Kerberized Handover Keying: A Media-Independent Handover Key Management Architecture

Yoshihiro Ohba
Toshiba America Research, Inc
P.O. Box 429, Piscataway
New Jersey 08854-4151, U.S.A.
+1-732-699-5305
yohba@tari.toshiba.com

Subir Das
Telcordia Technologies Inc
One Telcordia Drive, Piscataway
New Jersey 08854, U.S.A.
+1-732-699-2483
subir@research.telcordia.com

Ashutosh Dutta
Telcordia Technologies Inc
One Telcordia Drive, Piscataway
New Jersey 08854, U.S.A.
+1-732-699-3130
adutta@research.telcordia.com

ABSTRACT

This paper proposes a media-independent handover key management architecture that uses Kerberos for secure key distribution among a server, an authenticator, and a mobile node. With the proposed architecture, signaling for key distribution is based on re-keying and is decoupled from re-authentication that requires EAP (Extensible Authentication Protocol) and AAA (Authentication, Authorization and Accounting) signaling similar to initial network access authentication. In this framework, the mobile node is able to obtain master session keys required for dynamically establishing the security associations with a set of authenticators without communicating with them before handover. By separating re-key operation from re-authentication, the proposed architecture is more optimized for proactive mode of operation. It is also optimized for reactive mode of operation by reversing the key distribution roles between the mobile node and the target access node. This paper discusses how the proposed architecture is applicable to the existing link-layer technologies including IEEE 802.11 and 802.16 and across multiple AAA domains. This paper also describes how Kerberos is bootstrapped from initial access authentication using an EAP method.

Categories and Subject Descriptors

H.4.3 [**Communications Applications**]: Network Security Architecture and Protocols.

General Terms

Design, Security, Standardization.

Keywords

Kerberos, Handover, Signaling, Key Distribution, Network Access Authentication.

MobiArch'07, August 27--31, 2007, Kyoto, Japan.
Copyright 2007 ACM 978-1-59593-784-8/07/0008…$5.00.

1. INTRODUCTION

Wireless network technologies are evolving to allow seamless handover across multiple different link-layer technologies. IEEE 802.21[1] is defining a Media-Independent Handover (MIH) Function with unified interface to both link-layer and higher-layer protocols. This function facilitates handover by providing several services to mobility management entity and a protocol for carrying these services to another MIH Function in a remote node. While security signaling optimization during handover is one of the important factors for achieving seamless handover, it is currently out of the scope of IEEE 802.21 specification. Security signaling during handover includes network access authentication and subsequent key management signaling for enabling link-layer ciphering.

The IETF (Internet Engineering Task Force) defines EAP (Extensible Authentication Protocol) [2] that provides a unified mechanism for network access authentication with a support of a variety of authentication methods over several link-layer technologies such as Ethernet, IEEE 802.11 and IEEE 802.16 [3][4][5] as well as over UDP/IP[6][7]. An EAP method is a two-party authentication protocol that runs an authentication method between a peer (e.g., a mobile node) and a server (e.g., a backend authentication server) whereas EAP is just a container for conveying the EAP method through an authenticator (e.g., an access point in case of IEEE 802.11) as shown in Figure 1: EAP with pass-through authenticator.

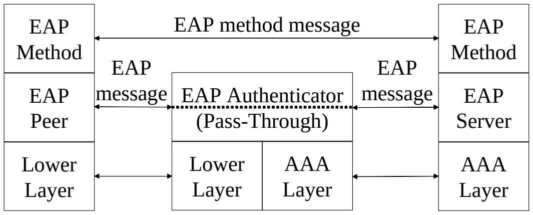


Figure 1: EAP with pass-through authenticator

Although EAP provides a media-independent mechanism for network access authentication, its basic design has not sufficiently taken into account to optimize its signaling when the peer changes one authenticator to another one due to a handover, except for specific EAP methods that address this issue by providing enhancements using session resumption to address this issue [8]. However, such enhancements are not usable when other EAP methods are used for initial network access authentication.

