A PK Authentication Scheme for Controller Area Networks

Nick Bravo, Matthew Chang, Skanda Koppula

Outline

1. **Motivating Problem**
2. **Introduction to CAN**
3. **Our PK CAN-Authentication Scheme**
    a. The PKI
    b. Hash-Chain
    c. Frame Splitting
4. **Analysis and Testing**
    a. Testing Framework
    b. Latency & Memory
    c. Thoughts on Security
5. **Previous Work**
6. **Applause**

Why?

1. Every electric car in the US:

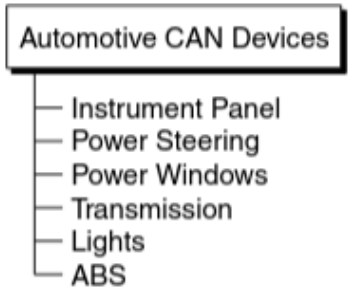
   OBD2 diagnostic port connecting to an internal Controller Area Network (CAN) network

Why?

1. Every electric car in the US:

   OBD2 diagnostic port connecting to an internal Controller Area Network (CAN) network

2. CAN network orchestrates the vehicle

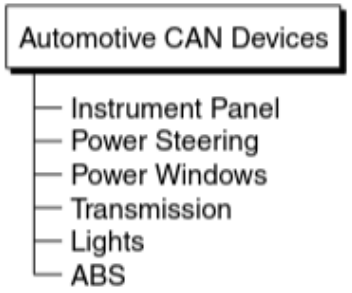


Why?

1. Every electric car in the US:

   OBD2 diagnostic port connecting to an internal Controller Area Network (CAN) network

2. CAN network orchestrates the vehicle



3. It's already insecure [DefCon 21]

How to Hack Your Mini Cooper: Reverse Engineering CAN Messages on Passenger Automobiles

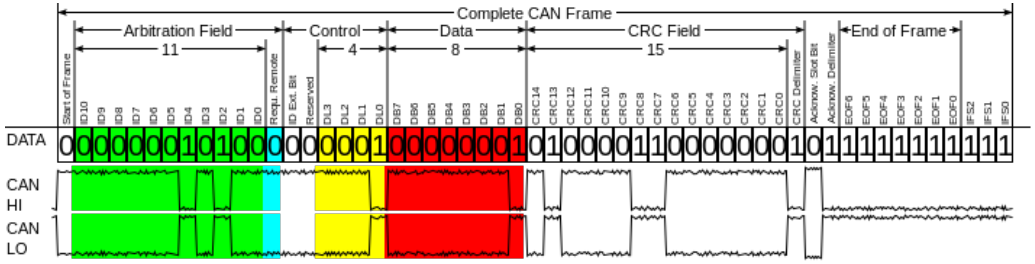
Jason Staggs

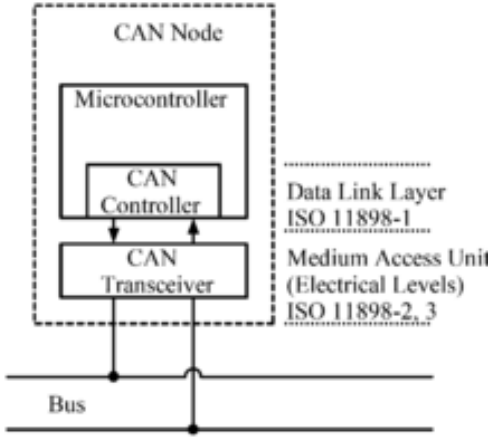
University of Tulsa

On **HACKADAY**

How to do CAN in 2 seconds

- Fixed size serial bit frames on bus
- Arbitration based on ID
  - Can’t mess around too much with ID
- **Standard**/Extended
- No source/source authentication!





What we want

- Backward compatible: only mess with payload (mostly)
- Low overhead: latency, memory for key storage, etc.
- Flexible network: add 3rd party devices to listen to bus
- Add authentication standards: our own special Bus Unforgeability Experiment

Bus-Frame Unforgeability Experiment
On a bus-based network with N nodes $n_1, n_2, ...n_N$, we define our scheme to be *bus unforgeable* if for all PPT adversaries A, that A succeeds after the following steps is negligible:

1. We choose i key-pairs, according to our key generation algorithm: $(SK_i, VK_i) \leftarrow KeyGen$. Limit $i \leq n$. Every node is assigned one SK_i.

