Universidade de Aveiro Electrónica, Telecomunicações e Informática
2008

Rodolphe António de Araújo Marques

Security and Mobility in 802.11 Structured Networks
Segurança e Mobilidade em Redes Estruturadas 802.11

Universidade de Aveiro Departamento de Electrónica, Telecomunicações e Informática
2008

Rodolphe António de Araújo Marques

Security and Mobility in 802.11 Structured Networks
Segurança e Mobilidade em Redes Estruturadas 802.11

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Engenharia Electrónica e Telecomunicações, realizado sob a orientação científica do Dr. André Zúquete, Professor Auxiliar do Departamento de Electrónica, Telecomunicações e Informática da Universidade de Aveiro.

o júri / the jury

presidente / president	**Doutor Atílio Manuel da Silva Gameiro** Professor Associado da Universidade de Aveiro
vogais / examiners committee	**Doutor Edmundo Heitor Silva Monteiro** Professor Associado com Agregação do Departamento de Engenharia Informática da Faculdade de Ciências e Tecnologia da Universidade de Coimbra
	Doutor André Ventura da Cruz Marnoto Zúquete (Orientador) Professor Auxiliar da Universidade de Aveiro
	Doutor Paulo Jorge Salvador Serra Ferreira (Co-Orientador) Professor Auxiliar Convidado da Universidade de Aveiro

agradecimentos / acknowledgements

Agradeço à colaboradora do projecto, Professora Susana Sargento e ao meu co-orientador, Professor Paulo Salvador por toda a disponibilidade demonstrada.

Agradeço especialmente ao meu orientador, Professor André Zúquete por todo o conhecimento transmitido, apoio e dedicação ao longo do desenvolvimento do projecto, e sem o qual a sua concretização não teria sido possível.

Deixo também um agradecimento a todas as pessoas que me são próximas e que, de alguma forma, contribuiram para a concretização deste projecto.

Resumo

Nesta tese é apresentado um protocolo que permite handovers rápidos e seguros em redes estruturadas 802.11. Este protocolo recupera o paradigma original do 802.11: autenticar primeiro, reassociar depois. Partindo deste paradigma, apresentamos duas novas operações 802.11 de autenticação e (re)associação, que permitem que uma estação móvel realize reautenticações e reassociações com as mesmas funcionalidades do 802.1X. Esta nova aproxiamação requer pouca mudança na arquitectura da rede, nomeadamente só necessita de um novo Servidor de Reautenticação, para armazenar os dados usados pelas estações móveis durante as reautenticações. Nesta tese é também apresentada uma extensão do nosso protocolo, de maneira a permitir uma migração rápida e segura entre ESS usando Mobile IP.

Abstract

This thesis presents a fast, secure handover protocol that recovers the original 802.11 paradigm: authenticate first, reassociate next. Following this paradigm, we present two new 802.11 authentication and (re)association operations which allow a mobile station to perform network reauthentications and reassociations with the same functionality of a complete 802.1X authentication. This new approach requires very little from the environment, namely it only requires a new, central network Reauthentication Service, for storing data used in the reauthentication of stations. This thesis also presents a layer 3 extension of our protocol, to support fast, secure transitions between ESS using Mobile IP.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The widespread deployment of 802.11 APs (Access Points) is a strong incentive to the mobility of people while keeping their network connections. However, while mobility is possible, the support for fast handovers of mobile stations (STAs) is still incipient. Namely, handovers in wireless networks requiring 801.1X authentications take a non-negligible time, which create noticeable outage periods.

Currently, 802.1X is the accepted standard for implementing strong authentication and access control in large wireless networks. Furthermore, this standard enables the mutual authentication of an STA and an access network provided by one organization, even when the user managing the STA is known and authenticated by another organization. Unfortunately, this standard, by design, is not suitable for implementing fast handovers of mobile STAs. This happens because, with 802.1X, the reassociation of an STA to a different AP takes place during the mutual (re)authentication between the STA and the AP (or network) where he got reassociated. This means that the time expended in the mutual authentication of STAs and APs will always contribute to the outage time during a handover (see Figure 1.1).

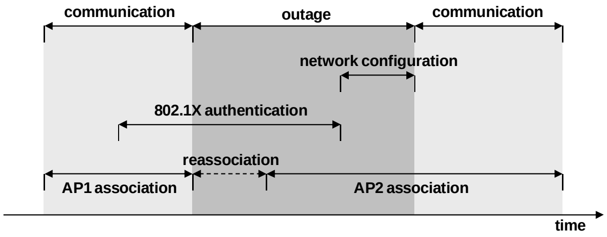


Figure 1.1: Outage periods during reassociations with 802.1X (re)authentications.

