Network Working Group	K. Hoeper, Ed.	
Internet-Draft	Motorola	
Intended status: Standards Track	Y. Ohba, Ed.	
Expires: October 5, 2009	Toshiba	
	April 03, 2009	

TOC

Distribution of EAP based keys for handover and re-authentication draft-ietf-hokey-key-mgm-06

Status of this Memo

Copyright Notice

Abstract

This document describes a mechanism for delivering root keys from an Extensible Authentication Protocol (EAP) server to another network server that requires the keys for offering security protected services, such as re-authentication, to an EAP peer. The distributed root key can be either a usage-specific root key (USRK), a domain-specific root key (DSRK) or a domain-specific usage-specific root key (DSUSRK) that has been derived from an Extended Master Session Key (EMSK) hierarchy previously established between the EAP server and an EAP peer. The document defines a key distribution exchange (KDE) protocol using Remote Authentication Dial In User Service (RADIUS) that can distribute these different types of root keys and discusses its security requirements.

Table of Contents

TOC

1. Introduction

The Extensible Authentication Protocol (EAP) [RFC3748] (Aboba, B., Blunk, L., Vollbrecht, J., Carlson, J., and H. Levkowetz, "Extensible Authentication Protocol (EAP)," June 2004.) is an authentication framework supporting authentication methods that are specified in EAP methods. By definition, any key-generating EAP method derives an Master Session Key (MSK) and an Extended Master Session Key (EMSK). [RFC5295] (Salowey, J., Dondeti, L., Narayanan, V., and M. Nakhjiri, "Specification for the Derivation of Root Keys from an Extended Master Session Key (EMSK)," August 2008.) reserves the EMSK for the sole purpose of deriving root keys that can be used for specific purposes called usages. In particular, [RFC5295] (Salowey, J., Dondeti, L., Narayanan, V., and M. Nakhjiri, "Specification for the Derivation of Root Keys from an Extended Master Session Key (EMSK)," August 2008.) defines how to create a usage-specific root key (USRK) for bootstrapping security in a specific application, a domain-specific root key (DSRK) for bootstrapping security of a set of services within a domain, and a usage-specific DSRK (DSUSRK) for a specific application within a domain.

MSK and EMSK may be used to derive further keying material for a variety of security mechanisms [RFC5247] (Aboba, B., Simon, D., and P. Eronen, "Extensible Authentication Protocol (EAP) Key Management Framework," August 2008.). For example, the MSK has been widely used for bootstrapping the wireless link security associations between the peer and the network attachment points. However, performance as well as security issues arise when using the MSK and the current bootstrapping methods in mobile scenarios that require handovers, as described in [RFC5169] (Clancy, T., Nakhjiri, M., Narayanan, V., and L. Dondeti, "Handover Key Management and Re-Authentication Problem Statement," March 2008.). To address handover latencies and other shortcomings, [RFC5296] (Narayanan, V. and L. Dondeti, "EAP Extensions for EAP Re-authentication Protocol (ERP)," August 2008.) specifies an EAP re-authentication protocol (ERP) that uses keys derived from EMSK or DSRK to enable efficient re-authentications in handover scenarios. [RFC5295] (Salowey, J., Dondeti, L., Narayanan, V., and M. Nakhjiri, "Specification for the Derivation of Root Keys from an Extended Master Session Key (EMSK)," August 2008.) and [RFC5296] (Narayanan, V. and L. Dondeti, "EAP Extensions for EAP Re-authentication Protocol (ERP)," August 2008.) both do not specify how root keys are delivered to the network server requiring the key. Such a key delivery mechanism is essential because the EMSK cannot leave the EAP server ([RFC5295] (Salowey, J., Dondeti, L., Narayanan, V., and M. Nakhjiri, "Specification for the Derivation of Root Keys from an Extended Master Session Key (EMSK)," August 2008.)) but root keys are needed by other network servers disjoint with the EAP server. For example, in order to enable an EAP peer to re-authenticate to a network during a handover, certain root keys need to be made available by the EAP server to the server carrying out the re-authentication.

