HOKEY Working Group T. Clancy, Editor
Internet-Draft LTS
Intended status: Informational January 24, 2007
Expires: July 28, 2007

Handover Key Management and Re-authentication Problem Statement
draft-ietf-hokey-reauth-ps-01

Status of this Memo

Copyright Notice

Abstract

This document describes the Handover Keying (HOKEY) problem statement. The current EAP keying framework is not designed to support re-authentication and handovers. This often cause unacceptable latency in various mobile wireless environments. HOKEY plans to address these HOKEY plans to address these problems by implementing a generic mechanism to reuse derived EAP keying material for hand-off.

Table of Contents

1. Authors

The following authors contributed to the HOKEY problem statement draft:

- Julien Bournelle, France Telecom R&D, julien.bournelle@orange-ftgroup.com
- Lakshminath Dondeti, QUALCOMM, ldondeti@qualcomm.com
- Rafael Marin Lopez, Universidad de Murcia, rafa@dif.um.es
- Madjid Nakhjiri, Huawei, mnakhjiri@huawei.com
- Vidya Narayanan, QUALCOMM, vidyan@qualcomm.com
- Mohan Parthasarathy, Nokia, mohan.parthasarathy@nokia.com
- Hannes Tschofenig, Siemens, Hannes.Tschofenig@siemens.com

2. Introduction

The extensible authentication protocol (EAP), specified in RFC3748 [RFC3748] is a generic framework supporting multiple authentication methods. The primary purpose of EAP is network access control. It also supports exporting session keys derived during the authentication. The EAP keying hierarchy defines two keys that are derived at the top level, the master session key (MSK) and the extended MSK (EMSK).

In many common deployment scenario, an EAP peer and EAP server authenticate each other through a third party known as the pass-through authenticator (hereafter referred to as simply "authenticator"). The authenticator is responsible for translating EAP packets from the layer 2 (L2) or layer 3 (L3) network access technology to the AAA protocol.

According to [RFC3748], after successful authentication, the server transports the MSK to the authenticator. The underlying L2 or L3 protocol uses the MSK to derive additional keys, including the transient session keys (TSK) used per-packet access encryption and enforcement. Figure Figure 1 depicts this process.

```
   +--------+       +---------+                     +--------+
   |  EAP   |<----->| Authen- |<--------------------->|  EAP   |
   | Client | L2/L3 | ticator |         AAA          | Server |
   +--------+       +---------+                     +--------+
        ^                                               ^
        |                           +---------+         |
        \-------------------------->| Authen- |<--------/
                     L2/L3          | ticator |   AAA
                                    +---------+

    Initial Authentication:

      <----------------- EAP Authentication ------------->
                       <------ MSK Transport -------------
      <- TSK Generation ->

    Re-Authentication / Hand-off:

      <----------------- EAP Authentication ------------->
                                        <- MSK Transport -
      <---------- TSK Generation --------->
```

Figure 1: Logical diagram of EAP authentication and key derivation using passthrough authenticator

Note that while the authenticator is one logical device, there can be many physical devices involved. For example, in the CAPWAP model [RFC3990] WTPs communicate using L2 protocols with the EAP client and ACs communicate using AAA to the EAP server, while using CAPWAP protocols to communicate with each other. Depending on the configuration, authenticator features can be split in a variety of ways between physical devices, however from the EAP perspective there is only one logical authenticator.

The current models of EAP authentication and keying are unfortunately not efficient in case of mobile and wireless networks. When a peer arrives at the new authenticator, or is expected to re-affirm its access through the current authenticator, the security restraints will require the peer to run an EAP method irrespective of whether it has been authenticated to the network recently and has unexpired keying material. A full EAP method execution involves several round trips between the EAP peer and the server.

