Network Working Group
Internet Draft
February 2007
Expires in six months

R. Housley
Vigil Security
B. Aboba
Microsoft

# Guidance for AAA Key Management
<draft-housley-aaa-key-mgmt-09.txt>

## Status of this Memo



## Abstract

This document provides guidance to designers of AAA key management protocols. The guidance is also useful to designers of systems and solutions that include AAA key management protocols. Given the complexity and difficulty in designing secure, long-lasting key management algorithms and protocols by experts in the field, it is almost certainly inappropriate for IETF working groups without deep expertise in the area to be designing their own key management algorithms and protocols based on Authentication, Authorization and Accounting (AAA) protocols. The guidelines in this document apply to documents requesting publication as IETF RFCs. Further, these guidelines will be useful to other standards development organizations (SDOs) that specify AAA key management.

Internet Draft draft-housley-aaa-key-mgmt-07.txt February 2007

Table of Contents



## 1. Introduction

This document provides architectural guidance to designers of AAA key management protocols. The guidance is also useful to designers of systems and solutions that include AAA key management protocols.

AAA Key Management often includes a collection of protocols, one of which is the AAA protocol. Other protocols are used in conjunction with the AAA protocol to provide an overall solution. These other protocols often provide authentication and security association establishment.

Given the complexity and difficulty in designing secure, long-lasting key management algorithms and protocols by experts in the field, it is almost certainly inappropriate for IETF working groups without deep expertise in the area to be designing their own key management algorithms and protocols based on Authentication, Authorization and Accounting (AAA) protocols. These guidelines apply to documents requesting publication as IETF RFCs. Further, these guidelines will be useful to other standards development organizations (SDOs) that specify AAA key management that depends on IETF specifications for protocols such as EAP [RFC3748], RADIUS [RFC2865], and Diameter [RFC3588].

In March 2003, at the IETF 56 AAA Working Group Session, Russ Housley gave a presentation on "Key Management in AAA" [H]. That presentation established the vast majority of the requirements contained in this document. Over the last three years, this





collection of requirements have become known as the "Housley Criteria".

## 1.1. Requirements Specification

The keywords MUST, MUST NOT, REQUIRED, SHALL, SHALL NOT, SHOULD, SHOULD NOT, RECOMMENDED, MAY, and OPTIONAL, when they appear in this document, are to be interpreted as described in RFC 2119 [RFC2119].

An AAA key management proposal is not compliant with this specification if it fails to satisfy one or more of the MUST or MUST NOT statements. An AAA key management proposal that satisfies all the MUST, MUST NOT, SHOULD and SHOULD NOT statements is said to be "unconditionally compliant"; one that satisfies all the MUST and MUST NOT statements but not all the SHOULD or SHOULD NOT requirements is said to be "conditionally compliant".

## 1.2. Mandatory to Implement

The guidance provided in this document is mandatory to implement. However, it is not mandatory to use. That is, configuration at the time of deployment may result in a deployed implementation that does not conform with all of these requirements.

For example, [RFC4072] enables EAP keying material to be delivered from a AAA server to a AAA client without disclosure to third parties. Thus, key confidentiality is mandatory to implement in Diameter [RFC3588]. However, key confidentiality is not mandatory to use.

## 1.3. Terminology

This section defines terms that are used in this document.

AAA
Authentication, Authorization and Accounting (AAA). AAA protocols include RADIUS [RFC2865] and Diameter [RFC3588].

Authenticator
The party initiating EAP authentication. The term authenticator is used in [IEEE-802.1X], and authenticator has the same meaning in this document.





Backend authentication server
A backend authentication server is an entity that provides an authentication service to an authenticator. This terminology is also used in [802.1X].

CHAP
Challenge Handshake Authentication Protocol; a one-way challenge/response authentication protocol defined in [RFC1994].

EAP
Extensible Authentication Protocol, defined in [RFC3748].

EAP server
The entity that terminates the EAP authentication method with the peer. In the case where no backend authentication server is used, the EAP server is part of the authenticator. In the case where the authenticator operates in pass-through mode, the EAP server is located on the backend authentication server.

Key Wrap
The encryption of one symmetric cryptographic key in another. The algorithm used for the encryption is called a key wrap algorithm or a key encryption algorithm. The key used in the encryption process is called a key-encryption key (KEK).

PAP
Password Authentication Protocol; a deprecated cleartext password PPP authentication protocol, originally defined in

[RFC1334].

Party
A party is a processing entity which can be identified as a single role in a protocol.

