Security Proposal for PMCI Standards and Protocols

Architecture for Version 1.0 Release

Work in Progress

Last Updated: 12/17/2018

Disclaimer

- The information in this presentation represents a snapshot of work in progress within the DMTF.
- This information is subject to change without notice. The standard specifications remain the normative reference for all information.
- For additional information, see the Distributed Management Task Force (DMTF) website.

Acknowledgement

- Some of the content in this presentation is derived from USB Type-C Authentication specification 1.0 at https://www.usb.org/sites/default/files/documents/usb_authentication_20180904.zip

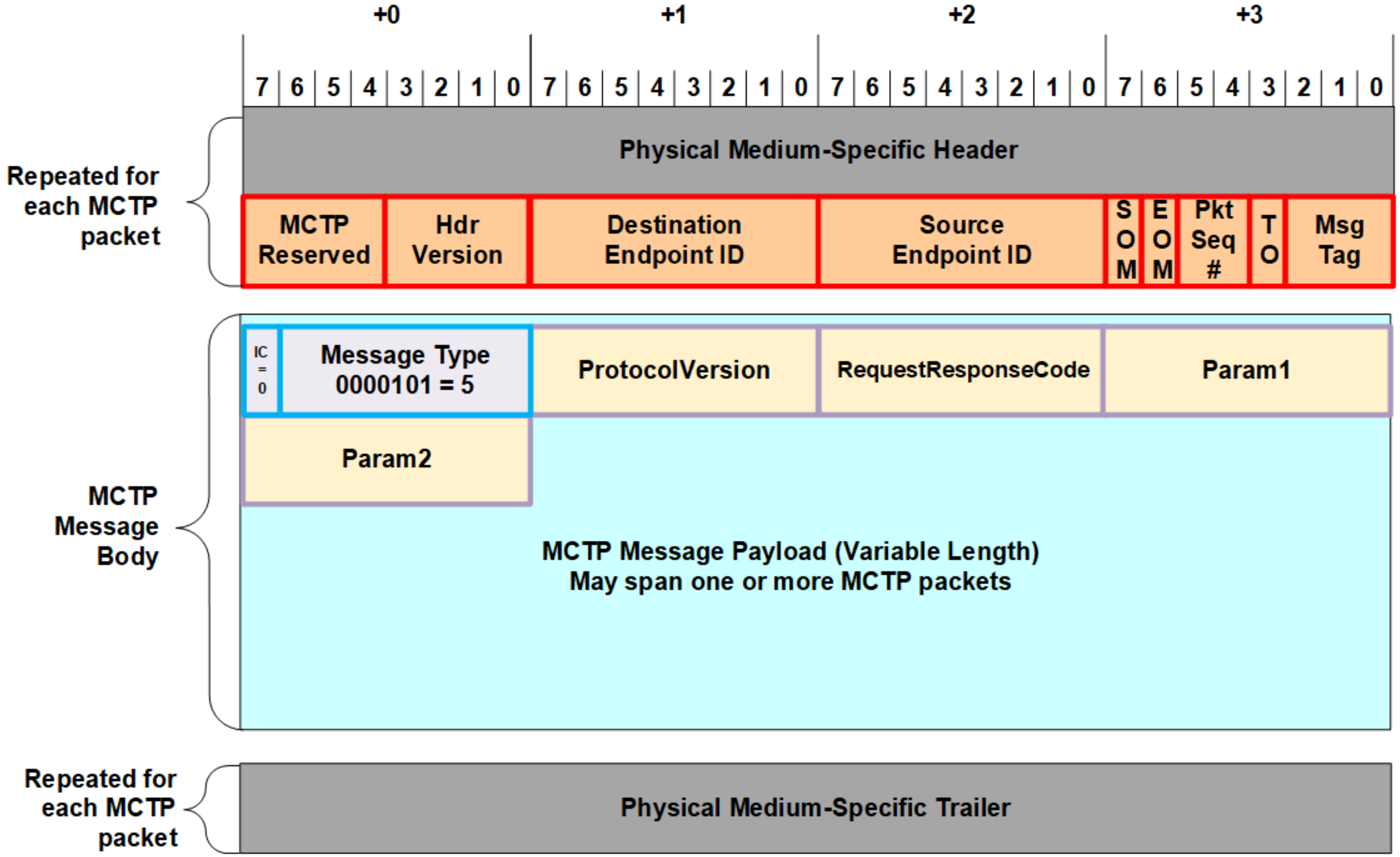
Guiding Principals

- Use MCTP message type 5 for all authentication commands including the future ones used for setting up secure sessions

- Use MCTP message type 6 for secured transport of encapsulated MCTP messages as appropriate (Future Version)

- Leverage USB Type-C format
  - No Completion Code field, uses request/response code to communicate errors
  - Refrain from optimizing a byte here and a byte there
  - Will reference USB Body's wherever appropriate and extend wherever appropriate.

