RIDAS: Real-time identification of attack sources on controller area networks

Jiwoo Shin*
Soongsil University
sy9254@gmail.com

Hyunghoon Kim*
Soongsil University
axolotl0210@gmail.com

Seyoung Lee
Korea University
seyoung0131@korea.ac.kr

Wonsuk Choi
Korea University
wonsuk85.choi@gmail.com

Dong Hoon Lee
Korea University
donghlee@korea.ac.kr

Hyo Jin Jo
Soongsil University
hyojinjo86@gmail.com

USENIX Security 2023, 9-11 August 2023

Soongsil University KOREA UNIVERSITY

* Co-first Authors

Outline

- Motivation
- Background
- Our Method
- Evaluation
- Discussion
- Conclusion

In-vehicle Communication System

Motivation

- ECU (Electronic Control Unit)
  - A small device in a vehicle's body that is responsible for controlling a driving-related function
- CAN (Controller Area Network)
  - In-vehicle network designed to communicate between ECUs
  - ISO 11898
  - Broadcasting
  - **No data encryption**
  - **No sender/receiver authentication**

→ Security is needed for CAN

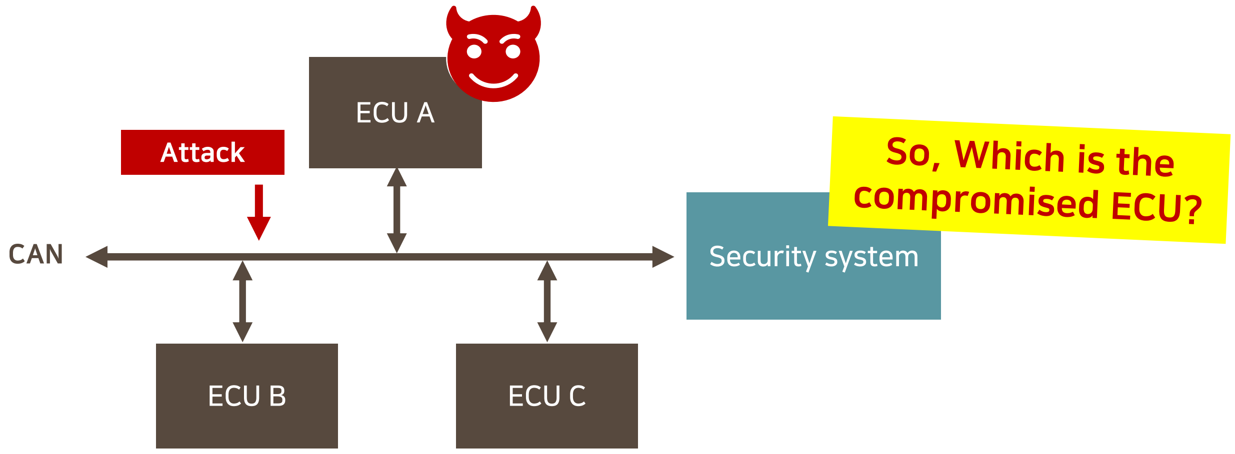
Jeep Cherokee hacking

Tesla Model S hacking

State-of-the-art

- Intrusion detection system (IDS)
  - Lots of Rule-based or AI-based methods have been proposed

    → **Attack detection only, attack source cannot be identified**



State-of-the-art

Motivation

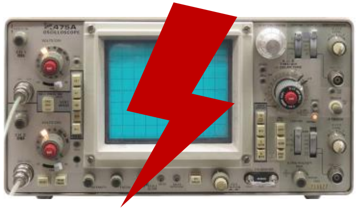
- Intrusion detection system (IDS)
  - Able to identify the compromised ECU
    - **The ECU's clock skewness-based method proposed in [1]**

      → The ECU's clock skew was found to **be corrupted by modifying the timing of transmitted messages [2]**

    - **The ECU's physical layer signal-based method proposed in [3,4]**

      → **Need such a type of electronic test instrument** that measures voltage signals

      → In addition, this device **cannot identify the attack sources with 100% accuracy** due to environmental factors such as its battery level, humidity, etc.



[1] K.-T. Cho and K. G. Shin, "Fingerprinting Electronic Control Units for Vehicle Intrusion Detection." (USENIX Security 16)
[2] Sagong, Sang Uk, et al. "Cloaking the clock: Emulating clock skew in controller area networks." (ICCPS 2018)
[3] W. Choi, H. J. Jo, "Identifying ECUs through Inimitable Characteristics of Signals in Controller Area Networks." (IEEE TVT 2018).
[4] W. Choi, H. J. Jo, "VoltageIDS: Low-Level Communication Characteristics for Automotive Intrusion Detection System." (IEEE TIFS 2018)

Contributions

- Proposal of **a novel real-time attack node identification method, called RIDAS**
  - Using the error handling rule of CAN
- Proposal of a methodology that deals with **RIDAS-aware attackers**
- Evaluation of RIDAS on **a CAN bus prototype** and **a real vehicle**

Outline

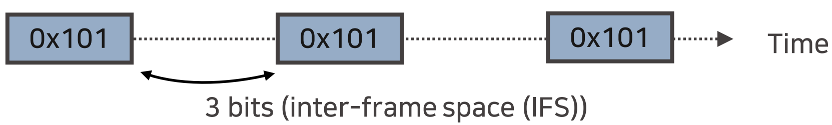
- Motivation
- Background
- Our Method
- Evaluation
- Discussion
- Conclusion

Controller Area Network (CAN)

Background

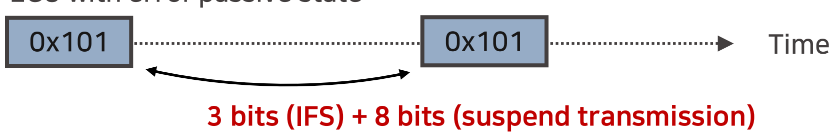
- Error handling and fault confinement
  - ECU has two registers: TEC, REC
  - ECU's error state: Active, Passive, Bus-off
    - The active state: default