Peer
The end of the link that responds to the authenticator. In [802.1X], this end is known as the supplicant.

PPP
Point-to-Point Protocol, defined in [RFC1661], provides support for multiprotocol serial datalinks. PPP is the primary IP datalink used for dial-in NAS connection service.





Secure Association Protocol
A protocol for managing security associations derived from EAP and/or AAA exchanges. The protocol establishes a security association, which includes symmetric keys and a context for the use of the keys. An example of a Secure Association Protocol is the 4-way handshake defined within [802.11i].

Session Keys
Keying material used to protect data exchanged after authentication has successfully completed, using the negotiated ciphersuite.

Network Access Server (NAS)
A device which provides an access service for a user to a network. The service may be a network connection, or a value added service such as terminal emulation, as described in [RFC2881].

4-Way Handshake
A Secure Association Protocol, defined in [802.11i], which confirms mutual possession of a Pairwise Master Key by two parties and distributes a Group Key.

## 2. AAA Environment Concerns

Examples of serious flaws plague the history of key management protocol development, starting with the very first attempt to define a key management protocol in the open literature, which was published in 1978 [NS]. A flaw and a fix were published in 1981 [DS], and the fix was broken in 1994 [AN]. In 1995 [L], a new flaw was found in the original 1978 version, in an area not affected by the 1981/1994 issue. All of these flaws were blindingly obvious once described, yet no one spotted them earlier. Note that the original protocol, if it were revised to employ certificates, which of course had yet to be invented, was only three messages. Many proposed AAA key management schemes are significantly more complicated.

This bit of history shows that key management protocols are subtle. Experts can easily miss a flaw. As a result, peer review by multiple experts is essential, especially since many proposed AAA key management schemes are significantly more complicated, In addition, formal methods can help uncover problems [M].

AAA-based key management is being incorporated into standards developed by the IETF and other standards development organizations (SDOs), such as IEEE 802. However, due to ad hoc development of AAA-based key management, AAA-based key distribution schemes have poorly





understood security properties, even when well-studied cryptographic algorithms are employed. More academic research is needed to fully understand the security properties of AAA-based key management in the diverse protocol environments where it is being employed today. In the absence of such research results, pragmatic guidance based on sound security engineering principles is needed.

In addition to the need for interoperability, cryptographic algorithm independent solutions are greatly preferable. Without algorithm independence, the AAA-based key management protocol must be changed whenever a problem is discovered with any of the selected algorithms. As AAA history shows, problems are inevitable. Problems can surface due to age or design failure.

DES [FIPS46] was a well designed encryption algorithm, and it

provided protection for many years. Yet, the 56-bit key size was eventually overcome by Moore's Law. No significant cryptographic deficiencies have been discovered in DES.

The history of AAA underlines the importance of algorithm independence as flaws have been found in authentication mechanisms such as CHAP, MS-CHAPv1 [SM1], MS-CHAPv2 [SM2], Kerberos [W][BM][DLS], and LEAP [B]. Unfortunately, RADIUS [RFC2865] mandates use of the MD5 algorithm for integrity protection, which has known deficiencies, and RADIUS has no provisions to negotiate substitute algorithms. Similarly, the vendor-specific key wrap mechanism defined in [RFC2548] has no provisions to negotiate substitute algorithms.

The principle of least privilege is an important design guideline. This principle requires that a party be given no more privilege than necessary to perform the task assigned to them. Ensuring least privilege requires clear identification of the tasks assigned to each party, and explicit determination of the minimum set of privileges required to perform those tasks. Only those privileges necessary to perform the tasks are granted. By denying to parties unneeded privileges, those denied privileges cannot be used to circumvent security policy or enable attackers. With this principle in mind, AAA key management schemes need to be designed in a manner where each party has only the privileges necessary to perform their role. That is, no party should have access to any keying material that is not needed to perform their own role. A party has access to a particular key if it has access to all of the secret information needed to derive it.

EAP is being used in new ways. The inclusion of support for EAP within IKEv2 and the standardization of robust Wireless LAN security [802.11i] based on EAP are two examples. EAP has also been proposed





within IEEE 802.16e [802.16e] and by the IETF PANA Working Group. AAA-based key management is being incorporated into standards developed by the IETF and other standards development organizations (SDOs), such as IEEE 802. However, due to ad hoc development of AAA-based key management, AAA-based key distribution schemes have poorly understood security properties, even when well-studied cryptographic algorithms are employed. More academic research is needed to fully understand the security properties of AAA-based key management in the

diverse protocol environments where it is being employed today. In the absence of research results, pragmatic guidance based on sound security engineering principles is needed.

