Security Challenges in Seamless Mobility —How to "Handover" the Keys?

Katrin Hoeper
National Institute of Standards and Technology (NIST)
Computer Security Division
Gaithersburg, MD, USA
khoeper@nist.gov

Lidong Chen
National Institute of Standards and Technology (NIST)
Computer Security Division
Gaithersburg, MD, USA
llchen@nist.gov

Antonio Izquierdo
National Institute of Standards and Technology (NIST)
Advanced Networking Technologies Division
Gaithersburg, MD, USA
aizquier@nist.gov

Nada Golmie
National Institute of Standards and Technology (NIST)
Advanced Networking Technologies Division
Gaithersburg, MD, USA
ngolmie@nist.gov

ABSTRACT

In this paper, we discuss key management challenges for seamless handovers across heterogeneous wireless networks. We focus on utilizing existing keying material from previous access authentications to expedite network entry. For a seamless handover, keys must be available at the target network at the time of the handover. Currently, industry is still exploring possible ways to handle keys for mobility. This paper identifies the challenges of secure derivation and timely distribution of such keys. We discuss solutions for intra- and inter-technology handovers within the same network and between networks with roaming agreements. The presented solutions include different types of handover key hierarchies and key distribution protocols. In addition, we analyze the tradeoffs between security and performance in the discussed solutions.

Categories and Subject Descriptors

C.2.1 [**Computer-Communication Networks**]: Network Architecture and Design—*Wireless communication* ; E.3 [**Data**]: Data Encryption

Keywords

Security, Key Management, Seamless Mobility

1. INTRODUCTION

The importance of securing wireless network access has been recognized, and media-specific security protocols are widely implemented (e.g., IEEE 802.11i [6] and IEEE 802.16e [8]). The security of a wireless network connection should be maintained when users switch from one point of attachment (PoA) to another, even across heterogeneous networks. Otherwise, mobile users might connect to a rogue PoA or be forced to send confidential data over an unprotected link. In addition, maintaining secure connections with roaming users is of great importance for network providers to ensure proper billing. In general, a secure connection between a mobile node (MN) and a PoA is established through the execution of an access authentication and key establishment protocol between the MN and the authentication server (AS) of the network. For the authentication, MN and AS use long-term authentication credentials that both parties share a priori (such as secret keys or passwords) and/or exchange during the protocol execution (such as public key certificates). The freshly derived keys are then used to protect subsequent communications between the MN and the PoA. In the remainder of this paper, we refer to this process as *full network authentication*.

The term *handover (HO)* originates from cellular networks and describes the switching from the current PoA to a target PoA. Here, all involved network entities share the information about a cellular subscriber's roaming so that existing keys, together with service information, are handed over by switches from one base station to another. However, for some widely deployed non-cellular wireless technologies, such as, IEEE 802.11 [5] and 802.16 [7], such dedicated handover infrastructures do not exist. We observe that "handovers" in such wireless networks is really not the same as the term originally implied when used for cellular networks.

A handover is referred to as a *seamless handover* when a new connection is established before the old one goes down ("make before break"). One of the primary HO challenges is that authentication and key establishment as part of the network access must be executed as quickly as possible such that used services are not disrupted while roaming. To en-

WICON '08, November 17-19, 2008, Maui, Hawaii, USA

Digital Object Identifier: 10.4108/ICST.WICON2008.4790
http://dx.doi.org/10.4108/ICST.WICON2008.4790

able seamless handovers, mobile nodes could initiate a full network access authentication with a target PoA through the current network connection. This method is called *pre-authentication* [12]. For pre-authentication, one of the main challenges is the provision of information necessary to initiate the process sufficiently ahead of time without disrupting the network connection. An approach to expedite secure network entry at the target PoA is to use existing keying material from a previous access authentication in the same network or networks with roaming agreements. This approach is referred to as *re-authentication* [4]. Re-authentications demand secure and timely HO key derivations and distributions such that keying material is available at the target network whenever the mobile user roams.

In this paper, we will discuss the security challenges of HO key management to enable seamless handover in heterogenous wireless environments without handover infrastructures. We limit our discussion to key management for re-authentications, because re-authentications—unlike pre-authentication—introduce new key hierarchies and key distribution issues. We are the first to explore the various aspects of this important problem space, since key management is out of the scope of several standard groups dealing with seamless mobility. Currently, industry is still exploring possible ways to handle keys for mobility applications, and this paper can serve as a guideline for implementers to make the right choices and be aware of the several trade-offs between security and performance. All our discussions are media-independent, but sometimes we use particular wireless technologies, such as IEEE 802.11 and IEEE 802.16 for illustration purposes.

The remainder of the paper is organized as follows. In the next section, we review related work. In Section 3, we summarize the underlying assumptions of our analysis and discussions. In Sections 4-7, we present several challenges related to secure re-authentication and outline solutions of how these challenges can be addressed. The discussed challenges include issues that arise from: the lack of a dedicated key distribution infrastructure; re-using existing keying material to derive HO key hierarchies; performing key updates and maintaining key synchronization throughout networks with roaming agreements; and roaming between networks with different trust models. In Section 8, we analyze security and performance tradeoffs of the presented solutions. Finally, we draw conclusions in the last section.

