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

Yoshihiro Ohba (yohba@tari.toshiba.com)
Toshiba America Research, Inc. (USA)

Subir Das (subir@research.telcordia.com)
Ashutosh Dutta (adutta@research.telcordia.com)
Telcordia Technologies (USA)

Problem description (1/2)

- Wireless access networks require cryptographic data protection at link-layer
- Enabling cryptographic data protection requires security signaling
- Security signaling takes time, especially for peer-entity authentication in a roaming environment where authentication credentials are stored in a AAA server
  - AAA servers are typically located away from access networks
- IETF HOKEY WG is working on EAP (Extensible Authentication Protocol) signaling optimization based on two approaches
  - Pre-authentication: a proactive handover optimization technique by which a peer runs EAP for a candidate target authenticator from the serving access network
  - Low-latency Re-authentication: an extension to EAP to minimize message roundtrips by utilizing keying material generated by a previous EAP session

Problem description (2/2)

- Pre-authentication applicability is limited to environments where proactive signaling can take effect
  - Optimization for reactive operation is missing

- Low-latency re-authentication still requires communication to an AAA server (in home or visited network) after handover
  - Difficult to support real-time applications

We propose another approach using Kerberos to address these issues

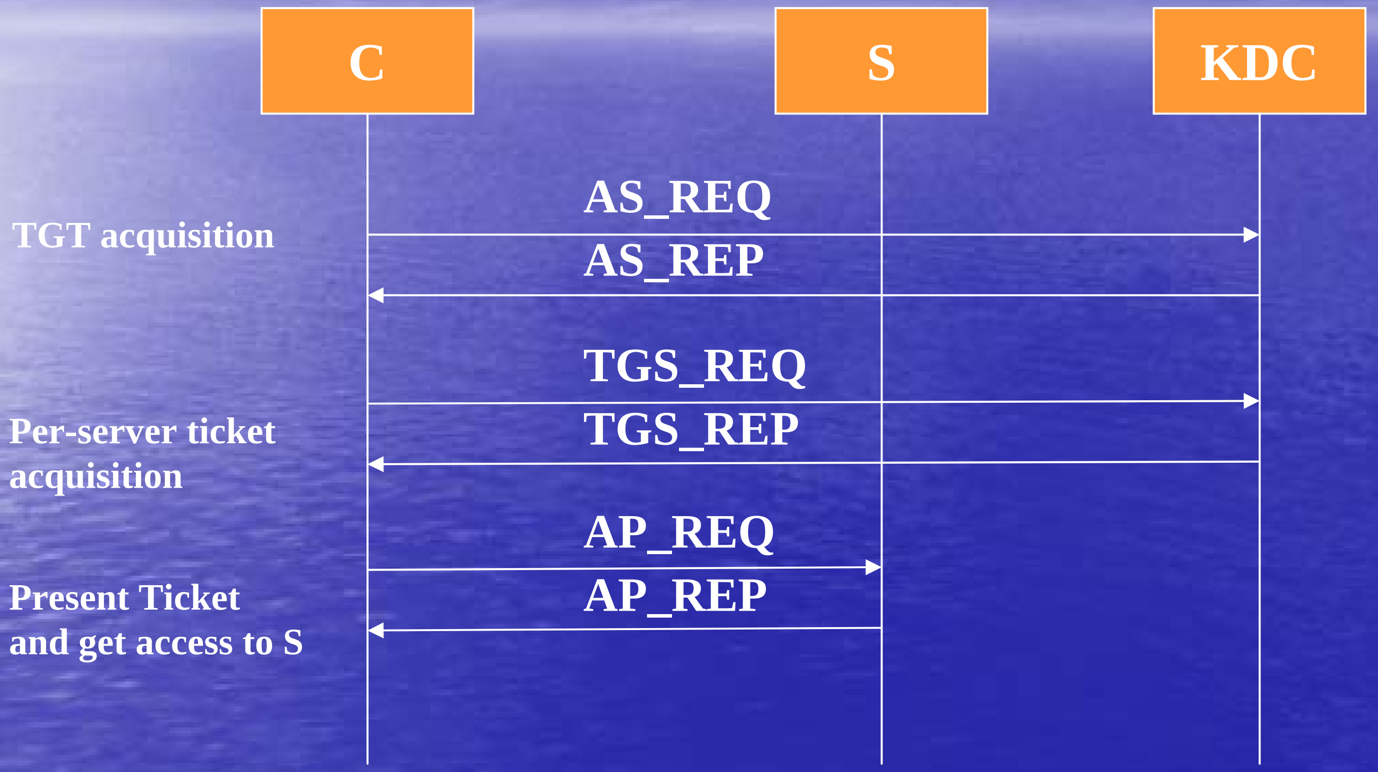
Kerberos overview

- Kerberos 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

Kerberos overview (cont'd)

- In Kerberos, a session key is generated by the KDC and distributed to the client
  - The session key is used by the client and server to securely establish an application session
- 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
  - The session key is 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
  - AS-REQ/AS-REP exchange for acquisition of a TGT (Ticket Granting Ticket)
  - TGS-REQ/TGS-REP exchange for acquisition of a ticket used for the server
  - AP-REQ/AS-REP exchange for installation of the ticket to the server

