P4-MACsec: Dynamic Topology Monitoring and Data Layer Protection with MACsec in P4-Based SDN

2. KuVS Fachgespräch "Network Softwarization“, 02.04.2020

Frederik Hauser, Marco Häberle, Mark Schmidt, Michael Menth

http://kn.inf.uni-tuebingen.de

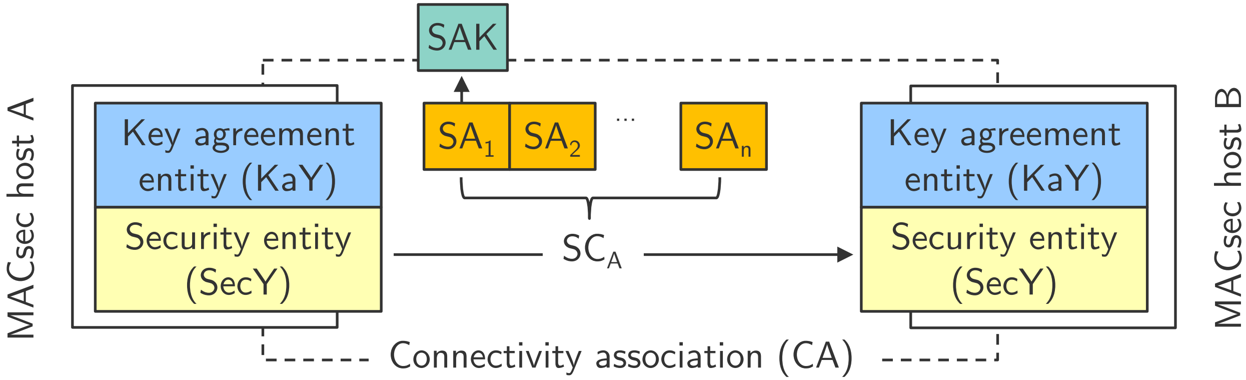
Outline

- Paper accepted for publication in IEEE ACCESS (2020-03-23)
  - Early access: https://ieeexplore.ieee.org/document/9044731

- Outline
  - Recap: MACsec (IEEE 802.1AE)
  - Problem statement
  - Concept of P4-MACsec
    - Secure link discovery
    - Automated setup/operation of MACsec
  - Experiences: prototypical implementations
    - BMv2
    - NetFPGA SUME
    - (EdgeCore Wedge with Tofino)
  - Recent/further work

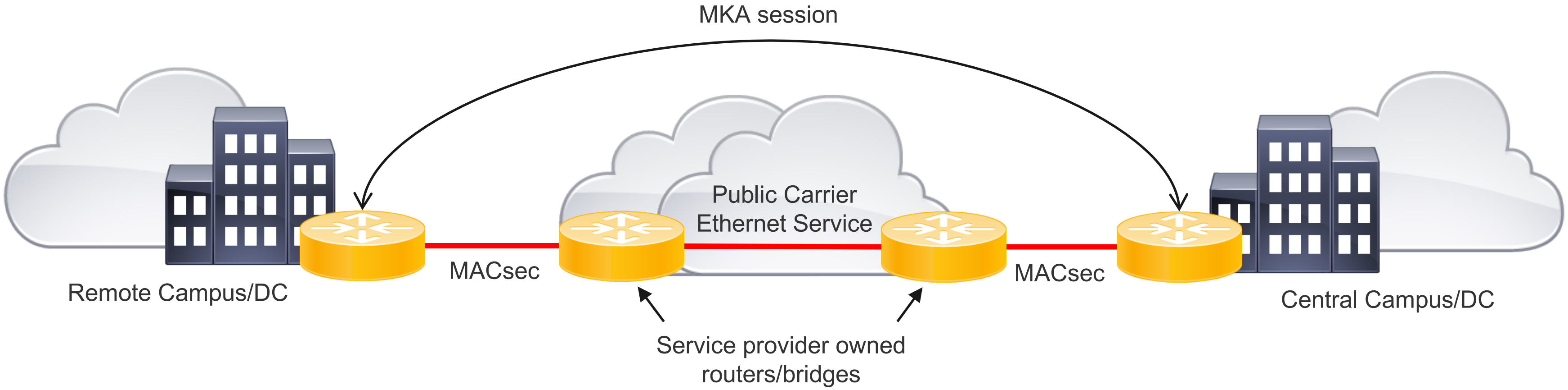
Recap: MACsec (1/2)