MCTP Message Type 5 (Security Commands) Format



+0 +1 +2 +3

7 6 5 4 3 2 1 0 7 6 5 4 3 2 1 0 7 6 5 4 3 2 1 0 7 6 5 4 3 2 1 0

Repeated for each MCTP packet

Physical Medium-Specific Header

MCTP Reserved | Hdr Version | Destination Endpoint ID | Source Endpoint ID | SOM | EOM | Pkt Seq # | TO | Msg Tag

MCTP Message Body

IC = 0 | Message Type 0000101 = 5 | ProtocolVersion | RequestResponseCode | Param1

Param2

MCTP Message Payload (Variable Length)
May span one or more MCTP packets

Repeated for each MCTP packet

Physical Medium-Specific Trailer

Acknowledgement: Some of the content in this page is derived from USB Type-C Authentication specification 1.0.
https://www.usb.org/sites/default/files/documents/usb_authentication_20180904.zip

MCTP Type 5 Message Header Format

Offset (byte)	Field Name (USB 3.1)	Field Name (MCTP)	Size (bytes)	Definition
0	*ProtocolVersion*	*ProtocolVersion*	1	Version of the spec being followed.
1	*MessageType*	*RequestResponseCode*	1	Identifies type of request or type of response.
2	*Param1*	*Param1*	1	Meaning is specific to the Request/Response Code
3	*Param2*	*Param2*	1	Meaning is specific to the Request/Response Code

Acknowledgement: Some of the content in this page is derived from USB Type-C Authentication specification 1.0.
https://www.usb.org/sites/default/files/documents/usb_authentication_20180904.zip

RequestResponseCode Field: Part 1

Value	Type	Name	Description
80h	Request	Reserved	
81h	Request	GET_DIGESTS	Retrieve Cert chain digest
82h	Request	GET_CERTIFICATE	Retrieve segment of cert chain
83h	Request	CHALLENGE	Initiate authentication
84h – DFh	Request	Reserved	
E0h	Request	GET_MEASUREMENTS	Retrieve signed firmware measurement
E1h	Request	GET_CAPABILITIES	Retrieve capabilities
E2h	Request	SET_CERTIFICATE	Install new cert chain (slots 1-7 only)
E3h	Request	NEGOTIATE	Negotiate Cryptographic Algorithms
E4h – FEh	Request	Reserved	
FFh	N/A	Reserved	Do not use.

Gray rows – Not part of USB 3.1

Acknowledgement: Some of the content in this page is derived from USB Type-C Authentication specification 1.0.
https://www.usb.org/sites/default/files/documents/usb_authentication_20180904.zip

RequestResponseCode Field: Part 2

Value	Type	Name	Description
00h	Response	Reserved	
01h	Response	DIGESTS	Response to GET_DIGEST request.
02h	Response	CERTIFICATE	Response to GET_CERTIFICATE request.
03h	Response	CHALLENGE_AUTH	Response to CHALLENGE.
04h – 5Fh	Response	Reserved	
60h	Response	MEASUREMENTS	Response to GET_MEASUREMENTS request.
61h	Response	CAPABILITIES	Response to GET_CAPABILITIES request.
62h	Response	SET_CERT_RESPONSE	Response to SET_CERTIFICATE request.
63h	Response	ALGORITHMS	Response to NEGOTIATE request
64h – 7Eh	Response	Reserved	
7Fh	Response	ERROR	Response to any unsuccessful request.

Gray rows – Not part of USB 3.1

Acknowledgement: Some of the content in this page is derived from USB Type-C Authentication specification 1.0.
https://www.usb.org/sites/default/files/documents/usb_authentication_20180904.zip

GET_DIGESTS Request (same as Type-C)

This Request is used to retrieve Certificate Chain digests.

