TECHNISCHE UNIVERSITÄT DRESDEN

Tim Lackorzynski, **Sebastian Rehm**, Tao Li, Stefan Köpsell, Hermann Härtig
Chair of Privacy and Data Security

Secure and Efficient Extension of MACSec Tunnels for Modern Industrial Use Case

ICSS 2021 – Industrial Control Systems Security Workshop. 7/12/2021

Agenda

Introduction and Background

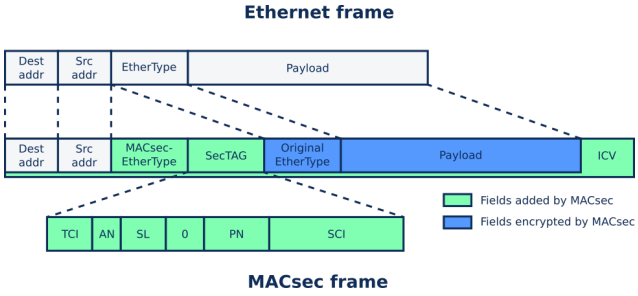
Design

Implementation

Evaluation and Conclusion

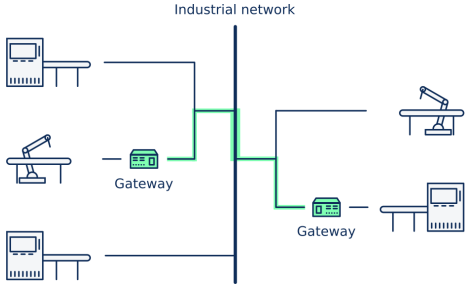
Introduction and Background – Agenda

01 MACSec

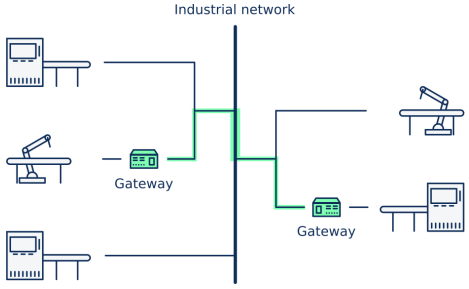


- Layer 2 security
- Payload confidentiality
- Integrity/authenticaton of whole frame
- MACsec Key Agreement protocol (MKA)

01 MACSec Middle Boxes

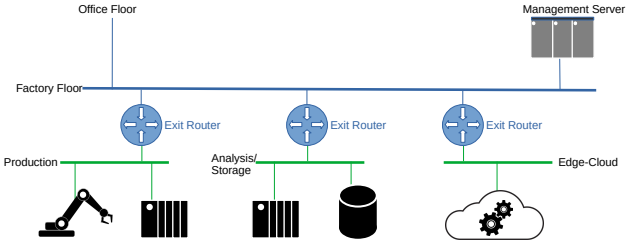


01 MACSec Middle Boxes

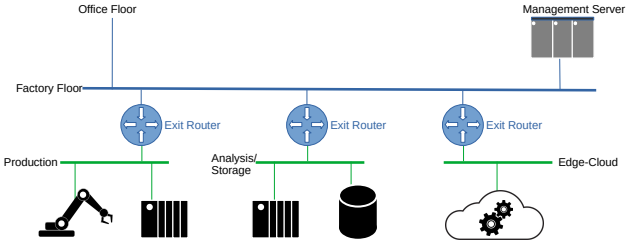


- Enhance security for *existing* industrial networks
- Transparent
- Agnostic of upper network layers

01 Industrial Network Bridging



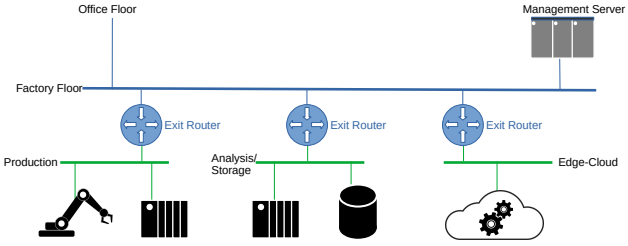
01 Industrial Network Bridging



Industry 4.0:

- Zero trust networks → Secure tunneling
- Data intensity → High performance required

01 Industrial Network Bridging



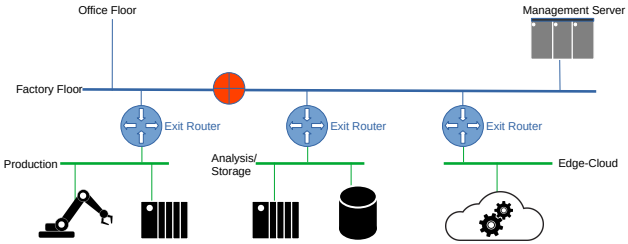
Industry 4.0:

- Zero trust networks → Secure tunneling
- Data intensity → High performance required

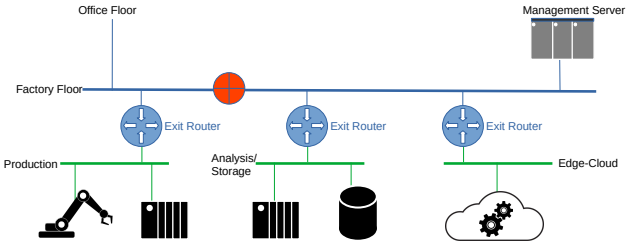
Two Channels:

- Data Channel (high throughput)
- Management Channel (slow)

01 Industrial Network Bridging



01 Industrial Network Bridging



Attacker capabilities:

- Passively intercept messages
- Delay and drop messages (including disordering)
- Replay messages
- Change message content
- Forge messages
- Relay messages

Design – Agenda

02 Tunneling

02 Tunneling

In principle the problem is solved: GRE, VXLAN, L2TP

02 Tunneling

In principle the problem is solved: GRE, VXLAN, L2TP

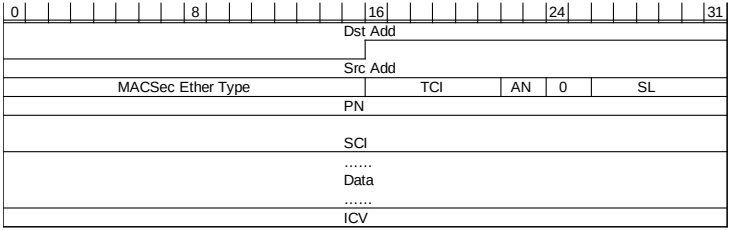
Guaranteed by MACSec:

- Integrity for complete L2 frame
- Confidentiality for the user data
- Authentication

Problems:

- No tunnel authentication → traffic injection possible (DOS against endpoints)
- Header confidentiality

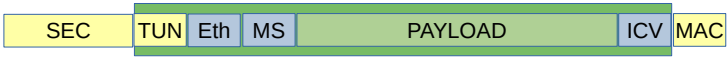
02 Problem Analysis



- Destination and source address relevant
- Ether Type not relevant
- TCI not relevant
- AN(SA) partly relevant
- SL not relevant
- PN relevant
- SCI relevant
- ICV not relevant

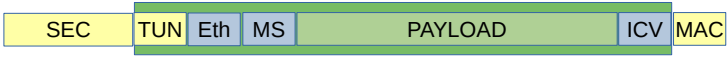
02 Secure Tunneling

Several existing solutions like Wireguard, IPSec, OpenVPN



02 Secure Tunneling

Several existing solutions like Wireguard, IPSec, OpenVPN



Huge performance loss

- Network stack layers:
  Ether-L2TP-IP/IP-IPSec
- Double encryption of the payload

