Vehicle Electronics and Architecture (VEA) & Cyber

SECURE HEAVY VEHICLE DIAGNOSTICS

Jeremy Daily, Associate Professor, Colorado State University
Prakash Kulkarni, Systems Engineer, DG Technologies

SYSTEMS ENGINEERING
COLORADO STATE UNIVERSITY

DG TECHNOLOGIES
Vehicle Network Solutions

Who has access to the vehicle?

Diagnostic and maintenance systems are frequently connected to trucks.

Brake Control Module

Engine Control Module

Transmission Control Module

120Ω

120Ω

Typical J1939 Vehicle

Diagnostic Port

Vehicle Diagnostic Adapter

RP1210 Driver

1 — USB

2

3

250 KBPS J1939

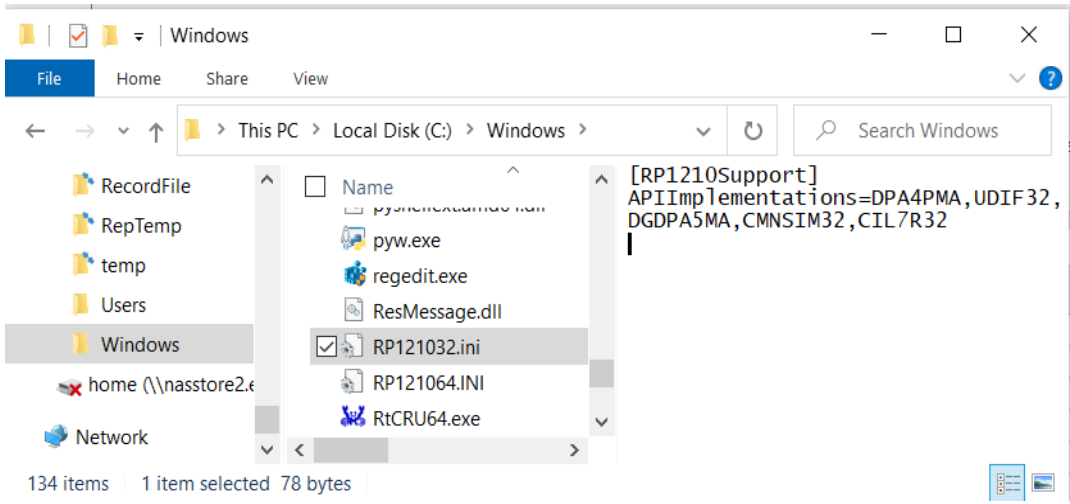
Attack Vectors

1. Malicious PC Software

2. DLL Driver

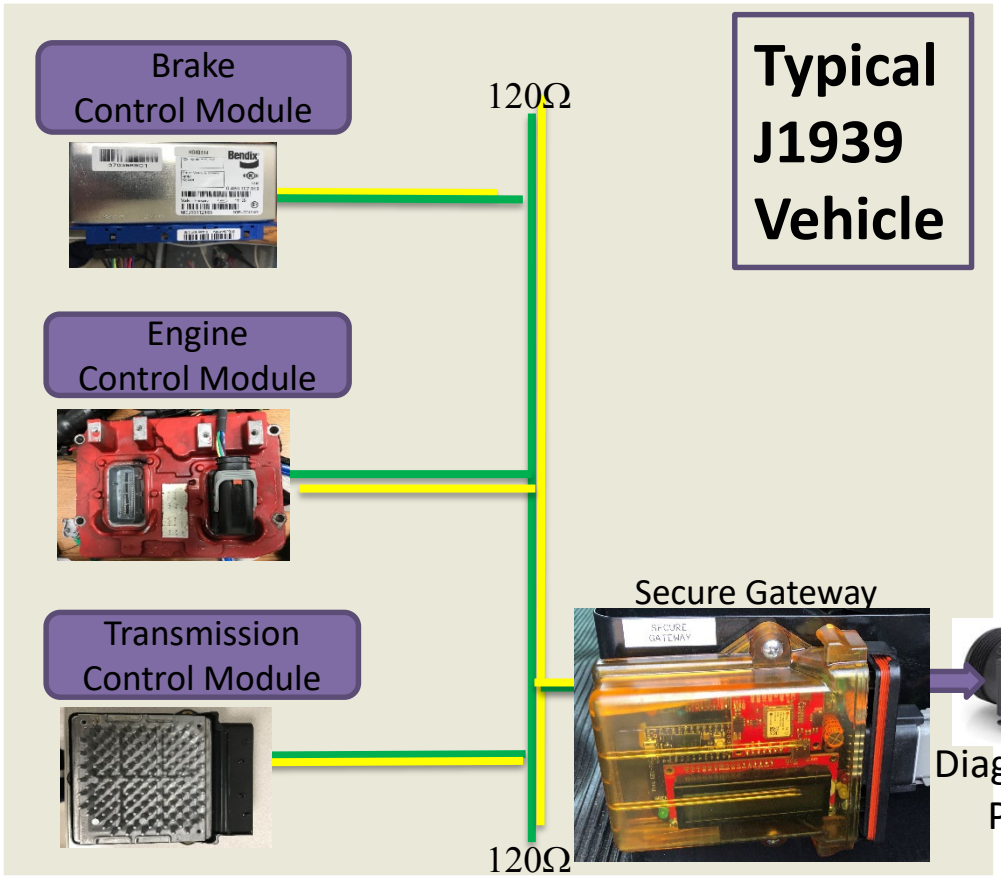
3. VDA Firmware

- Unknown and uncontrolled diagnostic systems are trusted on the J1939 network
- Technicians frequently connect to the Internet to update software
- Diagnostic Software is written to communicate through the RP1210 API.
  - Started with Windows 3.1
  - User selects driver with information in an INI file



