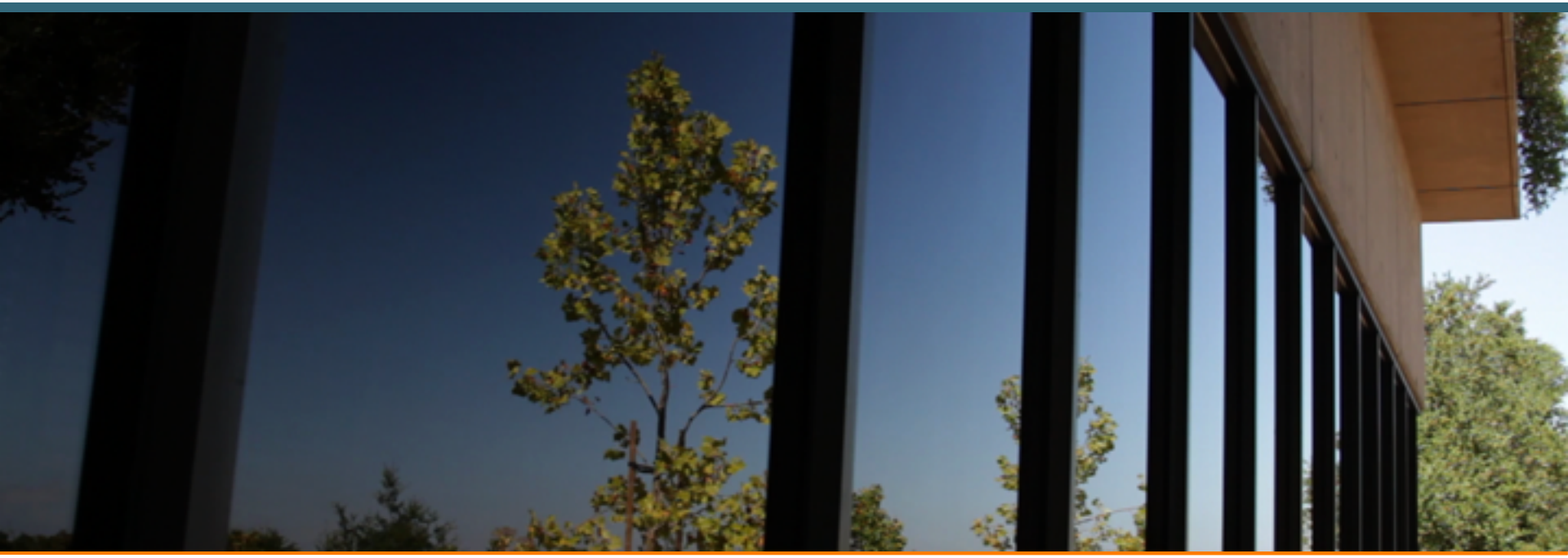
parc
A Xerox Company

CCNx 1.0 Wire Format Update
ICNRG @ IETF 90

Marc Mosko, Palo Alto Research Center

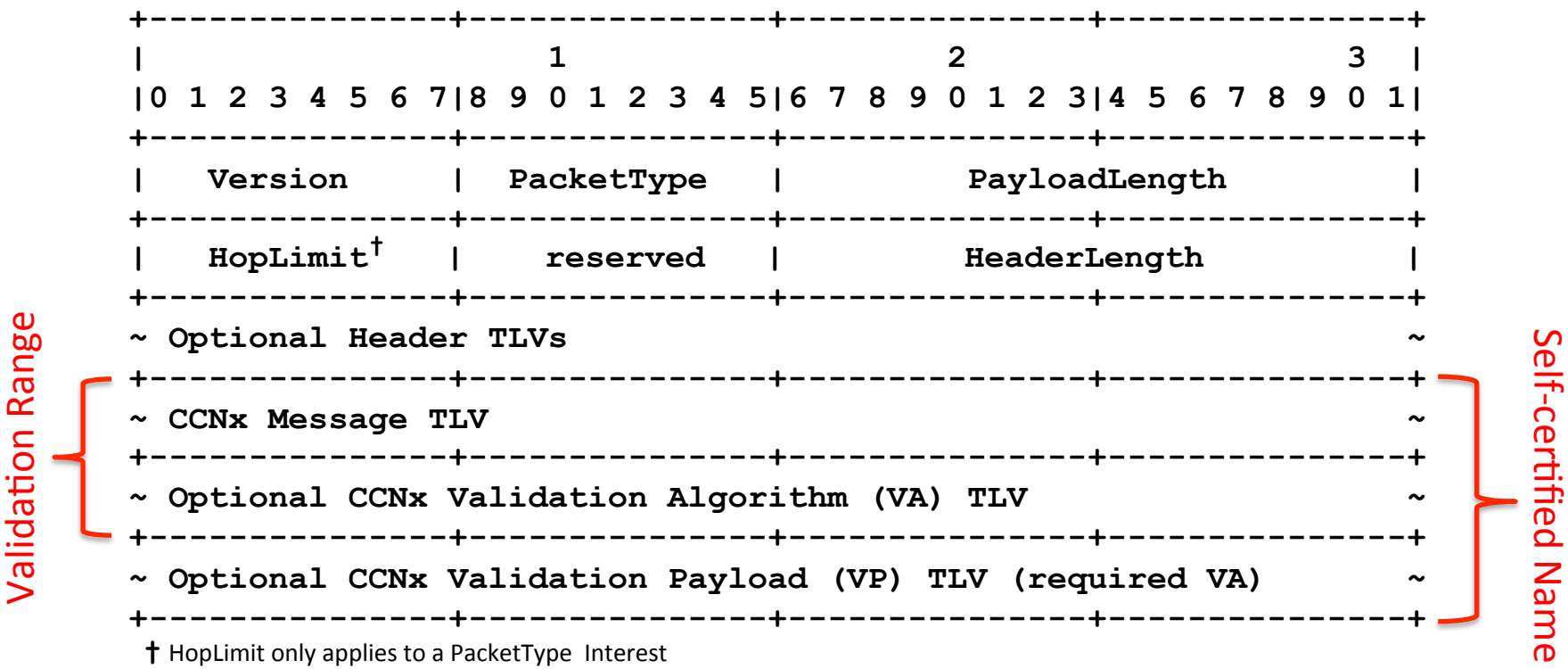


marc.mosko@parc.com |

Change Log from IETF 88

- Interests and Content Objects use same overall format.
- Make “validation” more modular and support CRC, Checksum, MACs, and Signatures.
- Split Validation out of the Content Object proper, can now be applied to any CCNx Message
- An Interest can carry Validation – we are not tied to IP protocols for end-to-end integrity checks.