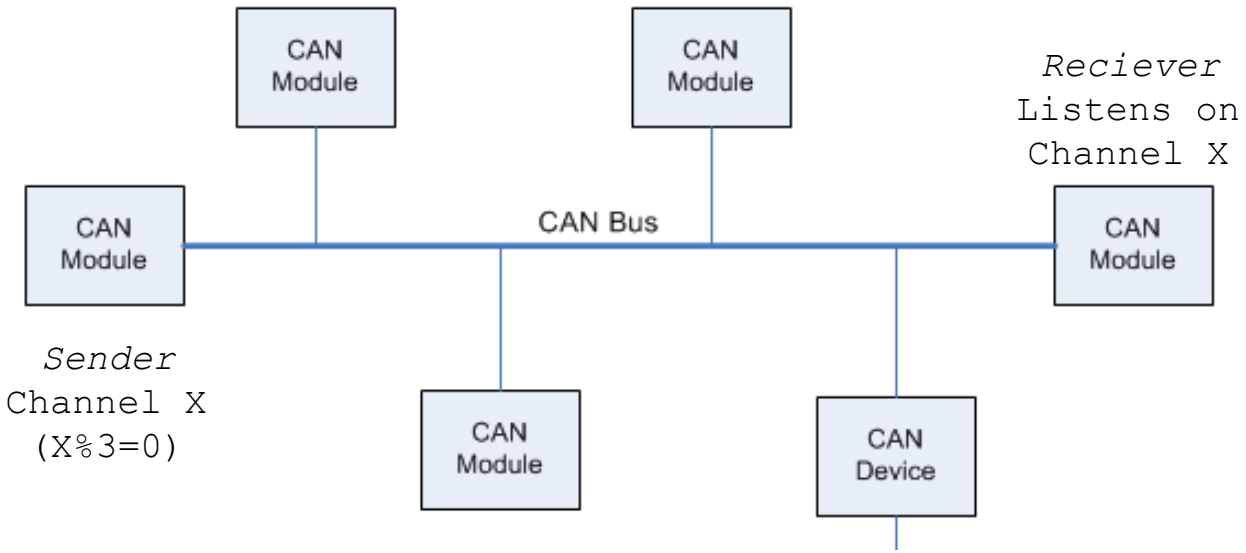
...

CANAuthScheme: Big Picture

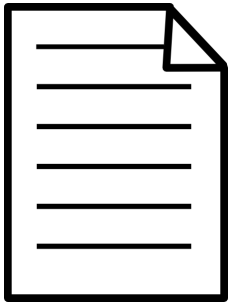
Each node:

- Unique ID
- Public Key/Private Key Pair

Establish channels of authenticated communication on bus



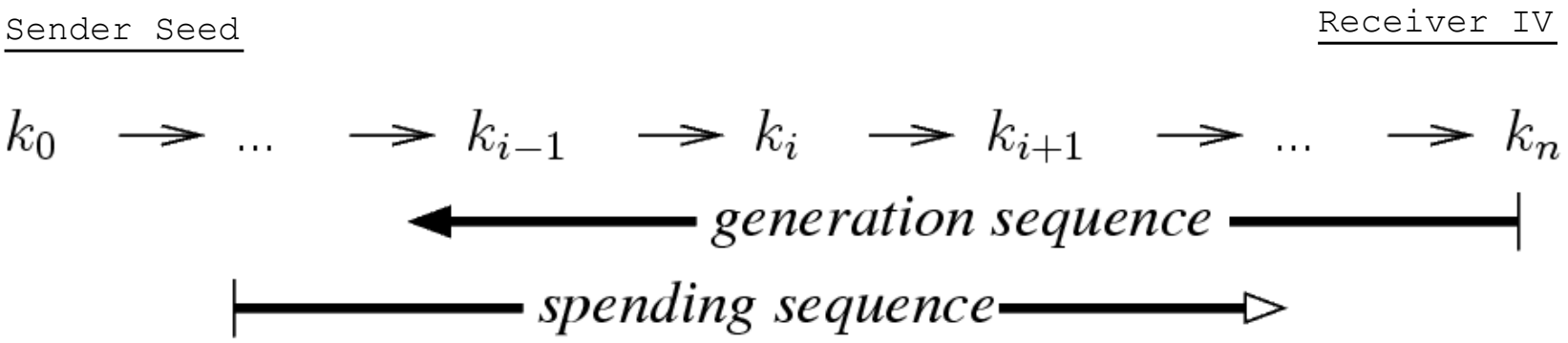
Trusted Directory
of PKs/ID/Channel



CANAuthScheme: Channels

Series of Auth Tags in an HMAC chain

broadcasted along with every message with message ID X



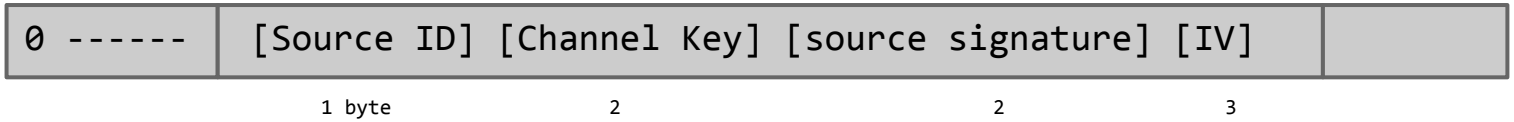
straightforward source verification process:

```
Keyed-Hash(k_current) = k_prev
```