- Remote attackers only need access to the diagnostics PC to perform a cyberattack.

Approach: Use a Secure Gateway



Attack Vectors

1. **Malicious PC Software**
2. **DLL Driver**
3. **VDA Firmware**

System Requirements

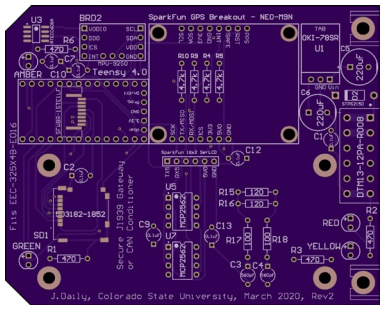
- **SR1:** Maintain compatibility with existing J1939 Architectures
- **SR2:** Provide a solution that is agnostic of the vehicle diagnostic adapter.
- **SR3:** Enable offline diagnostics sessions.
- **SR4:** Store CAN Data Logs based on event triggers.

Cybersecurity Requirements

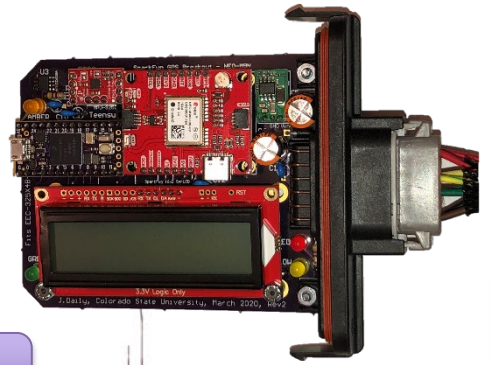
- **CR1:** Use unique key material so any key leakage does not compromise other systems.
- **CR2:** Use secure storage hardware for private key storage on the vehicle.
- **CR3:** Use existing best practices for cryptographic implementations
  - AES-128 for symmetric encryption
  - EEC P256
  - New or untested ciphers shall not be used.
- **CR4:** Any sensitive key material should be encrypted for storage.

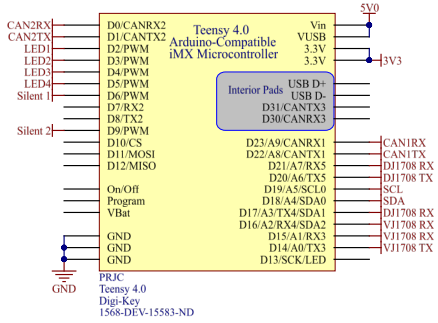
Hardware Prototype Design and Realization

