



LEGAL NOTICE:

© *Copyright 2008 to 2023 NVM Express®, Inc. ALL RIGHTS RESERVED.*

This Technical Proposal is proprietary to the NVM Express, Inc. (also referred to as "Company") and/or its successors and assigns.

NOTICE TO USERS WHO ARE NVM EXPRESS, INC. MEMBERS: Members of NVM Express, Inc. have the right to use and implement this Technical Proposal subject, however, to the Member's continued compliance with the Company's Intellectual Property Policy and Bylaws and the Member's Participation Agreement.

NOTICE TO NON-MEMBERS OF NVM EXPRESS, INC.: If you are not a Member of NVM Express, Inc. and you have obtained a copy of this document, you only have a right to review this document or make reference to or cite this document. Any such references or citations to this document must acknowledge NVM Express, Inc. copyright ownership of this document. The proper copyright citation or reference is as follows: "© 2008 to 2023 NVM Express, Inc. ALL RIGHTS RESERVED." When making any such citations or references to this document you are not permitted to revise, alter, modify, make any derivatives of, or otherwise amend the referenced portion of this document in any way without the prior express written permission of NVM Express, Inc. Nothing contained in this document shall be deemed as granting you any kind of license to implement or use this document or the specification described therein, or any of its contents, either expressly or impliedly, or to any intellectual property owned or controlled by NVM Express, Inc., including, without limitation, any trademarks of NVM Express, Inc.

LEGAL DISCLAIMER:

THIS DOCUMENT AND THE INFORMATION CONTAINED HEREIN IS PROVIDED ON AN "AS IS" BASIS. TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, NVM EXPRESS, INC. (ALONG WITH THE CONTRIBUTORS TO THIS DOCUMENT) HEREBY DISCLAIM ALL REPRESENTATIONS, WARRANTIES AND/OR COVENANTS, EITHER EXPRESS OR IMPLIED, STATUTORY OR AT COMMON LAW, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE, VALIDITY, AND/OR NONINFRINGEMENT.

All product names, trademarks, registered trademarks, and/or servicemarks may be claimed as the property of their respective owners.

The NVM Express® design mark is a registered trademark of NVM Express, Inc.

NVM Express Workgroup
c/o VTM, Inc.
3855 SW 153rd Drive
Beaverton, OR 97003
USA
info@nvmexpress.org

NVM Express® Technical Proposal (TP)

| | |
|----------------------------|--|
| Technical Proposal ID | 8019a - Authentication Verification Entity for DH-HMAC-CHAP |
| Revision Date | 2023.10.03 |
| Builds on Specification(s) | NVM Express Base Specification 2.0b, NVM Express TCP Transport Specification 1.0b NVM Express Management Interface Spec rev 1.2b |
| References | TP 8010a – NVMe-oF Centralized Discovery Controller TP 8016 – Subsystem Driven Zoning for Pull Registration |

Technical Proposal Author(s)

| Name | Company |
|-----------------|-------------------|
| Claudio DeSanti | Dell Technologies |
| | |
| | |
| | |
| | |
| | |
| | |
| | |

Technical Proposal Overview

Enable an NVMe-oF entity to delegate DH-HMAC-CHAP authentication verification to an Authentication Verification Entity.

Revision History

| Revision Date | Change Description |
|---------------|---|
| 2022.06.07 | Initial draft |
| 2022.06.21 | Second draft, incorporated multiple feedback |
| 2022.06.28 | Third draft, incorporated more feedback and updated for phase 2 exit |
| 2022.07.12 | Ready for phase 2 exit |
| 2022.07.14 | Added the TP name to Technical Proposal ID section |
| 2022.07.26 | Ready for phase 3 exit |
| 2022.07.28 | Clean version for phase 3 exit |
| 2022.09.01 | Incorporated feedback from Mike Allison, ready for preintegration |
| 2022.09.06 | Incorporated feedback from FMDS |
| 2022.11.16 | Integrated |
| 2022.11.21 | Integrated feedback from Mike Allison and Claudio Desanti |
| 2022.11.26 | Integrated feedback from Mike Allison and Claudio Desanti |
| 2022.12.14 | Integrated feedback from Mike Allison and David Black |
| 2023.08.01 | Start of TP 8019a. Fixed (in 8.13.TBD.2): <ul style="list-style-type: none">• PSK_{ea} computation• What to do when the DH-HMAC-CHAP secret representation does not specify a hash function• How to derive PSK_{ea} from a configured PSK |
| 2023.08.17 | Ready for 30-day member review |
| 2023.10.03 | Integrated |