Kerberos sequence

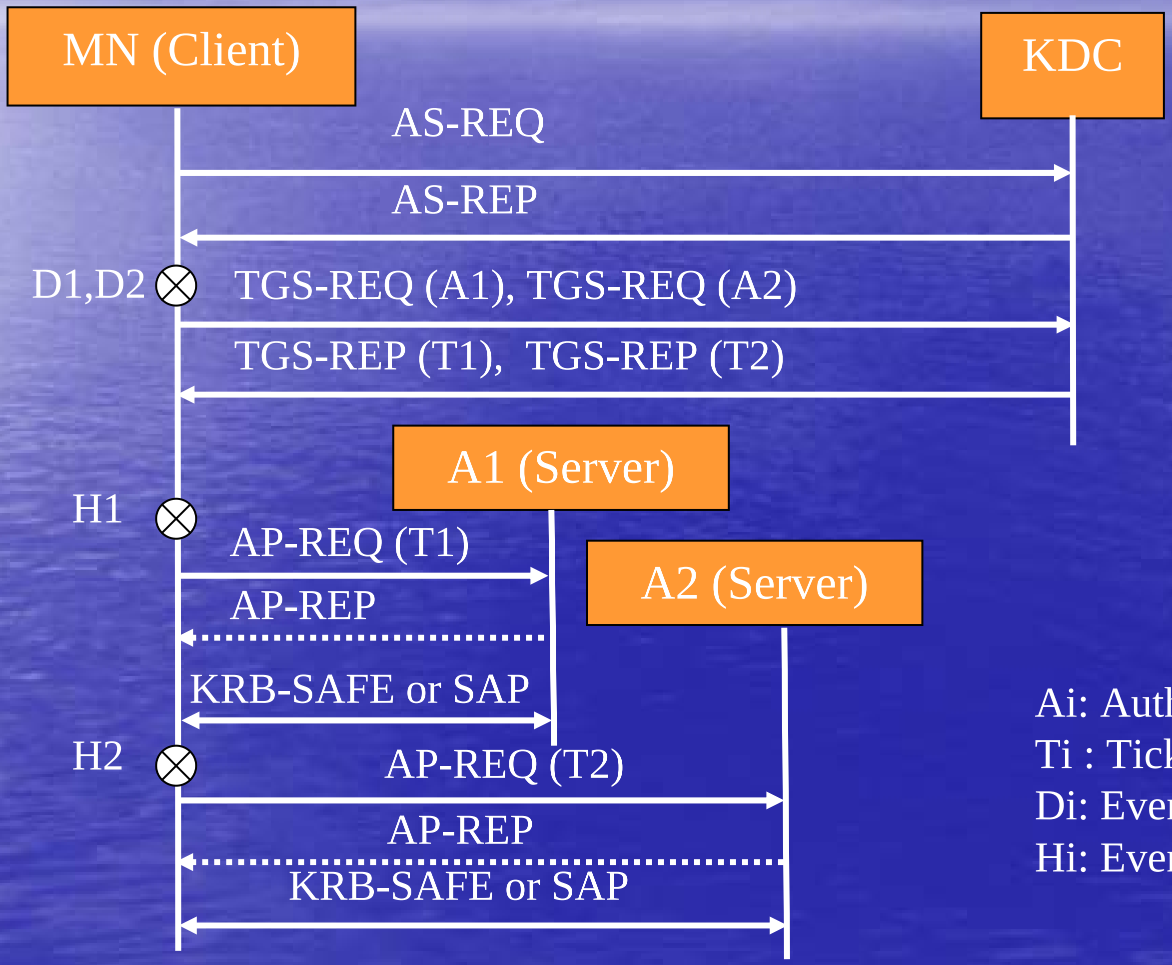


S: Application Server
C: Application Client
KDC: Key Distribution Center

KHK in a nutshell

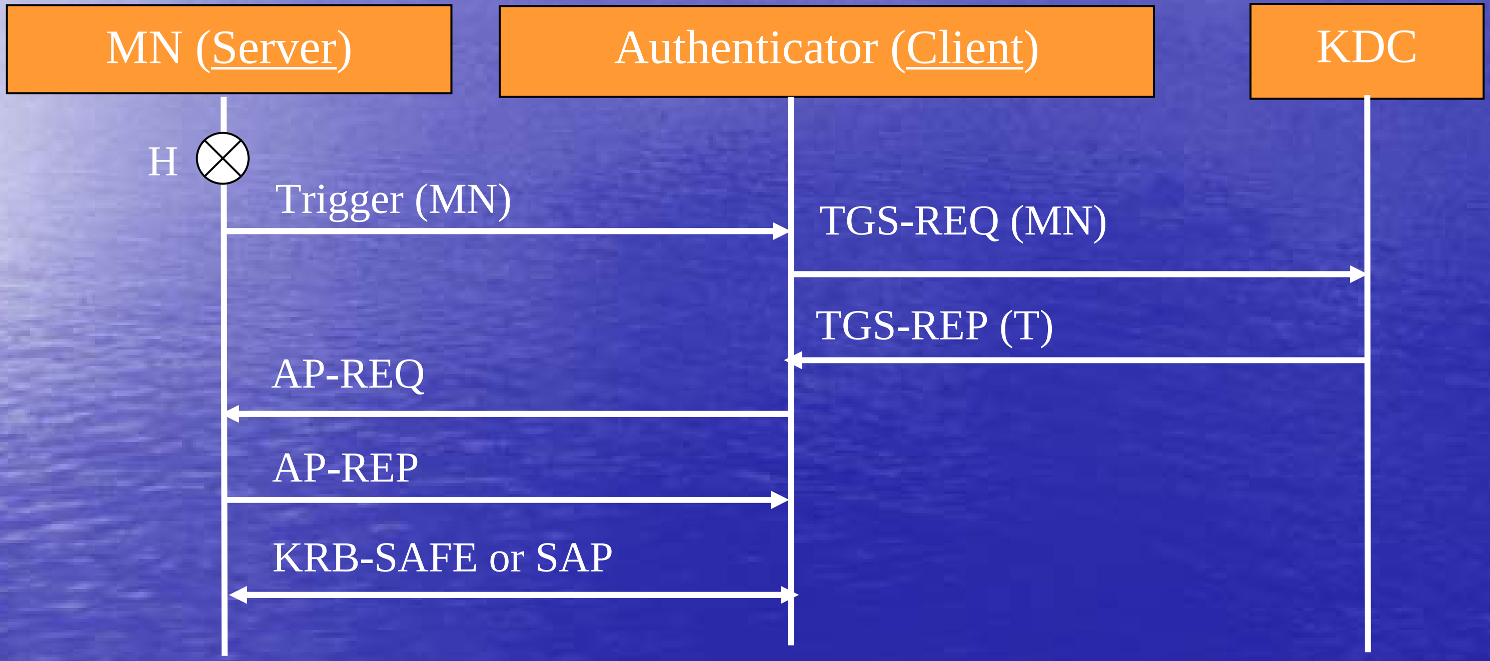
- A mobile node and an authenticator act as a client or a server of Kerberos
- The roles of client and server can be reversed depending on the timing when a ticket is delivered to the authenticator (role reversing)
- Two modes of operation
  - **Proactive mode** is the case in which ticket delivery to the MN happens before the handover
  - **Reactive mode** is the case in which ticket delivery to the MN 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 proactive mode, the signaling latency after handover is expected to be less than 20msec (comparable to IEEE 802.11i 4-way handshake)
- KHK does not require an authenticator to create any state for a mobile node before handover even in proactive mode (i.e., more efficient than pre-authentication)

