RPMC Support Using eSPI OOB (eRPMC)

Architecture Specification

Rev. 0.81
June 2025

Doc. No.: 739180, Rev.: 0.81

You may not use or facilitate the use of this document in connection with any infringement or other legal analysis concerning Intel products described herein.

No license (express or implied, by estoppel or otherwise) to any intellectual property rights is granted by this document.

All information provided here is subject to change without notice. Contact your Intel representative to obtain the latest Intel product specifications and roadmaps

All product plans and roadmaps are subject to change without notice.

The products described may contain design defects or errors known as errata, which may cause the product to deviate from published specifications. Current characterized errata are available on request.

Intel technologies' features and benefits depend on system configuration and may require enabled hardware, software or service activation. Performance varies depending on system configuration. No computer system can be absolutely secure. Check with your system manufacturer or retailer or learn more at intel.com.

Intel disclaims all express and implied warranties, including without limitation, the implied warranties of merchantability, fitness for a particular purpose, and non-infringement, as well as any warranty arising from course of performance, course of dealing, or usage in trade.

Intel, the Intel logo, and other Intel marks are trademarks of Intel Corporation or its subsidiaries.

*Other names and brands may be claimed as the property of others.

Contents

Revision History

Document Number	Revision Number	Description	Revision Date
739180	0.7	• Initial Release	August 2022
	0.7.1	• Updated Read RPMC Parameters response definition Updated: Byte count & length • Updated document chapterssssss numbering	March 2023
	0.8	• Updated Chapter 2	February 2025
	0.81	• Updated Legal Disclaimer	June 2025

References

1. Intel Serial Flash Hardening Product External Architecture Specification (EAS) Revision 0.7. (Link)
2. Intel Enhanced Serial Peripheral Interface (eSPI) Interface Base Specification (for Client and Server Platforms) January 2016 Revision 1.0. (Link)
3. Intel Enhanced Serial Peripheral Interface (eSPI) Compatibility Specification For 2017, 2018, 2019, 2020 Client, Desktop and Server Platforms February 2019 Revision 0.7
4. DMTF Management Component Transport Protocol (MCTP) Base Specification, DSP0236, Version: 1.3.1 (Link)
5. DMTF Management Component Transport Protocol (MCTP) SMBus/I2C Transport Binding Specification, DSP0237, Version: 1.1.0 4 (Link)
6. eRPMC enablement guide (search for “eRPMC enablement guide” at Link)

Introduction

RPMC functionality is defined in Intel Serial Flash Hardening Product External Architecture Specification (EAS) Revision 0.7 (document link).

Certain types of attacks can be detected by using monotonic counter values in the platform. RPMC implements authenticated commands to the protected monotonic counters.

Adhering to the above RPMC specification, this document outlines Firmware support for RPMC implemented in an Embedded Controller (EC device) over eSPI as Out-Of-Band (OOB) authenticated messages.

EC device must meet all the associated HW requirements for RPMC providing the necessary authentication and secure internal nonvolatile storage for RPMC keys and counters.

EC firmware **must** be cryptographically signed and serve as the Root of Trust (RoT), ensuring its integrity and authenticity through a secure boot process and verified firmware updates

eRPMC Implementation

Firmware support for RPMC is added in an EC device.

The RPMC commands are implemented as OOB (Tunneled SMBus) Messages (Ref [2]). These messages include SMBus Management Component Transport Protocol (MCTP) packets.

Communication over the eSPI OOB channel is with firmware in the Intel PCH's CSME microcontroller.

The ME controller will issue commands to the EC device as SMBus packets tunneled through eSPI as Out-Of-Band (OOB) messages.

- The entire SMBus packet including the SMBus Slave Address, SMBus Command Opcode, SMBus Byte Count, SMBus Data fields and optional PEC byte are sent as data within the eSPI OOB message packet
- The SMBus Data fields include SMBus Management Component Transport Protocol (MCTP) packets
- The RPMC commands are sent as data within the MCTP Data fields in the eSPI OOB message packet.

The OOB packets use SMBus addressing to specify the endpoints of communication.

The EC device will respond to the RPMC commands contained in the eSPI OOB packets with eSPI OOB response packets containing status or status and data, depending on the RPMC command received.

The following commands are supported over the OOB channel (Ref [1]):

- RPMC Write Root Key Register
- RPMC Update HMAC Key Register
- RPMC Increment Monotonic Counter
- RPMC Request Monotonic Counter
- Read RPMC Parameters
- The RPMC Read Data is not required since the status is returned in response to each command and both status and counter data is returned as a response to the RPMC Request Monotonic Counter command.

The root key must be stored in internal secure nonvolatile storage such as EEPROM. Size is 256 bits.

The HMAC Key is stored in HMAC Key Register(s)/RAM/NVRAM. Size is 256 bits per HMAC key per counter. HMAC keys get invalidated at each power cycle.

The device supports minimum 4 to 256 monotonic counters and can be extended to support more as the platform RPMC counter needs evolve.

The number of supported monotonic counters will be returned as a response to the “Read RPMC Parameters” command.

The commands support RPMC implemented in firmware in the EC device as well as passing commands to one or multiple RPMC SPI flash devices. Supports up to three devices including EC as RPMC device. RPMC Authenticated Commands

Certain types of attacks can be detected by using monotonic counter values in the platform. RPMC implements authenticated commands to the protect monotonic counters. The authenticated commands require the following:

- A Root key programmed into the device during manufacturing
- An HMAC key derived from the Root Key at run time. HMAC keys gets invalidated at each power cycle.

