RSUs and public safety OBUs do not need privacy protection

RSUs and public safety OBUs

- no need for privacy → straightforward PKI-style solution
- PKI structure can be ...
  - hierarchical
    - mirrors the naming and administrative hierarchy
    - imposes burden on the OBU (signatures should be verified for all certificates in the certificate chains and on related CRLs)
    - deployment requires each of the superior organizations to be operational prior to a subordinate entity
  - flat
    - single CA, but administrative hierarchy is kept
    - needs RAs for convenience
    - reduces the signature verification burden on OBUs
    - size of single CRL may be too long (but there exists optimization techniques such as partitioned, indirect, and delta CRLs)
    - deployment requires a national CA to be operational
    - the national CA must be highly available

RSUs and public safety OBUs

- certificate structure
  - X509 certificates are too large
  - certificates should be optimized and contain only
    - the public key of the certificate holder
    - the scope of the certificate (geographic or functional)
    - the validity window of the certificate (expiry time)
    - a signature over the certificate
  - no identity string in the certificate, all relevant authorization information is in the scope field

- processing
  - standard signature generation / verification
  - scope verification
  - OBUs are expected to cache verified certificates, thus reducing the burden of re-verification for new certificates
  - certificate pre-loading (e.g., at the border of geographic regions)
  - CRL distribution

Regular OBUs

- privacy protection is a requirement
  - anonymity – it should not be possible to determine a vehicle's identity from its transmissions
  - unlinkability – it should not be possible to determine that multiple transmissions originated from the same source

- approaches
  - anonymous certificates
  - group signatures
  - (anonymous self-enforcing certificates)
  - (static combinatoric schemes)
  - (dynamic combinatoric schemes)

Anonymous certs – a naïve solution

- each OBU has its own key pair certified using a PKI

- anonymous certificates
  - public key
  - validity period
  - geographic scope
  - identity of the signer
  - signature

- doesn’t protect privacy
  - each message may contain (a reference to) the certificate of the signer
  - the public key in the cert can be a unique identifier
  - messages signed by the same OBU can be linked through it

Anonymous certs – a better solution

- issue a set of anonymous certificates to each OBU
- OBUs would change their active certificate periodically (e.g., every hour)
- CA needs to maintain a list of which certificate has been issued to which OBU (for revocation and law enforcement purposes)
- OBUs should be able to refill their anonymous certificate set (e.g., at traffic lights or at gas stations using a high speed connection)
- improved privacy, but …
  - limited protection against insider attackers
  - changing pseudonyms is an effective mechanism only if the adversary's observational capabilities are limited (e.g., no global eavesdropping)

