Network Working Group	J. Salowey
Internet-Draft	Cisco Systems
Updates: eap-keying (RFC Ed to	L. Dondeti
replace this with RFC number)	V. Narayanan
(if approved)	Qualcomm, Inc
Intended status: Standards Track	M. Nakhjiri
Expires: December 25, 2008	Motorola
	June 23, 2008

TOC

Specification for the Derivation of Root Keys from an Extended Master Session Key (EMSK)
draft-ietf-hokey-emsk-hierarchy-07

Status of this Memo

Abstract

The Extensible Authentication Protocol (EAP) defined the Extended Master Session Key (EMSK) generation, but reserved it for unspecified future uses. This memo reserves the EMSK for the sole purpose of deriving root keys. Root keys are are master keys that can be used for multiple purposes, identified by usage definitions. This document also specifies a mechanism for avoiding conflicts between root keys by deriving them in a manner that guarantee cryptographic separation. Finally, this document also defines one such root key usage: domain specific root keys are root keys made available to and used within specific key management domains.

Table of Contents

1. Introduction

TOC

This document deals with keys generated by authenticated key exchange mechanisms defined within the EAP framework [RFC3748] (Aboba, B., Blunk, L., Vollbrecht, J., Carlson, J., and H. Levkowetz, “Extensible Authentication Protocol (EAP),” June 2004.). EAP defines two types of keying material; a Master Session Key (MSK) and an Extended Master Session Key (EMSK). The EAP specification implicitly assumes that the MSK produced by EAP will be used for a single purpose at a single device, however it does reserve the EMSK for future use. This document

defines the EMSK to be used solely for deriving root keys using the key derivation specified. The root keys are meant for specific purposes called usages; a special usage class is the domain specific root keys made available to and used within specific key management domains. This document also provides guidelines for creating usage definitions for the various uses of EAP key material and for the management of the root keys. In this document, the terms application and usage (or "usage definition") refer to a specific use case of the EAP keying material. Different uses for keys derived from the EMSK have been proposed. Some examples include hand off across access points in various mobile technologies, mobile IP authentication and higher layer application authentication. In order for a particular usage of EAP key material to make use of this specification it must specify a so-called usage definition. This document does not define how the derived Usage Specific Root Keys (USRK) are used, see the following section for discussion of applicable usages. It does define a framework for the derivation of USRKs for different purposes such that different usages can be developed independently from one another. The goal is to have security properties of one usage have minimal or no effect on the security properties of other usages.

This document does define a special class of USRK, called a Domain Specific Root Key (DSRK) for use in deriving keys specific to a key management domain. Each DSRK is a root key used to derive Domain Specific Usage Specific Root Keys (DSUSRK). The DSUSRKs are USRKs specific to a particular key management domain.

In order to keep root keys for specific purposes separate from one another, two requirements are defined in the following sections. One is coordinated key derivation and another is cryptographic separation.

1.1. Applicable usages of keys derived from the EMSK

TOC

The EMSK is typically established as part of network access authentication and authorization. It is expected that keys derived from EMSK will be used in protocols related to network access, such as handover optimizations, and the scope of these protocols is usually restricted to the endpoints of the lower layers over which EAP packets are sent.

In particular, it is inappropriate for the security of higher layer applications to solely rely on keys derived from network access authentication. Even when used together with another, independent security mechanism, the use of these keys needs to be carefully evaluated with regards to the benefits of the optimization and the need to support multiple solutions. Performance optimizations may not warrant the close tie-in that may be required between the layers in order to use EAP-based keys. Such optimizations may be offset by the complexities of managing the validity and usage of key materials. Keys