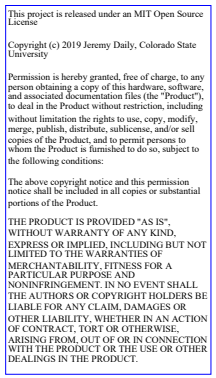
Printed Circuit Board

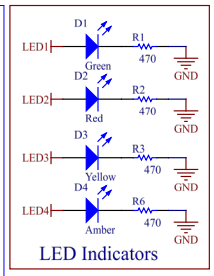


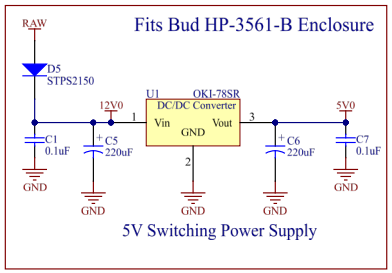
Assembled Circuit Board

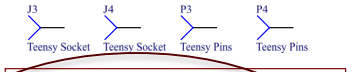


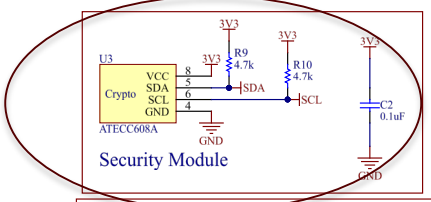


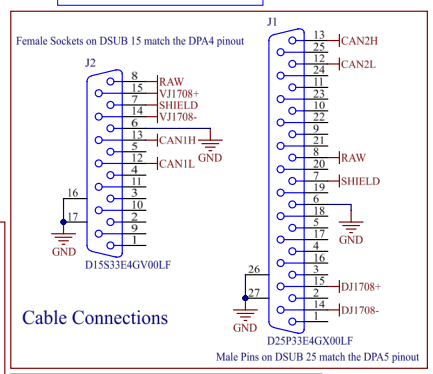


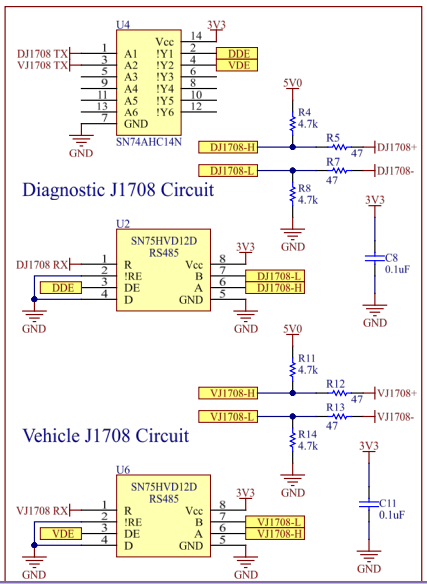


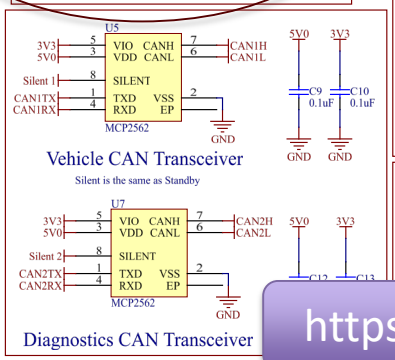


