Viden: Attacker Identification on In-Vehicle Networks

Kyong-Tak Cho and Kang G. Shin
University of Michigan, Ann Arbor

CCS 2017

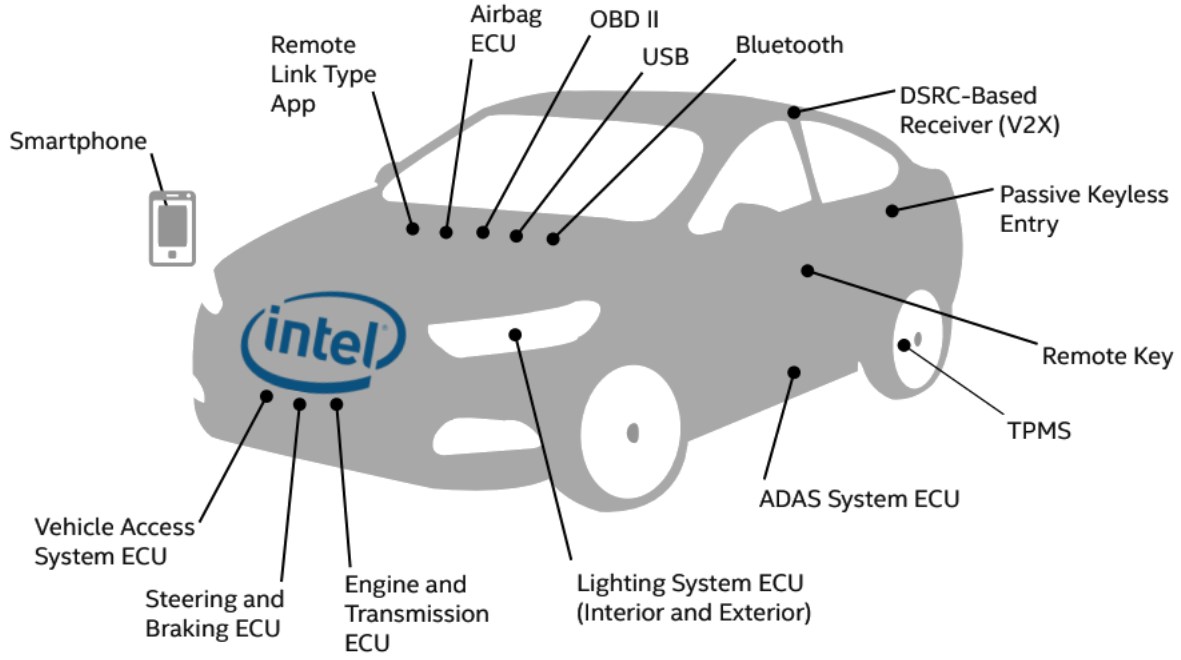
Presented By
Md Mahbubur Rahman
Wayne State University

Outline

- Motivation
- Scope
- **Viden**
- Implementation
- Evaluation
- Conclusion

Motivation

- Remote/Driverless control of a car is no longer a science fiction!
- Security/Safety vulnerability
  - In-vehicle network
- Electronic Control Unit (ECU)
  - Embedded device (microcontroller)
  - Reads sensor data and actuates accordingly
  - 150+ ECUs in today's cars



15 of the most hackable and exposed attack surfaces (2015)

Scope

- Clock-based **Intrusion Detection System** (C**IDS**) [Chao et al. USENIX SS'16]
  - Different clock-skews in different ECUs
  - Periodic messages for attacker identification
  - What if the attacker sends aperiodic messages?

