Encryption for 50G-EPON

Glen Kramer, Broadcom

Scope of the proposal

- Encryption is the last remaining big ticket item in IEEE 1904.4:

11	**Data encryption**	New 802.3ca behavior	1) Zero-overhead encryption as in SIEPON, pkg.A, but envelope-based instead of frame-based. 2) Add support for 256-bit keys. 3) Specify encryption using one key per ONU, not per LLID

- In 1904.1 SIEPON, the Pkg. A does not define encryption. It just references DPoE:

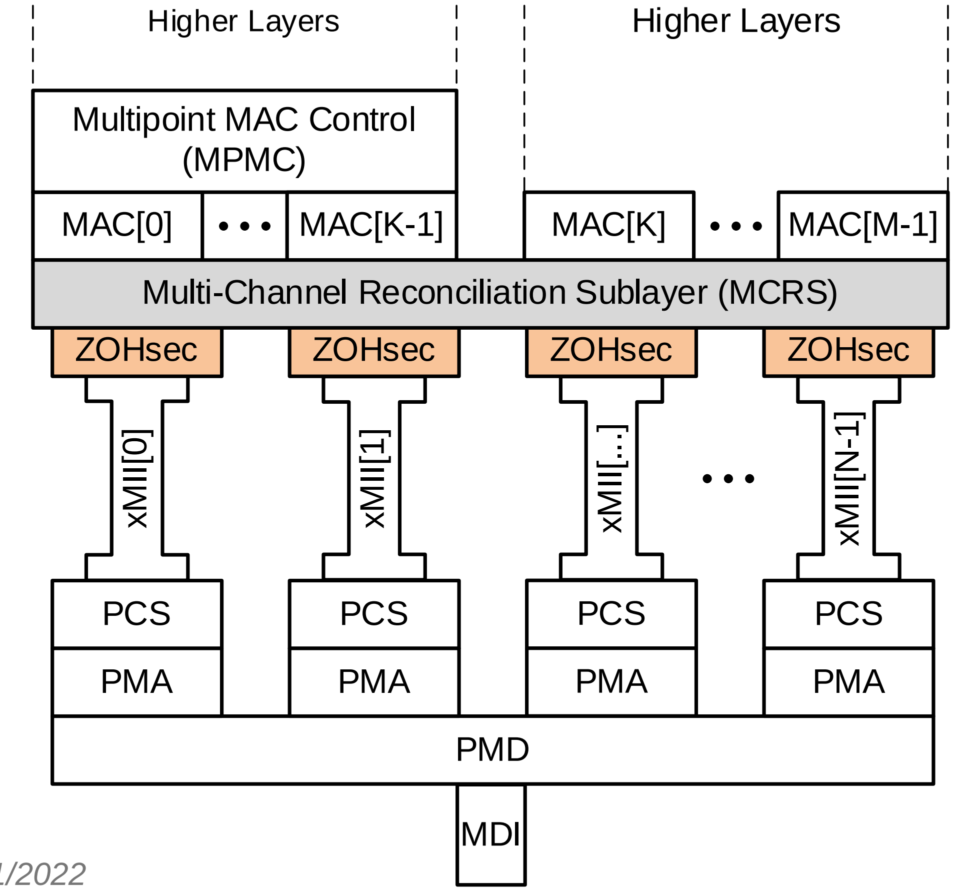
> **11.2.2 Data encryption in DPoE**
>
> Devices conforming to this profile shall implement data encryption and integrity protection mechanisms, as defined in DPoE-SP-SEC and DPoE-SP-OSSI.

- We have nothing we can reference for 25G and 50G-EPON. The complete encryption spec should be added to 1904.4 draft

Zero-Overhead encryption or MacSec?