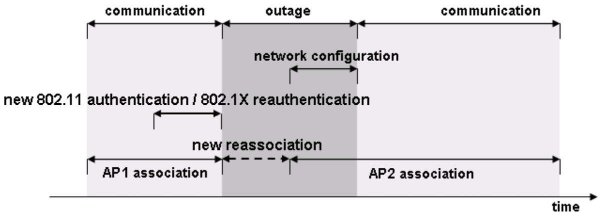


Figure 1.2: Outage periods during reassociations with our new 802.11 authentication and reassociation protocols.

In this document, we provide a new approach for dealing with intra-domain, mutual reauthentications of mobile STAs and 802.11 wireless networks. Our approach recovers the 802.11 principles for reassociation of mobile STAs, which are the following: first, the STA is authenticated by a candidate serving AP, while associated with a currently serving AP; then it gets reassociated to the new serving AP. With this model, outage periods during handovers are reduced to the time to perform reassociations and network configurations, but not (re)authentications (see Figure 1.2).

To design the new approach we isolated the functionalities provided by 802.1X and we implemented them in two new 802.11 authentication and reassociation operations. This way, we are able to perform a normal 802.11 reassociation, preceded by an 802.11 authentication, while keeping the security features of 802.1X. We also adopted the general reauthentication architecture and key hierarchy proposed by the HOKEY (HandOver KEYing) IETF Working Group [1, 2], which uses a local HOKEY service and key hierarchies starting in the 802.1X EMSK (Extended Master Session Key). Our proposal is also inline with a TGi (IEEE Task Group i) goal: handover to a new serving AP should not require a complete 802.1X reauthentication.

Another problem occurs when the STA roams to a different ESS. In this case the STA as to acquire a new IP usually, through DHCP. The process of obtaining a new IP address not only adds a significant latency to the already slow 802.1X authentication, but it also presents serious problems to the transport and application layer protocols that use the IP address as end point identifier. This means that all ongoing communications, services which depend on IP addresses, as opposed to hostnames; and some security mechanisms that provide access-privileges to node based upon their IP address, present in the STA would have to be terminated, with new connections being initiated by the STA at its new address. Thus, by definition, changing a mobile node's address as it moves does not solve the problem of node mobility, being *mobility* the ability of a node to change its point-of-attachment from one link to another while maintaining all existing communications and using the same IP address at its new link [3].

To deal with this problem we used Mobile IP [4]. Our approach consists in taking the following options: (i) advertising in advance the information needed by the STA to determine to which subnet the target AP belongs to, and the IP address of the MIP Foreign Agent, if existing in the served network, (ii) acquire and register if possible before the handover occurs the care-of address, reusing key material of our key hierarchy to mutually authenticate the registration messages.

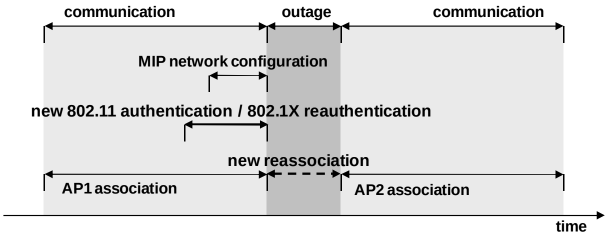


Figure 1.3: Outage periods during a layer 3 handover using Mobile IP.

In short, we present a solution using MIP were the STA registers its care-of address prior to its reassociation, this way eliminating the latency introduced by network configuration. We also defined new keys and security associations needed to secure the registration process, thus preventing denial-of-service attacks.

During our work an article called *Fast, Secure Handovers in 802.11: Back to the Basis* as been submitted presenting our layer 2 contribution to *The 4th ACM International Workshop on QoS and Security for Wireless and Mobile Networks (Q2SWinet 2008)*. By the time of this writing we were still waiting for the notification of acceptance.

This document is organized as follows. Chapter 2 overviews roaming in 802.11 networks; the

802.11 standard and its protocols, namely 802.11i; various other protocols and methodologies to achieve fast handovers; and the Mobile IP. Chapter 3 presents the related work. Chapter 4 and Chapter 5 describe our layer 2 and layer 3 fast, secure handover protocol. Chapter 6 and Chapter 7 evaluates the security of our protocol, goals achieved and some practical considerations about the new reauthentication and reassociation protocols. Chapter 8 describes a prototype implementation of our layer 2 protocol and Chapter 9 evaluates its performance, namely the reduction of outage delays. Finally, chapter 10 presents the conclusions.

