

An Attestation Capable Programmable Software Switch on FABRIC

Alexander Wolosewicz, Nishanth Shyamkumar, Nik Sultana

Introduction

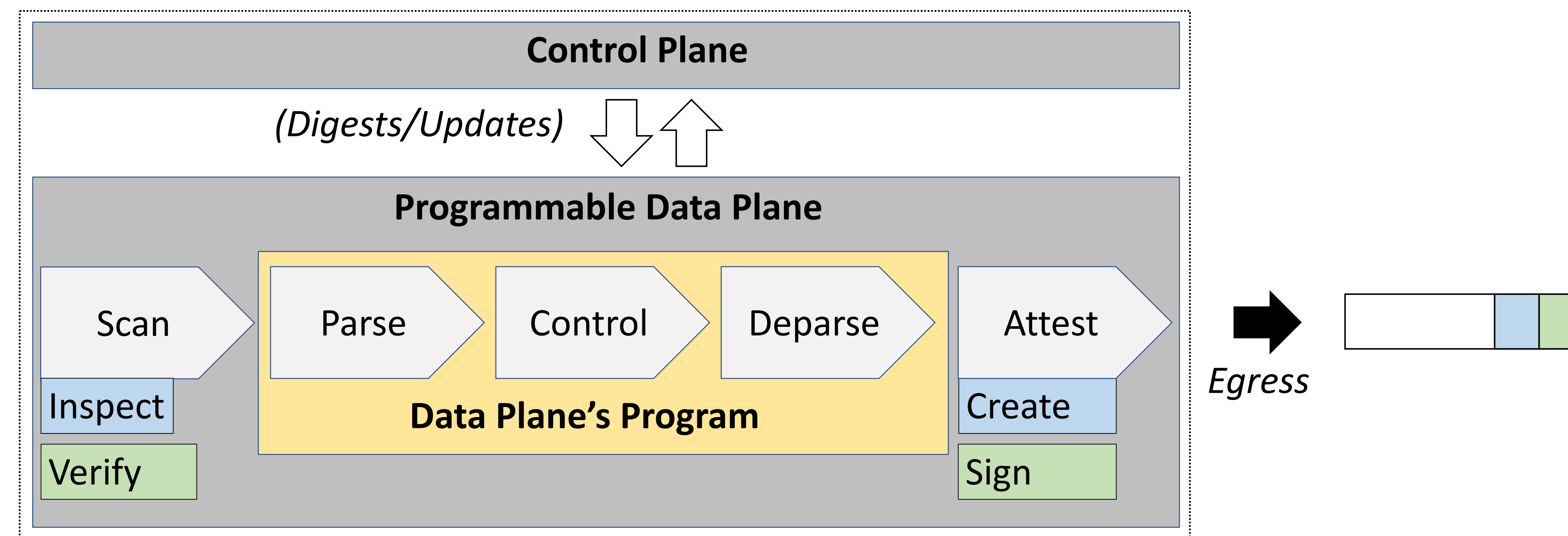
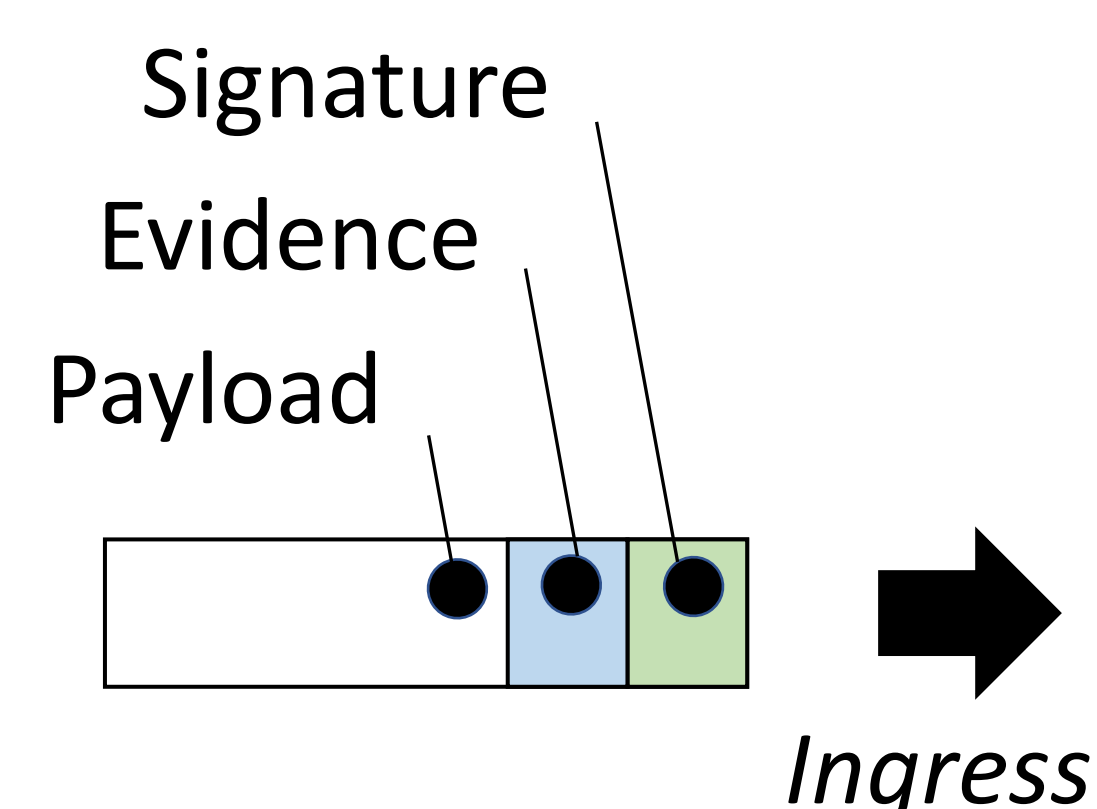
- Programmable networking elements provides great flexibility on the dataplane.
- But it also creates new risks of misconfiguration and of attacks that dynamically modify security-critical functionality.
- Using **Remote Attestation** techniques we can enable dynamic assessment of network security and configuration characteristics.
- We can create RA policies for programmable networks that specify the generation, collection and evaluation of **evidence of network program and control plane rules integrity**.
- By utilizing such policies network elements in a programmable network can participate in proving their own **trustworthiness**.