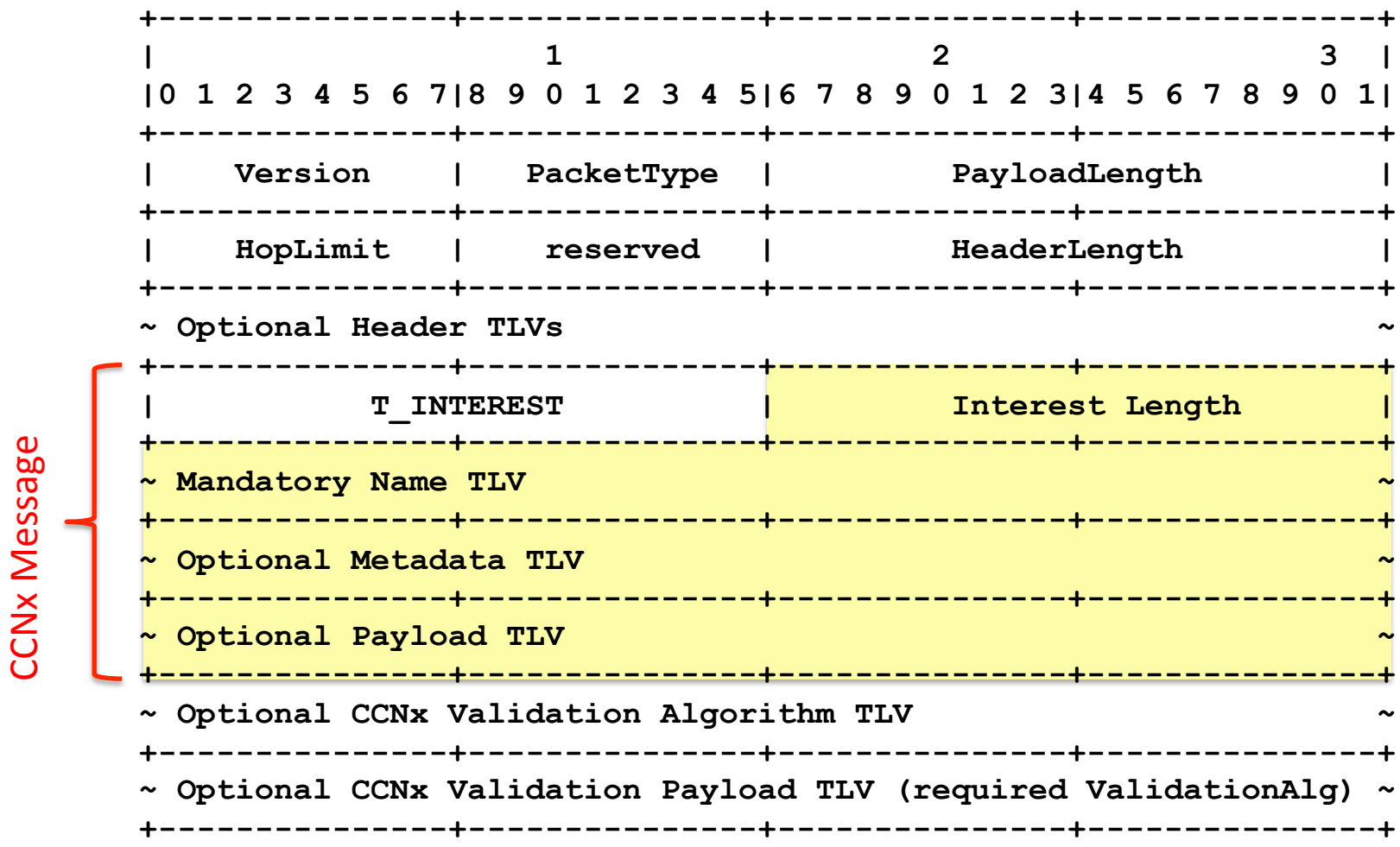
TLV PACKET



† HopLimit only applies to a PacketType Interest

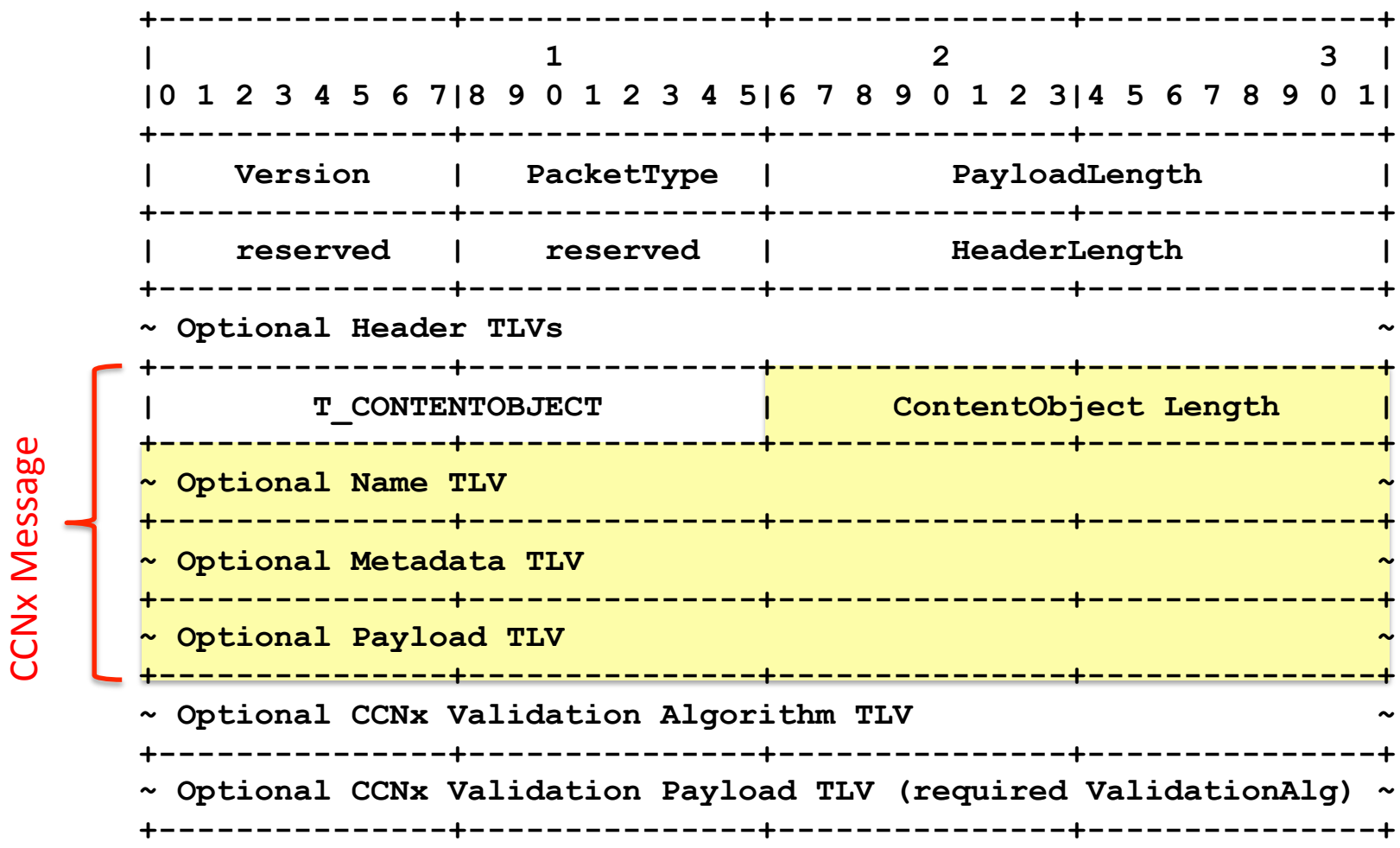
```
CCNx Message         = Interest / ContentObject / (other)
ValidationAlgorithm  = CRC32C / Checksums /HMAC-SHA256 / VMAC-128 /
                       RSA-SHA256 / EC-SECP-256K1 / EC-SECP-384R1
ValidationPayload    = (validation payload)
Validation Range     = (CCNx Message plus VA)
Self-certified Name  = SHA-256(CCNx Message, VA, VP)
```