generated from subsequent EAP authentications may be beyond the knowledge and control of applications.
From an architectural point of view, applications should not make assumptions about the lower layer technology (such as network access authentication) used on any particular hop along the path between the application endpoints.
From a practical point of view, making such assumptions would complicate using those applications over lower layers that do not use EAP, and make it more difficult for applications and network access technologies to evolve independently of each other.
Parties using keys derived from EMSK also need trust relationships with the EAP endpoints, and mechanisms for securely communicating the keys.
For most applications, it is not appropriate to assume that all current and future access networks are trusted to secure the application function. Instead, applications should implement the required security mechanisms in access independent manner.
Implementation considerations may also complicate communication of keys to an application from the lower layer. For instance, in many configurations applications may run on a different device than the one providing EAP-based network access to it.
Given all this, it is NOT RECOMMENDED to use keys derived from the EMSK as an exclusive security mechanism, when their usage is not inherently, and by permanent nature, tied to the lower layer where network access authentication was performed.
Keys derived from EAP are pairwise by nature and are not directly suitable for multicast or other group usages such as those involved in some routing protocols. It is possible to use keys derived from EAP in protocols that distribute group keys to group participants. The definition of these group key distribution protocols is beyond the scope of this document and would require additional specification.

1.2. 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.)
The following terms are taken from [RFC3748] (Aboba, B., Blunk, L., Vollbrecht, J., Carlson, J., and H. Levkowetz, “Extensible Authentication Protocol (EAP),” June 2004.): EAP Server, peer, authenticator, Master Session Key (MSK), Extended Master Session Key (EMSK), Cryptographic Separation.
Usage Definition

> An application of cryptographic key material to provide one or more security functions such as authentication, authorization, encryption or integrity protection for related applications or services. This document provides guidelines and recommendations for what should be included in usage definitions. This document does not place any constraints on the types of use cases or services that create usage definitions.

Usage Specific Root Key (USRK)

> Keying material derived from the EMSK for a particular usage definition. It is used to derive child keys in a way defined by its usage definition.

Key Management Domain

> A key management domain is specified by the scope of a given root key. The scope is the collection of systems authorized to access key material derived from that key. Systems within a key management domain may be authorized to (1) derive key materials, (2) use key materials, or (3) distribute key materials to other systems in the same domain. A derived key's scope is constrained to a subset of the scope of the key it is derived from. In this document the term domain refers to a key management domain unless otherwise qualified.

Domain Specific Root Key (DSRK)

> Keying material derived from the EMSK that is restricted to use in a specific key management domain. It is used to derive child keys for a particular usage definition. The child keys derived from a DSRK are referred to as domain specific usage specific root keys (DSUSRK). DSUSRKs are similar to the USRK, except in the fact that their scope is restricted to the same domain as the parent DSRK from which it is derived.

---

2. Cryptographic Separation and Coordinated Key Derivation

TOC

The EMSK is used to derive keys for multiple use cases, and thus it is required that the derived keys are cryptographically separate. Cryptographic separation means that when multiple keys are derived from an EMSK, given any derived key it is computationally infeasible to derive any of the other derived keys. Note that deriving the EMSK from any combinations of the derived keys must also be computationally infeasible. In practice this means that derivation of an EMSK from a derived key or derivation of one child key from another must require an

amount of computation equivalent to that required to, say, reversing a cryptographic hash function.

Cryptographic separation of keys derived from the same key can be achieved in many ways. Two obvious methods are as follows: it is plausible to use the IKEv2 PRF [RFC4306] (Kaufman, C., "Internet Key Exchange (IKEv2) Protocol," December 2005.) on the EMSK and generate a key stream. Keys of various lengths may be provided as required from the key stream for various uses. The other option is to derive keys from EMSK by providing different inputs to the PRF. However, it is desirable that derivation of one child key from the EMSK is independent of derivation of another child key. This allows child keys to be derived in any order, independent of other keys. Thus it is desirable to use the second option from above. That implies the additional input to the PRF must be different for each child key derivation. This additional input to the PRF must be coordinated properly to meet the requirement of cryptographic separation and to prevent reuse of key material between usages.