- MACsec (IEEE 802.1AE)
  - Point-to-point security between peers connected to a LAN
  - Integrity, confidentiality, and replay protection for Ethernet frames
- Principle
  - Secure channels (SCs) between MACsec hosts
  - Security associations (SAs) with secure association keys (SAKs)
  - KaY: discover other KaY + exchange keying material (MKA)
  - SecY: application of protect() and validate() functions to packets



Recap: MACsec (2/2)

- Application of MACsec
  - Enterprise / Campus networks
    - Protection against man-in-the-middle attacks
  - ”WAN MACsec” (Cisco)
    - Motivation: system capacity of IPsec limited (~ 40 Gbps)
    - Public Ethernet service as alternative to VPN



Source: https://www.cisco.com/c/dam/en/us/td/docs/solutions/Enterprise/Security/MACsec/WP-High-Speed-WAN-Encrypt-MACsec.pdf

Problem Statement

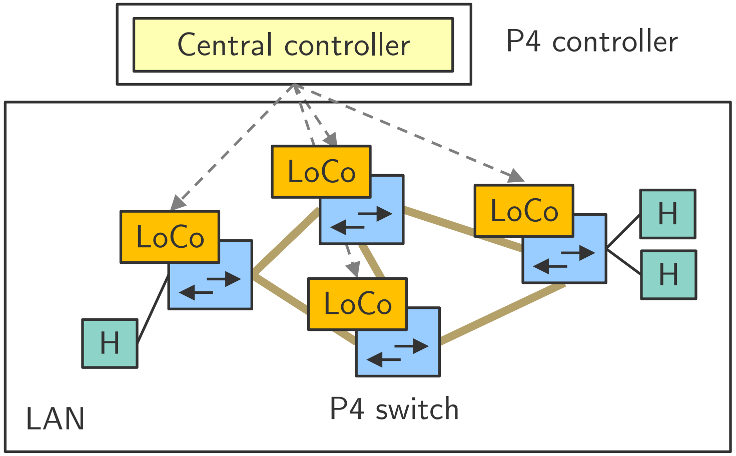
- Current MACsec deployment
  - Requirements: known topology with fixed links
  - MACsec setup and operation
    - Configuration of MACsec policies per interface
    - Static keys or MACsec key agreement (MKA) or EAP

- Related work: MACsec in SDN
  - Controller-based configuration of Linux nodes (Choi et al. 2018)
  - Many theoretical discussion of SDN-based deployment (Szyrkowiec et al. 2018, Vajaranta et al. 2016, Bentstuen and Flathagen 2018)
  - OpenCORD
    - SecY: part of the switch, KaY: control plane application
    - Configuration/operation via NETCONF
    - Only simulation, no implementation of packet encryption

P4-MACsec: Overview

- Functional components
  1. Secure link discovery / monitoring
  2. Automated deployment of MACsec

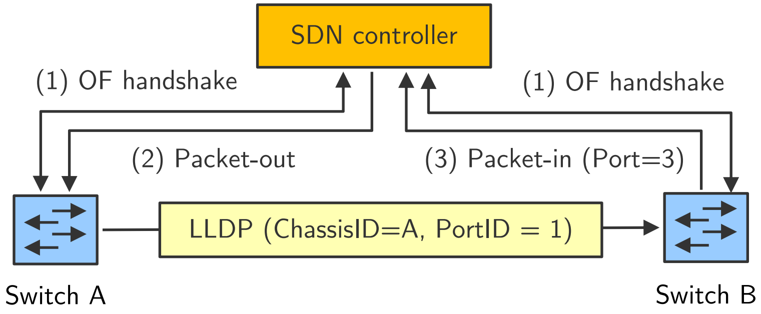
- Architecture
  - P4 switches
    - L2 packet forwarding
    - Packet-in / packet-out functions for secure link discovery
    - MACsec data plane functions
  - Two-tier control plane
    - Local controller (LoCo) assigned to each P4 switch
    - Central controller



P4-MACsec: Secure Link Discovery (1/3)