INTEREST



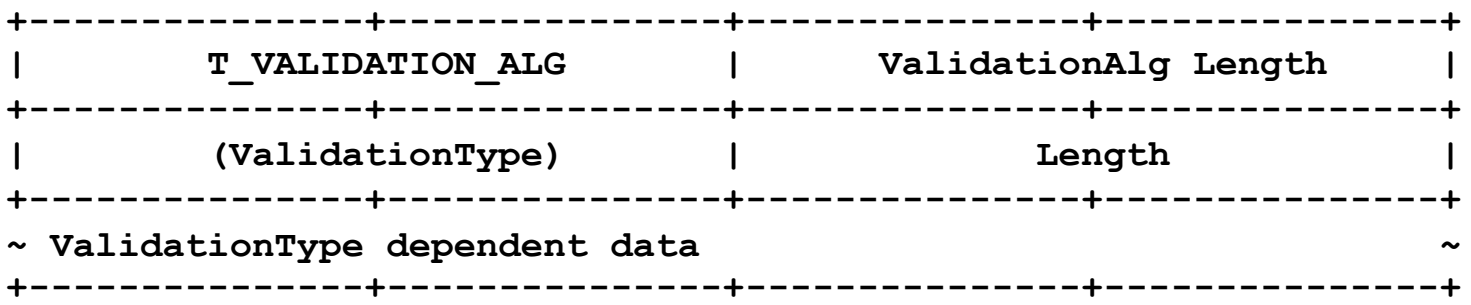

```
Headers    = Lifetime
Metadata   = KeyIdRestriction / ContentObjectHashRestriction
```

CONTENT OBJECT



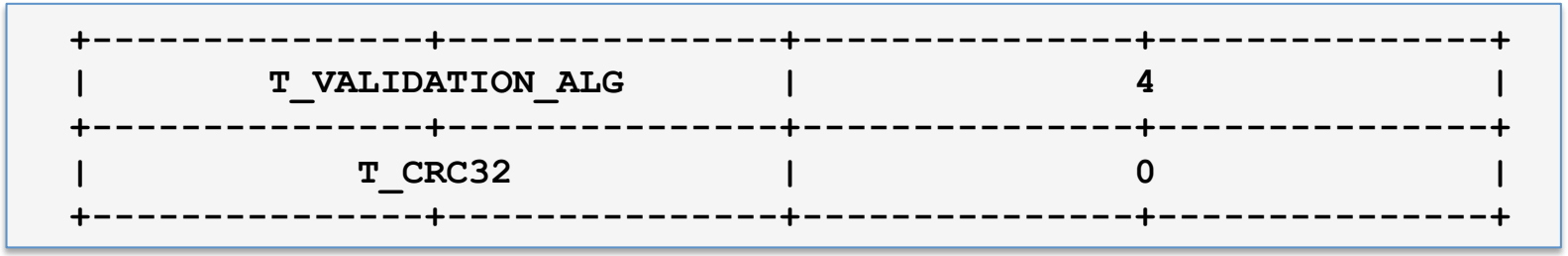

```
Headers     = Age
Metadata    = CreateTime / PayloadType / MaxAge
```