This document specifies a mechanism for the delivery of EMSK child keys from the server holding the EMSK or a root key to another network server that requests a root key for providing protected services (such as re-authentication and other usage and domain-specific services) to EAP peers. In the remainder of this document, a server delivering root keys is referred to as Key Delivering Server (KDS) and a server authorized to request and receive root keys from a KDS is referred to as Key Requesting Server (KRS). The Key Distribution Exchange (KDE) protocol defined in this document uses RADIUS [RFC2865] (Rigney, C., Willens, S., Rubens, A., and W. Simpson, "Remote Authentication Dial In User Service (RADIUS)," June 2000.), [RFC3579] (Aboba, B. and P. Calhoun, "RADIUS (Remote Authentication Dial In User Service) Support For Extensible Authentication Protocol (EAP)," September 2003.) and has several variants depending on the type of key that is requested and delivered (i.e. DRSK, USRK, and DSUSRK). The document also describes security requirements for the secure key delivery over RADIUS.

---

2. Terminology

TOC

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] (Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels," March 1997.).

USRK: Usage-Specific Root Key. A root key that is derived from the EMSK, see [RFC5295] (Salowey, J., Dondeti, L., Narayanan, V., and M. Nakhjiri, "Specification for the Derivation of Root Keys from an Extended Master Session Key (EMSK)," August 2008.).

USR-KH: USRK Holder. A network server that is authorized to request and receive a USRK from the EAP server. The USR-KH can be an AAA server or dedicated service server.

DSRK: Domain-Specific Root Key. A root key that is derived from the EMSK, see [RFC5295] (Salowey, J., Dondeti, L., Narayanan, V., and M. Nakhjiri, "Specification for the Derivation of Root Keys from an Extended Master Session Key (EMSK)," August 2008.).

DSR-KH: DSRK Holder. A network server that is authorized to request and receive a DSRK from the EAP server. The most likely implementation of a DSR-KH is an AAA server in a domain, enforcing the policies for the usage of the DSRK within this domain.

DSUSRK: Domain-Specific Usage-Specific Root Key. A root key that is derived from the DSRK, see [RFC5295] (Salowey, J., Dondeti, L.,

Narayanan, V., and M. Nakhjiri, “Specification for the Derivation of Root Keys from an Extended Master Session Key (EMSK),” August 2008.).

DSUSR-KH: DSUSRK holder. A network server authorized to request and receive a DSUSRK from the DSR-KH. The most likely implementation of a DSUSR-KH is an AAA server in a domain, responsible for a particular service offered within this domain.

RK: Root Key. An EMSK child key, i.e. a USRK, DSRK, or DSUSRK.

KDS: Key Delivering Server. A network server that holds an EMSK or DSRK and delivers root keys to KRS requesting root keys. The EAP server and DSR-KH can act as KDS.

KRS: Key Requesting Server. A network server that shares an interface with a KDS and is authorized to request root keys from the KDS. USR-KH, DSR-KH, and DSUSR-KH can all act as KRS.

---

3. Key Delivery Architecture

TOC

An EAP server carries out the EAP authentications with EAP peers but is typically not making any, potentially future, service authorization decisions involving peers. Authorizations as well as the service provisioning are handled by the respective network server offering the requested service. These servers can be AAA servers or other service servers. Whenever EAP-based keying material is used to protect a requested service, a network server needs to request the root key associated with the offered service from the respective KDS. This kind of key requests and distributions are necessary because an EMSK cannot leave the EAP server ([RFC5295] (Salowey, J., Dondeti, L., Narayanan, V., and M. Nakhjiri, “Specification for the Derivation of Root Keys from an Extended Master Session Key (EMSK),” August 2008.)). Hence, any root key that is directly derived from an EMSK must be derived and delivered by the EAP server itself, whereas root keys derived from EMSK child keys, such as a DSUSRK, can be requested from the respective root key holder. Hence, a KDS can be either the EAP server or a DSRK holder (DSR-KH), whereas a KRS can be either a USRK holder (USR-KH), a DSR-KH or a DSUSRK holder (DSUSR-KH).