EAP selects one end-to-end authentication mechanism. The mechanisms defined in [RFC3748] only support unilateral authentication, and they do not support mutual authentication or key derivation. As a result, these mechanisms do not fulfill the security requirements for many deployment scenarios, including Wireless LAN authentication [RFC4017].

To ensure adequate security and interoperability, EAP applications need to specify mandatory-to-implement algorithms. As described in [RFC3748], EAP methods seeking publication as an IETF RFC need to document their security claims. However, some EAP methods are not based on well-studied models, which makes the validity of these security claims difficult to determine.

In the context of EAP, the EAP peer and server are the parties involved in the EAP method conversation, and they gain access to key material when the conversation completes successfully. However, the lower layer needs keying material to provide the desired protection through the use of cryptographic mechanisms. As a result, a "pass-through" mode is used to provide the keying material, and the lower layer keying material is replicated from the AAA server to the authenticator. The only parties authorized to obtain all of the keying material are the EAP peer and server; the authenticator obtains only the keying material necessary for its specific role. No other party can obtain direct access to any of the keying material; however, other parties may receive keys that is derived from this keying material for a specific purpose as long as the requirements defined in the next section are met.





## 3. AAA Key Management Requirements

The overall goal of AAA key management is to provide cryptographic keying material in situations where key derivation cannot be used by the peer and authenticator. It may not be possible because the authenticator lacks computational power, because it lacks the resources necessary to implement the various authentication mechanisms that might be required, or because it is undesirable for each authenticator to engage in a separate key management conversation.

This section provides guidance to AAA protocol designers, EAP method designers, and security association protocol designers. Acceptable solutions MUST meet all of these requirements.

Cryptographic algorithm independent

The AAA key management protocol MUST be cryptographic algorithm independent. However, an EAP method MAY depend on a specific cryptographic algorithm. The ability to negotiate the use of a particular cryptographic algorithm provides resilience against compromise of a particular cryptographic algorithm. Algorithm independence is also REQUIRED with a Secure Association Protocol if one is defined. This is usually accomplished by including an algorithm identifier and parameters in the protocol, and by specifying the algorithm requirements in the protocol specification. While highly desirable, the ability to negotiate key derivation functions (KDFs) is not required. For interoperability, at least one suite of mandatory-to-implement algorithms MUST be selected. Note that without protection by IPsec as described in [RFC3579] Section 4.2, RADIUS [RFC2865] does not meet this requirement, since the integrity protection algorithm can not be negotiated.

This requirement does not mean that a protocol must support both public-key and symmetric-key cryptographic algorithms. It means that the protocol needs to be structured in such a way that multiple public-key algorithms can be used whenever a public-key algorithm is employed. Likewise, it means that the protocol needs to be structured in such a way that multiple symmetric-key algorithms can be used whenever a symmetric-key algorithm is employed.

Strong, fresh session keys

While preserving algorithm independence, session keys MUST be strong and fresh. Each session deserves an independent session key. Fresh keys are required even when a long replay counter (that is, one that "will never wrap") is used to ensure that loss of state does not cause the same counter value to be used more than once with the same session key.

Some EAP methods are capable of deriving keys of varying strength, and these EAP methods MUST permit the generation of keys meeting a minimum equivalent key strength. BCP 86 [RFC3766] offers advice on appropriate key sizes. The National Institute for Standards and Technology (NIST) also offers advice on appropriate key sizes in [SP800-57].

A fresh cryptographic key is one that is generated specifically for the intended use. In this situation, a secure association protocol is used to establish session keys. The AAA protocol and EAP method MUST ensure that the keying material supplied as an input to session key derivation is fresh, and the secure association protocol MUST generate a separate session key for each session, even if the keying material provided by EAP is cached. A cached key persists after the authentication exchange has completed. For the AAA/EAP server, key caching can happen when state is kept on the server. For the NAS or client, key caching can happen when the NAS or client does not destroy keying material immediately following the derivation of session keys.

Session keys MUST NOT be dependent on one another. Multiple session keys may derived from a higher-level shared secret as long as a one-time value, usually called a nonce, is used to ensure that each session key is fresh. The mechanism used to generate session keys MUST ensure that the disclosure of one session key does not aid the attacker in discovering any other session keys.

Limit key scope

Following the principle of least privilege, parties MUST NOT have access to keying material that is not needed to perform their role. A party has access to a particular key if it has access to all of the secret information needed to derive it.

