Channel Binding Mechanism based on Parameter Binding in Key Derivation
draft-ohba-eap-channel-binding-02

Status of this Memo

By submitting this Internet-Draft, each author represents that any applicable patent or other IPR claims of which he or she is aware have been or will be disclosed, and any of which he or she becomes aware will be disclosed, in accordance with Section 6 of BCP 79.

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/1id-abstracts.txt.

The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html.

This Internet-Draft will expire on June 6, 2007.

Copyright Notice

Copyright (C) The Internet Society (2006).
Abstract

This document describes a channel binding mechanism based on parameter binding in the key derivation procedure. The method cryptographically binds service information to a key without need to carry the service information in EAP methods.

Table of Contents

1. Introduction ........................................... 3
   1.1. Specification of Requirements .................... 4
2. Terminology ............................................ 5
3. Basic Channel Binding Mechanism .......................... 7
   3.1. Key Derivation Function ........................... 7
   3.2. Key Scope ........................................... 8
   3.3. Key Name ............................................. 8
   3.4. Key Lifetime ........................................ 8
   3.5. Key Caching ......................................... 9
4. Channel Binding Mechanism Variants ........................ 9
   4.1. Multiple CBKs ....................................... 9
   4.2. Transferring CBMK ................................... 9
   4.3. Hierarchical Channel Binding ....................... 9
5. EAP Authenticator Consideration .......................... 10
6. Authenticator-Supplicant Protocol Requirements ............ 11
7. EAP Method Requirements .................................. 12
8. Security Considerations .................................. 13
9. IANA Considerations ..................................... 14
10. Acknowledgments ........................................ 15
11. References ............................................. 16
   11.1. Normative References .............................. 16
   11.2. Informative References ............................ 16
Authors’ Addresses ......................................... 17
Intellectual Property and Copyright Statements ............... 18
1. Introduction

EAP (Extensible Authentication Protocol) is an authentication protocol which supports multiple authentication algorithms known as "EAP methods" [RFC3748]. In EAP, an EAP peer and an EAP server generates EAP keying material, i.e., MSK (Master Session Key) and EMSK (Extended Master Session Key). A detailed framework for the generation, transport and usage of MSK is described in [I-D.ietf-eap-keying].

Each access network has its own set of parameters advertised by the EAP authenticator to EAP peers. The identity of the EAP authenticator is one of such parameters. Such parameters are referred to as service information. The service information should be bound to the MSK in a secure way to avoid possible security flaws.

The Channel Binding mechanism that is described in [RFC3748] to create such a binding is based on communicating the service information over a protected channel of an EAP method to help the EAP peer and the authentication server detect a mismatch between the service information exchanged over the protected channel and the one advertised by the EAP authenticator to the EAP peer and the authentication server. As the service information needs to be explicitly configured on the server, there is not much use for the peer to explicitly send it through the EAP method as specified in [RFC3748]. Also, this Channel Binding mechanism does not create cryptographic binding between the service information and EAP keying material. Therefore, it is possible that no binding is actually made if the exchanged service information was not checked by the peer and/or the server.

In any Channel binding solution the authentication server should be able to authenticate the service information provided by the EAP authenticator. It is not sufficient if it just ensures that the same information is available at both the EAP peer and authentication server. The EAP authenticator can provide the same false information to both the EAP peer and authentication server. For this reason, the authentication server needs to be configured with the service information of the EAP authenticators (out-of band) for authenticating the service information.

On the other hand, in roaming situations, this becomes difficult if there is no direct trust relationship with the visited network. In that case, an intermediary should be able to authenticate the service information on behalf of the authentication server.

This document describes an alternative Channel Binding mechanism to create a binding between a key exported by EAP method and the service
information. The mechanism has the following characteristics:

- The mechanism retains EAP invariants, i.e., mode independence, media independence, method independence and ciphersuite independence.
- The mechanism does not necessarily require any change to existing EAP authenticators.
- The mechanism is scalable to support a large number of EAP authenticators.

1.1. Specification of Requirements

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].
2. Terminology

Channel Binding Key (CBK)

A key that is derived from a Channel Binding Master Key (CBMK) and cryptographically bound to a Key Binding Blob (KBB) using a Key Derivation Function (KDF). A CBK is at least 64 octets in length.

