Internet Engineering Task Force (IETF)
Request for Comments: 6697
Category: Informational
ISSN: 2070-1721

G. Zorn, Ed.
Network Zen
Q. Wu
T. Taylor
Huawei
Y. Nir
Check Point
K. Hoeper
Motorola Solutions, Inc.
S. Decugis
INSIDE Secure
July 2012

Handover Keying (HOKEY) Architecture Design

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 basic 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 defines 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.

Status of This Memo

This document is not an Internet Standards Track specification; it is published for informational purposes.

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Not all documents approved by the IESG are a candidate for any level of Internet Standard; see Section 2 of RFC 5741.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at http://www.rfc-editor.org/info/rfc6697.

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 a variety of 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 the probable increase in round-trip time.

Clancy, et al. [RFC5169] describe this problem more fully and establish design goals for solutions to reduce re-authentication delay for transfers within a single administrative domain. They also suggest a number of ways to achieve a solution:

- specification of a method-independent, efficient re-authentication protocol based upon EAP;
- reuse of keying material from the initial EAP 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.

Salowey, et al. [RFC5295] tackle the problem of the 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, while Cao, et al. [RFC6696] specify a method-independent re-authentication protocol (the EAP 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 co-located with an Authentication, Authorization, and Accounting (AAA) proxy within the domain.

Other work provides further pieces of the solution or insight into the problem. For the purpose of this memo, Hoeper, et al. [RFC5749] provide an abstract mechanism for distribution of keying material from the EAP server to re-authentication servers. Ohba, et al. [RFC5836] contrast the EAP Re-authentication (ER) strategy provided by ERP with an alternative strategy called "early

authentication". RFC 5836 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. 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. Early authentication includes direct and indirect pre-authentication as well as Authenticated Anticipatory Keying (AAK). All three early authentication mechanisms provide means to securely establish authenticated keying material on a Candidate Attachment Point (CAP) while still being connected to the Serving Attachment Point (SAP) but vary in their respective system assumptions and communication paths. In particular, direct pre-authentication assumes that clients are capable of discovering CAPs and all communications are routed through the SAP. On the other hand, indirect pre-authentication assumes an existing relationship between the SAP and CAP, whereas the discovery and selection of CAPs is outside the scope of AAK. Furthermore, both direct and indirect pre-authentication require a full EAP execution to occur before the handover of the peer takes place, while AAK techniques (like ERP [RFC6696]) use keys derived from the initial EAP authentication.

Both EAP re-authentication and early authentication enable faster inter-authenticator handovers. However, it is currently unclear how the necessary handover infrastructure can be deployed and integrated into existing EAP infrastructures. In particular, previous work has not described how ER servers that act as endpoints in the re-authentication process should be integrated into local and home domain networks. Furthermore, how EAP infrastructure can support the timely triggering of early authentications and aid with the selection of CAPs is currently unspecified.

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. 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. Finally, Section 7 provides an analysis of how AAA protocols can be applied in the HOKEY architecture.

2. Terminology

This document reuses terms defined in Section 2 of Ohba, et al. [RFC5836] and Section 2 of Cao, et al. [RFC6696]. In addition, it defines the following:

DS-rRK
: Domain-Specific re-authentication Root Key.

pMSK
: pre-established Master Session Key.

EAP Early Authentication
: The use of EAP by a mobile peer to establish authenticated keying material on a target attachment point prior to its arrival; see Ohba, et al. [RFC5836].

ER Key Management
: An instantiation of the mechanism described in Hoeper, et al. [RFC5749] for creating and delivering root keys from an EAP server to an ER server.

EAP Re-authentication (ER)
: The use of keying material derived from an initial EAP authentication to enable single-round-trip re-authentication of a mobile peer. For a detailed description of the keying material, see Section 4 of Cao, et al. [RFC6696].

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

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, enabling fault-tolerant re-authentication, and improving deployment scalability. These goals supplement those discussed in Section 4 of RFC 5169. Note that the identification and selection of CAPs is not a goal of the architecture, since those operations are generally specific to the lower layer in use.

