CANSentry: Securing CAN-Based CPS against Denial and Spoofing Attacks

Bo Luo

The University of Kansas

Collaborative work with Abdulmalik Humayed, Fengjun Li, and Jingqiang Lin

Slides courtesy of Dr. Abdulmalik Humayed, Jazan University, Saudi Arabia

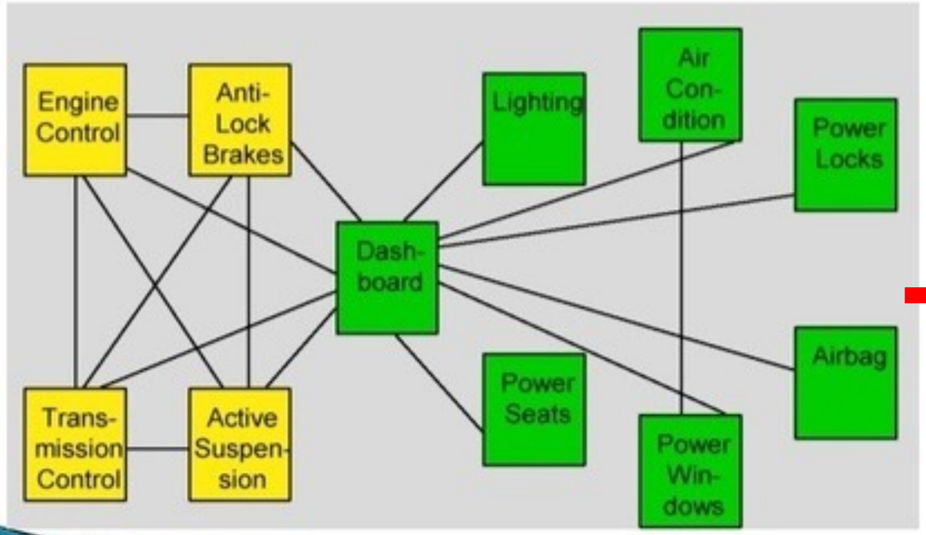
This paper was published in ESORICS 2020

The Controller Area Network (CAN)

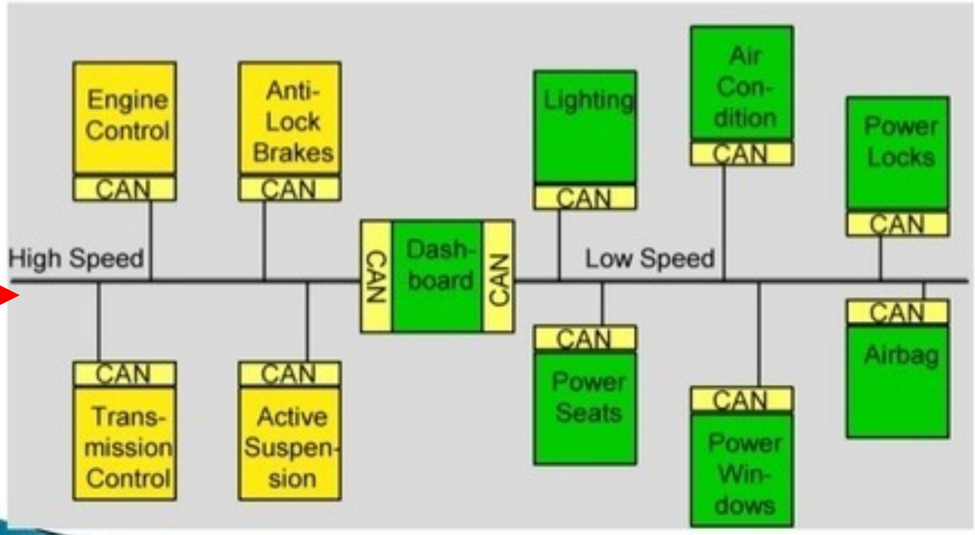
- The most common communication protocol for automotive and industrial applications
- On-Board Diagnostics (OBD-II) is mandated to be deployed in all cars for emission control with CAN only
- It allows data transmission in hostile environments
- Due to its bus topology, it greatly reduces vehicles’ cost and weight

The Controller Area Network (CAN)

Without CAN

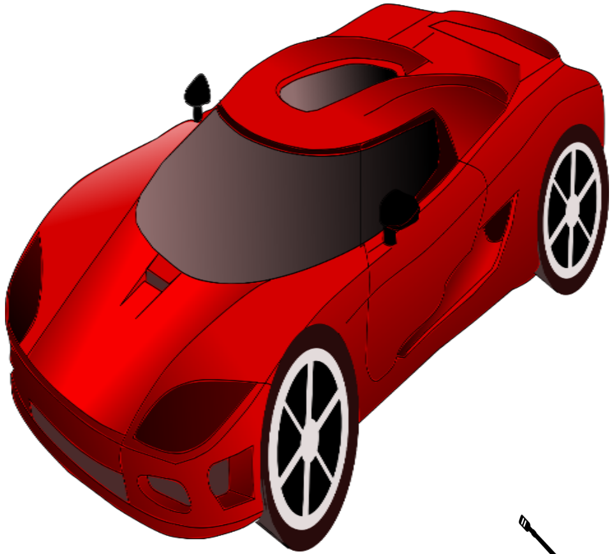


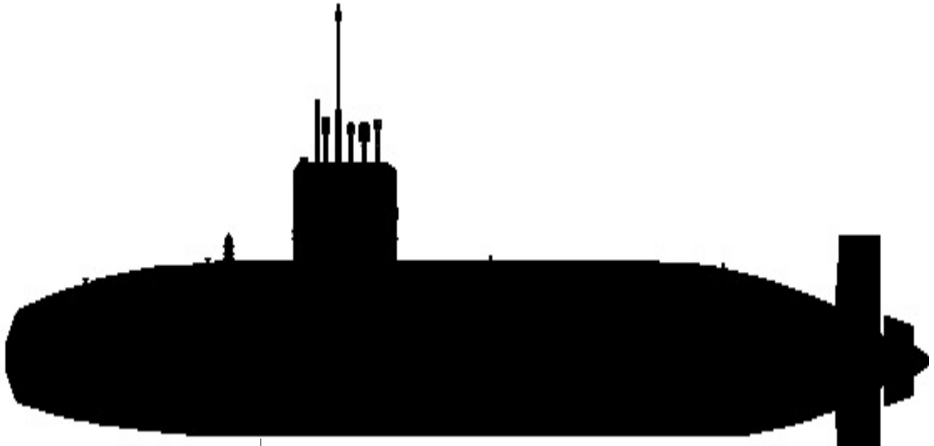
With CAN



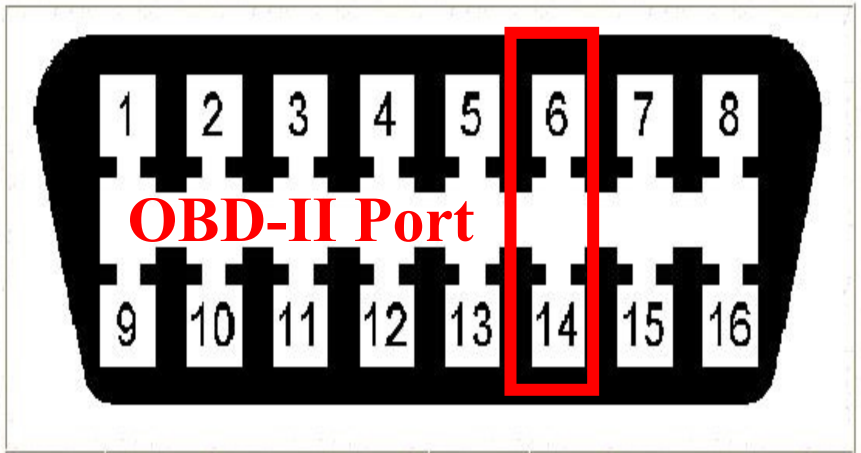
https://www.quora.com/Why-is-CAN-protocol-preferred-to-be-used-in-automotive-application

