

Internet Draft
Expiration: December 2002
File: [draft-walker-aaa-key-distribution-00.txt](#)

Jesse Walker
Intel Corporation
Russ Housley
RSA Labs
Nancy Cam-Winget

AAA Key Distribution
Last Updated: April 15, 2002

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>.

To view the entire list of current Internet-Drafts, please check the "lid-abstracts.txt" listing contained in the Internet-Drafts Shadow Directories on ftp.is.co.za (Africa), ftp.nordu.net (Northern Europe), ftp.nis.garr.it (Southern Europe), munnari.oz.au (Pacific Rim), ftp.ietf.org (US East Coast), or ftp.isi.edu (US West Coast).

Copyright (C) The Internet Society (2002). All Rights Reserved.

Abstract

This memo describes problems with the current AAA NASREQ key distribution mechanisms, and proposes enhancements to the NASREQ key distribution model to address these problems.

Please send comments on this document to the aaa-wg@merit.edu mailing list.

1. Introduction

The IETF AAA Working Group is developing solutions for Authentication, Authorization and Accounting as applied to network access. The AAA Working Group has defined protocols addressing the network access needs specified by the NASREQ, MOBILE IP, and ROAMOPS

Walker et al.

Expires December 2002

[Page 1]

Internet Draft

AAA Key Distribution

April 2002

Working Groups as well as TIA 45.6. The solution specified by the AAA Working Group is also thought to address the needs of IEEE 802.11 wireless networks. The solution is intended to augment and eventually replace RADIUS.

One area under the AAA Working Group's purview is the subject of session key distribution. For the purposes here, a session key is a cryptographic key that is used either directly or indirectly to protect traffic exchanged over a link between a Network Access Server (NAS) and one of its clients. [[NASREQ](#)] describes the key exchange mechanism. This document analyzes the NASREQ key distribution mechanism, identifies a number of security weaknesses, and proposes some enhancements that rectify the identified weaknesses.

1.1. Requirements Terminology

Keywords "MUST", "MUST NOT", "REQUIRED", "SHOULD", "SHOULD NOT" and "MAY" that appear in this document are to be interpreted as described in [[RFC2119](#)].

2. Description of NASREQ Key Distribution

[NASREQ] defines session key distribution from an AAA server to a NAS. The model (or models) it uses are never explicitly specified, but it is possible to infer the model from the documentation available.

The purpose of NASREQ key distribution is to securely establish a session key between the NAS and the NAS client. The AAA server may distribute more than one key to the NAS, each to provide different functions. For instance, it can distribute both encryption and data authenticity keys, or send and receive keys, or master keys that can be used to derive other keys.

[NASREQ] allows the AAA server to distribute a key to the NAS client using EAP, but does not specify how this is accomplished. EAP fails to specify mechanisms. As a result, all mechanisms assume that the AAA server and the NAS client already share a key that may be used directly to protect the link between the NAS and the NAS client, and so it is unnecessary to distribute any key to the NAS client. Since no specified instances of EAP key distribution to the NAS client exist, the implicit assumption has to be that such mechanisms are unimportant and will not be deployed as part of the AAA architecture. That is, the lack of any defined EAP key distribution mechanism, and the further lack of any work on such a mechanism, implies that the AAA Working Group implicitly assumes it is only necessary to distribute the key to the NAS. This de facto NASREQ key distribution architecture is the object of our critique.

To effect key distribution, at some (unspecified) point in the authentication process, the AAA server decides to distribute a key to the NAS. This is accomplished by inserting a NAS-Session-Key AVP in an Answer packet sent by the AAA server to the NAS. Each distributed key is included in its own NAS-Session-Key AVP.

The NAS-Session-Key AVP has the following structure:

```
NAS-Session-Key ::= < AVP Header: 408 >
                    { NAS-Key-Direction }
                    { NAS-Key-Type }
                    { NAS-Key }
                    { NAS-Key-Data }
                    [ NAS-Key-Binding ]
                    [ NAS-Key-Lifetime ]
                    [ NAS-IV ]
```

Here

- o NAS-Key-Direction tells whether the key is bi-directional, used for upstream communication, or for downstream communication;
- o NAS-Key-Type tells whether the key is a cipher key, an integrity key, or some type of master key, used for further key derivation;

- o NAS-Key is never defined;
- o NAS-Key-Data is the distributed key itself;
- o NAS-Key-Binding indicates the cryptographic primitive to use with the key, and it is optional;
- o NAS-Key-Lifetime gives the number of seconds remaining before the key expires, and it is optional; and
- o NAS-IV may be used as an initialization vector, and it is optional.