3.1. Reducing Signaling Overhead

3.1.1. Minimized Communications with Home Servers

ERP [RFC6696] requires only one round trip; however, this round trip may require communication between a peer and its home ER and/or home AAA server in explicit bootstrapping and communication between local servers and the home server in implicit bootstrapping 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 the home ER and home AAA servers may be distant from the peer and the network to which it is attached. To lower 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. Minimized User Interaction for Authentication

When the peer is initially attached to the network or moves between heterogeneous networks, full EAP authentication between the peer and EAP server occurs and user interaction may be needed, e.g., a dialog to prompt the user for credentials. To reduce latency, user interaction for authentication at each handover should be minimized. Ideally, user involvement should take place only during initial authentication and subsequent re-authentication should occur transparently.

3.2. Integrated Local Domain Name (LDN) Discovery

ERP bootstrapping must occur before (implicit) or during (explicit) a handover to transport the necessary keys to the local ER server involved. Implicit bootstrapping is preferable because it does not require communication with the home ER server during handover, but it requires that the peer know the domain name of the ER server before the subsequent local ERP exchange happens in order to derive the necessary re-authentication keying material. ERP [RFC6696] 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. However, domain name discovery happens after the implicit bootstrapping completes, which may introduce extra latency. To allow more efficient handovers, a HOKEY architecture should support an efficient domain name discovery mechanism (for example, see Zorn, Wu & Wang [RFC6440]) and allow its integration with ERP implicit bootstrapping. Even in the case of explicit bootstrapping, LDN discovery should be optimized such that it does not require contacting the home AAA server, as is currently the case.

3.3. Fault-Tolerant Re-Authentication

If all authentication services depend upon a remote server, a network partition can result in the denial of service to valid users. However, if for example an ER server exists in the local network, previously authenticated users can re-authenticate even though a link to the home or main authentication server doesn't exist.

3.4. Improved Deployment Scalability

To provide better deployment scalability, there should be no requirement for the co-location of entities providing handover keying services (e.g., ER servers) and AAA servers or proxies. Separation of these entities may cause problems with routing but allows greater flexibility in deployment and implementation.

4. Required Functionality

4.1. Authentication Subsystem Functional Overview

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

- discovery of CAPs by the peer, by the SAP, or by some other entity;
- discovery of the authentication services supported at a given CAP;
- discovery of the required server in the home domain when a CAP is not in the same domain as the SAP, or no local server is available;
- peer discovery of the 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. LDN discovery is a possible exception.

The major functions comprising the authentication subsystem and their interdependencies are discussed in greater detail below.

- When AAA is invoked to authorize network access, it uses one of two services offered by the authentication subsystem: full EAP authentication or EAP re-authentication. Note that although AAA may perform authentication directly in some cases, when EAP is

utilized AAA functions only as a transport for EAP messages and the encryption keys (if any) resulting from successful EAP authentication.

- Pre-authentication triggers AAA network access authorization at each CAP, 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.

- 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. AAK uses an extension to EAP re-authentication to communicate with ER/AAK servers.

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 CAPs and completion of network access authentication and authorization at each CAP in advance of handover. The operation of this function is described in general terms in Ohba, et al. [RFC5836]. No document is yet available to describe the implementation of pre-authentication in terms of specific protocols; pre-authentication support for the Protocol for Carrying Authentication for Network Access (PANA) [RFC5873] could be part of the solution.

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. RFC 5169 [RFC5169] provides the design objectives for an implementation of this function. ERP [RFC6696] describes a protocol to implement EAP re-authentication.

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. Aboba, et al. [RFC3748] define the associated protocol, while Ohba, et al. [RFC5836] describe 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 Ohba, et al. [RFC5836], although PANA [RFC5873] is a candidate.

4.5. Authenticated Anticipatory Keying (AAK) Function

The AAK function is responsible for pre-placing keying material derived from an initial full EAP authentication on CAPs. 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 one or more CAPs. The operation of the AAK function is described in very general terms in Ohba, et al. [RFC5836]. A protocol specification exists (see Cao, et al. [RFC6630]).

