   Handover Key Management and Re-Authentication Problem Statement

Status of This Memo

Abstract

   This document describes the Handover Keying (HOKEY) re-authentication
   problem statement. The current Extensible Authentication Protocol
   (EAP) keying framework is not designed to support re-authentication
   and handovers without re-executing an EAP method. This often causes
   unacceptable latency in various mobile wireless environments. This
   document details the problem and defines design goals for a generic
   mechanism to reuse derived EAP keying material for handover.

Table of Contents

1. Introduction

   The Extensible Authentication Protocol (EAP), specified in RFC 3748
   [RFC3748] is a generic framework supporting multiple authentication
   methods. The primary purpose of EAP is network access control. It
   also supports exporting session keys derived during the
   authentication. The EAP keying hierarchy defines two keys that are
   derived at the top level, the Master Session Key (MSK) and the
   Extended Master Session Key (EMSK).

   In many common deployment scenarios, an EAP peer and EAP server
   authenticate each other through a third party known as the pass-
   through authenticator (hereafter referred to as simply
   "authenticator"). The authenticator is responsible for encapsulating
   EAP packets from a network-access technology lower layer within the
   Authentication, Authorization, and Accounting (AAA) protocol. The
   authenticator does not directly participate in the EAP exchange, and
   simply acts as a gateway during the EAP method execution.

   After successful authentication, the EAP server transports the MSK to
   the authenticator. Note that this is performed using AAA protocols,
   not EAP itself. The underlying L2 or L3 protocol uses the MSK to
   derive additional keys, including the transient session keys (TSKs)
   used for per-packet encryption and authentication.

   Note that while the authenticator is one logical device, there can be
   multiple physical devices involved. For example, the CAPWAP model
   [RFC3990] splits authenticators into two logical devices: Wireless
   Termination Points (WTPs) and Access Controllers (ACs). Depending on
   the configuration, authenticator features can be split in a variety
   of ways between physical devices; however, from the EAP perspective,
   there is only one logical authenticator.

   Wireless handover between access points or base stations is typically
   a complex process that involves several layers of protocol execution.
   Often times executing these protocols results in unacceptable delays
   for many real-time applications such as voice [MSA03]. One part of
   the handover process is EAP re-authentication, which can contribute
   significantly to the overall handover time [MSPCA04]. Thus, in many
   environments we can lower overall handover time by lowering EAP re-
   authentication time.

   In EAP existing implementations, when a peer arrives at the new
   authenticator, it runs an EAP method irrespective of whether it has
   been authenticated to the network recently and has unexpired keying
   material. This typically involves an EAP-Response/Identity message
   from the peer to the server, followed by one or more round trips
   between the EAP server and peer to perform the authentication,

followed by the EAP-Success or EAP-Failure message from the EAP
server to the peer. At a minimum, the EAP exchange consists of 1.5
round trips. However, given the way EAP interacts with AAA, and
given that an EAP identity exchange is typically employed, at least 2
round trips are required to the EAP server. An even higher number of
round trips is required by the most commonly used EAP methods. For
instance, EAP-TLS (Extensible Authentication Protocol - Transport
Layer Security) requires at least 3, but typically 4 or more, round
trips.

There have been attempts to solve the problem of efficient re-
authentication in various ways. However, those solutions are either
EAP-method specific or EAP lower-layer specific. Furthermore, these
solutions do not deal with scenarios involving handovers to new
authenticators, or they do not conform to the AAA keying requirements
specified in [RFC4962].

This document provides a detailed description of efficient EAP-based
re-authentication protocol design goals. The scope of this protocol
is specifically re-authentication and handover between authenticators
within a single administrative domain. While the design goals
presented in this document may facilitate inter-technology handover
and inter-administrative-domain handover, they are outside the scope
of this protocol.

2. Terminology

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], with the
qualification that, unless otherwise stated, they apply to the design
of the re-authentication protocol, not its implementation or
application.

With respect to EAP, this document follows the terminology that has
been defined in [RFC3748] and [EAP-KEYING].

3. Problem Statement

Under the existing model, any re-authentication requires a full EAP
exchange with the EAP server to which the peer initially
authenticated [RFC3748]. This introduces handover latency from both
network transit time and processing delay. In service provider
networks, the home EAP server for a peer could be on the other side
of the world, and typical intercontinental latencies across the
Internet are 100 to 300 milliseconds per round trip [LGS07].