2. RELATED WORK

Some wireless technology standard groups are currently working on amendments to specify seamless handover solutions (e.g., IEEE 802.11r [9] and IEEE 802.16e). The current solutions define HO key hierarchies and abbreviated network access protocols utilizing the defined HO keys. The drafts are media-specific and, thus, only work in *intra-technology handovers*, i.e. from one PoA to a target PoA which both support the same wireless technology. Since some keying material is re-used, the specified solutions only apply to *intra-domain handovers*, i.e. handovers within one domain or between domains with roaming agreements.

The IETF handover keying working group (HOKEY WG) [1] is currently working on solutions enabling media-independent handovers, also called *inter-technology handovers*. The solutions are applicable to wireless access technologies based on the Extensible Authentication Protocol (EAP) [2]. The IEEE 802.21 [10] security study group plans on adopting the solutions of the HOKEY WG.

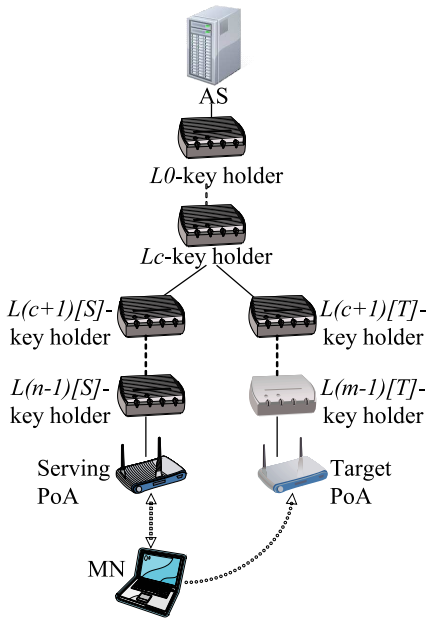

Figure 1: Single Domain Handover

Please notice that the mentioned IEEE standards specify media-dependent protocols between MNs and PoAs and do not consider HO key distribution. While proposed solutions for EAP-based HO key hierarchies seem fairly stable in the HOKEY WG, HO key distribution solutions are still being discussed. This paper is the first to discuss various security-related issues of media-independent key management for enabling re-authentication in heterogenous networks.

3. ASSUMPTIONS

In the remainder of this paper, we make the following assumptions about the considered HO scenarios:

1. *Network Architectures.* We consider two general network architectures: 1) a *single domain* consisting of one authentication server and several key holders connected to the authentication server (as illustrated in Figure 1), and 2) two single domains that have a roaming agreement with each other (as illustrated in Figure 2).

   In the first model a mobile node is currently attached to the serving PoA and plans to switch to a target PoA in the same domain. The serving PoA is connected to the AS through $(n-1)$ intermediate key holders, and the target PoA is connected to the same AS through a different path consisting of $(m-1)$ key holders. We only consider network entities that serve as key holders in the access authentication process and refer to them as $Lx[S]$- and $Lx[T]$-key holders, respectively, where index S denotes key holders in the serving network,

Digital Object Identifier: 10.4108/ICST.WICON2008.4790
http://dx.doi.org/10.4108/ICST.WICON2008.4790

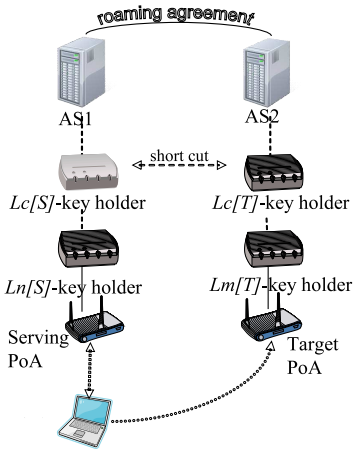

Figure 2: Handover from Domain 1 to Domain 2 with Roaming Agreements, and Optionally Shortcut between $Lc[S]$- and $Lc[T]$-key Holders.

T key holders in the target network, x the key holder level in the network with $x \in [0, n-1]$ for wireless access technology i and $x \in [0, m-1]$ for wireless access technology j, respectively. Serving and target key holder paths to the serving PoA and the target PoA may be the same from AS up to the Lc-key holder (referred to as *lowest common key holder*) and then split into different branches.

In the second model, we consider handovers from a domain 1 (with AS1) to another domain 2 (with AS2), where both domains have roaming agreements. Serving and target networks and the corresponding key holder paths to the respective PoA are disjoint. However, a key holder in the serving network might be able to directly communicate with a key holder in the target network, i.e. without going through the authentication servers. We refer to such a communication path as *short-cut* in the remainder of this paper. A shortcut between $Lc[S]$- and $Lc[T]$-key holders is illustrated in Figure 2.