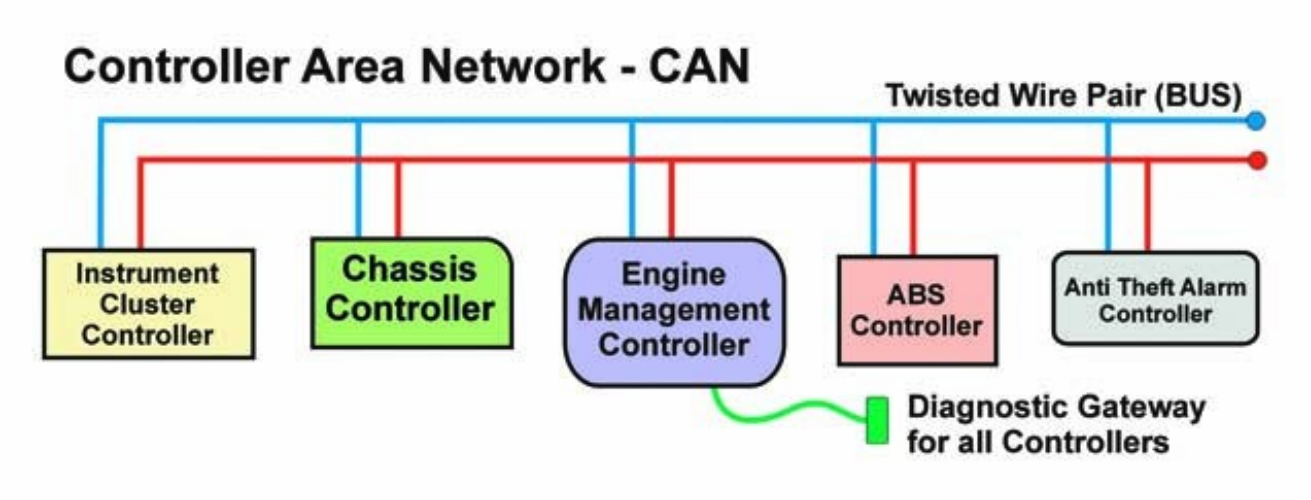
- Voltage-based IDS
  - Different **Mean Squared Errors** for voltage measurements of different ECUs
  - Works for low-speed connection between ECUs (10kbps)
  - Modern ECU network is 500kbps

- Other **time** and **frequency** domain-based ECU fingerprinting
  - RMS Amplitude
  - Use Supervised Learning Algorithms (e.g., SVM)
  - Not adaptable to changes

Scope

- Viden: Looks at attack messages from the perspective of ECU's **output voltages** on the in-vehicle network.

- In-vehicle network: **Controller Area Network (CAN)**
  - Each controller has a CAN transceiver



CAN Hight: CANH

Can Low: CANL

Source:

Controller Area Network (CAN)

- ECUs broadcast sensor data via CAN frame/message

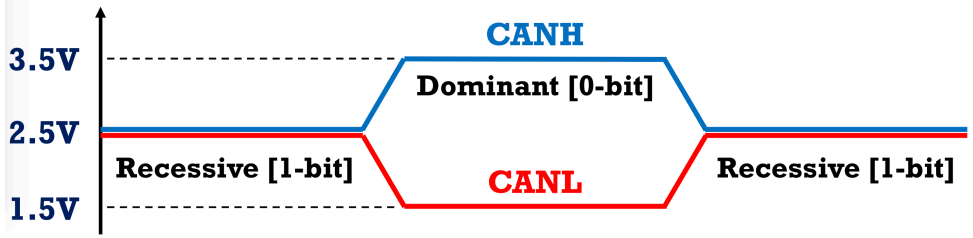
- Transmitter ECU send the frame on CAN BUS
  - Uses an ID (represents priority) instead of ECU address
  - Does not use the ACK field

- Receiver ECUs send an ACK on the CAN BUS

Controller Area Network (CAN)