Proactive mode



Ai: Authenticator
Ti : Ticket for Ai
Di: Event “Discovered Ai”
Hi: Event “Switched to Ai”

Reactive mode



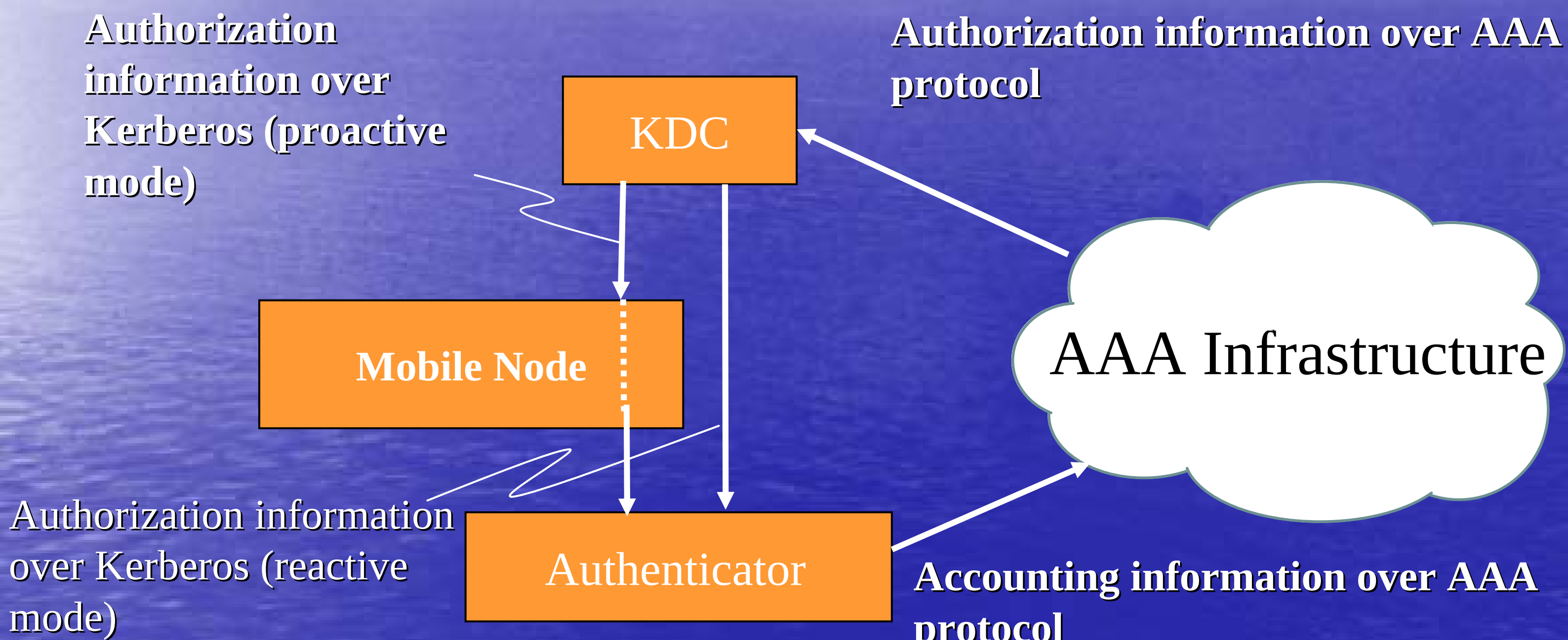
T : Ticket for MN
H: Event “Switched to Authenticator”