2. *Existing Key Hierarchy.* We assume that as a result of a previous successful network access authentication, a key hierarchy $\mathbb{S}$ already exists between the MN and the serving PoA. Hierarchies depend on the key holder path (indicated by S for the serving and T for the target network), the wireless access technology (denoted i and j, respectively), and the number of key holder levels (n and m, respectively). For example, the existing key hierarchy $\mathbb{S}$ for wireless access technology i consists of: $RK_i[S]$, $L0K_i[S]$ to $L(n-1)K_i[S]$ and $PTK_i[S]$. Such a key hierarchy is illustrated in Figure 3. Here, the root key $RK_i[S]$ is derived by the MN and the AS upon successful network access. The AS then derives the level 0 key $L0K_i[S]$ for the $L0$-key holder. Upon receiving $L0K_i[S]$, the $L0$-key holder derives a level 1 key $L1K_i[S]$ from $L0K_i[S]$ and sends it to the $L1$-key holder. Key derivation and distribution is repeated until the the lowest key holder in the chain—the PoA—receives $LnK_i[S]$ and derives transient protection keys ($TPK_i[S]$). Latter key is used to derive all necessary keys to protect the wireless link between the MN and the serving PoA. The MN derives all keys in the key hierarchy.

   For the handover, we assume that key hierarchy $\mathbb{S}$ is non-expired and non-compromised.

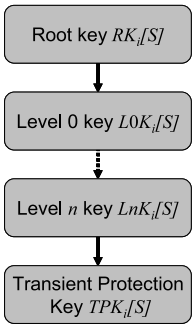

Figure 3: Key Hierarchy $\mathbb{S}$ for Wireless Technology i

3. *Inter Network Communication.* All communications among entities within the backend network are protected.

4. *Multiple Radios.* We assume that roaming MNs support the wireless access technologies of the serving as well as the target PoA. In the case that those technologies are different, i.e. in an inter-technology HO, the MN is able to communicate over these two radios simultaneously.

5. *New Link Security.* We say a seamless handover process is secure if the newly established link between the MN and the target PoA provides the same level of security as a link established between the same entities using a full network authentication. However, in this paper, we only discuss the security of the HO key hierarchy and key distribution and assume that protocols used to establish and protect the new link between the MN and the target PoA are secure.

4. KEY DISTRIBUTION INFRASTRUCTURE

We now identify the challenges of distributing HO keys in non-cellular wireless networks to enable seamless handovers through re-authentication. After that, we outline approaches addressing each of the identified challenges.

4.1 Challenges

As mentioned earlier, non-cellular wireless access networks do not have a dedicated handover infrastructure. As a result, no special entities are available to perform the HO key management, including triggering the distribution of HO keys and their actual distribution. To avoid the high costs associated with deploying new or extending an existing infrastructure, HO key management solutions should utilize already existing network infrastructure. The three questions that need be answered when designing a HO key distribution scheme are discussed in this section.

1. What triggers the key distribution?

Digital Object Identifier: 10.4108/ICST.WICON2008.4790
http://dx.doi.org/10.4108/ICST.WICON2008.4790

Certain roaming information is necessary to trigger the timely key distribution to the correct target network. However, in some wireless networks, network entities are unable to exchange information about mobile nodes to anticipate their roaming behavior.

2. Who distributes the keys?

Once key distribution is triggered, a network entity must distribute HO keys. This so-called *key distributor* must receive the trigger, be able to derive HO keys and distribute them to the target network.

3. How are the keys distributed?

The execution time, as well as the preparation time of the key distribution protocol, should be kept to a minimum. Long delays may result in disconnections or require early HO initiations which may cause unnecessary protocol executions. Furthermore, key distribution protocols should be efficient in terms of imposed communication and computational costs.

4.2 Approaches

4.2.1 Triggers

The network entity triggering the HO key distribution process must have access to real-time information that is sufficient for a timely and accurate prediction of mobile nodes' roaming behavior. However, in some of the networks under consideration, only a mobile node itself is able to predict its own roaming behavior (e.g. based on the signal strength to the serving PoA, the signal strengths to other PoAs, its speed and direction, etc.). When network entities acting as key distribution triggers are not available, HO keys can be distributed periodically or triggered by an event. For instance, after a successful network authentication, the authentication server could distribute the HO keys for this particular MN and session to all potential target PoAs. Such proactive key distributions can only be carried out by the serving authentication server, because it is the only entity having a network path to all PoAs in the network and all other authentication servers with which the network has roaming agreements with. This ensures that HO keys are already available at the target PoA by the time a MN roams there.

4.2.2 Key Distributor

The entity acting as key distributor must be a key holder in the current network connection (i.e be on the path from the serving PoA to the serving AS) and must have access to a key holder on the network path from the target AS to the target PoA. The key distribution path from key distributor to the target PoA might be across one or more other key holders. We distinguish three types of key distributors:

1. The authentication server of the serving network;
2. the lowest common key holder in the serving and target network;

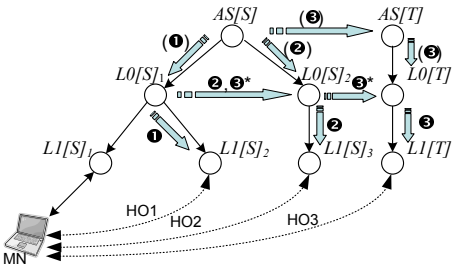

Figure 4: Key Distribution Scenarios

3. the lowest key holder in the serving network with access to the target network via a short cut.

If the serving PoA and target PoA share the same authentication server (see Figure 1), the authentication server may act as the key distributor. If serving PoA and target PoA reside in different networks with roaming agreements (see Figure 2), then the authentication server in the serving network (AS1 in the figure) may act as key distributor.