ECU with error active state



- **The passive state**: penalty in message sending

ECU with error passive state



- The bus-off state: blocked from the network

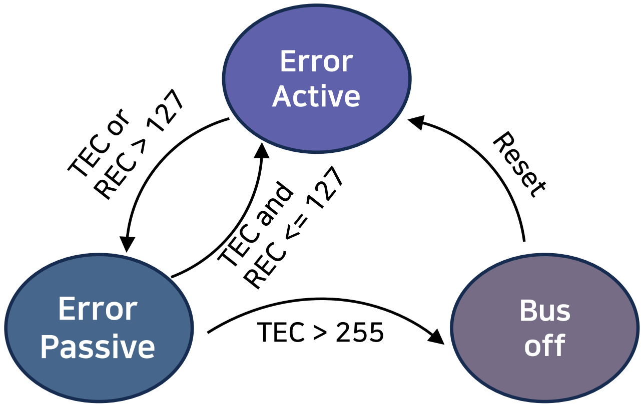
ECU with bus-off state

Active Error Flag 6 bits (dominant)	Error Delimiter 8 bits (recessive)

Active Error Frame

Passive Error Flag 6 bits (recessive)	Error Delimiter 8 bits (recessive)

Passive Error Frame

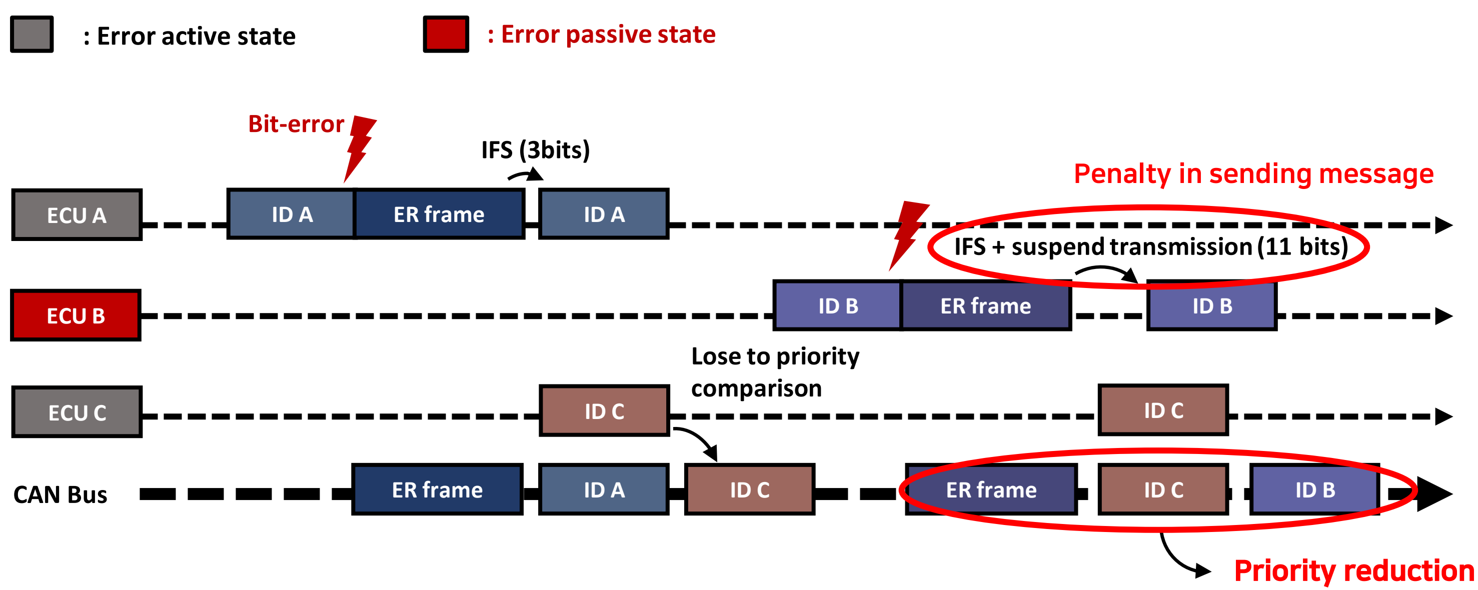


TEC: Transmission Error Counter, REC: Receive Error Counter

Priority Reduction

Background

- The occurrence only in an error passive state
  - It is that messages with lower priority are transmitted before messages with higher priority
    - ex) message priority: ID A and ID B > ID C



Outline

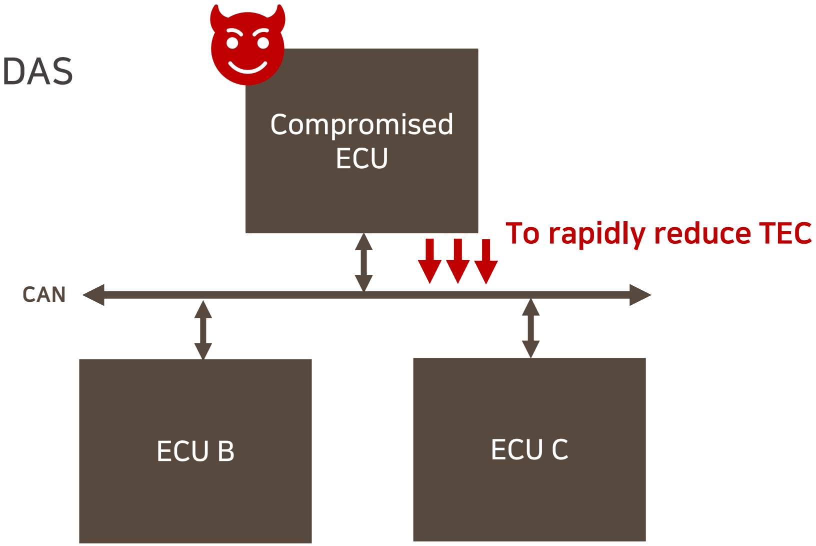
- Motivation
- Background
- Our Method
- Evaluation
- Discussion
- Conclusion