02 Flow

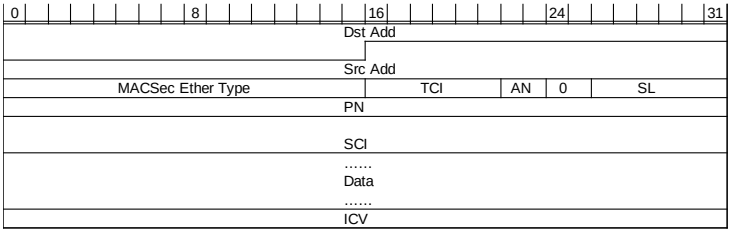
- End-to-end relationship (source + destination address + SCI)
- In MACSec: multiple associations for a given pair (Secure Associations SAs) → One Flow per SA

02 Flow

- End-to-end relationship (source + destination address + SCI)
- In MACSec: multiple associations for a given pair (Secure Associations SAs) → One Flow per SA

Every MACSec packet is attributed to a Flow by a tunnel gateway (Normally by a hash table - often done anyways for routing)

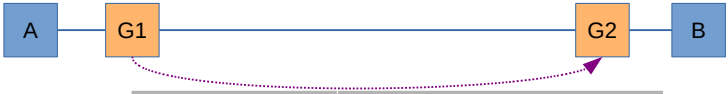
02 Predictability



- Destination and source address: constant
- Ether Type: constant
- SCI: constant
- SA: constant

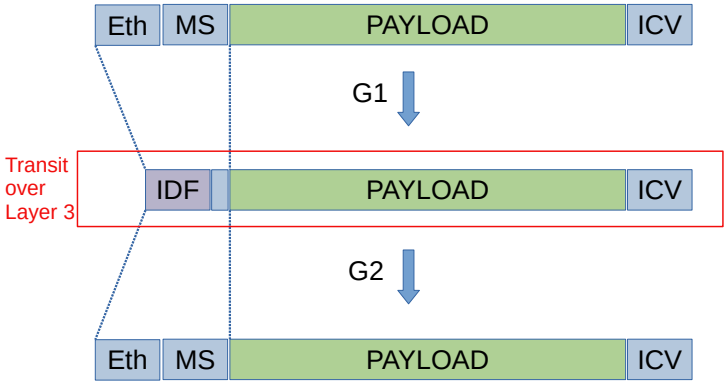
- PN: predictable
- SL: not predictable but in principle derivable
- TCI: partly predictable

02 Identifier Header Replacement



Identifier	Info
bd5f4aed67dcfa	Src: de:ad:be:ef:00:11 Dst: af:fe:00:99:00:11 SCI: 09876543212345 ... Nxt PN: 42
...	...

02 Identifier Header Replacement



02 Problems

Integrity ensured by MACSec
Confidentiality solved by identifier

02 Problems

Integrity ensured by MACSec
Confidentiality solved by identifier (to some extend: pseudonymous)

02 Problems

Integrity ensured by MACSec
Confidentiality solved by identifier (to some extend: pseudonymous)
No authentication

02 Problems

Integrity ensured by MACSec
Confidentiality solved by identifier (to some extend: pseudonymous)
No authentication

- Traffic injection

02 Problems

Integrity ensured by MACSec
Confidentiality solved by identifier (to some extend: pseudonymous)
No authentication

- Traffic injection

No counters/state-information

- Different states break the channel (used vs expected counter)
  Adding a counter additionally requires integrity checks

02 Rotating Identifier

– Authenticity is often realized through unpredictability (MACs/cookies)

02 Rotating Identifier

- Authenticity is often realized through unpredictability (MACs/cookies)
- Use a keyed derivation function for every identifier (use transfered identifer as key 'base identifier')
  `rotidf ← F ( baseidf , PN)`

02 Rotating Identifier

- Authenticity is often realized through unpredictability (MACs/cookies)
- Use a keyed derivation function for every identifier (use transfered identifer as key 'base identifier')
  `rotidf ← F ( baseidf , PN)`
- PN is implicitly represented in the rotating identifier → avoid async states in gateways