If both serving and target network paths have one or more common key holders, then the lowest common key holder may act as key distributor. This is illustrated as *Lc*-key holder in Figure 1. Note that the lowest common key holder might be the authentication server.

Sometimes, serving and target network paths are connected via one or more shortcuts (e.g. Lc_i-key holder to Lc_j-key holder in Figure 2). Then the lowest key holder in the serving network with access to such a short cut may act as key distributor distributing keys to the target branch via the short cut.

Some HO key distribution scenarios with different key distributors are illustrated in Figure 4. Note that an authentication server can always serve as HO key distributor, whereas the other two cases depend on the network architecture and interconnection of serving and target networks.

4.2.3 Key Distribution Protocols

There are two general methods for a target PoA to obtain HO keys from the key distributor:

1. requesting the keys using a *pull protocol*;
2. automatically receiving the keys prior to the HO as a result of a *push protocol*.

Pull protocols are on-demand key distribution protocols that are triggered by a MN either (a) through the current link or (b) through the target PoA. In variant (a), the serving PoA forwards the MN's request to the key distributor for the indicated target network. Note that this requires knowledge of which entity serves as key distributor in a target network. This is easy when the authentication server acts as key distributor but more difficult for the two other cases of key distributors mentioned in Section 4.2.2. Upon receiving the request, the key distributor sends HO keys to the

Digital Object Identifier: 10.4108/ICST.WICON2008.4790
http://dx.doi.org/10.4108/ICST.WICON2008.4790

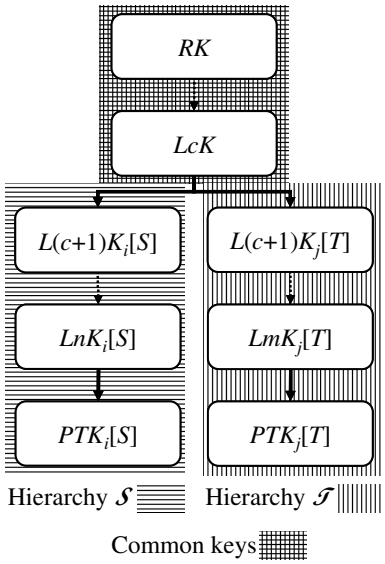

Figure 5: Merged Hierarchies $\mathbb{S}$ and $\mathbb{T}$ with LcK as Lowest Common Key.

target PoA. In variant (b), the MN sends a key distribution request to the target PoA, which forwards the request to the key distributor. The key distributor then returns HO keying material to the target PoA.

On the other hand, *push protocols* are pro-active, i.e. the HO key distributor of a network distributes HO keying material to all key holders. As observed in Section 4.2.1, only the authentication server can take over this role.

We can observe that variant (b) of the pull protocol requires a dual link (i.e. simultaneous connections between the MN and the serving PoA as well as between MN and the target PoA); variant (a) a single link between the MN and the serving PoA; and push protocols are completely independent of the MN's network connections.

5. SECURE RE-USE OF KEYING MATERIAL

The goal for deriving a HO key hierarchy $\mathbb{T}$ is to utilize keys from an existing key hierarchy $\mathbb{S}$ without the need of a full network authentication with the target network. To derive one or more keys of hierarchy $\mathbb{T}$, one or more keys of hierarchy $\mathbb{S}$ are used, and if necessary, all remaining keys of hierarchy $\mathbb{T}$ are freshly derived as specified for technology j. There are two general approaches, as illustrated in Figures 5 and 6, respectively. In the first approach, all keys from hierarchy $\mathbb{S}$ that are held by common key holders are directly re-used in key hierarchy $\mathbb{T}$, here $RK = RK_i[S] = RK_j[T]$ to $LcK = LcK_i[S] = LcK_j[T]$. The remaining keys of $\mathbb{T}$ are derived according to the full authentication in technology j. We refer to this approach as *merged key hierarchies*. In the latter approach, $LxK_i[S]$ is used to derive $LyK_j[T]$. Again, lower level keys can be derived according to technology j. We refer to this approach as *mapped key hierarchies*. We refer to both procedure as *re-using* keying material in the remainder of this paper. Re-using keying material enables re-authentication to expedite the network access authentication of roaming MNs to a target network.

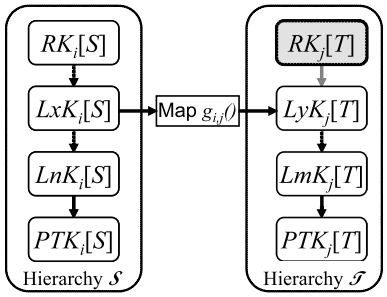

Figure 6: Mapped Key Hierarchies $\mathbb{S}$ and $\mathbb{T}$, Where $LxK_i[S]$ is mapped to $LyK_j[T]$ Using Mapping Function $g_{i,j}()$.

5.1 Challenges

In cellular networks, the same key that is shared between the MN and its serving base station is transferred to the target base station. This does not require any interaction with the MN, and the MN might not even be aware of the handover. However, the same approach is not applicable to non-cellular wireless networks for several reasons: 1) key distribution may have to be triggered by the MN; 2) the networks consist of more than one level of key holders; 3) PoAs and other key holders cannot be trusted to the same extent as cellular base stations.