- ECUs broadcast sensor data via CAN frame/message

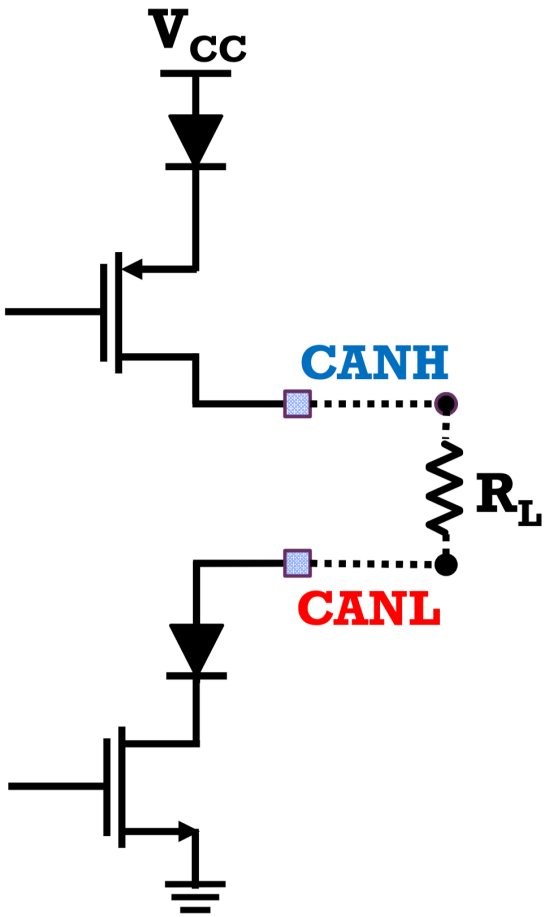
- Frame starts with a 0-bit (dominant bit) preamble
- Frame contains sequence of 0 (dominant) and 1 (recessive) bits



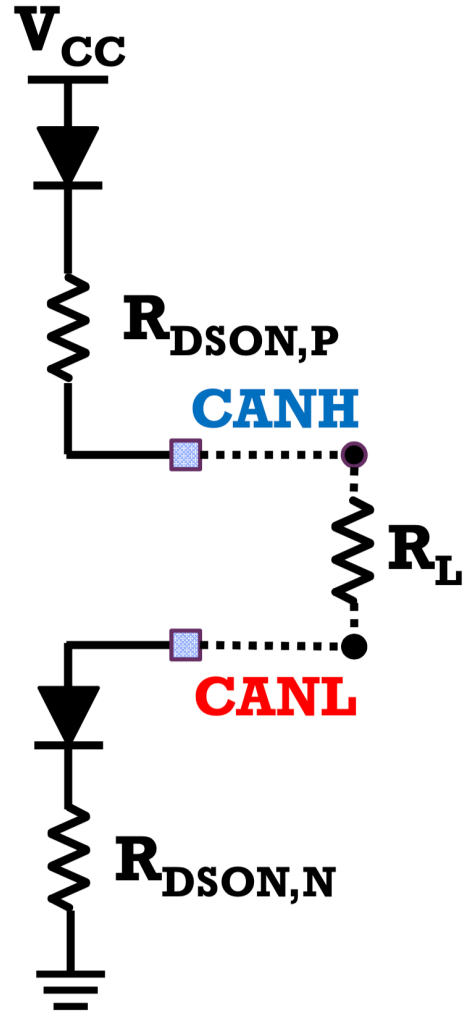
Differential BUS

- ACK: one 0-bit

CAN Transceiver



Transceiver Schematic



When Sending a 0-bit

Viden: Overview

- Different ECUs will have different voltage output
  - Creates voltage profiles (fingerprints)
- Updates fingerprints online
  - Adaptive
- Goal: Attacker Identification
  - Compromised ECUs