To protect the key distribution, [[NASREQ](#)] permits three different mechanisms:

1. IPsec can be used to protect the entire Answer packet conveying the NAS-Session-Key AVP. In this case, ESP would be used to provide confidentiality, integrity, and data origin authentication.
2. TLS can be used to protect the entire Answer packet, or TLS can be used to protect a portion of the packet.
3. CMS can be used to envelope any included NAS-Session-Key AVPs.

[[NASREQ](#)] specifies no mandatory-to-implement protections for the key distribution.

[3.](#) Analysis of [NASREQ](#) Key Distribution

This section argues there are five fundamental problems with the model assumed by the [NASREQ](#) key distribution mechanism:

1. It is incompatible with static keys.

2. It has an inadequate, non-existent key naming scheme, which opens the mechanism to numerous abuses.
3. It provides inadequate protection for distributed keys.

4. It introduces a novel key distribution architecture with poorly understood security properties, while key distribution is an area bedeviled by subtle bugs.
5. It is not feasible to design the key distribution exchange between the AAA server and the NAS in isolation of any associated handshaking between the NAS and the NAS client.

3.1. Architecture incompatible with static keys

In the de facto key distribution model described above, the key distributed by the AAA server to the NAS must not ever be used again, with either the same or with any other NAS. To permit the distributed key to be reused with the same NAS, both the NAS and the NAS client would have to maintain a record of the consumed sequence spaces, nonce spaces, and replay windows used with the key, to prevent inadvertent compromise on reuse, something that is not in general feasible or desirable. To permit the key to be used by a different NAS, a mechanism to prevent the original NAS from using the key to masquerade as the NAS client would be needed. Some people may argue that this latter problem is not a genuine concern because the NAS is trusted. However, the consequences of NAS compromise must be considered. The NAS is not immune from compromise; within the past year alone, for example, the Code Red virus and SNMP buffer overrun alert have applied to nearly every NAS, and almost all NASes required patches as a consequence. Therefore, NASREQ key distribution cannot be used with static keys; all distributed keys must be fresh, never-used-before keys.

Another approach exists that can safely employ static keys. Under this approach the static key remains a secret shared only between the AAA server and the NAS client; it is never shared with the NAS. The AAA server employs this static key to distribute a key to the NAS client at the same time it distributes a key to the NAS. The cost of this increased flexibility is that the AAA server generates a random key and distributes it both to the NAS client and to the NAS, not just the NAS, as in the present de facto architecture.

Proponents of the NASREQ approach might argue that the use of protocols like TTLS or PEAP will alleviate this problem. They reason that TTLS derives a fresh key at initial contact, and TLS-resume can be used to derive another fresh key at the time of reattachment. This is an attractive line of thinking, but suffers from two problems.

The first problem is that organizations deploying TTLS or PEAP still have to obtain an X.509 certificate for their AAA servers and deploy a trust anchor that allows the X.509 certificate to be validated on

the NAS client. The trust anchor only contains public data, but integrity must be maintained. If the trust anchor can be altered, then the NAS clients cannot properly authenticate the AAA server, and the NAS clients are subject to rogue NASes. This trust anchor provisioning can be costly. It is probably an unacceptable cost for many simple deployments, especially ad hoc wireless networks.

The second problem with this reasoning is that it leads to a more complex exchange between the NAS client and the AAA server than is actually necessary. An approach distributing the derived key only to the NAS requires at least a three packet exchange between the NAS client and the AAA server (EAP-TLS actually requires a four packet exchange for TLS-resume) to generate a fresh key, while an approach distributing a fresh randomly generated key to both the NAS and the NAS client requires only a two packet exchange. This means that at the time of a reattachment, when an existing key may be used, the de facto architecture implementation is 50% more complex than necessary. A 50% reduction in complexity is a giant win for security analysis, and, when amortized over all reattachments, it provides a significant performance enhancement for the case that is most time critical.