Any protocol that is used to establish session keys, MUST

specify the scope for session keys, clearly identifying the parties to whom the session key is available.





Replay detection mechanism

The AAA key management protocol exchanges MUST be replay protected, including AAA, EAP and Secure Association Protocol exchanges. Replay protection allows a protocol message recipient to discard any message that was recorded during a previous legitimate dialogue and presented as though it belonged to the current dialogue.

Authenticate all parties

Each party in the AAA key management protocol MUST be authenticated to the other parties with whom they communicate. Authentication mechanisms MUST maintain the confidentiality of any secret values used in the authentication process.

When a secure association protocol is used to establish session keys, the parties involved in the secure association protocol MUST identify themselves using identities that are meaningful in the lower layer protocol environment that will employ the session keys. In this situation, the authenticator and peer may be known by different identifiers in the AAA protocol environment and the lower layer protocol environment, making authorization decisions difficult without a clear key scope. If the lower layer identifier of the peer will be used to make authorization decisions, then the pair of identifiers associated with the peer MUST be authorized by the authenticator and/or the AAA server.

AAA protocols such as RADIUS [RFC2865] and Diameter [RFC3588] provide a mechanism for the identification of AAA clients; since the EAP authenticator and AAA client are always co-resident, this mechanism is applicable to the identification of EAP authenticators.

When multiple base stations and a "controller" (such as a WLAN switch) comprise a single EAP authenticator, the "base station identity" is not relevant; the EAP method conversation takes place between the EAP peer and the EAP server. Also, many base

stations can share the same authenticator identity. The authenticator identity is important in the AAA protocol exchange and the secure association protocol conversation.

Authentication mechanisms MUST NOT employ plaintext passwords. Passwords may be used provided that they are not sent to another party without confidentiality protection.





Peer and authenticator authorization

Peer and authenticator authorization MUST be performed. These entities MUST demonstrate possession of the appropriate keying material, without disclosing it. Authorization is REQUIRED whenever a peer associates with a new authenticator. The authorization checking prevents an elevation of privilege attack, and it ensures that an unauthorized authenticator is detected.

Authorizations SHOULD be synchronized between the peer, NAS, and backend authentication server. Once the AAA key management protocol exchanges are complete, all of these parties should hold a common view of the authorizations associated the other parties.

In addition to authenticating all parties, key management protocols need to demonstrate that the parties are authorized to possess keying material. Note that proof of possession of keying material does not necessarily prove authorization to hold that keying material. For example, within an IEEE 802.11i, the 4-way handshake demonstrates that both the peer and authenticator possess the same EAP keying material. However, by itself, this possession proof does not demonstrate that the authenticator was authorized by the backend authentication server to possess that keying material. As noted in RFC 3579 in section 4.3.7, where AAA proxies are present, it is possible for one authenticator to impersonate another, unless each link in the AAA chain implements checks against impersonation. Even with these checks in place, an authenticator may still claim different identities to the peer and the backend authentication server. As described in RFC

3748 in section 7.15, channel binding is required to enable the peer to verify that the authenticator claim of identity is both consistent and correct.

Keying material confidentiality and integrity

While preserving algorithm independence, confidentiality and integrity of all keying material MUST be maintained.

Confirm ciphersuite selection

The selection of the "best" ciphersuite SHOULD be securely confirmed. The mechanism SHOULD detect attempted roll back attacks.





Uniquely name keys

AAA key management proposals require a robust key naming scheme, particularly where key caching is supported. The key name provides a way to refer to a key in a protocol so that it is clear to all parties which key is being referenced. Objects that cannot be named cannot be managed. All keys MUST be uniquely named, and the key name MUST NOT directly or indirectly disclose the keying material. If the key name is not based on the keying material, then one can be sure that it cannot be used to assist in a search for the key value.

Prevent the Domino effect

Compromise of a single peer MUST NOT compromise keying material held by any other peer within the system, including session keys and long-term keys. Likewise, compromise of a single authenticator MUST NOT compromise keying material held by any other authenticator within the system. In the context of a key hierarchy, this means that the compromise of one node in the key hierarchy must not disclose the information necessary to compromise other branches in the key hierarchy. Obviously, the compromise of the root of the key hierarchy will compromise all of the keys; however, a compromise in one branch MUST NOT result in the compromise of other branches. There are many

implications of this requirement; however, two implications deserve highlighting. First, the scope of the keying material must be defined and understood by all parties that communicate with a party that holds that keying material. Second, a party that holds keying material in a key hierarchy must not share that keying material with parties that are associated with other branches in the key hierarchy.