System & Threat Model

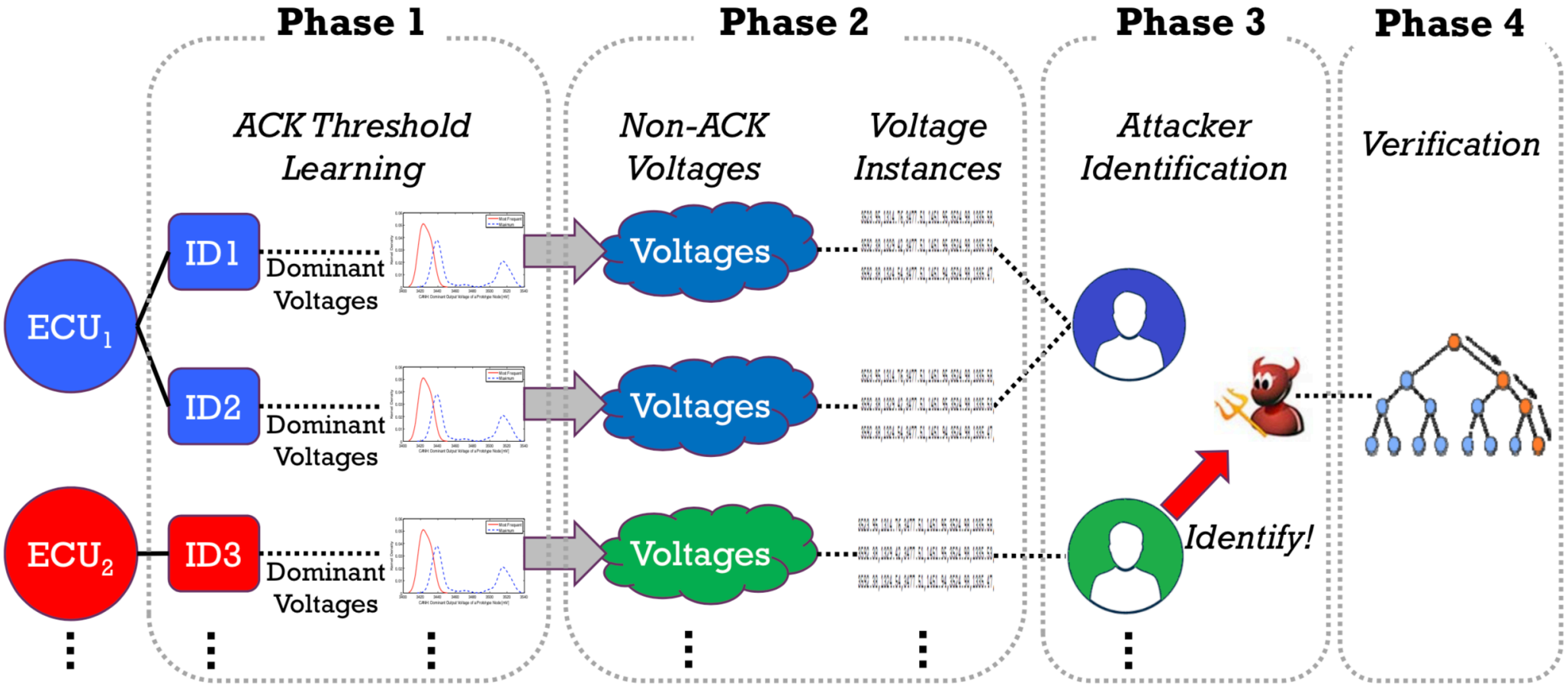
- System Model
  - CAN is equipped with **Intrusion Detection System** and **fingerprinting** (timing and voltage) devices
  - ECU/s are **remotely** compromised
  - For a given message ID, only one ECU is assigned
  - ECU∶ID∶ voltage_profile = 1∶N∶1

- Threat Model
  - Attacker can fabricate ECU (compromised) messages and control the vehicle menuever
  - Attacker can hide identity of the compromised ECU/s
  - Attacker is capable of impersonating ECUs: **Arbitrary** and **Targeted**