Attack Model

Our Method

- **Naïve attacker**
  - Using the default setting of the CAN controller

- **RIDAS-aware attacker**
  - Exploiting CAN controller's functions to evade RIDAS
    - **CAN controller reset**
    - **One-shot mode**
    - **Fast message transmission**

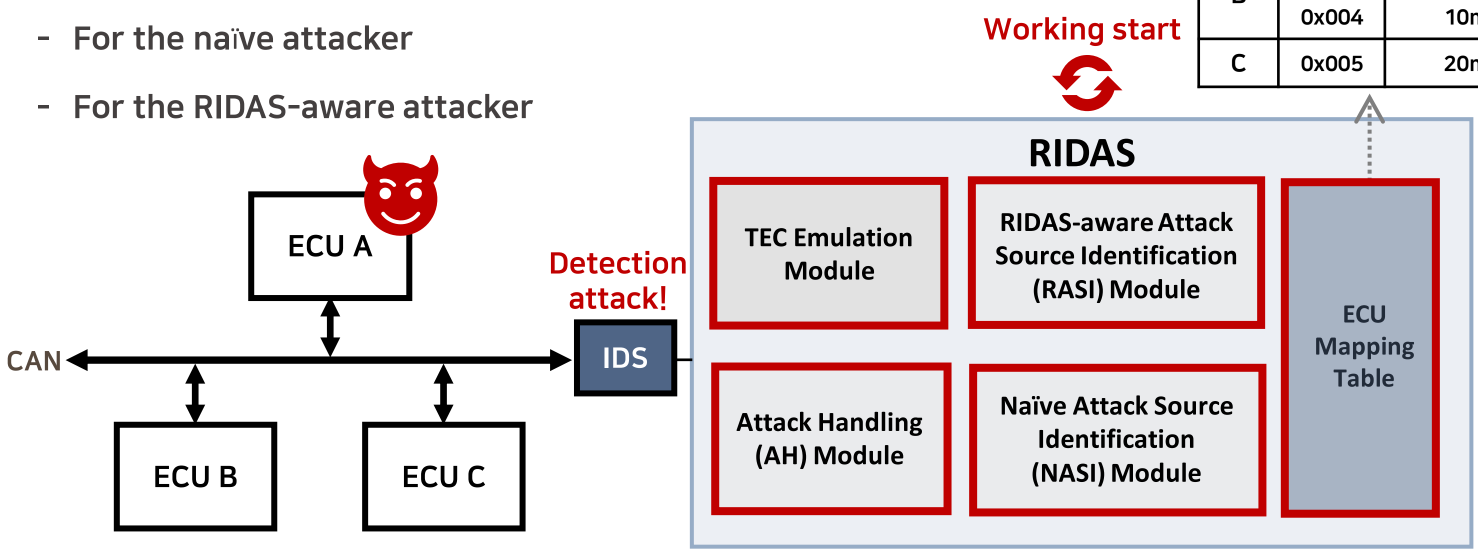


RIDAS: Workflow

Our Method

- System overview
  - Four modules
  - ECU mapping table
  - Two modes
    - For the naïve attacker
    - For the RIDAS-aware attacker

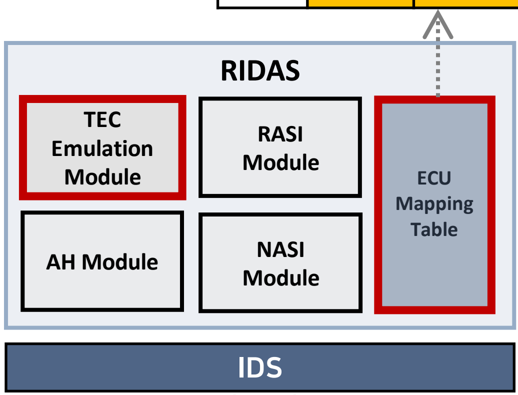
ECU	CAN ID	Transmission cycle
A	0x001	20ms
	0x002	10ms
B	0x003	20ms
	0x004	10ms
C	0x005	20ms

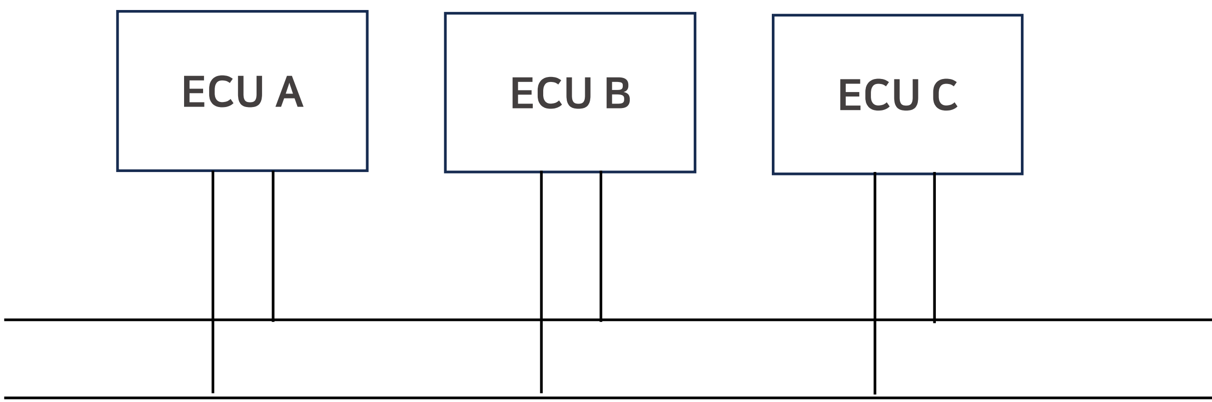


RIDAS: Workflow (for the naïve attacker)

- **First, initialization before starting RIDAS**
  - Start the TEC emulation
    - **Monitors the CAN bus in real-time** and **emulates the TEC of each ECU**
  - Set each representative ID (RID) for all ECUs
    - **CAN ID with the fastest transmission cycle and higher priority**