Group signatures

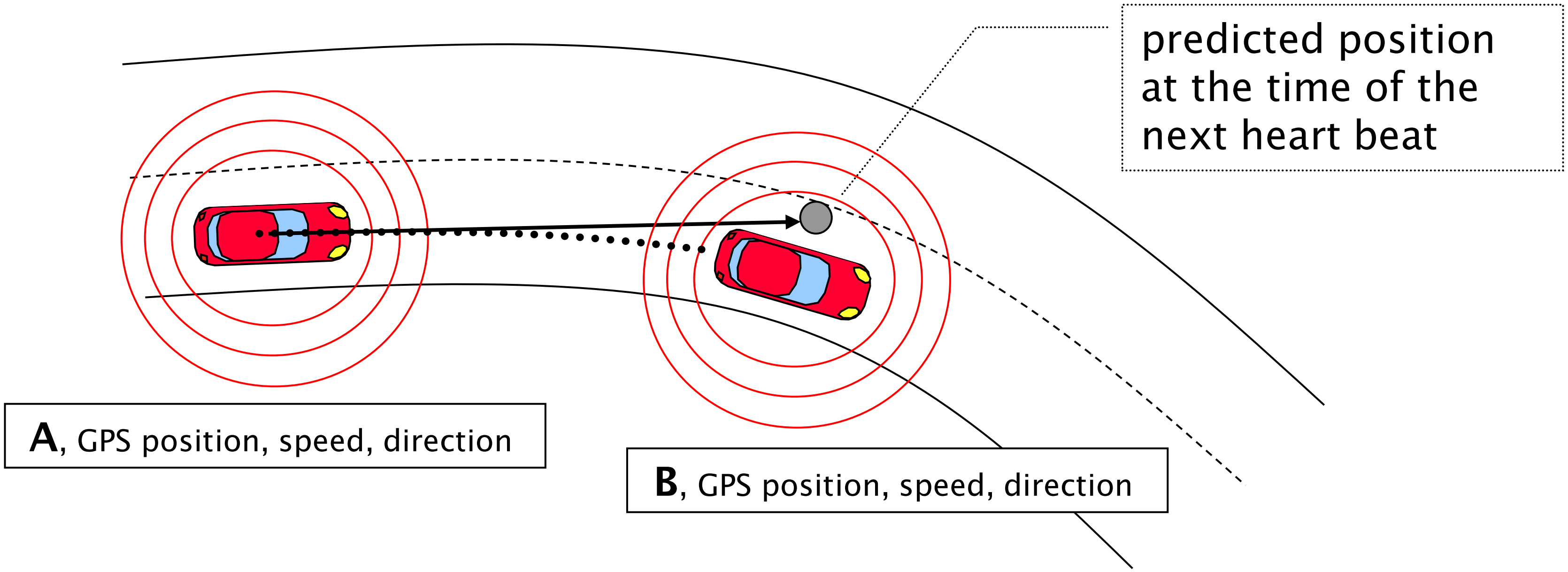
- operation
  - a group signature scheme has a single public key and a large number of private keys
  - a signature that is generated with any of the private keys can be verified with the public key
  - verifier learns only that the message was signed by a member of the group, but cannot tell which member
- all vehicles from the same country can form a group
- elegant but not very efficient yet
- could possibly be combined with anonymous certificates
  - vehicles can use a group signature scheme to issue pseudonyms for themselves (this would be done by a trusted hardware security module (HSM) in each vehicle)
  - a receiver may receive several messages signed under the same pseudonym (within the lifetime of a pseudonym), but needs to verify the group signature on corresponding certificate only once

  → efficiency of standard pseudonyms is retained

  → problem of running out of pseudonyms is eliminated

Changing pseudonyms

- changing pseudonyms can be ineffective if done in a naive way


predicted position at the time of the next heart beat
A, GPS position, speed, direction
B, GPS position, speed, direction

- pseudonyms should be changed by multiple vehicles in the same location in a coordinated way
  - designated *mix zones*
  - pseudonym change below a certain speed (intersection become mix zones!)

References

- S. Checkoway et al., Comprehensive Experimental Analyses of Automotive Attack Surfaces, Usenix Security Symposium, 2011.
  http://www.autosec.org/pubs/cars-usenixsec2011.pdf

- C. Miller, C. Valasek, Adventures in Automotive Networks and Control Units, Technical White Paper, IOActive, 2014.
  http://www.ioactive.com/pdfs/IOActive_Adventures_in_Automotive_Networks_and_Control_Units.pdf
  https://www.youtube.com/watch?v=n70hIu9lcYo

- C. Miller, C. Valasek, A Survey of Remote Automotive Attack Surfaces, Technical White Paper, IOActive, 2014.
  http://illmatics.com/remote%20attack%20surfaces.pdf
  https://www.youtube.com/watch?v=MAGacjNw0Sw

- A. Greenberg, Hackers Remotely Kill a Jeep on the Highway - With Me in It, Wired Magazin, July 21, 2015.
  https://www.wired.com/2015/07/hackers-remotely-kill-jeep-highway/
  https://www.youtube.com/watch?v=MK0SrxBC1xs

- A. Szijj, L. Buttyán, Zs. Szalay, Hacking cars in the style of Stuxnet, Hacktivity, 2015.
  http://www.hit.bme.hu/~buttyan/publications/carhacking-Hacktivity-2015.pdf
  https://www.youtube.com/watch?v=5UCsKQjB6ZE

- See other references on slides

Control questions

- What are the main characteristics of CAN networks?
- What remote attack surfaces exist in vehicles?
- What are the two main approaches in realizing integrity protection in CAN networks?
- What are the ideas behind intrusion detection approaches in CAN networks?
- Why regular PKI is not enough in V2X communication?