If cryptographic separation is not maintained then the security of one usage depends upon the security of all other usages that use key derived from the EMSK. If a system does not have this property then a usage's security depends upon all other usages deriving keys from the same EMSK, which is undesirable. In order to prevent security problems in one usage from interfering with another usage, the following cryptographic separation is required:

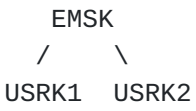
- It MUST be computationally infeasible to compute the EMSK from any root key derived from it.
- Any root key MUST be cryptographically separate from any other root key derived from the same EMSK or DSRK
- Derivation of USRKs MUST be coordinated so that two separate cryptographic usages do not derive the same key.
- Derivation of DSRKs MUST be coordinated so that two separate key management domains do not derive the same key.
- Derivation of DSRKs and USRKs MUST be specified such that no domain can obtain a USRK by providing a domain name identical to a Usage Key Label.

This document provides guidelines for a key derivation mechanism, which can be used with existing and new EAP methods to provide cryptographic separation between usages of EMSK. This allows for the development of new usages without cumbersome coordination between different usage definitions.

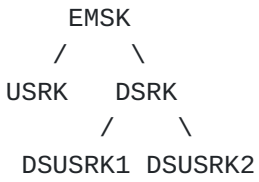
TOC

3. EMSK Key Root Derivation Framework

The EMSK key derivation framework provides a coordinated means for generating multiple root keys from an EMSK. Further keys may then be derived from the root key for various purposes, including encryption, integrity protection, entity authentication by way of proof of possession, and subsequent key derivation. A root key is derived from the EMSK for specific set of uses set forth in a usage definition described in Section 5 (Requirements for Usage Definitions).
The basic EMSK root key hierarchy looks as follows:



This document defines how to derive usage specific root keys (USRK) from the EMSK and also defines a specific USRK called a domain specific root key (DSRK). DSRK are root keys restricted to use in a particular key management domain. From the DSRK, usage specific root keys for a particular application may be derived (DSUSRK). The DSUSRKs are equivalent to USRKs that are restricted to use in a particular domain. The details of lower levels of key hierarchy are outside scope of this document. The key hierarchy looks as follows:



3.1. USRK Derivation

TOC

The EMSK Root Key derivation function (KDF) derives a USRK from the EMSK, a key label, optional data, and output length. The KDF is expected to give the same output for the same input. The basic key derivation function is given below.

```
USRK = KDF(EMSK, key label | "\0" | optional data | length)
```

Where:

| denotes concatenation
"\0" is a NULL octet (0x00 in hex)
length is a 2 octet unsigned integer in network byte order

The key labels are printable ASCII strings unique for each usage definition and are a maximum of 255 octets. In general they are of the form label-string@specorg where specorg is the organization that controls the specification of the usage definition of the Root Key. The key label is intended to provide global uniqueness. Rules for the allocation of these labels are given in Section 8 (IANA Considerations).

The NULL octet after the key label is used to avoid collisions if one key label is a prefix of another label (e.g. "foobar" and "foobarExtendedV2"). This is considered a simpler solution than requiring a key label assignment policy that prevents prefixes from occurring.

For the optional data the KDF MUST be capable of processing at least 2048 opaque octets. The optional data must be constant during the execution of the KDF. Usage definitions MAY use the EAP session-ID [I-D.ietf-eap-keying] (Aboba, B., Simon, D., and P. Eronen, "Extensible Authentication Protocol (EAP) Key Management Framework," November 2007.) in the specification of the optional data parameter that go into the KDF function. This provides the advantage of providing data into the key derivation that is unique to the session that generated the keys.

The KDF must be able to process input keys of up to 256 bytes. It may do this by providing a mechanism for "hashing" long keys down to a suitable size that can be consumed by the underlying derivation algorithm.

The length is a 2-octet unsigned integer in network byte order of the output key length in octets. An implementation of the KDF MUST be capable of producing at least 2048 octets of output, however it is RECOMMENDED that Root Keys be at least 64 octets long.

A usage definition requiring derivation of a Root Key must specify all the inputs (other than EMSK) to the key derivation function.

USRKs MUST be at least 64 octets in length.

3.1.1. On the KDFs

TOC

This specification allows for the use of different KDFs. However, in order to have a coordinated key derivation function the same KDF function MUST be used for all key derivations for a given EMSK. If no KDF is specified, then the default KDF specified in Section 3.1.2 (Default KDF) MUST be used. A system may provide the capability to negotiate additional KDFs. KDFs are assigned numbers through IANA following the policy set in section Section 8 (IANA Considerations).