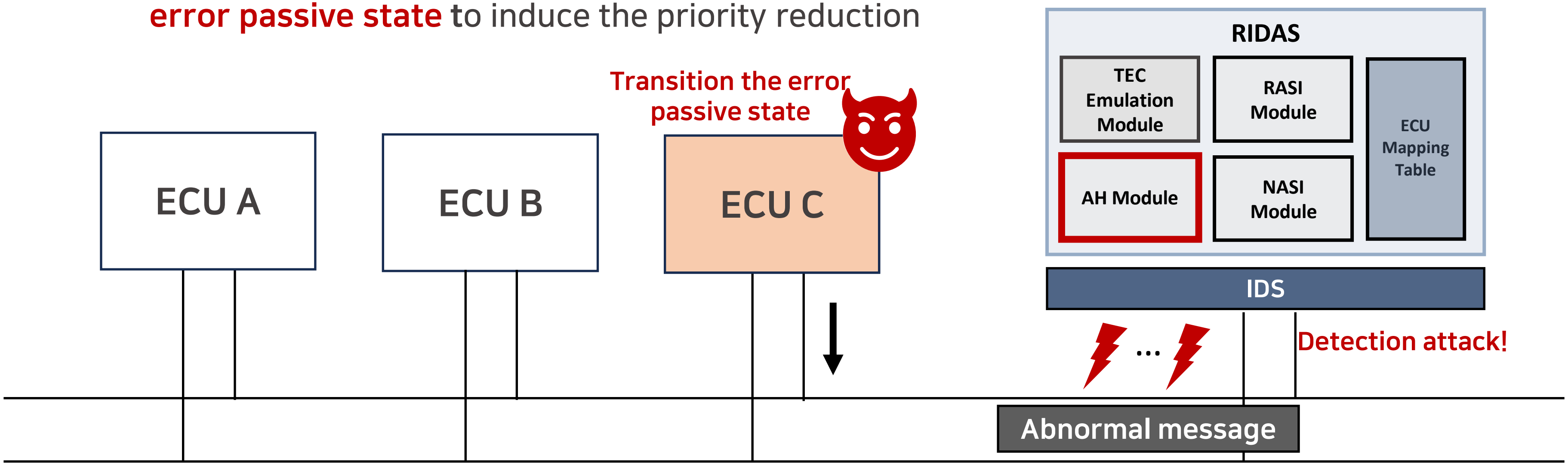
ECU	CAN ID	Transmission cycle
A	0x001	10ms
	0x002	20ms
B	0x003	20ms
	0x004	10ms
C	0x005	20ms





RIDAS: Workflow (for the naïve attacker)

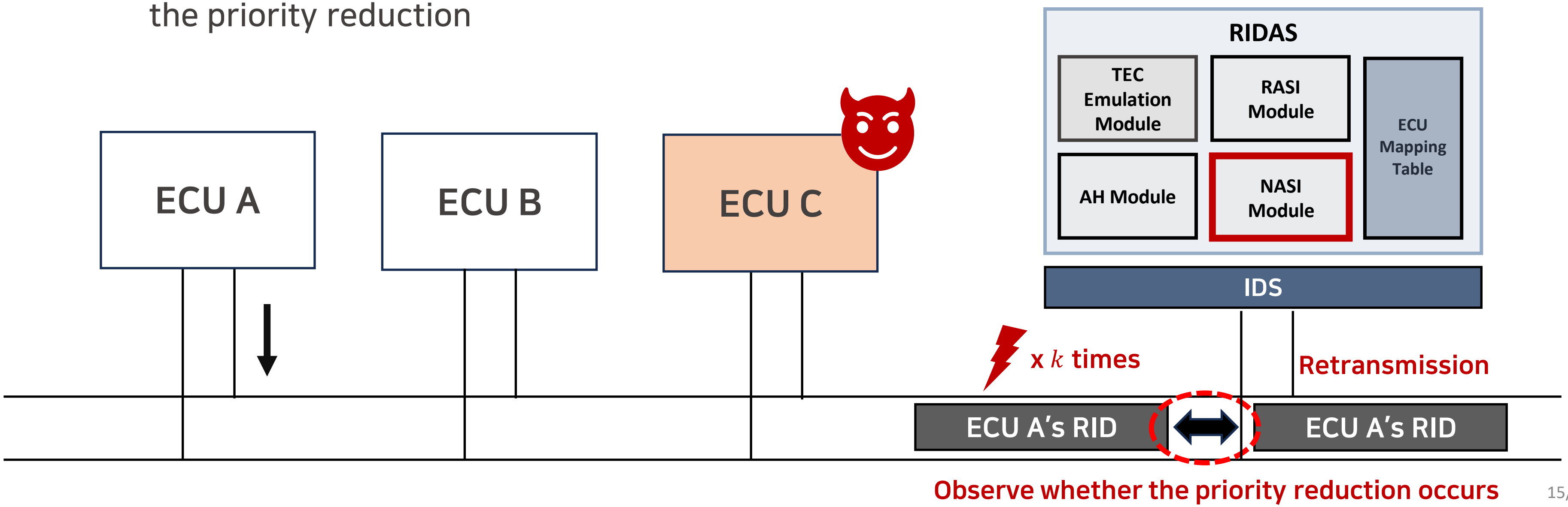
- **Second,** Transition the error state of the compromised ECU
  - When an attack message is detected, **the AH module injects continuous errors** before the message transmission is completed
  - AH module aims to **transition the compromised ECU from the error active state to the error passive state** to induce the priority reduction



RIDAS: Workflow (for the naïve attacker)