Wrapping up, our contributions are the following:

- A fast, 802.11 security protocol compatible with the 802.1X paradigms.
- An authenticated 802.11 reauthentication protocol.
- A new server for managing both layer 2 and layer 3 mobility.
- A usage case of MIP, implementing a way of adding security attributes to messages
- A very fast, secure handoff protocol.

Chapter 2

Context

2.1 802.11 Standard

2.1.1 Introduction

Nowadays the IEEE 802.11 WLAN standard is by far the most widely deployed wireless local area network technology in the world. Work on this standard began in the early 1990s. At that time, the 802.11 standard was seen largely as an untethered alternative to the 802.3 Ethernet cable that connected most of the world's enterprise users to the Internet. The concept of supporting QoS features such as voice was not one of the priorities of the design. But since the year 2000 it has become evident that 802.11 WLAN will be used for applications demanding QoS. Similary, in the initial design, relatively little attention was paid to the security related areas of authentication and encryption. It is now taken as given that securing the first hop of 802.11 connections is indeed a prerequisite for widespread enterprise deployment of the technology. In parallel with these changes, the physical layer of 802.11 has undergone a number of major updates, and in the process has greatly increased its speed range of technology. These concurrent factors of increased speed, greater range, QoS requirements, and security have combined to create a plethora of challenges for 802.11 implementations, and indeed for the IEEE 802.11 committee itself.

2.1.2 802.11 Network Architecture

The IEEE 802.11 standard allows two types of alternative network architectures which are illustrated in figure 2.1:

- ***Ad-hoc***. In this architecture a group of STAs are able to communicate with each other based on a P2P model. The group of STAs of an *ad-hoc* network form an IBSS (*Independent Basic Service Set*). In our work we wont be focusing this type of networks.

- **Structured**. In this architecture STAs communicate with APs. APs are bridges between the wireless world and the wired world. As bridges, then all APs have features that one would expect to see on a network bridge. They have at least two network interfaces: a wireless interface that understands the details of 802.11 and a second interface to connect to wired networks. A set of STAs that associate to a specific AP form a BSS (*Basic Service Set*). This set can be spread so it can contain other APs to form a ESS (*Extended Service Set*). The wired network that connects the various AP is called a DS (*Distribution System*). Our work will be focused in this type of network architecture.

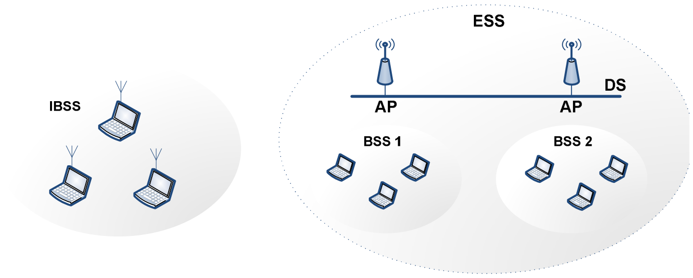


Figure 2.1: 802.11 network architecture: ad hoc (IBSS) and structured (BSS and ESS).

2.1.3 Communication in Structured Networks

In a 802.11 structured network, the communication between an STA and an AP is made using three basic types of frames, each one of them with several subtypes. The three basic types of frames are:

- **Data frames**. Data frames carry higher-level protocol data in the frame body.
- **Management frames**. Management frames perform supervisory functions; they are used for beaconing, probing, to join and leave wireless networks and move associations from AP to AP.
- **Control frames**. Control frames assist in the delivery of data frames. They administer access to the wireless medium (but not the medium itself) and provide MAC-layer reliability functions.

2.1.4 MAC Layer Services

Devices using the IEEE 802.11 PHY and MAC as part of a WLAN are called stations. Stations can be endpoints or APs. APs are stations that act as part of the DS and facilitate the distribution of data between endpoints. The MAC provides nine logical services: authentication, deauthentication, association, disassociation, reassociation, distribution, integration, privacy, and data delivery. An AP uses all nine services. An endpoint uses authentication, deauthentication, privacy, and data delivery. Each service utilizes a set of messages with information elements that are pertinent to the services.

- **Authentication**

  Because WLANs have limited physical security to prevent unauthorized access, 802.11 defines authentication services for controling the access to a WLAN. The goal of the authentication service is to provide access control equal to a wired LAN. The authentication service provides a mechanism for one station to identify another station. Without this proof of identity, the station is not allowed to use the WLAN for data delivery. All 802.11 stations, whether they are part of an IBSS or extended service set (ESS) network, must use the authentication service prior to communicating with another station.