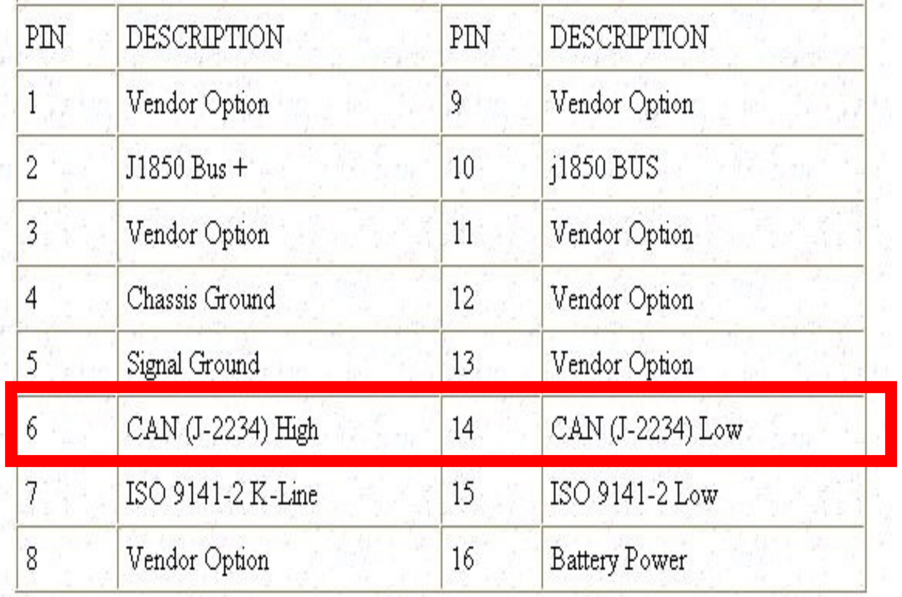
The Controller Area Network (CAN)





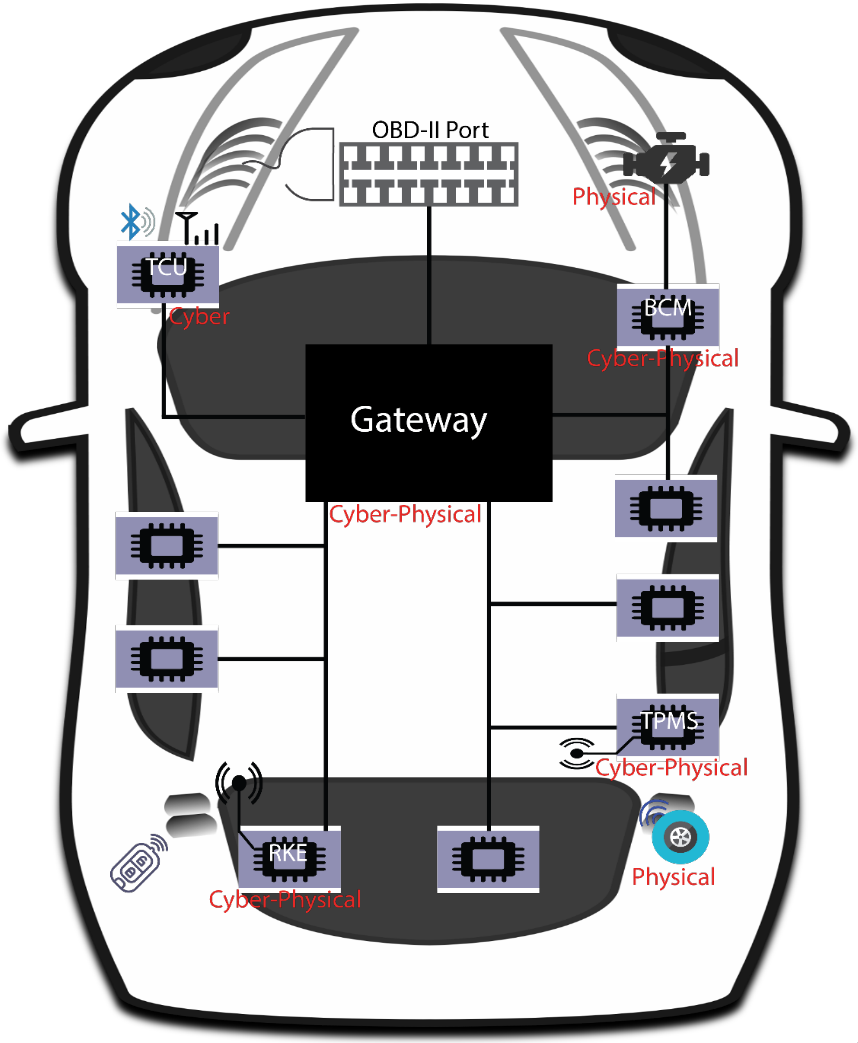
The Controller Area Network (CAN)





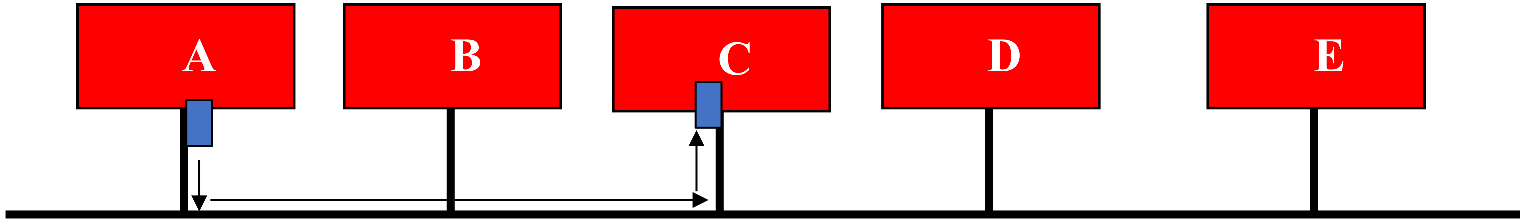
PIN	DESCRIPTION	PIN	DESCRIPTION
1	Vendor Option	9	Vendor Option
2	J1850 Bus +	10	j1850 BUS
3	Vendor Option	11	Vendor Option
4	Chassis Ground	12	Vendor Option
5	Signal Ground	13	Vendor Option
6	CAN (J-2234) High	14	CAN (J-2234) Low
7	ISO 9141-2 K-Line	15	ISO 9141-2 Low
8	Vendor Option	16	Battery Power

OBD-II Connector and Pinout



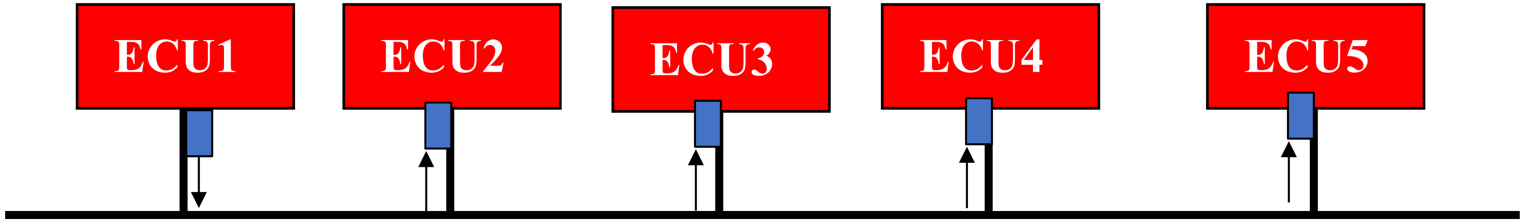
The Controller Area Network (CAN)

- Conventional peer-to-peer communication paradigm
  - Hey! I’m A, this is a message to C



Source: A	Dest: C	Message: Hello

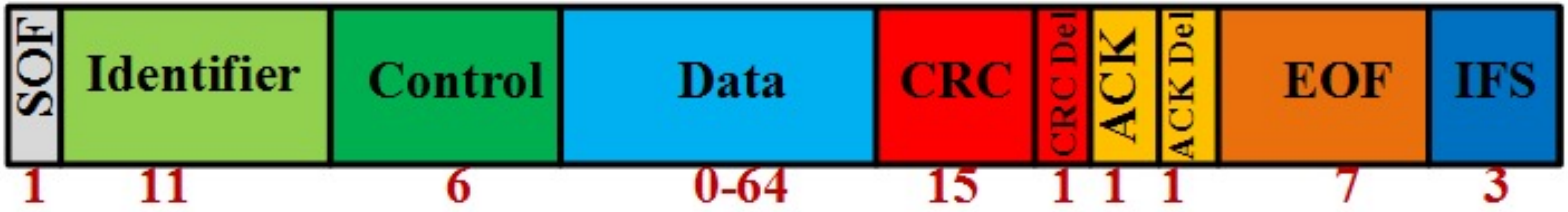
- In CAN
  - Hey! This is a message with ID 10 to everyone