For these reasons, HO keying material should be of the form of a key hierarchy $\mathbb{T}$ as shown in Figure 3. The main challenges for deriving such a HO key hierarchy $\mathbb{T}$ are as follows:

1. Which existing keying material can be re-used?

Basically, keys held by key holders which have the same role in both the serving and the target key holder paths can be re-used. However, in HOs between two domains with roaming agreements (see Figure 2), the set of key holders is completely disjoint. Furthermore, in inter-technology HOs the number of key holder levels in the target network may be different i.e. $n \neq m$ in Figures 1 and 2. Or the keying hierarchy of technology i might have completely different properties than the key hierarchy of technology j of the target network, such as key entropy, length, lifetime and derivations.

2. How are these keys re-used to derive the HO keys?

For the HO to be considered secure, the derived HO key hierarchy needs to provide at least the same level of security as a key hierarchy that is established in a full network authentication with the target PoA. All keys derived from existing keying material require the inclusion and verification of time variant information (such as nonces or sequence numbers) to prevent replay attacks. At the same time, the key derivations should be efficient, and the MN needs to be capable of deriving all HO keys.

5.2 Approaches

Digital Object Identifier: 10.4108/ICST.WICON2008.4790
http://dx.doi.org/10.4108/ICST.WICON2008.4790

5.2.1 Intra-Technology HO Key Hierarchies

In intra-technology HOs, the serving and target PoA both support the same key hierarchy, i.e. $i = j$ (we drop this index in the remainder of this section). In addition, existing wireless technology standards specify a fixed number of key holder levels (e.g. IEEE 802.11 and IEEE 802.16), so that $n = m$ in intra-technology handovers (we drop this index in the remainder of this section).

In the case that the serving and target PoA are both connected to the same authentication server (see Figure 1), hierarchy $\mathbb{T}$ could be derived by merging with hierarchy $\mathbb{S}$. In that case, keys of hierarchy $\mathbb{S}$ are directly re-used in HO key hierarchy $\mathbb{T}$ for all common key holders. In other words, $RK[S] = RK[T] = RK$, and $LxK[S] = LxK[T]$ for all $x \in \{0, \ldots, c\}$. Remaining keys are derived as specified for technology i. A HO hierarchy $\mathbb{T}$ derived using this approach is illustrated in Figure 5. Please note that this type of re-using keying material cannot prevent replay attacks and some other security features because the re-used keys do not contain any time-variant information.

To prevent replay attacks and provide other security features, the key mapping approach can be applied (see Figure 6). Here, the existing keys of all common key holders are combined with time-variant information $info_Z$ to derive HO keys. For example, a one-way function $h()$ can be used to derive $RK[T] = h(RK[S], info_T)$ and so forth. Function $h()$ must be one-way and publicly known, while $info_T$ is only known to the key distributor and the MN. To enable MN to derive all keys in $\mathbb{T}$, $info_Z$ must be pre-known (e.g. timestamps) or exchanged over a protected channel (e.g. nonces). In some wireless technologies time-variant information is included in the key derivations, e.g. nonces exchanged as part of the authentication process. In that case the same format and exchange method as described for the full authentication should be applied for the HO key derivations as well.

In the case that the serving and target PoAs are connected to different authentication servers AS1 and AS2 (see Figure 2), keys cannot be directly re-used (merging approach) for two reasons: 1) serving and target networks do not have any common key holders; and 2) despite roaming agreements, the keys from the serving network should never be directly passed to the target network. Instead, the mapping approach, as described in the previous paragraph, needs to be used. To enable the derivation of HO keying material, either AS1 passes the derived HO root key (e.g. $RK[T] = h(RK[S], info_Z)$) to AS2 via a backend connection or, if applicable, the $Lx[S]$-key holder passes derived keys (e.g. $LyK[T] = h(LxK[S], info_Z)$) to the $Ly[T]$-key holder via a shortcut. In the latter case, keys should only be passed from a higher level key holder to a key holder of the same or lower level, i.e. $y \geq x$, to maintain the trust level. Please refer to Section 7 for a discussion on trust levels.

5.2.2 Inter-Technology HO Key Hierarchies

If two different wireless access technologies i and j have a common key in their hierarchies, the same merging method as described in the previous section can be used. For example, IEEE 802.11i and IEEE 802.16e both utilize EAP to derive keying material and, thus, both establish root keys of the same format (namely the master session key MSK and the extended master session key $EMSK$).

The key hierarchy mapping approach must be used whenever both technologies do not share any common keys. Such an inter-technology key re-use requires a secure mapping $g_{i,j}()$ from a key in hierarchy $\mathbb{S}$ to a key in hierarchy $\mathbb{T}$ as illustrated in Figure 6, where an existing key $LxK_i[S]$ is mapped to $LyK_j[T]$ with $LyK_j[T] = g_{i,j}(LxK_i[S], info_T)$. This is very different from the mapping approach for intra-technology HOs, because instead of simply including time-variant information, the mapping function $g_{i,j}()$ and its input format must be designed such that the derived keys meet all security requirements of the target access technology. To achieve this the following requirements need to be met:

1. Input key $LxK_i[S]$ has at least the security strength as required for $LyK_j[T]$.
2. $info_T$ contains the same information as defined for the respective key in access technology j.
3. Mapping function $g_{i,j}()$ is one-way.