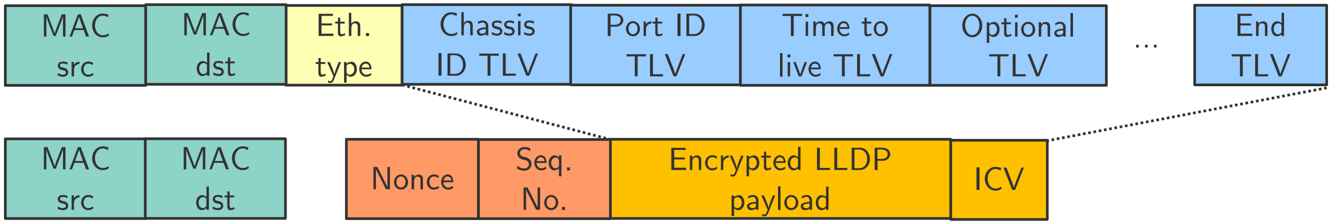
- Topology discovery in SDN
  - Switches: known by the controller (OF handshake, P4R setup)
  - Links: discovery mechanism / protocol

- Current approach: OpenFlow Discovery Protocol (OFDP)
  - Procedure (LDDP-based)
    - Create packets on controller
    - Output via packet-outs
    - Learn links via packet-ins
  - Problems
    - Efficiency: packet-outs + single controller
    - Security
      - Spoofing: LLDP injection for traffic redirection
      - Replay: incorrect topology view



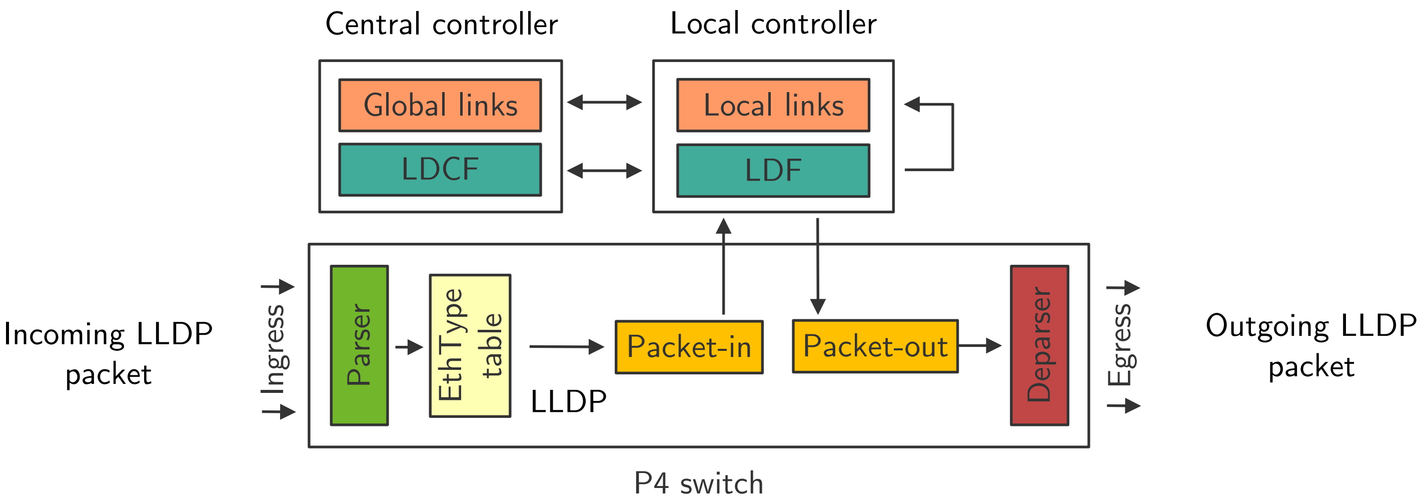
P4-MACsec: Secure Link Discovery (2/3)

- Secure link discovery in P4-MACsec (1/2)
  - Protect LLDP with AES-GCM
    - Add authentication and confidentiality to LLDP packets
    - Common encryption key among all switches
  - Nonce + sequence number
    - Protection against replay attacks