The KRS needs to share an interface with the KDS to be able to send all necessary input data to derive the requested key and to receive the requested key. The provided data includes the Key Derivation Function (KDF) that should be used to derive the requested key. The KRS uses the received root key to derive further keying material in order to secure its offered services. Every KDS is responsible for storing and protecting the received root key as well as the derivation and

distribution of any child key derived from the root key. An example of a key delivery architecture is illustrated in Figure 1 (Example Key Delivery Architecture for the Different KRS and KDS) showing the different types of KRS and their interfaces to the KDS.

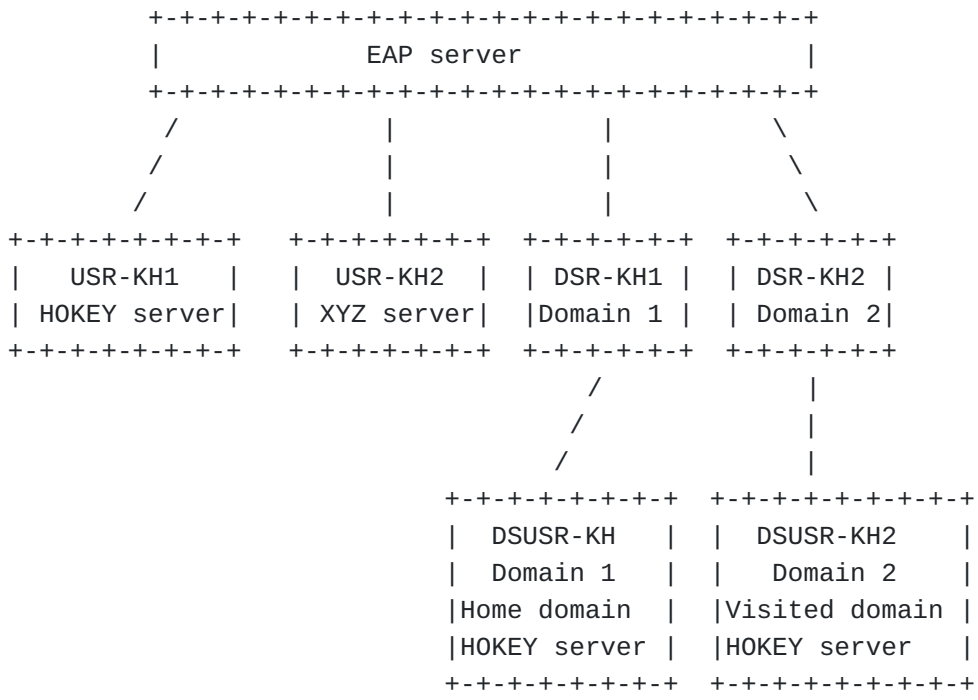


Figure 1: Example Key Delivery Architecture for the Different KRS and KDS

4. Key Distribution Exchange (KDE)

TOC

In this section, a generic mechanism for a key distribution exchange (KDE)over RADIUS is described in which a root key (RK) is distributed from a KDS to a KRS. It is required that the communication path between the KDS and the KRS is protected by the use of an appropriate RADIUS transport security mechanism (see Section 8 (Security Considerations)). Here, it is assumed that the KRS and the KDS are separate entities, logically if not physically, and the delivery of the requested RK is specified accordingly.

