Integrity for Car-Computing

A cryptographic vision for integrity in vehicle networks

Eran Tromer
The first vehicle computer
D-17B *Minuteman I* guidance system
The first vehicle computer
D-17B *Minuteman I* guidance system
In-car integrity

• Modern cars contain dozens of Electronic Control Units
• Can you trust them?
  – Hardware supply chain
  – Bad software
  – Errors
  – Bad updates
  – Attacks
Example: engaging ABS

PAM ECU
Parking Aid Module

PSCM ECU
Power Steering Control Module
07 36: 0B 34 00 44 ...
"SJB ECU: Program firmware"
07 36: 03 B1 7E 80 ...
"SJB ECU: Crash"

SJB ECU
Smart Junction Box

ABS ECU
Anti-Lock Brake System
07 60: 04 B1 00 3C FF ...
"ABS ECU: Engage brakes fully"

PCM ECU
Powertrain Control Module

Transmission

Parking brake switch

Brake pedal switch

Accelerator pedal position sensor

Brake pedal position sensor

See [Miller Valasek 2013]
Approach: proof-carrying data

“My message is [...] and here’s a proof that I computed it correctly based on a correct message from the PAM ECU and signed sensor data.”

“My message is [...] and here’s a proof that I computed it correctly based on a correct message from the Transmission ECU and signed sensor data.”

“My message is BRAKE and here’s proof that it was computed correctly based on all of the above.”

“My message is [...] and here’s a proof that I computed it correctly.”
Integrity via Proof-Carrying Data

- Diverse network, containing untrustworthy parties and unreliable components.
- Enforce correctness of the messages and ultimate results.
• Every message is augmented with a **proof** attesting to its **compliance** with a prescribed policy.
• Compliance can express any property that can be verified by locally checking every node.
• Proofs can be verified efficiently and **retroactively**.
• If the final proof is OK, we can trust the result.
# The road to Proof-Carrying Data

The correct execution of arbitrary C programs can be verified in 5 milliseconds using 230-byte proofs.

## Feasibility

<table>
<thead>
<tr>
<th>Feasibility</th>
<th>Network</th>
<th>C program size</th>
<th>Program running time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Theory</td>
<td>Prototype</td>
<td>Fast</td>
<td>1 hop</td>
</tr>
</tbody>
</table>

- [Micali 94] [Groth 2010]
- [Chiesa Tromer 2010]
- [Ben-Sasson Chiesa Genkin Tromer Virza 2013]
- [Parno Gentry Howell Raykova 2013]
- [Ben-Sasson Chiesa Tromer Virza 2014]

Used in Zerocash: anonymous Bitcoin

[Ben-Sasson Chiesa Garman Green Miers Tromer Virza 2013]
The road to Proof-Carrying Data on the road

• More efficient PCD: cost, latency
• Formally defining the critical security properties within a vehicle, and then applying PCD to enforce them
• Extending to V2V and V2I
  – Trusting other cars
    (that trust other cars
    (that trust other cars
    (that trust infrastructure (and other cars))))
  – Protecting privacy using zero-knowledge proofs

SCIPR Lab scipr-lab.org