Group keys are an obvious exception. Since all members of the group have a copy of the same key, compromise of any one of the group members will result in the disclosure of the group key.





Bind key to its context

Keying material MUST be bound to the appropriate context. The context includes the following.

- The manner in which the keying material is expected to be used.
- The other parties that are expected to have access to the keying material.
- The expected lifetime of the keying material. Lifetime of a child key SHOULD NOT be greater than the lifetime of its parent in the key hierarchy.

Any party with legitimate access to keying material can determine its context. In addition, the protocol MUST ensure that all parties with legitimate access to keying material have

the same context for the keying material. This requires that the parties are properly identified and authenticated, so that all of the parties that have access to the keying material can be determined.

The context will include the peer and NAS identities in more than one form. One (or more) name form is needed to identify these parties in the authentication exchange and the AAA protocol. Another name form may be needed to identify these parties within the lower layer that will employ the session key.

## 4. AAA Key Management Recommendations

Acceptable solutions SHOULD meet all of these requirements.

### Confidentiality of Identity

In many environments it is important to provide confidentiality protection for identities. However, this is not important in other environments. For this reason, EAP methods are encouraged to provide a mechanism for identity protection of EAP peers, but such protection is not a requirement.





### Authorization restriction

If peer authorization is restricted, then the peer SHOULD be made aware of the restriction. Otherwise, the peer may inadvertently attempt to circumvent the restriction. For example, authorization restrictions in an IEEE 802.11 environment include:

- Key lifetimes, where the keying material can only be used for a certain period of time;

- SSID restrictions, where the keying material can only be used with a specific IEEE 802.11 SSID;

- Called-Station-ID restrictions, where the keying material can only be used with a single IEEE 802.11 BSSID; and

- Calling-Station-ID restrictions, where the keying material can only be used with a single peer IEEE 802 MAC address.

## 5. Security Considerations

This document provides architectural guidance to designers of AAA key management protocols. The guidance is also useful to designers of systems and solutions that include AAA key management protocols.

In some deployment scenarios, more than one party in the AAA key management protocol can reside on the same host. For example, the EAP authenticator and AAA client are expected to reside on the same entity. Colocation enables a single unique authenticator identity to be sent by the authenticator to the AAA server as well as by the authenticator to the EAP peer. Use of the same identity in both conversations enables the peer and AAA server to confirm that the authenticator is consistent in its identification, avoiding potential impersonation attacks. If the authenticator and AAA client are not colocated, then the authenticator and AAA client identities will differ, and the key scope will not be synchronized between the EAP peer, authenticator and server. Lack of key scope synchronization enables a number of security vulnerabilities, including impersonation. For this reason, a design needs to include mechanisms to ensure that the key scope and key naming are unambiguous.

The AAA server is a trusted entity. When keying material is present at all, it establishes keying material with the peer and distributes keying material to the authenticator using the AAA protocol. It is trusted to only distribute keying material to the authenticator that was established with the peer, and it is trusted to provide that





keying material to no other parties. In many systems, keying material established by the EAP peer and EAP server are combined with publicly available data to derive other keys. The AAA server is trusted to refrain from deriving these same keys even though it has

access to the secret values that are needed to do so.

The authenticator is also a trusted party. The authenticator is trusted not to distribute keying material provided by the AAA server to any other parties. If the authenticator uses a key derivation function to derive additional keying material, the authenticator is trusted to distribute the derived keying material only to the appropriate party that is known to the peer, and no other party. When this approach is used, care must be taken to ensure that the resulting key management system meets all of the principles in this document, confirming that keys used to protect data are to be known only by the peer and authenticator.

EAP is used to authenticate the peer to the AAA/EAP Server. Following successful authentication, the AAA/EAP server authorizes the peer. In many situations, this is accomplished by sending keying material to the authenticator and the peer in separate protocol messages. The authenticator is not directly authenticated to the peer. Rather, the peer determines that the authenticator has been authorized by the AAA/EAP Server by confirming that the authenticator has the same AAA/EAP Server-provided keying material. In some systems, explicit authenticator and peer mutual authentication is possible. This is desirable since it greatly improves accountability.

When MIB modules are developed for AAA protocols or EAP methods, these MIB modules might include managed objects for keying material. The existence of managed objects associated with keying material offers an additional avenue for key compromise if these objects include the keying material itself. Therefore, these MIB modules MUST NOT include objects for private keys or symmetric keys. However, these MIB modules MAY include management objects that expose names and context associated with keys, and they MAY provide a means to delete keys.