The rules for negotiating a KDF are as follows:

* If no other KDF is specified the KDF specified in this document MUST be used. This is the "default" KDF.

*The initial authenticated key exchange MAY specify a favored KDF. For example an EAP method may define a preferred KDF to use in its specification. If the initial authenticated key exchange specifies a KDF then this MUST override the default KDF.

Note that usage definitions MUST NOT concern themselves with the details of the KDF construction or the KDF selection, they only need to worry about the inputs specified in Section 3 (EMSK Key Root Derivation Framework).

3.1.2. Default KDF

TOC

The default KDF for deriving root keys from an EMSK is taken from the PRF+ key expansion specified in [RFC4306] (Kaufman, C., “Internet Key Exchange (IKEv2) Protocol,” December 2005.) based on HMAC-SHA-256 [SHA256] (National Institute of Standards and Technology, “Secure Hash Standard,” August 2002.). The PRF+ construction was chosen because of its simplicity and efficiency over other mechanisms such as those used in [RFC4346] (Dierks, T. and E. Rescorla, “The Transport Layer Security (TLS) Protocol Version 1.1,” April 2006.). The motivation for the design of PRF+ is described in [SIGMA] (Krawczyk, H., “SIGMA: the 'SIGn-and-MAc' Approach to Authenticated Diffie-Hellman and its Use in the IKE Protocols,” 2003.). The definition of PRF+ from [RFC4306] (Kaufman, C., “Internet Key Exchange (IKEv2) Protocol,” December 2005.)is given below:

```
PRF+ (K,S) = T1 | T2 | T3 | T4 | ...
```

Where:

```
T1 = PRF (K, S | 0x01)
T2 = PRF (K, T1 | S | 0x02)
T3 = PRF (K, T2 | S | 0x03)
T4 = PRF (K, T3 | S | 0x04)
```

continuing as needed to compute the required length of key material. The key, K, is the EMSK and S is the concatenation of key label, the NULL octet, optional data and length defined in Section 3.1 (USRK Derivation). For this specification the PRF is taken as HMAC-SHA-256 [SHA256] (National Institute of Standards and Technology, “Secure Hash Standard,” August 2002.). Since PRF+ is only defined for 255 iterations it may produce up to 8160 octets of key material.

TOC

3.2. EMSK and USRK Name Derivation

The EAP keying framework [I-D.ietf-eap-keying] (Aboba, B., Simon, D., and P. Eronen, "Extensible Authentication Protocol (EAP) Key Management Framework," November 2007.) specifies that the EMSK MUST be named using the EAP Session-Id and a binary or textual indication. Following that requirement, the EMSK name SHALL be derived as follows:

```
EMSKname = KDF ( EAP Session-ID, "EMSK" | "\0" | length )
```

Where:

```
| denotes concatenation
"EMSK" consists of the 4 ASCII values for the letters
"\0" = is a NULL octet (0x00 in hex)
length is the 2 octet unsigned integer 8 in network byte order
```

It is RECOMMENDED that all keys derived from the EMSK are referred to by the EMSKname and the context of the descendant key usage. This is the default behavior. Any exceptions SHALL be signaled by individual usages.
USRKs MAY be named explicitly with a name derivation specified as follows:

```
USRKName =
     KDF(EAP Session-ID, key label|"\0"|optional data|length)
```

Where:

```
key label and optional data MUST be the same as those used
  in the corresponding USRK derivation
length is the 2 octet unsigned integer 8 in network byte order
```

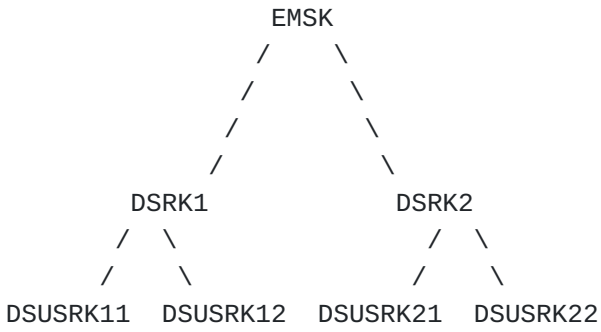
USRKName derivation and usage is applicable when there is ambiguity in the referencing the keys using the EMSKname and the associated context of the USRK usage. The usage SHALL signal such an exception in key naming, so both parties know the key name used.