The following commands are supported:

- Command to write 256 bit “Root Key”.
  - The root key is stored in secure internal non-volatile memory and is not readable from outside. This includes test modes. A non “0FF..FF” root key is programmed only one-time during system manufacturing.
  - When this request is received error-free only the corresponding Monotonic Counter is initialized to 0 if previously uninitialized. This state is used to leave the monotonic counters at the current value when a subsequent error free Root Key Register Write operation is received. (Both 256’HFF..FF and non 256’HFF..FF)
- Authenticated commands/responses are signed using the “HMAC Key”. The signature is verified using HMAC-SHA-256.
  - The HMAC key is stored inside the device and is not readable including via test modes.
- Authenticated “HMAC key update command” to derive a 256-bit HMAC key. The HMAC key is derived from the Root Key and Key data supplied during the command using HMAC-SHA-256. This command performs two HMAC-SHA-256 operations: one to derive the HMAC key and one to verify the signature.
- Authenticated commands to support following monotonic counter operations:
  - Increment counter
  - Read counter
- There is no mechanism to circumvent Authenticated commands including via test modes.

> **Note:** Recommendation is EC to complete all eRPMC commands within 50ms to reduce and boot time impact; must complete within a max of 100ms.

eRPMC - RPMC Commands as OOB over eSPI (Tunneled SMBus Messages – MCTP)

The eSPI OOB block write packet is used for this implementation, where the SMBus payload includes the SMBus source Slave address, MCTP header and RPMC command.

The below figure shows the bytes in the OOB MCTP packet.

Figure 3-1. OOB MCTP Packet

diagram1.png

Format Used for RPMC OOB Commands

RPMC OOB MCTP Command, PCH to eSPI Slave:

Byte #	7	6	5	4	3	2	1	0
0	eSPI Cycle Type: OOB Message=21h							
1	Tag[3:0]=0h				Length[11:8]=0h			
2	Length[7:0]							
3	Dest Slave Addr[7:1]=07h (eSPI Slave/EC)							0
4	SMB Command Code = 0Fh (MCTP)							
5	Byte Count							
6	Source Slave Address[7:1]= 08h (PCH)							1
7	MCTP Reserved=0h				Header Version			

8	Destination Endpoint ID				
9	Source Endpoint ID				
10	SOM	EOM	Packet Seq #	TO	Message Tag
11	IC	Message Type			
12	1st Message Header/Data				
13	2nd Message Header/Data				
...	...				

RPMC OOB MCTP Command, eSPI Slave to PCH:

Byte #	7	6	5	4	3	2	1	0
0	eSPI Cycle Type: OOB Message=21h							
1	Tag[3:0]=0h				Length[11:8]=0h			
2	Length[7:0]							
3	Dest Slave Addr[7:1]=08h (PCH)							0
4	SMB Command Code = 0Fh (MCTP)							
5	Byte Count							

6	Source Slave Address[7:1]= 07h (eSPI Slave/EC)				1
7	MCTP Reserved=0h			Header Version	
8	Destination Endpoint ID				
9	Source Endpoint ID				
10	SOM	EOM	Packet Seq #	TO	Message Tag
11	IC	Message Type			
12	1st Message Header/Data				
13	2nd Message Header/Data				
...	...				

The optional PEC optional byte for SMBus may follow the Payload:

- The SMBus Byte Count field does not include the PEC byte
- The Length field of the OOB message includes the PEC byte
  - The presence of the PEC byte is signaled by a 1-byte discrepancy between the eSPI Received Length register and the Byte Count as given in the SMBus header.

Maximum Payload Size (MPS)

The EC device supports an OOB MPS of 64 bytes (73 bytes for MCTP, refer Ref [2]). This requires any OOB message with an MCTP payload size greater than 64 bytes to be broken up into two messages.

Note: An expansion of the signature field for an extension of the RPMC standard to SHA-384 will also require several commands to be broken up into two messages.

SMB/MCTP Parameters

The SMB/MCTP message parameters are as follows:

SMB/MCTP Parameters:

Parameter	Value	Comment
eSPI Cycle Type	21h	OOB Message
Tag[3:0]	0h	Ignored
Length[11:8]	0h	
Length[7:0]	Command dependent	
Destination/Source Slave Address[7:1]	08h for Intel PCH/ME OOB MCTP Handler 07h for eSPI Slave/EC	Intel ME OOB MCTP Handler address[7:0]=10h(Dest)/11h(Src) eSPI/EC Address[7:0]=0Eh(Dest)/0Fh(Src)
SMB Command Code	0Fh	0Fh for MCTP over SMBus messages. Ref[5]
Byte Count	Command dependent	
Header Version	1h	Ref[5]
Destination Endpoint ID	CSME: 0x50; EC: 0x40	EID (Endpoint identifier) for the endpoint to receive MCTP packet.
Source Endpoint ID	CSME: 0x50; EC: 0x40	EID of the originator of MCTP packet.

SOM (Start of Message flag)	0b/1b	Set to 1b if this packet is the first packet of a message. Ref[4]
EOM (End of Message flag)	0b/1b	Set to 1b if this packet is the last packet of a message. Ref[4]
Packet Seq # (Packet sequence number)	00b/01b	00b for messages that do not span multiple packets. Ref[4]
TO (Tag Owner)	1b	Set to 1b to indicate that the source of the message originated the message tag. Ref[4]
Message Tag	0	Message tag (along with the Source Endpoint IDs and TO field) identifies a unique message at the MCTP transport level
IC (Integrity Check bit)	0b	0b = No MCTP message integrity check. Ref[4]
Message Type	7Dh	Message Type to be requested from DMTF
RPMC Device	00h-03h	Up to four devices including EC as RPMC device and SPI flash devices
MCTP Message Payload	Command dependent	Includes the following: IC/Message Type byte RPMC Device RPMC packet payload

RPMC Commands

RPMC Command Codes:

Command	Command OpCode	OpCode Value	OP1 CmdType	Response
Write Root Key Register	OP1	9Bh	00h	RPMC Device, Counter Addr & Status
Update HMAC Key Register	OP1	9Bh	01h	RPMC Device, Counter Addr & Status
Increment Monotonic Counter	OP1	9Bh	02h	RPMC Device, Counter Addr & Status
Request Monotonic Counter	OP1	9Bh	03h	RPMC Device, Counter Addr, Status and Counter Data
Read RPMC Parameters	N/A	9Fh (Note)	N/A	RPMC Parameters

Note: This is not a standard RPMC command and therefore 9Fh is not an RPMC defined OpCode.