Our Method

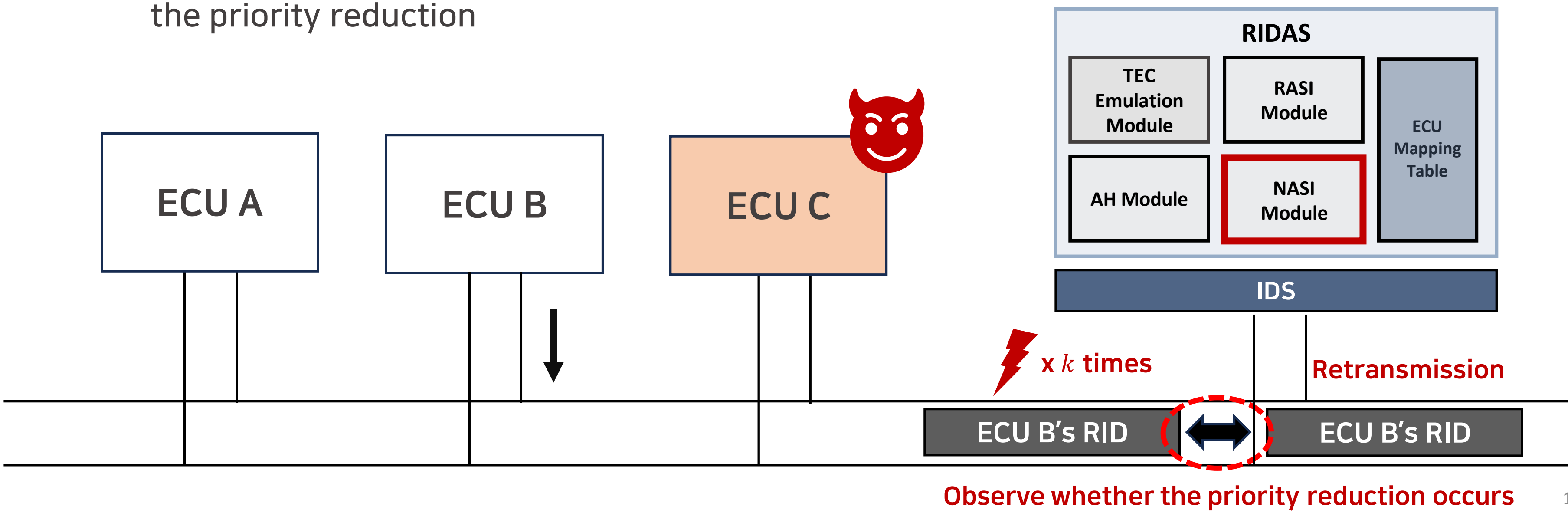
- **Third,** Identification of the ECU where the error state has transitioned
  - To **identify the compromised ECU (i.e., the naïve attacker)** who has transitioned to the error passive state, the NASI module sequentially inspects all ECUs
  - NASI module **generates bit-errors pre-defined number of times (k) for all RIDs** to observe the priority reduction



RIDAS: Workflow (for the naïve attacker)

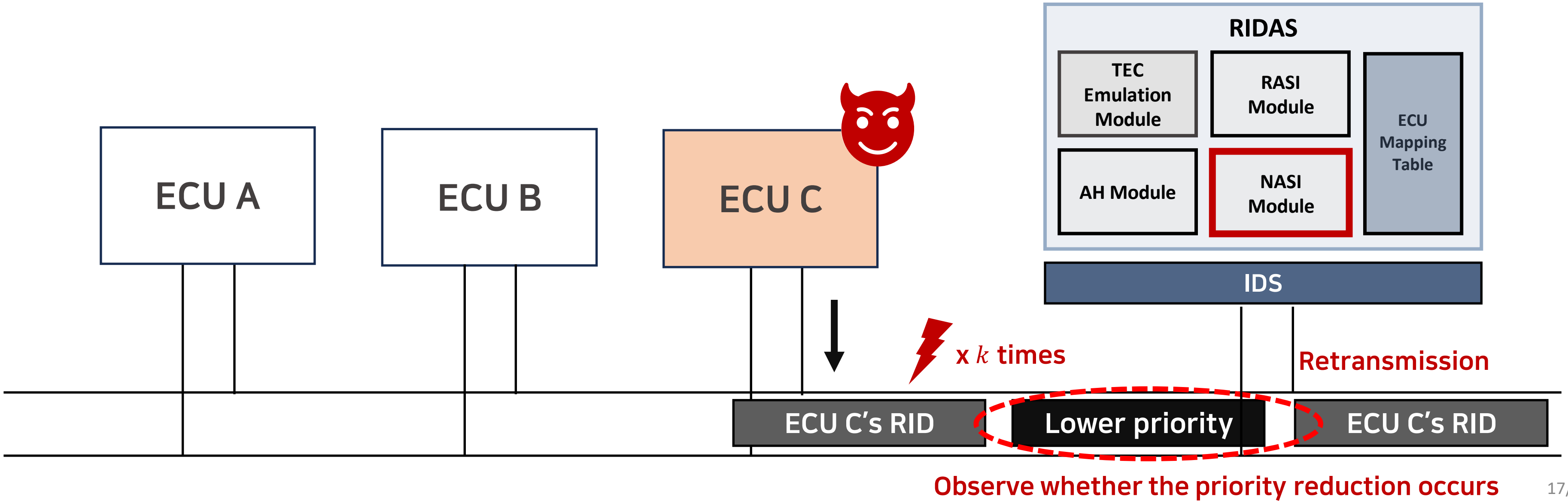
Our Method

- **Third,** Identification of the ECU where the error state has transitioned
  - To **identify the compromised ECU (i.e., the naïve attacker)** who has transitioned to the error passive state, the NASI module sequentially inspects all ECUs
  - NASI module **generates bit-errors pre-defined number of times (k) for all RIDs** to observe the priority reduction



RIDAS: Workflow (for the naïve attacker)