---

4. Domain Specific Root Key Derivation

TOC

A specific USRK called a Domain Specific Root Key (DSRK) is derived from the EMSK for a specific set of usages in a particular key management domain. Usages derive specific keys for specific services from this DSRK. The DSRK may be distributed to a key management domain for a specific set of usages so keys can be derived within the key management domain for those usages. DSRK based usages will follow a key hierarchy similar to the following:



The DSRK is a USRK with a key label of "dsrk@ietf.org" and the optional data containing a domain label. The optional data MUST contain an ASCII string representing the key management domain that the root key is being derived for. The DSRK MUST be at least 64 octets long.
Domain Specific Usage Specific Root Keys (DSUSRK) are derived from the DSRK. The KDF is expected to give the same output for the same input. The basic key derivation function is given below.

```
DSUSRK = KDF(DSRK, key label | "\0" | optional data | length)
```

The key labels are printable ASCII strings unique for each usage definition within a DSRK usage and are a maximum of 255 octets. In general they are of the form label-string@specorg where specorg is the organization that controls the specification of the usage definition of the DSRK. The key label is intended to provide global uniqueness. Rules for the allocation of these labels are given in Section 8 (IANA Considerations). For the optional data the KDF MUST be capable of processing at least 2048 opaque octets. The optional data must be constant during the execution of the KDF. The length is a 2-octet unsigned integer in network byte order of the output key length in octets. An implementation of the KDF MUST be capable of producing at least 2048 octets of output, however it is RECOMMENDED that DSUSRKs be at least 64 octets long.
Usages that make use of the DSRK must define how the peer learns the domain label to use in a particular derivation. A multi-domain usage must define how both DSRKs and specific DSUSRKs are transported to different key management domains. Note that usages may define alternate ways to constrain specific keys to particular key management domains.
To facilitate the use of EMSKname to refer to keys derived from DSRKs, EMSKname SHOULD be sent along with the DSRK. The exception is when a DSRKname is expected to be used. The usage SHALL signal such an exception in key naming, so both parties know the key name used.
DSUSRKs MAY be named explicitly with a name derivation specified as follows:

```
DSUSRKName =
     KDF(EMSKName,key label | "\0" | optional data | length)
```

where length is the 2 octet unsigned integer 8 in network byte order.

4.1. Applicability of Multi-Domain usages

TOC

When a DSRK is distributed to a domain the domain can generate any DSUSRKs it wishes. This keys can be used to authorize entities in a domain to perform specific functions. In cases where it is appropriate for only a specific domain to be authorized to perform a function the usage SHOULD NOT be defined as multi-domain.
In some cases only certain domains are authorized for a particular Multi-Domain usage. In this case domains that do not have full authorization should not receive the DSRK and should only receive DSUSRKs for the usages which they are authorized. If it is possible for a peer to know which domains are authorized for a particular usage without relying on restricting access to the DSRK to specific domains then this recommendation may be relaxed.

---

5. Requirements for Usage Definitions

TOC

In order for a usage definition to meet the guidelines for USRK usage it must meet the following recommendations:

- The usage must define if it is a domain enabled usage.

- The usage definition MUST NOT use the EMSK in any other way except to derive Root Keys using the key derivation specified in Section 3 (EMSK Key Root Derivation Framework) of this document. They MUST NOT use the EMSK directly.

- The usage definition SHOULD NOT require caching of the EMSK. It is RECOMMENDED that the Root Key derived specifically for the usage definition rather than the EMSK should be used to derive child keys for specific cryptographic operations.

- Usage definition MUST define distinct key labels and optional data used in the key derivation described in Section 3 (EMSK Key Root Derivation Framework). Usage definitions are encouraged to use the key name described in Section 3.2 (EMSK and USRK Name Derivation) and include additional data in the optional data to provide additional entropy.