All individual fields are Byte wide fields. For a multi-byte field, Most Significant Byte is issued first; Last Significant Byte is issued last. Within a Byte, Most Significant Bit is issued first; Least Significant Bit is issued last.

When an OP1 command is received, the EC device responds a shown in the “RPMC Command Codes” table above when the command completes.

Note: All commands defined below use SHA-256. SHA-384 is left for a future update.

Write Root Key Register

This command is used to initialize the Root Key Register corresponding to the received Counter Address with the received Root Key. It is expected to be used in an OEM manufacturing environment.

RPMC Write Root Key Register OOB Command (all Payload Bytes):

Byte #	7	6	5	4	3	2	1	0
0	eSPI Cycle Type: OOB Message=21h							
1	Tag[3:0]=0h				Length[11:8]=0h			
2	Length[7:0]=4Ah							
3	Dest Slave Addr[7:1]=07h (eSPI Slave/EC)							0
4	Command Code = 0Fh							
5	Byte Count=47h							
6	Source Slave Address[7:1]= 08h (PCH)							1
7	MCTP Reserved=0h				Header Version			
8	Destination Endpoint ID							
9	Source Endpoint ID							
10	SOM	EOM	Packet Seq #			TO	Message Tag	
11	IC	Message Type=7Dh						
12	RPMC Device							
13	Opcode = 9Bh							

14	Cmd Type = 00h
15	Counter Addr[7:0]
16	Rsvd=00h
17	Root Key[255:248]
...	...
48	Root Key[7:0]
49	TruncatedSignature[223:216]
...	...
74	TruncatedSignature[23:16]
75	TruncatedSignature[15:8] (In second message, see below)
76	TruncatedSignature[7:0] (In second message, see below)

Since the number of bytes in the MCTP payload of this message is greater than 64, it will be sent in two messages as follows:RPMC Write Root Key Register OOB Command, First Message:

Byte #	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
0	eSPI Cycle Type: OOB Message=21h							
1	Tag[3:0]=0h				Length[11:8]=0h			

2	Length[7:0]=48h				
3	Dest Slave Addr[7:1]=07h (eSPI Slave/EC)				0
4	Command Code = 0Fh				
5	Byte Count=45h				
6	Source Slave Address[7:1]= 08h (PCH)				1
7	MCTP Reserved=0h			Header Version	
8	Destination Endpoint ID				
9	Source Endpoint ID				
10	SOM=1b	EOM= 0b	Packet Seq #=00b	TO	Message Tag
11	IC	Message Type=7Dh			
12	RPMC Device				
13	Opcode = 9Bh				
12	Cmd Type = 00h				
13	Counter Addr[7:0]				
14	Rsvd=00h				

15	Root Key[255:248]
...	...
46	Root Key[7:0]
47	TruncatedSignature[223:216]
...	...
72	TruncatedSignature[23:16]

RPMC Write Root Key Register OOB Command, Second Message:

Byte #	7	6	5	4	3	2	1	0
0	eSPI Cycle Type: OOB Message=21h							
1	Tag[3:0]=0h				Length[11:8]=0h			
2	Length[7:0]=0Bh							
3	Dest Slave Addr[7:1]=07h (eSPI Slave/EC)							0
4	Command Code = 0Fh							
5	Byte Count=08h							
6	Source Slave Address[7:1]= 08h (PCH)							1

7	MCTP Reserved=0h				Header Version
8	Destination Endpoint ID				
9	Source Endpoint ID				
10	SOM=0b	EOM=1b	Packet Seq #=01b	TO	Message Tag
11	IC	Message Type=7Dh			
12	TruncatedSignature[15:8]				
13	TruncatedSignature[7:0]				

Response:

Byte #	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
0	eSPI Cycle Type: OOB Message=21h							
1	Tag[3:0]=0h				Length[11:8]=0h			
2	Length[7:0]=0Ch							
3	Dest Slave Addr[7:1]=08h							0
4	Command Code = 0Fh							
5	Byte Count=09h							

6	Source Slave Address[7:1]= 07h				1
7	MCTP Reserved=0h			Header Version	
8	Destination Endpoint ID				
9	Source Endpoint ID				
10	SOM	EOM	Packet Seq #	TO	Message Tag
11	IC	Message Type=7Dh			
12	RPMC Device				
13	Counter Addr[7:0]				
14	Extended Status[7:0]				

After the command is issued, the device ensures that the received transaction is error free. This includes checking following conditions:

- RPMC message payload size is correct (including OP1 is 64 bytes)
- Counter Address falls within the range of supported counters.
- The Root Key Register corresponding to the requested Counter Address was previously uninitialized or initialized using a test/temporary key.
- Truncated signature field is the same as least significant 224 bits of HMAC-SHA- 256 based signature computed based on received input parameters:
  - HMAC message[31:0] = (OpCode[7:0], CmdType[7:0], CounterAddr[7:0], Reserved[7:0])
  - HMAC Key[255:0] = Root_Key[255:0]

If the received transaction is error free the device successfully executes the command and posts “successful completion” extended status. This command is executed to ensure that power cycling in the middle of command execution is properly handled. This requires that the internal state tracking the root key register initialization is written as the last operation of the command execution (Ref [1]).

Root Key Register Write with root key is = 256’HFF...FF is used as a temporary key. When this request is received error-free only the corresponding Monotonic Counter is initialized to 0 if

previously uninitialized. This state is used to leave the monotonic counters at the current value when a subsequent error free Root Key Register Write operation is received. (Both 256'HFF..FF and non 256'HFF..FF)

Once this command is successfully executed with a non 256'hFF..FF Root Key, the device will not accept the "Write Root Key Register" command any more, and the Root Key value cannot be read out by any instructions.

