



The Standards People

# A Proxy Transport Layer Secure Protocol

TLMSP, profile for fine grained access control

Presented by: **Roger Eriksson**

For: **3GPP**

26.2.2019

# Agenda

---

- ✔ Introduction & Background
- ✔ Transport Layer MSP (TLMSP) – profile for fine grained access control
- ✔ Comparison TLMSP and M32

# Introduction & Background

# Introduction & Goals

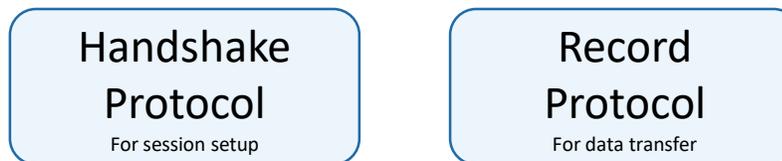
---

- ✔ Presentation based on an ongoing work in ETSI TC Cyber group
  - ✔ DTS/CYBER-0027
- ✔ End to End Encryption AND Proxy enabled functionality with fine grained access control to payload data
- ✔ End points in control of the whole data access chain
- ✔ End points in control of all allowed Proxies in the chain
- ✔ Audit tracking of changes
- ✔ Identity of Proxies cryptographically verified
- ✔ No requirement for Proxy re-encryption

# Based on

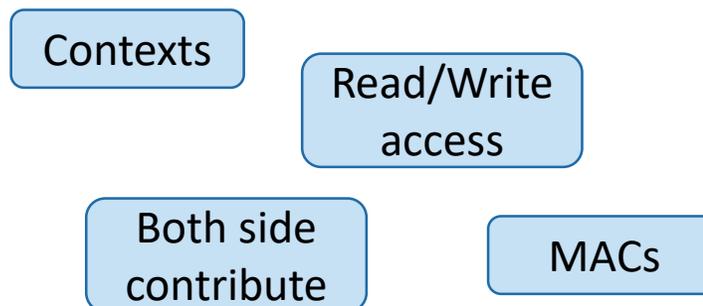
## ✓ TLS - Two part with

- ✓ Entity Authentication
- ✓ Payload Secrecy
- ✓ Payload Integrity



## ✓ mcTLS (<https://mctls.org/>) multipart with

- ✓ Entity Authentication
- ✓ Payload Secrecy
- ✓ Payload Integrity
- ✓ Visibility & Control
- ✓ Least privilege



# TLMSP

# TLMSP context



## Example Use Cases

- Corporate networks
  - Company firewalls, DLP-systems, Other types of information gateways
- Home network
  - Verifying that their data isn't being stolen by key-loggers, malicious apps, apps changing their privacy settings without user visibility
- Visiting a website
  - it would enable you to verify the security defenses of a web server before you trust it with sensitive information – gives assurance that they do indeed have proper security to protect you.
- IoT devices
  - Might not be upgradable to withstand latest threats
- Legacy devices (hospitals, CNI)
  - Devices that for economic reason can't be replaced
- Battery powered
  - Internal protection drains battery life

# TLMSP - highlights

## Record, Containers, Fragment

```
+-----+-----+-----+-----+-----+-----+
| TYPE | VERSION | TOT_LENGTH | C1 | C2 | ... | Cn |
+-----+-----+-----+-----+-----+-----+
<----- TLMSP header -----> <- container(s) ->
```

```
+-----+-----+-----+-----+-----+-----+-----+
| CTXT_ID | FLAGS | M_INFO (OPTIONAL) | LEN | FRAGMENT | WM | FM | nAF | Additional FMs |
+-----+-----+-----+-----+-----+-----+-----+
<----- container header -----> <- (OPTIONAL) ->
```

## Container insertion/deletion/audit

- Flags contain information about insertion, deletion, audit and FM for each container