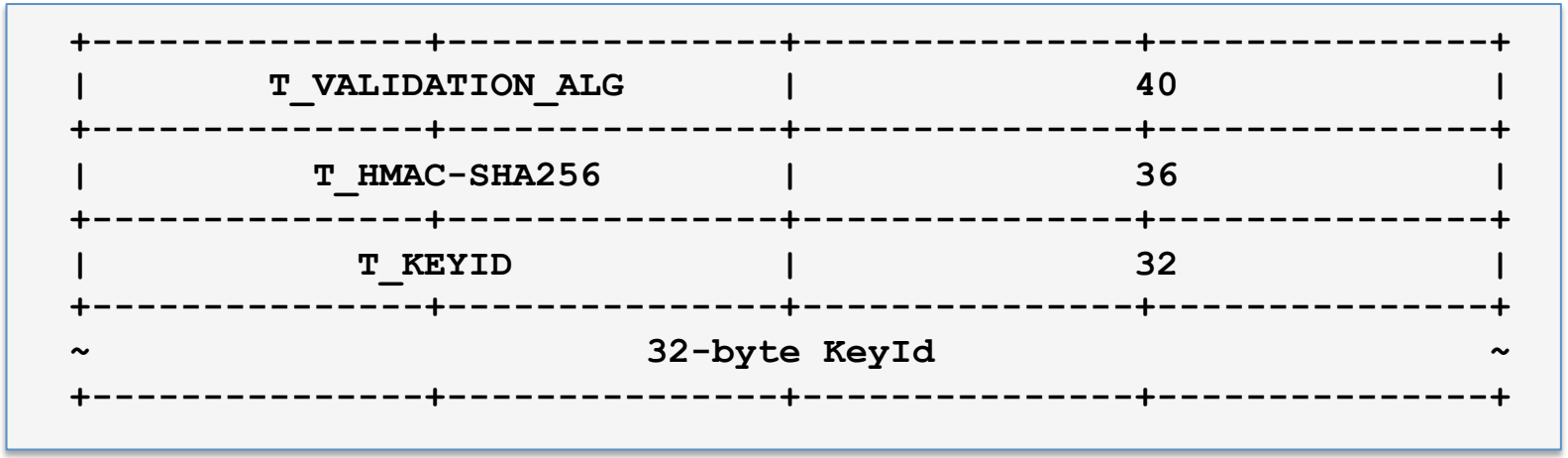
VALIDATION ALGORITHM



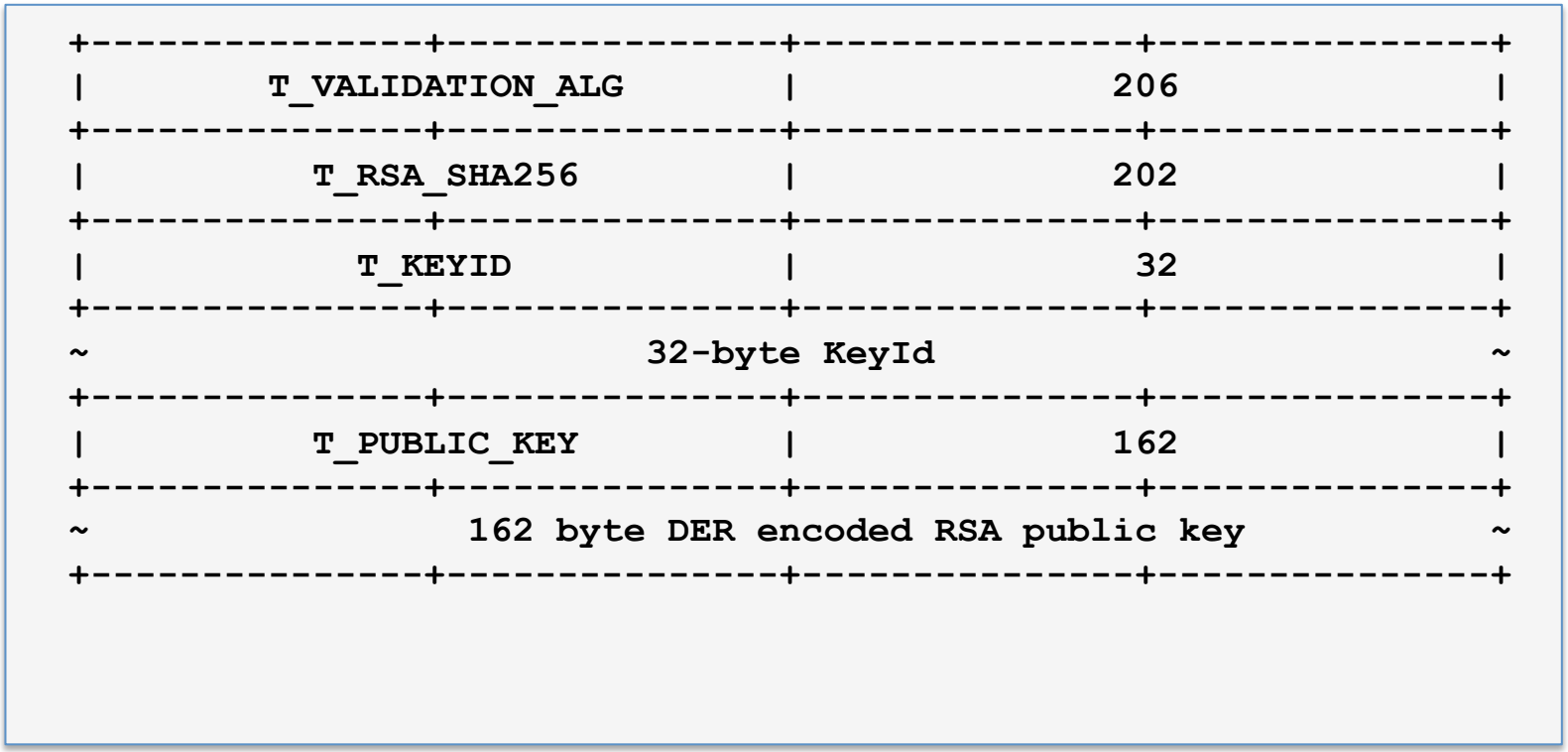

```
# Initial list of supported algorithms
ValidationType = CRC32C / RFC793 / HMAC-SHA256 / VMAC-128 /
                 RSA-SHA256 / EC-SECP-256K1 / EC-SECP-384R1

CRC32C      = (empty)
RFC793      = (emtpy)
HMAC-SHA261 = KeyId
VMAC-128    = KeyId
RSA-SHA256  = KeyId [KeyLocator]
EC-*        = KeyId [KeyLocator]
KeyLocator  = PublicKey / Certificate / KeyName
PublicKey   = (DER encoded public key)