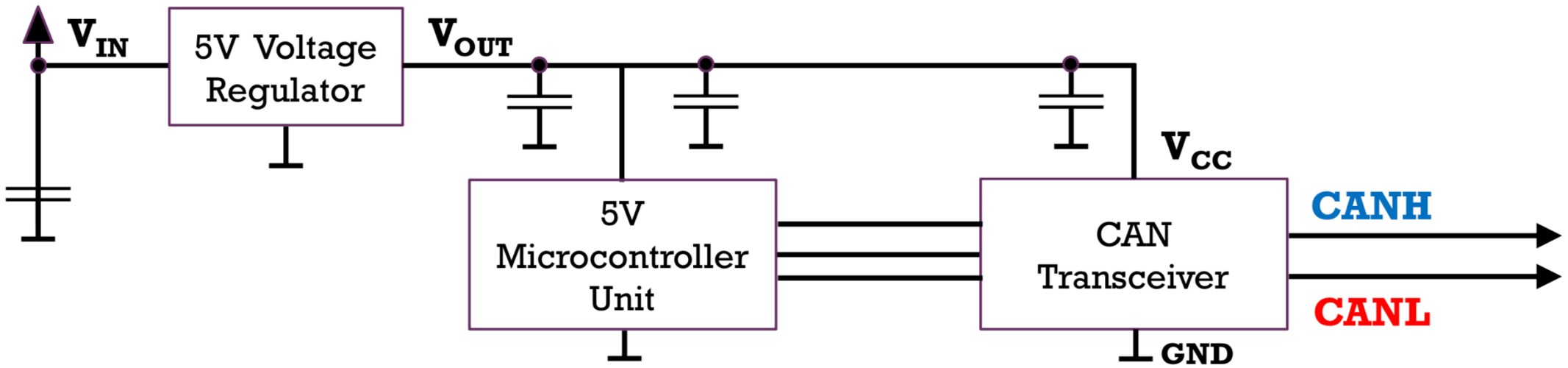
- Attacker types
  - **Naïve**
  - **Timing-aware**:
  - **Timing-voltage-aware**

Viden

- Fingerprints ECU via voltage measurements and achieves attacker identification in four phases



ECU Voltage Characteristics Observations



A typical connection of ECU to CAN

VCC is stabilized using voltage regulator and the capacitors

There exist differences/variations in CAN transceivers' nominal supply voltage, ground voltage, and $R_{DSON,P/N}$ values, especially during the trans- mission of a 0-bit.

ECU Voltage Characteristics Observations

Variations in VCC, ground, and $R_{DSON,P/N}$ result in different ECUs with different CANH and CANL dominant voltages.

ISO11898-2

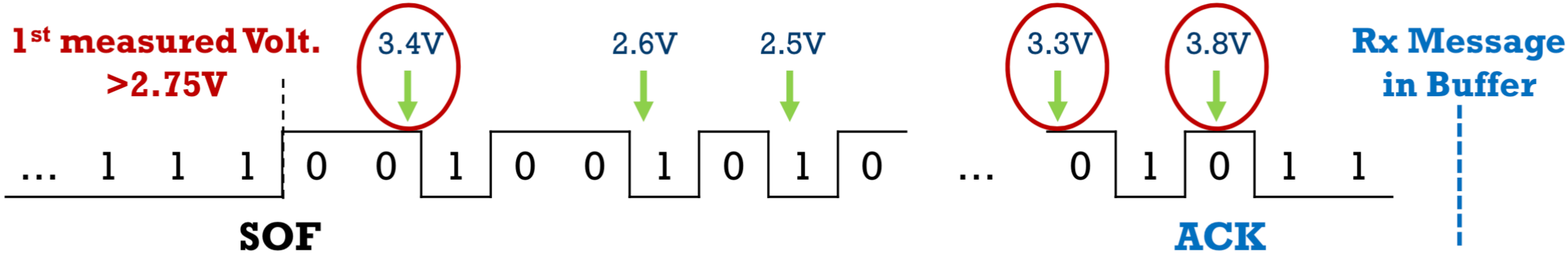
CANH = 2.75~4.5V & CANL: 0.5~2.25

Transient changes in the ECU temperature and driver's input/output affect $R_{DSON,P/N}$, and thus make VCANH and VCANL temporarily deviate in the "opposite" direction.

Transient changes in VCC and ground are significantly smaller than those in CANH and CANL, i.e., their values remain relatively constant.