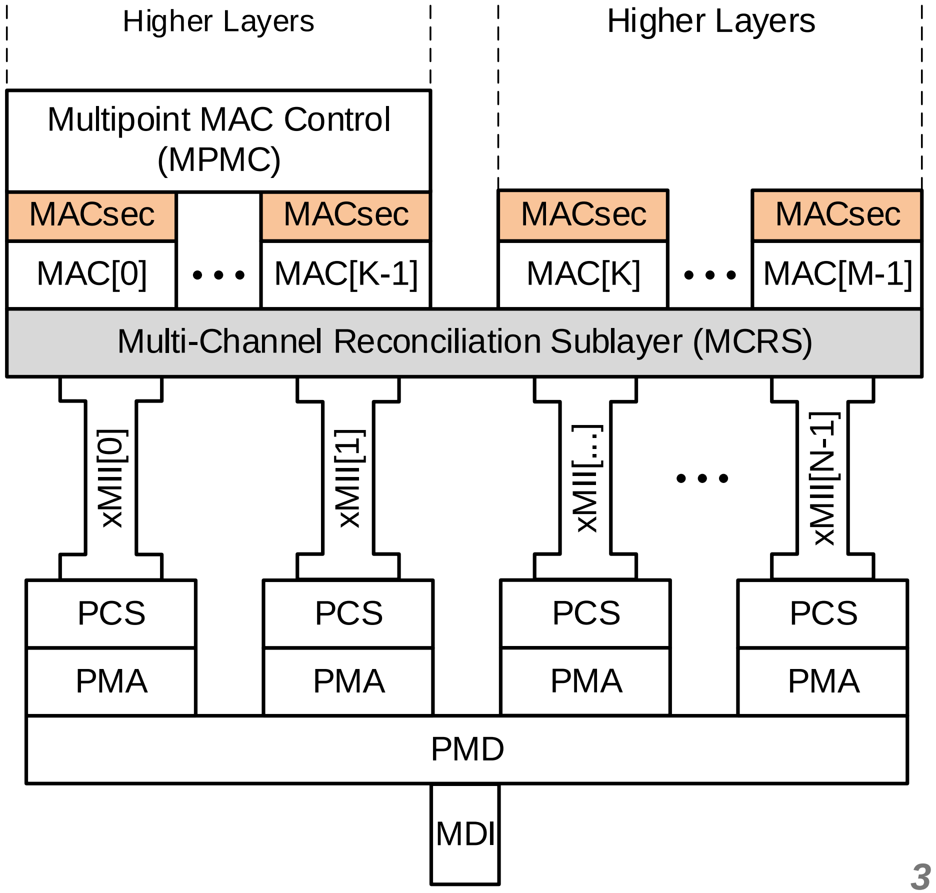
Zero-overhead encryption

- Same method as in DPoE, but extended to envelopes
- Zero overhead



IEEE 802.1X MacSec

- 32 bytes of overhead per frame
- No work for us



Envelope-based instead of frame-based

- Zero-overhead encryption method relied on 3 fields in frame preamble
  - 6b MPCP low-end bits
  - Encryption enable/disable
  - Encryption key index (0/1)
- The same fields have been added to the envelope headers by 802.3ca
- For the encryption function, there is no difference whether it encrypts an entire packet or a fragment

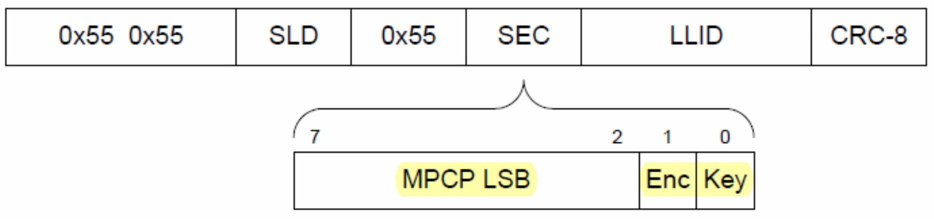


Figure 6 - Security Octet (10G Zero Overhead)