If any of the above conditions is not met, then the security of the HO key hierarchy cannot be ensured. Note that suitable mapping functions can be applied at any level and may map a key from hierarchy $\mathbb{S}$ to a higher or lower level key in hierarchy $\mathbb{T}$. A different mapping $g_{i,j}()$ must be carefully designed for each pair of technologies (i, j) and each HO direction. Hence, the solution is not very scalable.

6. KEY UPDATE AND SYNCHRONIZATION

With HO keys that have been derived from existing keying material and been distributed to different branches in the network (see Figure 1) or even to other networks with roaming agreements (see Figure 2), problems with key synchronization arise.

6.1 Challenges

If a key needs to be updated—because the key or its key holder has been compromised—all keys that have been derived from this key need to be updated as well. In addition, no further keys should be derived from outdated or compromised keys. Consequently, all affected network entities, as well as the MN, must be aware of key updates. Keys can be updated in two different ways, namely by:

1. A full authentication to update the root key and, thus, the entire key hierarchy;
2. A protocol to update a certain key between a key holder and the MN.

After identifying who can initiate key updates and which keys need to be updated, the following main challenges for key updates and synchronization need to be addressed:

1. Who can execute key updates?

Digital Object Identifier: 10.4108/ICST.WICON2008.4790
http://dx.doi.org/10.4108/ICST.WICON2008.4790

In order to derive fresh updated keys, the key update executor needs to communicate with the MN. However, not all key holders are capable of that. That is, for a certain key holder, it may not have a protocol with the MN to update the key.

2. How can key updates be synchronized with all key holders across networks and the MN?

The entity executing the key update needs to distribute keys to all affected key holders. Again, not all network entities are capable of that. The new updated portion of the key hierarchy should be as secure as the previous replaced keys in the key hierarchy.

6.2 Approaches

Key updates are either (1) executed periodically or (2) triggered by an event. The first method can only be executed by the authentication server to ensure the network-wide distribution of updated keys. This corresponds to the push protocol for HO key distribution as described in Section 4.2.3. In the latter case, several trigger events can be envisioned. For example, a successfully completed full authentication can serve as such a trigger, in which case the authentication server serves as key distributor in a push protocol. Alternatively, a key update could be initiated by a compromised key holder itself, which corresponds to the pull protocol in Section 4.2.3. Note that a MN can act as such an initiator to execute a key update with the key holder of the compromised key.

All descendant keys that are derived from a source key are typically referred to as *child keys*. For instance in Figure 5, $L(c+1)K_i[S]$ to $LnK_i[S]$, $PTK_i[S]$, $L(c+1)K_j[T]$ to $L(c+1)K_m[T]$ and $PTK_j[T]$ are all child keys of LcK. In Figure 6, all keys in hierarchy $\mathbb{S}$ excluding $RK_i[S]$ itself are child keys of $RK_i[S]$ as well as $LyK_j[T]$ and all lower level keys in hierarchy T. Consequently, if a key $LxK_i[S]$ is updated, all its child keys in key hierarchy $\mathbb{S}$ as well as all its child keys in HO key hierarchies $\mathbb{T}$ need to be updated.

6.2.1 Key Update Executors

The key holder initiating the key update might not be able to execute it. For instance, key updates require the communication with all affected key holders including the MN. Typically, only the authentication server and the PoAs support protocols that enables them to communicate with the MN. Key holders who do not have these capabilities cannot serve as key update executors. While the authentication server can act as update executor as long as the root key has not been compromised, PoAs can only serve as key update executor if: (1) only the PTKs are compromised; (2) the LnKs are still valid; and (3) the update was initiated by the MN or the PoA itself. For synchronization purposes, pull and push protocols should only be supported simultaneously if the authentication server acts as the key distributor in both protocols, i.e. if updates by PoAs are disabled.

6.2.2 Update Mechanisms

In case root key RK is compromised, the authentication server and the MN need to execute a full network authentication to derive a fresh key hierarchy. This is straight forward and will not be discussed any further. In all other cases, it is desirable to employ a more efficient key update mechanism. Therefore—after a key update has been triggered—the key update executor derives the updated keys. If the PoA acts as the executor, keys do not need to be distributed. On the other hand, if another key holder acts as the update executor, the updated keys need to be distributed to all lower key holders in the network. In any case, the MN needs to derive all updated keys and must be informed by the key update executor about the key updates.

To derive secure, cryptographic independent updated keys, an uncompromised key from the current session must be used as an input. For example, the authentication server acting as the update executor uses the root key $RK_i[S]$ as input to derive $L0K'_i[S]$, whereas the PoA uses $LnK_i[S]$ to derive $PTK'_i[S]$. To prevent replay attacks, time-variant information $info_T$ should be used as additional input in key derivations. To be able to derive all updated keys, the MN must know $info_T$, which again may require the interaction with the key update executor. For example, if timestamps are used as predictable non-repeating data, the clock between the MN and the update executor must be synchronized. However, such synchronization is not easy to realize in mobile wireless networks. Also, sequence numbers require synchronization, and all MNs, PoAs and the authentication server need to keep track of the sequence numbers for each session. This seems especially challenging in networks that support more than one key update executor. For instance, problems may occur if a key was previously updated by a PoA and the MN then switches to another PoA that is not aware of those previous updates. In that case, target PoA and MN may derive different keys and the re-authentication will fail resulting into service disruption. This includes cases where target authentication servers are not aware of previous updates by PoAs. To resolve this potential issue, some coordination (and thus communication) among PoAs and the authentication server is needed. As another time-variant information, nonces can be used. The use of nonces requires interactive communications between the update executor and the MN to exchange this information, for example, using a 4-way handshake.