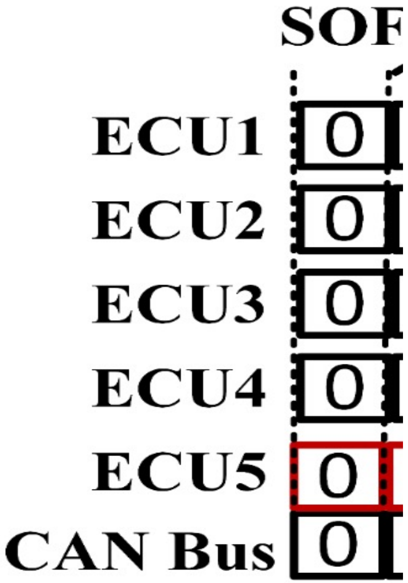
Message ID: 10	Message: Hello

CAN Frames & Arbitration

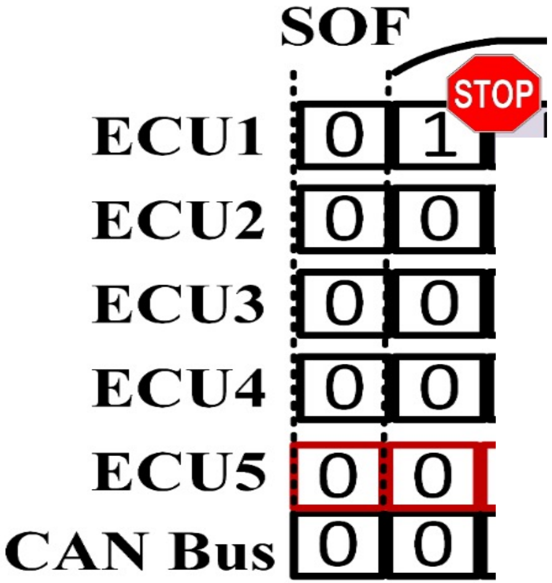
- Format of CAN frames
  - Each frame is identified by its arbitration ID
  - The frame with the lowest ID wins the arbitration and dominates the bus
  - Different types of frames use different ID
  - Ideally, IDs should be used uniquely across ECUs



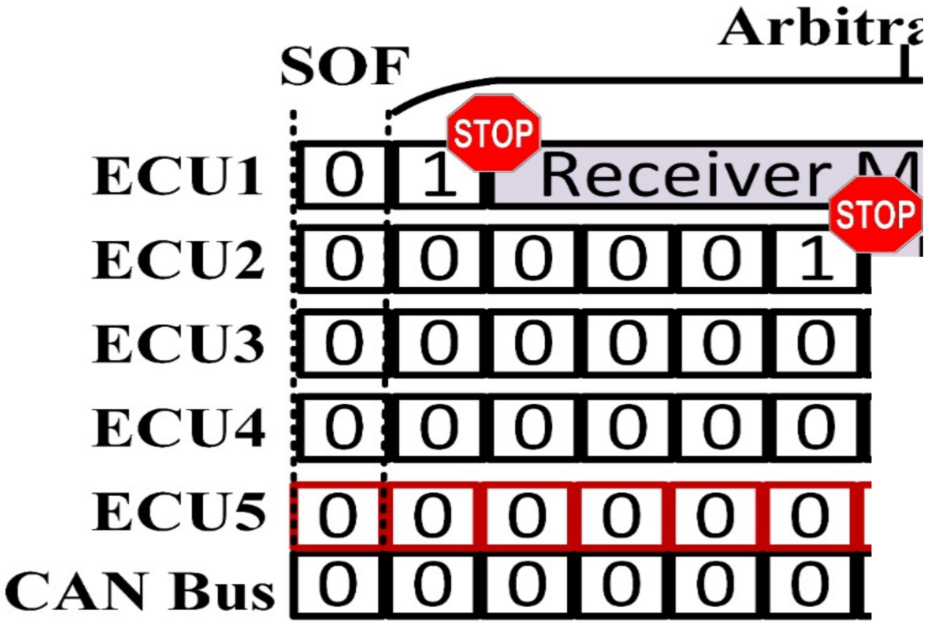
CAN Frames & Arbitration



CAN Frames & Arbitration

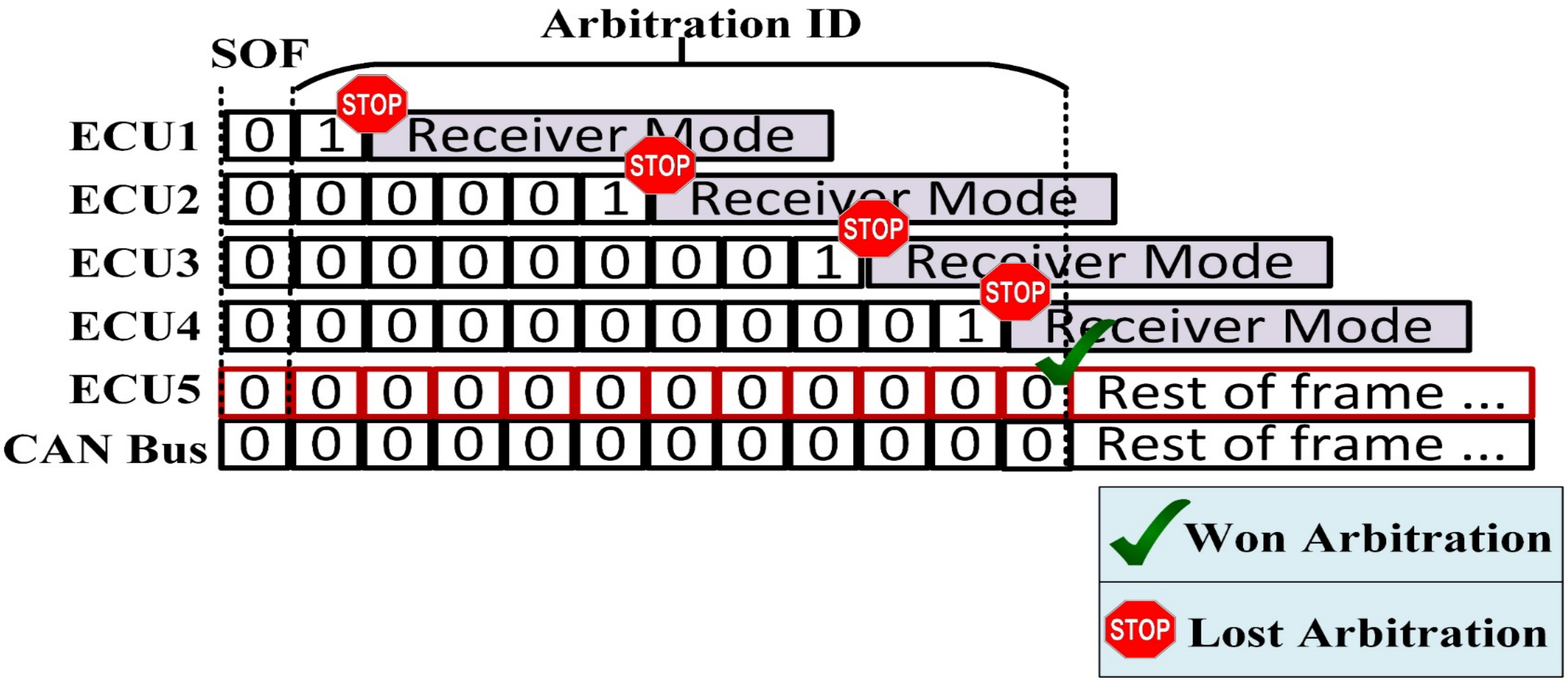


CAN Frames & Arbitration



ECU1	0	1	Receiver M			
ECU2	0	0	0	0	0	1
ECU3	0	0	0	0	0	0
ECU4	0	0	0	0	0	0
ECU5	0	0	0	0	0	0
CAN Bus	0	0	0	0	0	0

CAN Frames & Arbitration



	SOF	Arbitration ID											
ECU1	0	1	Receiver Mode										
ECU2	0	0	0	0	0	1	Receiver Mode						
ECU3	0	0	0	0	0	0	0	0	1	Receiver Mode			
ECU4	0	0	0	0	0	0	0	0	0	0	1	Receiver Mode	
ECU5	0	0	0	0	0	0	0	0	0	0	0	0	Rest of frame …
CAN Bus	0	0	0	0	0	0	0	0	0	0	0	0	Rest of frame …