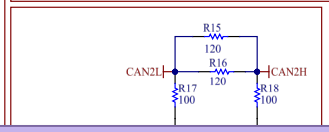






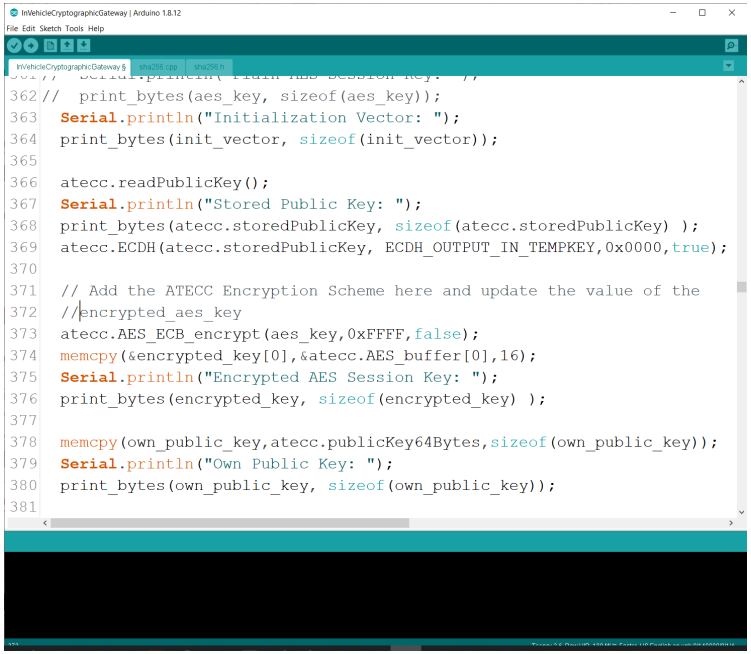




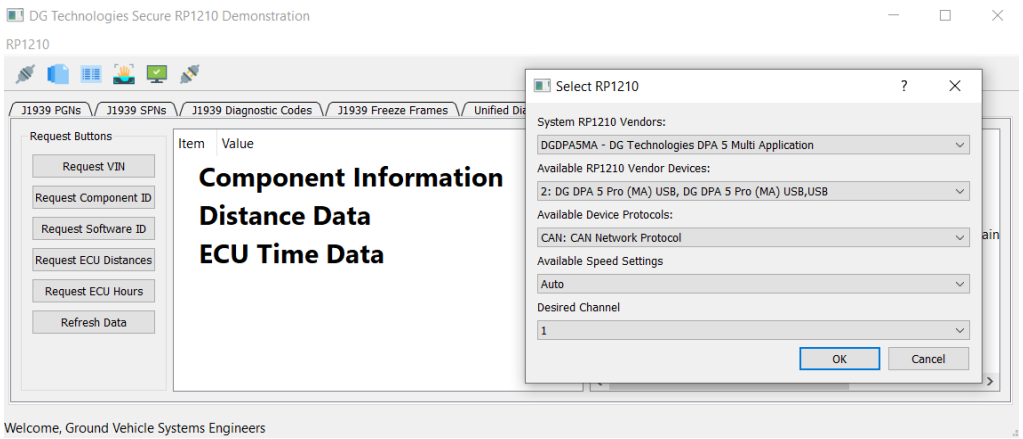


https://github.com/SystemsCyber/CANWatermarking

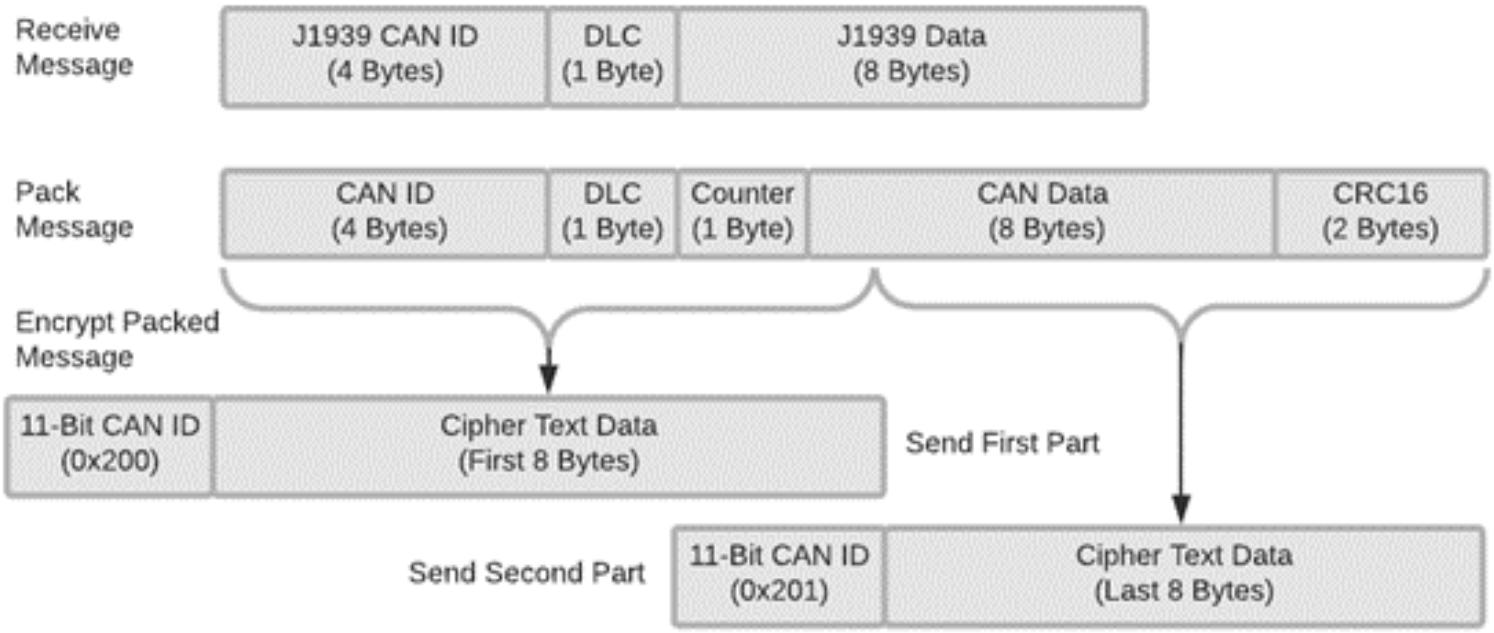
- Embedded Firmware
  - Provision
  - Write Configuration
  - Executable Firmware



- Proof of Concept Secure RP1210 App
  - Python/PyQt5
  - Requests for API Connection