The IETF has recently started the HOKEY (Handover Keying)[9] working group to define mechanisms and protocols for optimizing EAP for handover. The HOKEY WG is defining three components, such as low-latency re-authentication (or HOKEY re-authentication), handover key management and pre-authentication. HOKEY re-authentication defines an extension to EAP to minimize message roundtrips by utilizing keying material generated by a previous EAP session. Handover key management defines a new key hierarchy that spans multiple authenticators as well as a key distribution mechanism from the server to the authenticators. Pre-authentication is a proactive handover optimization technique by which a peer runs EAP for a candidate target authenticator from the serving access network [10].

However, there are two issues on the HOKEY components. First, HOKEY re-authentication still requires a peer to communicate with the server that is typically co-located with an AAA server that resides in the home or a visited operator's network[1] and is likely to be physically away from the peer's location. Thus, it is difficult for HOKEY re-authentication to reduce the handover latency to the extent that does not affect the performance of real-time applications. Second, HOKEY pre-authentication requires an accurate anticipation on movement of the mobile node. However, movement anticipation is difficult when there are a number of candidate authenticators in the neighboring networks.

This paper proposes Kerberized Handover Keying (KHK), a new architecture for handover key management using Kerberos [11] in order to address the issues on HOKEY. KHK is aimed to provide the following features: KHK does not require post-handover AAA signaling for authentication and authorization as long as the mobile node proactively obtains per-authenticator keys. The post-handover signaling latency is expected to be reduced to the order of propagation delay within the access network based on a few message exchanges between the peer and the authenticator for Kerberos ticket installation and execution of a lower-layer secure association protocol. KHK can reduce the size of required key cache for proactive keying operation since each authenticator does not need to store a key for a mobile node before handover.

There is an existing work EAP-GSS [12] that uses Kerberos for network access authentication and key management. The work was initially considered as a candidate for IEEE 802.11i. However, unlike KHK, the work neither considers handover optimization nor allows any EAP method to be used for initial network access authentication.

2. Kerberos Overview

Kerberos [10] is a three-party authentication and key management protocol based on symmetric keys. There are three principals in Kerberos; a client, a server, and a key distribution center (KDC). KDC provides two special servers: an Authentication Server (AS) and a Ticket Granting Server (TGS). It is assumed that each client and server has a pre-established trust relationship with KDC based on a secret key.

In Kerberos, a session key that is used by the client and server to securely establish a session is generated by the KDC and distributed to the client. The client then distributes the session key to the server using a *ticket*, or a record generated by the KDC to help a client authenticate itself to a server. The ticket contains the identity of the client, a session key, a timestamp and other information, where all pieces of information, except for a ticket version number, a realm and a server name, are encrypted using the server's secret key shared only with the KDC. The Kerberos protocol consists of three exchanges where the initial exchange is performed only once. Figure 2: Kerberos sequence shows a typical protocol sequence of Kerberos.

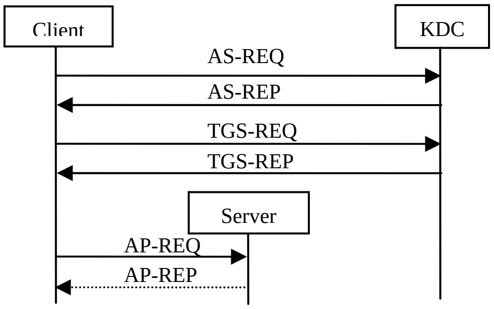


Figure 2: Kerberos sequence

In the initial exchange (AS-REQ/AS-REP exchange), the client requests a Ticket Granting Ticket (TGT), or a special ticket used for generating other tickets, from the AS. The AS generates a TGT, which contains a session key for TGS (a TGS session key), and sends the client the TGT together with a copy of the TGS session key that is encrypted with the secret key shared only with the client.