The key distribution exchange consists of one roundtrip, i.e. two messages between the KRS and the KDS, as illustrated in Figure 2 (KDE Message Flow). First, the KRS sends a KDE-Request consisting of a

RADIUS Access-Request message with a KDE attribute in which the K-flag is cleared. As a response, the KDS sends a KDE-Response consisting of a RADIUS Access-Accept message with a KDE attribute in which the K-flag set. The RADIUS KDE attribute used in this exchange is defined in Section 5 (RADIUS KDE Attribute).

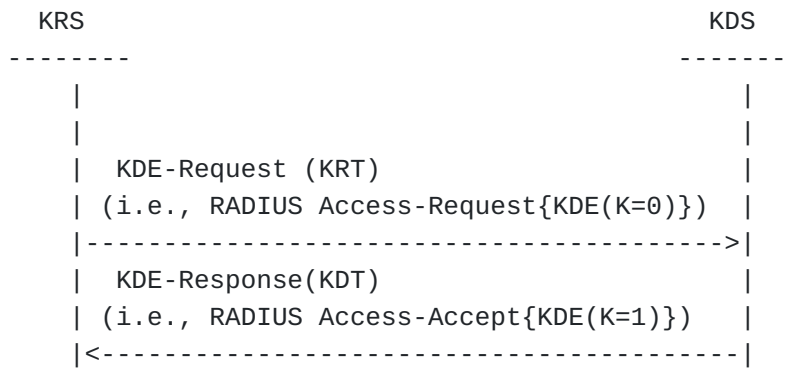


Figure 2: KDE Message Flow

KDE-Request: The KRS sends a Key Request Token (KRT) to the KDS. The contents of KRT are detailed below.

KDE-Response: As a response, the KDS sends the requested RK to the KRS wrapped inside a token called Key Delivery Token (KDT). The contents of KDT are detailed below.

KRT : (PID, KT, KL)
KRT carries the identifiers of the peer (PID), the key type (KT) and the key label (KL). See [RFC5295] (Salowey, J., Dondeti, L., Narayanan, V., and M. Nakhjiri, "Specification for the Derivation of Root Keys from an Extended Master Session Key (EMSK)," August 2008.) for the specification of key labels.

KDT : (KT, KL, RK, KN_RK, LT_RK)
KDT carries the root key (RK) to be distributed to the KRS, as well as the key type (KT), the key label (KL), the key name (KN_RK) and the lifetime of RK (LT_RK).

TOC

4.1. Context and Scope for Distributed Keys

The key lifetime of each distributed key MUST NOT be greater than that of its parent key.
The key context of each distributed key is determined by the sequence of KTs in the key hierarchy. When a DSRK is being delivered and the DSRK applies to only a specific set of services, the service types may need to be carried as part of context for the key. Carrying such a specific set of services is outside the scope of this document.
The key scope of each distributed key is determined by the sequence of (PID, KT, KL)-tuples in the key hierarchy. The KDF used to generate the requested keys includes context and scope information, thus, binding the key to the specific channel [RFC5295] (Salowey, J., Dondeti, L., Narayanan, V., and M. Nakhjiri, “Specification for the Derivation of Root Keys from an Extended Master Session Key (EMSK),” August 2008.).

4.2. Key Distribution Exchange Scenarios

TOC

Given the three types of KRS, there are three scenarios for the distribution of EMSK child keys. For all scenarios, the trigger and mechanism for key delivery may involve a specific request from an EAP peer and/or another intermediary (such as an authenticator). For simplicity, it is assumed that USR-KHs reside in the same domain as the EAP server.

> **Scenario 1: EAP server to USR-KH:** In this scenario, the EAP server delivers a USRK to a USR-KH.
>
> **Scenario 2: EAP server to DSR-KH:** In this scenario, the EAP server delivers a DSRK to a DSR-KH.
>
> **Scenario 3: DSR-KH to DSUSR-KH:** In this scenario, a DSR-KH in a specific domain delivers keying material to a DSUSR-KH in the same domain.