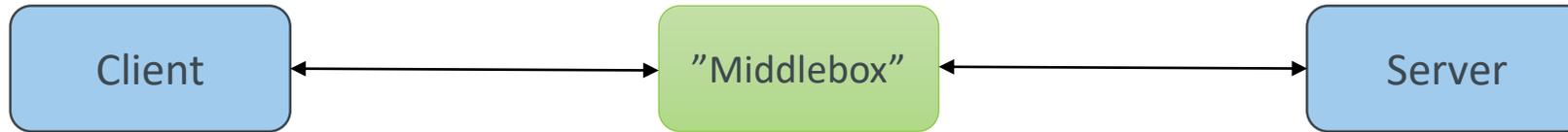
- Proxy key confirmation

- Dynamic discovery of Proxies

- Session resumption

- Support for fallback to TLS

# TLMSP – Handshake Overview



- Hello messages
  - get information on all parts in the chain
  - Establish context use
  - Certificate & Cryptographic information
- Key material and Key confirmation (cryptographically bind entities)
  - Client – Server
  - Client – Proxy (one for each proxy in the chain)
  - Server – Proxy (one for each proxy in the chain)

✔ Based on TLS 1.2 handshake

✔ Extensions to support:

✔ Proxy access rights

✔ Allow client and server to verify the correct use of security parameters, including the agreed permissions of Proxies

