Exploring Flexible Data-Rate CAN and Encryption

JT Stancil and Duy Van
Advisor: Dr. Jeremy Daily

THE UNIVERSITY of TULSA
Student CyberTruck Experience

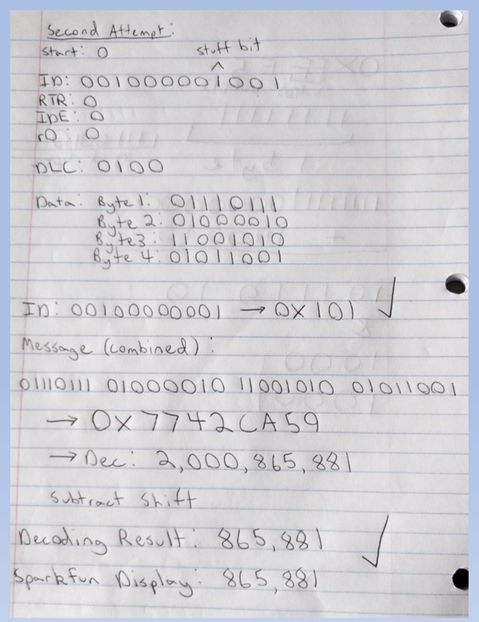
Decoding CAN

Purpose:
- Understand the structure of CAN messages and how data is transmitted across CAN
- Decode a CAN message captured on an oscilloscope
- Write Python code to parse CAN data

Hardware:
- Oscilloscope
- Any active CAN bus

Testing:
- Decoded CAN frames and compared to their known values/information

Skills Learned:
- Python Programing
- CAN-FD Structure Analyzing
- Oscilloscope Reading



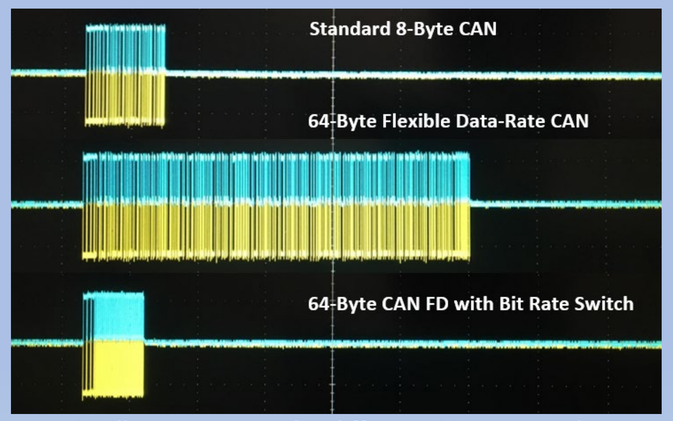
Decoding CAN frames by hand

Flexible Data-Rate (CAN-FD)

Purpose:
- Interface with CAN-FD using PEAK-System Application Programming Interface (API)
- Utilize CAN-FD fast speed (up to 12Mbit/s) with Bit Rate Switch (BRS) and larger data transfer rate (up to 64 bytes/frame) to potentially enable encryption

Hardware:
- PCAN PCI Express FD
- Oscilloscope
- D-Sub 9-pin cable

Testing:
- Connect the 2 channels on PCAN PCI using the D-Sub cable
- Transmit and receive messages using Python script



Oscilloscope traces for different CAN protocols

```
byte transport0[8] = {32,0,16,3,totalBlocks,0xD6,0xFE,0};
byte transport [3][8];
for( int i = 0; i < 3; i++){
 transport[i][0] = i+1;
 for( int j = 1; j < 8; j++){
 transport[i][j]= bob.ciphertext[j-1+(7*i)];
 spot++;
 if(spot>16){
  transport[i][j] = 0xFF;
 }//end if
 }//end for
}//end for
 spot = 0;
 //send those values
 for (int i = 0;i<8;i++) txmsg.buf[i]=transport0[i];
 Can0.write(txmsg);
 delay(3);
 for(int i = 0; i < 3; i++){
  //Serial.print("Packet ");
  //Serial.print(i+1);
  //Serial.print(":      ");
  for(int j = 0; j < 8; j++){
   txmsg.buf[j] = transport[i][j];
   //Serial.print(txmsg.buf[j]);
   //Serial.print(" ");
  }// end for
  Can0.write(txmsg);
```

Writing to the transport layer

```
//starts first half of DH
void beginDH(){
    Serial.println("Diffie-Hellman key exchange:");
    Serial.print("Generate random k/f for Alice ... ");
    Serial.flush();
    unsigned long start = micros();
    Curve25519::dh1(alice_k, alice_f);
    unsigned long elapsed = micros() - start;
    Serial.print("elapsed ");
    Serial.print(elapsed);
    Serial.println(" us");
   Serial.println("Alice's K values...");
    for(int i = 0; i<32; i++){
      Serial.print(alice_k[i]);
      Serial.print(" ");
      if((i+1)%7 == 0){
        Serial.println();
      }
    }
    Serial.println();
}//end beginDH
```

Encrypt and decrypt functions

```
SendEncoderWithEncryption
288
289 void decryptBlock(BlockCipher *cipher,TestVector *test){
290   Serial.println("Decryption ... ");
291   cipher->decryptBlock( test1,test ->ciphertext );
292   Serial.println("the decrypted block");
293   for(int j = 0; j<16; j++){
294
295     recievedData[(j)+(16*(blockNumber-1))] = test1[j];
296     Serial.print(char(recievedData[(j)+(16*(blockNumber-1))]));
297     test->ciphertext[j] = ' ';
298   }
299 }
300 void encryptBlock(BlockCipher *cipher,  TestVector *test){
301
302
303   Serial.println("Encryption ... ");
304
305   cipher->setKey(aes_key, cipher->keySize());
306   cipher->encryptBlock(buffer, test->plaintext);
307   for(int i = 0; i<16; i++){
308     Serial.print(char(buffer[i]));
309     Serial.print(" ");
310     bob.ciphertext[i] = buffer[i];
311   }
312   Serial.println();
313 }
314 void listenForSerial(){
```

Assigning Diffie Hellman values

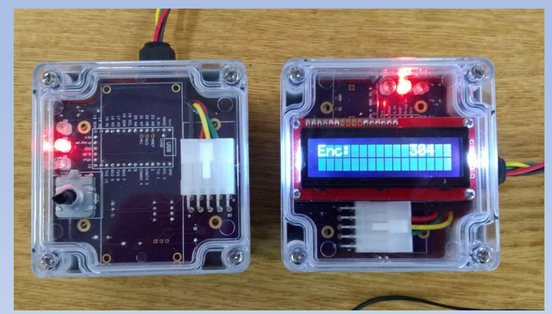
Encrypted CAN

Purpose:
- Introduction to common encryption standards
- Determine the possibility and viability of encrypting CAN messages
- Develop circuit boards that will allow for encryption between units and some visual indication of communication

Hardware:
- Teensy 3.2

Testing:
- Used known/tested AES libraries
- Performed a secured Diffie Hellman key exchange
- Sent encrypted CAN data (the encoder position) across the secure tunnel and decrypted that message at the receiving node
- Displaying the decrypted encoder position on LCD monitor

Skills Learned:
- Electronics design, C programming
- Basic cryptography



Prototype encoder and decoder for experimenting

NMFTA National Motor Freight Traffic Association, Inc.

PEOPLENET A TRIMBLE COMPANY

GEOTAB management by measurement