

Internet Draft
Document: [draft-black-ips-iscsi-dhchap-01.txt](#)
Expires: October 2002

D. Black
EMC
April 2002

DH-CHAP: Diffie-Hellman Enhanced CHAP for iSCSI

Status of this Memo

This document is an Internet-Draft and is in full conformance with all provisions of [Section 10 of RFC2026](#).

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at
<http://www.ietf.org/ietf/lid-abstracts.txt>

The list of Internet-Draft Shadow Directories can be accessed at
<http://www.ietf.org/shadow.html>.

Abstract

This draft describes an authentication mechanism based on enhancing CHAP [[RFC 1994](#)] with a Diffie-Hellman Exchange (see Section 22.1 of [[Schneier](#)]) in order to prevent a passive eavesdropper from acquiring sufficient information to perform an off-line dictionary attack on the CHAP secret. The use of this authentication mechanism with iSCSI [[iSCSI](#)] is discussed, along with a brief comparison to the existing CHAP and SRP authentication mechanisms in iSCSI.

Caution should be exercised in drawing inferences from the fact that the author of this draft is one of the chairs of the IP Storage Working Group. This draft is an individual submission that the IP Storage Working Group is free to adopt, modify, reject, fold, spindle, and/or mutilate as it sees fit.

DH-CHAP: Diffie-Hellman Enhanced CHAP for iSCSI April 2002

Table of Contents

1.	Overview.....	2
2.	Conventions used in this document.....	3
3.	Basic CHAP Protocol.....	3
4.	Basic Anonymous Diffie-Hellman Key Exchange.....	4
5.	DH-CHAP, Diffie-Hellman Enhanced CHAP.....	6
	5.1 Bi-directional DH-CHAP.....	7
	5.2 DH-CHAP additions to DH and CHAP.....	7
	5.3 iSCSI text keys and values.....	8
6.	Brief Security Comparison of DH-CHAP with CHAP and SRP.....	8
	6.1 Passive Eavesdropping.....	9
	6.2 Use of Secret Information to Verify Authentication.....	9
	6.3 Impersonation and Man-in-the-Middle Attacks.....	9
	6.4 Reflection Attacks.....	10
	6.5 IPsec and DH-CHAP.....	11
	6.6 Passwords vs. Machine-generated keys.....	11
7.	External Authentication Server Considerations.....	12
	7.1 DH-CHAP and EAP.....	12
	7.1.1 Successful Authentication.....	13
	7.2.2 Unsuccessful Authentication.....	13
8.	Security Considerations.....	13
9.	Open Issues.....	13
10.	Change Log.....	14
	10.1 -00 to -01.....	14
	References.....	14

[1.](#) Overview

DH-CHAP is a new combination of an unauthenticated Diffie-Hellman (DH) key exchange (see Section 22.1 of [[Schneier](#)]) with the existing CHAP algorithm [[RFC 1994](#)]. CHAP is vulnerable to an offline dictionary attack in that an eavesdropper who observes the CHAP challenge and response obtains information sufficient to test an unlimited number of candidates for the CHAP secret off-line. This offline attack succeeds more often than random chance would predict because CHAP secrets are often human-selected passwords, and humans aren't very good at selecting random passwords. For example, they often use words that can be found in a dictionary. DH-CHAP strengthens CHAP in a fashion that requires an attacker to perform an online attack in order to capture the information required to mount an off-line dictionary attack. The three primary design goals of DH-CHAP are:

- (1) Prevent a passive dictionary attack on CHAP via use of a DH exchange. An active attacker (e.g., impersonator or man-in-the-middle) can still gain sufficient information to mount an off-line dictionary attack.

DH-CHAP: Diffie-Hellman Enhanced CHAP for iSCSI April 2002

- (2) Stay as close to CHAP as possible. The ability to use existing RADIUS servers to verify authentication of DH-CHAP is desirable, although there are security considerations involved in doing so.
- (3) Invent as little as possible. Every innovation in this sort of security protocol is an opportunity for subtle errors that can have major security consequences.

The basic idea behind DH-CHAP is to augment the CHAP challenge with the key that results from the Diffie-Hellman exchange so that the CHAP response depends on both of them. [Section 3](#) describes the basic CHAP algorithm, [Section 4](#) describes a basic unauthenticated Diffie-Hellman key exchange, and [Section 5](#) specifies how DH-CHAP combines them.