✔ Information to authorize Read or Read/Write access

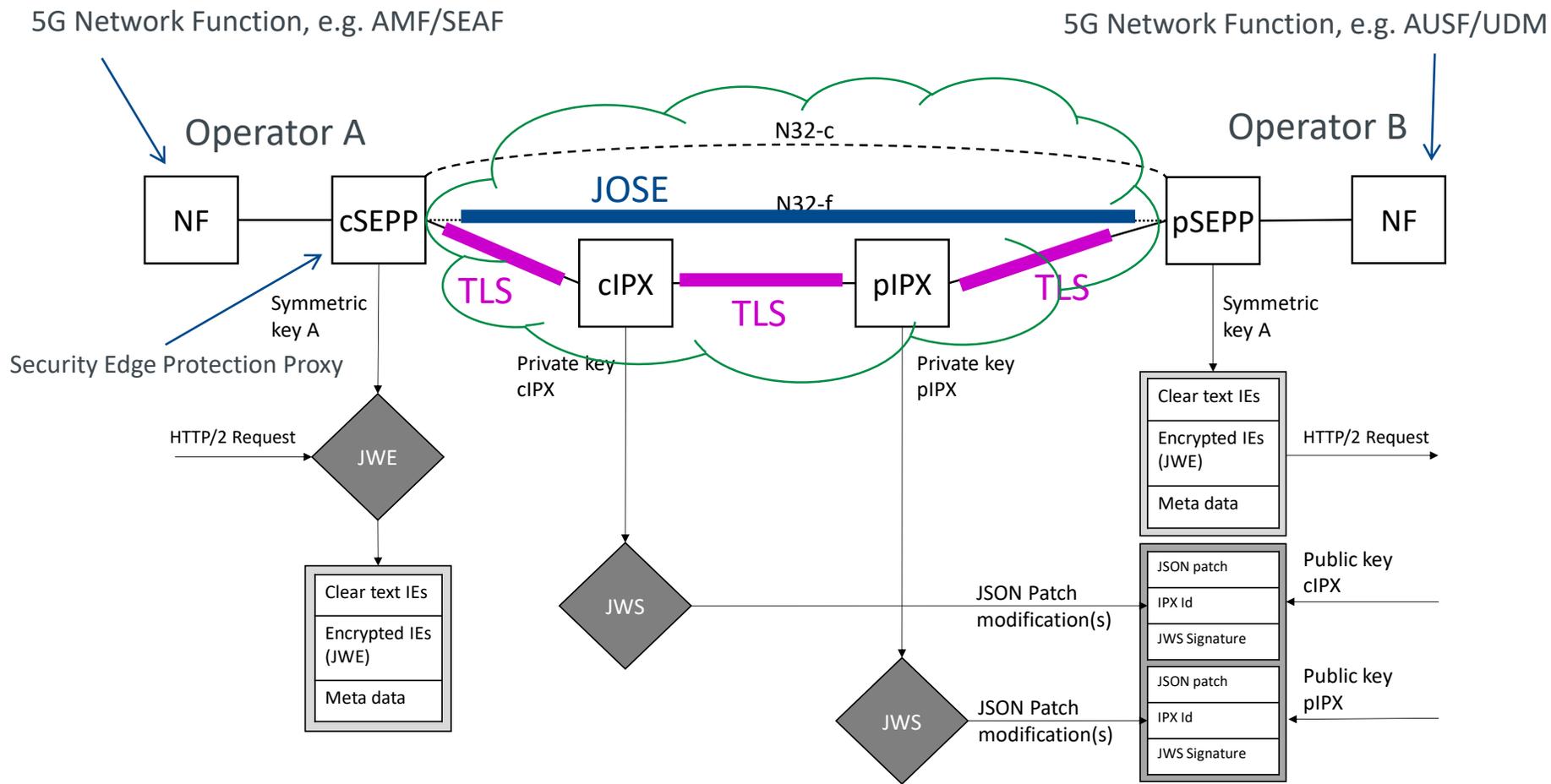
# Security Considerations

---

- ✔ Both parts (client/server) contribute with key material
- ✔ Proxy key confirmation
- ✔ Keys are bound to set of identities
- ✔ Audit trail
- ✔ Proxy may authenticate the client

# Comparison between TLMSP & N32

# N32-security model (1/2)



- E2E application layer security association between cSEPP and pSEPP
- Hop-by-hop TLS security associations between IPX-proxies

## N32-security model (2/2)

1. A policy is shared between SEPPs, the policy determining(\*):
  - a) which parts of data that MUST NOT be accessible to IPX-proxies
  - b) which parts of data that MAY be read by IPX-proxies
  - c) which parts of data that MAY be modified (and thus read) by IPX proxies
    - in case (c), also per-IPX info on which type of modifications that are allowed
2. Processing at source SEPP
  - i. Parts of data (a) are protected by blue e2e "inner" security using JWE
  - ii. All data (incl (b, c)) protected by purple hop-by-hop "outer" TLS security
    - in case (c), also by e2e JWE (AEAD) integrity protection from (i), no encryption
3. Processing at IPX-proxies
  - Read access: IPX-proxy terminates incoming TLS encryption and simply reads cleartext data from the JWE object, then forwards in outgoing TLS
  - Write access: as above, and then adds (signed) JSON object which contains the modification , finally forwards in outgoing TLS
4. Processing at destination SEPP
  - i. Terminate incoming TLS
  - ii. Verify e2e integrity of JWE object and decrypt the encrypted parts
  - iii. Check signature(s) of JSON-protected modification object(s)
  - iv. Verify that performed modification(s) in (iii) are allowed by policies(\*)

(\*) It is assumed that operator A agrees its modification policy with pIPX and operator B agrees its policy with cIPX. Additionally, the SEPPs exchange certificates of both IPX:es to allow both operators (SEPPs) to verify the modifications made by each IPX-proxy.

# Mapping of N32-Sec to TLMSP

---

The two SEPPs map to client/server

✔ depending on REQUEST-RESPONSE direction

The IPX-proxies map to TLMSP “proxies”

Parts of data that:

✔ MUST NOT be accessible to IPX proxies map to “context 1”

✔ MAY be read by IPX proxies map to “context 2”