02 Rotating Identifier

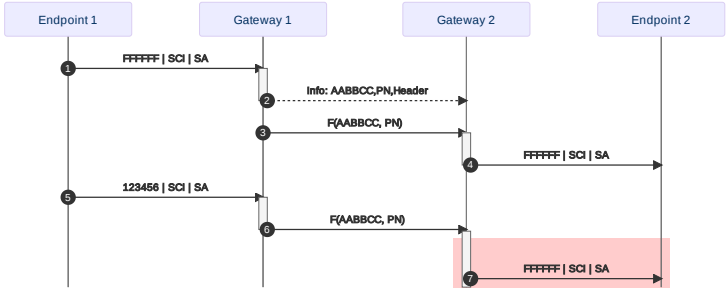
- Authenticity is often realized through unpredictability (MACs/cookies)
- Use a keyed derivation function for every identifier (use transfered identifer as key 'base identifier')
  `rotidf ← F ( baseidf , PN)`
- PN is implicitly represented in the rotating identifier → avoid async states in gateways
- Lost packets: window → precalculate identifiers for the complete window
  → accept packets inside of the window

02 Rotating Identifier

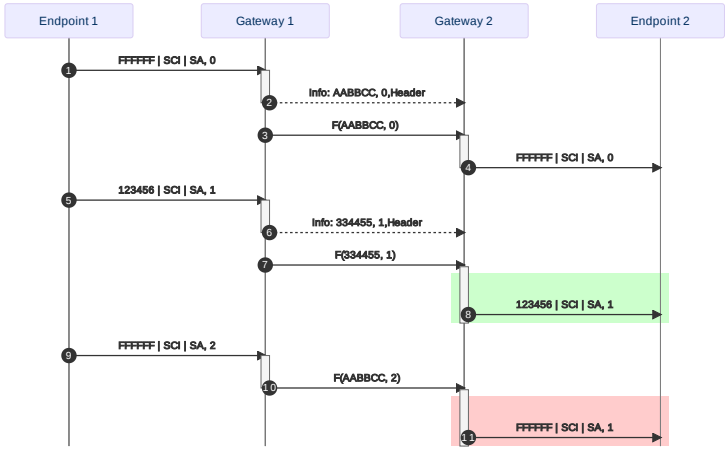
- Authenticity is often realized through unpredictability (MACs/cookies)
- Use a keyed derivation function for every identifier (use transfered identifer as key 'base identifier')
  `rotidf ← F ( baseidf , PN)`
- PN is implicitly represented in the rotating identifier → avoid async states in gateways
- Lost packets: window → precalculate identifiers for the complete window
  → accept packets inside of the window

PN	Identifier	Flow Info
123	19618bce812b57b2	Src,Dst,Flow 0,key
124	ea13661c0bee40cf	Src,Dst,Flow 0,key
125	4b262c434cb5f526	Src,Dst,Flow 0,key
55	18fe7b9ae3484f67	Src,Dst,Flow 1,key
56	b2d4ba2fe2a6b7ac	Src,Dst,Flow 1,key
57	ea13661c0bee40cf	Src,Dst,Flow 1,key

02 Flow Binding



02 Flow Binding



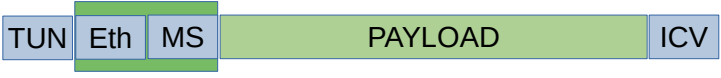
02 Header Encryption

The simple approach:

02 Header Encryption

The simple approach:

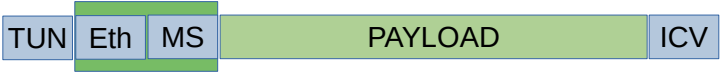
This is possible via the concept of Flows:



02 Header Encryption

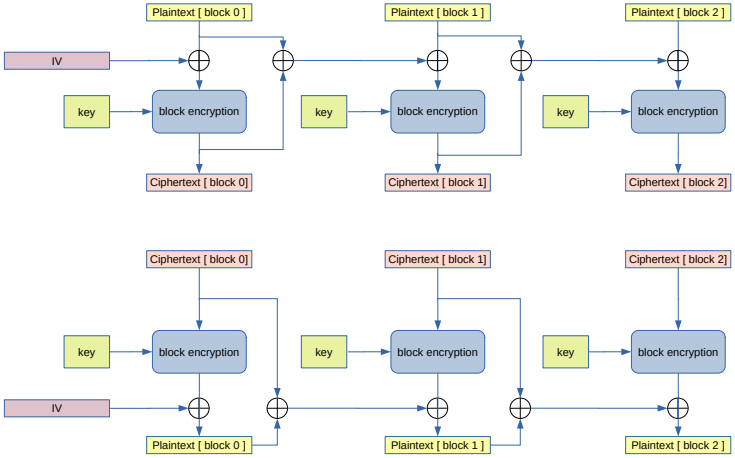
The simple approach:

This is possible via the concept of Flows:



- Use block encryption (not stream-like CTR)
  → Leverage diffusion of data
- Use underlying encryption and PN as IV
- Attacker does not have access to secret key → not able to generate ciphertext which decrypts to valid plaintext
- Check validity after decryption through constant data lookup → in case of change the lookup will fail → Analogous to symmetric authentication

02 Block Cipher Mode - PCBC



Implementation – Agenda

03 DPDK

Framework for fast packet processing

03 DPDK

Framework for fast packet processing

- Pure software solution (in contrast to e.g. SmartNICs)

03 DPDK

Framework for fast packet processing

- Pure software solution (in contrast to e.g. SmartNICs)
- Bypasses the OS network stack

03 DPDK

Framework for fast packet processing

- Pure software solution (in contrast to e.g. SmartNICs)
- Bypasses the OS network stack
- Some features:
  - Allows for busy polling of the NIC
  - Makes packets in the direct access memory accessible in user space
  - Reduces the handover time between layers/applications
  - Batch transfer of pointers (packets are processed individually in user space)

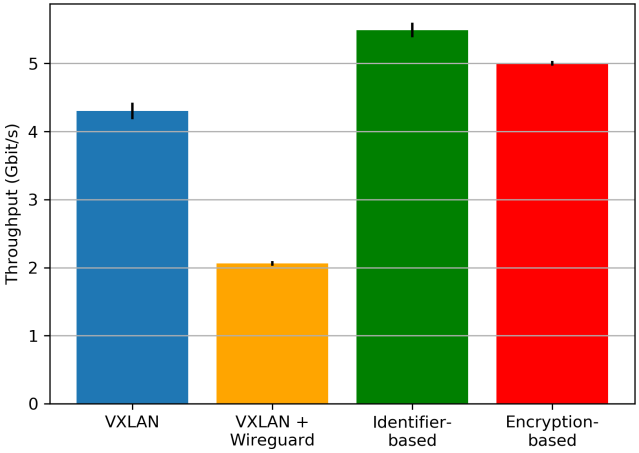
03 Derivation Function

Sip-Hash (Aumasson, J. P., & Bernstein, D. J. (2012, December))

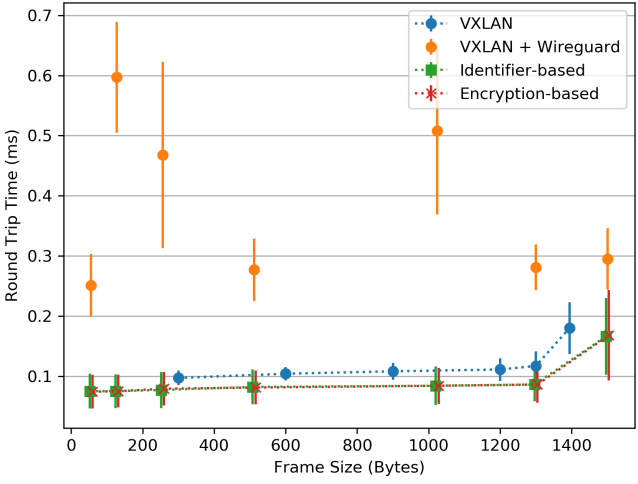
- Keyed hash function
- Designed for short inputs of 64 bits and a key of 128
- Designed for unpredictable hash-table entries and use as MACs