- Usage definitions MUST define the length of their Root Keys. It is RECOMMENDED that the Root Keys be at least as long as the EMSK (at least 64 octets).

*Usage definitions MUST define how they use their Root Keys. This includes aspects of key management covered in the next section on Root Key Management guidelines.

*

5.1. Root Key Management Guidelines

TOC

This section makes recommendations for various aspects of key management of the Root Key including lifetime, child key derivation, caching and transport.

It is RECOMMENDED that the Root Key is only used for deriving child keys. A usage definition must specify how and when the derivation of child keys should be done. It is RECOMMENDED that usages following similar considerations for key derivation are as outlined in this document for the Root Key derivation with respect to cryptographic separation and key reuse. In addition, usages should take into consideration the number of keys that will be derived from the Root Key and ensure that enough entropy is introduced in the derivation to support this usage. It is desirable that the entropy is provided by the two parties that derive the child key.

Root Keys' lifetimes should not be more than that of the EMSK. Thus, when the EMSK expires, the Root Keys derived from it should be removed from use. If a new EMSK is derived from a subsequent EAP transaction then a usage implementation should begin to use the new Root Keys derived from the new EMSK as soon as possible. Whether or not child keys associated with a Root Key are replaced depends on the requirements of the usage definition. It is conceivable that some usage definition forces the child key to be replaced and others allow child keys to be used based on the policy of the entities that use the child key.

Recall that the EMSK never leaves the EAP peer and server. That also holds true for some Root Keys; however, some Root Keys may be provided to other entities for child key derivation and delivery. Each usage definition specification will specify delivery caching and/or delivery procedures. Note that the purpose of the key derivation in Section 3 (EMSK Key Root Derivation Framework) is to ensure that Root Keys are cryptographically separate from each other and the EMSK. In other words, given a Root Key, it is computationally infeasible to derive the EMSK, any other Root Keys, or child keys associated with other Root Keys. In addition to the Root Key, several other parameters may need to be sent.

Root Key names may be derived using the EAP Session ID, and thus the key name may need to be sent along with the key. When Root Keys are delivered to another entity, the EMSKname and the lifetime associated with the specific root keys MUST also be transported to that entity.

Recommendations for transporting keys are discussed in the security considerations (Key Distribution).
Usage definition may also define how keys are bound to particular entities. This can be done through the inclusion of usage parameters and identities in the child key derivation. Some of this data is described as "channel bindings" in [RFC3748] (Aboba, B., Blunk, L., Vollbrecht, J., Carlson, J., and H. Levkowetz, “Extensible Authentication Protocol (EAP),” June 2004.).

---

6. Requirements for EAP System

TOC

The system that wishes to make use of EAP root keys derived from the EMSK must take certain things into consideration. The following is a list of these considerations:

- The EMSK MUST NOT be used for any other purpose than the key derivation described in this document.
- The EMSK MUST be secret and not known to someone observing the authentication mechanism protocol exchange.
- The EMSK MUST be maintained within a protected location inside the entity where it is generated. Only root keys derived according to this specification may be exported from this boundary.
- The EMSK MUST be unique for each EAP session
- The EAP method MUST provide an identifier for the EAP transaction that generated the key
- The system MUST define which usage definitions are used and how they are invoked.
- The system may define ways to select an alternate PRF for key derivation as defined in Section 3.1 (USRK Derivation).

The system MAY use the MSK transmitted to the NAS in any way it chooses in accordance with [RFC3748] (Aboba, B., Blunk, L., Vollbrecht, J., Carlson, J., and H. Levkowetz, “Extensible Authentication Protocol (EAP),” June 2004.) [I-D.ietf-eap-keying] (Aboba, B., Simon, D., and P. Eronen, “Extensible Authentication Protocol (EAP) Key Management Framework,” November 2007.) and other relevant specifications. This is required for backward compatibility. New usage definitions following this specification MUST NOT use the MSK. If more than one usage uses the MSK, then the cryptographic separation is not achieved. Implementations MUST prevent such combinations.