The key distribution exchanges for Scenario 3 can be combined with the key distribution exchanges for Scenario 2 into a single roundtrip exchange as shown in Figure 3 (Combined Message Exchange). Here, KDE-Request and KDE-Response are messages for Scenarios 2, whereas KDE-Request' and KDE-Response' are messages for Scenarios 3.

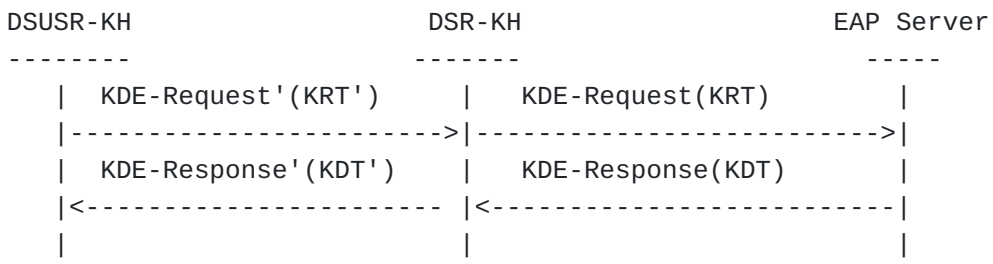


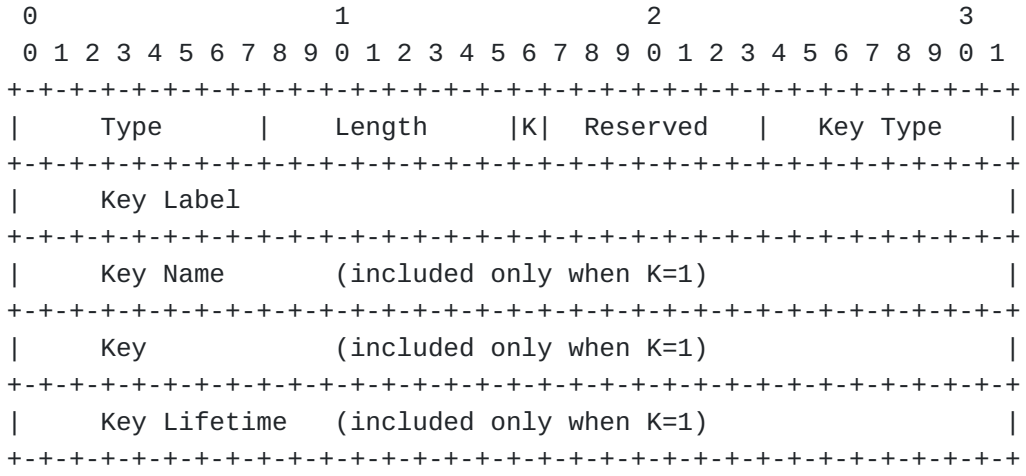
Figure 3: Combined Message Exchange

---

5. RADIUS KDE Attribute

TOC

This section defines the format of the RADIUS KDE attribute. See Section 8 (Security Considerations) for security requirements on transporting this RADIUS attribute.



Type
= X (KDE) [X to be assigned by IANA].

Length
>4

K (Key included)
A flag to indicate whether this attribute contains a Key field. This flag is set for a KDE-Response. This flag is cleared for a KDE-Request.

Reserved

Reserved bits. All reserved bits MUST be set to 0 by the sender and ignored by the recipient.

Key Type

A field to contain a KT. The following KT values are defined: 0 (DSRK), 1 (USRK) and 2 (DSUSRK).

Key Label

A field to contain a key label (KL). The first octet contains the length of the rest of this field in octets.

Key Name

A field to contain a KN_RK. The first octet contains the length of the rest of this field in octets. This field is contained if and only if K-flag is set.

Key

A field to contain a RK. The first octet contains the length of the rest of this field in octets. This field is contained if and only if K-flag is set.

Key Lifetime

A 4-octet unsigned integer to indicate a LT_RK. This field is contained if and only if K-flag is set.