There have been attempts to solve the problem of efficient re-authentication in various ways. However, those solutions are either EAP-method specific, EAP lower-layer specific, or are otherwise

limited in scope. Furthermore, these solutions do not deal with scenarios involving handovers to new authenticators, or do not conform to the AAA keying requirements specified in [I-D.housley-aaa-key-mgmt].

This document provides a detailed description of EAP efficient re-authentication protocol requirements.

3. Terminology

In this document, several words are used to signify the requirements of the specification. These words are often capitalized. 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 [RFC2119].

This document follows the terminology that has been defined in RFC3748 [RFC3748] and the EAP Keying Framework [I-D.ietf-eap-keying].

4. Problem Statement

When a peer needs to re-affirm access to an authenticator or moves from one authenticator and reattaches to another authenticator, the current EAP keying model requires the peer to engage in a full EAP exchange with the authentication server in its home domain [RFC3748].

An EAP conversation with a full EAP method run takes several round trips and significant time to complete, causing delays in re-authentication and hand-off times. Some methods [RFC4187] specify the use of keys and state from the initial authentication to finish subsequent authentications in fewer round trips. However, even in those cases, several round trips to the EAP server are still involved. Furthermore, many commonly-used EAP methods do not offer such a fast re-authentication feature. In summary, it is undesirable to have to run a full EAP method each time a peer associates with a new authenticator or needs to extend its current association with the same authenticator. Furthermore, it is desirable to specify a method-independent, efficient, re-authentication protocol. Keying material from the full authentication can be used to enable efficient re-authentication.

Another problem with respect to authentication is when the EAP server is several hops away from the peer, causing too much delay in executing the re-authentication. It is desirable to allow a locally reachable server with EAP efficient re-authentication capability with which the peer can execute such re-authentication without having to

involve the original EAP server all the time. An EAP re-authentication solution defined MUST NOT prevent its extension to a fast re-authentication protocol that operates between EAP servers, and the defined keying hierarchy MUST be designed such that this could be supported.

These problems are the primary issue to be resolved. In solving them, there are a number of constraints to conform to and those result in some additional work to be done in the area of EAP keying.

5. Design Goals

The following are the goals and constraints in designing the EAP re-authentication and key management protocol:

Lower latency operation: The protocol MUST be responsive to handover and re-authentication latency performance objectives within a mobile access network. A solution that reduces latency as compared to a full EAP authentication will be most favorable.

EAP lower-layer independence: Any keying hierarchy and protocol defined MUST be lower layer independent in order to provide the capability over heterogeneous technologies. The defined protocols MAY require some additional support from the lower layers that use it. Any keying hierarchy and protocol defined MUST accommodate inter-technology heterogeneous handover.

EAP method independence: Changes to existing EAP methods MUST NOT be required as a result of the re-authentication protocol. There MUST be no requirements imposed on future EAP methods. Note that the only EAP methods for which independence is required are those that conform to the specifications of [I-D.ietf-eap-keying] and [RFC4017].

AAA protocol compatibility and keying: Any modifications to EAP and EAP keying MUST be compatible with RADIUS and Diameter. Extensions to both RADIUS and Diameter to support these EAP modifications are acceptable. The designs and protocols must satisfy the AAA key management requirements specified in [I-D.housley-aaa-key-mgmt].

Compatability: Compatibility and co-existence with compliant ([RFC3748] [I-D.ietf-eap-keying]) EAP deployments SHOULD be provided. The keying hierarchy or protocol extensions MUST NOT preclude the use of CAPWAP or IEEE 802.11r.

6. Security Goals

The section draws from the guidance provided in [I-D.housley-aaa-key-mgmt] to further define the security goals to be

achieved by a complete re-authentication keying solution.

6.1. Key Context and Domino Effect

Any key MUST have a well-defined scope and MUST be used in a specific context and for the intended use. This specifically means the lifetime and scope of each key MUST be defined clearly so that all entities that are authorized to have access to the key have the same context during the validity period. In a hierarchical key structure, the lifetime of lower level keys MUST NOT exceed the lifetime of higher level keys. This requirement MAY imply that the context and the scope parameters have to be exchanged. Furthermore, the semantics of these parameters MUST be defined to provide proper channel binding specifications. The definition of exact parameter syntax definition is part of the design of the transport protocol used for the parameter exchange and that may be outside scope of this protocol.