✓ Won Arbitration

STOP Lost Arbitration

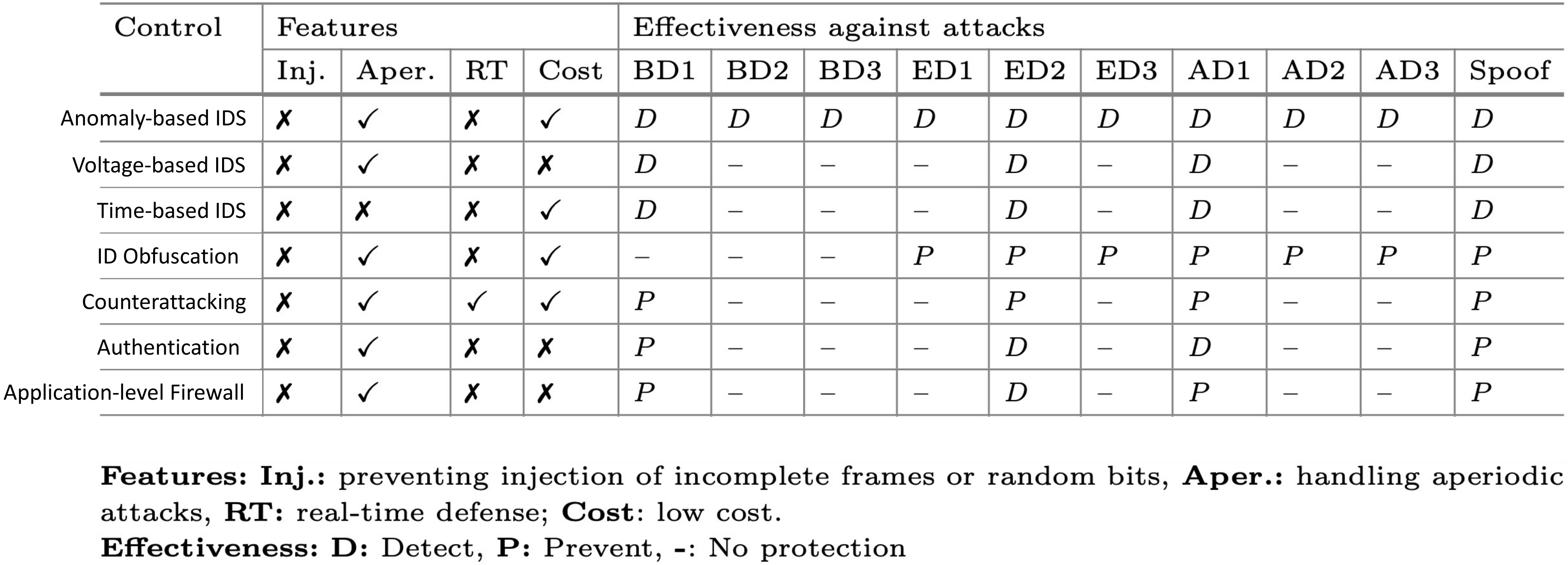
Existing Attacks

- **Denial attacks**
  [S&P’10, CCS’16, DIMVA’17, ARES’17, ESORICS’17]
  - Bus Denial (BD): 0x0 ID, dominant bits via Test Mode exploitation or custom ECU
  - ECU Denial (ED) : CAN Controller abuse or bypass & Error Handling abuse
  - Arbitration (AD) : injection of high priority IDs, dominant bit, or fake partial frames
- **Spoofing attacks**
  [S&P’10, DefCon’13, arXiv preprint’19, BlackHat’15]
  - An attacker sends any CAN ID of her choice to spoof other ECUs
  - ECUs could be compromised through a remote channel, and CAN frames are sent to unlock doors, stall the engine, or control the steering wheel

Existing Controls

- **Controls**
  - Node identification and IDS [USENIX Sec’16, CCS’17, CCS’18, TIFS’18, ACSAC’19]
  - CAN-ID Obfuscation [escar’15, SCAV’17, TODAES’17, Access’19]
  - Counterattacking [VTC’12, escar’14, SafeComp’18]
  - Authentication [DATE’09, DATE’13, ICCPS’13]
  - Firewalls [Micro’18]
- **Problems with existing controls**
  - Many require major software & hardware modifications
  - Changes to the protocol may require ALL ECUs to be updated
  - May introduce overheads for key management and crypto operations
  - Cannot defend against abused or compromised CAN controllers
  - None can handle attacks based on incomplete frames

Existing Controls



Control	Features				Effectiveness against attacks									
	Inj.	Aper.	RT	Cost	BD1	BD2	BD3	ED1	ED2	ED3	AD1	AD2	AD3	Spoof
Anomaly-based IDS	✗	✓	✗	✓	*D*	*D*	*D*	*D*	*D*	*D*	*D*	*D*	*D*	*D*
Voltage-based IDS	✗	✓	✗	✗	*D*	–	–	–	*D*	–	*D*	–	–	*D*
Time-based IDS	✗	✗	✗	✓	*D*	–	–	–	*D*	–	*D*	–	–	*D*
ID Obfuscation	✗	✓	✗	✓	–	–	–	*P*	*P*	*P*	*P*	*P*	*P*	*P*
Counterattacking	✗	✓	✓	✓	*P*	–	–	–	*P*	–	*P*	–	–	*P*
Authentication	✗	✓	✗	✗	*P*	–	–	–	*D*	–	*D*	–	–	*P*
Application-level Firewall	✗	✓	✗	✗	*P*	–	–	–	*D*	–	*P*	–	–	*P*

Features: Inj.: preventing injection of incomplete frames or random bits, **Aper.:** handling aperiodic attacks, **RT:** real-time defense; **Cost**: low cost.
Effectiveness: D: Detect, **P:** Prevent, **-**: No protection

Threat Model

- Attackers have remote access (via wireless access points) or brief physical access (via OBD-II port) to the CAN bus
- **The CAN Abuser**
  - Has complete control over ECU's software but not hardware
  - Abuse arbitration and error handling mechanisms to achieve malicious goals
- **The Skipper**
  - Skips CAN controller to directly access CAN bus
    - Uses a custom MCU directly connected to the bus
    - Manipulate CAN controller's GPIO pins to directly access to the bus
  - Attacker does not comply with CAN standards