Description for Changes Document for TP 8019

New Features:

- Authentication Verification Entity (AVE) for DH-HMAC-CHAP (optional feature)

Markup Conventions:

| | |
|-------------------------------|--|
| Black: | Unchanged (however, hot links are removed) |
| Red Strikethrough: | Deleted |
| Blue: | New |
| Orange: | Text from reference TPs |
| Blue Highlighted: | TBD values, anchors, and links to be inserted in new text. |
| <Green Bracketed>: | Notes to editor |

Description of Specification Changes for NVMe Base Specification 2.0b

1.8 References

Add the following references:

RFC 5869, H. Krawczyk, P. Eronen, "HMAC-based Extract-and-Expand Key Derivation Function (HKDF)", May 2010. Available from <https://www.ietf.org/rfc.html>.

RFC 6520, R. Seggelmann, M. Tuexen, M. Williams, "Transport Layer Security (TLS) and Datagram Transport Layer Security (DTLS) Heartbeat Extension", February 2012. Available from <https://www.ietf.org/rfc.html>.

3.1.2.1.2 Log Page Support

Modify Figure 24 (I/O Controller – Log Page Support) as shown below:

Figure 24: I/O Controller – Log Page Support

| Log Page Name | Log Page Support Requirements ¹ |
|--|--|
| ... | ... |
| Discovery | P |
| Host Discovery | P |
| AVE Discovery | P |
| Notes: | |
| 1. O/M/P definition: O = Optional, M = Mandatory, P = Prohibited | |

3.1.2.2.2 Log Page Support

Modify Figure 29 (Administrative Controller – Log Page Support) as shown below:

Figure 29: Administrative Controller – Log Page Support

| Log Page Name | Log Page Support Requirements ¹ |
|--|--|
| ... | ... |
| Discovery | P |
| Host Discovery | P |
| AVE Discovery | P |
| Notes: | |
| 1. O/M/P definition: O = Optional, M = Mandatory, P = Prohibited | |

3.1.2.3.3 Log Page Support

Modify Figure 33 (Discovery Controller – Log Page Support) as shown below:

Figure 33: Discovery Controller – Log Page Support

| Log Page Name | Log Page Support Requirements ¹ |
|----------------|--|
| ... | ... |
| Discovery | M |
| Host Discovery | O ² |
| AVE Discovery | O ² |

Figure 33: Discovery Controller – Log Page Support

| Log Page Name | Log Page Support Requirements ¹ |
|--|--|
| Notes: 1. O/M/P definition: O = Optional, M = Mandatory, P = Prohibited 2. Mandatory for CDCs and optional for Discovery controllers that are not a CDC. | |

5.2.1 Command Completion

Modify Figure 146 (Asynchronous Event Information – Notice) as shown below:

Figure 146: Asynchronous Event Information – Notice

| Value | Description |
|--------------------------|--|
| ... | ... |
| F0h | Discovery Log Page Change: A change has occurred to one or more of the Discovery log pages. The host or Discovery controller should submit a Get Log Page command to receive updated Discovery log pages. |
| F2h | AVE Discovery Log Page Change: A change has occurred to the AVE Discovery log page. The host or controller should submit a Get Log Page command to receive an updated AVE Discovery log page. |
| F4 F3h to FFh | Reserved for future NVMe over Fabrics Asynchronous Event Notifications |

5.16.1 Log Specific Information

Modify Figure 202 (Get Log Page – Log Page Identifiers) as shown below:

Figure 202: Get Log Page – Log Page Identifiers

| Log Identifier | Scope | Log Page Name | Reference Section |
|--------------------------|----------|---------------|-------------------|
| ... | ... | | |
| 19h to 6Fh | Reserved | | |
| 70h | | Discovery | 5.16.1.23 |
| 72h | | AVE Discovery | 5.16.1.NEW |
| 72 73h to 7Fh | Reserved | | |

Add a new section 5.16.1.NEW as shown below:

5.16.1.NEW AVE Discovery Log Page (Log Identifier 72h)

The format of the AVE Discovery Log Page is shown in Figure NEW.1.