In the second exchange (TGS-REQ/TGS-REP exchange), the client sends the server's identity and the TGT to the TGS, together with the credentials generated using the TGS session key so that the TGS can verify that the client possesses the same TGS session key. After successful verification of the credentials, the TGS generates a ticket which contains a session key for the server and sends the client the ticket and a copy of the session key that is encrypted with the secret key shared only with the client.

In the third exchange (AP-REQ/AP-REP exchange), the client sends the ticket obtained in the second exchange, together with the message authentication code computed by the client using the session key so that the server can verify that the client possesses the session key. The AP-REP message may be omitted if the client does not need to authenticate the server. After successful verification of the credentials, the client and server are able to use the session key for protecting their application protocol.

[1] An AAA server in the visited operator's network also serves as the home AAA server for subscribers of the operator.

3. Kerberized Handover Keying (KHK)

In KHK, a mobile node and an authenticator (i.e., an access point or a base station) act as a client or a server of Kerberos, where the roles of client and server can be reversed depending on the timing when a ticket is delivered to the authenticator. Proactive mode is the case in which ticket delivery to the authenticator happens before the handover. Reactive mode is the case in which ticket delivery to the authenticator happens after the handover. Proactive mode is more optimized than reactive mode since it does not require for a mobile node to communicate with KDC after handover. In such cases, the signaling latency after handover is expected to be similar to that for IEEE 802.11i 4-way handshake [4] and is known to be less than 20msec [13]. KHK does not require an authenticator to create any state for a mobile node before handover even in proactive mode.

Initially, the mobile node obtains the identity of the KDC and the secret key shared with the AS during initial network access authentication by using the bootstrapping mechanism as described in Section 4.

After initialization, the three steps of Kerberos explained in Section 2 are executed. Those three steps are executed differently in proactive and reactive modes. The detailed differences are explained in Sections 3.1 and 3.2.

3.1 Proactive Mode

In this section, proactive mode is explained using Figure 3: KHK proactive mode.

First, the mobile node (MN) runs an AS-REQ/AS-REP exchange with the KDC to obtain a TGT.

When the MN discovers one or more authenticators (e.g., events D1 and D2) by using authenticator discovery mechanisms as described in Section 3.4, it runs a TGS-REQ/TGS-REP exchange with the KDC to obtain a ticket for each discovered authenticator (A1 and A2). When the MN makes a handover to one of the discovered authenticators (e.g., event H1), it runs an AP-REQ/AP-REP exchange with the authenticator (AP-REP message is optional). When the MN makes another handover to a different authenticator (e.g., event H2), it runs an AP-REQ/AP-REP exchange with the authenticator. In this case, the mobile node and the target authenticator act as a client and a server of Kerberos, respectively.

After the AP-REQ/AP-REP exchange, one or more additional KRB-SAFE messages or link-layer specific SAP (Secure Association Protocol) messages are exchanged between the mobile node and the target authenticator to establish a link-layer security association between the mobile node and the authenticator. These messages carry link-layer specific parameters such as link-layer ciphersuites parameters. The use of KRB-SAFE messages allows the architecture to be independent of link-layer technologies; each link-layer technology only needs to define a Kerberos transport between a mobile node and an authenticator as well as the format and semantics of the link-layer specific parameters to be carried in KRB-SAFE messages.

In the case of IEEE 802.11, link-layer authentication frames can be used as the Kerberos transport between a mobile node and an authenticator, and 802.11i 4-way handshake is used as the SAP instead of KRB-SAFE messages. Similarly, in the case of IEEE 802.16, new PKM (Privacy Key Management) message types can be defined to carry Kerberos messages between a mobile station and a base station, and PKM 3-way handshake is used as the SAP instead of KRB-SAFE messages. In both the cases it requires modifications to the link-layer specifications.