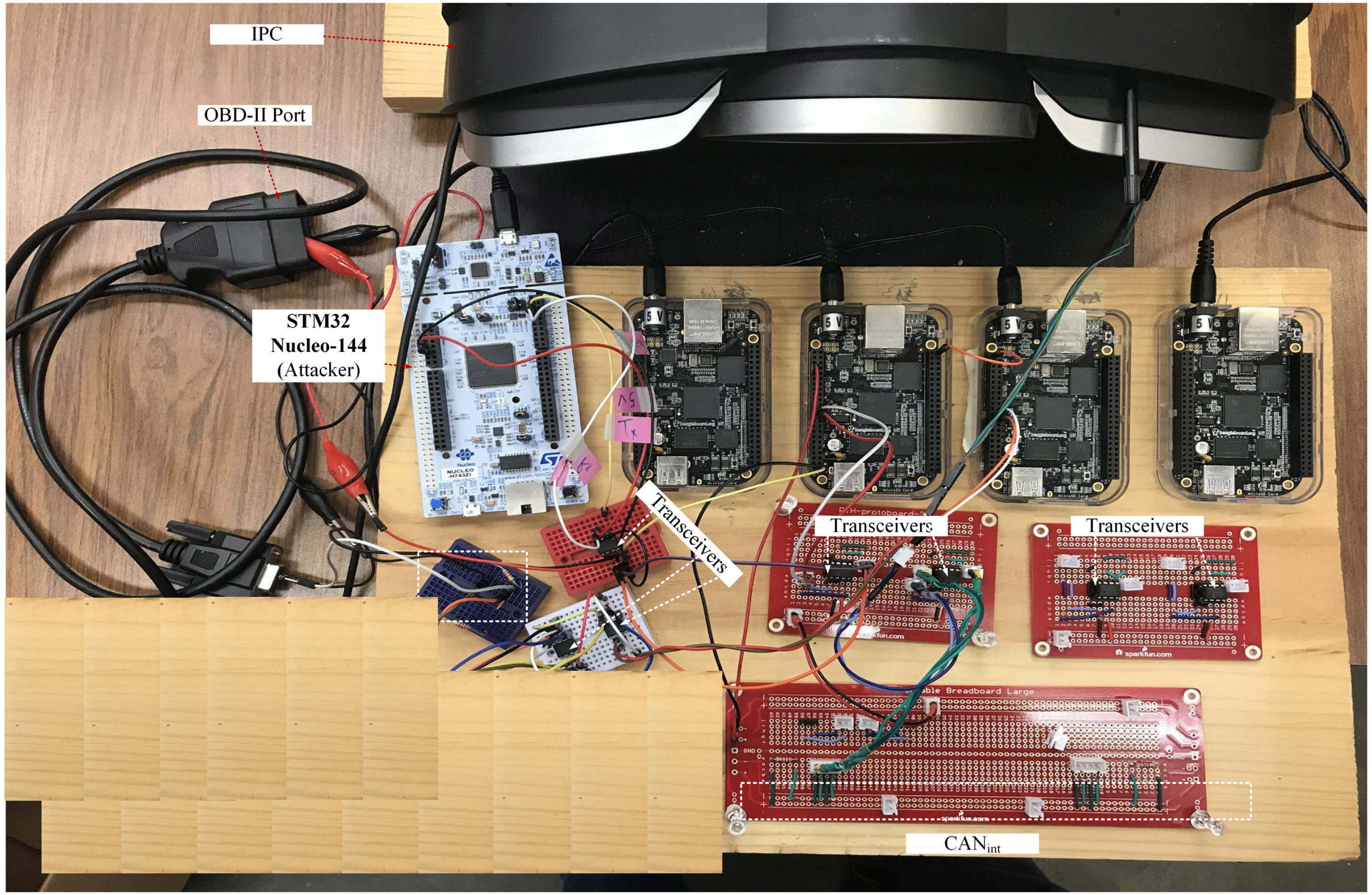
Threat Model

- **Denial attacks**: disable certain functionalities in a target ECU or bus
  - ECU is shutdown (bus-off state)
  - Bus is occupied
  - Specific CAN ID cannot win arbitration
- **Spoofing attacks**: transmit an ID belonging to another ECU
  - Receiving ECUs are spoofed resulting in:
    - Disabling brakes
    - Taking control of the steering wheel
    - False data injection

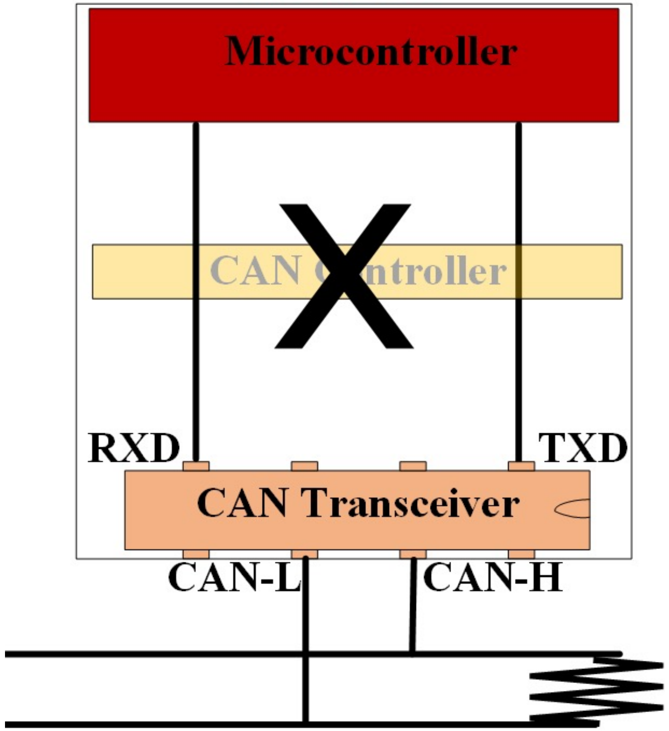
A Novel Stealthy Arbitration Denial Attack

- New objectives: Selective, Stealthy, & Practical
- Overview of the attack
  - Passively monitors the bus to detect a targeted ID in the arbitration phase
  - Overwrites the last recessive bit in the target ID to win arbitration
  - Completes the transmission with a fake frame
- Challenges
  - Existing tools only deal with complete CAN frames
  - High degree of precision is needed
  - Unexpected delays, premature injection, or malformed frames may cause incomplete frames resulting in bus errors

A Novel Stealthy Arbitration Denial Attack

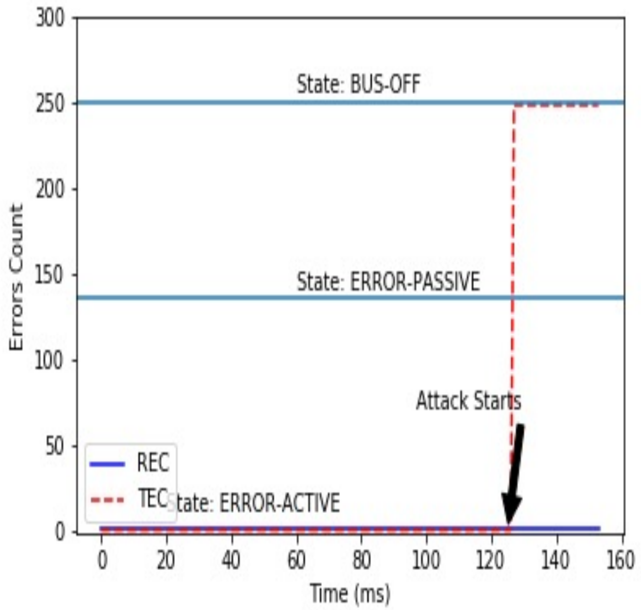


The Skipper

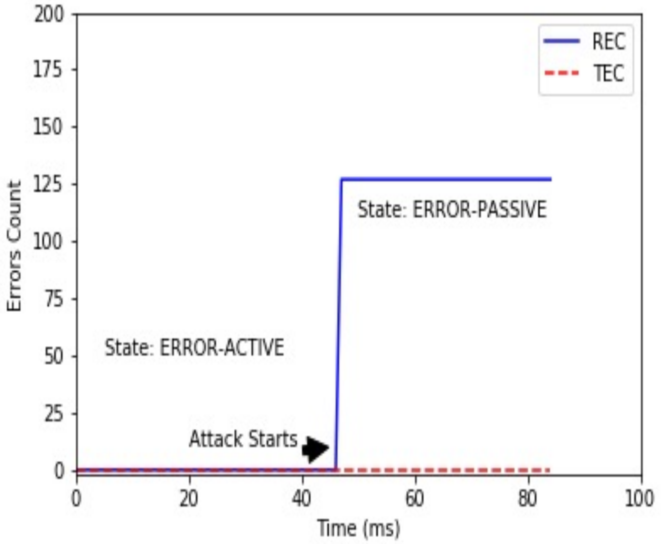


A Novel Stealthy Arbitration Denial Attack

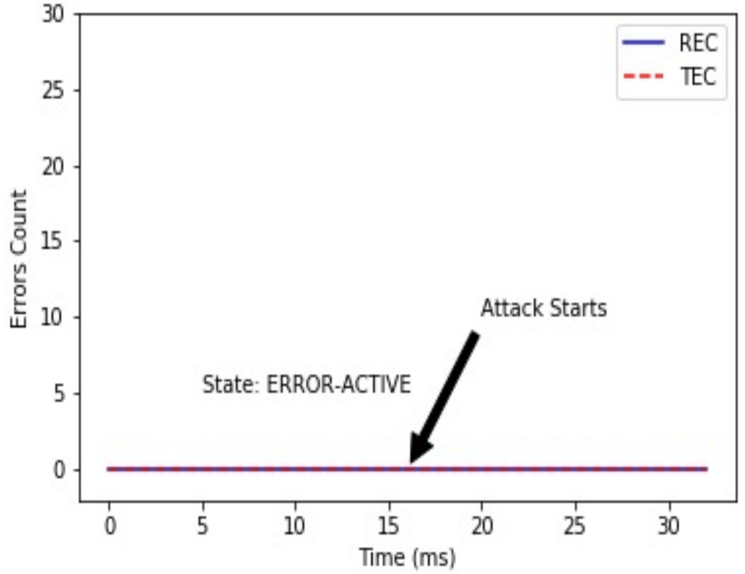
- Stealthy - the new attack does not incur any error
- Selective - the new attack only affects the targeted ECU or CAN ID



ECU Denial (ED)



Arbitration Denial 2 (AD2)