Channel Binding Master Key (CBMK)

A key from which a CBK is derived using a KDF. An MSK or an EMSK is a CBMK. A CBMK is at least 64 octets in length.

Key Binding Blob (KBB)

An octet-string that is constructed from the service information advertised by an authenticator using an Authenticator-Supplicant Protocol (ASP).

Key Derivation Function (KDF)

A function by which a CBK is generated using a CBMK and a KBB as input.

Server

An entity that creates a CBK and transfers it to the authenticator. An EAP server is a server of this mechanism.

Authenticator

A network-side entity that uses a CBK for an Authenticator-Supplicant Protocol (SAP). An authenticator may or may not be co-located with the server in the same equipment. An EAP authenticator is an authenticator of this mechanism.

Supplicant

A user-side entity that uses a CBK for an Authenticator-Supplicant Protocol (SAP). An EAP peer is a supplicant of this mechanism.

Authenticator-Supplicant Protocol (ASP)

A protocol that is executed between an supplicant and an authenticator and uses a CBK for binding an authenticated identity to the protocol. The supplicant obtains the service information from the authenticator using the ASP where the service information is used for constructing a KBB. The specification of the ASP
defines how to construct a KBB from the service information. The
notion of ASP is broader than Secure Association Protocol (SAP)
[I-D.ietf-eap-keying] in that an ASP is used not only for secure
association but also for advertising the service information.
IKEv2, PANA, IEEE 802.11i, IEEE 802.11r and IEEE 802.16e are ASPs.
3. Basic Channel Binding Mechanism

The basic channel binding mechanism is described as follows (see Figure 1.

--- Diagram ---

<table>
<thead>
<tr>
<th>Server</th>
<th>Authenticator</th>
<th>Supplicant</th>
</tr>
</thead>
<tbody>
<tr>
<td>KBB CBMK</td>
<td>v v</td>
<td>+----+ CBMK</td>
</tr>
<tr>
<td>v v</td>
<td></td>
<td>v v</td>
</tr>
<tr>
<td>+--------</td>
<td></td>
<td>+--------+</td>
</tr>
<tr>
<td></td>
<td>KDF</td>
<td>(1)</td>
</tr>
<tr>
<td>+--------</td>
<td></td>
<td>+--------+</td>
</tr>
<tr>
<td></td>
<td>CBK</td>
<td>(2)</td>
</tr>
<tr>
<td>+-----------------</td>
<td>--</td>
<td>ASPI</td>
</tr>
<tr>
<td></td>
<td>+----+</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ASPI</td>
<td>&lt;-------------&gt;</td>
</tr>
</tbody>
</table>

ASPI: Instance of ASP

Figure 1: Basic Channel Binding Mechanism

1. CBK Creation: A server and supplicant creates a CBK used for an authenticator. The CBK is derived from a CBMK and bound to a KBB associated with the authenticator using a KDF. The KBB is pre-configured on the server.

2. CBK Transfer: The server transfers the CBK to the authenticator.

3. CBK Verification: The supplicant and authenticator verifies proof of possession of the CBK over the ASP. After successful verification of proof of possession of the CBK, the supplicant and authenticator are able to use the CBK in the ASP.

3.1. Key Derivation Function

A CBK is computed using prf+ defined in IKEv2 [RFC4306] in the following way.

\[
\text{CBK} = \text{prf+}(\text{CBMK}, \text{KBB})
\]
3.2. Key Scope

The scope of a CBK MUST be within the pair of the supplicant and authenticator that use the CBK.

3.3. Key Name

The name of a CBK is the string concatenation of the name of the CBMK and "CBK".

3.4. Key Lifetime

The lifetime of a CBK is determined based on the guidelines on exported and calculated key lifetimes described in [I-D.ietf-eap-keying].

3.5. Key Caching

Where explicitly supported by the ASP, the ASP MAY cache a CBK.
4. Channel Binding Mechanism Variants

4.1. Multiple CBKs

In this scenario, a server creates multiple CBKs from a single CBMK for multiple authenticators by using different KBBs for different authenticators.

4.2. Transferring CBMK

An entity that owns a CBMK MAY transfer the ownership of the CBMK to a trusted entity by transferring the CBMK to that entity. The recipient or the new owner of the CBMK MAY then act as the server on behalf of the previous owner or further transfer the ownership to another trusted entity.