If a key hierarchy is deployed, compromising lower level keys MUST NOT result in a compromise of higher level keys which they were used to derive the lower level keys. The compromise of keys at each level MUST NOT result in compromise of other keys at the same level. The same principle applies to entities that hold and manage a particular key defined in the key hierarchy. Compromising keys on one authenticator MUST NOT reveal the keys of another authenticator. Note that the compromise of higher-level keys has security implications on lower levels.

Guidance on parameters required, caching, storage and deletion procedures to ensure adequate security and authorization provisioning for keying procedures MUST be defined in a solution document.

All the keying material MUST be uniquely named so that it can be managed effectively.

6.2. Key Freshness

As [I-D.housley-aaa-key-mgmt] defines, a fresh key is one that is generated for the intended use. This would mean the key hierarchy MUST provide for creation of multiple cryptographically separate child keys from a root key at higher level. Furthermore, the keying solution needs to provide mechanisms for refreshing each of the keys within the key hierarchy.

6.3. Authentication

Each party in the handover keying architecture MUST be authenticated to any other party with whom it communicates, and securely provide

its identity to any other entity that may require the identity for defining the key scope. The identity provided MUST be meaningful according to the protocol over which the two parties communicate.

6.4. Authorization

The EAP Key management document [I-D.ietf-eap-keying] discusses several vulnerabilities that are common to handover mechanisms. One important issue arises from the way the authorization decisions might be handled at the AAA server during network access authentication. For example, if AAA proxies are involved, they may also influence in the authorization decision. Furthermore, the reasons for making a particular authorization decision are not communicated to the authenticator. In fact, the authenticator only knows the final authorization result. The proposed solution MUST make efforts to document and mitigate authorization attacks.

6.5. Channel Binding

Channel Binding procedures are needed to avoid a compromised intermediate authenticator providing unverified and conflicting service information to each of the peer and the EAP server. In the architecture introduced in this document, there are multiple intermediate entities between the peer and the back-end EAP server. Various keys need to be established and scoped between these parties and some of these keys may be parents to other keys. Hence the channel binding for this architecture will need to consider layering intermediate entities at each level to make sure that an entity with higher level of trust can examine the truthfulness of the claims made by intermediate parties.

6.6. Transport Aspects

Depending on the physical architecture and the functionality of the elements involved, there may be a need for multiple protocols to perform the key transport between entities involved in the handover keying architecture. Thus, a set of requirements for each of these protocols, and the parameters they will carry, MUST be developed. Following the requirement specifications, recommendations will be provided as to whether new protocols or extensions to existing protocols are needed.

As mentioned, the use of existing AAA protocols for carrying EAP messages and keying material between the AAA server and AAA clients that have a role within the architecture considered for the keying problem will be carefully examined. Definition of specific parameters, required for keying procedures and to be transferred over any of the links in the architecture, are part of the scope. The

relation of the identities used by the transport protocol and the identities used for keying also needs to be explored.

7. Use Cases and Related Work

In order to further clarify the items listed in scope of the proposed work, this section provides some background on related work and the use cases envisioned for the proposed work.

7.1. IEEE 802.11r Applicability

One of the EAP lower layers, IEEE 802.11, is in the process of specifying a mechanism to avoid the problem of repeated full EAP exchanges in a limited setting, by introducing a two-level key hierarchy. The EAP authenticator is collocated with what is known as an R0 Key Holder (R0-KH), which receives the MSK from the EAP server. A pairwise master key (PMK-R0) is derived from the last 32 octets of the MSK. Subsequently, the R0-KH derives an PMK-R1 to be handed out to the attachment point of the peer. When the peer moves from one R1-KH to another, a new PMK-R1 is generated by the R0-KH and handed out to the new R1-KH. The transport protocol used between the R0-KH and the R1-KH is not specified.

