Mobile Vehicle Security Bridge

By: sdmay23-14

Ryan Scehovic, Ryan Campbell, Cody Stricker, Levi Jansen, Josue Torres, Drake Ridgeway, Riley Lawson

Intro

Our goal is provide security for CAN messages being sent between ECUs by using CAN FD, an upgraded version of CAN.

Controller Area Network (CAN) is a message protocol that allows Electronic Control Units (ECUs) to communicate.

Methodology

When designing our CAN FD frame we had to consider how many CAN frames (11 bytes each) we wanted it to contain, as well as how much room we needed to allocate towards security related values.

For security values we choose to use a CMAC tag to help ensure message authenticity/integrity and a freshness value to allow us to keep track of messages we've already seen.

Storing these values in our CAN FD frame allows us to validate the message on the receiving end to see if it was tampered with or replayed.

Overview

CAN	VS	CAN FD
• PGN = 2 bytes • Source Address = 1 byte • Data payload = 8 bytes		• PGN = 2 bytes • Source Address = 1 byte • Data payload = 64 bytes

Using CAN FD gives us 8x as much data payload to work with. We decided our CAN FD frame will consist of 5 CAN messages (11 bytes each), a freshness value, and a CMAC tag.

Impact

This project shows that it is possible to secure CAN messages by using the upgraded CAN FD version. The real world impact of this is that it increases the safety of vehicle operators in their daily tasks.

The CAN protocol is commonly used in cars, tractors, and boats, so possible vulnerabilities could impact millions of people.

Implementation

Packing CAN FD Frame

On the right you can see 5 CAN frames being received and then a CAN FD frame at the bottom being sent out from a Bridge.

When a CAN frame is received the PGN + Source Address (3 bytes) are added to the FD frame (pink highlight). After that it adds in the 8 data bytes (yellow highlight). This repeats until 5 CAN messages are added. Then it adds the CMAC tag (orange highlight) and freshness value (green highlight).

```
Timestamp: 1682784133.434768    ID: 001e001e    X
      DLC:  8    00 40 00 00 00 00 00 00      Channel: vcan0
Timestamp: 1682784133.455083    ID: 0006ef00    X
      DLC:  8    64 15 17 f0 c6 23 d8 21      Channel: vcan0
Timestamp: 1682784133.455260    ID: 0006ef00    X
      DLC:  8    64 15 17 f0 c6 23 d8 21      Channel: vcan0
Timestamp: 1682784133.475654    ID: 0006ef00    X
      DLC:  8    64 15 18 f0 c6 23 d8 21      Channel: vcan0
Timestamp: 1682784133.475811    ID: 0006ef00    X
      DLC:  8    64 15 18 f0 c6 23 d8 21      Channel: vcan0
Timestamp:        0.000000    ID: 00abc123    X     F
    DLC: 64    1e 00 1e 00 40 00 00 00 00 00 00 00 06 ef 00 64
 15 17 f0 c6 23 d8 21 06 ef 00 64 15 17 f0 c6 23 d8 21 06 e
f 00 64 15 18 f0 c6 23 d8 21 06 ef 00 64 15 18 f0 c6 23 d8
21 08 08 83 4d 00 00 00 00 c9
```

Message Validation

An FD frame is accepted when the freshness value and CMAC tag pass the validation checks.

Counter Check: freshness value has already been seen or message is stale

CMAC Check: CMAC is recalculated after received to see if any of the data bytes were tampered with

```
 Message Accepted:   Timestamp: 1682784131.510908    ID: 00abc123    X     F
    DLC: 64    06 ef 00 64 15 1f f0 3f 24 d8 21 06 ef 00 64 15 10 f0 2c 24 d8 21 0
6 ef 00 64 15 10 f0 2c 24 d8 21 1e 00 1e 00 40 00 00 00 00 00 00 13 ef f0 64 19 81
 ff ff ff ff 00 34 b2 c1 a1 00 00 00 00 a1    Channel: vcan1
 Message Fails Counter check:   Timestamp: 1682784131.510921    ID: 00abc123    X
    F          DLC: 64    06 ef 00 64 15 1f f0 3f 24 d8 21 06 ef 00 64 15 10 f0 2c
 24 d8 21 06 ef 00 64 15 10 f0 2c 24 d8 21 1e 00 1e 00 40 00 00 00 00 00 00 13 ef
f0 64 19 81 ff ff ff ff 00 34 b2 c1 a1 00 00 00 00 a1    Channel: vcan1
 Message Fails CMAC check:   Timestamp: 1682784131.545252    ID: 00abc123    X
 F          DLC: 64    06 ef 00 64 15 11 f0 2c 24 d8 21 03 f0 05 84 ff ff ff ff ff
 ff ff 06 ef 00 64 15 12 f0 13 24 d8 21 06 ef 00 64 15 12 f0 13 24 d8 21 1e 00 1e
00 40 00 00 00 00 00 00 f8 f8 d6 42 00 00 00 00 a2    Channel: vcan1
 Message Fails Both Counter and CMAC:   Timestamp: 1682784131.545336    ID: 00abc1
23    X     F          DLC: 64    06 ef 00 64 15 11 f0 2c 24 d8 21 03 f0 05 84 ff
ff ff ff ff ff ff 06 ef 00 64 15 12 f0 13 24 d8 21 06 ef 00 64 15 12 f0 13 24 d8 2
1 1e 00 1e 00 40 00 00 00 00 00 00 f8 f8 d6 42 00 00 00 00 a2    Channel: vcan1
```

Results

Our program can process through about 10,000 messages per second. These messages are processed and validated as they are routed through the system. In our testing 100% of replayed or invalid messages are caught during this time.

Conclusion

As security becomes more prominent, a security implementation for the CAN bus becomes critical. Given the challenges that implementing security on a CAN bus imposes, a solution is possible. Utilizing the methods mentioned, CAN messages can be secured within the low overhead constraints.