Viden: Phase 1: ACK Threshold Learning

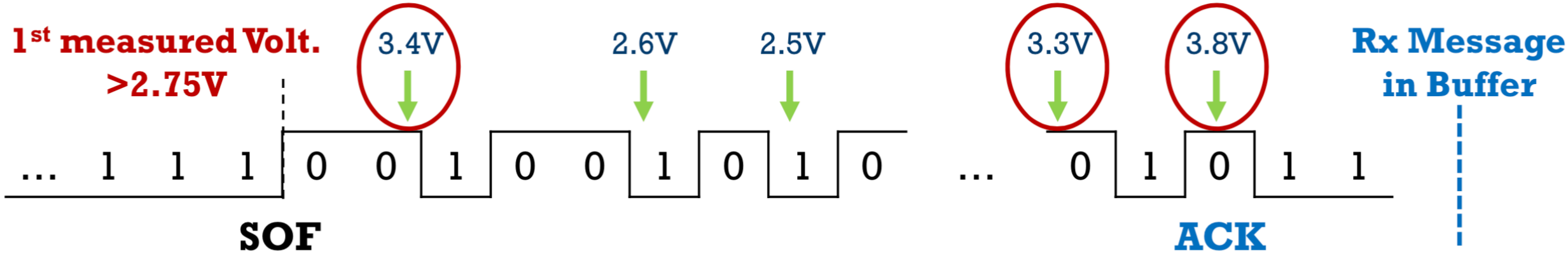
- Measure dominant voltages
  - Ignores any reading if CANH < 2.75 & CANL > 2.25



- Viden's measurement triggers whenever a CANH voltage exceeds 2.75V
- Continues until any message is received into Viden's message buffer

Viden: Phase 1: ACK Threshold Learning

- Extract Non-ACK voltages



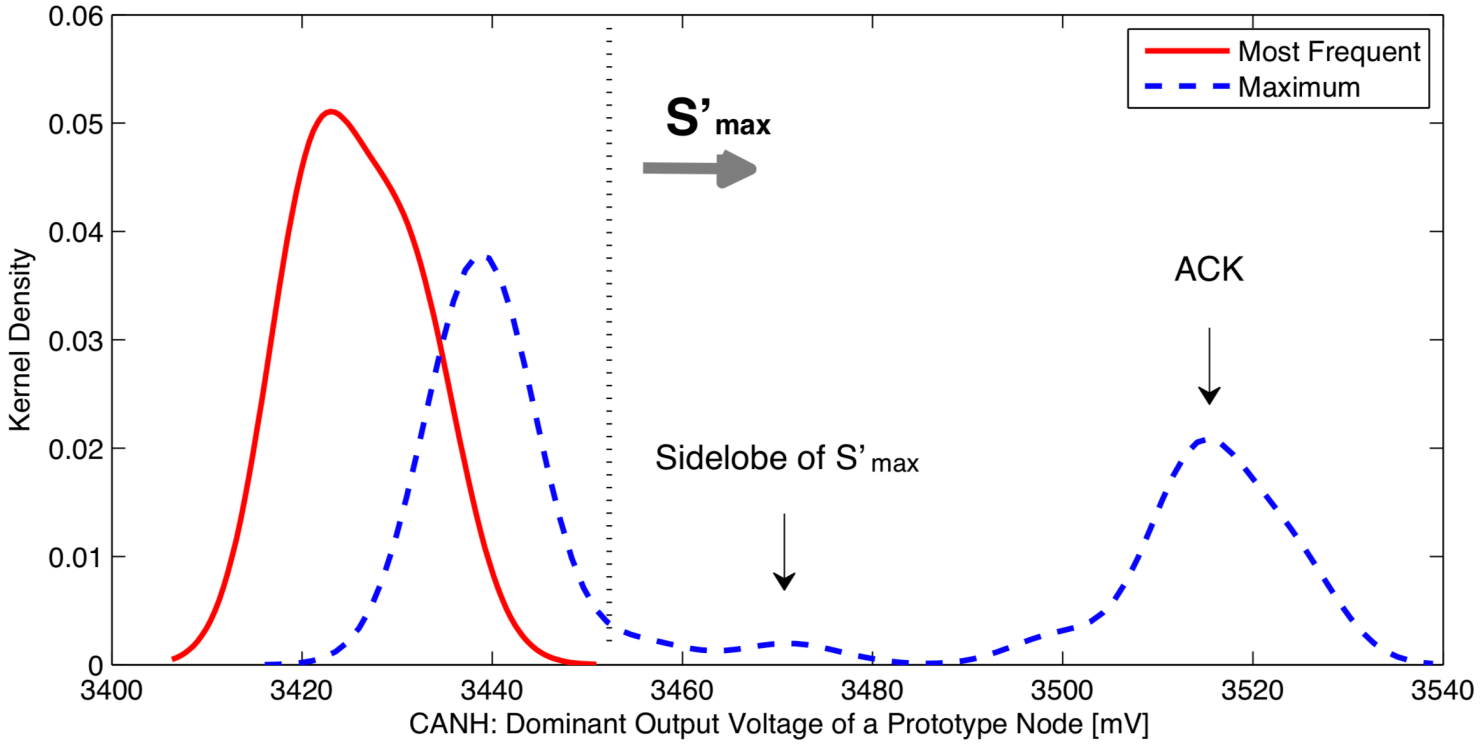
Low probability: Since ACK is only 1 bit long, when measuring dominant voltages during a message reception, most of them would be outputted from the message transmitter.