## 6. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", RFC 2119, March, 1997.

## 7. Informative References

[802.1X] IEEE Standards for Local and Metropolitan Area Networks: Port based Network Access Control, IEEE Std 802.1X-2004, December 2004.

[802.11i] Institute of Electrical and Electronics Engineers, "Supplement to Standard for Telecommunications and Information Exchange Between Systems -- LAN/MAN Specific Requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: Specification for Enhanced Security", IEEE 802.11i, July 2004.

[802.16e] Institute of Electrical and Electronics Engineers, "Supplement to Standard for Telecommunications and Information Exchange Between Systems -- LAN/MAN Specific Requirements - Part 16: Air Interface for Fixed and Mobile Broadband Wireless Access Systems -- Amendment for Physical and Medium Access Control Layers for Combined Fixed and Mobile Operation in Licensed Bands", Draft, IEEE 802.16e/D8, May 2005.

[AN] M. Abadi and R. Needham, "Prudent Engineering Practice for Cryptographic Protocols", Proc. IEEE Computer Society Symposium on Research in Security and Privacy, May 1994.

[B] Brewin, B., "LEAP attack tool author says he wants to alert users to risks", Computerworld, October 17, 2003.

[BM] Bellovin, S. and M. Merrit, "Limitations of the Kerberos authentication system", Proceedings of the 1991 Winter USENIX Conference, pp. 253-267, 1991.

[DDNN39.2] DCA DDN Program Management Office, "MILNET TAC Access Control", Defense Data Network Newsletter, DDN News 39, Special Issue, 26 Apr 1985. [http://sunsite.uakom.sk/doc/rfc/ddn-news.n39.2]

[DLS] Dole, B., Lodin, S. and E. Spafford, "Misplaced trust: Kerberos 4 session keys", Proceedings of the Internet Society Network and Distributed System Security Symposium, pp. 60-70, March 1997.

[DS] D. Denning and G. Sacco. "Timestamps in key distributed protocols", Communication of the ACM, 24(8):533--535, 1981.

[FIPS46] Federal Information Processing Standards Publication (FIPS PUB) 46, Data Encryption Standard, 1977 January 15.

[H] Housley, R., "Key Management in AAA", Presentation to the AAA WG at IETF 56, March 2003. [http://www.ietf.org/proceedings/03mar/slides/aaa-5/index.html]

[L] G. Lowe. "An attack on the Needham-Schroeder public key authentication protocol", Information Processing Letters, 56(3):131--136, November 1995.

[M] Meadows, C., "Analysis of the Internet Key Exchange Protocol using the NRL Protocol Analyser", Proceedings of the 1999 IEEE Symposium on Security & Privacy, Oakland, CA, USA, IEEE Computer Society, May 1999. [http://chacs.nrl.navy.mil/publications/CHACS/1999/1999meadows-IEEE99.pdf]

[NS] R. Needham and M. Schroeder. "Using encryption for authentication in large networks of computers", Communications of the ACM, 21(12), December 1978.

[RFC0927] Anderson, B.A., "TACACS user identification Telnet option", RFC 927, December 1984.

[RFC1334] Lloyd, B. and B. Simpson, "PPP Authentication Protocols", RFC 1334, October 1992, Obsoleted by RFC 1994.

[RFC1492] Finseth, C., "An Access Control Protocol, Sometimes Called TACACS", RFC 1492, July 1993.

[RFC1661] Simpson, W., "The Point-to-Point Protocol (PPP)", RFC 1661, July 1994.

[RFC1968] Meyer, G., "The PPP Encryption Protocol (ECP)", RFC 1968, June 1996.





[RFC1994] Simpson, W., "PPP Challenge Handshake Authentication Protocol (CHAP)", RFC 1994, August 1996.

[RFC2409] Harkins, D. and D. Carrel, "The Internet Key Exchange (IKE)", RFC 2409, November 1998.

[RFC2419] Sklower, K. and G. Meyer, "The PPP DES Encryption Protocol, Version 2 (DESE-bis)", RFC 2419, September 1998.

[RFC2420] Hummert, K., "The PPP Triple-DES Encryption Protocol (3DESE)", RFC 2420, September 1998.

[RFC2433] Zorn, G. and S. Cobb, "Microsoft PPP CHAP Extensions", RFC 2433, October 1998.

[RFC2548] Zorn, G., "Microsoft Vendor-specific RADIUS Attributes", RFC 2548, March 1999.