CANAuthScheme: Nitty Gritty of Channels

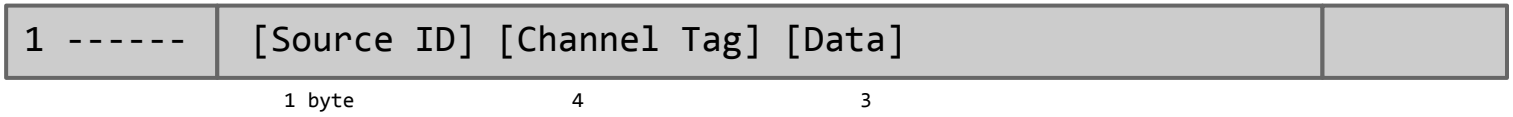
Initialization

Send Reciever IV, HMAC Key **in signed, marked frame**



x 64

Data Transmission



Why HMAC?

Replay same Tag-Data might be used again in future chain!

CANAuthScheme: Nitty Gritty Other Things

- **Not Purely HMAC: XOR Message**

  and other trickery

CANAuthScheme: Nitty Gritty Other Things

- **Not Purely HMAC: XOR Message**

  and other trickery

- **Channel ID assignment**

  Published random algorithms to ensure fairness on bus given CAN arbitration rules

CANAuthScheme: Nitty Gritty Other Things

- **Not Purely HMAC: XOR Message**

  and other trickery

- **Channel ID assignment**

  Published random algorithms to ensure fairness on bus given CAN arbitration rules

- **Where to store PKs?**

  Either in chip memory or query master for PK (for super-lightweight nodes!)

CANAuthScheme: Nitty Gritty Other Things

- **Not Purely HMAC: XOR Message**
  and other trickery

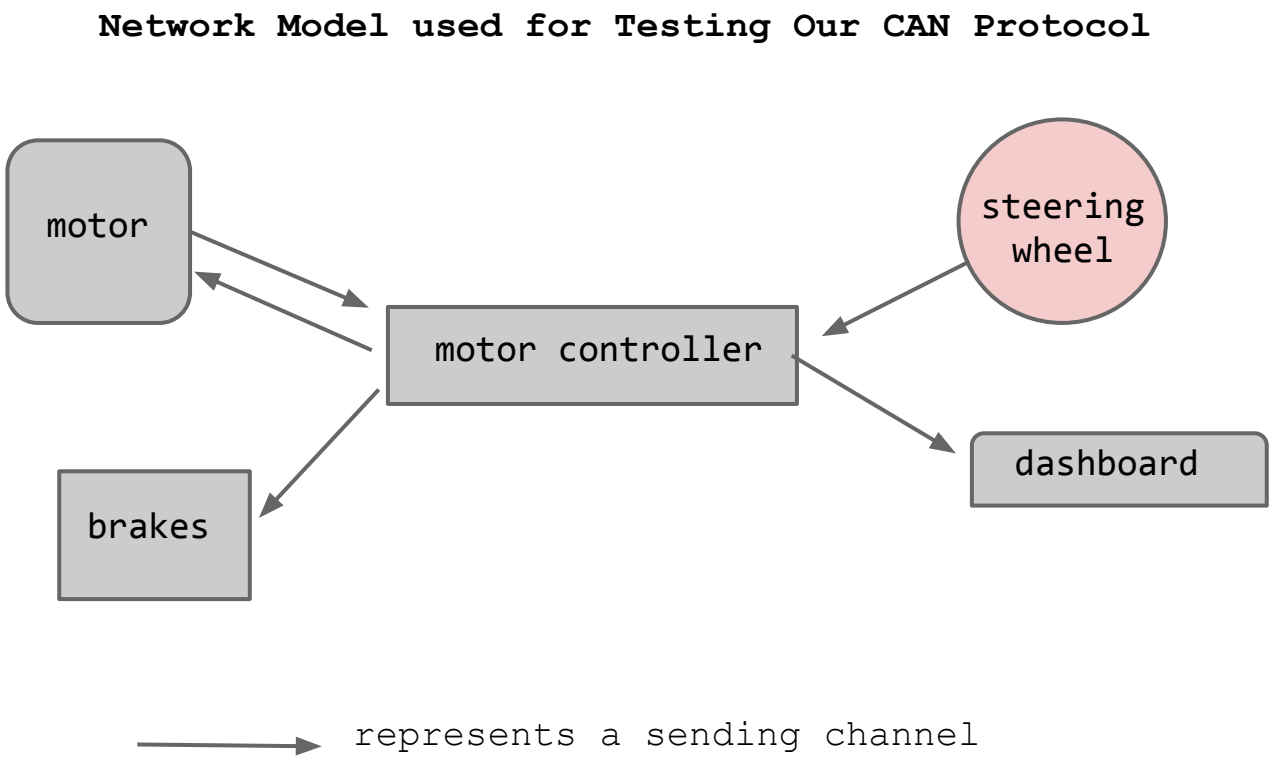
- **Channel ID assignment**
  Published random algorithms to ensure fairness on bus given CAN arbitration rules