4.6. Management of EAP-Based Handover Keys

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

- handover key derivation
- handover key distribution

The derivation of handover keys is specified in Salowey, et al. [RFC5295], and AAA-based key distribution is specified in Hoeper, Nakhjiri & Ohba [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 SAP and CAPs;
- the EAP server;
- the ER server; and
- the ER/AAK server [RFC6630], either in the home domain or local to the authenticator.

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 CAPs. 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. Participates in ERP exchange with the ER server.
-	-
AAK	Determines that AAK is possible based on discovery or serving authenticator prompting. Discovers CAPs. Participates in ERP/AAK exchange, requesting distribution of keying material to the CAPs.
-	-
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 CAPs. 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 CAP.
-	-
EAP re-authentication	No role.
-	-
AAK	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 ERP frames on the peer side and AAA transport toward the ER server.
-	-
AAK	Receives and saves the 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	Terminates EAP signaling between it and the peer via the candidate authenticator. Determines whether network access authentication succeeds or fails. Provides the MSK to the authenticator (via AAA).
-	-
Direct pre-authentication	Same as for EAP authentication.
-	-
Indirect pre-authentication	Same as for EAP authentication.
-	-
EAP re-authentication	Provides an rRK or DS-rRK to the ER server (via AAA).
-	-
AAK	Same as for EAP re-authentication.
-	-
ER key management	Creates an rRK or DS-rRK and distributes it to the 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.

Function	ER Server Role
EAP authentication	No role.
-	-
Direct pre-authentication	No role.
-	-
Indirect pre-authentication	No role.
-	-
EAP re-authentication	Acquires an rRK or DS-rRK as applicable when necessary. Terminates ERP signaling between it and the peer via the candidate authenticator. Determines whether network access authentication succeeds or fails. Provides an MSK to the authenticator.
-	-
AAK	Acquires an rRK or DS-rRK as applicable when necessary. Derives pMSKs and passes them to the CAPs.
-	-
ER key management	Receives and saves an rRK or DS-rRK as applicable.

Table 5: Functions of the ER Server

6. Usage Scenarios

Depending upon whether a change in a domain or access technology is involved, we have the following usage scenarios.

6.1. Simple Re-Authentication

The peer remains stationary and re-authenticates to the original access point. Note that in this case, the SAP takes the role of the CAP in the discussion above.

6.2. Intra-Domain Handover

The peer moves between two authenticators in the same domain. In this scenario, the peer communicates with the ER server via the ER authenticator within the same network.

6.3. Inter-Domain Handover

The peer moves between two different domains. In this scenario, the peer communicates with more than one ER server via one or two different ER authenticators. One ER server is located in the current network as the peer, and one is located in the previous network from which the peer moves. Another ER server is located in the home network to which the peer belongs.

6.4. Inter-Technology Handover

The peer moves between two heterogeneous networks. In this scenario, the peer needs to support at least two access technologies. The coverage of two access technologies usually is overlapped during handover. In this case, only authentication corresponding to intra-domain handover is required; i.e., the peer can communicate with the same local ER server to complete authentication and obtain keying material corresponding to the peer.

7. AAA Considerations

This section provides an analysis of how the AAA protocol can be applied in the HOKEY architecture in accordance with Section 4.1 ("Authentication Subsystem Functional Overview").

7.1. Authorization

Authorization is a major issue in deployments. Wherever the peer moves around, the home AAA server provides authorization for the peer during its handover. However, it is unnecessary to couple authorization with authentication at every handover, since authorization is only needed when the peer is initially attached to the network or moves between two different AAA domains. The EAP key management document [RFC5247] discusses several vulnerabilities that are common to handover mechanisms. One important issue arises from the way that the authorization decisions might be handled at the AAA server during network access authentication. For example, if AAA proxies are involved, they may also influence authorization decisions. Furthermore, the reasons for choosing a particular decision are not communicated to the AAA clients. In fact, the AAA client only knows the final authorization result. Another issue relates to session management. In some circumstances, when the peer

moves from one authenticator to another, the peer may be authenticated by the different authenticator during a period of time, and the authenticator to which the peer is currently attached needs to create a new AAA user session; however, the AAA server should not view these handoffs as different sessions. Otherwise, this may affect user experience and also cause accounting or logging issues. For example, session ID creation, in most cases, is done by each authenticator to which the peer attaches. In this sense, the new authenticator acting as AAA client needs to create a new AAA user session from scratch, which forces its corresponding AAA server to terminate the existing user session with the previous authenticator and set up a new user session with the new authenticator. This may complicate the setup and maintenance of the AAA user session.