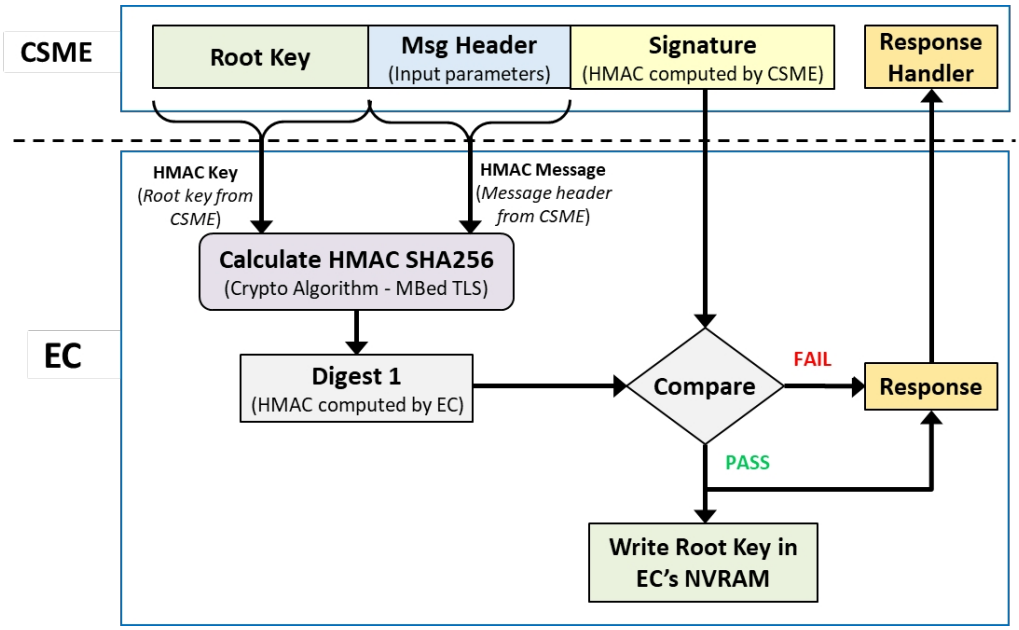
If the received transaction has errors, the device does not execute the transaction and posts the corresponding error in extended status.

Expected Extended Status [7:0] results:

Extended Status [7:0]	Applicable CmdType(s)	Description
10000000	00h	Successful completion
0XXXXXX1	00h	N/A. This bit cannot be read as 1.
0XXXXX1X	00h	This bit is only set when correct payload size is received. It is set on Root Key Register Overwrite or Counter Address is out of range or when there is a truncated signature mismatch error
0XXXX1XX	00h	This bit is set on Counter Address out of range when correct payload size is received; or CmdType is out of range; or incorrect payload size is received.

Flow Diagram for Command:

Figure 3-2. Write Root Key



Update HMAC Key Register

This command is used to update the HMAC-Key register corresponding to the received Counter Address with a new HMAC key calculated based on received input.

This command is issued on every power cycle event. This HMAC key is stored in volatile memory.

RPMC Update HMAC Key Register OOB Command:

Byte #	7	6	5	4	3	2	1	0
0	eSPI Cycle Type: OOB Message=21h							
1	Tag[3:0]=0h				Length[11:8]=0h			
2	Length[7:0]=32h							
3	Dest Slave Addr[7:1]=07h							0
4	Command Code = 0Fh							

5	Byte Count=2Fh				
6	Source Slave Address[7:1]= 08h				1
7	MCTP Reserved=0h			Header Version	
8	Destination Endpoint ID				
9	Source Endpoint ID				
10	SOM	EOM	Packet Seq #	TO	Message Tag
11	IC	Message Type=7Dh			
12	RPMC Device				
13	Opcode = 9Bh				
14	Cmd Type = 01h				
15	Counter Addr[7:0]				
16	Rsvd=00h				
17	Key Data[31:24]				
...	...				
20	Key Data[7:0]				

21	Signature[255:248]
...	...
52	Signature[7:0]

Response:

Byte #	7	6	5	4	3	2	1	0
0	eSPI Cycle Type: OOB Message=21h							
1	Tag[3:0]=0h				Length[11:8]=0h			
2	Length[7:0]=0Ch							
3	Dest Slave Addr[7:1]=08h							0
4	Command Code=0Fh							
5	Byte Count=09h							
6	Source Slave Address[7:1]= 07h							1
7	MCTP Reserved=0h				Header Version			
8	Destination Endpoint ID							
9	Source Endpoint ID							

10	SOM	EOM	Packet Seq #	TO	Message Tag
11	IC	Message Type=7Dh			
12	RPMC Device				
13	Counter Addr[7:0]				
14	Extended Status[7:0]				

After the command is issued, the device ensures that the received transaction is error free. This includes checking following conditions:

- RPMC message payload size is correct (including OP1 = 40 bytes)
- Counter Address falls within the range of supported counters.
- The Monotonic Counter corresponding to the requested Counter Address was previously initialized.
- Signature matches the HMAC-SHA-256 based signature computed based on received input parameters. This command performs two HMAC-SHA-256 operations.
  - HMAC-SHA-256 Operation 1 Output = HMAC_Storage[255:0]
    - HMAC Message[31:0] = KeyData[31:0]
    - HMAC Key[255:0] = Root_Key_Register[CounterAddr][255:0]
  - HMAC-SHA-256 Operation 2 Output = HMAC-SHA-256 based signature[255:0]
    - HMAC message[63:0] = (OpCode[7:0],CmdType[7:0].CounterAddr[7:0].Reserved[7:0], KeyData[31:0])
    - HMAC Key[255:0] = HMAC_Storage[255:0]

If the received transaction is error free the device successfully executes the command and posts “successful completion” extended status.