---

6. KDE used in the EAP Re-authentication Protocol (ERP)

TOC

This section describes how the presented KDE should be used to request and deliver the root keys used for re-authentication in the EAP Re-authentication Protocol (ERP) defined in [RFC5296] (Narayanan, V. and L. Dondeti, “EAP Extensions for EAP Re-authentication Protocol (ERP),” August 2008.). ERP supports two forms of bootstrapping, implicit as well as explicit bootstrapping, and KDE is discussed for both cases in the remainder of this section.

In implicit bootstrapping the local EAP Re-authentication (ER) server requests the DSRK from the home AAA server during the initial EAP exchange. Here, the local ER server acts as the KRS and the home AAA server as the KDS. In this case, the local ER server requesting the DSRK MUST include a KDE attribute with the K-flag cleared in the RADIUS Access-Request message that carries the first EAP-Response message from the peer. A value of the RADIUS User-Name attribute is used as the PID. Upon receiving a valid KDE-Request, the home AAA server includes a KDE attribute with K-flag set in the RADIUS Access-Accept message that carries the EAP-Success message.

Explicit bootstrapping is initiated by a peer if it doesn't know the domain. Here, EAP-Initiate and EAP-Finish messages are exchanged between the peer and the home AAA server, with the bootstrapping flag in the EAP-Initiate message set. In this case, the local ER server (acting as KRS) MUST include a KDE attribute with the K-bit cleared in a RADIUS Access-Request message that carries an EAP-Initiate message with the bootstrapping flag turned on. A value of the RADIUS User-Name attribute is used as the PID. In its response, the home AAA server (acting as KDS) MUST include a KDE attribute with K-flag set in a RADIUS Access-Accept message that carries an EAP-Finish message for which the bootstrapping flag is set.

---

7. Conflicting Messages

TOC

In addition to the rules specified in Section 2.6.3. of [RFC3579] (Aboba, B. and P. Calhoun, “RADIUS (Remote Authentication Dial In User Service) Support For Extensible Authentication Protocol (EAP),” September 2003.), the following combinations SHOULD NOT be sent by a RADIUS Server:

```
Access-Accept/EAP-Message/EAP-Finish with 'R' flag set to 1
Access-Reject/EAP-Message/EAP-Finish with 'R' flag set to 0
Access-Reject/Keying-Material
Access-Reject/KDE
Access-Challenge/EAP-Message/EAP-Initiate
Access-Challenge/EAP-Message/EAP-Finish
Access-Challenge/KDE
```

---

8. Security Considerations

TOC

This section provides security requirements and an analysis on transporting EAP keying material using RADIUS.

8.1. Requirements on RADIUS Key Transport

TOC

RADIUS messages that carry a KDE attribute MUST be encrypted, integrity-protected and replay-protected with a security association created by a RADIUS transport protocol such as TLS [I-D.ietf-radext-radsec] (Winter, S., McCauley, M., Venaas, S., and K. Wierenga, “TLS encryption for RADIUS,” March 2010.). When there is an

intermediary such as a RADIUS proxy on the path between the KRS and the KDS, there will be a series of hop-by-hop security associations along the path. The use of hop-by-hop security associations implies that the intermediary on each hop can access the distributed keying material. Hence the use of hop-by-hop security SHOULD be limited to an environment where an intermediary is trusted not to abuse the distributed key material.

8.2. Distributing RK without Peer Consent

TOC

When a KDE-Request message is sent as a result of explicit ERP bootstrapping [RFC5296] (Narayanan, V. and L. Dondeti, "EAP Extensions for EAP Re-authentication Protocol (ERP)," August 2008.), cryptographic verification of peer consent on distributing a RK is provided by the integrity checksum of the EAP-Initiate message with the bootstrapping flag turned on.
When a KDE-Request message is sent as a result of implicit ERP bootstrapping [RFC5296] (Narayanan, V. and L. Dondeti, "EAP Extensions for EAP Re-authentication Protocol (ERP)," August 2008.), cryptographic verification of peer consent on distributing a RK is not provided. As a result, it is possible for a KRS to request a RK from the home server and obtain the RK even if the peer does not support ERP, which can lead to an unintended use of a RK and failed authentication attempts.