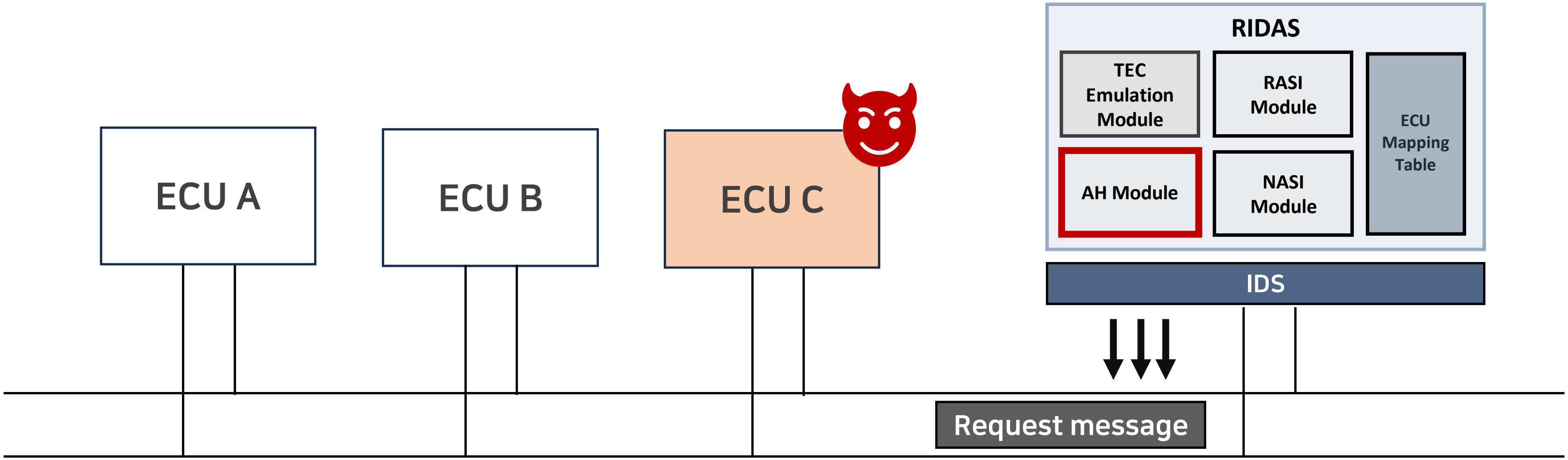
- **Third,** Identification of the ECU where the error state has transitioned
  - The ECU of RID in which priority reduction has occurred is the compromised ECU



RIDAS: Workflow (for the naïve attacker)

Our Method

- **Forth,** Restart RIDAS
  - Before restarting RIDAS, **the AH module reduces increased ECU's TEC** by generating request messages (e.g., remote frame or UDS message) for all ECUs

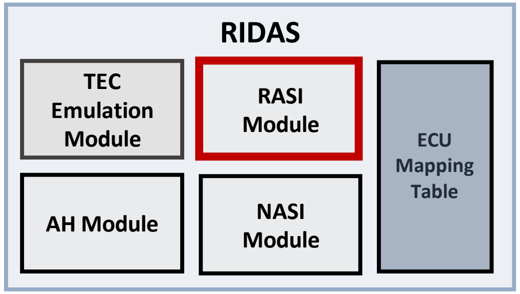


UDS: Unified Diagnostic Services

RIDAS: Workflow (for the RIDAS-aware attacker)

Our Method

- RASI module deals with attackers who evade RIDAS by monitoring the CAN bus
  - **CAN controller reset**
    - Detection of **the change in the transmission cycle** of certain CAN packets
  - **One-shot mode**
    - Detection of **the non-retransmission**
  - **Fast message transmission**
    - Whenever 8 fast messages are detected, a bit-error is injected **to restore the compromised node's TEC** to its original value



Outline

- Motivation
- Background
- Our Method
- Evaluation
- Discussion
- Conclusion

Experimental Setup

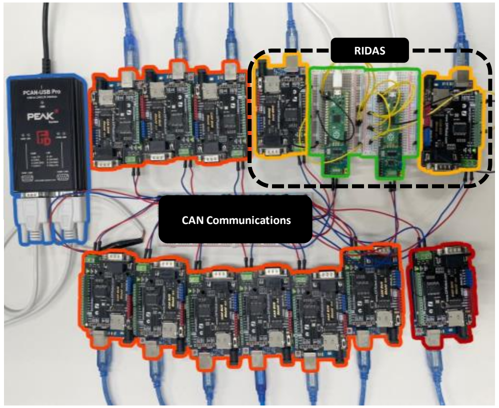
- CAN bus prototype
  - ECU: Arduino Uno with CAN Bus Shield (x10)
  - RIDAS: CAN Pico (x2), ECU (x2)
  - Monitoring tool: PCAN-USB Pro FD

Arduino Uno

CAN Bus Shield

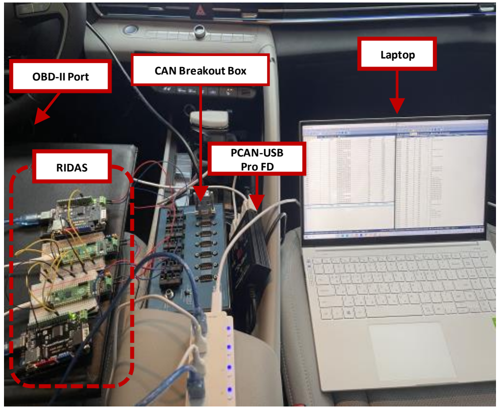
CAN Pico

PCAN-USB Pro FD



CAN bus prototype

- Real vehicle
  - RIDAS
  - Monitoring tool
  - CAN DBC: openDBC
  - Vehicle: Hyundai Avante CN7 2020



Real vehicle (Inside)