[2.](#) Conventions used in this document

I - Initiator
R - Responder
| - Concatenation operation
H[] - One-way hash function

CHAP requires the MD5 one-way hash function for interoperability, and there is existing equipment that only supports MD5. SHA-1 is cryptographically preferable for new protocols. In order to use SHA-1, it will need to be registered in the IANA PPP Authentication Algorithms registry (<http://www.iana.org/assignments/ppp-numbers>, PPP AUTHENTICATION ALGORITHMS section).

Additional notation for each protocol is introduced in the section describing that protocol.

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [[RFC 2119](#)].

[3.](#) Basic CHAP Protocol

CHAP is specified in [[RFC 1994](#)]. This is an authentication algorithm where the Initiator proves to the Responder that the Initiator knows a secret (e.g., password) that is also known to the responder without passing the secret in the clear to the Responder. The following notation is used here:

C_k - CHAP secret, shared by I and R
C_i - CHAP identifier, 1 octet
C_c - CHAP challenge, 16 octets when MD5 is used, MUST be generated from a source of real randomness
C_r - CHAP response, 16 octets when MD5 is used, computed from the above three

DH-CHAP: Diffie-Hellman Enhanced CHAP for iSCSI April 2002

The CHAP secret, C_k is often a human-selected password, and hence is the source of CHAP's vulnerability to dictionary attacks. The CHAP protocol consists of the following steps:

- (1) The Initiator requests service from the Responder and sends a list of acceptable CHAP hash algorithms:

I ---- algorithm list ----> R

- (2) The responder selects a hash algorithm, generates a random challenge, C_c, chooses an identifier, C_i and replies:

I <---- algorithm, C_i, C_c ---- R

- (3) The Initiator computes the response, $C_r = H[C_i|C_k|C_c]$ and replies:

I ---- C_r ----> R

- (4) The responder independently computes $C_{r'} = H[C_i|C_k|C_c]$. Authentication succeeds if $C_r == C_{r'}$. I and R both know C_i and C_c, hence this equality demonstrates to R that I knows C_k.

The computation of C_r' and the comparison may be outsourced to a RADIUS server or other external server capable of verifying CHAP authentication. R sends the username, C_i, C_c, and C_r to the server and asks if the response is correct. This allows the secret, C_k to be stored in the external server rather than the Responder.

[RFC 1994](#) specifies that the CHAP Identifier (C_i) is returned at step (3). iSCSI prohibits this, and C_i is omitted from step 3 in the above description to match the iSCSI specification.

4. Basic Anonymous Diffie-Hellman Key Exchange

A description of the anonymous Diffie-Hellman key exchange protocol can be found in Section 22.1 of [[Schneier](#)]. This is a key exchange protocol where the Initiator and Responder agree on a secret key whose contents are computationally intractable to determine from the messages they exchange. For brevity, the acronym DH will be used to refer to this protocol. The following notation is used here:

- x - first random number
- y - second random number
- n - field prime
- g - generator

Black

Expires - October 2002

[Page 4]

DH-CHAP: Diffie-Hellman Enhanced CHAP for iSCSI April 2002

k - generated key

The random numbers MUST be generated from sources of real randomness. DH is based on modular (mod n) arithmetic in a finite field, which is often referred to as a group. The security properties of DH depend strongly on the size of the field and numerical properties of both the field prime and generator; see [[Schneier](#)] for further discussion. Current practice is to specify fixed values that are known to be secure for a small selection of different field sizes.

The protocol is shown here with Bob as the Initiator and Alice as the Responder in order to match the structure in which it is used by DH-CHAP.

(1) The Initiator requests service from the Responder:

I ---- request for service ----> R

(2) The Responder generates the first random number, x, computes $g^x \text{ mod } n$ and replies:

I <---- $g^x \text{ mod } n$ ----> R

- (3) The Initiator generates the second random number, y , computes $g^y \bmod n$ and replies:

I ---- $g^y \bmod n$ ----> R

- (4) Both parties can now calculate the resulting key:
- The Initiator generated y and received $g^x \bmod n$, and so calculates $k = (g^x \bmod n)^y \bmod n = g^{xy} \bmod n$.
 - The Responder generated x and received $g^y \bmod n$, and so calculates $k' = (g^y \bmod n)^x \bmod n = g^{xy} \bmod n$.