Figure NEW.1: Get Log Page – AVE Discovery Log Page

| Bytes | Description |
|-----------------|---|
| Header | |
| 07:00 | Generation Counter (GENCTR): This field indicates the version of the discovery information, starting at a value of 0h. For each change in the AVE Discovery log page, this field shall be incremented by one. If the value of this field is FFFFFFFF_FFFFFFFFh, then the field shall be cleared to 0h when incremented (i.e., rolls over to 0h). |
| 15:08 | Number of Records (NUMREC): Indicates the number of records contained in the log page. |
| 17:16 | Record Format (RECFMT): This field indicates the format of the AVE Discovery log page. If a new format is defined, this value is incremented by one. The format of the record specified in this definition shall be 0h. |
| 19:18 | Reserved |
| 23:20 | Total AVE Discovery Log Page Length (TADLPL): This field indicates the length in bytes of the entire AVE Discovery log page. |
| 1023:24 | Reserved |
| Entries | |
| TEL + 1023:1024 | AVE Discovery Log Page Entry 0: This field contains the first AVE Discovery Log Page Entry as defined in Figure NEW.2. TEL is the size indicated in the Total Entry Length (TEL) field of the AVE Discovery Log Page Entry. |
| ... | ... |

Figure NEW.1: Get Log Page – AVE Discovery Log Page

| Bytes | Description |
|-------------------------|--|
| TADLPL - 1:TADLPL – TEL | AVE Discovery Log Page Entry NUMREC-1: This field contains the NUMREC-1 AVE Discovery Log Page Entry as defined in Figure NEW.2 (if present). TEL is the size indicated in the Total Entry Length (TEL) field of the AVE Discovery Log Page Entry and TADLPL is the size indicated in the Total AVE Log Page Length (TADLPL) field. |

The format of the AVE Discovery Log Page Entry is shown in [Figure NEW.2](#).

Figure NEW.2: Get Log Page – AVE Discovery Log Page Entry

| Bytes | Description |
|-----------------------|--|
| 03:00 | Total Entry Length (TEL): This field indicates the length in bytes of the AVE Discovery Log Page Entry. |
| 227:04 | AVE NQN (AVENQN): This field indicates the NQN (represented as a null-terminated string, NULL padded as necessary to the 224-byte maximum length) of the AVE. |
| 228 | Number of AVE Transport Records (N): This field indicates the number of subsequent AVE transport records. |
| 231:229 | Reserved |
| 251:232 | AVE Transport Record #1, if any |
| 271:252 | AVE Transport Record #2, if any |
| ... | ... |
| (N*20)+231:(N*20)+212 | AVE Transport Record #N, if any |

The format of the AVE Transport Record is shown in [Figure NEW.3](#).

Figure NEW.3: AVE Transport Record

| Bytes | Description | | | | | | | | | | |
|-------|--|-------|------------|-------|------|-------|------------|-------|------|-------|----------|
| 00 | AVE Address Family (AVEADRFAM): This field identifies the IP address family. This field shall be set to one of the following values: <ul style="list-style-type: none"> 1h: IPv4 address family; or 2h: IPv6 address family. | | | | | | | | | | |
| 01 | Reserved | | | | | | | | | | |
| 03:02 | AVE Transport Service Identifier (AVETRSCID): This field identifies the TCP port. | | | | | | | | | | |
| 19:04 | AVE Transport Address (AVETRADDR): This field identifies the IP address. An IPv6 address is encoded in binary as follows: <table border="1"> <tr> <th>Bytes</th><th>Definition</th></tr> <tr> <td>15:00</td><td>Used</td></tr> </table> An IPv4 address is encoded in binary as follows: <table border="1"> <tr> <th>Bytes</th><th>Definition</th></tr> <tr> <td>03:00</td><td>Used</td></tr> <tr> <td>15:04</td><td>Reserved</td></tr> </table> | Bytes | Definition | 15:00 | Used | Bytes | Definition | 03:00 | Used | 15:04 | Reserved |
| Bytes | Definition | | | | | | | | | | |
| 15:00 | Used | | | | | | | | | | |
| Bytes | Definition | | | | | | | | | | |
| 03:00 | Used | | | | | | | | | | |
| 15:04 | Reserved | | | | | | | | | | |

5.27.1.8 Asynchronous Event Configuration (Feature Identifier 0Bh)

Modify Figure 326 (Asynchronous Event Configuration – Command Dword 11) as shown below:

Figure 326: Asynchronous Event Configuration – Command Dword 11

| Bits | Description |
|-------------------|--|
| 31 | Discovery Log Page Change Notification: This bit indicates that the Discovery controller reports Discovery Log Page Change Notifications. If set to '1', the Discovery controller shall send a notification if Discovery log page changes occur. |
| ... | |
| 29 | AVE Discovery Log Page Change Notification: This bit indicates that the Discovery controller reports AVE Discovery Log Page Change Notifications. If set to '1', then the Discovery controller shall send a notification if AVE Discovery log page changes occur. |
| 30 :28 | Reserved |
| ... | ... |

Modify a portion of section 8.13.5.1 (Protocol Operations) as shown below:

To authenticate another entity, an entity is required to either:

- know the key associated with the entity to be authenticated; or
- rely on a third party that knows the key to verify the authentication (refer to section 8.13.TBD).

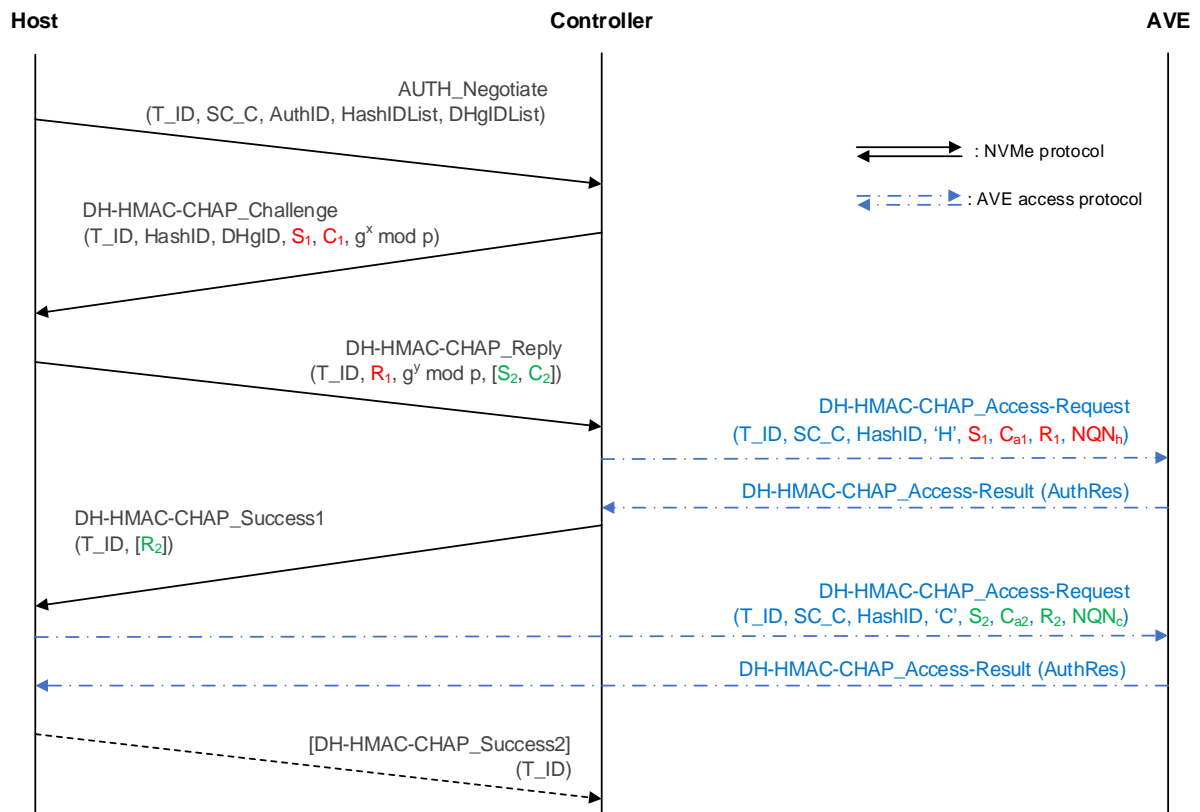
Add a new section 8.13.TBD as shown below:

8.13.TBD DH-HMAC-CHAP Authentication Verification Entity

8.13.TBD.1 Overview

A DH-HMAC-CHAP Authentication Verification Entity (AVE) is a service that performs the DH-HMAC-CHAP authentication verification function on behalf of an NVMe entity (i.e., a host or a controller). An example of a DH-HMAC-CHAP authentication transaction with AVE is shown in Figure NEW.4, using the notation shown in Figure 444.