- **Open System Authentication (OSA)**

  This is the default authentication method, which is a very simple, two-step process. First, the station wanting to authenticate with another station sends an authentication management frame containing the sending station's identity. The receiving station then sends back a frame alerting whether it recognizes the identity of the authenticating station.

- **Shared Key Authentication (SKA)**

  This type of authentication assumes that each station has received a secret shared key through a secure channel independent of the 802.11 network. Stations authenticate through shared knowledge of the secret key. Use of shared key authentication requires the implementation of encryption via the Wired Equivalent Privacy (WEP) algorithm.

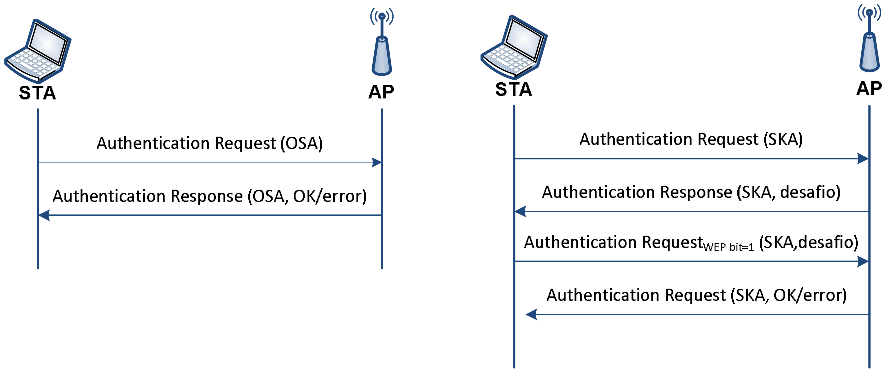

STA
AP
Authentication Request (OSA)
Authentication Response (OSA, OK/error)
STA
AP
Authentication Request (SKA)
Authentication Response (SKA, desafio)
Authentication Request$_{\text{WEP bit=1}}$ (SKA,desafio)
Authentication Request (SKA, OK/error)

Figure 2.2: WEP authentication models: OSA, in the left, and SKA, in the right.

- **Deauthentication**

  This type removes any state related with a previous authentication. The deauthentication service is used to eliminate a previously authorized user from any further use of the network. Once a station is deauthenticated, that station is no longer able to access the WLAN without performing the authentication function again. Deauthentication is a notification and cannot be refused. For example, when a station wants to be removed from a BSS, it can send a deauthentication management frame to the associated AP to notify the AP of the removal from the network. An AP can also deauthenticate a station by sending a deauthentication frame to the station.

- **Association**

  Association maps a station to an AP and enables the AP to distribute data to and from the station. The association service is used to make a logical connection between a mobile station and an AP. Each station must become associated with an AP before it is allowed to send data through the AP onto the DS. The connection is necessary in order for the DS to know where and how to deliver data to the mobile station.

  The mobile station invokes the association service once and only once, typically when the station enters the BSS. Each station can associate with one AP, although an AP can associate with multiple stations.

- **Disassociation**

  This breaks an existing association relationship. The disassociation service is used either to force a mobile station to eliminate an association with an AP or for a mobile station to inform an AP that it no longer requires the services of the DS. When a station becomes disassociated, it must begin a new association to communicate with an AP again.

  An AP may force a station or stations to disassociate because of resource restraints; the AP is shutting down or being removed from the network for a variety of reasons. When a mobile station is aware that it will no longer require the services of an AP, it may invoke the disassociation service to notify the AP that the logical connection to the services of the AP from this mobile station is no longer required.

Stations should disassociate when they leave a network, although nothing in the architecture ensures this will happen. Disassociation is a notification and can be invoked by either associated party. Neither party can refuse the termination of the association.

- **Reassociation**

  This type transfers an association between APs. Reassociation enables a station to change its current association with an AP. The reassociation service is similar to the association service, with the exception that it includes information about the AP with which a mobile station has been previously associated. A mobile station will use the reassociation service repeatedly as it moves throughout the ESS, loses contact with the AP with which it is associated, and needs to become associated with a new AP.

  By using the reassociation service, a mobile station provides information to the AP to which it will be associated and information pertaining to the AP to which it will be disassociated. This enables the newly associated AP to contact the previously associated AP to obtain frames that may be waiting there for delivery to the mobile station as well as other information that may be relevant to the new association. The mobile station always initiates reassociation.