A passive eavesdropper will find it computationally daunting to calculate $g^{xy} \bmod n$ from the $g^x \bmod n$ and $g^y \bmod n$ that she can observe. In essence this requires calculating x from $g^x \bmod n$ or y from $g^y \bmod n$, and calculation of these sort of discrete logarithms in a finite field is very difficult.

The key exchange ends here, but two more things usually happen when the exchanged key is used in practice:

- (1) k and k' are employed in a fashion where something goes seriously wrong if $k \neq k'$.
- (2) k and k' tend to be large numbers (e.g., kilobits) with low relative entropy, in that they possess randomness equivalent to true random numbers with far fewer bits. A one-way hash is a

DH-CHAP: Diffie-Hellman Enhanced CHAP for iSCSI April 2002

useful tool to produce smaller numbers with higher entropy (e.g., that make good session keys).

[5.](#) DH-CHAP, Diffie-Hellman Enhanced CHAP

DH-CHAP is specified here. This is an authentication algorithm where the Initiator proves to the Responder that the Initiator knows a secret (e.g., password) that is also known to the responder without passing the secret in the clear to the Responder, or passing information between the parties that is sufficient for a passive eavesdropper to mount a dictionary attack against the secret.

The basic idea is to augment the CHAP challenge, C_c , with the key, k , that results from the Diffie-Hellman exchange so that the CHAP response, C_r , depends on both of them. The following additional notation is used here:

C_ca - the augmented challenge from which the DH-CHAP response is generated.

The DH-CHAP protocol proceeds as follows:

- (1) The Initiator requests service from the Responder and sends a list of acceptable CHAP hash algorithms:

I ---- algorithm list ----> R

- (2) The Responder selects a hash algorithm, generates the initial CHAP challenge, C_c, and the first random number x. The Responder then computes $g^x \bmod n$, chooses a CHAP identifier, C_i, and replies:

I <---- algorithm, $g^x \bmod n$, C_i, C_c ---- R

- (3) The Initiator generates the second random number, y, computes:
 $g^y \bmod n$
 $C_{ca} = H[C_c | (g^x \bmod n)^y \bmod n] = H[C_c | g^{xy} \bmod n]$
 $C_r = H[C_i | C_k | C_{ca}]$

and replies:

I ---- C_r, $g^y \bmod n$ ----> R

- (4) The Responder now computes:
 $C_{ca}' = H[C_c | (g^y \bmod n)^x \bmod n] = H[C_c | g^{xy} \bmod n]$
 $C_r' = H[C_i | C_k | C_{ca}']$

Authentication succeeds if $C_r == C_r'$.

The computation of C_r' and the comparison could be outsourced to a RADIUS server or other external server capable of verifying CHAP authentication. R sends the username, C_i, C_ca', and C_r to the external server and asks if the response is correct. This allows the secret, C_k, to be stored in the external server rather than the Responder. A one-way hash is used to generate C_ca in order to make this possible (if C_ca were just $C_c | g^{xy} \bmod n$ without the hash, it would be much larger than these servers expect, and using a subset of $g^{xy} \bmod n$'s bits would sacrifice much of the strength of the key exchange). Outsourcing this verification has security

implications that are discussed in [Section 7](#).

[5.1](#) Bi-directional DH-CHAP

Section 10.5 of [[iSCSI](#)] describes the use of two overlapped instances of CHAP to accomplish bi-directional authentication. (NOTE: This is not "mutual" authentication as the authentications in each direction are not cryptographically linked.) Each instance requires a hash on each side, so two instances would require two hash calculations (and in each case, one could be outsourced to a RADIUS server).

DH-CHAP adds two exponentiations and an additional hash calculation to each side. A simpleminded application of DH-CHAP to the two CHAP instances in the previous paragraph would result in a total of four exponentiations on each side to perform two Diffie-Hellman exchanges. Exponentiations are sufficiently expensive calculations to merit optimizing, and the optimization is straightforward. The $g^{xy} \bmod n$ generated from a single DH exchange can be used for both DH-CHAP exchanges because in each case, $g^{xy} \bmod n$ is concatenated to a different random challenge (C_c) before applying the hash that calculates C_{ca} and C_{ca}' . DH-CHAP may qualify as mutual authentication, because both participants "sign" the same key derived from a DH exchange, but mutual authentication was not among DH-CHAP's design goals.