Processing delays average a couple of milliseconds for symmetric-key
operations and hundreds of milliseconds for public-key operations.

An EAP conversation with a full EAP method run can take two or more
round trips to complete, causing delays in re-authentication and
handover times. Some methods specify the use of keys and state from
the initial authentication to finish subsequent authentications in
fewer round trips and without using public-key operations (detailed
in Section 6.1). However, even in those cases, multiple round trips
to the EAP server are required, resulting in unacceptable handover
times.

In summary, it is undesirable to run an EAP Identity and complete EAP
method exchange each time a peer authenticates to a new authenticator
or needs to extend its current authentication with the same
authenticator. Furthermore, it is desirable to specify a method-
independent, efficient, re-authentication protocol. Keying material
from the initial authentication can be used to enable efficient re-
authentication. It is also desirable to have a local server with
low-latency connectivity to the peer that can facilitate re-
authentication. Lastly, a re-authentication protocol should also be
capable of supporting scenarios where an EAP server passes
authentication information to a remote re-authentication server,
allowing a peer to re-authenticate locally, without having to
communicate with its home re-authentication server.

These problems are the primary issues to be resolved. In solving
them, there are a number of constraints to conform to, and those
result in some additional work to be done in the area of EAP keying.

4. Design Goals

The following are the goals and constraints in designing the EAP re-
authentication and key management protocol:

Lower-latency operation: The protocol MUST be responsive to handover
   and re-authentication latency performance objectives within a
   mobile access network. A solution that reduces latency as
   compared to a full EAP authentication will be most favorable,
   since in networks that rely on reactive re-authentication this
   will directly impact handover times.

EAP lower-layer independence: Any keying hierarchy and protocol
   defined MUST be lower-layer independent in order to provide
   capabilities over heterogeneous technologies. The defined
   protocols MAY require some additional support from the lower
   layers that use it, but should not require any particular lower
   layer.

   EAP method independence: Changes to existing EAP methods MUST NOT be
      required as a result of the re-authentication protocol. There
      MUST be no requirements imposed on future EAP methods, provided
      they satisfy [EAP-KEYING] and [RFC4017]. Note that the only EAP
      methods for which independence is required are those that
      currently conform to the specifications of [EAP-KEYING] and
      [RFC4017]. In particular, methods that do not generate the keys
      required by [EAP-KEYING] need not be supported by the re-
      authentication protocol.

   AAA protocol compatibility and keying: Any modifications to EAP and
      EAP keying MUST be compatible with RADIUS [RADEXT-DESIGN] and
      Diameter [DIME-APP-DESIGN]. Extensions to both RADIUS and
      Diameter to support these EAP modifications are acceptable. The
      designs and protocols must be configurable to satisfy the AAA key
      management requirements specified in RFC 4962 [RFC4962].

   Compatibility: Compatibility and coexistence with compliant
      ([RFC3748] [EAP-KEYING]) EAP deployments MUST be provided.
      Specifically, the protocol should be designed such that a peer not
      supporting fast re-reauthentication should still function in a
      network supporting fast re-authentication, and also a peer
      supporting fast re-authentication should still function in a
      network not supporting fast re-authentication.

   Cryptographic Agility: Any re-authentication protocol MUST support
      cryptographic algorithm agility, to avoid hard-coded primitives
      whose security may eventually prove to be compromised. The
      protocol MAY support cryptographic algorithm negotiation, provided
      it does not adversely affect overall performance (i.e., by
      requiring additional round trips).

   Impact to Existing Deployments: Any re-authentication protocol MAY
      make changes to the peer, authenticator, and EAP server, as
      necessary to meet the aforementioned design goals. In order to
      facilitate protocol deployment, protocols should seek to minimize
      the necessary changes, without sacrificing performance.

5. Security Goals

   This section draws from the guidance provided in [RFC4962] to further
   define the security goals to be achieved by a complete re-
   authentication keying solution.

5.1. Key Context and Domino Effect

   Any key must have a well-defined scope and must be used in a specific
   context and for the intended use. This specifically means the
   lifetime and scope of each key must be defined clearly so that all
   entities that are authorized to have access to the key have the same
   context during the validity period. In a hierarchical key structure,
   the lifetime of lower-level keys must not exceed the lifetime of
   higher-level keys. This requirement may imply that the context and
   the scope parameters have to be exchanged. Furthermore, the
   semantics of these parameters must be defined to provide proper
   channel binding specifications. The definition of exact parameter
   syntax definition is part of the design of the transport protocol
   used for the parameter exchange, and that may be outside scope of
   this protocol.