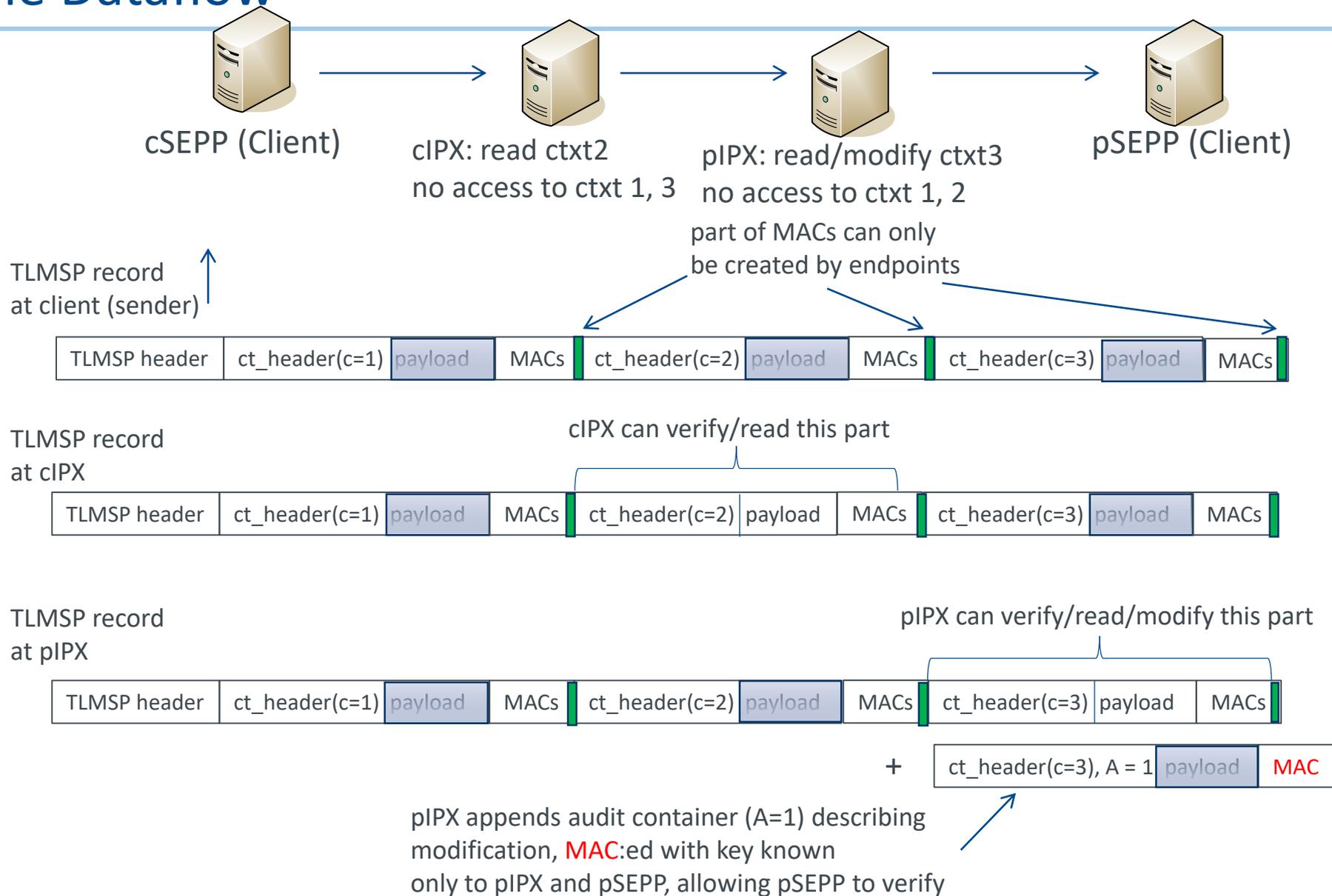
✔ MAY be modified IPX proxies map to “context 3”

The “allowed modification policies” map to themselves

Each TLMSP record contains up to 3 containers  
(i.e. one container for each context)

Assume cIPX is allowed to read context 2, pIPX allowed  
to modify (and read) context 3

# Example Dataflow



# Comparison

Feature	N32-Sec	TLMSP	Comment
Allows full e2e protection of selected data	Yes	Yes	
Explicit authn and authzn of <u>all</u> proxies toward both endpoints	No	Yes*	*Explicit authenticated handshake with <u>each</u> IPX , individual key material contributions
Cryptographic (key mgmt based) access control for each individual “part of data”	Yes/No*	Yes	*Yes for the e2e protected part and for “modify”, but not for “read”
Allows per-IPX read policy	No*	Yes	*Data that can be read can be read by <u>any</u> IPX
Allows per-IPX modify policy	Yes	Yes	
Protects IPX-readable/modifiable data from outside 3 <sup>rd</sup> parties	Yes	Yes	
Verification that all proxies have been granted access	No	Yes*	*Through key confirmation
Allows endpoint audit of IPX modifications	Yes	Yes*	*Using forwarding MAC and audit containers
Verification of modifications can use symmetric crypto	No	Yes	

---

Thank You