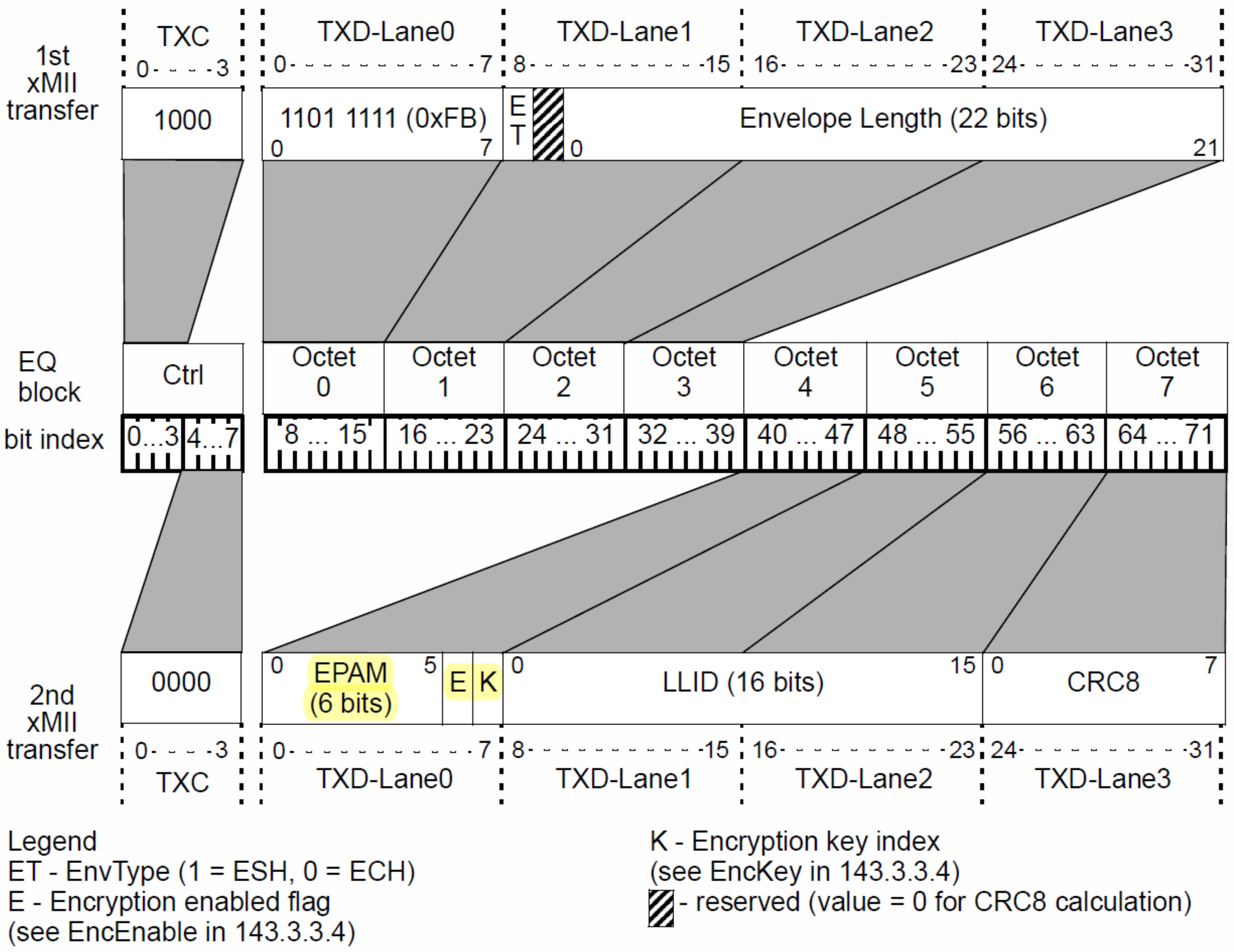


Figure 143–10—Mapping of envelope header fields into two xMII transfers

Support for 256-bit keys

- The communications industry is transitioning to 256-bit AES
  - ITU-T G.9804.2 (CommTC) supports AES-128 (mandatory), AES-256 (mandatory), Camellia-128, Camellia-256, and SM4(-128) .
  - DOCSIS includes CBC-mode 256-bit AES