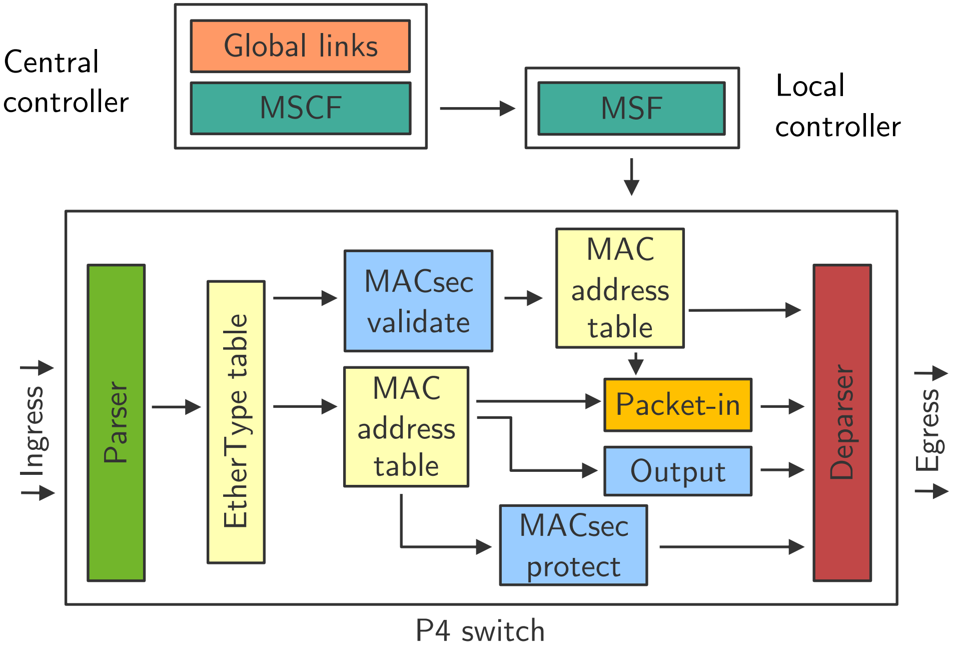
P4-MACsec: Secure Link Discovery (3/3)

- Secure link discovery in P4-MACsec (2/2)
  - Two-tier control plane function
    - Central controller: global link map + link discovery controller function (LDCF)
    - Local controllers: link discovery function (LDF)
      - Create and send out LLDP packets (via packet-out)
      - Receive and analyze LLDP packets (via packet-in)



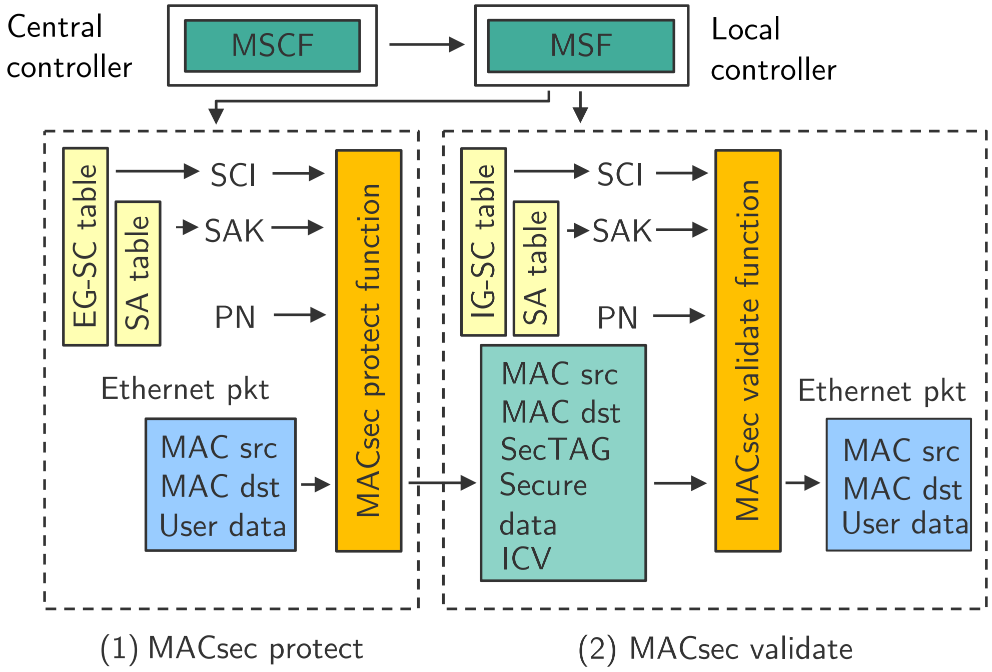
P4-MACsec: Automated Operation (1/2)

- Automated deployment of MACsec (1/2)
  - MACsec configuration via match-and-action table writes
  - Two-tier control plane function
    - Central controller: MACsec configuration function (MSCF)
    - Local controllers: MACsec function (MSF)



P4-MACsec: Automated Operation (2/2)

- Automated deployment of MACsec (2/2)
  - P4 data plane implementation
    - protect() and validate() functions implemented as P4 externs
    - Packet number counters using P4 counters