In some cases, a mobile may seldom move beyond the domain of the R0-KH and this model works well. A full EAP authentication will generally be repeated when the PMK-R0 expires. However, in general cases mobiles may roam beyond the domain of R0-KHs (or EAP authenticators), and the latency of full EAP authentication remains an issue.

Another consideration is that there needs to be a key transfer protocol between the R0-KH and the R1-KH; in other words, there is either a star configuration of security associations between the key holder and a centralized entity that serves as the R0-KH, or if the first authenticator is the default R0-KH, there will be a full-mesh of security associations between all authenticators. This is undesirable.

The proposed work on EAP efficient re-authentication protocol aims at addressing re-authentication in a lower layer agnostic manner that also can fill some of the gaps in IEEE 802.11r.

7.2. CAPWAP Applicability

The IETF CAPWAP WG is developing a protocol between what is termed an Access Controller (AC) and Wireless Termination Points (WTP). The AC and WTP can be mapped to a WLAN switch and Access Point respectively.

The CAPWAP model supports both split and integrated MAC architectures, with the authenticator always being implemented at the AC.

The proposed work on EAP efficient re-authentication protocol addresses an inter-authenticator hand-off problem from an EAP perspective, which applies during hand-off between ACs. Inter-controller hand-offs is a topic yet to be addressed in great detail and the re-authentication work can potentially address it in an effective manner.

7.3. Inter-Technology Hand-Off

EAP is used for access authentication by several technologies and is under consideration for use over several other technologies going forward. Given that, it should be feasible to support smoother hand-offs across technologies. That is one of the big advantages of using a common authentication protocol. Authentication procedures typically add substantial hand-off delays.

An EAP peer that has multiple radio technologies (802.11 and GSM, for instance) must perform the full EAP exchange on each interface upon every horizontal or vertical hand-off. With a method independent EAP efficient re-authentication, it is feasible to support faster hand-offs even in the vertical hand-off cases, when the peer may be roaming from one technology to another.

8. Security Considerations

This document details the HOKEY problem statement. Since HOKEY is an authentication protocol, there are a myriad of security-related issues surrounding its development and deployment.

In this document, we have detailed a variety of security properties inferred from [I-D.housley-aaa-key-mgmt] to which HOKEY must conform, including the management of key context, scope, freshness, and transport; resistance to attacks based on the domino effect; and authentication and authorization. See section Section 6 for further details.

9. IANA Considerations

This document does not introduce any new IANA considerations.

10. References

10.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

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

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

[I-D.ietf-eap-keying] Aboba, B., "Extensible Authentication Protocol (EAP) Key Management Framework", draft-ietf-eap-keying-17 (work in progress), January 2007.

[I-D.housley-aaa-key-mgmt] Housley, R. and B. Aboba, "Guidance for AAA Key Management", draft-housley-aaa-key-mgmt-06 (work in progress), November 2006.

10.2. Informative References

[RFC3990] O'Hara, B., Calhoun, P., and J. Kempf, "Configuration and Provisioning for Wireless Access Points (CAPWAP) Problem Statement", RFC 3990, February 2005.

[RFC4187] Arkko, J. and H. Haverinen, "Extensible Authentication Protocol Method for 3rd Generation Authentication and Key Agreement (EAP-AKA)", RFC 4187, January 2006.

Author's Address

T. Charles Clancy, Editor
DoD Laboratory for Telecommunications Sciences
8080 Greenmead Drive
College Park, MD 20740
USA

Email: clancy@LTSnet.net

Full Copyright Statement

Intellectual Property

Acknowledgment

Funding for the RFC Editor function is provided by the IETF Administrative Support Activity (IASA).