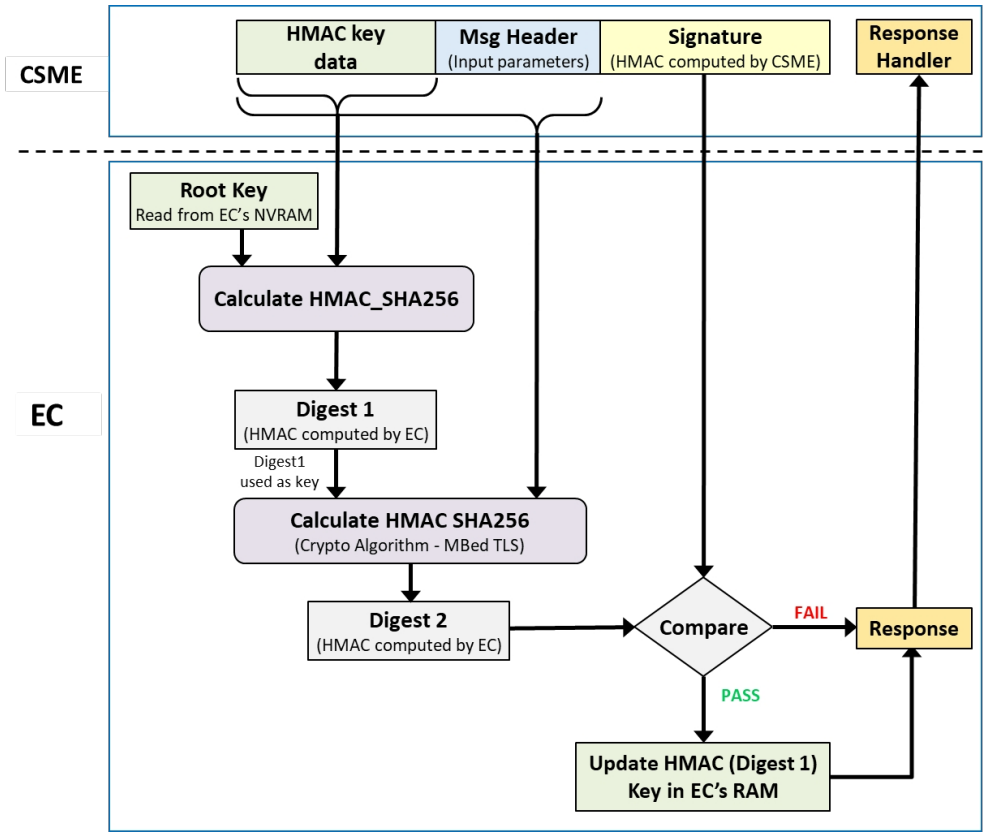
If the received transaction has errors, the device does not execute the transaction and posts the corresponding error in extended status.

Expected Extended Status [7:0] results:

Extended Status [7:0]	Applicable CmdType(s)	Description
10000000	01h	This status is set on successful completion (no errors) of OP1 command.
0XXXXXX1	01h	N/A. This bit cannot be read as 1.
0XXXXX1X	01h	This bit is set only when the correct payload size is received. This bit is set when the corresponding monotonic counter is uninitialized
0XXXX1XX	01h	This bit is set on Signature Mismatch, Counter Address out of range when correct payload size is received; or CmdType is out of range; or incorrect payload size is received.

Flow Diagram for Command:

Figure 3-2. Update HMAC Key



Increment Monotonic Counter

This command is used to increment the Monotonic counter by 1.

RPMC Increment Monotonic Counter OOB Command:

Byte #	7	6	5	4	3	2	1	0
0	eSPI Cycle Type: OOB Message=21h							
1	Tag[3:0]=0h				Length[11:8]=0h			
2	Length[7:0]=32h							
3	Dest Slave Addr[7:1]=07h							0

4	Command Code=0Fh				
5	Byte Count=2Fh				
6	Source Slave Address[7:1]= 08h				1
7	MCTP Reserved=0h			Header Version	
8	Destination Endpoint ID				
9	Source Endpoint ID				
10	SOM	EOM	Packet Seq #	TO	Message Tag
11	IC	Message Type=7Dh			
12	RPMC Device				
13	Opcode = 9Bh				
14	Cmd Type = 02h				
15	Counter Addr[7:0]				
16	Rsvd=00h				
17	Counter Data[31:24]				
...	...				

20	Counter Data[7:0]
21	Signature[255:248]
...	...
52	Signature[7:0]

Response:

Byte #	7	6	5	4	3	2	1	0
0	eSPI Cycle Type: OOB Message=21h							
1	Tag[3:0]=0h				Length[11:8]=0h			
2	Length[7:0]=0Ch							
3	Dest Slave Addr[7:1]=08h							0
4	Command Code=0Fh							
5	Byte Count=09h							
6	Source Slave Address[7:1]= 07h							1
7	MCTP Reserved=0h				Header Version			
8	Destination Endpoint ID							

9	Source Endpoint ID				
10	SOM	EOM	Packet Seq #	TO	Message Tag
11	IC	Message Type=7Dh			
12	RPMC Device				
13	Counter Addr[7:0]				
14	Extended Status[7:0]				

After the command is issued, the device ensures that the received transaction is error free. This includes checking following conditions:

- RPMC message payload size is correct. (Including OP1 = 40 bytes)
- Counter Address falls within the range of supported counters.
- The Monotonic Counter corresponding to the requested Counter Address was previously initialized.
- The HMAC Key Register corresponding to the requested Counter Address was previously initialized.
- The requested Signature matches the HMAC-SHA-256 based signature computed based on received input parameters.
    - HMAC Message[63:0] = (OpCode[7:0], CmdType[7:0]. CounterAddr[7:0]. Reserved[7:0], CounterData[31:0])
    - HMAC Key[255:0] = HMAC_Key_Register[Counter_Addreess][255:0]
- The received Counter Data matches the current value of the counter read from the device.

If the received transaction is error free the device successfully executes the command and posts “successful completion” extended status. The increment counter implementation must ensure that the counter increment operation is performed within the allowed command timeout.