Different voltage level for ACK: Since ACK responders are connected in parallel and turned on concurrently, when receiving the ACK, the measured voltages are much higher on CANH and much lower on CANL than those when receiving non-ACK bits.

Viden: Phase 1: ACK Threshold Learning

- Collects N rounds of M dominant samples for a given message ID
  - Most frequent set: S_{freq}
  - Maximum/minimum set: $S_{max/min}$

- Derivation of ACK threshold: kernel density function



ACK Threshold

$\Gamma\downarrow ACK\uparrow H$

$\Gamma\downarrow ACK\uparrow L$

Viden: Phase 2: Deriving voltage Instance

- Viden collects dominant voltages continuously for a given message ID
  - 2.75 < CANH < $\Gamma\downarrow ACK\uparrow H$ and 2.25 > CAHL > $\Gamma\downarrow ACK\uparrow H$
  - Only non-ACK voltages

- On each κ (<M) collection of dominant voltages, Viden derives a new voltage instance based on 6 tracking points
  - F1-F2: Most frequent values. Keeps track of the median of CANH and CANL
  - F3-F6: Dispersions: 75^{th}, 90^{th} −percentile of CANH and 25^{th}, 10^{th} −percentile of CANL

Viden: Phase 3: Attacker Identification

- *Cumulative Voltage Derivation (CVD)*: how much the transmitter's dominant voltage changes over time
- Updates CVD of tracking points/features *F1-F6*

Elapsed time since step *(n-1)*

$$CVD_x[n] = CVD_x[n-1] + \Delta[n]\,(1 - \nu_x[n]/\nu_x^*)$$

For feature F_x at step n

Value of F_x at step n

Desired value of F_x

CANH: 3.5, CANL: 1.5

Viden: Phase 3: Attacker Identification

- Suppressing Transient Changes

$$\Psi[n] = \sum_{x=1}^{6} CVD_x[n]$$

Transient changes in the ECU temperature and driver's input/output affect $R_{DSON,P/N}$, and thus make VCANH and VCANL temporarily deviate in the "opposite" direction.