Figure NEW.4: Example of DH-HMAC-CHAP authentication transaction with AVE



As shown in **Figure NEW.4**, a controller using the AVE service delegates to the AVE the verification of the response R_1 received from the host by passing the relevant DH-HMAC-CHAP authentication transaction parameters to the AVE through a DH-HMAC-CHAP_Access-Request message. A host using the AVE service delegates to the AVE the verification of the response R_2 received from the controller by passing the relevant DH-HMAC-CHAP authentication transaction parameters to the AVE through a DH-HMAC-CHAP_Access-Request message. In both cases, the AVE replies with a DH-HMAC-CHAP_Access-Result message containing the result of the authentication verification (refer to section **8.13.TBD.3**).

Use of the AVE service by an NVMe entity is optional and is determined by configuration of the NVMe entity. If an NVMe entity uses the AVE, then provisioning of DH-HMAC-CHAP information on that entity is reduced to only that entity's DH-HMAC-CHAP secret (refer to section **8.13.5.8**) and the parameters (refer to section **8.13.TBD.2**) for accessing the AVE (i.e., no DH-HMAC-CHAP keys are required to be provisioned for verification of responses received from other NVMe entities).

An AVE is required to maintain the following information for each NVMe entity:

- the NQN of that entity (i.e., NQN_e),
- the DH-HMAC-CHAP key associated with that entity (i.e., K_e), and
- the PSK shared between that entity and the AVE (i.e., PSK_{ea}).

K_e is used to perform the authentication verification function (refer to section **8.13.TBD.3**) and PSK_{ea} is used to establish a secure connection with the AVE (refer to section **8.13.TBD.2**). An AVE shall support all hash functions defined for DH-HMAC-CHAP (refer to section **8.13.5.2**).

To facilitate dynamic discovery of the transport addresses of an AVE through a Discovery Controller (refer to section **8.13.TBD.4**) and to simplify establishing a secure connection to an AVE (refer to section **8.13.TBD.2**), an AVE is identified at by an NQN (NQN_{AVE}).

8.13.TBD.2 AVE Connections

An NVMe entity (i.e., a host or a controller) connection with a DH-HMAC-CHAP AVE shall use TLS version 1.3 (refer to RFC 8446) with pre-shared key (PSK) authentication, as specified for NVMe/TCP (refer to the NVM Express TCP Transport Specification).

In order to establish a TLS connection with an AVE, an NVMe entity requires a PSK shared between that entity and the AVE (i.e., PSK_{ea}) for authentication of the TLS connection. PSK_{ea} shall be either derived from the DH-HMAC-CHAP secret, if the DH-HMAC-CHAP secret representation specifies a hash function (refer to section 8.13.5.8), or provisioned on the NVMe entity through a configured PSK, as specified for NVMe/TCP.

Derivation of PSK_{ea} from a DH-HMAC-CHAP secret shall use the HKDF-Extract and HKDF-Expand-Label operations (refer to RFC 5869 and RFC 8446) in the following order:

1. $PRK = \text{HKDF-Extract}(0, \text{DH-HMAC-CHAP secret})$; and
2. $PSK_{ea} = \text{HKDF-Expand-Label}(PRK, \text{"A-V-Entity"} \parallel NQN_{AVE}, NQN_e, \text{Length}(\text{DH-HMAC-CHAP secret}))$,

where NQN_{AVE} is the NQN of the AVE and NQN_e is the NQN of the NVMe entity. The hash function used with HKDF shall be the one specified in the DH-HMAC-CHAP secret representation (refer to section 8.13.5.8). This transform requires that the NVMe entity knows NQN_{AVE} .

Derivation of PSK_{ea} from a DH-HMAC-CHAP secret is not possible if the DH-HMAC-CHAP secret representation does not specify a hash function. In this case PSK_{ea} shall be provisioned on the NVMe entity through a configured PSK, as specified for NVMe/TCP, and that configured PSK should be different than the DH-HMAC-CHAP secret for that entity.

