Network Working Group
Internet-Draft
Intended status: Informational
Expires: January 13, 2011

K. Hoeper
Motorola
S. Decugis
NICT
G. Zorn
Network Zen
Q. Wu
T. Taylor
Huawei

July 12, 2010

Handover Keying (HOKEY) Architecture Design
draft-hoeper-hokey-arch-design-03

Abstract

The Handover Keying (HOKEY) Working Group seeks to minimize handover delay due to authentication when a peer moves from one point of attachment to another. Work has progressed on two different approaches to reduce handover delay: early authentication (so that authentication does not need to be performed during handover), and reuse of cryptographic material generated during an initial authentication to save time during re-authentication. A starting assumption is that the mobile host or "peer" is initially authenticated using the Extensible Authentication Protocol (EAP), executed between the peer and an EAP server as defined in RFC 3748.

This document documents the HOKEY architecture. Specifically, it describes design objectives, the functional environment within which handover keying operates, the functions to be performed by the HOKEY architecture itself, and the assignment of those functions to architectural components. It goes on to illustrate the operation of the architecture within various deployment scenarios that are described more fully in other documents produced by the HOKEY Working Group.

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."

This Internet-Draft will expire on January 13, 2011.

Copyright Notice

Table of Contents

1. Introduction

The Extensible Authentication Protocol (EAP) [RFC3748] is an authentication framework that supports different types of authentication methods. Originally designed for dial-up connections, EAP is now commonly used for authentication in wireless access networks.

When a host (or "peer", the term used from this point onward) changes its point of attachment to the network, it must be re-authenticated. If a full EAP authentication must be repeated, several message round-trips between the peer and the home EAP server may be involved. The resulting delay will result in degradation or in the worst case loss of any service session in progress if communication is suspended while re-authentication is carried out. The delay is worse if the new point of attachment is in a visited network rather than the peer's home network, because of the extra procedural steps involved as well as because of the probable increase in round-trip time.

[RFC5169] describes this problem more fully and establishes design goals for solutions to reduce re-authentication delay for transfers within a single administrative domain. [RFC5169] also suggests a number of ways to achieve a solution:

- specification of a method-independent, efficient, re-authentication protocol;
- reuse of keying material from the initial authentication;
- deployment of re-authentication servers local to the peer to reduce round-trip delay; and
- specification of the additional protocol needed to allow the EAP server to pass authentication information to the local re-authentication servers.

[RFC5295] tackles the problem of reuse of keying material by specifying how to derive a hierarchy of cryptographically independent purpose-specific keys from the results of the original EAP authentication. [RFC5296] specifies a method-independent re-authentication protocol (ERP) applicable to two specific deployment scenarios:

- where the peer's home EAP server also performs re-authentication; and
- where a local re-authentication server exists but is collocated with a AAA proxy within the domain.

Other work provides further pieces of the solution or insight into the problem. [RFC5749] provides an abstract mechanism for distribution of keying material from the EAP server to re-authentication servers. (The stated scope is more general, but is restated in terms consistent with the present summary.) [RFC5836] contrasts the EAP re-authentication (ER) strategy provided by [RFC5296] (which [RFC5836] calls "vertical context transfer") with an alternative strategy called "early authentication".

[RFC5836] defines EAP early authentication as the use of EAP by a mobile peer to establish authenticated keying material on a target attachment point prior to its arrival. Here, a full EAP execution occurs before the handover of the peer takes place. Hence, the goal of EAP early authentication is to complete all EAP-related communications, including AAA signaling, in preparation for the handover, before the mobile device actually moves.

Both EAP re-authentication and early authentication enable faster inter-authenticator handovers. However, it is currently unclear how the necessary handover infrastructure for ER is deployed and can be integrated into existing EAP infrastructures. In particular, previous work has not described how ER servers (as defined below) should be integrated into local and home domain networks.

This document proposes a general HOKEY architecture and demonstrates how it can be adapted to different deployment scenarios. To begin with, Section 3 recalls the design objectives for the HOKEY architecture. Section 4 reviews the functions that must be supported within the architecture. Section 5 describes the components of the HOKEY architecture. Finally, Section 6 describes the different deployment scenarios that the HOKEY Working Group has addressed and the information flows that must occur within those scenarios, by reference to the documents summarized above where possible and otherwise within this document itself.

2. Terminology

This document contains no normative language, hence [RFC2119] language does not apply.