   If a key hierarchy is deployed, compromising lower-level keys must
   not result in a compromise of higher-level keys that were used to
   derive the lower-level keys. The compromise of keys at each level
   must not result in compromise of other keys at the same level. The
   same principle applies to entities that hold and manage a particular
   key defined in the key hierarchy. Compromising keys on one
   authenticator must not reveal the keys of another authenticator.
   Note that the compromise of higher-level keys has security
   implications on lower levels.

   Guidance on parameters required, caching, and storage and deletion
   procedures to ensure adequate security and authorization provisioning
   for keying procedures must be defined in a solution document.

   All the keying material must be uniquely named so that it can be
   managed effectively.

5.2. Key Freshness

   As [RFC4962] defines, a fresh key is one that is generated for the
   intended use. This would mean the key hierarchy must provide for
   creation of multiple cryptographically separate child keys from a
   root key at higher level. Furthermore, the keying solution needs to
   provide mechanisms for refreshing each of the keys within the key
   hierarchy.

5.3. Authentication

   Each handover keying participant must be authenticated to any other
   party with whom it communicates to the extent it is necessary to
   ensure proper key scoping, and securely provide its identity to any
   other entity that may require the identity for defining the key
   scope.

5.4. Authorization

   The EAP Key management document [EAP-KEYING] discusses several
   vulnerabilities that are common to handover mechanisms. One
   important issue arises from the way the authorization decisions might
   be handled at the AAA server during network access authentication.
   Furthermore, the reasons for making a particular authorization
   decision are not communicated to the authenticator. In fact, the
   authenticator only knows the final authorization result. The
   proposed solution must make efforts to document and mitigate
   authorization attacks.

5.5. Channel Binding

   Channel Binding procedures are needed to avoid a compromised
   intermediate authenticator providing unverified and conflicting
   service information to each of the peer and the EAP server. To
   support fast re-authentication, there will be intermediate entities
   between the peer and the back-end EAP server. Various keys need to
   be established and scoped between these parties and some of these
   keys may be parents to other keys. Hence, the channel binding for
   this architecture will need to consider layering intermediate
   entities at each level to make sure that an entity with a higher
   level of trust can examine the truthfulness of the claims made by
   intermediate parties.

5.6. Transport Aspects

   Depending on the physical architecture and the functionality of the
   elements involved, there may be a need for multiple protocols to
   perform the key transport between entities involved in the handover
   keying architecture. Thus, a set of requirements for each of these
   protocols, and the parameters they will carry, must be developed.

   The use of existing AAA protocols for carrying EAP messages and
   keying material between the AAA server and AAA clients that have a
   role within the architecture considered for the keying problem will
   be carefully examined. Definition of specific parameters, required
   for keying procedures and for being transferred over any of the links

in the architecture, are part of the scope. The relation between the
identities used by the transport protocol and the identities used for
keying also needs to be explored.

6. Use Cases and Related Work

   In order to further clarify the items listed in scope of the proposed
   work, this section provides some background on related work and the
   use cases envisioned for the proposed work.

6.1. Method-Specific EAP Re-Authentication

   A number of EAP methods support fast re-authentication. In this
   section, we examine their properties in further detail.

   EAP-SIM [RFC4186] and EAP-AKA [RFC4187] support fast re-
   authentication, bootstrapped by the keys generated during an initial
   full authentication. In response to the typical EAP-Request/
   Identity, the peer sends a specially formatted identity indicating a
   desire to perform a fast re-authentication. A single round-trip
   occurs to verify knowledge of the existing keys and provide fresh
   nonces for generating new keys. This is followed by an EAP success.
   In the end, it requires a single local round trip between the peer
   and authenticator, followed by another round trip between the peer
   and EAP server. AKA is based on symmetric-key cryptography, so
   processing latency is minimal.

   EAP-TTLS [EAP-TTLS] and PEAP (Protected EAP Protocol)
   [JOSEFSSON-PPPEXT] support using TLS session resumption for fast re-
   authentication. During the TLS handshake, the client includes the
   message ID of the previous session he wishes to resume, and the
   server can echo that ID back if it agrees to resume the session.
   EAP-FAST [RFC4851] also supports TLS session resumption, but
   additionally allows stateless session resumption as defined in
   [RFC5077]. Overall, for all three protocols, there are still two
   round trips between the peer and EAP server, in addition to the local
   round trip for the Identity request and response.