Derivation of PSK_{ea} from a configured PSK shall use the HKDF-Extract and HKDF-Expand-Label operations in the following order:

1. $PRK = \text{HKDF-Extract}(0, \text{Configured PSK})$; and
2. $PSK_{ea} = \text{HKDF-Expand-Label}(PRK, \text{"A-V-Entity"} \parallel NQN_{AVE}, NQN_e, \text{Length}(\text{Configured PSK}))$,

where NQN_{AVE} is the NQN of the AVE and NQN_e is the NQN of the NVMe entity. The hash function used with HKDF shall be the one specified in the PSK interchange format (refer to the NVM Express TCP Transport Specification). If no hash function is specified in the PSK interchange format, then the configured PSK shall be used as PSK_{ea} . This transform requires that the NVMe entity knows NQN_{AVE} .

The TLS connection with the AVE shall be performed as specified in the TLS PSK and PSK Identity Derivation section of the NVM Express TCP Transport Specification, with PSK_{ea} used as the Retained PSK, NQN_e used as the host NQN, and NQN_{AVE} used as the controller NQN. Mandatory and recommended cipher suites for this TLS connection are specified in the Mandatory and Recommended Cipher Suites section of the NVM Express TCP Transport Specification. This TLS connection may be used for multiple authentication verifications. An NVMe entity may terminate this TLS connection and re-establish it as required. An AVE may terminate this TLS connection after some period of inactivity (e.g., 10 minutes). An NVMe entity may avoid termination of this TLS connection by using the TLS heartbeat extension (refer to RFC 6520).

8.13.TBD.3 AVE Access Protocol

Communication with a DH-HMAC-CHAP AVE uses two PDUs, DH-HMAC-CHAP_Access-Request and DH-HMAC-CHAP_Access-Result, that are sent directly over the TLS connection with the AVE (refer to section 8.13.TBD.2). The format of the DH-HMAC-CHAP_Access-Request PDU is shown in Figure NEW.5.

Figure NEW.5: DH-HMAC-CHAP_Access-Request PDU format

| Bytes | Description |
|-------|--|
| 00 | PDU-Type: AEh for DH-HMAC-CHAP_Access-Request |
| 01 | FLAGS: Reserved |
| 02 | Header Length (HLEN): Fixed length of 8 bytes (08h) |

| Bytes | Description |
|-------------------------|--|
| 03 | PDU Data Offset (PDO): Reserved |
| 07:04 | PDU Length (PLEN): total length of the PDU in bytes |
| 15:08 | ID: 64-bit identifier used to match Access-Request and Access-Result PDUs |
| 16 | Hash Length (HL): Length in bytes of the selected hash function |
| 17 | HashID: Identifier of selected hash function |
| 19:18 | T_ID: 16-bit authentication transaction identifier |
| 20 | SC_C: Secure Channel concatenation indication |
| 21 | Responder's Role: 'H' if host, 'C' if controller |
| 22 | NQNRlen: Length of the responder's NQN |
| 23 | Reserved |
| 27:24 | Sequence Number: Sequence number S |
| 27+HL:28 | Augmented Challenge Value: Challenge C _a |
| 27+2*HL:28+HL | Response Value: Response R |
| NQNRlen+27+2*HL:28+2*HL | NQNR: Responder's NQN |

The DH-HMAC-CHAP_Access-Request PDU contains the parameters exchanged by the host and the controller during a DH-HMAC-CHAP authentication transaction. The responder is the entity that replied to a DH-HMAC-CHAP challenge sent by an authenticator.

Referring to [Figure NEW.4](#), when the controller transmits the DH-HMAC-CHAP_Access-Request PDU, the parameters are instantiated as follows:

- Responder's Role: 'H'
- Sequence Number: S₁
- Augmented Challenge Value: C_{a1}
- Response Value: R₁
- NQNR: NQN_h
- HashID, T_ID, SC_C: the correspondent DH-HMAC-CHAP parameters

When the host transmits the DH-HMAC-CHAP_Access-Request PDU, the parameters are instantiated as follows:

- Responder's Role: 'C'
- Sequence Number: S₂
- Augmented Challenge Value: C_{a2}
- Response Value: R₂
- NQNR: NQN_c
- HashID, T_ID, SC_C: the correspondent DH-HMAC-CHAP parameters

Upon receiving a DH-HMAC-CHAP_Access-Request PDU, the AVE shall perform the following steps in order:

1. Lookup the DH-HMAC-CHAP key of the responder (i.e., K_r) from NQNR;
2. If the Responder's Role is 'H', compute the expected response R' as:

$$R' = \text{HMAC}(K_r, C_a \parallel S \parallel T_ID \parallel SC_C \parallel \text{"HostHost"} \parallel NQN_r \parallel 00h \parallel NQN_a)$$
where NQN_a is the NQN of the authenticator;