Kerberos role is reversed between MN and Authenticator (Role Reversing)

Authorization and accounting

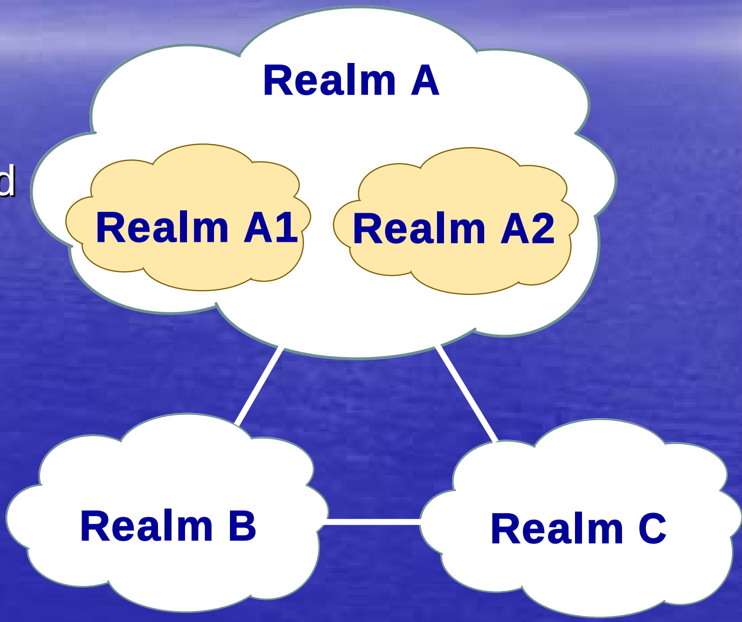
- Kerberos tickets also carry authorization information
- The authorization information must come from AAA
  - KDC needs to be an AAA client <u>for authorization</u>
- Accounting is still performed at each authenticator
  - Authenticators are an AAA client <u>for accounting</u> (as well as initial authentication)

Authorization and accounting (cont'd)



Multi-realm operation in Kerberos

- 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
  - Multi-realm operation addresses the scalability issue
- Realms can be formed hierarchically



Mapping between Kerberos realms and AAA domains

- In general, Kerberos realms and AAA domains are independent
  - However, for simplicity, we introduce an operationally reasonable model
- KHK uses DNS domain name as Kerberos realm name and AAA domain name
- The relationship between an AAA domain and Kerberos realms

$$D(n) = Rn$$

  - D(n) : a AAA domain whose DNS domain name is n
  - Rn : a set of Kerberos realms for which the realm name contain n in their suffix

AAA domain “mydomain.com”

Kerberos Realm “r1.mydomain.com”	Kerberos Realm “r2.mydomain.com”

Bootstrapping Kerberos

- One problem with Kerberos is lack of a mechanism to dynamically create the principal name of the local KDC and the secret key
- A mechanism to dynamically bootstrap Kerberos is needed for KHK to work across multiple AAA domains
- Bootstrapping should be made available from initial network access authentication using EAP

Kerberos bootstrapping using EAP-EXT

- 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
  - No modification to existing EAP methods is needed
- We propose to define a mechanism to bootstrap Kerberos using EAP-EXT