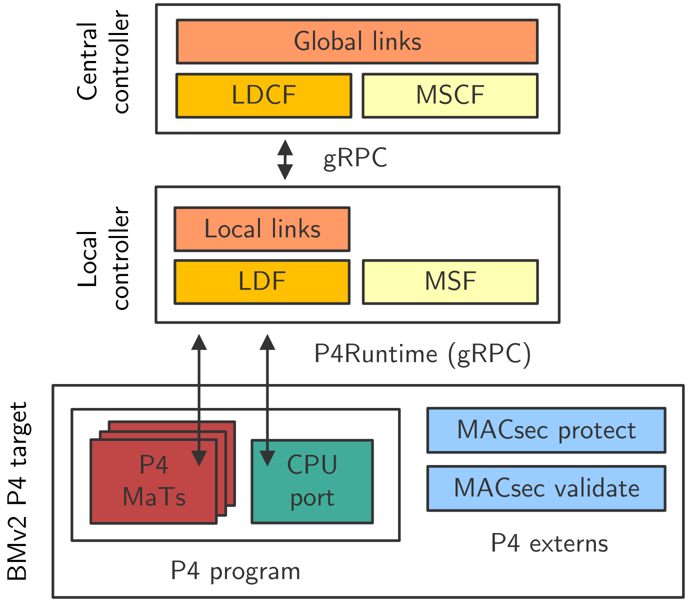
Prototypical Implementations (1/2)

- Software prototype: BMv2
  - simple_switch_grpc target
  - Two externs: MACsec protect() and validate() function
    - Implemented in C++ with help of the OpenSSL library

- Control plane
  - Implemented in Python 2.7
  - gRPC for interface in between

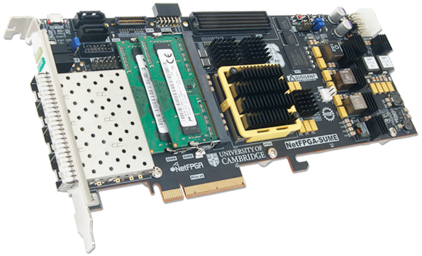
- OpenSource release on GitHub
  - Apache v2 license
  - Discussion about integration in BMv2 codebase

https://github.com/uni-tue-kn/p4-macsec



Prototypical Implementations (2/2)

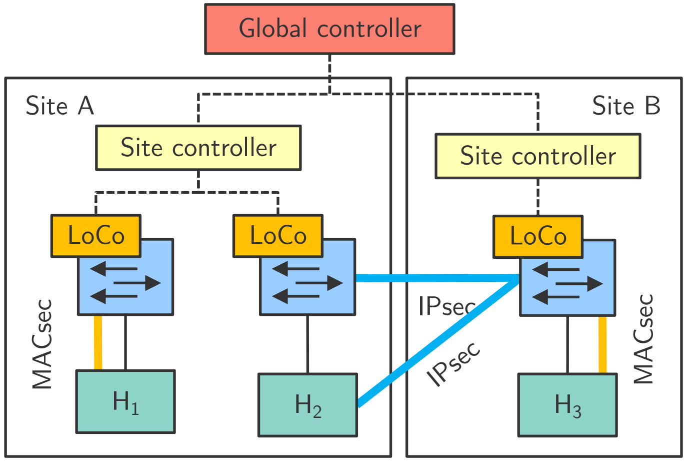
- Hardware prototype: NetFPGA SUME
  - Reusage of IP cores (AES-GCM from OpenCores)
  - No support for parsing variable-length payloads
  - No packet streaming function (data exchange limited to 128 byte per packet)



- Hardware prototype: EdgeCore Wedge with Barefoot Tofino
  - No support for P4 externs
  - Workaround
    - CPU port for interaction with main CPU module
    - Implement functionality in software running on CPU module

Recent/Further Work: P4sec

- Encompassing system: automated security in distributed Enterprise and Campus networks
  - MACsec (P4-MACsec)
    - Host-to-switch
    - Switch-to-switch
  - IPsec (P4-IPsec)
    - Site-to-site (SD-WAN)
    - Host-to-site (roadwarrior access)
  - 802.1X: PNAC



- Three-tier control plane
  - Local controller (per switch)
  - Site controller
  - Global controller
- Fully working prototype (based on BMv2)
  - Open source codebase + publication (in queue ☺)