- The 1904.4 should support both AES-128 and AES-256

One key per ONU, not per LLID

- One of key goals of 802.3ca was to make LLIDs more “lightweight”, so that LLIDs can be easily added/removed dynamically based on services needed at the moment.
  - LLIDs in 802.3ca are more similar to Service Flows in DOCSIS
  - ONUs generally are expected to support more LLIDs than in previous generations.
- Using an independent key for each LLID in the same ONU is undesirable
  - Waste of resources and key memory (2x256b/LLID x Number_of_LLIDs). This is an even bigger issue for the OLT.
  - Makes it hard for LLIDs to be allocated dynamically.
    - When a user starts a phone call, the NMS sends ConfigLLID action to the ONU and gets a new LLID up and running (using the existing key that ONU and OLT already know). But with a separate key per LLID, the ONU has to generate a new key, send it to the OLT, get a response from the OLT that a key is accepted, and only then start sending encrypted data.
  - The eavesdropping threat is between different ONUs (a malicious device can be connected to PON), not between different LLIDs on the same ONU. Traffic on different LLIDs may still be encrypted end-to-end by user applications. But the PON domain-specific threat is sufficiently resolved with the per-ONU encryption.
- Multicast LLIDs are different - the same key value must be used by multiple ONUs.

Proposed encryption architecture (1/4)

- Encryption of unicast (bidirectional) LLIDs
  - All unicast LLIDs in one ONU use the same encryption key.
  - Upon OLT's request, the ONU generates the key and sends it to the OLT (upstream is more secure).
  - The same key is used in both directions.
  - The entire upstream burst is encrypted with the same key, regardless of how many small fragments are in this burst.
    - A key fetch for every envelope (frame fragment) in not needed.
- Encryption of multicast (downstream-only) LLIDs
  - Each multicast LLID is encrypted using a separate key.
  - The OLT generates the initial key and shares it with each member ONU via the encrypted unicast MLID channel.
  - For large or long-lived multicast groups, the NMS may establish a mirror MLID multicast group and distribute subsequent key updates using encrypted multicast (single-copy) OAMPDUs.
  - ONUs that are not members of the given multicast group do not see the key and cannot decrypt the multicast traffic.

Number of keys to maintain
N = number of ONUs **M** = number of multicast groups at the OLT $\mathbf{m}_i$ = number of multicast groups at the ONU$_i$
Number of keys in OLT Encryption: **2×(N + M)** Decryption: **2×N**
Number of keys in ONU$_i$ Encryption: **2** Decryption: **2 + 2×m$_i$**

Two keys (current and next) are stored per each encrypted channel to guarantee seamless key switch

Proposed encryption architecture (2/4)

- Cryptographic Method
  - Use AES Counter Mode (same as the DPoE method for 10G-EPON)
  - Key size is 128 or 256 bits (selectable, global per OLT PON port)
  - In 50G-PON, the keys on both channels are the same, but the IV values are different