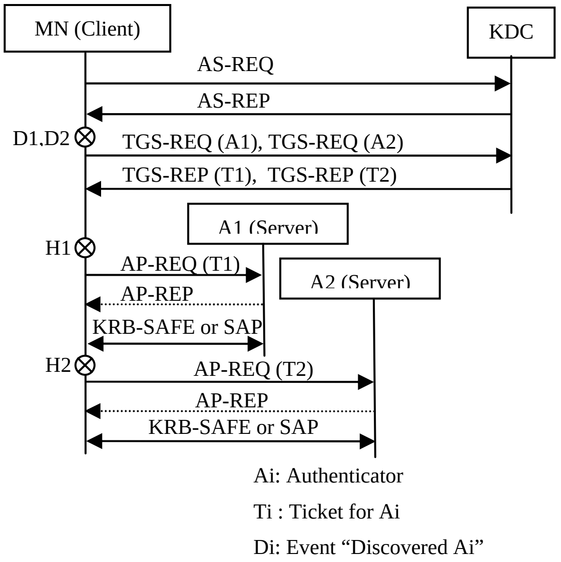


Figure 3: KHK proactive mode

3.2 Reactive Mode

In this section, reactive mode is explained using Figure 4: KHK reactive mode.

First, the mobile node (MN) runs an AS-REQ/AS-REP exchange with the KDC to obtain a TGT, in the same way as proactive mode.

In reactive mode, the Kerberos roles of the mobile node and the target authenticator are reversed, i.e., the mobile node and the target authenticator act as a server and a client, respectively. After a handover to the target authenticator, the mobile node first sends a trigger message to the target authenticator. The target authenticator then runs a TGS-REQ/TGS-REP exchange with the KDC to obtain a ticket for the mobile node and then runs an AP-REQ/AP-REP exchange with the mobile node (AP-REP message is mandatory to authenticate the MN before KRB-safe or SAP exchange).

After the AP-REQ/AP-REP exchange, one or more additional KRB-SAFE messages or link-layer specific SAP (Secure Association Protocol) messages are exchanged between the mobile node and the target authenticator to establish a link-layer security association between the mobile node and the authenticator. This exchange is performed in a similar fashion as the proactive mode.

Since the trigger message is unprotected, a resource consumption DoS (Denial of Service) attack is possible for reactive mode. An additional mechanism may be needed to mitigate such a DoS attack.

The signaling latency after handover for reactive mode is expected to be the one for proactive mode plus one round trip between the authenticator and the KDC. Therefore, the handover performance for real-time applications for reactive mode depends on the location of the KDC. As described in Section 3.6, the KDC can be placed at a location closer to the authenticators using cross-realm operation. However, use of proactive mode is more recommended whenever possible than reactive mode for which the handover performance does not depend on the location of the KDC.

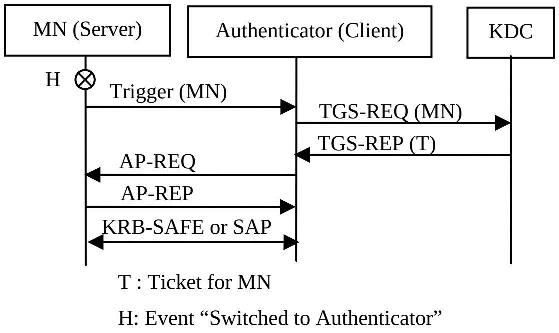


Figure 4: KHK reactive mode

3.3 Key Lifetime

Since a Kerberos ticket contains a key lifetime, it is possible to assign different key lifetimes (or different authorization lifetimes if the key lifetime is same as the authorization time) for different authenticators depending on (but not limited to) link-layer type and location, to provide flexibility in key management for heterogeneous link-layer technologies.

3.4 Authenticator Discovery

In proactive mode of KHK, the mobile node needs to discover authenticators in neighboring networks. An authenticator discovery mechanism needs to provide at least the following information.

- Discovering the identity of the authenticator so that the mobile node can identify it when obtaining a ticket for the authenticator from the KDC. Note that a single authenticator identity may be used for multiple network points of attachment (e.g., access points, base stations or access routers).
- Discovering an address of the authenticator so that the mobile node can communicate with the authenticator for an AP-REQ/AP-REP exchange. The address may be a link-layer address or an IP address. When a single authenticator identity is used for multiple network points of attachments, the authenticator identity is associated with multiple addresses.