7. TRUST MODELS AND SERVER-CENTRIC TRUST

Unlike in cellular networks, we cannot assume the same trust model for the network entities across all non-cellular wireless access technologies and domains. While in cellular networks the same keys are shared among base stations, in the networks we consider, different entities operating at the same key holder level may not have the same physical protection. For example, an IEEE 802.11 access point and an IEEE 802.16 base station may be both LnK-key holders, but while IEEE 802.11 access points are low cost and numerous, IEEE 802.16 base stations are rather expensive and harder to access. This suggests that base stations can be protected more efficiently. In addition, if the serving and target network have a different number of key holder levels, it might be difficult to match the associated trust level of key holders in different networks.

Only in a single domain with one authentication server in which each PoA supports the same wireless access technol-

Digital Object Identifier: 10.4108/ICST.WICON2008.4790
http://dx.doi.org/10.4108/ICST.WICON2008.4790

ogy does a trust comparison of network entities seem feasible. However, the actual level of trust associated with each key holder level may still be hard to determine. In all other networks, such a comparison is very difficult. To avoid security problems, but still allow handovers between networks with differing trust models, the key lifetime of HO keys should be rather limited. This forces the MN to execute a full network authentication shortly after entering the target network without disrupting the MN's connectivity. This approach helps to release the liability of the serving network shortly after the HO and—at the same time—assures the target network that its new connection to the MN is secure.

It can be observed that the authentication server is often the most trusted key holder in the network and can be considered as a center of trust. Some security properties of re-authentication protocols depend on the involvement of such a center of trust. For example, some re-authentication protocols utilize sequence numbers to prevent replay attacks (such as [11]), which requires a trusted entity to verify the sequence numbers provided by an MN. The trusted entity must be capable of keeping track of sequence numbers of all MNs that previously accessed the network and update them accordingly with each re-authentication. Furthermore, providing channel binding (as defined in [3]) requires the target authentication server to verify the information advertised by the target PoA. Finally, as discussed in Section 5.2.1, the key holder deriving HO keys must be trusted to only include valid HO-specific information to derive HO keys. Otherwise, this key holder could include false key holder identifiers or tamper with other security-sensitive HO-specific information. We can observe that the authentication server is likely to be the only network entity capable and sufficiently trusted to execute these tasks.

8. PERFORMANCE AND SECURITY DISCUSSIONS

For our discussions, we consider several combinations of the presented distribution methods with different key distributors, as illustrated in Figure 4. In handover scenario 1 (denoted as HO1 in the figure), the MN switches from its serving PoA, acting as $L1[S]_1$-key holder, to a target PoA, acting as $L1[S]_2$. Here, the $L0[S]_1$-key holder is the lowest common key holder and may act as the key distributor. In another scenario, the authentication server $AS[S]$ can serve as the key distributor (indicated with brackets in the figure). In handover scenario 2 (HO2), the MN switches from the $L1[S]_1$-key holder to $L1[S]_2$. Here, the lowest common key holder is $AS[S]$, which can distribute keys via $L0[S]_2$ to the target PoA. An alternative path for key distribution can use a short cut between $L0[S]_1$ and $L0[S]_2$ (denoted with *). In the third handover scenario (HO3), no common key holder is present, since the target PoA is located in another network with the authentication server $AS[T]$. Keys can be either distributed by $AS[S]$ via backbone connections to $AS[T]$ or, if applicable, over a short cut (e.g. from $L0[S]_2$ to $L0[T]$).

We can observe that for optimizing the efficiency of HO key distribution, the lowest common key holder should be used whenever applicable, and short cuts are preferable over key distribution through the authentication server(s) in the backbone. However, from a security perspective, only the authentication server acting as key distributor can ensure the security properties discussed in Sections 6 and 7.

In addition to the location of the key distributor, the method of key distribution heavily affects the overall performance of the key distribution scheme. The push protocol introduces by far the most network traffic and likely leads to some unnecessarily distributed keys that will never be used. Key distribution can only be optimized with the aid of additional information, e.g. the location information of the MN and the PoAs in the network. It is important to note that the push protocol only affects the HO preparation time and network traffic of the first HO after a full network authentication. All subsequent HOs require neither additional key derivations nor key distributions. This is beneficial if an MN frequently performs intra-domain HOs in a short period time. In both variants of the pull protocol described in Section 4.2.3, only keying material for a specific target PoA is distributed, where network delay and traffic depend on the location of the key distributor. Variant (b) introduces the least network traffic, but is the only method that requires a dual link between the MN and the serving PoA, as well as between the MN and the target PoA.