[5.2](#) DH-CHAP additions to DH and CHAP

The only addition that DH-CHAP makes to the base DH and CHAP protocols is the calculation of C_{ca} and C_{ca}' as a hash of the DH key appended to the transmitted CHAP challenge, specifically:

$$\begin{aligned}C_{ca} &= H[C_c | (g^x \bmod n)^y \bmod n] = H[C_c | g^{xy} \bmod n] \\C_{ca}' &= H[C_c | (g^y \bmod n)^x \bmod n] = H[C_c | g^{xy} \bmod n]\end{aligned}$$

This was added in this form to ensure that DH-CHAP can produce challenges and responses in the forms expected by existing servers that verify CHAP responses (e.g., RADIUS servers). All other

computations specified above and quantities sent in DH-CHAP messages are present in the original DH or CHAP algorithms.

DH-CHAP retains the challenge, C_c , for similarity to CHAP, and

because it avoids having to invent a way to use the same DH exchange for both directions of bi-directional authentication. The existence of C_c poses a slight increase of difficulty to an attacker who has somehow defeated the DH exchange, but in practice defeating the DH exchange is primary obstacle facing such an attacker.

[5.3](#) iSCSI text keys and values

DH-CHAP is a different algorithm from CHAP. To negotiate DH-CHAP for iSCSI authentication, a "DHCHAP" value needs to be added to the set of values defined for the AuthMethod key. The following keys need to be defined for DHCHAP's usage:

- DHCHAP_A - DH-CHAP algorithm list or algorithm, similar to CHAP_A
- DHCHAP_I - DH-CHAP identifier, similar to CHAP_I
- DHCHAP_C - DH-CHAP challenge, similar to CHAP_C
- DHCHAP_N - DH-CHAP name, similar to CHAP_N
- DHCHAP_R - DH-CHAP response, similar to CHAP_R
- DHCHAP_GX - $g^x \bmod n$ sent from Responder to Initiator
- DHCHAP_GY - $g^y \bmod n$ sent from Initiator to Responder

When the DH-CHAP authentication method is the result of AuthMethod negotiation, the CHAP keys (CHAP_*) MUST NOT be used. When the CHAP authentication method is the result of AuthMethod negotiation, the DH-CHAP keys (DHCHAP_*) MUST NOT be used.

In addition, the DH_CHAP field prime and generator MUST be communicated from Initiator to Responder at step (1) (same approach as used for SRP with iSCSI):

- DHCHAP_DHN - DH field prime, n in Sections [4](#) and [5](#)
- DHCHAP_DHG - DH generator, g in Sections [4](#) and [5](#)

As is the case for SRP authentication in iSCSI, these are announced values that cannot be negotiated; the Responder MUST either accept them or close the connection. There is an open issue in this area; see [Section 9](#).

[6](#). Brief Security Comparison of DH-CHAP with CHAP and SRP

This section attempts to summarize important points of comparison among DH-CHAP and the CHAP and SRP authentication methods already specified in iSCSI. This is not intended to be a complete list,

but rather a listing of the important points. The author intends to revise this section to reflect important points from discussion on the mailing list as they develop.

[6.1](#) Passive Eavesdropping.

CHAP allows a passive eavesdropper to gain information sufficient to mount an off-line dictionary attack on the CHAP secret. DH-CHAP prevents this, but in a fashion that may restrict DH-CHAP usage of existing servers (e.g., RADIUS) that verify CHAP authentication; see [Section 7](#). SRP protects its secret from passive eavesdropping.

[6.2](#) Use of Secret Information to Verify Authentication

SRP does not require secret information to be stored at the Responder if bi-directional authentication is not required; the SRP verifier can be made public without revealing secret information. Both CHAP and DH-CHAP require that the Responder or the system that verifies authentication for the Responder have access to the CHAP or DH-CHAP secret, C_k .

[6.3](#) Impersonation and Man-in-the-Middle Attacks

An impersonation attack involves an attacker who does not know the secret impersonating a communicating party. A man-in-the-middle attack involves the attacker inserting himself between the communicating parties to read and modify communication between them. SRP prevents impersonation and man-in-the-middle attacks from obtaining the secret or information sufficient to mount a dictionary attack on it.