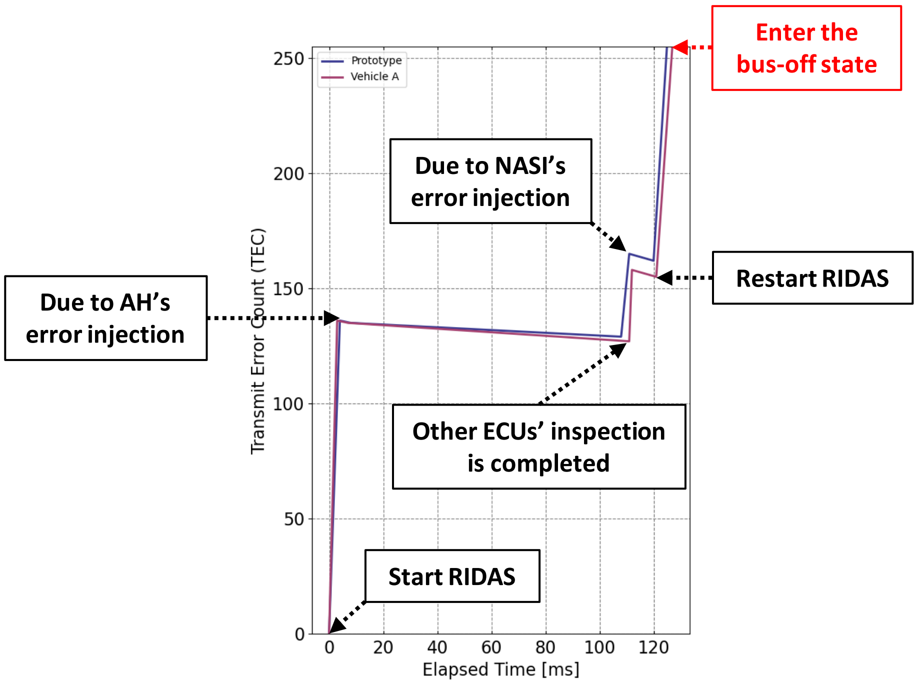
Real vehicle (Outside)