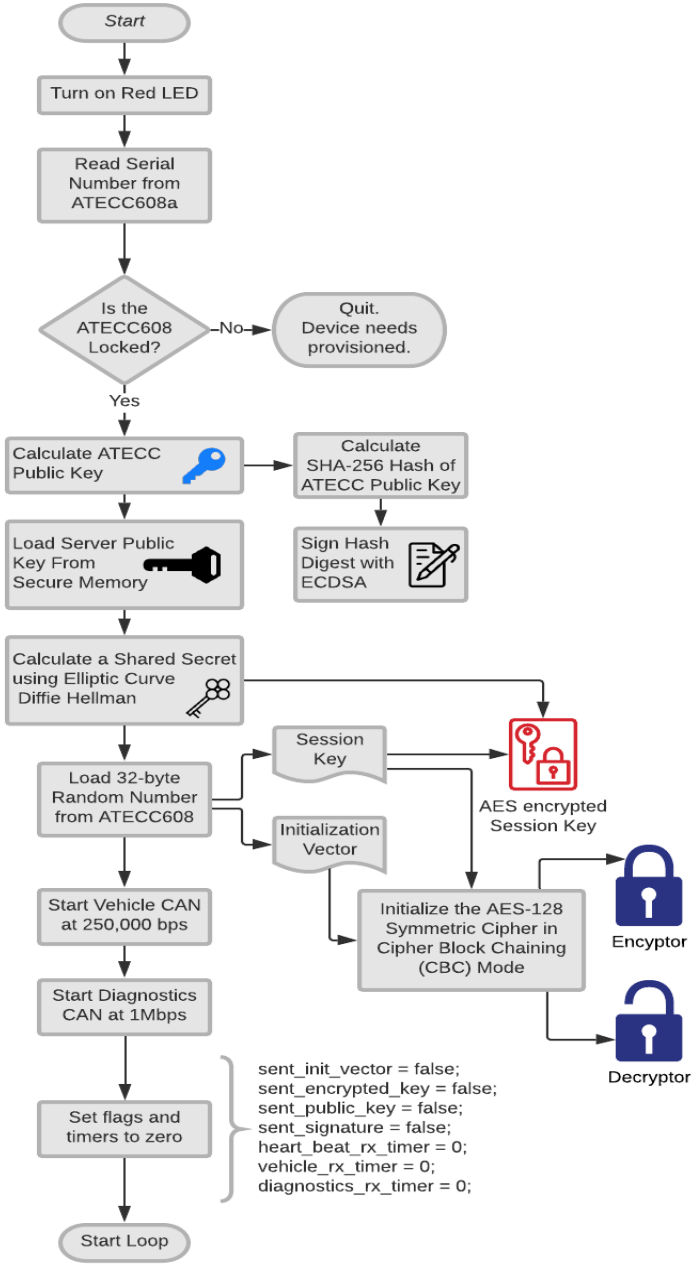


Core Concept: Encrypt CAN from J1939 to Application



Separating a J1939 Frame into two 8-byte frames for AES-128 blocks

Encryption Setup



Make sure device is provisioned;
If provisioned, use the hardware security module for setting up ephemeral symmetric session keys.

Cryptographic Gateway

1. Generate Random
   - Initialization Vector
   - Session AES Key
2. Encrypt Session Key
3. Wait for Crytpo Data Request
4. Start Session
5. Reflect Heartbeat

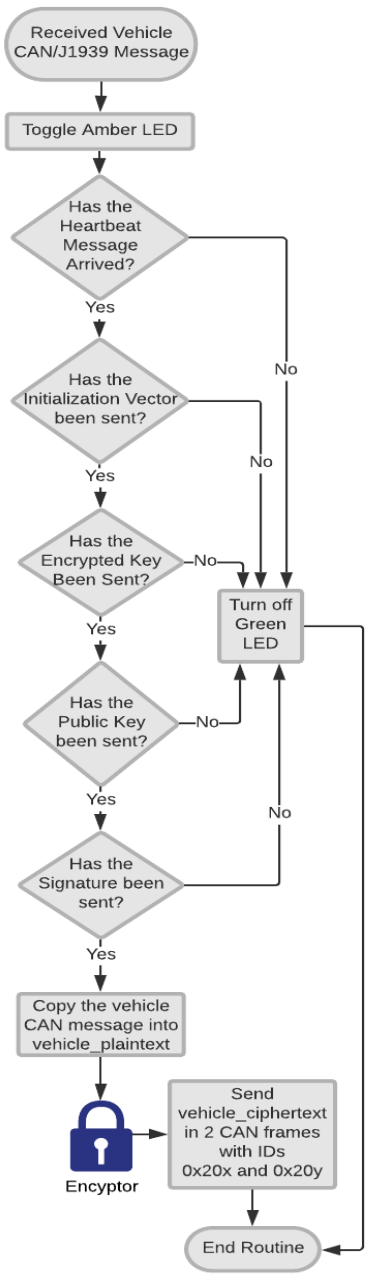
- Encrypt Vehicle Traffic
- Decrypt Application (VDA) Traffic