Impersonation and man-in-the-middle attacks on CHAP and DH-CHAP disclose sufficient information to mount an off-line dictionary attack against the secret (C_k). CHAP and DH-CHAP Initiators are vulnerable to both sorts of attacks. CHAP and DH-CHAP Responders are not vulnerable to impersonation attacks because bi-directional CHAP and DH-CHAP as used in iSCSI require the Initiator to authenticate first, and forbid the Responder from sending its response if that authentication fails (see [Section 6.4](#)). CHAP Responders are vulnerable to man-in-the-middle attacks because the man-in-the-middle need only imitate a passive eavesdropper to succeed. DH-CHAP Responders are not vulnerable to man-in-the-middle attacks because the man-in-the-middle (M) has to conduct two independent Diffie-Hellman exchanges with I and R in order to capture the information required to mount a dictionary attack on DH-CHAP. In this case:

DH-CHAP: Diffie-Hellman Enhanced CHAP for iSCSI April 2002

I <-- DH-CHAP <1> --> M <-- DH-CHAP <2> --> R

k (at I) and k' (at R) will not be the same because they depend on different random numbers whose generation (at I and R) M cannot control. As a result, authentication of I will fail because I will have sent a response (C_r) based on k which is different from k'; thus failure causes R to not send its response. The failed authentication of I occurs too late to protect I; M already has I's response, but this failure may serve as a warning that an attack may have taken place.

Impersonation attacks against one-way authentication are hard to detect because the impersonating attacker is not required to authenticate, and hence can simulate any one of a number of plausible reasons to terminate the connection after obtaining the information of interest. For SRP, this is of no consequence, as the attacker obtains no useful information about the secret, but this is a risk for both CHAP and DH-CHAP.

Impersonation attacks against bidirectional authentication may simulate failure of Initiator authentication (e.g., by closing the connection instead of responding). iSCSI Initiators MUST treat any login failure that causes bi-directional CHAP or DH-CHAP to fail to complete after the Initiator has sent its response as a potential security issue (e.g., treat the error in the same fashion as an authentication failure).

[6.4](#) Reflection Attacks

The topic of reflection attacks was raised on the list, described as "the Target sends you a challenge, the Initiator sends the same challenge back to the Target". If the same CHAP or DH-CHAP secret (C_k) is being used for both directions of a bi-directional authentication, the Initiator and Target responses (C_r) are identical if the identifier (C_i) is also reflected. In this situation, if the Target responds to the challenge, it provides a rogue Initiator with the exact response (C_r) that is required to authenticate the Initiator to the Target. Needless to say, this must not be permitted to occur.

As CHAP and DH-CHAP are used in iSCSI, this reflection attack is almost prevented by the requirement that a Target MUST NOT continue if authentication of the Initiator fails. That requirement needs to be strengthened to require that a Target MUST NOT send its CHAP or DH-CHAP response if the Initiator has not successfully authenticated.

DH-CHAP: Diffie-Hellman Enhanced CHAP for iSCSI April 2002

For example, the following exchange:

```
I->R    CHAP_A(A1,A2,...)
R->I    CHAP_A, CHAP_C, CHAP_I
I->R    CHAP_A, CHAP_C, CHAP_I
```

MUST result in the Responder (Target) closing the iSCSI TCP connection because the Initiator has failed to authenticate (there is no CHAP_R in the third message). In addition Initiators MUST NOT reuse the CHAP or DH-CHAP challenge sent by the Target for the other direction of a bi-directional authentication. Targets MUST check for this condition and close the iSCSI TCP connection if it occurs.

[6.5](#) IPsec and DH-CHAP

Neither CHAP nor DH-CHAP defend against all active attacks such as impersonation and man-in-the-middle, and CHAP does not defend against a passive eavesdropper. In environments where these or other active attacks are a concern, DH-CHAP SHOULD NOT be used without additional protection. IPsec (IKE and ESP) provides the iSCSI defenses against these classes of attacks.

The iSCSI requirement for IPsec is "MUST implement," not "MUST use," and one can expect that administrators will choose not to use IPsec for a variety of reasons. To deal with such situations, the "MUST implement" iSCSI authentication mechanism needs to have an appropriate level of security on its own, although this level of security need not defend against all the attacks that IPsec prevents. For example, sending passwords in the clear and relying on IPsec to address all security issues would not be acceptable.