Motivation

- 1) Configuration transparency of programmable networking elements in a federated testbed.
- 2) Using configuration transparency for improved diagnostic ability, and reproducibility of research.

Acknowledgement

Our collaborators Ben Ujchich (Georgetown University) and Deborah Shands (SRI Intl), Vinod Yegneswaran (SRI Intl), and Ashish Gehani (SRI Intl).



Approach

- Define security primitives (state elements) that generate evidence of programmable device's dynamic working state.
- Evidence consists of **md5 hash digests** for switch and path state.
- Evidence is transported using **IPv6 Hop by Hop Extension Headers** and ultimately checked by the verifier.
- We extend a programmable network element (**BMv2 switch**) to accommodate our Remote Attestation implementation.
- Conduct verification and performance tests to confirm the working of the programmable element as an attester.

Results

- We display the evidence of the switch STATE and PATH evidence using the command line to query the switch.
- We compare it with the HBH header as seen at the receiver and verify that the state values have been transmitted successfully and correctly.

(Evidence seen from the control plane)

```
RuntimeCmd: get_ra_data
Registers: D41D8CD98F00B204E9800998ECF8427E
Tables: 083908AB3929001D4F94CDC290DC6C53
Program: 99914B932BD37A50B983C5E7C90AE93B
```

(Evidence seen from the data plane)

```
Source Address: fec0:db8:0:f000::10
Destination Address: fec0:db8:0:f001::100
  ~ IPv6 Hop-by-Hop Option
    Next Header: UDP (17)
    Length: 12
    [Length: 104 bytes]
  ~ Unknown IPv6 Option (55)
    > Type: Unknown (0x37)
    Length: 100
    > Unknown Option Payload: 00000000d41d8cd98f00
    0030 00 00 00 00 01 00 11 0c 37 64 00 00 00 00 d4 1d
    0040 8c d9 8f 00 b2 04 e9 80 09 98 ec f8 42 7e 08 39
    0050 08 ab 39 29 00 1d 4f 94 cd c2 90 dc 6c 53 99 91
    0060 4b 93 2b d3 7a 50 b9 83 c5 e7 c9 0a e9 3b d4 1d
    0070 8c d9 8f 00 b2 04 e9 80 09 98 ec f8 42 7e 08 39
    0080 08 ab 39 29 00 1d 4f 94 cd c2 90 dc 6c 53 99 91
    0090 4b 93 2b d3 7a 50 b9 83 c5 e7 c9 0a e9 3b c5 bd
    00a0 14 51 05 1c 35 00 00 71 f6 66 00 04 e0 31 00 00
    00b0 00 0c ac 98 08 bb ad 47 3f ef 35 12 85 8b de dd
    00c0 5a 88 88 a8 bb 5c c2 41 5b 14 28 11 84 77 35 a7
    00d0 06 bb a9 6f 4a 9c 5c 34 1a 9a 97 c6 08 a8 11 3f
```