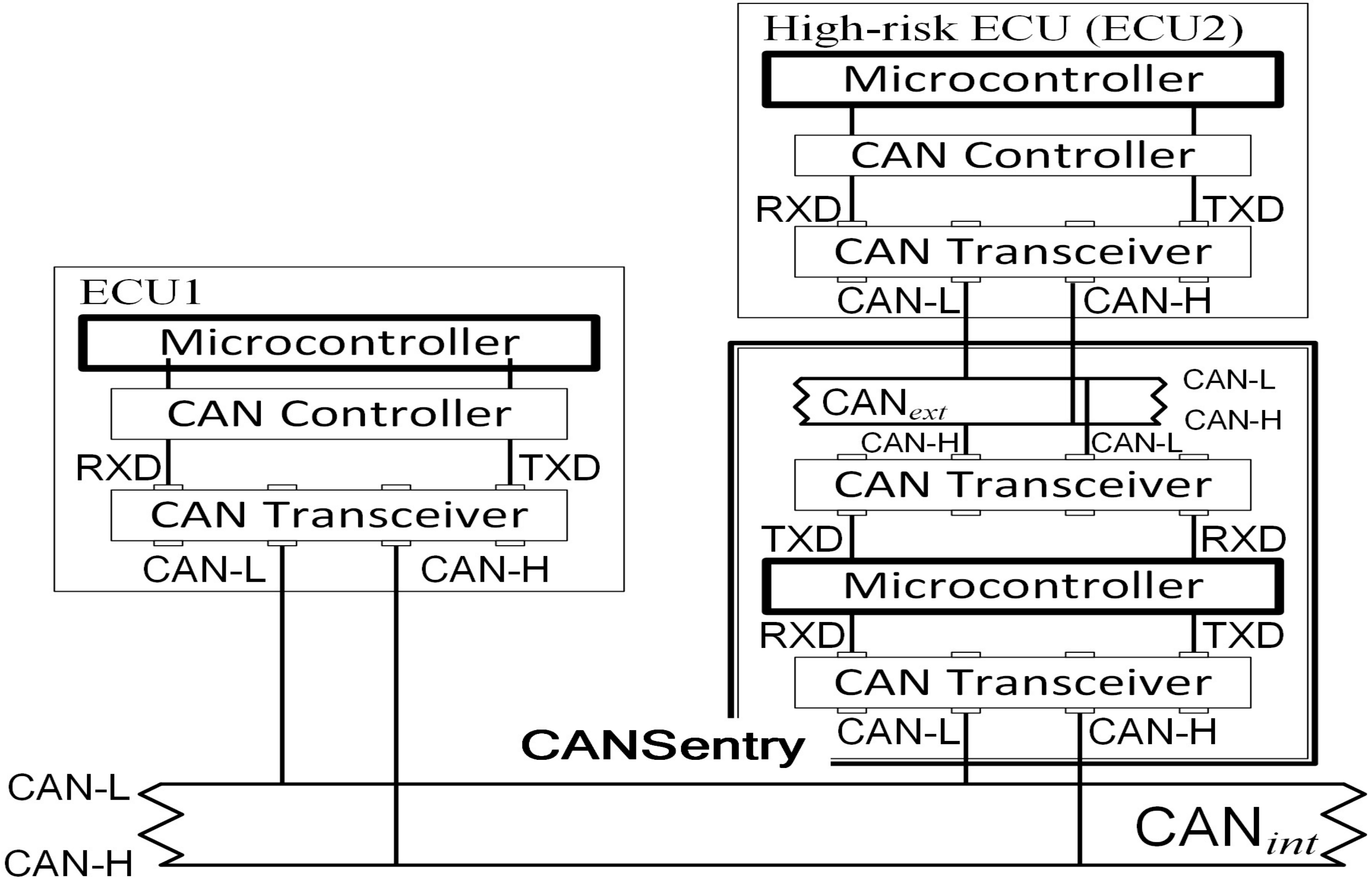


Arbitration Denial 3 (AD3)

CANSentry: Overview

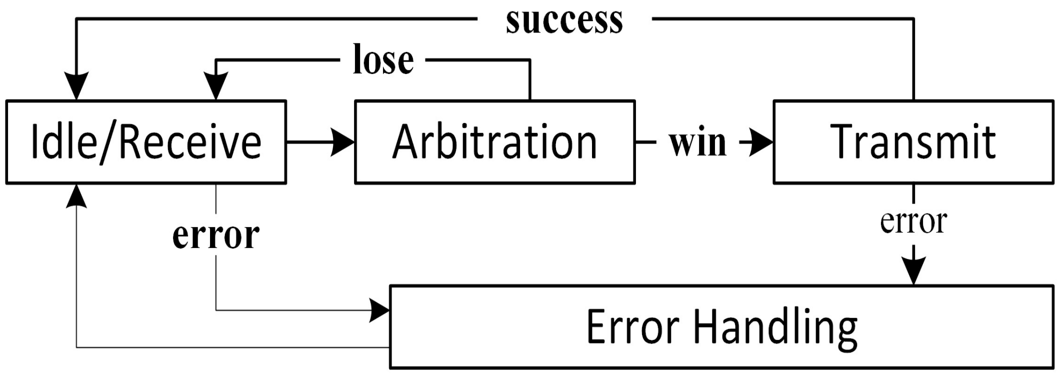
- A novel firewall sitting between any high-risk ECU and the bus
  - High-risk ECU: an ECU with remote access (entertainment system, Bluetooth) or open hardware access (OBD-II)
- Monitors incoming traffic from the ECU
- Ensures the consistency between the CAN Bus state and the ECU state
  - E.g., when another ECU wins arbitration and transmits data, the protected ECU could only receive. It cannot interrupt the BUS traffic.
- Uses firewall rules to block illegal traffic
- Low-cost and highly efficient implementation

CANSentry: Architecture

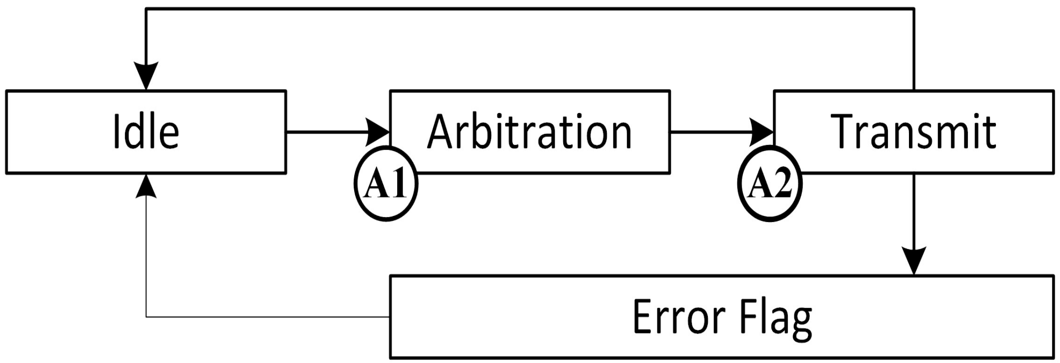


CANSentry: States

- States and states transitions of CAN bus and nodes



States of a CAN Node



States of the CAN Bus

CANSentry: State Transition Rules

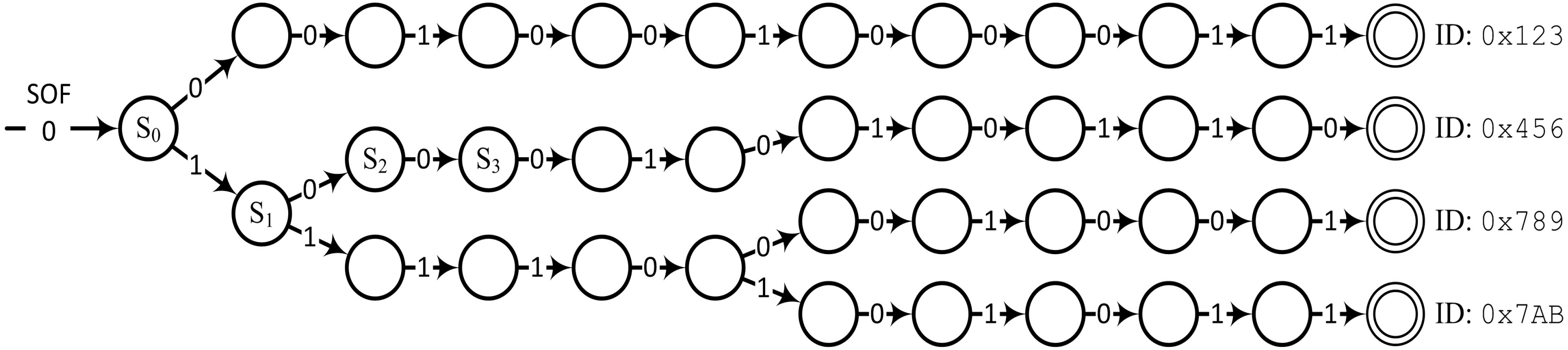
- **Main principle**: The fundamental principle of the firewall is to ensure that at any time high-risk nodes on the external bus operate in a state consistent with the state of the internal bus.