Even though the push method is more efficient for frequent HOs and has some advantages with key synchronization, only the pull approach allows the inclusion of fresh information $info_Z$ in HO keys to provide some security features. For example, such information is necessary for replay prevention and, in case the authentication server acts as a key distributor, to provide channel binding.

9. CONCLUSIONS

In this paper, we are the first to explore the various security aspects of providing key management to enable seamless mobility in heterogenous networks. Current standards only cover parts of the problem, e.g. media-dependent solutions (IEEE 802.11i and IEEE 802.16e) or EAP-based HO key hierarchies (IETF HOKEY WG). Industry is still exploring possible ways to handle keys for mobility, and this paper can serve as an implementation guideline to identify security challenges and choose a suitable solution strategy, based on our security and performance trade-off analysis.

We identified the derivation of a secure HO key hierarchy and the timely distribution of these keys as two crucial components of secure seamless HOs, and showed that HO security and performance depend on the method used to derive the HO key hierarchy, the network position of the entity acting as the key distributor and the protocol used to distribute HO keys. As part of the presented solutions, we introduced generic HO key hierarchies for network technologies that share a common key, and summarized requirements for key mapping functions for technologies that do not share such common keys. Furthermore, we showed that three special network entities are suitable to derive and distribute keys, namely the serving authentication server, a common key holder, and an entity with short-cut access to the target network. Finally, we presented three HO key distribution protocols: a push protocol and two variants of pull protocols.

All presented solutions and their variants constitute a trade-

Digital Object Identifier: 10.4108/ICST.WICON2008.4790
http://dx.doi.org/10.4108/ICST.WICON2008.4790

off between security and performance, which we explored in our analysis. Our results show that some security features can only be provided with the involvement of an authentication server or another central trusted network entity. These features include network-wide key synchronization, homogenous trust modeling, channel binding, and replay prevention using sequence numbers or timestamps. However, the required on-line interaction with the server during the HO prevents the use of proactive push protocols for key distribution. Hence, if such security properties are required, only pull protocols can be used.

In conclusion, we observe that many security features require server access during the HO. This introduces communication delays and network traffic that can significantly slow down the HO. A risk assessment analysis of the system is necessary to evaluate whether some of these security features can be suspended for a limited period of time. To re-establish a link with all desirable security properties, a policy could enforce a full network authentication a short time after a successful seamless HO. In that case, the connection and service continuity can be maintained at all times. On the other hand, if none of these security features can be suspended at any time, accessing the target server during the HO is unavoidable. Under these circumstances, a solution enabling a timely initiation of the re-authentication protocol is crucial.

10. REFERENCES

[1] Handover keying working group (hokey wg). Internet Engineering Task Force. http://www.ietf.org/html.charters/hokey-charter.html.

[2] B. Aboba, L. Blunk, J. Vollbrecht, and J. Carlson. *RFC 3748, Extensible Authentication Protocol (EAP).* Internet Engineering Task Force, June 2004.

[3] T. Clancy and K. Hoeper. *Channel Binding Support for EAP Methods.* Internet Engineering Task Force, <draft-clancy-emu-chbind-01>, June 2008. Work in progress.

[4] T. Clancy, M. Nakhjiri, V. Narayanan, and L. Dondeti. *RFC 5169, Handover Key Management and Re-Authentication Problem Statement.* Internet Engineering Task Force, March 2008.

[5] Institute of Electrical and Electronics Engineers. *IEEE 802.11-1999, Standard for Local and metropolitan area networks -specific requirements - part 11: Wireless LAN Medium Access Control and Physical Layer specifications*, 1999.

[6] Institute of Electrical and Electronics Engineers. *IEEE 802.11i, Supplement to Standard for Telecommunications and Information Exchange Between Systems - LAN/MAN Specific Requirements - part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: Specification for Enhanced Security*, July 2004.

[7] Institute of Electrical and Electronics Engineers. *IEEE 802.16-2004, Standard for Local and metropolitan area networks -specific requirements - part 16: Air Interface for Fixed Broadband Wireless Access Systems*, October 2004.

[8] Institute of Electrical and Electronics Engineers. *IEEE 802.16e, Standard for Local and metropolitan area networks -Air Interface for Fixed and Mobile Broadband Wireless Access Systems*, February 2006. Amendment 2: Physical and Medium Access Control Layers for Combined Fixed and Mobile Operation in Licensed Bands, Corrigendum 1.

[9] Institute of Electrical and Electronics Engineers. *IEEE 802.11r/D6.0, Draft Standard for Local and metropolitan area networks - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications*, May 2007. Amendment 2: Fast BSS Transition.

[10] Institute of Electrical and Electronics Engineers. *IEEE 802.21/D8.0, Draft Standard for Local and Metropolitan Area Networks: Media Independent Handover Services*, December 2007.

[11] V. Narayanan and L. Dondeti. *EAP Extensions for EAP Re-authentication Protocol (ERP).* Internet Engineering Task Force, <draft-ietf-hokey-erx-14>, March 2008. Work in progress.

[12] Y. Ohba. *EAP Pre-authentication Problem Statement.* Internet Engineering Task Force, <draft-ietf-hokey-preauth-ps-03>, June 2008. Work in progress.

Digital Object Identifier: 10.4108/ICST.WICON2008.4790
http://dx.doi.org/10.4108/ICST.WICON2008.4790