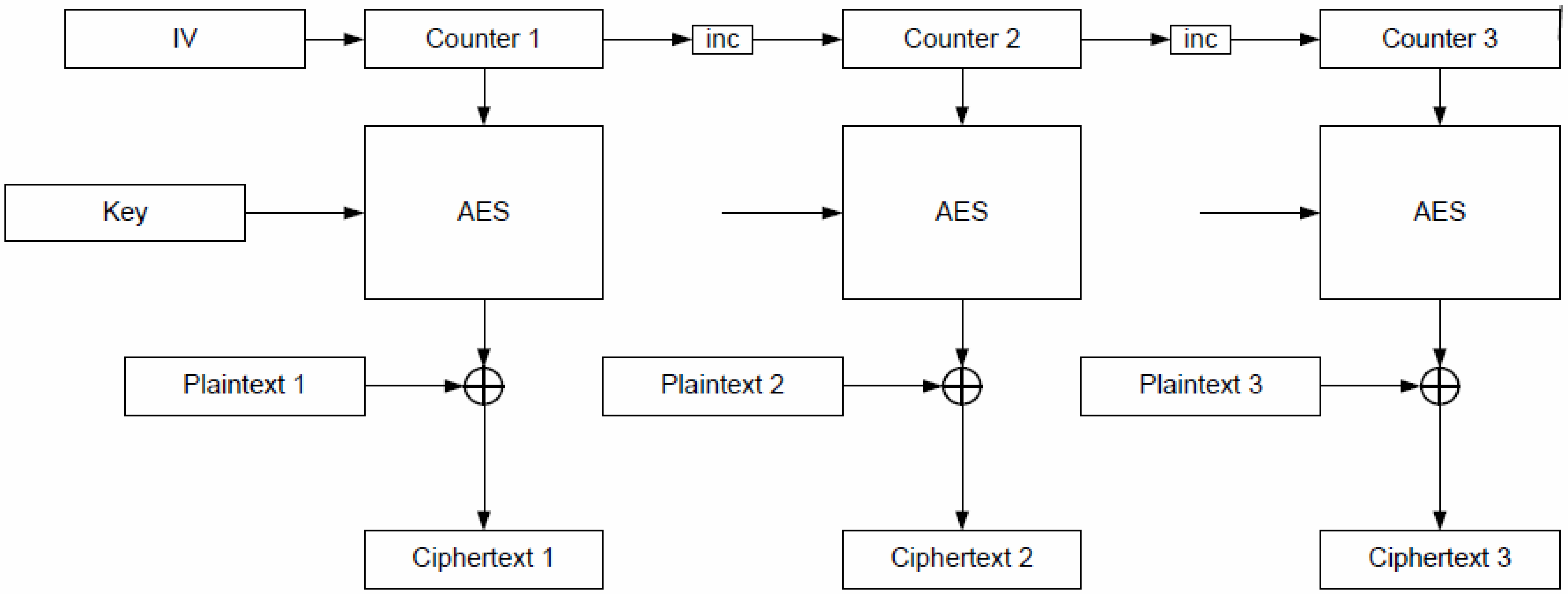
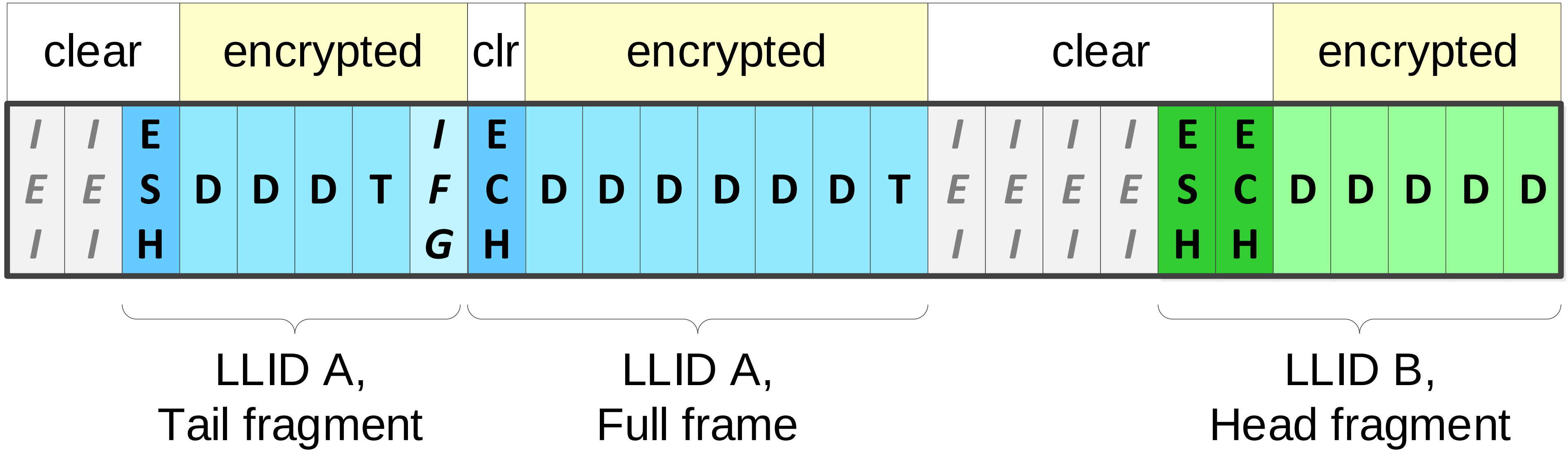