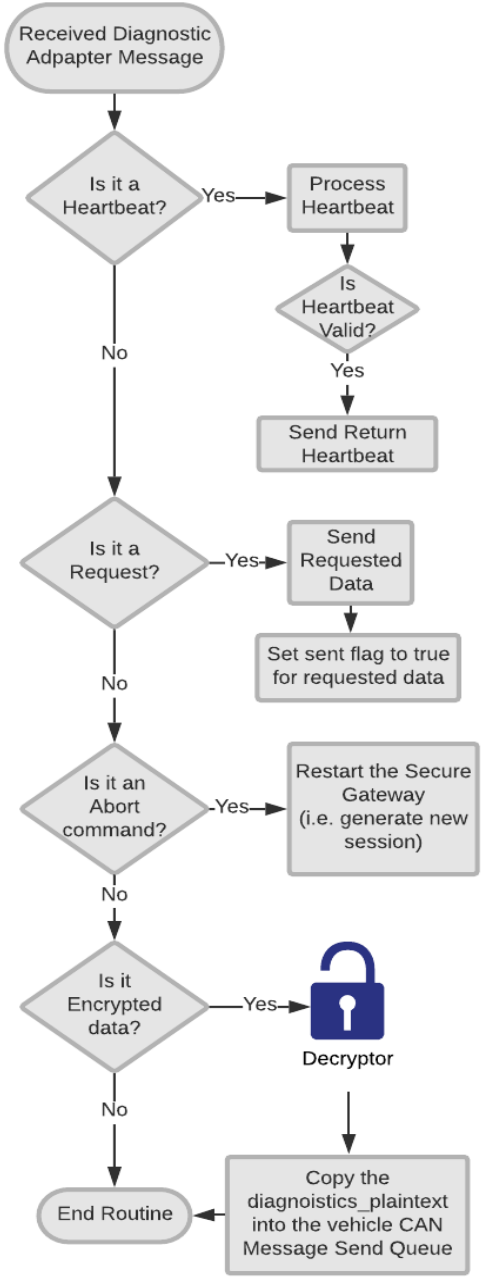
PC Diagnostics App

1. Connect RP1210 with CAN at 1Mbps
2. Request Crypto Data
   - Gateway Public Key
   - Initialization Vector
   - Encrypted Public Key
3. Start Session
4. Send Heartbeat