In IEEE 802.11r [14], R0KH-ID and BSSID advertised in Beacon frame correspond to the identity and an address of the authenticator, respectively, but it is applicable to IEEE 802.11 link-layer only. On the other hand, a media-independent authenticator discovery mechanism is highly demanded to provide all pieces of information described above. IEEE 802.21 Information Service [1] is considered to be such a mechanism because it provides various pieces of information on neighboring networks to facilitate handover decision making process and is designed to work over any media.

3.5 Authorization and Accounting

Kerberos allows authorization information to be embedded in a ticket's authorization data when encapsulated by the KDC-issued authorization data element. If the authorization credentials issued by the KDC contain the entire authorization information that is needed by the authenticator to perform access control, it is possible to eliminate AAA signaling after handover not only for authentication but also for authorization.

The authorization model used for Kerberized handover keying is described as follows:

- The node that implements KDC also implements AAA client to communicate with an AAA server for authorization purpose.
- When the KDC receives a TGS-REQ message from a Kerberos client (i.e., a mobile node in proactive mode or an authenticator in reactive mode), it asks the AAA client for authorization information for the Kerberos client. The AAA client obtains the authorization information from the AAA server using an AAA protocol and returns the obtained information to the KDC.
- The KDC embeds the authorization information in the authorization data field of the ticket to be contained in a TGS-REP message.

There are two requirements for this authorization model to work. First, the format and semantics of authorization credentials need to be standardized for interoperability. Second, in the case of reactive mode, the authorization information needs to be carried in the encrypted data part of TGS-REP message so that the Kerberos client in reactive mode (i.e., an authenticator) is able to obtain the authorization information to be used for the mobile node.

Accounting is performed on the authenticator as is done in existing link-layer technologies, i.e., the node that implements authenticator also implements AAA client for accounting. In order to associate accounting records with an appropriate authorization session, an authorization session identifier needs to be contained in the authorization credentials. The AAA interaction for authorization and accounting is illustrated in Figure 5: AAA interaction for authorization and accounting. The difference between proactive mode and reactive mode with regard to authorization and accounting is that the KDC indirectly passes the authorization information to the authenticator via the mobile node in proactive mode whereas the KDC directly passes authorization information to the authenticator in reactive mode.

Note that the AAA client on the authenticator node also has authentication functionality for initial network access.

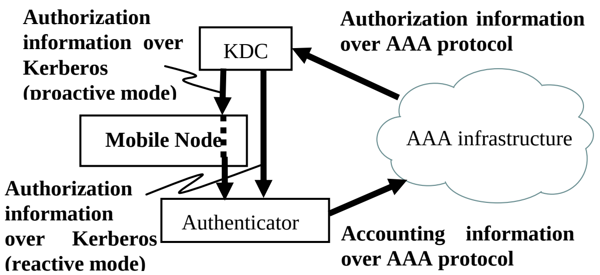

Authorization information over Kerberos (proactive mode)
KDC
Authorization information over AAA protocol
Mobile Node
AAA infrastructure
Authorization information over Kerberos (reactive mode)
Authenticator
Accounting information over AAA protocol

Figure 5: AAA interaction for authorization and accounting

3.6 Mapping Kerberos realms to AAA domains

Kerberos is designed to operate across organizational boundaries. A client in one organization can be authenticated to a server in another. Each organization wishing to run a Kerberos server establishes its own "realm". The name of the realm in which a client is registered is referred to as the local realm.

By establishing "inter-realm" keys, the administrators of two realms can allow a client authenticated in the local realm to prove its identity to the servers in other realms. The exchange of inter-realm keys registers the ticket-granting service of each realm as a principal in the other realm. A client is then able to obtain a TGT for the remote realm's ticket-granting service from its local realm. When that TGT is used, the remote ticket-granting service uses the inter-realm key (which usually differs from its own normal TGS key) to decrypt the TGT. Tickets issued by the remote ticket-granting service will indicate to the end-service that the client was authenticated from another realm.