This document reuses most of the terms defined in Section 2.2 of [RFC5836]. In addition, it defines the following:

EAP Early Authentication
See Section 3.3.2 of [RFC5836].

EAP Re-authentication (ER)
The use of keying material derived from an initial EAP authentication to enable single-roundtrip re-authentication of a mobile peer. For a detailed description of the keying material see Section 3 of [RFC5296].

ER Server
A component of the HOKEY architecture that terminates the EAP re-authentication exchange with the peer.

ER Key Management
An instantiation of the mechanism provided by [RFC5749] for creating and delivering root keys from an EAP server to an ER server.

3. Design Goals

This section investigates the design goals for the HOKEY architecture. These include reducing the signaling overhead for re-authentication and early authentication, integrating local domain name discovery, and improving deployment scalability. These goals supplement the discussion in [RFC5169].

3.1. Reducing Signalling Overhead

3.1.1. Minimized Communications with Home Servers

ERP requires only one round trip, however, this roundtrip may require communications between a peer and its home ER and/or home AAA server even if the peer is currently attached to a visited (local) network. As a result, even this one round trip may introduce long delays because home ER and home AAA servers may be distant from the peer. To lower the signaling overhead, communication with the home ER server and home AAA server should be minimized. Ideally, a peer should only need to communicate with local servers and other local entities.

3.1.2. Integrated Local Domain Name (LDN) Discovery

Ideally, whenever a peer performs a handover, ERP is executed between the peer and a local ER server, thus, reducing handover latency by avoiding a full EAP authentication with the peer's home EAP server. For this to work, ERP bootstrapping must occur before (implicit) or during (explicit) a handover to transport the necessary re-authentication root keys to the local ER server involved. Implicit bootstrapping is preferable because it does not require communication with the home ER server during handover (see previous section), but

it requires the peer to know the domain name of the ER server in order to derive the necessary re-authentication keying material. [RFC5296] does not specify such a domain name discovery mechanism and suggests that the peer may learn the domain name through the EAP-Initiate/Re-auth-Start message or via lower layer announcements. To allow more efficient handovers, a HOKEY architecture should support an efficient domain name discovery mechanism and allow its integration with ERP implicit bootstrapping. Even in the case of explicit bootstrapping, local domain name discovery should be optimized such that it does not require contacting the home AAA server, as is currently the case.

3.2. Better Deployment Scalability

To provide better deployment scalability, it should not be required that the HOKEY server and AAA servers or proxies are collocated. Separation of these entities may cause problems with routing, but allows flexibility in deployment and implementation.

4. Functions That Must Be Supported

4.1. System Overview

This section views the HOKEY architecture as the implementation of a subsystem providing authentication services to AAA. Not only does AAA depend on the authentication subsystem, but the latter also depends on AAA as a means for the routing and secure transport of messages internal to the operation of network access authentication.

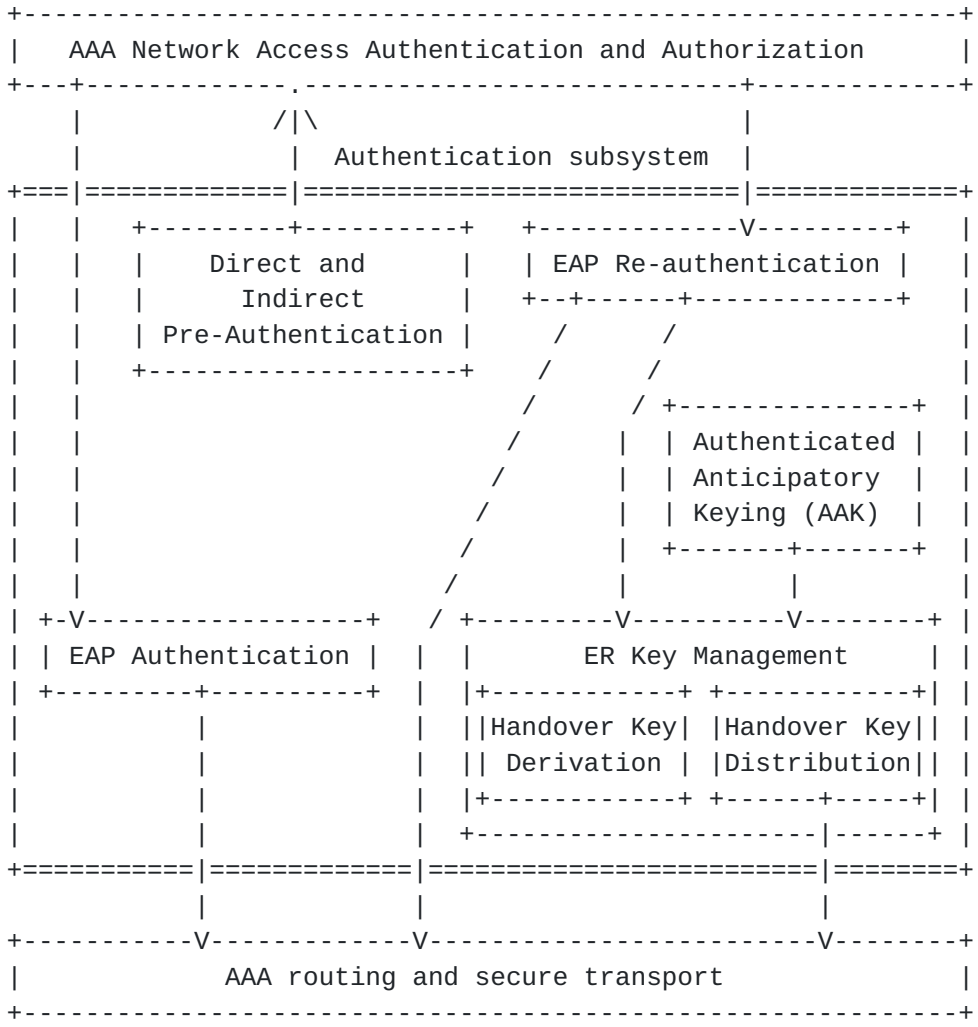
The operation of the authentication subsystem also depends on the availability of a number of discovery functions:

- discovery of candidate access points, by the peer, by the serving attachment point, or by some other entity;
- discovery of the authentication services supported at a given candidate access point;
- discovery of the required server in the home domain when a candidate access point is not in the same domain as the serving attachment point, or no local server is available;
- peer discovery of the local domain name (LDN) when EAP re-authentication is used with a local server.

It is assumed that these functions are provided by the environment within which the authentication subsystem operates, and are outside

the scope of the authentication subsystem itself. Local domain name discovery is a possible exception.

Figure 1 shows the major functions comprising the authentication subsystem and their interdependencies. These functions are described below. [EDITOR'S NOTE: These probably need refinement. The relationship of pre-authentication to EAP authentication, for instance, is currently not totally correct, when one takes account of the roles described in Section 5. AAK also needs an extension of ER key management.]



Arrows show the direction of functional dependency.

Figure 1: Authentication Subsystem Functional Overview

Figure 1 shows the following dependencies:

- When AAA is invoked to authenticate and authorize network access, it uses one of two services offered by the authentication subsystem: full EAP authentication, or EAP re-authentication.

- Pre-authentication triggers AAA network access authentication and authorization at each candidate access point, which in turn causes full EAP authentication to be invoked.

- EAP re-authentication invokes ER key management at the time of authentication to create and distribute keying material to ER servers.

- Authenticated anticipatory keying (AAK) relies on ER key management to establish keying material on ER/AAK servers, but uses an extension to ER key management to derive and establish keying material on candidate authenticators.

EAP authentication, EAP re-authentication, and handover key distribution depend on the routing and secure transport service provided by AAA. Discovery functions and the function of authentication and authorization of network entities (access points, ER servers) are not shown. As stated above, these are external to the authentication subsystem.

4.2. Pre-Authentication Function (Direct or Indirect)

The pre-authentication function is responsible for discovery of candidate access points and completion of network access authentication and authorization at each in advance of handover. The operation of this function is described in general terms in [RFC5836]. No document is yet available to describe the implementation of pre-authentication in terms of specific protocols. [RFC5873] could be part of the solution, but is Experimental rather than Standards Track.

4.3. EAP Re-authentication Function

The EAP re-authentication function is responsible for authenticating the peer at a specific access point using keying material derived from a prior full EAP authentication. [RFC5169] provides the design objectives for an implementation of this function. [RFC5296] describes a protocol to implement EAP re-authentication subject to the architectural restrictions noted above. Work is in progress to relax those restrictions.

4.4. EAP Authentication Function

The EAP authentication function is responsible for authenticating the peer at a specific access point using a full EAP exchange. [RFC3748] defines the associated protocol. [RFC5836] shows the use of EAP as part of pre-authentication. Note that the HOKEY Working Group has not specified the non-AAA protocol required to transport EAP frames over IP that is shown in Figures 3 and 5 of [RFC5836], although [RFC5873] is a candidate.