This is not to say that TTLS and PEAP do not provide any value. The argument is rather they do not solve the problem being raised here.

Presumably, the NASREQ vision can be completed by enhancing EAP to distribute an analog of the NAS-Session-Key AVP, delivering a session key to the NAS client. Mandating this usage going forward would improve flexibility by restoring the use of static keys, and would simplify the overall system design.

[3.2.](#) Inadequate key naming

Since NASREQ did not tackle the fundamental issue of key freshness, it also fails to address the issue of binding keys to particular sessions between particular entity pairs. The protocol fails to explicitly name the distributed keys. This is a much more serious problem than the failure to work with static keys, because the history of key distribution protocols shows that failing to properly identify keys presents a major opportunity for compromise of the distributed keys. It is a major vulnerability exploited to launch

attacks.

To prevent these kinds of weaknesses, it is necessary to specify both the NAS client and the NAS identities in the key distribution, to allow their peers to detect when either cheats or uses the key with unintended parties. It is further necessary to explicitly bind the key to a particular session between the NAS and the NAS client, to detect key reuse problems.

The sort of key naming required represents an assertion by the AAA server that the parties may not use the key with any other party, nor with a different session. Thus, the only reasonable identification has to include the AAA identities of the intended pair of peers and some session identifier. This is not needless overhead.

The NAS-Session-Key AVP can be easily enhanced to provide this additional information.

[3.3.](#) Inadequate protections for NAS-Session-Key AVPs

The NAS-Session-Key AVP does not require any explicit mandatory protection. Instead, the NASREQ specification permits implementations to rely upon TLS or IPsec to protect the AVP. The problem with this approach is a practical implementation necessity: AAA server implementations are typically software only, running under general-purpose operating systems; many NAS devices are likewise based on public domain UNIX implementation. In either environment it is usually easy for an attacker to insert a Trojan horse that intercepts the data stream between modules, e.g., between the TCP/IP stack and the socket layer. Such a Trojan horse can read and replace distributed keys if the cryptographic protections are applied by a separate module. This defect in the design may be remedied by mandating that the NAS-Session-Key AVP be CMS encapsulated and design due diligence applied to keep the cryptographic operations from crossing module boundaries.

The issue is deeper than just the layer at which the cryptographic protections are applied. To minimize the immunity of a key distribution protocol to attack, it is necessary to explicitly bind

together the information in a way that the recipient of the distributed key can validate, and this binding typically crosses message boundaries and often must even relate to messages from all three parties in the protocol; [section 3.5](#) below touches further on this theme. These are application protocol issues, and it is simply not reasonable to expect that bilateral mechanisms at other layers can effectively solve these problems. [\[NASREQ\]](#) as it stands today does not exhibit any evidence that this issue has even been contemplated. For instance, the relationship between information in AAA Request messages and Answer messages is never spelled out, being left entirely as a method-specific detail. For some aspects of key distribution to work properly, this relationship has to be defined. Key distribution is not merely a data transport operation; it is also a mechanism for building transitive trust; it is simply infeasible to specify a secure key distribution without binding data across several messages.

As an example of this complaint, even CMS wrapping the NAS-Session-Key AVP does not explicitly protect the key distribution from replay. Thus, although the NAS itself can assume a CMS-wrapped key is genuine and issued from the AAA server, the NAS cannot determine that the AVP it receives has not been issued already for some prior session. Instead, the NAS must assume that the AAA server is playing by the rules and issuing only fresh keys, and that there is no man-in-the-middle replacing AVPs before or after they are unwrapped by a lower-layer security mechanism. Many session establishment handshakes do not define adequate key confirmation handshakes, so the NAS could end up sending new data encrypted under already-used keys and IVs before the problem can be detected, potentially compromising previously sent data. What is needed to defeat this kind of attack is to require the

AAA server to incorporate a challenge from the NAS (and/or the NAS client) into the NAS-Session-Key AVP prior to wrapping. (Inserting a timestamp into the AVP is another option, but maintaining synchronized time in many of the environments served by an AAA server introduces its own problems.) Relying on TLS or IPsec does not solve the problem, because the replay protection afforded by such low-level mechanisms is not adequately bound to the AVP to prevent a Trojan horse from substituting an old AVP for the new one without detection.

