Secure Communication for Intra-Vehicular CANFD Network

Yutian Gui, Ali Shuja Siddiqui, Jim Plusquellic* and Fareena Saqib
Dept. of Electrical Engineering, University of North Carolina at Charlotte and University of New Mexico*
ygui@uncc.edu, asiddiq6@uncc.edu, jimp@ece.unm.edu, fsaqib@uncc.edu

MOTIVATION AND INTRODUCTION

Electronic Control Units (ECU) are enriched devices that control mechanical parts through control signals. ECUs are connected via Controller Area Network (CAN) bus to communicate with each other. Recently, the security risk of intra-vehicular communication has emerged due to the unprotected architecture of ECU and CAN bus. Some works have performed successful attack on in-car communication system. Therefore, there is a critical need to enhance the security of intra-vehicular communication. Trusted platform module (TPM) is a specialized chip which is designed to enhance the security of computing devices. TPM provides the functionality of Root of Trust (RoT) realized by a series of operations, including key generation, key storage, and authentication.

In this demo, we introduce the vulnerability of the communication between different ECUs and propose a novel secure communication model to protect the intra-vehicular communication from attacks aimed at the message frames such as replay and message spoofing attacks. The demo includes fast speed latest Controller Area Network Flexible Data-rate (CAN-FD) standard.

Keywords—CAN bus; ECU; secure communication; hardware security

PROPOSED ARCHITECTURE AND WORKFLOW

In the demo, we demonstrate a full-covered solution of secure communication including secure key generation and storage, data encryption, data verification and data signing. In this demo, all the message frames are transmitted in CAN-FD standard. The proposed secure framework with enriched ECUs connected over CAN bus is shown in Figure 1.

Each ECU node implements a TPM. TPM provides the functionality of key generation and secure storage for keys. All the public keys are shared between trusted nodes before communication. 3.Once deployed in field, each node encrypts and signs (optional) message by TPM module before sending. The encrypted and signed message is encapsulated in an extend CANFD frame. On the receiver side, the encrypted data is verified with sender's public key. If the verification fails, the sender will be considered as a corrupted node and the incident will be logged.

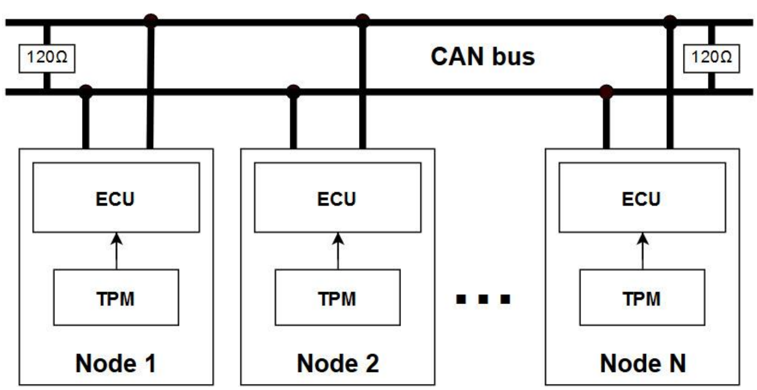


Fig. 1. Proposed Framework

HARDWARE USED AND SETUP

Hardware used in our demo is as follows:

- Microcontrollers based components communicating over CANBus.
- CAN with Flexible Data-Rate (CAN-FD) controller and transceiver.
- Trusted Platform Module (TPM).

The setup of proposed demo is shown in Figure 2. In this demo, all the components are connected using two SPI interfaces on the Raspberry PI. All the secure functionalities are provided by TPM.

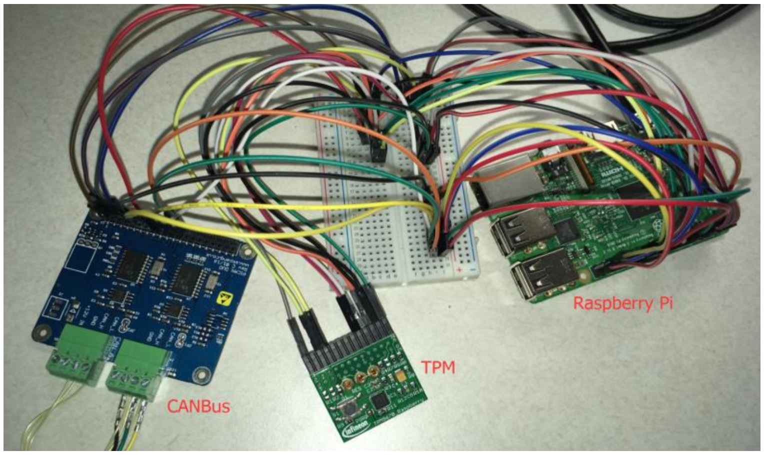
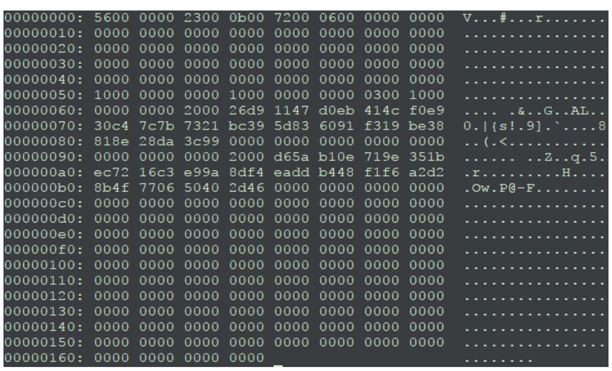