Figure 25 - Encrypting Data with CTR Mode

Proposed encryption architecture (3/4)

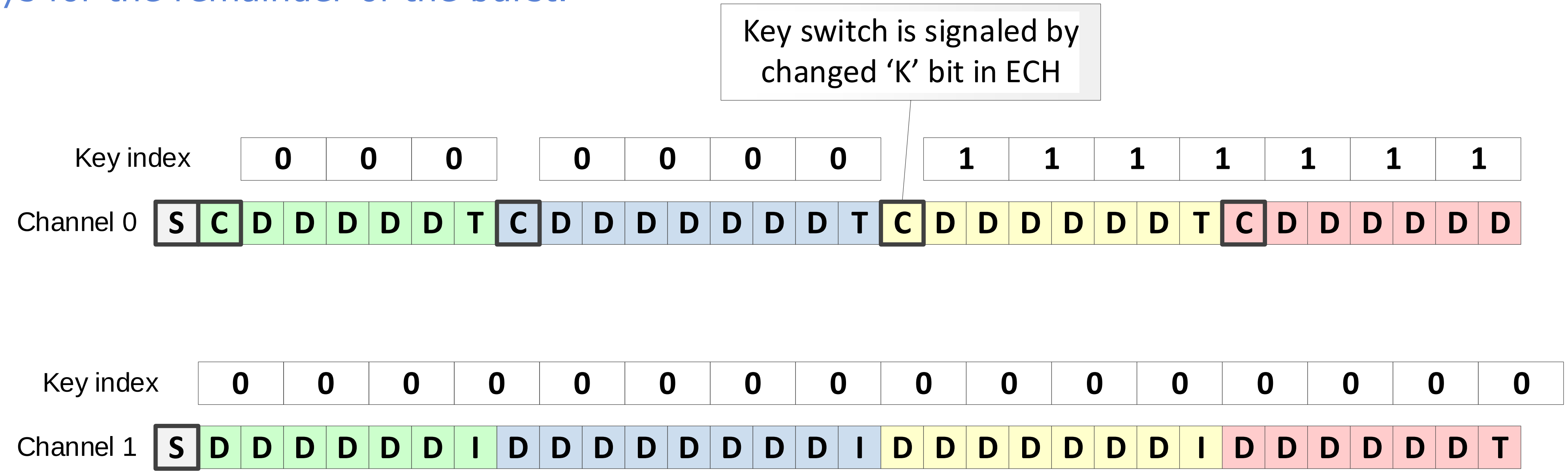
- Envelope headers are transmitted in clear text
- Inter-envelope-idles (IEIs) are transmitted in clear text
- Envelope payloads, including the IFG, are encrypted



Proposed encryption architecture (4/4)

Upstream key switchover options

- Option 1: keys may switch only between bursts.
  - The entire burst is always encrypted with the same key.
  - 50G burst uses the same key on both lanes
- Option 2: keys may switch during any envelope header transmission/reception
  - In 50G burst, the keys on two channels may be switched at different times, depending on envelope header locations on each lane
  - Very rarely, packets sizes in a burst may align in such a way that all envelope headers are located on one lane. If the key switch happened in the middle of such burst, the two lanes may be using different keys for the remainder of the burst.



Main focus areas for the new specification

- Location of the encryption function within the 802.3 layering architecture

- Construction of Initialization Vector
  - Both OLT and ONU shall be able to construct the same IV.
  - IV shall be time-dependent to prevent using the same value twice during lifetime of a key
  - IV shall be channel-dependent to prevent the use of the same value on both channels in 50G-EPON.