- **Privacy**

  With a wireless network, all stations and other devices can hear data traffic taking place within range on the network, seriously impacting the security level of a wireless link. IEEE 802.11 counters this problem by offering a privacy service option that raises the security of the 802.11 network to that of a wired network. This type prevents the unauthorized viewing of data through the use of the WEP algorithm. The privacy service of IEEE 802.11 is designed to provide an equivalent level of protection for data on the WLAN as that provided by a wired network with restricted physical access. This service protects that data only as it traverses the wireless medium. It is not designed to provide complete protection of data between applications running over a mixed network. The privacy service, applying to all data frames and some authentication management frames, is an encryption algorithm based on 802.11.

- **Distribution**

  Distribution is the primary service used by an 802.11 station. A station uses the distribution service every time it sends MAC frames across the DS. The distribution service provides the distribution with only enough information to determine the proper destination BSS for the MAC frame.

  The three association services (association, reassociation, and disassociation) provide the necessary information for the distribution service to operate. Distribution within the DS does not necessarily involve any additional features outside of the association services, although a station must be associated with an AP for the distribution service to forward frames properly.

- **Data delivery**

  This provides data transfer between stations.

- **Integration**

This provides data transfer between the DS of an IEEE 802.11 LAN and a non-IEEE 802.11 LAN. The station providing this function is called a portal . The integration service connects the 802.11 WLAN to other LANs, including one or more wired LANs or 802.11 WLANS. A portal performs the integration service. A portal is an abstract architectural concept that typically resides in an AP, although it could be part of a separate network component entirely. The integration service translates 802.11 frames to frames that can traverse other networks.

2.1.5 STA State Machine

When an STA wants to connect to a WLAN, knowing already the SSID, it completes this process in two stages: authentication and association. Figure 2.3 shows the 802.11 state machine. This states are used to determine what type of frames the STA and AP can exchange in each moment of the network association process.

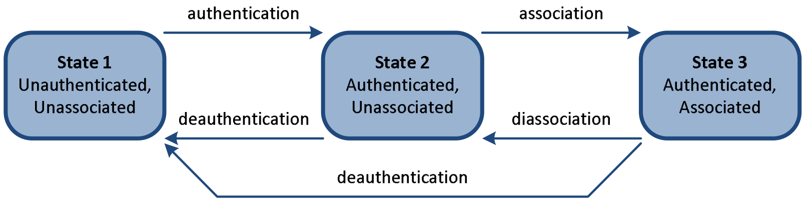


Figure 2.3: 802.11 state machine.

2.2 Roaming, Handover, and Mobility

Mobility and handover; in essence they mean the same thing with a slight difference: Mobility usually is used for wired systems, in particular for Internet Protocol (IP) networks, while handover is used for wireless systems. Handover is sometimes also known as handoff. Achieving inter-domain and intra-domain handover is a challenge for the future growth of WLAN technology. Another term, roaming, also used for handover in WLAN. For the purposes of this work we will consider terms like roaming, handover, handoff or mobility like being the same.

Since the concept of roaming is of great importance to this work here are some different notions of roaming.

Webster's New World Dictionary defines roaming as follows: **Roaming:** *Traveling about, especially in search of adventure: Errant, roving, wandering.*

The Wi-Fi Planet Webopedia defines roaming as: **Roaming:** *The extending of connectivity service of a network into a new region.*

As an example we can think of a person who is working on an 802.11 enabled laptop surfing the Internet, while using a wireless Internet access. Figure 2.4 depicts the situation.

In this example we have a person in motion who is using a laptop to connect to a wireless network. When a wireless connection is in use, there can be a definite motion taking place, which causes the user to switch APs.

Figure 2.4: Roaming between APs.

2.2.1 Mobility Support

Mobility is the major motivation for deploying an 802.11 network. Stations can move while connected to the network and transmit frames while in motion. Mobility can cause one of three types of transition:

- **No transition**

  When stations do not move out of their current AP's service area, no transition is necessary. This state occurs because the station is not moving or it is moving within the basic service area of its current AP. Although this is not actually a transition it is defined in the specification [5].

- **BSS transition**

  Stations continuously monitor the signal strength and quality from all APs administratively assigned to cover an extended service area. Within an extended service area, 802.11 provides MAC layer mobility. Network equipment connected to the distribution system can send out frames addressed to the MAC address of a mobile station and let the APs handle the final hop to the mobile station. Thus, they do not need to be aware of a mobile station's location as long as it is within the same extended service area.

  Figure 2.5 illustrates a BSS transition. The three APs in the picture are all assigned to the same ESS. The laptop with an 802.11 network card is sitting within AP 1 basic service area and is associated with AP 1. When the laptop moves out of AP 1 basic service area and into AP 2, a BSS transition occurs. The mobile station uses the reassociation service to associate with AP 2, which then starts sending frames to the mobile station.