CAN_{INT} State	Consistent State in CAN_{EXT}
IDLE	IDLE/RECIEVE, ARBITRATION, and TRANSMIT
ARBITRATION	IDLE/RECIEVE and ARBITRATION
TRANSMIT or ERROR FLAG	IDLE/RECEIVE

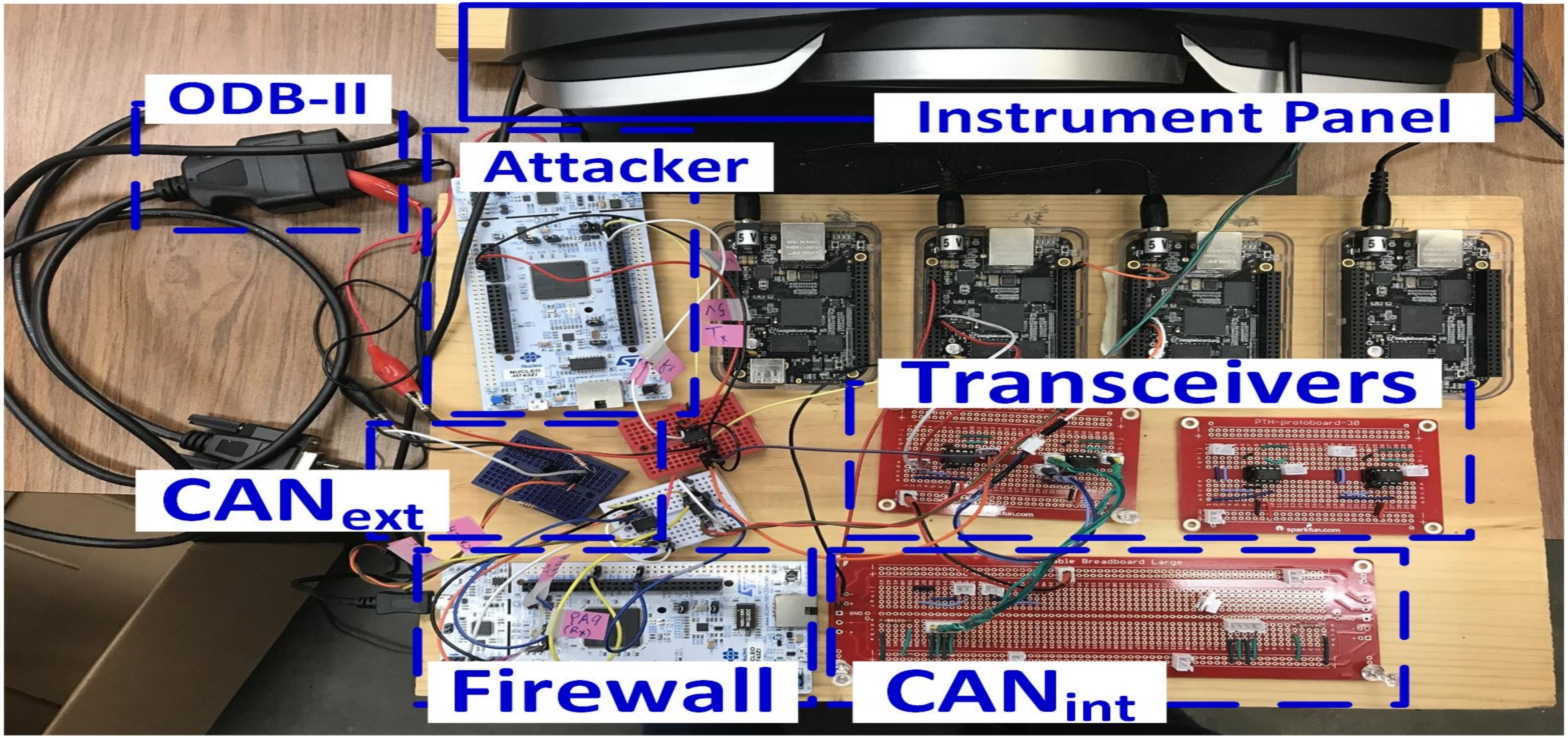
Example: ***R_1***: When the internal bus is in either $\text{TRANSMIT}_{\text{int}}$ or ERRORFLAG state, the firewall always forwards the traffic from CAN_{int} to CAN_{ext} and blocks the traffic from external to internal, regardless of high-risk node's state.

CANSentry: CAN ID filtering in arbitration

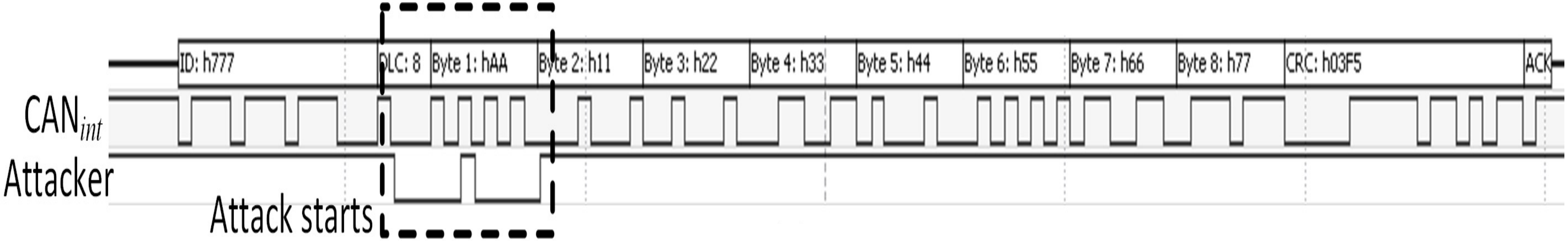
- When the internal bus is in ARBITRATION state, the firewall forwards traffic that has a CAN ID in the **arbitration whitelist** and conforms to CAN specifications from CAN$_{ext}$ to CAN$_{int}$.
- Prevent spoofing attacks
- Uses automata for efficient CAN ID matching
  - Example: this automata allows four CAN IDs: 0x123, 0x456, 0x789, 0x7AB