Offset	Field	Size	Value
0	*ProtocolVersion*	1	V1.0 = 01h
1	*Request/Response Code*	1	81h = GET_DIGESTS
2	*Param1*	1	Reserved
3	*Param2*	1	Reserved

Acknowledgement: Some of the content in this page is derived from USB Type-C Authentication specification 1.0.
https://www.usb.org/sites/default/files/documents/usb_authentication_20180904.zip

Successful DIGESTS Response

Offset	Field	Size	Value
0	*ProtocolVersion*	1	V1.0 = 01h
1	*Request/Response Code*	1	01h = DIGESTS
2	*Param1*	1	Capabilities Field; shall be set to 01h for this specification. All other values reserved.
3	*Param2*	1	Slot mask. The bit in position K of this byte shall be set to 1b if and only if slot number K contains a Certificate Chain for the protocol version in the *ProtocolVersion* field. (Bit 0 is the least significant bit of the byte.) The number of digests returned shall be equal to the number of bits set in this byte. The digests shall be returned in order of increasing slot number.
4	*Digest[0]*	H	H-byte digest of the first Certificate Chain. H is the size of the hashing algorithm output mutually agreed via NEGOTIATE request. This field is big endian.
…	…	…	…
4 + (H * (n -1))	*Digest[n-1]*	H	H-byte digest of the last (nth) Certificate Chain. H is the size of the hashing algorithm output mutually agreed via NEGOTIATE request. This field is big endian.

Acknowledgement: Some of the content in this page is derived from USB Type-C Authentication specification 1.0.
https://www.usb.org/sites/default/files/documents/usb_authentication_20180904.zip

GET_CERTIFICATE Request (same as Type-C)

This Request is used to retrieve Certificate Chains, one chunk at a time.

Offset	Field	Size	Value
0	*ProtocolVersion*	1	V1.0 = 01h
1	*Request/Response Code*	1	82h = GET_CERTIFICATE
2	*Param1*	1	Slot number of the target Certificate Chain to read from. The value in this field shall be between 0 and 7 inclusive.
3	*Param2*	1	Reserved
4	*Offset*	2	Offset in bytes from the start of the Certificate Chain to where the read request begins. This field is little endian.
6	*Length*	2	Length in bytes of the read request. This field is little endian.

Acknowledgement: Some of the content in this page is derived from USB Type-C Authentication specification 1.0.
https://www.usb.org/sites/default/files/documents/usb_authentication_20180904.zip

Successful CERTIFICATE Response (Same as Type-C)

Offset	Field	Size	Value
0	*ProtocolVersion*	1	V1.0 = 01h
1	*Request/Response Code*	1	02h = CERTIFICATE
2	*Param1*	1	Slot number of the Certificate Chain returned
3	*Param2*	1	Reserved
4	*CertChain*	*Length*	Data Requested contents of target Certificate Chain. Format defined in USB Type-C Authentication specification 1.0.

Acknowledgement: Some of the content in this page is derived from USB Type-C Authentication specification 1.0.
https://www.usb.org/sites/default/files/documents/usb_authentication_20180904.zip

CHALLENGE Request

This Request is used to authenticate an endpoint.

Offset	Field	Size	Value
0	*ProtocolVersion*	1	V1.0 = 01h
1	*Request/Response Code*	1	83h = CHALLENGE
2	*Param1*	1	Slot number of the recipient's Certificate Chain that will be used for Authentication
3	*Param2*	1	Reserved
4	*Nonce*	H	Random H-byte nonce chosen by the Authentication Initiator. H is the size of the hashing algorithm output mutually agreed via NEGOTIATE request.

Acknowledgement: Some of the content in this page is derived from USB Type-C Authentication specification 1.0.
https://www.usb.org/sites/default/files/documents/usb_authentication_20180904.zip

Successful CHALLENGE_AUTH Response

Offset	Field	Size	Value
0	*ProtocolVersion*	1	V1.0 = 01h
1	*Request/Response Code*	1	03h = CHALLENGE_AUTH
2	*Param1*	1	Shall contain the Slot number in the *Param1* field of the corresponding CHALLENGE Request
3	*Param2*	1	Slot mask. The bit in position K of this byte shall be set to 1b if and only if slot number K contains a Certificate Chain for the protocol version in the *ProtocolVersion* field. (Bit 0 is the least significant bit of the byte.)
4	*MinProtocolVersion*	1	Minimum protocol version supported by this Device
5	*MaxProtocolVersion*	1	Maximum protocol version supported by this Device
6	*Capabilities*	1	Set to 01h for this specification. All other values reserved
7	*Reserved*	1	Reserved
8	*CertChainHash*	H	Hash of the Certificate Chain used for Authentication. H is the size of the hashing algorithm output mutually agreed via NEGOTIATE request. This field is big endian.
8+H	*Salt*	H	Value chosen by the Authentication Responder. H is the size of the hashing algorithm output mutually agreed via NEGOTIATE request. *Note: the Salt can be random, fixed, or any other value*
8+2H	*Context Hash*	H	Hash over device specific information. This field is big endian.
8+3H	*Signature*	S	Signature over all bytes in the Authentication request and response payload (starting with ProtocolVersion fields) excluding this field. S is the size of the asymmetric signing algorithm output mutually agreed via NEGOTIATE request. This field is little endian.