Certificate = (DER encoded X.509 certificate)
KeyName     = Link
Link        = Name [KeyIdRestriction] [ContentObjectHashRestriction]
```

VALIDATION ALG EXAMPLES (1)

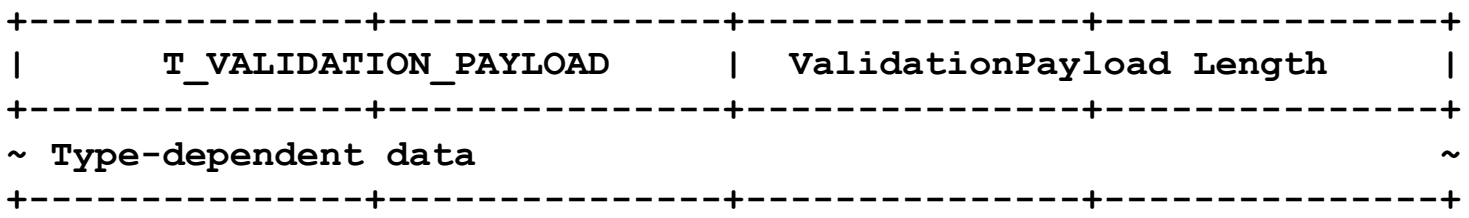




VALIDATION INFO EXAMPLES (2)



VALIDATION PAYLOAD



The validation payload depends on the verification type in the previous TLV.

For a CRC32, it's the 32-bytes of CRC.
For an HMAC-SHA256, it is the 32-byte SHA256 output.
For RSA-SHA256, it's the RSA signature.
Etc.