In general, Kerberos realms and AAA domains are independent. However, for simplicity, we introduce an operationally reasonable model. We assume that Kerberized handover keying uses DNS domain name as Kerberos realm name and AAA domain name.

Let D(n) denote a AAA domain whose DNS domain name is n. Let Rn be a set of Kerberos realms for which the realm name contain n in their suffix. The relationship between an AAA domain and Kerberos realms in KHK is represented as follows.

$$D(n) = Rn.$$

This means that an AAA domain consists of a set of Kerberos realms and that a mobile node can tell whether a particular Kerberos realm belongs to a particular AAA domain by using the name of the Kerberos realm and the name of the AAA domain. An example mapping between an AAA domain and Kerberos realms is shown in Figure 6: Example relationship between AAA domain and Kerberos realms. Defining multiple realms within a single AAA domain is important for KHK to be scalable as well as to reduce signaling latency by placing a KDC at a location physically close to the mobile node as much as possible.

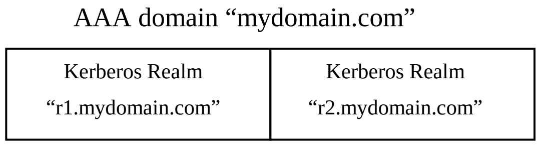


Figure 6: Example relationship between AAA domain and Kerberos realms

To support seamless handover across AAA domains, we also assume that there are pre-established Kerberos inter-realm keys between two AAA domains that have a roaming relationship with each other. When a mobile node moves from the serving authenticator to the target authenticator across AAA domains, it acquires a cross-realm TGT valid for the remote KDC in the visited AAA domain by contacting the local KDC in the home AAA domain. If there are one or more intermediate realms between the local and remote realms, the mobile node iterates the TGT acquisition procedure along the authentication path. The authorization credentials generated by the local KDC need to be preserved in the cross-realm TGT used for the remote KDC. Since the two AAA domains typically belong to different branches of the DNS domain hierarchy, the determination process of the authentication path is not trivial. In this case, the authentication path may be dynamically resolved using referrals of KDCs as specified in [15]. A more optimized mechanism that eliminates the iteration of the TGT acquisition procedure also exists [16].

Cross-realm operation of Kerberos is also possible for handover within the same AAA domain. When the mobile node moves from the serving authenticator to the target authenticator across Kerberos realms within the same AAA domain, it acquires a cross-realm TGT valid for the target KDC (i.e., the KDC for the target authenticator) by contacting the serving KDC (i.e., the KDC for the serving authenticator). In case there are one or more intermediate realms between the two KDCs in the same AAA domain, the mobile node iterates the TGT acquisition procedure along the authentication path. The authorization credentials generated by the serving KDC need to be preserved in the cross-realm TGT used for the target KDC. The authentication path may be constructed based on the DNS domain hierarchy, which makes the traversal of the authentication path easier than the inter-domain case.

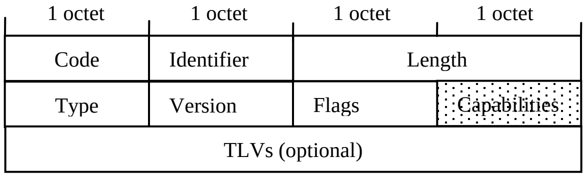
In proactive mode, the iteration of the TGT acquisition procedure needed for cross-realm operation is performed by the target authenticator instead of the mobile node. This effectively reduces Kerberos message exchanges over the link between the mobile node and the authenticator.

4. Bootstrapping Kerberos from EAP

To support roaming among multiple AAA domains, it is required to define a mechanism to dynamically configure the principal name of the local KDC and the secret key used for it as much as possible. For this purpose, we propose a mechanism to bootstrap Kerberos from network access authentication credentials using EAP-EXT[17] which is a new EAP method.