Acknowledgement: Some of the content in this page is derived from USB Type-C Authentication specification 1.0.
https://www.usb.org/sites/default/files/documents/usb_authentication_20180904.zip

GET_MEASUREMENTS Request

This Request is used to retrieve measurements of mutable firmware component(s) that the recipient endpoint is executing.

Offset	Field	Size	Value
0	*ProtocolVersion*	1	V1.0 = 01h
1	*Request/Response Code*	1	E0h = GET_MEASUREMENTS
2	*Param1*	1	Reserved
3	*Param2*	1	Reserved
4	*Reserved*	2	Reserved to be compatible with the Cerberus definition.
6	*Nonce*	H	Random H-byte nonce chosen by the Authentication Initiator. H is the size of the hashing algorithm output mutually agreed via NEGOTIATE request.

Successful MEASUREMENTS Response

Offset	Field	Size	Value
0	*ProtocolVersion*	1	V1.0 = 01h
1	*Request/Response Code*	1	60h = MEASUREMENTS
2	*Param1*	1	Reserved
3	*Param2*	1	Reserved
4	*Length*	2	Length in bytes
6	*NumberofMeasurements (N)*	1	Number of Measurement hashes
7	*MeasurementLength (L)*	1	Length in bytes for each Measurement hash
8	*Measurements*	*L*N*	Concatenation of all Measurement hashes
8 + (*L*N*)	*Signature*	*S*	Signature of the GET_MEASUREMENT Request and MEASUREMENT Response messages, excluding the Signature field and signed using the Device Private Key. The size of the Signature field depends on the asymmetric signing algorithm that was mutually agreed upon via NEGOTIATE.

GET_CAPABILITIES Request

This Request is used to discover endpoint capabilities.

Offset	Field	Size	Value
0	*ProtocolVersion*	1	V1.0 = 01h
1	*Request/Response Code*	1	E1h = GET_CAPABILITIES
2	*Param1*	1	Reserved
3	*Param2*	1	Reserved

Successful CAPABILITIES Response

Offset	Field	Size	Value
0	*ProtocolVersion*	1	V1.0 = 01h
1	*Request/Response Code*	1	61h = CAPABILITIES
2	*Param1*	1	Reserved
3	*Param2*	1	Reserved
4	*DetailedVersion*	3	The remaining 3 bytes that are concatenated with Offset 0 to form the complete PMCI specification version. Offset 0 describes the major version.
7	*CT*	1	Timeout value associated with CHALLENGE and GET_MEASUREMENT operations in uS, expressed in logarithmic (base 2) scale. This value is added to media specific timeout value when deriving Request-to-response timeout for CHALLENGE and GET_MEASUREMENT requests.
8	*Flags*	4	Bit 0 – Reserved for future version Bit 1 – Supports GET_DIGEST, GET_CERTIFICATE and CHALLENGE requests Bit 2 – Supports SET_CERTIFICATE request Bit 3 – Support GET_MEASUREMENT All other bits are reserved for future extension.

SET_CERTIFICATE Request

This Request is used to install new certificate chain(s).

Offset	Field	Size	Value
0	*ProtocolVersion*	1	V1.0
1	*Request/Response Code*	1	E2h = SET_CERTIFICATE
2	*Param1*	1	Slot number of the target Certificate Chain to read from. The value in this field shall be between 1 and 7 inclusive. If 0 is used in this parameter, the Device shall return an Error Response message.
3	*Param2*	1	Reserved
4	*CertChain*	*Length*	Data Contents of target Certificate Chain to be updated.