   To improve performance, fast re-authentication needs to reduce the
   number of overall round trips. Optimal performance could result from
   eliminating the EAP-Request/Identity and EAP-Response/Identity
   messages observed in typical EAP method execution, and allowing a
   single round trip between the peer and a local re-authentication
   server.

6.2. IEEE 802.11r Applicability

   One of the EAP lower layers, IEEE 802.11 [IEEE.802-11R-D9.0], is in
   the process of specifying a fast handover mechanism. Access Points
   (APs) are grouped into mobility domains. Initial authentication to
   any AP in a mobility domain requires execution of EAP, but handover
   between APs within the mobility domain does not require the use of
   EAP.

   Internal to the mobility domain are sets of security associations to
   support key transfers between APs. In one model, relatively few
   devices, called R0-KHs, act as authenticators. All EAP traffic
   traverses an R0-KH, and it derives the initial IEEE 802.11 keys. It
   then distributes cryptographically separate keys to APs in the
   mobility domain, as necessary, to support the client mobility. For a
   deployment with M designated R0-KHs and N APs, this requires M*N
   security associations. For small M, this approach scales reasonably.
   Another approach allows any AP to act as an R0-KH, necessitating a
   full mesh of N2 security associations, which scales poorly.

   The model that utilizes designated R0-KHs is architecturally similar
   to the fast re-authentication model proposed by HOKEY. HOKEY,
   however, allows for handover between authenticators. This would
   allow an IEEE 802.11r-enabled peer to handover from one mobility
   domain to another without performing an EAP authentication.

6.3. CAPWAP Applicability

   The CAPWAP (Control and Provisioning of Wireless Access Points)
   protocol [CAPWAP-PROTOCOL-SPEC] allows the functionality of an IEEE
   802.11 access point to be split into two physical devices in
   enterprise deployments. Wireless Termination Points (WTPs) implement
   the physical and low-level Media Access Control (MAC) layers, while a
   centralized Access Controller (AC) provides higher-level management
   and protocol execution. Client authentication is handled by the AC,
   which acts as the AAA authenticator.

   One of the many features provided by CAPWAP is the ability to roam
   between WTPs without executing an EAP authentication. To accomplish
   this, the AC caches the MSK from an initial EAP authentication, and
   uses it to execute a separate four-way handshake with the station as
   it moves between WTPs. The keys resulting from the four-way
   handshake are then distributed to the WTP to which the station is
   associated. CAPWAP is transparent to the station.

   CAPWAP currently has no means to support roaming between ACs in an
   enterprise network. The proposed work on EAP efficient re-
   authentication addresses is an inter-authenticator handover problem

from an EAP perspective, which applies during handover between ACs.
Inter-AC handover is a topic yet to be addressed in great detail and
the re-authentication work can potentially address it in an effective
manner.

7. Security Considerations

This document details the HOKEY problem statement. Since HOKEY is an
authentication protocol, there is a myriad of security-related issues
surrounding its development and deployment.

In this document, we have detailed a variety of security properties
inferred from [RFC4962] to which HOKEY must conform, including the
management of key context, scope, freshness, and transport;
resistance to attacks based on the domino effect; and authentication
and authorization. See Section 5 for further details.

8. Contributors

This document represents the synthesis of two problem statement
documents. In this section, we acknowledge their contributions, and
involvement in the early documents.

   Mohan Parthasarathy
   Nokia
   EMail: mohan.parthasarathy@nokia.com

   Julien Bournelle
   France Telecom R&D
   EMail: julien.bournelle@orange-ftgroup.com

   Hannes Tschofenig
   Siemens
   EMail: Hannes.Tschofenig@siemens.com

   Rafael Marin Lopez
   Universidad de Murcia
   EMail: rafa@dif.um.es

9. Acknowledgements

The authors would like to thank the participants of the HOKEY working
group for their review and comments including: Glen Zorn, Dan
Harkins, Joe Salowey, and Yoshi Ohba. The authors would also like to
thank those that provided last-call reviews including: Bernard Aboba,
Alan DeKok, Jari Arkko, and Hannes Tschofenig.

10. References

10.1. Normative 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.

   [RFC4017]   Stanley, D., Walker, J., and B. Aboba,
                           "Extensible Authentication Protocol (EAP)
                           Method Requirements for Wireless LANs",
                           RFC 4017, March 2005.