Fig. 2. Hardware Setup

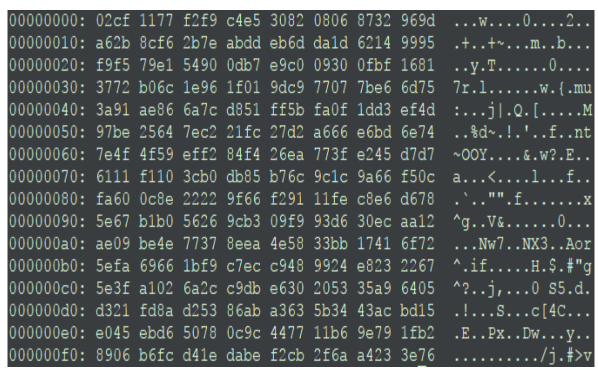
OBSERVABLES

In our demo, we will firstly demonstrate message transmission based on CAN-FD standard between two unprotected CANBus nodes. We will show the secure communication between two authorized nodes based on the proposed framework step by step. The whole process includes message encryption and authentication by TPM, traffic control on the sender side by thresholding module and the access control on both of sender side and receiver side. The One example of public key and encrypted data frame is shown in figures 3 and figure 4.



```
00000000: 5600 0000 2300 0b00 7200 0600 0000 0000  V...#...r.......
00000010: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00000020: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00000030: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00000040: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00000050: 1000 0000 0000 1000 0000 0000 0300 1000  ................
00000060: 0000 0000 2000 26d9 1147 d0eb 414c f0e9  .... .&..G..AL..
00000070: 30c4 7c7b 7321 bc39 5d83 6091 f319 be38  0.|{s!.9].`....8
00000080: 818e 28da 3c99 0000 0000 0000 0000 0000  ..(.<...........
00000090: 0000 0000 0000 2000 d65a b10e 719e 351b  ...... ..Z..q.5.
000000a0: ec72 16c3 e99a 8df4 eadd b448 f1f6 a2d2  .r.........H....
000000b0: 8b4f 7706 5040 2d46 0000 0000 0000 0000  .Ow.P@-F........
000000c0: 0000 0000 0000 0000 0000 0000 0000 0000  ................
000000d0: 0000 0000 0000 0000 0000 0000 0000 0000  ................
000000e0: 0000 0000 0000 0000 0000 0000 0000 0000  ................
000000f0: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00000100: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00000110: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00000120: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00000130: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00000140: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00000150: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00000160: 0000 0000 0000 0000                      ........
```

Fig. 3. Public key



```
00000000: 02cf 1177 f2f9 c4e5 3082 0806 8732 969d  ...w....0....2..
00000010: a62b 8cf6 2b7e abdd eb6d da1d 6214 9995  .+..+~...m..b...
00000020: f9f5 79e1 5490 0db7 e9c0 0930 0fbf 1681  ..y.T......0....
00000030: 3772 b06c 1e96 1f01 9dc9 7707 7be6 6d75  7r.l......w.{.mu
00000040: 3a91 ae86 6a7c d851 ff5b fa0f 1dd3 ef4d  :...j|.Q.[.....M
00000050: 97be 2564 7ec2 21fc 27d2 a666 e6bd 6e74  ..%d~.!.'..f..nt
00000060: 7e4f 4f59 eff2 84f4 26ea 773f e245 d7d7  ~OOY....&.w?.E..
00000070: 6111 f110 3cb0 db85 b76c 9c1c 9a66 f50c  a...<....l...f..
00000080: fa60 0c8e 2222 9f66 f291 11fe c8e6 d678  .`.."".f.......x
00000090: 5e67 b1b0 5626 9cb3 09f9 93d6 30ec aa12  ^g..V&......0...
000000a0: ae09 be4e 7737 8eea 4e58 33bb 1741 6f72  ...Nw7..NX3..Aor
000000b0: 5efa 6966 1bf9 c7ec c948 9924 e823 2267  ^.if.....H.$.#"g
000000c0: 5e3f a102 6a2c c9db e630 2053 35a9 6405  ^?..j,...0 S5.d.
000000d0: d321 fd8a d253 86ab a363 5b34 43ac bd15  .!...S...c[4C...
000000e0: e045 ebd6 5078 0c9c 4477 11b6 9e79 1fb2  .E..Px..Dw...y..
000000f0: 8906 b6fc d41e dabe f2cb 2f6a a423 3e76  ........../j.#>v
```

Fig. 4. Encrypted data frame