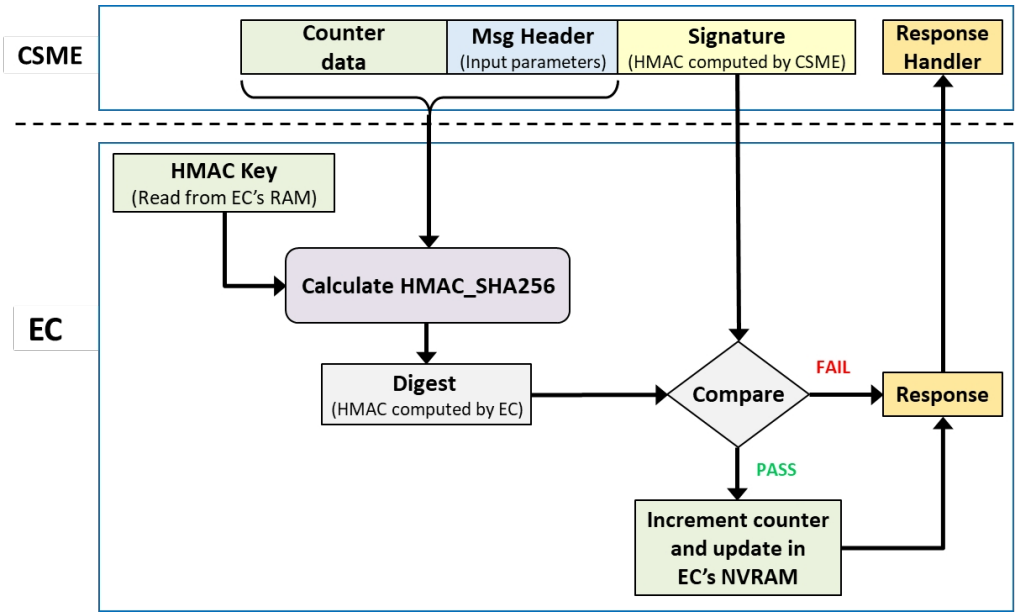
If the received transaction has errors, the device does not execute the transaction and posts the corresponding error in extended status.

Expected Extended Status [7:0] results:

Extended Status [7:0]	Applicable CmdType(s)	Description
10000000	02h	This status is set on successful completion (no errors) of OP1 command.
0XXXXXX1	02h	N/A. This bit cannot be read as 1.
0XXXX1XX	02h	This bit is set on Signature Mismatch, Counter Address out of range when correct payload size is received; or CmdType is out of range; or incorrect payload size is received.
0XXX1XXX	02h	This bit is set only when the correct payload size is received. This bit is set on HMAC Key Register (or Monotonic Counter) is uninitialized on previous OP1 command.
0XX1XXXX	02h	This bit is set only when the correct payload size is received. The bit is set when the received counter data filed does not match the actual counter value read from the device.

Flow Diagram for Command:

Figure 3-4. Increment Monotonic Counter



Request Monotonic Counter

This command is used to request the Monotonic counter value.

RPMC Request Monotonic Counter OOB Command:

Byte #	7	6	5	4	3	2	1	0
0	eSPI Cycle Type: OOB Message=21h							
1	Tag[3:0]=0h				Length[11:8]=0h			
2	Length[7:0]=3Ah							
3	Dest Slave Addr[7:1]=07h							0
4	Command Code=0Fh							
5	Byte Count=37h							

6	Source Slave Address[7:1]= 08h				1
7	MCTP Reserved=0h			Header Version	
8	Destination Endpoint ID				
9	Source Endpoint ID				
10	SOM	EOM	Packet Seq #	TO	Message Tag
11	IC	Message Type=7Dh			
12	RPMC Device				
13	Opcode = 9Bh				
14	Cmd Type = 03h				
15	Counter Addr[7:0]				
16	Rsvd=00h				
17	Tag[95:88]				
...	...				
28	Tag[7:0]				
29	Signature[255:248]				

...	...
60	Signature[7:0]

Response:

Byte #	7	6	5	4	3	2	1	0
0	eSPI Cycle Type: OOB Message=21h							
1	Tag[3:0]=0h				Length[11:8]=0h			
2	Length[7:0]=3Ch							
3	Dest Slave Addr[7:1]=08h							0
4	Command Code=0Fh							
5	Byte Count=39h							
6	Source Slave Address[7:1]= 07h							1
7	MCTP Reserved=0h				Header Version			
8	Destination Endpoint ID							
9	Source Endpoint ID							
10	SOM	EOM	Packet Seq #		TO	Message Tag		

11	IC	Message Type=7Dh
12	RPMC Device	
13	Counter Addr[7:0]	
14	Extended Status[7:0]	
15	Tag[95:88]	
...	...	
26	Tag[7:0]	
27	CounterReadData[31:24]	
...	...	
30	CounterReadData[7:0]	
31	Signature[255:248]	
...	...	
62	Signature[7:0]	

After the command is issued, the device ensures that the received transaction is error free. This includes checking following conditions:

- RPMC message payload size is correct. (Including OP1 = 48 bytes)
- Counter Address falls within the range of supported counters.
- The Monotonic Counter corresponding to the requested Counter Address was previously initialized.

- The HMAC Key Register corresponding to the requested Counter Address was previously initialized.
- The requested Signature matches the HMAC-SHA-256 based signature computed based on received input parameters.
    - HMAC Message[127:0] = (OpCode[7:0], CmdType[7:0]. CounterAddr[7:0]. Reserved[7:0], Tag[95:0]) we
    - HMAC Key[255:0] = HMAC_Key_Register[Counter_Address][255:0]

If the received transaction is error free device successfully executes the command and posts "successful completion" extended status. In response to this command, the device reads the monotonic counter addressed by counter address. It calculates HMAC-SHA-256 signatures the second time, based on following parameters:

- HMAC Message[127:0] = Tag [95:0], Counter_Data_Read[31:0]
- HMAC Key[255:0] = HMAC_Key_Register[Counter_Address][255:0]

It loads the outgoing response buffer[383:0] with the resulting signature:

- Outgoing response buffer[383: 288] = Tag[95:0].
- Outgoing response buffer[287:256] = Counter_Data[31:0]
- Outgoing response buffer[255:0] = Signature[255:0])

The tag, count value and signature are returned in the response to this command.