7.2. Transport Aspect

The existing AAA protocols can be used to carry EAP and ERP messages between the AAA server and AAA clients. AAA transport of ERP messages is specified in Hoeper, Nakhjiri & Ohba [RFC5749] and Bournelle, et al. [DIAMETER-ERP]. AAA transport of EAP messages is specified in [RFC4072]. Key transport also can be performed through a AAA protocol. Zorn, Wu & Cakulev [DIAMETER-AVP] specify a set of Attribute-Value Pairs (AVPs) providing native Diameter support of cryptographic key delivery.

8. Security Considerations

This document does not introduce any new security vulnerabilities.

9. Acknowledgments

The authors would like to thank Mark Jones, Zhen Cao, Semyon Mizikovsky, Stephen Farrell, Ondrej Sury, Richard Barnes, Jari Arkko, and Lionel Morand for their reviews and comments.

10. References

10.1. Normative References

[RFC3748] Aboba, B., Blunk, L., Vollbrecht, J., Carlson, J., and H. Levkowetz, Ed., "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.

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

[RFC6696] Cao, Z., He, B., Shi, Y., Wu, Q., Ed., and G. Zorn, Ed., "EAP Extensions for the EAP Re-authentication Protocol (ERP)", RFC 6696, July 2012.

10.2. Informative References

[DIAMETER-AVP]
Zorn, G., Wu, Q., and V. Cakulev, "Diameter Attribute-Value Pairs for Cryptographic Key Transport", Work in Progress, August 2011.

[DIAMETER-ERP]
Bournelle, J., Morand, L., Decugis, S., Wu, Q., and G. Zorn, "Diameter Support for the EAP Re-authentication Protocol (ERP)", Work in Progress, June 2012.

[RFC4072] Eronen, P., Ed., Hiller, T., and G. Zorn, "Diameter Extensible Authentication Protocol (EAP) Application", RFC 4072, August 2005.

[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.

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

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

[RFC6440] Zorn, G., Wu, Q., and Y. Wang, "The EAP Re-authentication Protocol (ERP) Local Domain Name DHCPv6 Option", RFC 6440, December 2011.

[RFC6630] Cao, Z., Deng, H., Wu, Q., and G. Zorn, Ed., "EAP Re-authentication Protocol Extensions for Authenticated Anticipatory Keying (ERP/AAK)", RFC 6630, June 2012.

Authors' Addresses

Glen Zorn (editor)
Network Zen
227/358 Thanon Sanphawut
Bang Na, Bangkok 10260
Thailand
Phone: +66 (0) 909 201060
EMail: glenzorn@gmail.com

Qin Wu
Huawei Technologies Co., Ltd.
101 Software Avenue, Yuhua District
Nanjing, JiangSu 210012
China
Phone: +86-25-84565892
EMail: bill.wu@huawei.com

Tom Taylor
Huawei Technologies Co., Ltd.
Ottawa, Ontario
Canada
EMail: tom.taylor.stds@gmail.com

Yoav Nir
Check Point
5 Hasolelim St.
Tel Aviv 67897
Israel
EMail: ynir@checkpoint.com

Katrin Hoeper
Motorola Solutions, Inc.
1301 E. Algonquin Road
Schaumburg, IL 60196
USA
EMail: khoeper@motorolasolutions.com>

Sebastien Decugis
INSIDE Secure
41 Parc Club du Golf
Aix-en-Provence 13856
France
Phone: +33 (0)4 42 39 63 00
EMail: sdecugis@freediameter.net