While it was poorly implemented, the RADIUS authenticator played a

useful role. It allowed the NAS to detect replay. By removing the authenticator when going forward to DIAMETER, AAA created a structurally weaker protocol than RADIUS. This omission is an opportunity, however, because it would be better to include an authenticator field in the NAS-Session-Key AVP than as part of the larger Answer packet, so it may be more tightly bound to the distributed key.

Finally, the AVP wrapping algorithm is not specified with sufficient granularity. Key distribution protocols make very specific assumption about what is encrypted, what is authenticated but not encrypted, and what must be sent without any protection. [\[NASREQ\]](#) appears to apply the same protection to the entire AVP. This again argues in favor of a mandatory application-specific security protocol.

[3.4.](#) Novel solution to a problem bedeviled by subtle failures

The cryptographic community has not studied the de facto architecture. Key distribution as defined in [\[NASREQ\]](#) is a three party protocol, with the AAA server, the NAS, and the NAS client all parties with an interest in the exchange. Cryptographers have defined protocols that are known to protect the interests of all three parties in such an exchange, but the NASREQ key distribution does not resemble any of these well-studied protocols. This is particularly troubling, because three-party key distribution of this sort appears to be one of the hardest problems cryptographers have attempted to address. Almost all of the initial proposals to this problem have been flawed. There are examples where minor flaws have remained undiscovered for 20 years after the protocol was published. This repeated history of failure puts a premium on the most conservative possible design, restricting it to well scrutinized protocols.

It may be possible to address many of the problems discussed here without adopting a classical, well-studied three-party protocol. It may be, for instance, feasible to clean up the NASREQ architecture to securely distribute keys in an environment where TTLS or PEAP is mandated. However, many years of analysis and scrutiny may be necessary to develop the confidence that this kind of approach is secure from practical attack. It is safer to deploy a protocol known to work correctly than to gamble that the novel design won't be broken after the NASREQ application is widely deployed in the Internet.

3.5. Key Distribution Protocols not Properly Bound

The IETF and the AAA WG have followed the time-honored custom of decomposing problems into separate pieces. In the case of key distribution, the decomposition is into the NASREQ/DIAMETER exchange between the AAA server and the NAS on one hand, and the AAA server and NAS client on the other, the latter being under the purview of the EAP WG. Unfortunately, this decomposition is not correct. It is incorrect because it is difficult to create separate key distribution sub-protocols that, when reassembled into a single system, guarantee the security needs of all three parties involved. A valid key distribution protocol requires that the sub-protocols interact in subtle and non-trivial ways and thus the key distribution specification has to span the domains of at least two Working Groups.

As an example of this, [[NASREQ](#)] permits the distribution of several keys for the same function, e.g., several Master session keys. On the other hand, while allowing this flexibility, it provides no means of indicating the sequence in which these keys are used, and when to change from one key to another. The point is that, to be useful, the usage of the distributed key must be synchronized on the NAS and the NAS client, and the protocol does not provide the information necessary for synchronization.

As a second example, in environments such as IEEE 802.11, an EAP-Success message is inappropriate to signal the open link between the NAS and the NAS client for general traffic. Rather, a key confirmation handshake is required; it is inappropriate to open the link to data traffic until the peer has signaled that its session keys are in place. It is not feasible to design all the details of the key confirmation handshake without also binding the handshake to the details of the key distribution. The reason is that key confirmation requires additional information to be exchanged that would not be configured when there is no trusted third party.

An objection has been raised that an approach based on a classical three-party protocol might be more complex than the present course of the AAA Working Group. This objection is wrong on two accounts. First, it is a necessary complexity, because composition of separately designed protocols does not necessarily lead to a secure overall protocol. Second, increasing the local complexity by binding together protocols which are intrinsically linked reduces the coupling of session key establishment from the remainder of the system, and hence also reduces global complexity.

4. NASREQ Key Distribution Requirements

A key distribution protocol should provide a secure means for affecting use of the session key. In addition to the requirements already spelled out for NASREQ key distribution, the following requirements are also needed for such a key distribution mechanism:

1. Must ensure the session key is fresh;
2. Must name the key;
3. Must bind the key to the intended session between the NAS and NAS client; and
4. Must enforce protection of the session key.