- Decrypt Gateway Traffic
- Encrypt Application Traffic

Interrupt Service Routines





CAN messages are handled using ISRs

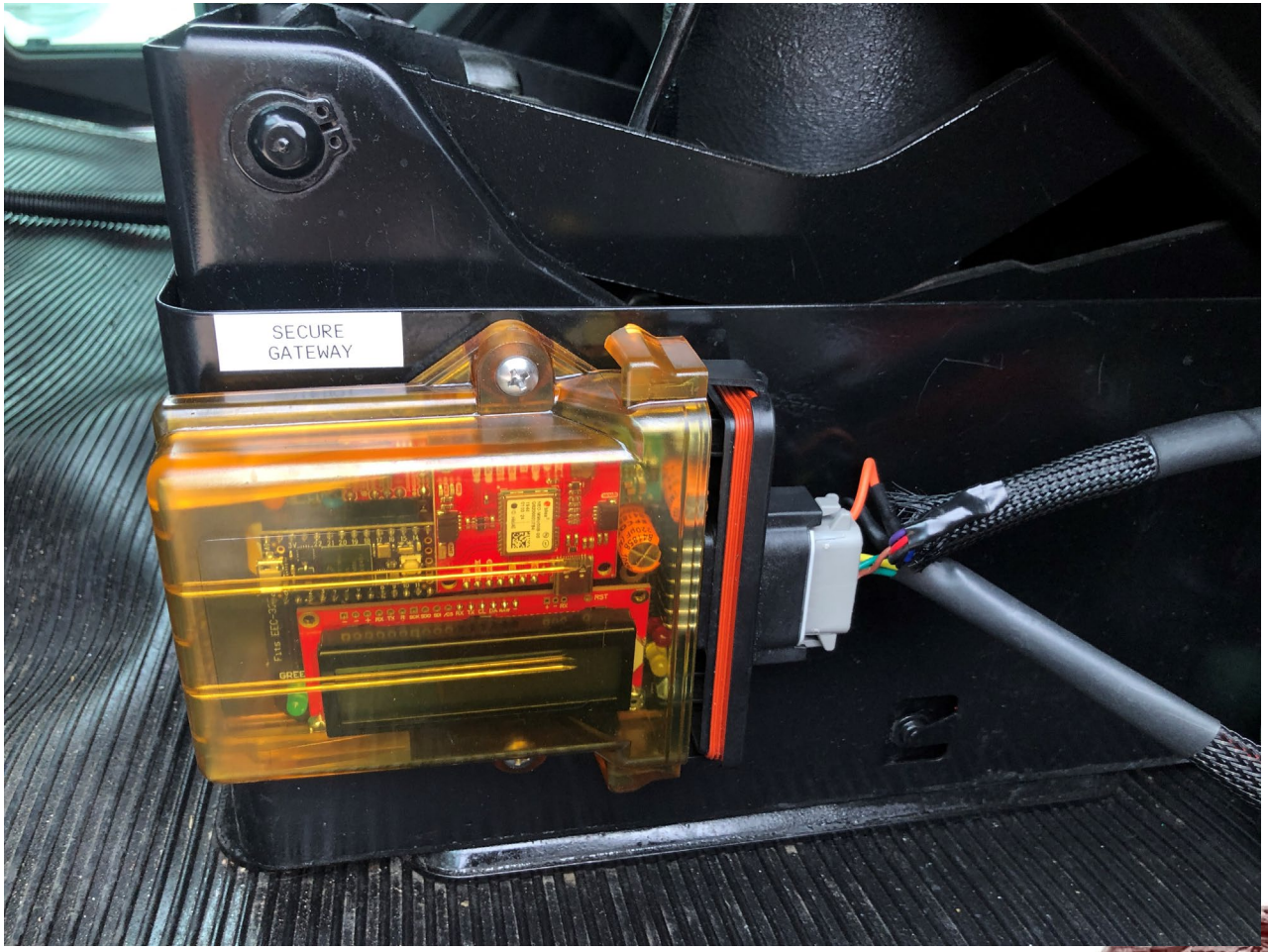
Special messages:

- Heartbeat
- Requests
- Abort
- Initialization Vector
- Encrypted Ephemeral Session Key
- Encrypted data

Test Truck

2014 Kenworth T270 Class 6
PACCAR PX7 Engine
Allison Transmission

Secure Gateway Module Installation



- **Attacks were implemented with two entry points**
  1. DLL Driver
     - Corrupt Shim DLL intercepts messages between the Diagnostic PC and the VDA, changes the data and passes the changed data
     - Shim DLL works in conjunction with the vendor-supplied DLL
  2. VDA Firmware
     - Compromised VDA with modified firmware that does not faithfully transfer the message traffic between the vehicle network and the PC
- **Attacks are representative of Man-in-the-Middle attacks**
- **Attacks target two types of messages**
  - Single Frame J1939 Messages
  - Multiframe messages using the J1939 Transport Protocol
- **A representative PC Application provides the Diagnostic User Interface**


Cardinals
DELL

Example of Compromising a Multi-Packet Message

DG Tech

RP1210

J1939 PGNs | J1939 SPNs | J1939 Diagnostic Codes | J1939 Freeze Frames | Unified Diagnostic Services | Componer

Request Buttons

- Request VIN
- Request Component ID
- Request Software ID
- Request ECU Distances
- Request ECU Hours
- Refresh Data

Item	Value
Component Information	
Cab Controller - Primary on J1939	
VIN	1HVBBAAM1YH314841*
Engine #1 on J1939	
VIN	1HVBBAAM1YH314841*
Distance Data	
Cab Controller - Primary on J1939	

- Attacks sought out VIN and source address patterns to affect traffic
- Followed well understood J1939 and RP1210 definitions
- All data passing through RP1210 can be logged and exfiltrated
  - Reveal operational readiness