IV construction in DPoE

127–80	80–64	64–32	32–0
Transmitter SA	LLID	MPCP	Yincr

Figure 26 - Octet Order within the Initial Vector (10G)

- Transmitter SA
  - Downstream: MAC address associated with the OLT PON port
  - Upstream: MAC address associated with the ONU PON port
- LLID
  - Tx: LLID value associated with the MAC instance that sourced the packet to be encrypted
  - Rx: LLID value taken from the preamble of the encrypted packet
- MPCP
  - TX: 32 bit LocalTime counter at the time the packet was encrypted.
  - Rx: Receiver derives this exact time with the help of 6 low-order bits of the MPCP clock transmitted in the frame preamble.
  - MPCP rollover limits the lifetime of a key to about 68 seconds ($2^{32} \times$TQ)
- Yincr
  - Counter that begins at a value of 1 at the DA of each frame, and is incremented with each successive 128-bit block of data in the frame.

Proposed IV construction

- A new IV value is generated every time the encryptor transmits an envelope header or the decryptor receives an envelope header. Envelope headers are sent in the clear.

127 120	119 72	71 56 55 24	23 0
Ch.	48b MAC SA	48b Extended MPCP Clock	24b Block Counter

- **Ch** (Channel ID)
  - Encodes the channel on which this IV is used. 8 bits allow extensibility in case more channels are allowed in the future.
  - This field guarantees counter uniqueness when envelopes with the same LLID (i.e., same key) are transmitted on both channels (50G transmission) at the same time (i.e., at the same MPCP clock)
- **MAC SA**
  - MAC address associated with the PON port that transmitted the envelope. (Downstream: MAC address of the OLT PON port; Upstream: MAC address of the ONU PON port)
  - Both the OLT and ONU discover the peer MAC addresses at the time of registration.
- **Extended MPCP clock**
  - In 25G-EPON, the 32-bit MPCP counter rolls over every 11 seconds. This makes the key lifetime too short and will require frequent key updates. Extending the MPCP clock to 48 bits increases the maximum key lifetime to 200 hours.
  - The 48b MPCP clock is dejittered using the EPAM field in the envelope header in order to ensure that identical IV values are used by the ecryptor and decryptor
- **Block Counter**
  - The value is set to the *EnvLength* field received in the ESH or ECH.
  - The counter is decremented by one for every 128-bit block following the envelope header (see next slide)

Example of Block Counter Operation

IV Block Counter			**31**		**30**		**29**			**24**		**23**		**22**		**21**			**15**		**14**		**13**		**12**		**7**		**6**		**5**	
Residual Env Len	32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
Channel 0	S	C	D	D	D	D	D	T	C	D	D	D	D	D	D	D	T	C	D	D	D	D	D	D	T	C	D	D	D	D	D	D

IV Block Counter		**32**		**31**		**30**		**29**		**28**		**27**		**26**		**25**		**24**		**23**		**22**		**21**		**20**		**19**		**18**		**17**
Residual Env Len	32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
Channel 1	S	D	D	D	D	D	D	I	D	D	D	D	D	D	D	D	I	D	D	D	D	D	D	D	I	D	D	D	D	D	D	T

- When an envelope header is sent or received, the Block Counter is set to the value of *EnvLength.*
- Block counter is decremented for each 128-bit block of envelope payload (i.e., for every 2 EQs)

Uniqueness of counter values

Scenario	Channel ID	MAC SA	Ext. MPCP Clock	Block Counter
OLT or ONU transmitted two envelopes <u>on the same channel</u> (i.e., at different times)	Same	Same	Different	May be same
OLT or ONU transmitted two envelopes <u>at the same time</u> on two channels	Different	Same	Same	May be same
Blocks transmitted within the same envelope	Same	Same	Same	Different
OLT received envelopes from different ONUs on two channels at the same time	Different	Different	Same	May be same