   [RFC4962]   Housley, R. and B. Aboba, "Guidance for
                           Authentication, Authorization, and Accounting
                           (AAA) Key Management", BCP 132, RFC 4962,
                           July 2007.

10.2. Informative References

   [CAPWAP-PROTOCOL-SPEC] Calhoun, P., Montemurro, M., and D. Stanely,
                           "CAPWAP Protocol Specification", Work
                           in Progress, March 2008.

   [DIME-APP-DESIGN]   Fajardo, V., Asveren, T., Tschofenig, H.,
                           McGregor, G., and J. Loughney, "Diameter
                           Applications Design Guidelines", Work
                           in Progress, January 2008.

   [EAP-KEYING]   Aboba, B., Simon, D., and P. Eronen,
                           "Extensible Authentication Protocol (EAP) Key
                           Management Framework", Work in Progress,
                           November 2007.

   [EAP-TTLS]   Funk, P. and S. Blake-Wilson, "EAP Tunneled
                           TLS Authentication Protocol Version 0 (EAP-
                           TTLSv0)", Work in Progress, March 2008.

   [IEEE.802-11R-D9.0]   "Information technology - Telecommunications
                          and information exchange between systems -
                          Local and metropolitan area networks -
                          Specific requirements - Part 11: Wireless LAN
                          Medium Access Control (MAC) and Physical
                          Layer (PHY) specifications - Amendment 2:
                          Fast BSS Transition", IEEE Standard 802.11r,
                          January 2008.

   [JOSEFSSON-PPPEXT]   Josefsson, S., Palekar, A., Simon, D., and G.
                          Zorn, "Protected EAP Protocol (PEAP) Version
                          2", Work in Progress, October 2004.

   [LGS07]   Ledlie, J., Gardner, P., and M. Selter,
                          "Network Coordinates in the Wild",
                          USENIX Symposium on Networked System Design
                          and Implementation, April 2007.

   [MSA03]   Mishra, A., Shin, M., and W. Arbaugh, "An
                          Empirical Analysis of the IEEE 802.11 MAC-
                          Layer Handoff Process", ACM SIGCOMM Computer
                          and Communications Review, April 2003.

   [MSPCA04]   Mishra, A., Shin, M., Petroni, N., Clancy,
                          T., and W. Arbaugh, "Proactive Key
                          Distribution using Neighbor Graphs",
                          IEEE Wireless Communications, February 2004.

   [RADEXT-DESIGN]   Weber, G. and A. DeKok, "RADIUS Design
                          Guidelines", Work in Progress, December 2007.

   [RFC3990]   O'Hara, B., Calhoun, P., and J. Kempf,
                          "Configuration and Provisioning for Wireless
                          Access Points (CAPWAP) Problem Statement",
                          RFC 3990, February 2005.

   [RFC4186]   Haverinen, H. and J. Salowey, "Extensible
                          Authentication Protocol Method for Global
                          System for Mobile Communications (GSM)
                          Subscriber Identity Modules (EAP-SIM)",
                          RFC 4186, January 2006.

   [RFC4187]   Arkko, J. and H. Haverinen, "Extensible
                          Authentication Protocol Method for 3rd
                          Generation Authentication and Key Agreement
                          (EAP-AKA)", RFC 4187, January 2006.

   [RFC4851]   Cam-Winget, N., McGrew, D., Salowey, J., and
                          H. Zhou, "The Flexible Authentication via
                          Secure Tunneling Extensible Authentication
                          Protocol Method (EAP-FAST)", RFC 4851,
                          May 2007.

   [RFC5077]   Salowey, J., Zhou, H., Eronen, P., and H.
                          Tschofenig, "Transport Layer Security (TLS)
                          Session Resumption without Server-Side
                          State", RFC 5077, January 2008.

Authors' Addresses

   T. Charles Clancy, Editor
   Laboratory for Telecommunications Sciences
   US Department of Defense
   College Park, MD
   USA

   EMail: clancy@LTSnet.net

   Madjid Nakhjiri
   Motorola

   EMail: madjid.nakhjiri@motorola.com

   Vidya Narayanan
   Qualcomm, Inc.
   San Diego, CA
   USA

   EMail: vidyan@qualcomm.com

   Lakshminath Dondeti
   Qualcomm, Inc.
   San Diego, CA
   USA

   EMail: ldondeti@qualcomm.com

Full Copyright Statement

Intellectual Property