openDBC: https://github.com/commaai/opendbc

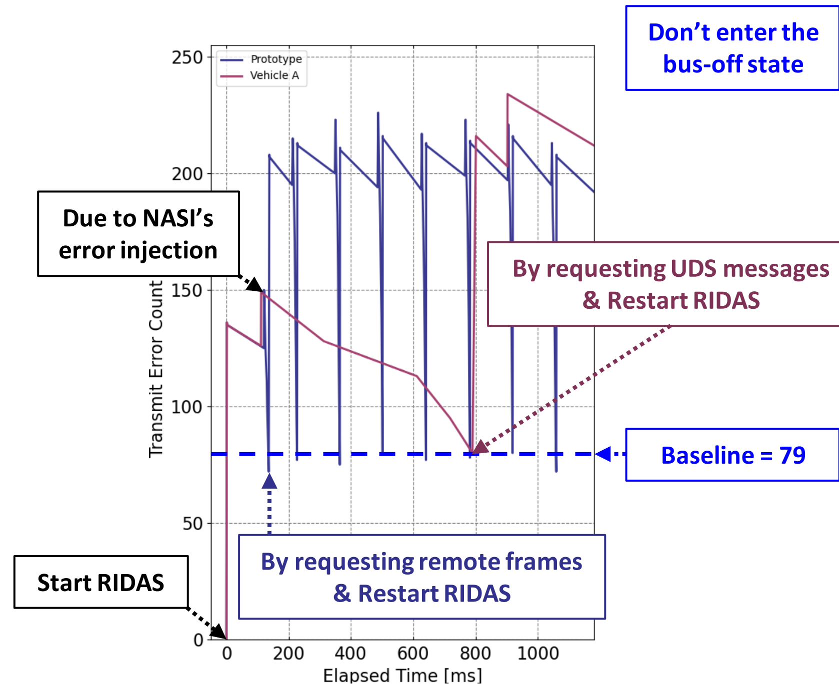
Evaluation of AH

Evaluation

- The AH module prevents driving the ECU into the bus-off state



without the AH module



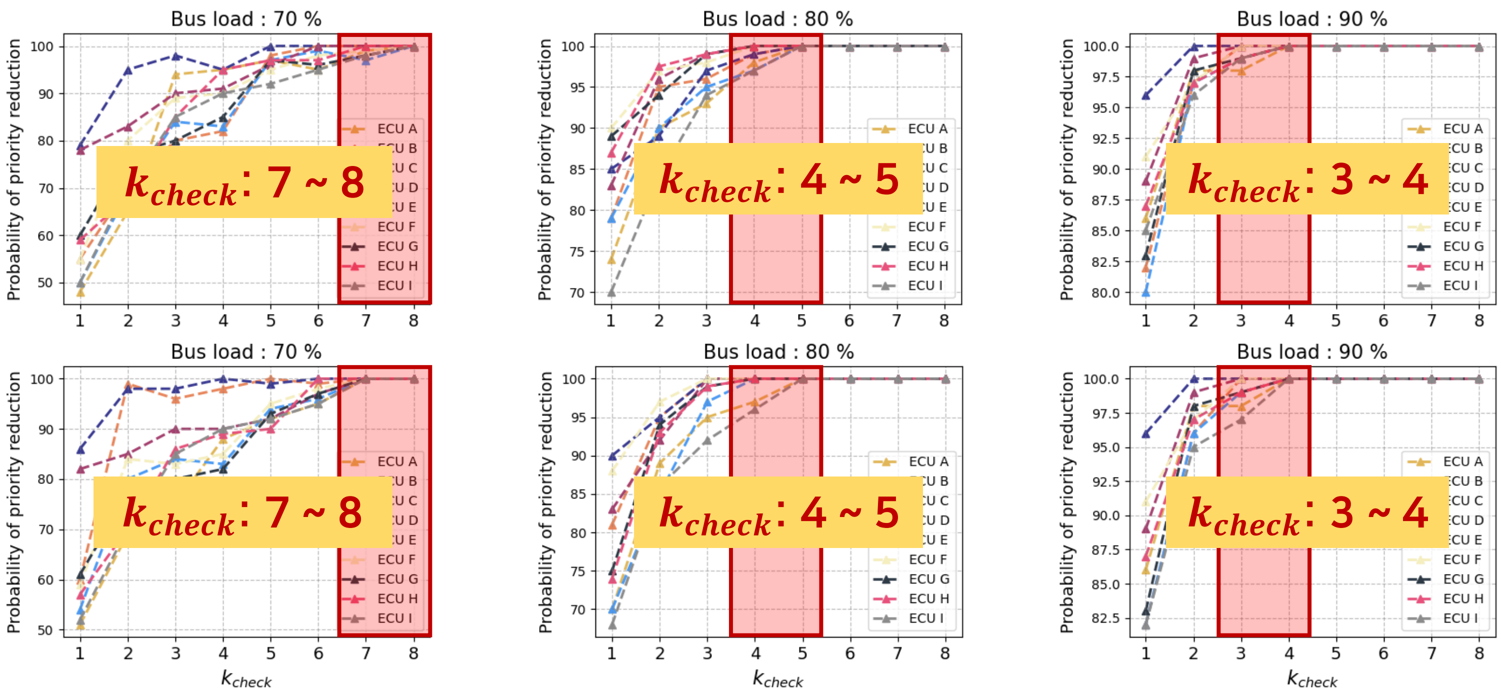
with the AH module

Evaluation of NASI (identification rate)

- The probability of priority reduction according to the bus load and the number of error injections (k_{check})

CAN bus prototype

Real vehicle



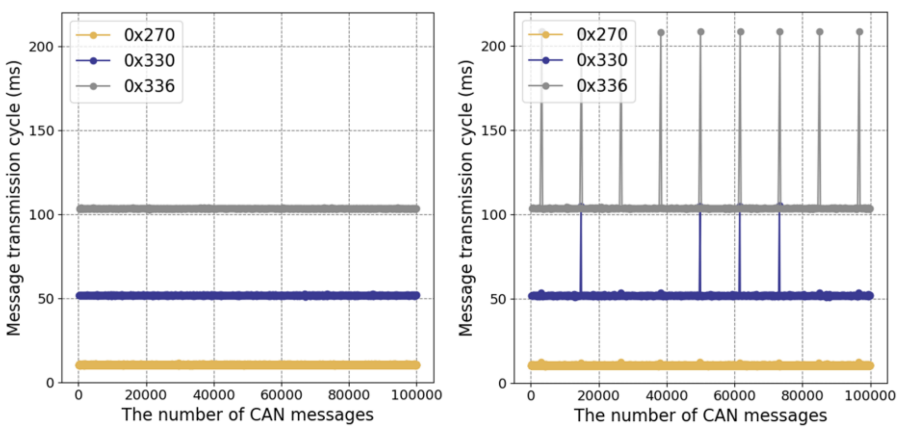
Evaluation of NASI (identification time)

- The average identification time of the NASI module
  - The identification time does not exceed 500ms in Avante CN7
    - Baseline = 79, k_{check} = 5, bus load = 80%

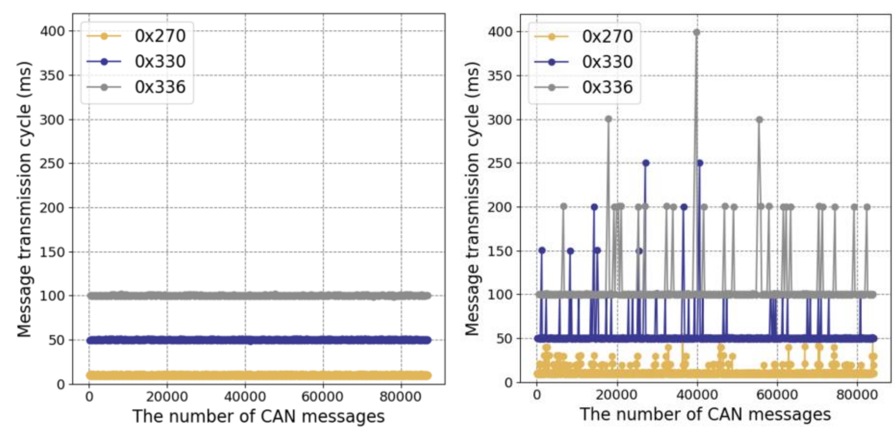
	Compromised ECU	A	B	C	D	E	F	G	H	I
Completion Time (ms)	Prototype	164.6	75.7	106.3	73	42.4	43.7	33.2	**5.4**	**563.5**
	Avante	150.1	75.8	99.6	72.3	38.4	40.1	29.7	**5.1**	**458.6**

Evaluation of RASI (for the RIDAS-aware attacker)

- Response to the ECU reset and the use of one-shot mode
  - There is a notable change in the message transmission cycle of the ECU



Message transmission cycle variations of ECU
(left: reset off / right: reset on)



Message transmission cycle variations of ECU
(left: default mode / right: one-shot mode)

Outline

- Motivation
- Background
- Our Method
- Evaluation
- Discussion
- Conclusion

Discussion

- Intrusion detection system
  - The IDS with RIDAS must detect attacks before the messages are completely transmitted
    - The worst-case response time-based IDS [5]

- Limitation of RIDAS
  - Direct TEC manipulation attack
    - Cannot drive a compromised ECU into the error passive state
  - ID reuse attack
    - Only nodes that attempt a masquerade attack can be identified

[5] Olufowobi, Habeeb, et al. "Saiducant: Specification-based automotive intrusion detection using controller area network (can) timing." (IEEE TVT 2019)

Outline

- Motivation
- Background
- Our Method
- Evaluation
- Discussion
- Conclusion

Conclusion

- Proposed a novel real-time attack node identification method, called RIDAS
  - RIDAS identifies an attack source using the priority reduction of an ECU’s error passive state
- Evaluated RIDAS on a CAN bus prototype and a real vehicle
  - RIDAS is capable of identifying the attack source without affecting driving
  - RIDAS is robust against changes in a vehicle’s environment
- In future research, we plan to integrate a lightweight IDS into RIDAS

Q&A

Thank you

Soongsil University KOREA UNIVERSITY