When IPsec is not in use, an attacker may choose to wait until

authentication is complete and attack (e.g., hijack) the TCP connection, but attacking CHAP or DH-CHAP may yield something more valuable - a secret that could be used to authenticate in the future. Hence defending against dictionary attacks can still be important when IPsec is not in use.

[6.6](#) Passwords vs. Machine-generated keys

The dictionary attacks against CHAP and DH-CHAP are based on the assumption that the CHAP and DH-CHAP secrets are human-generated passwords. If these secrets were instead machine-generated keys with sufficient randomness (e.g., 128 bits would suffice), vulnerability to dictionary attack would no longer be a concern (e.g., an off-line exhaustive search attack against a 128-bit CHAP key generated from real randomness is prohibitively expensive to mount). The downside of such keys is that they are difficult for

DH-CHAP: Diffie-Hellman Enhanced CHAP for iSCSI April 2002

humans to handle and hence require administrative mechanisms to transfer and install keys (e.g., instead of writing the password on a scrap of paper, an administrator could carry a floppy between systems assuming that all systems involved have floppy drives). Systems that use IP Storage (especially iSCSI) may have a circular dependency if the IP Storage may be required to boot the system to the point that the mechanism to accept the key required to access the IP Storage becomes operational.

[7.](#) External Authentication Server Considerations

If a RADIUS or some other external server is used to verify DH-CHAP responses, the connection between the Responder and that server may be the weak link because the C_{ca}' and C_r sent over that connection provide sufficient information to mount a passive dictionary attack on C_k . If an eavesdropper can observe these values by monitoring that connection, DH-CHAP's additional protection against passive attack gained from the Diffie-Hellman exchange is lost. Any such connection to an external server to verify DH-CHAP responses MUST use confidentiality (e.g., IPsec ESP) or be protected from eavesdroppers via other means. Examples of "other means" include use of a separate isolated network for all RADIUS traffic to protect against eavesdroppers, and the use of traffic filters to prevent RADIUS traffic from escaping into areas of the network that are vulnerable to eavesdroppers. For bi-directional usage of DH-CHAP, this requirement also applies to any

connection from an Initiator to an external response verification server.

[7.1](#) DH-CHAP and EAP

The Extensible Authentication Protocol (EAP) (defined in [RFC 2284], being updated in [[2284bis](#)]) describes a framework that allows the use of multiple authentication mechanisms. This section (based loosely on section 6 of [[EAP-SRP](#)]) shows examples on how DH-CHAP might be used with EAP, but does not give the formal definition that would be needed to actually do so.

With the extensions to RADIUS (as defined in [[RFC 2869](#)]), the authenticating endpoint can be moved to a RADIUS server, or even beyond to a back end authentication server. This avoids some of the security issues discussed in the previous section on RADIUS by not exposing any more information than what is already exposed between the peer systems. Note that a RADIUS server would have to be upgraded though (in some way) to support EAP DH-CHAP, or any new EAP protocol.

In the following examples, the named parameters (g , n , C_i , C_c , C_r) are the same as described in the previous sections. DH-CHAP

DH-CHAP: Diffie-Hellman Enhanced CHAP for iSCSI April 2002

represents the EAP Type that would be assigned for EAP DH-CHAP. The `id` parameter is used to match EAP-Responses with EAP-Requests. Note that if DH-CHAP is always used with EAP, the `C_i` parameter could be removed.

[7.1.1](#) Successful Authentication

In the case where the EAP DH-CHAP authentication is successful, the conversation may appear as follows:

Authenticatee	Authenticator
-----	-----
	<- EAP-Request id=65 / Identity
EAP-Response id=65 / Identity ("Initiator") ->	
	<- EAP-Request id=66 / DH-CHAP ($g^x \bmod n$, g , n , C_i , C_c)
EAP-Response id=66 / DH-CHAP ($g^y \bmod n$, C_r) ->	

<- EAP-Success id=67

[7.2.2](#) Unsuccessful Authentication

In the case where the EAP DH-CHAP authentication is unsuccessful, the conversation may appear as follows:

Authenticatee	Authenticator
	<- EAP-Request id=79 / Identity
EAP-Response id=79 / Identity ("Initiator")	->
	<- EAP-Request id=80 / DH-CHAP ($g^x \bmod n$, g , n , C_i , C_c)
EAP-Response id=80 / DH-CHAP ($g^y \bmod n$, C_r)	->
	<- EAP-Failure id=81

[8](#). Security Considerations

This entire draft is about security.