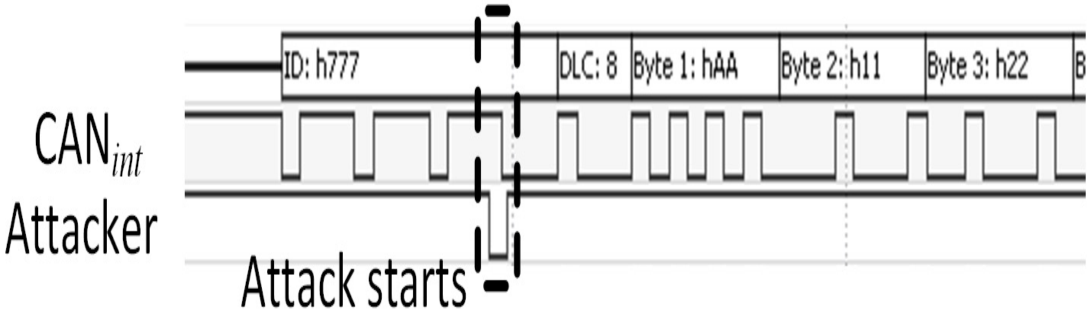
CANSentry: Implementation



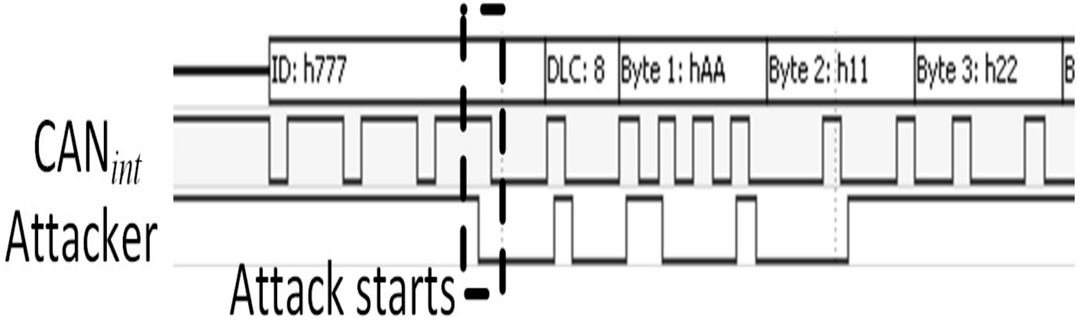
CANSentry: Evaluation



Bus-Off Attack



Arbitration Denial Attack

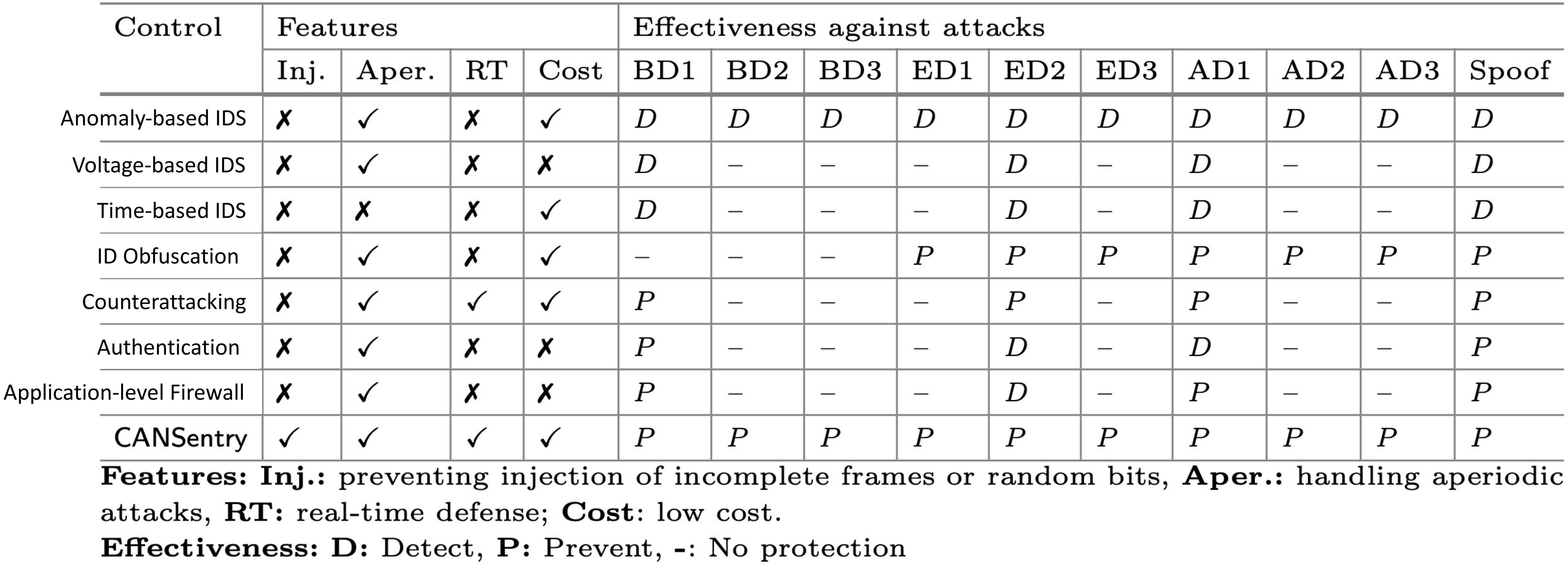


Stealthy Arbitration Denial Attack

Security Analysis

- CANSentry nodes are deployed in a physically secure environment
  - Makes it difficult for an adversary to bypass or alter
- CANSentry only has two network interfaces - CAN_{ext} and CAN_{int}
  - The limited communication channel and the simplicity of CAN makes it impractical to compromise the operations of the firewall from CAN_{ext}.
- The simplicity of the firewall makes it unlikely to have significant software faults

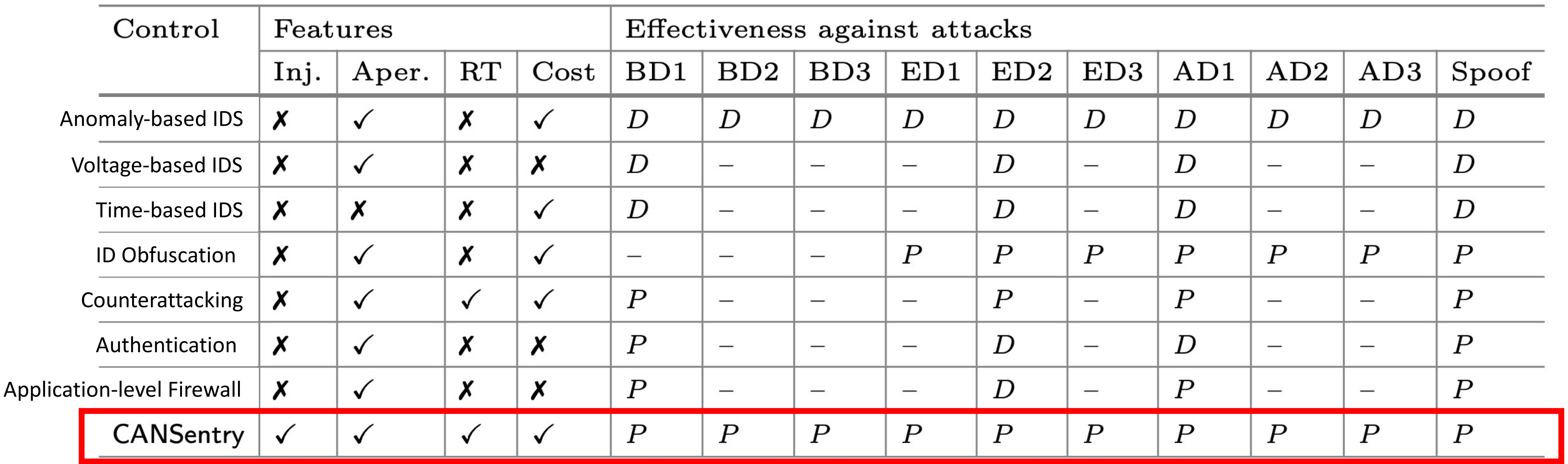
Security Analysis



Control	Features				Effectiveness against attacks									
	Inj.	Aper.	RT	Cost	BD1	BD2	BD3	ED1	ED2	ED3	AD1	AD2	AD3	Spoof
Anomaly-based IDS	✗	✓	✗	✓	*D*	*D*	*D*	*D*	*D*	*D*	*D*	*D*	*D*	*D*
Voltage-based IDS	✗	✓	✗	✗	*D*	–	–	–	*D*	–	*D*	–	–	*D*
Time-based IDS	✗	✗	✗	✓	*D*	–	–	–	*D*	–	*D*	–	–	*D*
ID Obfuscation	✗	✓	✗	✓	–	–	–	*P*	*P*	*P*	*P*	*P*	*P*	*P*
Counterattacking	✗	✓	✓	✓	*P*	–	–	–	*P*	–	*P*	–	–	*P*
Authentication	✗	✓	✗	✗	*P*	–	–	–	*D*	–	*D*	–	–	*P*
Application-level Firewall	✗	✓	✗	✗	*P*	–	–	–	*D*	–	*P*	–	–	*P*
CANSentry	✓	✓	✓	✓	*P*	*P*	*P*	*P*	*P*	*P*	*P*	*P*	*P*	*P*

Features: Inj.: preventing injection of incomplete frames or random bits, **Aper.:** handling aperiodic attacks, **RT:** real-time defense; **Cost**: low cost.
Effectiveness: D: Detect, **P:** Prevent, **-**: No protection

Security Analysis



Control	Features				Effectiveness against attacks									
	Inj.	Aper.	RT	Cost	BD1	BD2	BD3	ED1	ED2	ED3	AD1	AD2	AD3	Spoof
Anomaly-based IDS	✗	✓	✗	✓	*D*	*D*	*D*	*D*	*D*	*D*	*D*	*D*	*D*	*D*
Voltage-based IDS	✗	✓	✗	✗	*D*	–	–	–	*D*	–	*D*	–	–	*D*
Time-based IDS	✗	✗	✗	✓	*D*	–	–	–	*D*	–	*D*	–	–	*D*
ID Obfuscation	✗	✓	✗	✓	–	–	–	*P*	*P*	*P*	*P*	*P*	*P*	*P*
Counterattacking	✗	✓	✓	✓	*P*	–	–	–	*P*	–	*P*	–	–	*P*
Authentication	✗	✓	✗	✗	*P*	–	–	–	*D*	–	*D*	–	–	*P*
Application-level Firewall	✗	✓	✗	✗	*P*	–	–	–	*D*	–	*P*	–	–	*P*
CANSentry	✓	✓	✓	✓	*P*	*P*	*P*	*P*	*P*	*P*	*P*	*P*	*P*	*P*

Features: Inj.: preventing injection of incomplete frames or random bits, **Aper.:** handling aperiodic attacks, **RT:** real-time defense; **Cost**: low cost.
Effectiveness: D: Detect, **P:** Prevent, **-**: No protection

]

Conclusions

- We summarized existing DoS and spoofing attacks on CAN
- We proposed and implemented a novel stealthy selective arbitration DoS attack
- We designed a novel *CANSentry* firewall to defend against attacks that violate the CAN standard or abuse CAN's error-handling mechanism
  - *CANSentry* is the first solution that detects and prevents a broad spectrum of CAN denial and spoofing attacks
  - *CANSentry* does not introduce noticeable overhead or delay
  - It is very cost-effective

Acknowledgment

- Fengjun Li and Bo Luo were sponsored in part by NSF CNS-1422206, DGE-1565570, NSA Science of Security Initiative H98230-18-D-0009, and the Ripple University Blockchain Research Initiative.