4.3. Hierarchical Channel Binding

This channel binding mechanism allows CBKs to form a hierarchy (see Figure 2). As the service information needs to be explicitly configured for each authenticator at the server, it may become cumbersome to support large number of authenticators. The hierarchical Channel Binding mechanism helps solve the scalability problems when such large number of authenticators are present in a single visited network domain.

```
   CBMK0
  /  .. \   
 /    \   
CBK0_1  CBK0_n
 (=CBMK1_1)  (=CBMK1_n)
   /  .. \   /  .. \ 
  /    \   /    \ 
 ..     ..     ..     ..     ..
```

Figure 2: Hierarchical Channel Binding

In hierarchical Channel Binding, CBK verification may be performed at each level of the key hierarchy or only at the lowest level of the hierarchy. In the latter case, the lowest level authenticator MUST advertise, in the lowest level ASP, the service information necessary to construct a KBB at each level of the hierarchy.

Also, different hash algorithms used for prf+ may be negotiated at different levels and different branches of the hierarchy.
5. EAP Authenticator Consideration

When this mechanism is used, an EAP authenticator will receive and process a CBK as if it were an MSK. This enables the mechanism to work with the already deployed EAP authenticators without any modifications to them.
6. Authenticator-Supplicant Protocol Requirements

Any ASP that claims to support this mechanism MUST define how a KBB is constructed from the service information specific to the ASP where the KBB construction mechanism MUST satisfy all of the following requirements:

- Only static service information such as the identity of the authenticator is used for constructing KBBs.

- Probability of KBB collision in which the same KBB is associated with different authenticators of the ASP is either zero or reasonably low. A KBB collision may occur (i) when the KBB is computed as a hash of the service information or (ii) if the authenticators use different ASPs among which uniqueness of KBB is not guaranteed. A KBB collision may lead to domino effect [I-D.housley-aaa-key-mgmt] among authenticators associated with the collided KBB.
7. EAP Method Requirements

Any EAP method that claims to support the mechanism described in this document MUST provide at least all of the following functionalities.

1. Negotiation on enabling this mechanism. The EAP method MUST support the following negotiation over a protected channel.

* A functionality for the EAP peer to indicate the EAP server the desire to use this mechanism.

* A functionality to enable this mechanism when the EAP peer indicated the desire to use this mechanism and the EAP server implements this mechanism.

* A functionality to disable this mechanism when the EAP peer did not indicate the desire to use this mechanism.

* A functionality to fail the EAP method conversation when the EAP peer indicated the desire to use this mechanism while the EAP server does not implement this mechanism.

2. Negotiation on or specification of a hash algorithm. The EAP method MUST either provide, over a protected channel, a mechanism for negotiating on a hash algorithm used for prf+, or specify one hash algorithm used for prf+.
8. Security Considerations

The mechanism described in this document improves the security characteristics of the EAP key management framework in the following aspects.

- A secure association can never be established via the ASP if there is a difference in the KBB advertised by the EAP authenticator to the EAP peer and the KBB configured on the EAP server. This means that the mechanism can only be applied where EAP methods generating key material are used along with lower layers that utilize the keying material. For example, this mechanism would not enable verification of Channel Bindings on wired IEEE 802 networks using IEEE 802.1X, however, if a lower layer does not utilize the keying material, various attacks including a man-in-the-middle attack is possible.

- Even if a CBK is somehow transferred to an authenticator other than the intended one, the CBK can never be used by the non-intended authenticator as long as the KBB used for deriving the CBK does not collide between the intended and unintended authenticators.

The security level of this mechanism depends on probability of KBB collision among authenticators of the same ASP and among authenticators of different ASPs.
9. IANA Considerations

This document has no actions for IANA.
10. Acknowledgments

TBD.
11. References

11.1. Normative References


11.2. Informative References

Authors’ Addresses

Yoshihiro Ohba
Toshiba America Research, Inc.
1 Telcordia Drive
Piscataway, NJ  08854
USA
Phone: +1 732 699 5365
Email: yohba@tari.toshiba.com

Mohan Parthasarathy
313 Fairchild Drive
Mountain View CA  94043
USA
Email: mohan.parthasarathy@nokia.com

Mayumi Yanagiya
NTT Network service systems laboratories, NTT Corporation
9-11, Midori-Cho, 3-Chome
Musashino-Shi, Tokyo 180-8585
Japan
Email: yanagiya.mayumi@lab.ntt.co.jp