If the received transaction has errors, the device does not execute the transaction and posts the corresponding error in extended status.

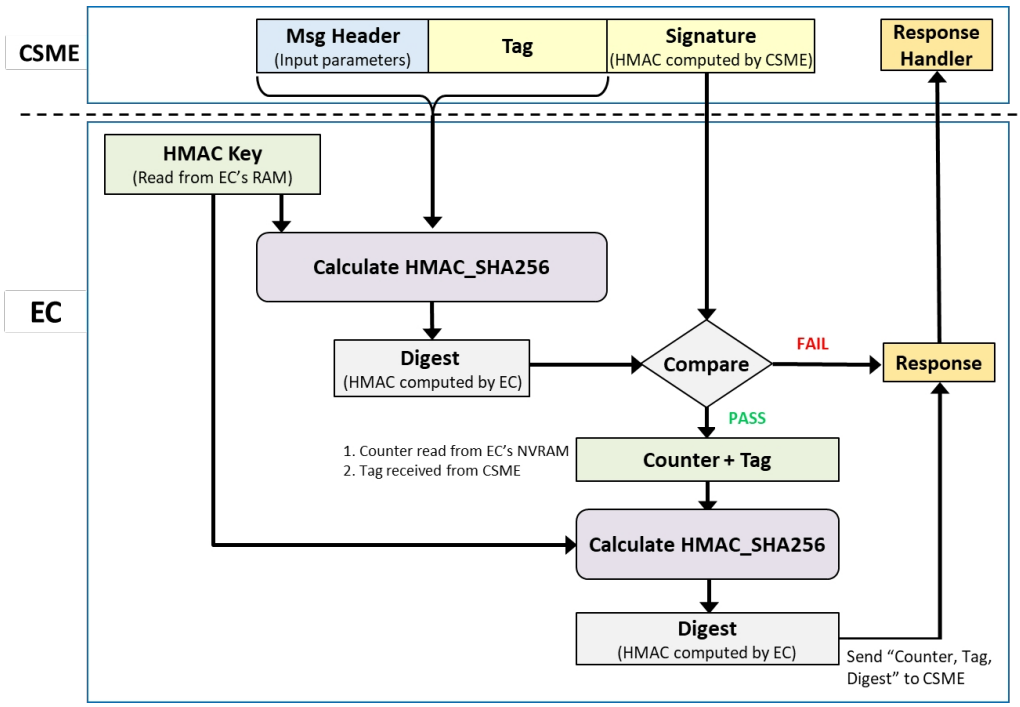
Expected Extended Status [7:0] results:

Extended Status [7:0]	Applicable CmdType(s)	Description
10000000	03h	This status is set on successful completion (no errors) of OP1 command.
0XXXXXX1	03h	N/A. This bit cannot be read as 1.
0XXXX1XX	03h	This bit is set on Signature Mismatch, Counter Address out of range when correct payload size is received; or CmdType is out of range; or incorrect payload size is received.

0XXX1XXX	03h	This bit is set only when the correct payload size is received. This bit is set on HMAC Key Register (or Root Key register or Monotonic Counter) is uninitialized on previous OP1 command.
0X1XXXXX	-	Fatal Error, e.g., program fail, no valid counter found after initialization

Flow Diagram for Command:

Figure 3-5. Request Monotonic Counter



Read RPMC Parameters

This command is used to read RPMC parameters values.

RPMC Parameter table is as defined below:

Bits	Description
31:8	Reserved

7:4	Document Version. Default=0.
3:0	Num_RPMC: Number of supported RPMC Devices (Note)

RPMC Parameters, RPMC Device n:

Bits	Description
31:28	Update_Rate: Rate of update = 5*(2**Update_Rate) seconds
27:26	RPMC Device
25	MC_Size: 0= Monotonic counter size is 32 bits, 1=Reserved. Default =0.
24	SHA_Size: 0=SHA-256, 1=SHA-384. Default =0.
23:16	Reserved
15:8	OP1: OP1 Opcode for RPMC Devices. Default= 9Bh
7:0	Num_Counter-1: Number of supported Monotonic Counters-1 (Note)

1. If RPMC is supported in FW in the EC device only, reports one device and number of supported counters. Num_RPMC is zero if RPMC is not supported in EC device.
2. If RPMC is supported by EC device and/or the EC device passing commands to SPI RPMC devices, reports the number of flash devices supported and the number of monotonic counters in each RPMC device.

Read RPMC Parameters Command:

Byte #	7	6	5	4	3	2	1	0

0	eSPI Cycle Type: OOB Message=21h				
1	Tag[3:0]=0h			Length[11:8]=0h	
2	Length[7:0]=0Bh				
3	Dest Slave Addr[7:1]=07h				0
4	Command Code=0Fh				
5	Byte Count=08h				
6	Source Slave Address[7:1]= 08h				1
7	MCTP Reserved=0h			Header Version	
8	Destination Endpoint ID				
9	Source Endpoint ID				
10	SOM	EOM	Packet Seq #	TO	Message Tag
11	IC	Message Type=7Dh			
12	RPMC Device= 00h				
13	Opcode = 9Fh				

Response:

Byte #	7	6	5	4	3	2	1	0
0	eSPI Cycle Type: OOB Message=21h							
1	Tag[3:0]=0h				Length[11:8]=0h			
2	Length[7:0]=12h							
3	Dest Slave Addr[7:1]=08h							0
4	Command Code=0Fh							
5	Byte Count=0Fh							
6	Source Slave Address[7:1]= 07h							1
7	MCTP Reserved=0h				Header Version			
8	Destination Endpoint ID							
9	Source Endpoint ID							
10	SOM	EOM	Packet Seq #		TO	Message Tag		
11	IC	Message Type=7Dh						
12	Extended Status[7:0]							
13	RPMC Parameter Table[31:24]							

...	...
16	RPMC Parameter Table[7:0]
17	RPMC Parameters, RPMC Device 0[31:24]
...	...
20	RPMC Parameters, RPMC Device 0[7:0]

Note: This command may return up to three additional DWs, depending on number of RPMC Devices.