Evaluation and Conclusion – Agenda

04 Performance Measurements - Throughput



04 Performance Measurements - Latency



04 Conclusion and Future Work

Our Solutions:

- Two performant tunneling solutions for MACsec frames
- Confidentiality
- Integrity/Authentication
- Identifier based approach allows pre-calculation of expensive crypto

Possible ways to explore:

- Examine the function F
- Additional multiprocessing options
- Window sliding algorithm
- Flow binding
- Implement status synchronization (analogous to MKA)
- Implement MKA monitoring

Contact:
TU Dresden – Chair for Privacy and Security
Sebastian Rehms
sebastian.rehms@tu-dresden.de

04 Derivation Function

- Pre-image resistance is required. An attacker should not be able to recover the original base identifier as she could then predict all rotating identifiers for a Flow.
- Second pre-image resistance is not strongly required: Assume an attacker observes a $\mathrm{ridf_n}$ and that F is a good PRF. If she is able to find any $\mathrm{bidf'} \neq \mathrm{bidf}$ there is no problem if

$$\mathrm{F\ bidf', n+1} \neq \mathrm{F\ bidf, n+1}$$

  . Property not strongly required.
- Collision resistance is required. The probability for obtaining the same rotating identifier for two different PNs should be small, although if it happens seldom this does not break our protocol.

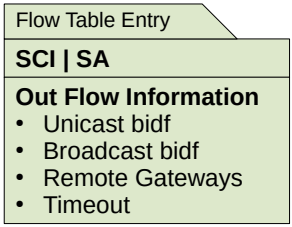


Figure 1: Uplink Flow Table entires

Figure 2: Uplink Flow Table entires

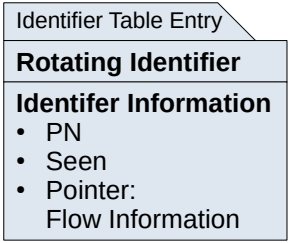
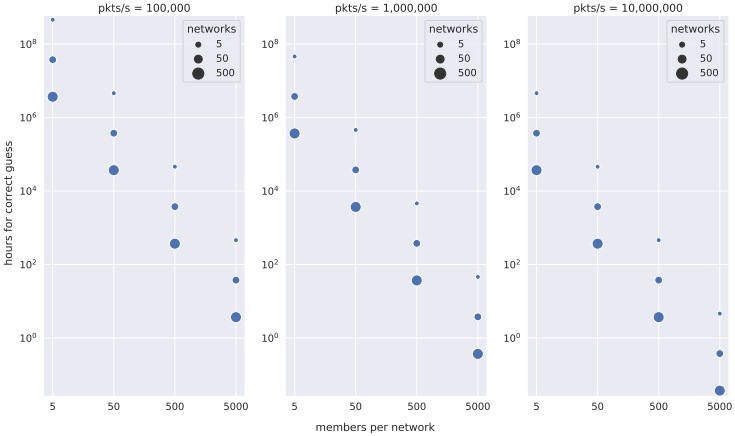


Figure 3: Uplink Flow Table entires

04 Brute Force Times

$s = 2w(n - 1)m^2$, $T = \frac{2^e+1}{2gs}$



04 Block Encryption Overview

Encrypt backwards!

- A: Packet number
- B: Constant data
- C: Encrypted data
- B: Constant Data

04 Multiprocess Implementation

Multiple possibilities:

- Ingress andegress is independent of each other → distribute this to multiple cores
- One core to burst packets and distribute then to worker cores → based on round robin / hitorical Flow load
- Offload expensive window sliding/hash-derivation to additional cores