  BSS transitions require the cooperation of APs. In this scenario, AP 2 needs to inform AP 1 that the mobile station is now associated with AP 2. 802.11 does not specify the details of the communications between APs during BSS transitions. 802.11f appeared as a set of recommendations to attempt to provide inter AP communication between multi-vendor systems. The IEEE 802 Executive Committee approved its withdrawal on November 18, 2005.

- **ESS transition**

  An ESS transition refers to the movement from one ESS to a second distinct ESS. 802.11 does not support this type of transition, except to allow the station to associate with an AP in the second ESS once it leaves the first. Higher-layer connections are almost guaranteed to be interrupted. It would be fair to say that 802.11 supports ESS transitions only to the extent that it is relatively easy to attempt associating with an AP in the new extended service area. Maintaining higher-level connections requires support from the protocol suites in question. In the case of TCP/IP, Mobile IP is required to seamlessly support an ESS transition.

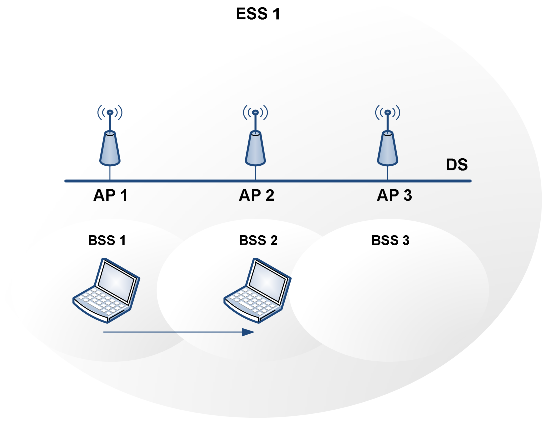


Figure 2.5: BSS Transition.

Figure 2.6 illustrates an ESS transition. Four basic service areas are organized in two extended service areas. Seamless transitions from the lefthand ESS to the righthand ESS are not supported. ESS transitions are supported only because the mobile station will quickly associate with an AP in the second ESS. Any active network connections are likely to be dropped when the mobile station leaves the first ESS.

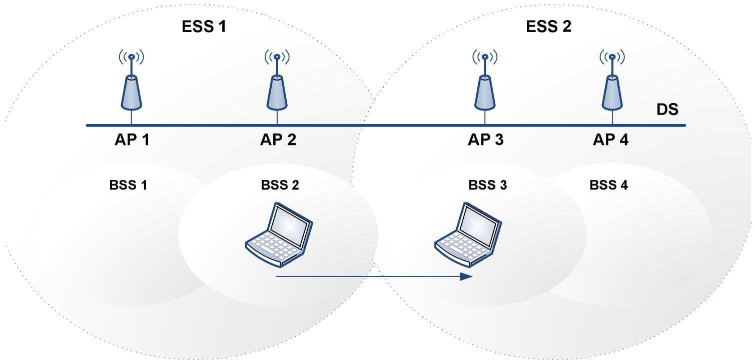


Figure 2.6: ESS Transition.

2.2.2 Handoff Notions

The great benefit of wireless networks is that a user can remain connected to a network even while in motion. A key concept in making sure that a user sees continuity in connectivity is the notion of a *handoff*. A handoff consists of the processes involved in switching a user's traffic in a cellar network from one antenna to another. Ideally, a handoff occurs seamlessly without the user even being aware that a handoff has taken place.

The technical challenges of this process have not yet been mastered, as evidenced by how often 802.11 network users are aware that the handoff is taking place, due to interruption and often temporary loss of connectivity. Nevertheless, one major goal of roaming in cellular networks is to provide a transparent handoff.

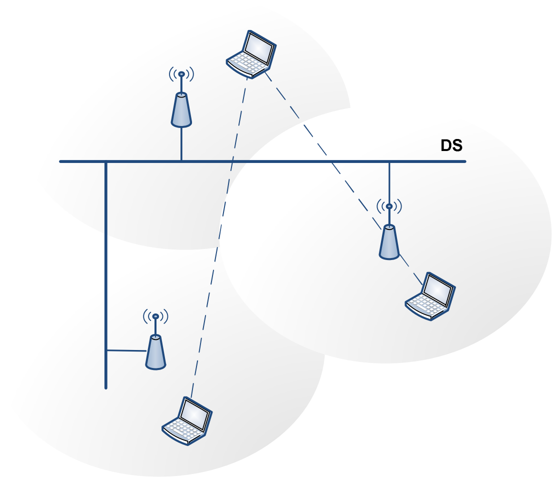


Figure 2.7: Example of an handoff.