If the received transaction has errors, the device does not execute the transaction and posts the corresponding error in extended status.

Expected Extended Status [7:0] results:

Extended Status [7:0]	Description
10000000	Successful completion.
0XXXXXX1	N/A. This bit cannot be read as 1, eRPMC not supported.
0XXXXX1X	This bit is only set when incorrect payload size is received.

Examples of RPMC Devices Supported

The following show examples of different RPMC devices (EC, SPI flash devices or both). Parameter table values and relevant MCTP command parameters for each example are shown.

EC is RPMC Device

RPMC Parameter Table:

Bits	Description	Value
31:8	Reserved	000000h
7:4	Document Version. Default=0.	0h
3:2	Reserved	00b
1:0	Num_RPMC: Number of supported RPMC Devices	01b

RPMC Parameters, RPMC Device 0:

Bits	Description	Value
31:28	Update_Rate	0h
27:26	RPMC Device	00b
25	MC_Size	0b
24	SHA_Size: 0=SHA-256, 1=SHA-384. Default =0.	0b
23:16	Reserved	00h
15:8	OP1: OP1 Opcode for RPMC Devices. Default= 9Bh	9Bh
7:0	Num_Counter-1: Number of supported Monotonic Counters-1	0-FFh

RPMC OOB Command Parameters, RPMC Device 0:

Parameter	Value
RPMC Device	00h
Counter Addr[7:0]	00h-FFh

Two RPMC SPI Flash Devices, Each with Four Monotonic Counters

RPMC Parameter Table:

Bits	Description	Value
31:8	Reserved	000000h
7:4	Document Version. Default=0.	0h
3:2	Reserved	00b
1:0	Num_RPMC: Number of supported RPMC Devices	02b

RPMC Parameters, RPMC Device 0:

Bits	Description	Value
31:28	Update_Rate	0h
27:26	RPMC Device	00b

25	MC_Size	0b
24	SHA_Size: 0=SHA-256, 1=SHA-384. Default =0.	0b
23:16	Reserved	00h
15:8	OP1: OP1 Opcode for RPMC Devices. Default= 9Bh	9Bh
7:0	Num_Counter-1: Number of supported Monotonic Counters-1	03h

RPMC OOB Command Parameters, RPMC Device 0:

Parameter	Value
RPMC Device	00h
Counter Addr[7:0]	00h-03h

RPMC Parameters, RPMC Device 1:

Bits	Description	Value
31:28	Update_Rate	0h
27:26	RPMC Device	01b
25	MC_Size	0b

24	SHA_Size: 0=SHA-256, 1=SHA-384. Default =0.	0b
23:16	Reserved	00h
15:8	OP1: OP1 Opcode for RPMC Devices. Default= 9Bh	9Bh
7:0	Num_Counter-1: Number of supported Monotonic Counters-1	03h

RPMC OOB Command Parameters, RPMC Device 1:

Parameter	Value
RPMC Device	01h
Counter Addr[7:0]	00h-03h

EC as RPMC Device and Two SPI RPMC Flash Devices

RPMC Parameter Table:

Bits	Description	Value
31:8	Reserved	000000h
7:4	Document Version. Default=0.	0h
3:2	Reserved	00b
1:0	Num_RPMC: Number of supported RPMC Devices	03b

RPMC Parameters, RPMC Device 0 (EC):

Bits	Description	Value
31:28	Update_Rate	0h
27:26	RPMC Device	00b
25	MC_Size	0b
24	SHA_Size: 0=SHA-256, 1=SHA-384. Default =0.	0b
23:16	Reserved	00h
15:8	OP1: OP1 Opcode for RPMC Devices. Default= 9Bh	9Bh
7:0	Num_Counter-1: Number of supported Monotonic Counters-1	FFh

RPMC OOB Command Parameters, RPMC Device 0 (EC):

Parameter	Value
RPMC Device	00h
Counter Addr[7:0]	00h-FFh

RPMC Parameters, RPMC Device 1:

Bits	Description	Value

31:28	Update_Rate	0h
27:26	RPMC Device	01b
25	MC_Size	0b
24	SHA_Size: 0=SHA-256, 1=SHA-384. Default =0.	0b
23:16	Reserved	00h
15:8	OP1: OP1 Opcode for RPMC Devices. Default= 9Bh	9Bh
7:0	Num_Counter-1: Number of supported Monotonic Counters-1	03h

RPMC OOB Command Parameters, RPMC Device 1:

Parameter	Value
RPMC Device	01h
Counter Addr[7:0]	00h-03h

RPMC Parameters, RPMC Device 2:

Bits	Description	Value
31:28	Update_Rate	0h

27:26	RPMC Device	02b
25	MC_Size	0b
24	SHA_Size: 0=SHA-256, 1=SHA-384. Default =0.	0b
23:16	Reserved	00h
15:8	OP1: OP1 Opcode for RPMC Devices. Default= 9Bh	9Bh
7:0	Num_Counter-1: Number of supported Monotonic Counters-1	03h

RPMC OOB Command Parameters, RPMC Device 2:

Parameter	Value
RPMC Device	02h
Counter Addr[7:0]	00h-03h

Reserved Command-type

If the device receives any of the reserved command-types, the device returns Error status in Extended Status Register. It asserts bit 2 to indicate that a reserved command-type was issued.

Expected Extended Status [7:0] results:

Extended Status [7:0]	Applicable CmdType(s)	Description

0XXXX1XX	04h-FFh	CmdType out of range

Monotonic Counters

eRPMC supports up to 256 monotonic counters. Monotonic counter size is 32 bits.

Note: Refer to Intel platform specific documentation for minimum number of counters required for a specific platform. 4 is the current minimum; it may increase in the future platforms.

Monotonic Counter addresses are as follows:

Monotonic Counter Number	Address
0	00h
1	01h
2	02h
3	03h
…	…
254	FEh
255	FFh