[RFC2637] Hamzeh, K., Pall, G., Verthein, W., Taarud, J., Little, W. and G. Zorn, "Point-to-Point Tunneling Protocol (PPTP)", RFC 2637, July 1999.

[RFC2716] Aboba, B. and D. Simon, "PPP EAP TLS Authentication Protocol", RFC 2716, October 1999.

[RFC2759] Zorn, G., "Microsoft PPP CHAP Extensions, Version 2", RFC 2759, January 2000.

[RFC2865] Rigney, C., Willens, S., Rubens, A. and W. Simpson, "Remote Authentication Dial In User Service (RADIUS)", RFC 2865, June 2000.

[RFC2881] D. Mitton, M. Beadles, "Network Access Server Requirements Next Generation (NASREQNG) NAS Model", RFC 2881, July 2000.

[RFC3078] Pall, G. and G. Zorn, "Microsoft Point-To-Point Encryption (MPPE) Protocol", RFC 3078, March 2001.

[RFC3079] Zorn, G., "Deriving Keys for use with Microsoft Point-to-Point Encryption (MPPE)", RFC 3079, March 2001.

[RFC3579] Aboba, B. and P. Calhoun, "RADIUS (Remote Authentication Dial In User Service) Support For Extensible Authentication Protocol (EAP)", RFC 3579, September 2003.





[RFC3588] Calhoun, P., Loughney, J., Guttman, E., Zorn, G. and J. Arkko, "Diameter Base Protocol", RFC 3588, September 2003.

[RFC3748] Aboba, B., Blunk, L., Vollbrecht, J., Carlson, J. and H. Levkowetz, "Extensible Authentication Protocol (EAP)", RFC 3748, June 2004.

[RFC3766] Orman, H. and P. Hoffman, "Determining Strength for Public Keys Used For Exchanging Symmetric Keys", RFC 3766, April 2004.

[RFC4017] Stanley, D., Walker, J. and B. Aboba, "Extensible Authentication Protocol (EAP) Method Requirements for Wireless LANs", RFC 4017, March 2005.

[RFC4072] Eronen, P., Ed., Hiller, T., and G. Zorn, "Diameter Extensible Authentication Protocol (EAP) Application", RFC 4072, August 2005.

[RFC4306] Kaufman, C., Ed., "Internet Key Exchange (IKEv2) Protocol", RFC 4306, December 2005.

[SM1] Schneier, B. and Mudge, "Cryptanalysis of Microsoft's Point-to-Point Tunneling Protocol", Proceedings of the 5th ACM Conference on Communications and Computer

Security, ACM Press, November 1998.

[SM2] Schneier, B. and Mudge, "Cryptanalysis of Microsoft's PPTP Authentication Extensions (MS-CHAPv2)", CQRE 99, Springer-Verlag, 1999, pp. 192-203.

[SP800-57] National Institute of Standards and Technology, "Recommendation for Key Management", Special Publication 800-57, May 2006.

[W] Wu, T., "A Real-World Analysis of Kerberos Password Security", Proceedings of the 1999 ISOC Network and Distributed System Security Symposium, [http://www.isoc.org/isoc/conferences/ndss/99/ proceedings/papers/wu.pdf]





## Appendix: AAA Key Management History

Protocols for Authentication, Authorization and Accounting (AAA) were originally developed to support deployments of Network Access Servers (NASes). In the ARPAnet, the Terminal Access Controller (TAC) provided a means for "dumb terminals" to access the network, and the TACACS [RFC0927][RFC1492] AAA protocol was designed by BBN under contract to the Defense Data Network Program Management Office (DDN PMO) for this environment. [RFC1492] documents a later version of TACACS, not the original version that was widely deployed in ARPAnet and MILNET [DDNN39.2].

Later, additional AAA protocols were developed to support deployments of NASes providing access to the Internet via PPP [RFC1661]. In deployments supporting more than a modest number of users, it became impractical for each NAS to contain its own list of users and associated credentials. As a result, additional AAA protocols were developed, including RADIUS [RFC2865] and Diameter [RFC3588]. These protocols enabled a central AAA server to authenticate users

requesting network access, as well as providing authorization and accounting.

While PPP [RFC1661] originally supported only PAP [RFC1334] and CHAP [RFC1661] authentication, the limitations of these authentication mechanisms became apparent. For example, both PAP and CHAP are unilateral authentication schemes supporting only authentication of the PPP peer to the NAS. Since PAP is a cleartext password scheme, it is vulnerable to snooping by an attacker with access to the conversation between the PPP peer and NAS. In addition, the use of PAP creates vulnerabilities within RADIUS as described in Section 4.3 of [RFC3579]. As a result, use of PAP is deprecated. While CHAP, a challenge-response scheme based on MD5, offers better security than cleartext passwords, it does not provide for mutual authentication, and CHAP is vulnerable to dictionary attack.