2.2.3 Hard handoff / Soft Handoff

The hard handoff implies an abrupt break in the path that the user signaling takes, as it is routed into the network infrastructure.

Unlike a hard handoff, the soft handoff implies that both APs may simultaneously be receiving the user traffic from the mobile device. When both user traffic streams are being received by the network infrastructure, the network infrastructure can relatively simply switch between the stream arriving from the different APs.

In general, a soft handoff will provide a much greater likelihood of a seamless transition during the roam, but a soft handoff is not always feasible, either due to the particular technology in use or even due to where in the network topology the handoff occurs. The 802.11 technology has always been built on the principle of hard handoffs of user traffic.

2.2.4 Phases of 802.11 Roaming

There are many factors that actually contribute to the roaming delay as perceived by the 802.11 user. The problem with providing straightforward measurements for 802.11 roaming delays is that delay may differ in its meaning based on context.

In the presentation [6] the author shows the latency of the various phases of a 802.11 roaming. This values are show in the table 2.1.

One complication is with regard to deciding when to initiate a roam. Intuitively, one would guess that a good approach would always be to associate with the AP that currently presentees the strongest signal. However, this strategy does not work in practice. The RF domain is one that is subject to many variations, even when the STA is stationary. Nowadays the target AP is selected among those that display the strongest signal strength and the strongest match to other implementation-dependent criteria. Various factors that contribute to the roaming delay, as perceived by the user, are the following:

Discovery delay The discovery process consists of scanning and other measurement processes required for the STA (a) to determine that it needs to roam and (b) to determine the best AP to roam to. Part (a) is essentially a determination that indicates that the current AP is unacceptable. There are different metrics that can be used to make this determination:

Layer	Item	Time (ms)
L2	802.11 scan (passive)	0 (cached), 1000 (wait for Beacon)
	802.11 scan (active)	40 to 300
	802.11 assoc/reassoc (no IAPP)	2
	802.11 assoc/reassoc (with IAPP)	40
	802.1X authentication (full)	1000
	802.1X authentication (fast resume)	250
	Fast handoff (4-way handshake only)	60
L3	DHCPv4	1000
	Initial RS/RA	5
	Wait for subsequent RA	1500
	DAD (full)	1000
	Optimistic DAD	0
L4	TCP parameter adjustment (status quo)	5000 (802.11/CDMA) - 2000(802.11/GPRS)
Best case	All fixes	150
Average case	6to4, RR, Active scan	1300
No TCP changes, full EAP auth, IAPP, DHCPv4		2500

Table 2.1: Latency of 802.11 phases

- Observing an increase in retry counts due to transmission errors;
- A downward shift of the transmission rate due to the inability to communicate at higher speeds;
- Too many missed beacons;
- Too many frames received with errors;
- A direct measurement of decreasing Radio Signal Strength, as measured by the RSSI (*receive signal strength indicator*) parameter.

Reassociation delay This delay is the time required to complete the association with the new AP. It may involve the exchange of 802.11 authentication frames and involves an association-request/association-response exchanges. The beginning of the reassociation delay marks the definitive end of application data being passed through the current AP.

Authentication delay This delay is attributed to the authentication conversation between the STA and the network server. Depending on what version of 802.11 security is in place, this exchange may include several frames, often 13 or more, and theoretically hundreds of frames when the network admission control is running.

Key management delay This delay is the time taken by the four-way exchange of key management frames that are used to derive the session keys that will be used to protect the wireless link.

Application resumption delay Once the 802.11 peers have been programmed with the keys derived during the key management phase, there is some additional internal delay when the driver sends a LINK UP event to the upper-layer protocols and before they react to allow the application traffic to resume. Modern STA implementation will send an ARP frame to the default gateway to determine if the IP subnet has changed. In the event that there was a change of IP subnet, the delay involved using DHCP to obtain a new IP address is part of application resumption delay.

Infrastructure routing delay This last phase of delay captures delays that may occur in the infrastructure after the AP and the STA are completely ready to resume user-data flow.

In the case of a local roam, even thought the AP and STA are completely ready to resume data flow, in order for traffic destined from the network to the STA to reach the STA, it has to be routed via the new AP. Until the infrastructure reacts to this move, there is still a loss of connectivity to the network from the application's point of view.

Our work does not cover the *discovery delay* phase, nor the subsequent event that will trigger the handover. For the purpose of our work we assume that some discovery procedure as taken place, gathering the necessary information of the surroundings, and with that information we may prepare the STA for a possible handover using preauthentication. We also assume that there is a handover decision mechanism that will initiate the handover phase, and once that happens our protocol performs a fast reassociation.