3. If the Responder's Role is 'C', compute the expected response R' as:
 $R' = \text{HMAC}(K_r, C_a \parallel S \parallel T_ID \parallel SC_C \parallel \text{"Controller"} \parallel NQN_r \parallel 00h \parallel NQN_a)$
 where NQN_a is the NQN of the authenticator; and
4. Compare the expected response R' with the response value R received in the DH-HMAC-CHAP_Access-Request PDU. If $R' = R$ then the authentication is successful; if $R' \neq R$ then the authentication has failed.

The NQN of the authenticator (i.e., NQN_a) is retrieved from the TLS identity associated to the TLS connection with the AVE (refer to section 8.13.TBD.2).

The result of an authentication verification is returned to the NVMe entity in a DH-HMAC-CHAP_Access-Result PDU. The format of the DH-HMAC-CHAP_Access-Result PDU is shown in Figure NEW.6.

Figure NEW.6: DH-HMAC-CHAP_Access-Result PDU format

| Bytes | Description | | | | | | | | | | |
|------------------|--|-------|------------|-----|--|-----|------------------------------------|------------------|---------------------------------|------------------|----------|
| 00 | PDU-Type: AFh for DH-HMAC-CHAP_Access-Result | | | | | | | | | | |
| 01 | FLAGS: Reserved | | | | | | | | | | |
| 02 | Header Length (HLEN): Fixed length of 8 bytes (08h) | | | | | | | | | | |
| 03 | PDU Data Offset (PDO): Reserved | | | | | | | | | | |
| 07:04 | PDU Length (PLEN): Fixed length of 20 bytes (14h) | | | | | | | | | | |
| 15:08 | ID: 64-bit identifier used to match Access-Request and Access-Result PDUs | | | | | | | | | | |
| 16 | Authentication verification result (AuthRes): <table> <tr> <th>Value</th><th>Definition</th></tr> <tr> <td>01h</td><td>Authentication Verification Successful</td></tr> <tr> <td>02h</td><td>Authentication Verification Failed</td></tr> <tr> <td>All other values</td><td>Reserved</td></tr> </table> | Value | Definition | 01h | Authentication Verification Successful | 02h | Authentication Verification Failed | All other values | Reserved | | |
| Value | Definition | | | | | | | | | | |
| 01h | Authentication Verification Successful | | | | | | | | | | |
| 02h | Authentication Verification Failed | | | | | | | | | | |
| All other values | Reserved | | | | | | | | | | |
| 17 | Reason Code: Additional explanation when authentication verification failed <table> <tr> <th>Value</th><th>Definition</th></tr> <tr> <td>00h</td><td>No additional explanation</td></tr> <tr> <td>01h</td><td>Authentication failure</td></tr> <tr> <td>02h</td><td>Selected hash function unusable</td></tr> <tr> <td>All other values</td><td>Reserved</td></tr> </table> | Value | Definition | 00h | No additional explanation | 01h | Authentication failure | 02h | Selected hash function unusable | All other values | Reserved |
| Value | Definition | | | | | | | | | | |
| 00h | No additional explanation | | | | | | | | | | |
| 01h | Authentication failure | | | | | | | | | | |
| 02h | Selected hash function unusable | | | | | | | | | | |
| All other values | Reserved | | | | | | | | | | |
| 19:18 | Reserved | | | | | | | | | | |

8.13.TBD.4 AVE Discovery

The AVE transport addresses may be configured on an NVMe entity or may be discovered by interacting with a Discovery Controller (e.g., a CDC). An NVMe entity should randomly select any of the discovered AVE transport addresses to connect to the AVE. The AVE Discovery log page is defined to facilitate this discovery (refer to section 5.16.1.NEW).

Description of Specification Changes for NVM Express Management Interface Specification 1.2b

Modify Figure 122 (Management Endpoint - Log Page Support) as shown below:

Figure 122: Management Endpoint - Log Page Support

| Log Page Name ³ | Log Identifier | Requirements ¹ | |
|----------------------------|----------------|---------------------------|----------------|
| | | NVMe Storage Device | NVMe Enclosure |
| ... | | | |
| Discovery | 70h | O | O |
| AVE Discovery | 72h | O | O |
| ... | | | |