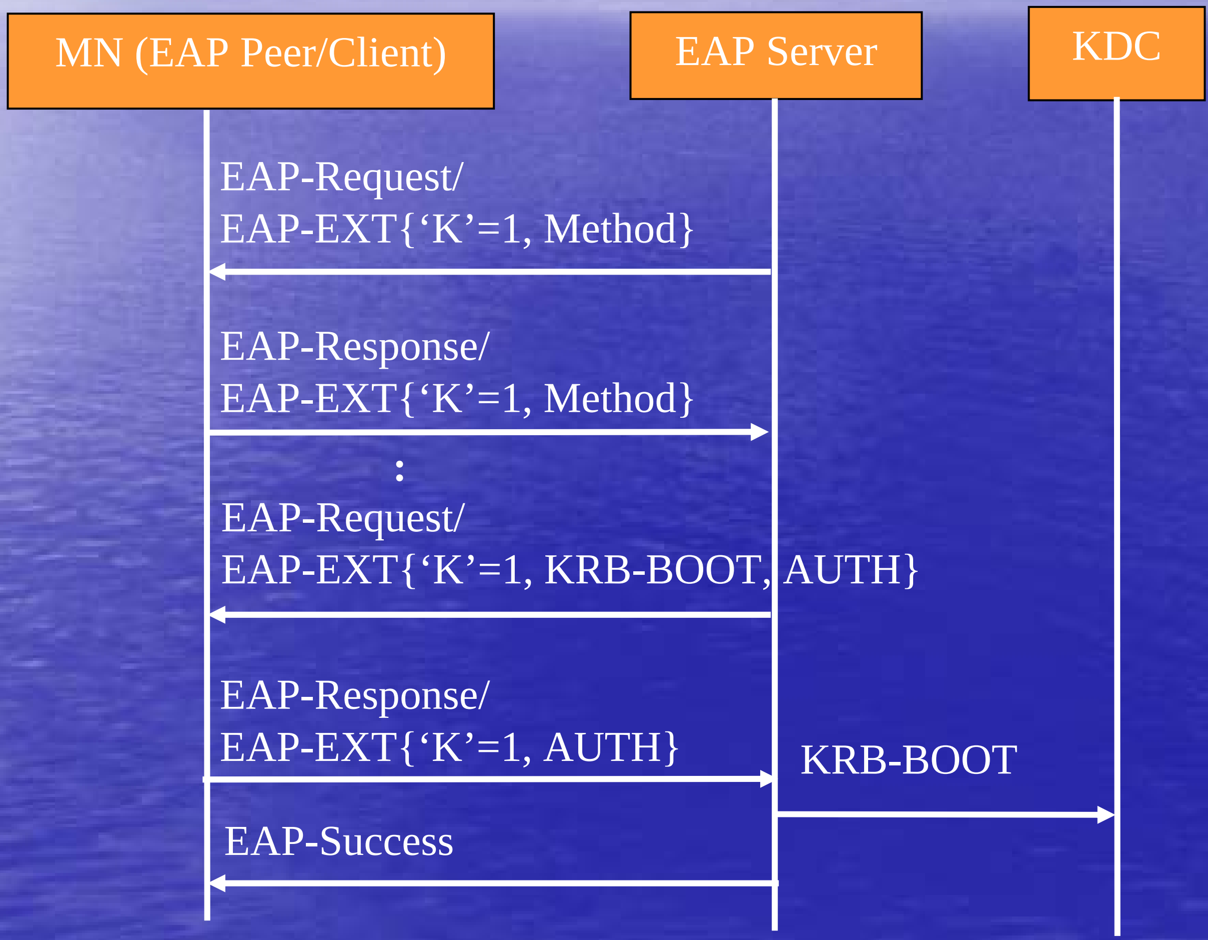
EAP-EXT message format with Kerberos bootstrapping

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)**
- Bits 3-7: Reserved

Kerberos bootstrapping sequence using EAP-EXT



Conclusion and Future Work

- Conclusion
  - We proposed a new media-independent key management architecture using Kerberos to achieve seamless handover across multiple technologies
  - We recommend that network equipment vendors and network operators investigate the cost for deploying KHK
- Future Work
  - A proof of concept based on an implementation to an existing wireless link-layer technology, especially with the support of inter-domain operations
  - Performance evaluation to compare with other secure handover architectures such as HOKEY
  - Investigation on how to interwork with IEEE 802.21
  - Investigation on expanding bootstrapping Kerberos from initial network access authentication to support SSO (Single-Sign On)

Thank you!