EAP-EXT is a tunneling method that encapsulates any EAP authentication method and provides capabilities negotiation by which newly defined functionality can be enabled. EAP-EXT provides backward compatibility to the existing EAP authentication methods, as it makes the new functionality available while still using existing EAP methods without any modification to them.

EAP-EXT currently defines two capabilities, i.e., channel binding and re-authentication. We propose to add a new capability on Kerberos to EAP-EXT. EAP-EXT message format with an additional capability bit (i.e., 'K' bit) for Kerberos bootstrapping is shown in Figure 7: EAP-EXT message format with Kerberos bootstrapping capability.



1 octet	1 octet	1 octet	1 octet
Code	Identifier	Length	
Type	Version	Flags	Capabilities
TLVs (optional)			

Capabilities:

Bit 0: 'R' bit (Re-authentication)

Bit 1: 'C' bit (Channel Binding)

Bit 2: 'K' bit (Kerberos)

Figure 7: EAP-EXT message format with Kerberos bootstrapping capability

When both the peer and server set 'K' bit in the final EAP-EXT message exchange which is integrity protected using the key exported from an inner EAP authentication method, the peer and server bootstrap Kerberos. The following information is required to bootstrap Kerberos and carried in a Kerberos-Boot (KRB-BOOT) TLV (Type-Length-Value) in the final EAP-EXT request message with 'K' bit set.

- The length and the lifetime of the secret key (EAP-KRB-KEY) to be shared between the mobile node and local KDC
- The principal name and realm of the local KDC
- IP address of the local KDC

EAP-KRB-KEY is derived from EMSK (EAP Master Session Key) as a USRK (Usage Specific Root Key) [18] as follows, where KDF denotes a key derivation function defined in [18] and length denotes the length of the derived key.

EAP-KRB-KEY = KDF(EMSK, "EAP-EXT-Kerberos-Boot-Key", length).

The EAP server also installs the information carried in the KRB-BOOT TLV to the local KDC as shown in Figure 8: Kerberos bootstrapping sequence, where "Method" denotes a Method TLV that carries an EAP method payload and AUTH denotes an AUTH TLV that carries integrity protection data.

In order to simplify the Kerberos bootstrapping procedure, it is strongly recommended that the EAP server and the local KDC be implemented on the same node using the same identifier for the EAP server identity and the local KDC. Otherwise, a three-party key distribution protocol would be required for key distribution among the EAP peer, the EAP server and the local KDC to securely transport the secret key.

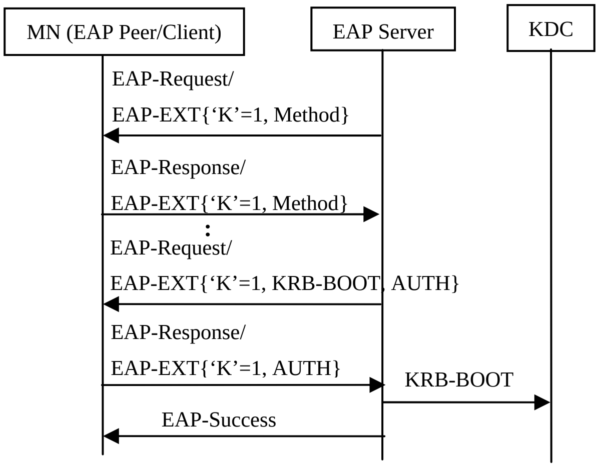


Figure 8: Kerberos bootstrapping sequence

5. Conclusion and Future Work

This paper proposed a new media-independent handover key management architecture using Kerberos to address several issues including those that are under discussion within the IETF HOKEY working group. This paper also discussed how the proposed architecture works across multiple AAA domains and explained how Kerberos is bootstrapped from initial access authentication using an EAP method. It is recommended that network equipment vendors and network operators investigate the cost for deploying KHK. If the technical advantage of KHK and the deployment cost are well balanced, it is further recommended that relevant standard bodies including IETF and IEEE 802 should define a set of protocols required for KHK, including modification to the Kerberos protocol, Kerberos bootstrapping, and link-layer transport of Kerberos.