Successful SET_CERTIFICATE_RESPONSE

Offset	Field	Size	Value
0	*ProtocolVersion*	1	V1.0 = 01h
1	*Request/Response Code*	1	62h = GET_CERT_RESPONSE
2	*Param1*	1	Reserved
3	*Param2*	1	Reserved

NEGOTIATE Request

Offset	Field	Size	Value
0	*ProtocolVersion*	1	V1.0
1	*Request/Response Code*	1	E3h = NEGOTIATE
2	*Param1*	1	Reserved
3	*Param2*	1	Reserved
4	*Length*	2	Length of the request packet in bytes
6	*BaseAsymAlgo*	2	Bit vector listing PMCI enumerated asymmetric algorithms supported by requestor. Bit 0 – RSA 2048; Bit 1 – RSA 4096; Bit 2 – ECDSA 256; Bit 3 – ECDSA 384
8	*BaseHashAlgo*	2	Bit vector listing PMCI enumerated hashing algorithms supported by requestor. Bit 0 – SHA2-256 ; Bit 1 – SHA3-512
10	*Reserved*	4	Reserved for future use
14	*ExtAsymCount*	1	Number of extended Asymmetric algorithms supported by requestor (=A)
15	*ExtHashCount*	1	Number of extended Hashing algorithms supported by requestor (=H)
16	*Reserved*	2	Reserved for future use
18	*ExtAsym*	2A	List of the extended asymmetric algorithms supported by requestor (for encoding, see TCG)
18+2A	*ExtHash*	2H	List of the extended Hashing algorithms supported by requestor (for encoding, see TCG)
18+2A+2H	*Reserved*	-	Reserved for future expansion. Consult the Length field (offset 4) to determine the number of bytes in the request.

Successful ALGORITHMS Response

Offset	Field	Size	Value
0	*ProtocolVersion*	1	V1.0
1	*Request/Response Code*	1	63h = ALGORITHMS
2	*Param1*	1	Reserved
3	*Param2*	1	Reserved
4	*Length*	2	Length of the request packet in bytes
6	*BaseAsymSel*	2	Bit vector listing PMCI enumerated asymmetric algorithms selected. No more than 1 bit can be set.
8	*BaseHashSel*	2	Bit vector listing PMCI enumerated hashing algorithms selected. No more than 1 bit can be set.
10	*Reserved*	4	Reserved for future use
14	*ExtAsymSelCount*	1	The number of extended Asymmetric algorithms selected. Either 0 or 1. (=A)
15	*ExtHashSelCount*	1	The number of extended Hashing algorithms selected. Either 0 or 1. (=H)
16	*Reserved*	2	
18	*ExtAsymSel*	2A	List of the extended asymmetric algorithms selected (for encoding, see TCG)
18+2A	*ExtHashSel*	2H	List of the extended Hashing algorithms selected (for encoding, see TCG)
18+2A+2H	*Reserved*	-	Reserved for future expansion. Consult the length field (offset 4) to determine the number of bytes in the response.

ERROR Response

Offset	Field	Size	Value
0	*ProtocolVersion*	1	Minimum Supported protocol version, V1.0 for now
1	*Request/Response Code*	1	7Fh = ERROR
2	*Param1*	1	Error Code.
3	*Param2*	1	Error Data. See Table 5-18.

Error Code	Value	Description	Error Data
Reserved	00h	Reserved	Reserved
INVALID_REQUEST	01h	One or more Request fields are invalid	00h
UNSUPPORTED_PROTOCOL	02h	Requested Security Protocol Version is not supported	Maximum supported Security Protocol Version1
BUSY	03h	Device cannot respond now, but will be able to respond in the future	00h
UNSPECIFIED	04h	Unspecified error occurred	00h
Reserved	05h-EFh	Reserved	Reserved
Vendor Defined	F0h- FFh	Vendor defined	Vendor defined

Acknowledgement: Some of the content in this page is derived from USB Type-C Authentication specification 1.0.
https://www.usb.org/sites/default/files/documents/usb_authentication_20180904.zip

Timeouts

Timing Specification	Symbol	Min	Max	Description
Number of request retries	AN1	2	See description	Total of three tries, minimum: the original try plus two retries. The maximum number of retries for a given request is limited by the requirement that all retries shall occur within MT4 from the corresponding media spec, max of the initial request.
Request-to-response time	AT1	-	MT1+CT	MT1 is the request-to-response timing defined in the media binding spec. CT is the allowance for crypto operations and reported via CAPABILITY response.
Time-out waiting for a response	AT2	AT1 max + MT2 min – MT1 max	MT4 min	MT2 min and MT1 max are defined in the appropriate media binding specification.

Acknowledgement: Some of the content in this page is derived from USB Type-C Authentication specification 1.0.
https://www.usb.org/sites/default/files/documents/usb_authentication_20180904.zip