Viden: Phase 3: Attacker Identification

- Voltage Profile
  - $\boldsymbol{\psi}$ represents the consistent factors in voltage instance: V_{CC}, GRND, usual voltage drop across transistors
  - $\boldsymbol{\psi}$ is constant, but different for different ECUs at each time instance, according to the observations
  - Accumulated sum of $\boldsymbol{\psi}$, $\Psi_{accum}[n] = \sum_{k=1}^{n} \Psi[k]$ becomes linear and changes differently for different ECUs

$$\Psi_{accum}[n] = \Upsilon[n]t[n] + e[n]$$

  - Viden uses Recursive Least Square Algorithm to determine voltage profile

Transient changes in VCC and ground are significantly smaller than those in CANH and CANL, i.e., their values remain relatively constant.

Viden: Phase 3: Attacker Identification

- Identifying the attacker
  - IDS systems determines whether there's an attack or not
  - Viden filters voltage outputs only from attack messages
  - Creates an *intrusion voltage profile*
  - Compares with existing profiles

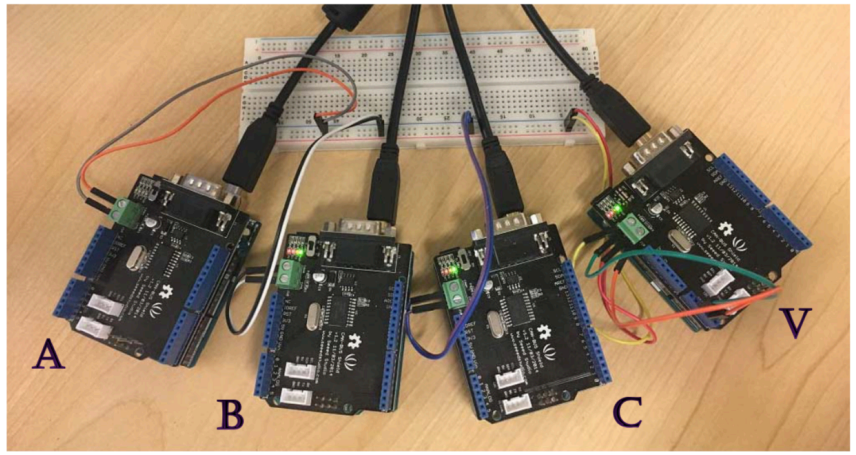
Viden: Phase 4: Verification

- Voltage profile collision
  - Multiple ECUs can have same profile
  - However, narrower set of ECUs to look at

- Targeted impersonation
  - Classifiers are run with (momentary) voltage instances as inputs and F1-F6 as their features

- Phase 4 only complements phase 3, cannot replace!

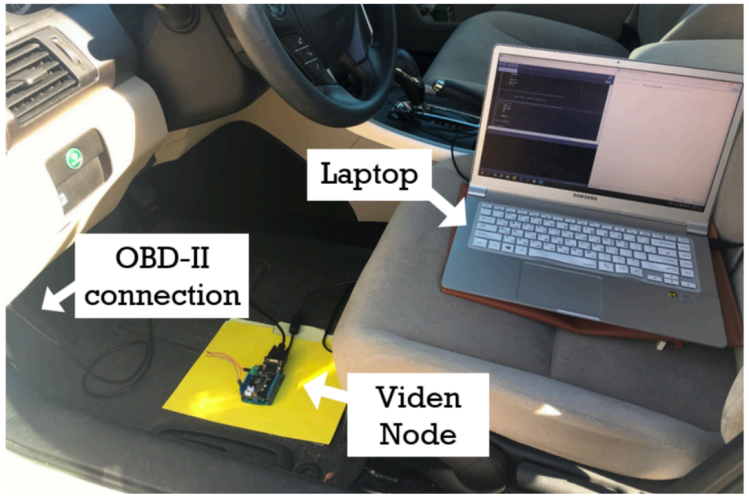
Implementation



(a) CAN bus prototype.

(b) 2013 Honda Accord.

(c) 2015 Chevrolet Trax.



(d) Connection to the vehicle.

False identification rate < 0.2%

Conclusions

- Limitations
  - Cannot har
  - At least one

Questions?