[4.1.](#) Session key freshness

Key distribution mechanisms must ensure that the session key distributed is statistically unique. That is, a session key must never be used again in either subsequent sessions or reused with another NAS. The protocol must allow both the NAS and the NAS client some means of verifying the freshness of the key distribution.

[4.2.](#) Key naming

Each key must be properly identified to a given session corresponding to a NAS and NAS client. The protocol and key naming scheme together must allow the participants to detect the unauthorized use of a distributed key. The protocol must allow each party to verify that the session peer is the other intended recipient of the distributed key.

[4.3.](#) Key binding

A key must be properly bound to a particular NAS and NAS client

session. The key binding is critical to allow both the NAS and NAS client to properly synchronize to a session key. Since the key is ultimately used to establish communications between the NAS and the NAS client, the protocol must be explicit on when the distributed key becomes active as well as allowing the NAS and NAS client to validate and confirm receipt of the key.

4.4. Protection of the session key

The protocol must provide assurances to all three parties (the AAA server, the NAS, and the NAS client) that no other parties have access to the distributed session key, assuming none of the three publishes it either intentionally or inadvertently.

5. Proposed NASREQ Key Distribution Architecture and Enhancements

< To be supplied by 3 May 2002 >

Walker et al.

Expires December 2002

[Page 9]

Internet Draft

AAA Key Distribution

April 2002

6. Security Considerations

This document concerns the security of [[NASREQ](#)]. The authors hope that the analysis presented here will be embraced by the AAA Working Group, resulting in a more secure protocol.

7. Acknowledgements

8. References

- [PROB] Calhoun, P., Aboba, B., Guttman, E., Mitton, D., Nelson, D., Schoenwaelder, J., Wolff, B., Zhang, X., "AAA Problem Statements", work in progress, [draft-ietf-aaa-issues-05.txt](#), January 2002.
- [TRANS] Aboba, B., Wood, J., "Authentication, Authorization, and Accounting (AAA) Transport Profile", work in progress, [draft-ietf-aaa-transport-05.txt](#), November 2001.

- [DIAM] Calhoun, P., Akhtar, H., Arkko, J., Guttman, E., Rubens, A., Zorn, G., "Diameter Base Protocol", work in progress,
- [NASREQ] Calhoun, P., Bulley, W., Rubens, A.C., Haag, J., Zorn., G. "Diameter NASREQ Application", work in progress, [draft-ietf-aaa-diameter-nasreq-08.txt](#), November 2001.
- [CMS] Calhoun, P., Farrell, S., Bulley, W., "The Diameter CMS Security Application", work in progress, [draft-ietf-aaa-diameter-cms-03.txt](#), November 2001.
- [TTLS] Funk, P., Blake-Wilson, S., "EAP Tunneled TLS Authentication Protocol", work in progress, [draft-ietf-pppext-eap-ttls-01.txt](#), February 2002
- [PEAP] Anderson, H., Josefsson, S., Zorn, G., Simon, D., Palekar, A., "Protected EAP Protocol (PEAP)", work in progress, [draft-josefsson-pppext-eap-tls-eap-02.txt](#), February 2002

9. Author Addresses

Jesse Walker
Intel Corporation
2111 N.E. 25th Avenue
Hillsboro, OR 97214
USA
jesse.walker@intel.com

Russell Housley
RSA Laboratories
918 Spring Knoll Drive
Herndon, VA 20170
USA
rhousley@rsasecurity.com

Nancy Cam-Winget
Mountain View, CA 94040
USA
nance@winget.net

Walker et al.

Expires December 2002

[Page 10]

Internet Draft

AAA Key Distribution

April 2002

10. Full Copyright Statement

Copyright (C) The Internet Society (2002). All Rights Reserved.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works. In addition, the ASN.1 modules presented in Appendices A and B may be used in whole or in part without inclusion of the copyright notice. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to the Internet Society or other Internet organizations, except as needed for the purpose of developing Internet standards in which case the

procedures for copyrights defined in the Internet Standards process shall be followed, or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by the Internet Society or its successors or assigns. This document and the information contained herein is provided on an "AS IS" basis and THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