There are several future work items related to KHK. First, the architecture needs a proof of concept based on an implementation to an existing wireless link-layer technology, especially with the support of inter-domain operations. Second, performance evaluation is needed to show the advantage of the proposed architecture to other secure handover architectures such as HOKEY. Third, more investigations on how the architecture can interwork with IEEE 802.21 handover services are needed. Finally, bootstrapping Kerberos from initial network access authentication as described in Section 4 allows not only bootstrapping KHK but also bootstrapping security for many other applications such as, SSO (Single-Sign On) that may or may not be related to handover. However, further studies on integrating SSO with KHK are required.

6. REFERENCES

[1] "Draft IEEE Standard for Local and Metropolitan Area Networks: Media Independent Handover Services", IEEE LAN/MAN Draft IEEE P802.21/D05.00, April 2007.

[2] B. Aboba, L. Blunk, J. Vollbrecht, J. Carlson and H. Levkowetz, "Extensible Authentication Protocol (EAP)", RFC 3748, June 2004.

[3] "IEEE Standard for Local and Metropolitan Area Networks - Port-Based Network Access Control", IEEE Std 802.1X–2004.

[4] "IEEE Standard for Local and metropolitan area networks - Specific requirements Part 11: Wireless LAN MAC and PHY specifications Amendment 6: Medium Access Control (MAC) Security Enhancements", IEEE 802.11i–2004.

[5] "IEEE Standard for Local and metropolitan area networks Part 16: Air Interface for Fixed and Mobile Broadband Wireless Access Systems Amendment 2", IEEE 802.16e-2005 and IEEE Std 802.16-2004/Cor1-2005.

[6] C. Kaufman, "Internet Key Exchange (IKEv2) Protocol", Dec 2005.

[7] D. Forsberg, Y. Ohba, B. Patil, H. Tschofenig and A. Yegin, "Protocol for Carrying Authentication for Network Access (PANA)", Internet-Draft, work in progress, May 2007.

[8] B. Aboba, D. Simon, "PPP EAP TLS Authentication Protocol", RFC 2716, October 1999.

[9] IETF HOKEY WG charter, http://www.ietf.org/html.charters/hokey-charter.html.

[10] A Dutta, T. Zhang, Y. Ohba, K. Taniuchi and H. Schulzrinne, "MPA assisted Optimized Proactive Handoff Scheme", Mobiquitous 2005.

[11] C. Neuman, T. Yu, S. Hartman and K. Raeburn, "The Kerberos Network Authentication Service (V5)", RFC 4120, July 2005.

[12] B. Aboba, "EAP GSS Authentication Protocol", http://tools.ietf.org/html/draft-aboba-pppext-eapgss-12, August 2002.

[13] R. Lopez, A. Dutta, Y. Ohba, H. Schulzrinne and A. Skarmeta, "Network-Layer Assisted Mechanism for reducing authentication delay during handoff in 802.11 Networks", to appear in Mobiquitous 2007.

[14] "Draft IEEE Standard for Local and metropolitan area networks - Specific requirements Part 11: Wireless LAN MAC and PHY specifications Amendment 2: Fast BSS Transition", IEEE P802.11r/D6.0, May 2007.

[15] K. Raeburn and L. Zhu, "Generating KDC Referrals to Locate Kerberos Realms", Internet-Draft, work in progress, March 2007.

[16] S. Zrelli, Y. Shinoda, S. Sakane, K. Kamada and M. Ishiyama, "XTGSP, the Inter-TGS protocol for cross-realm operations in Kerberos", Internet-Draft, work in progress, March 2007.

[17] Y. Ohba, S. Das and R. Lopez, "An EAP Method for EAP Extension (EAP-EXT)", Internet-Draft, work in progress, March 2007.

[18] J. Salowey, L. Dondeti, V. Narayanan and M. Nakhjiri, "Specification for the Derivation of Usage Specific Root Keys (USRK) from an Extended Master Session Key (EMSK)", Internet-Draft, work in progress, January 2007.