---

9. IANA consideration

TOC

This document defines a new namespace for maintaining Key Type used to identify the type of the root key RK. The range of values 0 - 255 are for permanent, standard message types, allocated by IETF Review [IANA] (Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs," May 2008.). This document defines the values 0 (DSRK), 1 (USRK) and 2 (DSUSRK).
This document defines a new RADIUS Attribute Type for KDE in Section 5 (RADIUS KDE Attribute).

---

10. Acknowledgements

TOC

The author would like to thank Dan Harkins, Chunqiang Li, Rafael Marin Lopez and Charles Clancy for their valuable comments.

---

11. Contributors

TOC

The following people contributed to this document.

Madjid Nakhjiri (madjid.nakhjiri@motorola.com)

Kedar Gaonkar (kgaonkar3@gatech.edu)

Lakshminath Dondeti (ldondeti@qualcomm.com)

Vidya Narayanan (vidyan@qualcomm.com)

Glen Zorn (glenzorn@comcast.net)

---

12. References

TOC

12.1. Normative References

TOC

[RFC2119]	Bradner, S., “Key words for use in RFCs to Indicate Requirement Levels,” BCP 14, RFC 2119, March 1997 (TXT, HTML, XML).
[RFC2865]	Rigney, C., Willens, S., Rubens, A., and W. Simpson, “Remote Authentication Dial In User Service (RADIUS),” RFC 2865, June 2000 (TXT).
[RFC3748]	Aboba, B., Blunk, L., Vollbrecht, J., Carlson, J., and H. Levkowetz, “Extensible Authentication Protocol (EAP),” RFC 3748, June 2004 (TXT).
[RFC5295]	Salowey, J., Dondeti, L., Narayanan, V., and M. Nakhjiri, “Specification for the Derivation of Root Keys from an Extended Master Session Key (EMSK),” RFC 5295, August 2008 (TXT).
[RFC5296]	Narayanan, V. and L. Dondeti, “EAP Extensions for EAP Re-authentication Protocol (ERP),” RFC 5296, August 2008 (TXT).
[IANA]	Narten, T. and H. Alvestrand, “Guidelines for Writing an IANA Considerations Section in RFCs,” BCP 26, RFC 5226, May 2008.

12.2. Informative references

TOC

[RFC3579]	Aboba, B. and P. Calhoun, “RADIUS (Remote Authentication Dial In User Service) Support For Extensible Authentication Protocol (EAP),” RFC 3579, September 2003 (TXT).
[RFC5247]	Aboba, B., Simon, D., and P. Eronen, “Extensible Authentication Protocol (EAP) Key Management Framework,” RFC 5247, August 2008 (TXT).
[RFC5169]	Clancy, T., Nakhjiri, M., Narayanan, V., and L. Dondeti, “Handover Key Management and Re-Authentication Problem Statement,” RFC 5169, March 2008 (TXT).
[I-D.ietf-radext-radsec]	Winter, S., McCauley, M., Venaas, S., and K. Wierenga, “TLS encryption for RADIUS,” draft-ietf-radext-radsec-06 (work in progress), March 2010 (TXT).

Authors' Addresses

TOC

	Katrin Hoeper (editor)
	Motorola
	1301 E Algonquin Road
	Schaumburg, IL 60196
	USA
Phone:	+1 847 576 4714
Email:	khoeper@motorola.com
	Yoshihiro Ohba (editor)
	Toshiba America Research, Inc.
	1 Telcordia Drive
	Piscataway, NJ 08854
	USA
Phone:	+1 732 699 5305
Email:	yohba@tari.toshiba.com