4.5. Authenticated Anticipatory Keying (AAK) Function

The authenticated anticipatory keying function is responsible for pre-placing keying material derived from an initial full EAP authentication on candidate access points. The operation is carried out in two steps: ER key management (with trigger not currently specified) places root keys derived from initial EAP authentication onto an ER/AAK server associated with the peer. When requested by the peer, the ER/AAK server derives and pushes predefined master session keys to a list of candidate access points. The operation of the authenticated anticipatory keying function is described in very general terms in [RFC5836]. A protocol implementation is being specified in [I-D.hokey-erp-aak].

4.6. EAP-Based Handover Key Management

EAP-based handover key management consists of EAP method independent key derivation and distribution and comprises the following specific functions:

- handover key derivation; and
- handover key distribution.

The derivation of handover keys is specified in [RFC5295], and key distribution is specified in [RFC5749].

5. Components of the HOKEY Architecture

This section describes the components of the HOKEY architecture, in terms of the functions they perform. The components cooperate as described in this section to carry out the functions described in the previous section. Section 6 describes the different deployment scenarios that are possible using these functions.

The components of the HOKEY architecture are as follows:

- the peer;
- the authenticator, which is a part of the serving access point and candidate access points;
- the EAP server; and
- the ER server, either in the home domain or local to the authenticator.

[EDITOR'S NOTE: probably have to add the ER/AAK server named in [I-D.hokey-erp-aak] to this list.]

5.1. Functions of the Peer

The peer participates in the functions described in Section 4 as shown in Table 1.

Function	Peer Role
EAP authentication	Determines that full EAP authentication is needed based on context (e.g., initial authentication), prompting from the authenticator, or discovery that only EAP authentication is supported. Participates in the EAP exchange with the EAP server.
-	-
Direct pre-authentication	Discovers candidate access points. Initiates pre-authentication with each, followed by EAP authentication as above, but using IP rather than L2 transport for the EAP frames.
-	-
Indirect pre-authentication	Enters into a full EAP exchange when triggered, using either L2 or L3 transport for the frames.
-	-
EAP re-authentication	Determines that EAP re-authentication is possible based on discovery or authenticator prompting. Discovers ER server. Participates in ERP exchange with ER server.
-	-

Authenticated anticipatory keying	Determines that AAK is possible based on discovery or serving authenticator prompting. Discovers candidate access points. Sends request to serving authenticator to distribute keying material to the candidate access points.
-	-
ER key management	No role.

Table 1: Functions of the Peer

5.2. Functions of the Serving Authenticator

The serving authenticator participates in the functions described in Section 4 as shown in Table 2.

Function	Serving Authenticator Role
EAP authentication	No role.
-	-
Direct pre-authentication	No role.
-	-
Indirect pre-authentication	Discovers candidate access points. Initiates an EAP exchange between the peer and the EAP server through each candidate authenticator. Mediates between L2 transport of EAP frames on the peer side and a non-AAA protocol over IP toward the candidate access point.
-	-
EAP re-authentication	No role.
-	-
Authenticated anticipatory keying	Mediates between L2 transport of AAK frames on the peer side and AAA transport toward the ER/AAK server.
-	-
ER key management	No role.

Table 2: Functions of the Serving Authenticator

5.3. Functions of the Candidate Authenticator

The candidate authenticator participates in the functions described in Section 4 as shown in Table 3.

Function	Candidate Authenticator Role
EAP authentication	Invokes AAA network access authentication and authorization upon handover/initial attachment. Mediates between L2 transport of EAP frames on the peer link and AAA transport toward the EAP server.
-	-
Direct pre-authentication	Invokes AAA network access authentication and authorization when the peer initiates authentication. Mediates between non-AAA L3 transport of EAP frames on the peer side and AAA transport toward the EAP server.
-	-
Indirect pre-authentication	Same as direct pre-authentication, except that it communicates with the serving authenticator rather than the peer.
-	-
EAP re-authentication	Invokes AAA network access authentication and authorization upon handover. Discovers or is configured with the address of the ER server. Mediates between L2 transport of a ERP frames on the peer side and AAA transport toward the ER server.
-	-
Authenticated anticipatory keying	Receives and saves pMSK.
-	-
ER key management	No role.

Table 3: Functions of the Candidate Authenticator

5.4. Functions of the EAP Server

The EAP server participates in the functions described in Section 4 as shown in Table 4.