- **Where to store PKs?**
  Either in chip memory or query master for PK (for super-lightweight nodes!)

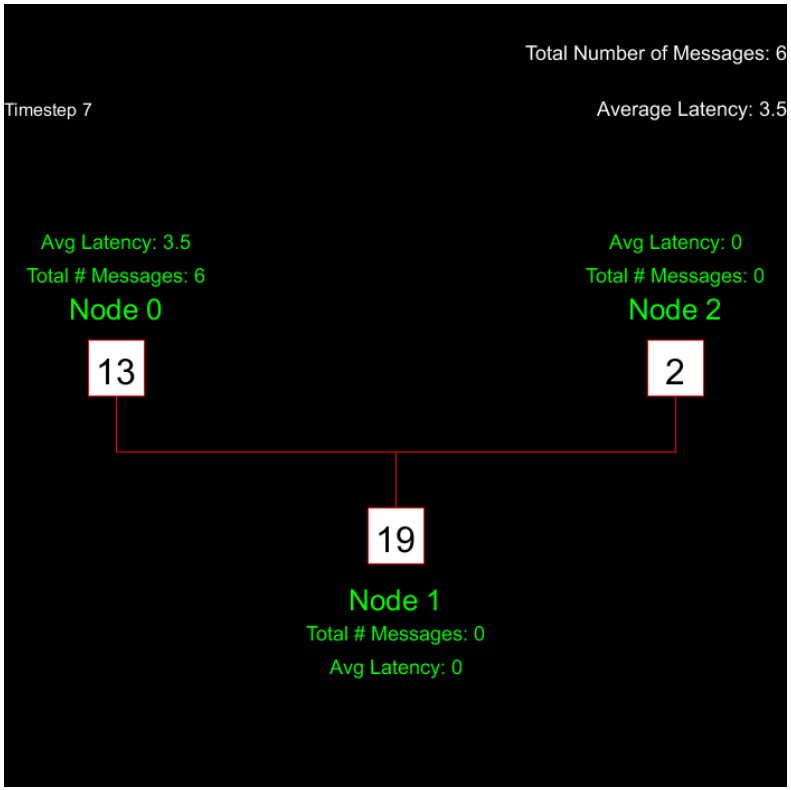
- **Frame Structure: really a tradeoff**

Enough theory, now practice

- Software simulation of CAN bus network [Python, Processing]
- No hardware prototypes (yet!)



If Our Protocol Demo Works Hopefully You'll See More Than A Screenshot



High Level Analysis

- Memory
  - $O(n^2)$ system key storage
  - Roughly $<160n$ bytes per node
- Security
  - Forging message is 2^{-32} probability
  - Adversary with $1/20$th bus usage expected time to brute force roughly 6 years
- Congestion
  - One time cost of 64 messages per node
  - Authentication doubles the traffic

Some numbers

Average Latency

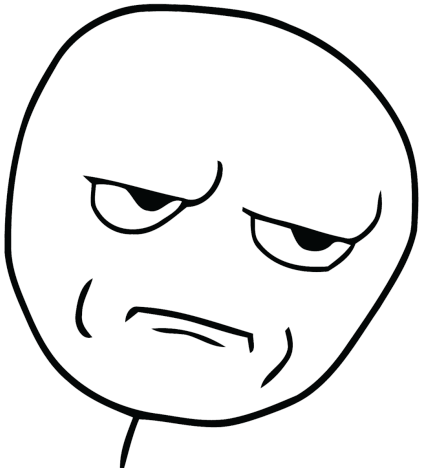
Total Messages sent

Previous Work (!)

LiBr-CAN

TESLA

…



- Purely shared key systems
- No support for 3rd party listeners
- Requires # keys ~ n choose n/2
- All published within the last three years
- No complete source authentication

Problems (for everyone)

DOS attacks (!)

Added memory requirements
not acceptable for pure-circuit nodes

Overhead latency and traffic very high

Questions?

skoppula@mit.edu
nbravo@mit.edu
m_chang@mit.edu

6.857 Spring 2015