With the addition of the Encryption Control Protocol (ECP) to PPP [RFC1968] as well as the definition of PPP ciphersuites in [RFC2419] [RFC2420][RFC3078] the need arose to provide keying material for use with link layer ciphersuites. As with user authentication, provisioning of static keys on each NAS did not scale well.

Additional vendor-specific PPP authentication protocols such as MS-CHAP [RFC2433] and MS-CHAPv2 [RFC2759] were developed to provide mutual authentication as well as key derivation [RFC3079] for use with negotiated ciphersuites, and they were subsequently adapted for use with PPP-based VPNs [RFC2637]. As with PAP and CHAP, flaws were subsequently found in these new mechanisms [SM1][SM2].





Even though PPP provided for negotiation of authentication algorithms, addressing the vulnerabilities found in authentication mechanisms still proved painful, since new code needed to be deployed on PPP peers as well as on the AAA server. In order to enable more rapid deployment of new authentication mechanisms, as well as fixes for vulnerabilities found in existing methods, the Extensible Authentication Protocol (EAP) [RFC3748] was developed, along with support for centralized authentication via RADIUS/EAP [RFC3579].

By enabling "pass through" authentication on the NAS, EAP enabled deployment of new authentication methods or updates to existing methods by revising code only on the EAP peer and AAA server. The

initial authentication mechanisms defined in [RFC2284] (MD5-Challenge, One-Time Password (OTP), and Generic Token Card (GTC)) only supported unilateral authentication, and these mechanisms do not support key derivation. Subsequent authentication methods such as EAP-TLS [RFC2716] supported mutual authentication and key derivation.

In order to support the provisioning of dynamic keying material for link layer ciphersuites in an environment supporting centralized authentication, a mechanism was needed for the transport of keying material between the AAA server and NAS. Vendor-specific RADIUS attributes were developed for this purpose [RFC2548]. Vulnerabilities were subsequently found in the key wrap technique, as described in Section 4.3 of [RFC3579].

In theory, public key authentication mechanisms such as EAP-TLS are capable of supporting mutual authentication and key derivation between the EAP peer and NAS without requiring AAA key distribution. However, in practice such pure two-party schemes are rarely deployed. Operation of a centralized AAA server significantly reduces the effort required to deploy certificates to NASes, and even though a AAA server may not be required for key derivation and possibly authentication, its participation is required for service authorization and accounting.

"Pass-through" authentication and AAA key distribution has retained popularity even in the face of rapid improvements in processor and memory capabilities. In addition to producing NAS devices of increased capability for enterprise and carrier customers, implementers have also produced low cost/high volume NAS devices such as 802.11 Access Points, causing the resources available on an average NAS to increase more slowly than Moore's law. Despite widespread support for certificate handling and sophisticated key derivation mechanisms such as IKEv1 [RFC2409] within host operating systems, these security capabilities are rarely deployed on low-end NASes and clients.





Even on more capable NASes, such as VPN servers, centralized authentication and AAA key management has proven popular. For example, one of the major limitations of IKEv1 [RFC2409] was the lack of integration with EAP and AAA, requiring proprietary extensions to enable use of IPsec VPNs by organizations deploying password or

authentication tokens. These limitations were addressed in IKEv2 [RFC4306], which while handling key derivation solely between the VPN client and server, supports EAP methods for user authentication. In order to enable cryptographic binding of EAP user authentication to keys derived within the IKEv2 exchange, the transport of EAP-derived keys within AAA is required where the selected EAP method supports key derivation.

## Acknowledgments

Many thanks to James Kempf, Sam Hartman, and Joe Salowey for their quality review and encouragement.

Thanks to the IETF AAA Working Group and the IETF EAP Working Group for their review and comment. The document is greatly improved by their contribution.

## Authors' Address

Russell Housley
Vigil Security, LLC
918 Spring Knoll Drive
Herndon, VA 20170
USA
Email: housley@vigilsec.com
Phone: +1 703-435-1775
Fax: +1 703-435-1274

Bernard Aboba
Microsoft Corporation
One Microsoft Way
Redmond, WA 98052
USA
Email: bernarda@microsoft.com
Phone: +1 425-706-6605
Fax: +1 425-936-7329

## Intellectual Property Statement



## Disclaimer of Validity



## Copyright Statement