Function	EAP Server Role
EAP authentication	Authenticates and authorizes the candidate access point to act as authenticator. Terminates EAP signalling between it and the peer via the candidate authenticator. Determines whether network access authentication succeeds or fails. Provides MSK to authenticator.
-	-
Direct pre-authentication	As for EAP authentication.
-	-
Indirect pre-authentication	As for EAP authentication.
-	-
EAP re-authentication	Mutually authenticates with the ER server and authorizes it for receiving keying amterial. Provides rRK or DSrRK to the ER server.
-	-
Authenticated anticipatory keying	As for EAP re-authentication.
-	-
ER key management	Creates rRK or DSrRK and distributes it to ER server requesting the information.

Table 4: Functions of the EAP Server

5.5. Functions of the ER Server

The ER server participates in the functions described in Section 4 as shown in Table 5. [EDITOR'S NOTE: Need discussion of respective roles of local and home ER server, or whether there should even be such a distinction.]

Function	ER Server Role
EAP authentication	No role.
-	-
Direct pre-authentication	No role.
-	-
Indirect pre-authentication	No role.
-	-
EAP re-authentication	Authenticates and authorizes the candidate access point to act as authenticator. Authenticates itself to the EAP server and acquires rRK or DSrRK as applicable when necessary. Terminates ERP signalling between it and the peer via the candidate authenticator. Determines whether network access authentication succeeds or fails. Provides MSK to authenticator.
-	-
Authenticated anticipatory keying	Authenticates itself to the EAP server and acquires rRK or DSrRK as applicable when necessary. Authenticates and authorizes the candidate access points to act as authenticator. Derives pMSKs and passes them to the candidate access points.
-	-
ER key management	Receives and saves rRK or DSrRK as applicable.

Table 5: Functions of the ER Server

6. Deployment Scenarios

The necessity for this section and its contents are TBD.

7. AAA Consideration

7.1. Standalone HOKEY server

TBD.

8. Security Considerations

TBD

9. IANA Considerations

This document has no actions for IANA.

10. Acknowledgments

The authors would like to thank Qin Wu, Mark Jones, and Zhen Cao for their reviews of the previous version of this draft.

11. Informative 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.

[RFC5169] Clancy, T., Nakhjiri, M., Narayanan, V., and L. Dondeti, "Handover Key Management and Re-Authentication Problem Statement", RFC 5169, March 2008.

[RFC5247] Aboba, B., Simon, D., and P. Eronen, "Extensible Authentication Protocol (EAP) Key Management Framework", RFC 5247, August 2008.

[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.

[RFC5296] Narayanan, V. and L. Dondeti, "EAP Extensions for EAP Re-authentication Protocol (ERP)", RFC 5296, August 2008.

[RFC5749] Hoeper, K., Nakhjiri, M., and Y. Ohba, "Distribution of EAP-Based Keys for Handover and Re-Authentication", RFC 5749, March 2010.

[RFC5836] Ohba, Y., Wu, Q., and G. Zorn, "Extensible Authentication Protocol (EAP) Early Authentication Problem Statement", RFC 5836, April 2010.

[RFC5873] Ohba, Y. and A. Yegin, "Pre-Authentication Support for the Protocol for Carrying Authentication for Network Access (PANA)", RFC 5873, May 2010.

[I-D.hokey-erp-aak] Cao, Z., Deng, H., Wang, Y., Wu, Q., and G. Zorn, "EAP Re-authentication Protocol Extensions for Authenticated Anticipatory Keying (ERP/AAK) (Work in Progress)", May 2010.

Authors' Addresses

Katrin Hoeper
Motorola, Inc.
1301 E. Algonquin Road
Schaumburg, IL 60196
USA

Email: khoeper@motorola.com

Sebastien Decugis
NICT
4-2-1 Nukui-Kitamachi
Tokyo, Koganei 184-8795
Japan

Email: sdecugis@nict.go.jp

Glen Zorn
Network Zen
1310 East Thomas Street
Seattle, Washington 98102
USA

Email: gwz@net-zen.net

Qin Wu
Huawei Technologies Co.,Ltd
Site B, Floor 12F, Huihong Mansion, No.91 Baixia Rd.
Nanjing, JiangSu 210001
China

Phone: +86-25-84565892
Email: sunseawq@huawei.com

Tom Taylor
Huawei Technologies Co., Ltd
Ottawa
Canada

Email: tom111.taylor@bell.net