7. Security Considerations

TOC

7.1. Key strength

TOC

The effective key strength of the derived keys will never be greater than the strength of the EMSK (or a master key internal to an EAP mechanism).

7.2. Cryptographic separation of keys

TOC

The intent of the KDF is to derive keys that are cryptographically separate: the compromise of one of the usage specific root keys (USRKs) should not compromise the security of other USRKs or the EMSK. It is believed that the KDF chosen provides the desired separation.

7.3. Implementation

TOC

An implementation of an EAP framework should keep the EMSK internally as close to where it is derived as possible and only provide an interface for obtaining Root Keys. It may also choose to restrict which callers have access to which keys. A usage definition MUST NOT assume that any entity outside the EAP server or EAP peer EAP framework has access to the EMSK. In particular it MUST NOT assume that a lower layer has access to the EMSK.

7.4. Key Distribution

TOC

In some cases it will be necessary or convenient to distribute USRKs from where they are generated. Since these are secret keys they MUST be transported with their integrity and confidentiality maintained. They MUST be transmitted between authenticated and authorized parties. It is also important that the context of the key usage be transmitted along with the key. This includes information to identify the key and constraints on its usage such as lifetime.

This document does not define a mechanism for key transport. It is up to usage definitions and the systems that use them to define how keys are distributed. Usage definition designers may enforce constraints on key usage by various parties by deriving a key hierarchy and by providing entities only with the keys in the hierarchy that they need.

7.5. Key Lifetime

TOC

The key lifetime is dependent upon how the key is generated and how the key is used. Since the Root Key is the responsibility of the usage definition it must determine how long the key is valid for. If key lifetime or key strength information is available from the authenticated key exchange then this information SHOULD be used in determining the lifetime of the key. If possible it is recommended that key lifetimes be coordinated throughout the system. Setting a key lifetime shorter that a system lifetime may result is keys becoming invalid with no convenient way to refresh them. Setting a key lifetime to longer may result in decreased security since the key may be used beyond its recommended lifetime.

7.6. Entropy consideration

TOC

The number of root keys derived from the EMSK is expected to be low. Note that there is no randomness required to be introduced into the EMSK to root key derivation beyond the root key labels. Thus, if many keys are going to be derived from an Root Key it is important that Root Key to child key derivation introduce fresh random numbers in deriving each key.

---

8. IANA Considerations

TOC

The keywords "Private Use", "Specification Required" and "IETF Consensus" that appear in this document when used to describe namespace allocation are to be interpreted as described in [RFC5226] (Narten, T. and H. Alvestrand, “Guidelines for Writing an IANA Considerations Section in RFCs,” May 2008.).

TOC

8.1. Key Labels

This specification introduces a new name space for "USRK key labels". Key labels MUST be printable US-ASCII strings, and MUST NOT contain the characters at-sign ("@") except as noted below, comma (","), whitespace, control characters (ASCII codes 32 or less), or the ASCII code 127 (DEL). Labels are case-sensitive, and MUST NOT be longer than 64 characters.

Labels can be assigned based on Specification Required policy [RFC5226] (Narten, T. and H. Alvestrand, “Guidelines for Writing an IANA Considerations Section in RFCs,” May 2008.). In addition, the labels "experimental1" and "experimental2" are reserved for experimental use. The following considerations apply to their use:

Production networks do not necessarily support the use of experimental code points. The network scope of support for experimental values should carefully be evaluated before deploying any experiment across extended network domains, such as the public Internet. The potential to disrupt the stable operation of EAP devices is a consideration when planning an experiment using such code points.

The network administrators should ensure that each code point is used consistently to avoid interference between experiments. Particular attention should be given to security vulnerabilities and the freedom of different domains to employ their own experiments. Cross-domain usage is NOT RECOMMENDED.

Similarly, labels "private1" and "private2" have been reserved for Private Use within an organization. Again, cross-domain usage of these labels is NOT RECOMMENDED.