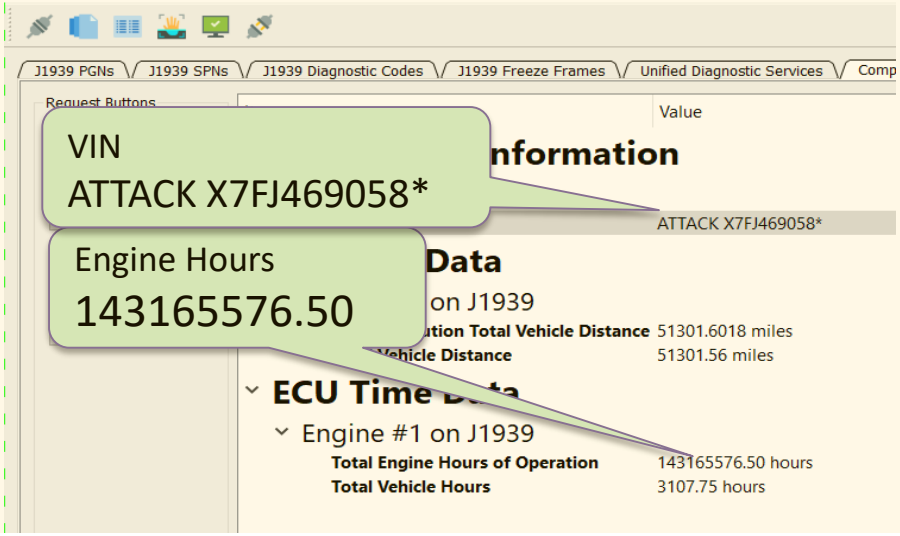
- When AES encrypted, all session data is patternless
  - No pattern matching attack could work
  - Exfiltration is also meaningless

Correct Engine Control Module Data

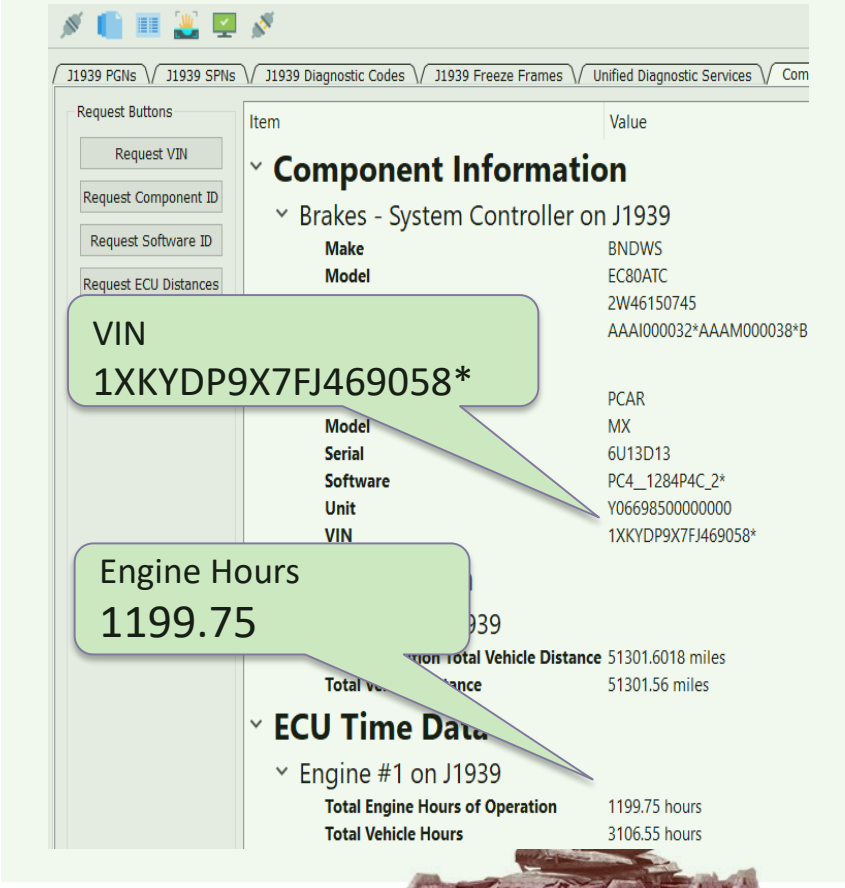
Vehicle VIN: 1XKYDP9X7FJ469058*

Engine Hours of Operation: 1199.75

Attack Scenario



Secure Diagnostics Attack Mitigation



- Vulnerabilities exist in heavy vehicle diagnostics systems.
  - VDA Firmware Updaters
  - Shim DLLs to wrap authentic drivers
- Connections of diagnostic services and hardware must be included in threat assessments.
- Access to VDA hardware and firmware show unique attacks
- Proposed Solution Highlights
  - PC Diagnostic Application communications are encrypted
  - Share ephemeral symmetric session keys using ECDH
  - Secure private key storage with the ATECC608 hardware security module
  - Solution can be retro-fitted to existing vehicles in the field
  - Open source physical hardware prototype