[9](#). Open Issues

Group support. Need a list of Diffie-Hellman groups or group sizes that MUST be supported and a list of g and n values that SHOULD be used for various sizes of groups. This is also the case for SRP, for which Section 7.2 of [[iSCSI](#)] says:

The strength of the SRP authentication method (specified in Chapter 13) is dependent on the characteristics of the group

being used (i.e., the prime modulus N and generator g). As described in [[RFC2945](#)], N is required to be a Sophie-German prime (of the form $N = 2q + 1$, where q is also prime) and the generator g is a primitive root of $GF(n)$. In iSCSI authentication, the prime modulus N MUST be at least 768 bits.

Upon receiving N and g from the Target, the Initiator MUST verify that they satisfy the above requirements (and otherwise, abort the connection). This verification MAY start by trying to match N and g with a well-known group that satisfies the above requirements. Well-known SRP groups are provided in [SEC-IPS].

A better approach may be to explicitly require support and use of specific groups in order to avoid the need to test for N being a Sophie-German prime and g being a primitive root of GF(n).

The current design of bi-directional DH-CHAP protects responders from rogue initiators, but not vice-versa. This could be changed by having the responder (target) authenticate first rather than the initiator. It's not clear that this makes a significant difference, as a successful dictionary attack against a responder secret can be used to impersonate the responder to the initiator to attack the initiator directly or obtain information to mount a dictionary attack on the initiator's secret.

[10.](#) Change Log

[10.1](#) -00 to -01

- Changed author from "Hain" to "Black" in pp.2+ footers.
- Rewrote [section 6.4](#) to incorporate explanation and example of reflection attack from Paul Koning.
- Removed "(which will generally lead to an authentication failure)" from the Overview, as it is not true of impersonation attacks on one-way authentication.
- Strengthened the warning in [Section 6.5](#) that IPsec needs to be used when active attacks against DH-CHAP are a concern.
- Lots of editorial changes.

References

- [2284bis] L. Blunk, J. Vollbrecht, B Aboba, "Extensible Authentication Protocol (EAP)," [draft-ietf-pppext-rfc2284bis-03.txt](#), Work in Progress, 2 April 2002.
- [EAP-SRP] J. Carlson, B. Aboba, H. Haverinen, "EAP SRP-SHA1 Authentication Protocol", [draft-ietf-pppext-eap-srp-03.txt](#), Work in Progress, July 2001.

- [iSCSI] Satran, J., et. al., "iSCSI", [draft-ietf-ips-iscsi-12.txt](#), Work in Progress, March 2002.
- [RFC 1994] Simpson, W., "PPP Challenge Handshake Authentication

Protocol (CHAP)", [RFC 1994](#), August, 1996.

[RFC 2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [RFC 2119](#), [BCP 14](#), March, 1997.

[RFC 2284] L. Blunk, J. Vollbrecht, "PPP Extensible Authentication Protocol (EAP)", [RFC 2284](#), March 1998.

[RFC 2869] C. Rigney, W. Willats, P. Calhoun, "RADIUS Extensions", [RFC 2869](#), June 2000.

[RFC 2945] Wu, T., "The SRP Authentication and Key Exchange System", [RFC 2945](#), September, 2000.

[Schneier] Schneier, B., "Applied Cryptography, Second Edition", New York: John Wiley & Sons, Inc., 1996.

Acknowledgements

A combination of Diffie-Hellman with CHAP was originally suggested by Steve Bellovin. The augmentation approach of concatenating the DH key to the CHAP challenge was suggested by Uri Blumenthal. Steve Senum contributed the text on EAP in [Section 7.1](#) and its subsections. The explanation of reflection attacks and the example in [Section 6.4](#) are largely based on Paul Koning's discussion of this topic on the IPS mailing list. Improvements have resulted from comments on earlier versions of this draft by a number of people, including Ofer Biran and Mark Bakke. Additional comments on various topics from the IPS WG mailing list have been incorporated.

Author's Address

David L. Black
EMC Corporation
42 South Street
Hopkinton, Mass., 01748, USA

Phone: +1 (508) 249-6449
Email: black_david@emc.com