Labels starting with a string and followed by the "@" and a valid, fully qualified Internet domain name [RFC1034] can be requested by by the person or organization who are in control of the domain name. Such labels can be allocated based on Expert Review with Specification Required. Besides the review needed for Specification Required (see Section 4.1 of [RFC5226] (Narten, T. and H. Alvestrand, “Guidelines for Writing an IANA Considerations Section in RFCs,” May 2008.)), the expert needs to review the proposed usage for conformance to this specification, including the suitability of the usage according to the applicability statement outlined in Section 1.1 (Applicable usages of keys derived from the EMSK). It is RECOMMENDED that the specification contain the following information:

- *A description of the usage
- *The key label to be used
- *Length of the Root Key
- *If optional data is used, what it is and how it is maintained

*How child keys will be derived from the Root Key and how they will be used

*How lifetime of the Root Key and its child keys will be managed

*Where the Root Keys or child keys will be used and how they are communicated if necessary

The following labels are reserved by this document: "EMSK", "dsrk@ietf.org".

8.2. PRF numbers

TOC

This specification introduces a new number space for "EMSK PRF numbers". The numbers are int he range 0 to 255 Numbers from 0 to 220 are assigned through the policy IETF Consensus and numbers in the range 221 to 255 are left for Private Use. The initial registry should contain the following values:

> 0 RESERVED
>
> 1 HMAC-SHA-256 PRF+ (Default)

---

9. Acknowledgements

TOC

This document expands upon previous collaboration with Pasi Eronen. This document reflects conversations with Bernard Aboba, Jari Arkko, Avi Lior, David McGrew, Henry Haverinen, Hao Zhou, Russ Housley, Glen Zorn, Charles Clancy, Dan Harkins, Alan DeKok, Yoshi Ohba and members of the EAP and HOKEY working groups.
Thanks to Dan Harkins for the idea of using a single root key name to refer to all keys.

---

10. References

TOC

10.1. Normative References

TOC

[I-D.ietf-eap-keying]	Aboba, B., Simon, D., and P. Eronen, "Extensible Authentication Protocol (EAP) Key Management Framework," draft-ietf-eap-keying-22 (work in progress), November 2007 (TXT).
[RFC2119]	Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels," BCP 14, RFC 2119, March 1997 (TXT, HTML, XML).
[RFC3748]	Aboba, B., Blunk, L., Vollbrecht, J., Carlson, J., and H. Levkowetz, "Extensible Authentication Protocol (EAP)," RFC 3748, June 2004 (TXT).
[RFC4306]	Kaufman, C., "Internet Key Exchange (IKEv2) Protocol," RFC 4306, December 2005 (TXT).
[RFC5226]	Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs," BCP 26, RFC 5226, May 2008 (TXT).
[SHA256]	National Institute of Standards and Technology, "Secure Hash Standard," FIPS 180-2, August 2002. With Change Notice 1 dated February 2004

10.2. Informative References

TOC

[RFC0822]	Crocker, D., "Standard for the format of ARPA Internet text messages," STD 11, RFC 822, August 1982 (TXT).
[RFC1034]	Mockapetris, P., "Domain names - concepts and facilities," STD 13, RFC 1034, November 1987 (TXT).
[RFC4282]	Aboba, B., Beadles, M., Arkko, J., and P. Eronen, "The Network Access Identifier," RFC 4282, December 2005 (TXT).
[RFC4346]	Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.1," RFC 4346, April 2006 (TXT).
[SIGMA]	Krawczyk, H., "SIGMA: the 'SIGn-and-MAc' Approach to Authenticated Diffie-Hellman and its Use in the IKE Protocols," LNCS 2729, Springer, 2003. Available at http://www.informatik.uni-trier.de/~ley/db/conf/crypto/crypto2003.html

---

Authors' Addresses

TOC

	Joseph Salowey
	Cisco Systems
Email:	jsalowey@cisco.com

	Lakshminath Dondeti
	Qualcomm, Inc
Email:	ldondeti@qualcomm.com
	Vidya Narayanan
	Qualcomm, Inc
Email:	vidyan@qualcomm.com
	Madjid Nakhjiri
	Motorola
Email:	madjid.nakhjiri@motorola.com

Full Copyright Statement

TOC

Intellectual Property