Our work covers the rest of the handover phases in 802.11: the *reassociantion, authentication and key management delay* by conceiving a new 802.11 authentication and reassociation protocol, and the *application resumption and infrastructure routing delays* by using Mobile IP, and by performing network configuration before the handover takes place.

2.2.5 Network Discovery

When an STA wants to connect to a WLAN, it first needs to know the SSID (*Service Set ID*) of the network. There are two ways for a STA to know the SSID of the WLAN that he is trying to connect to: *passive scanning* and *active scanning* (see figure 2.8).

In passive scanning, an STA will iteratively listen on all channels that are available to it. While listening, an STA will receive beacons, learning the channels being used, the SSID that they are advertising, plus capability information and information elements.

The process of active scanning differs from passive scanning in that the STA actively searches for an SSID to which the STA wishes to connect. In order to discover an AP using a particular SSID of interest, probe requests are transmitted on the channels that are available to that STA until an AP having that SSID responds or the STA gives up.

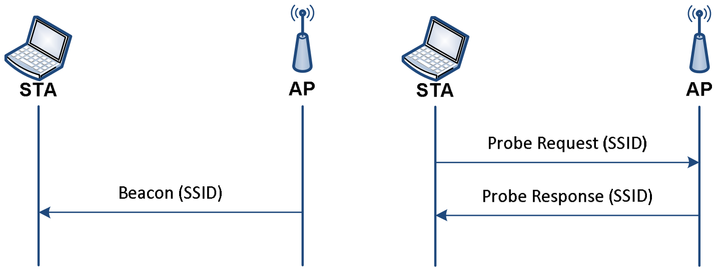


Figure 2.8: Network Discovery: passive scanning (left) and active scanning (right).

Our work does not address the problem of network discovery, but being this a critical phase of the 802.11 roaming we show here a good solution to address this problem, called *SyncScan* [7].

The major problem with network discovery is that the more time we spent scanning other channels, the less time we spent attending to the active channel and thus, more chance of missed frames. Thus, any heuristic that can increase the amount of usable information obtained per unit time spent away from the active channel would benefit the overall roaming throughput performance. This is precisely what SyncScan attempts to do

An important premise of the SyncScan work is that beacons are emitted by most APs at regular intervals. If the STA receives a beacon from an AP and knows the periodicity of that AP's beacon transmission, it follows that the STA can accurately predict when future beacons will be sent. AS the periodicity, or *beacon interval* of an AP is included as data within the beacon itself, the future beacon schedule for an AP can be calculated once the first beacon from that AP is received. The STA can organize its own background scanning schedule for an AP as a function of its predicted beacon schedule. By performing this operation on the set of APs that represent viable roaming candidates, the STA can gradually reduce the amount of time spent away from the active channel and still have a very high probability of receiving a beacon from one of those roaming candidates. This strategy allows the SyncScan enabled STA to maintain truly up-to-date information about signal strength from the roaming candidates without unduly affecting the active channel.

2.3 IEEE 802.11f (Recommended Practice for Inter Access Point Protocol)

IEEE 802.11f or *Inter-Access Point Protocol* is a recommendation that describes an optional extension to IEEE 802.11 that provides wireless access-point communications among multivendor systems.

2.3.1 Layer 2 Update Frame

The 802.11f defines a layer-2 update frame to be broadcasted by the new AP to all the layer-2 devices within the same subnet. Sent with the newly associated STA as the source, the layer-2 update frames will update the forwarding table in all the Medium Access Control (MAC) bridges and layer-2 switches so that future packets addressed to the STA will be forwarded to the new AP.

As defined in IEEE802.11f, upon the reassociation of a handoff STA, the new AP will send out a layer-2 update frame before any further actions. Since the layer-2 update frame is a layer-2 broadcast frame, the old AP (as a layer-2 device itself) will receive the frame as well. Next, the wireless interface driver for the old AP will receive this broadcast frame. By checking the source address of this frame, the old AP can identify that the STA was once in its BSS. The reception of this layer-2 update frame, therefore, signals that the STA has just moved to a new AP.

2.4 802.11i

2.4.1 Introduction

The 802.11i task group began in 2001 with the goal of providing safe secure access to 802.11 networks. This effort started as a result of the market's very negative reaction to the perceived vulnerabilities of WEP, which at that time was the standard way of protecting 802.11 network connection. The 802.11i amendment was the result of three years of intense debate and compromise, and it was finally ratified in 2004. The economic consequences of getting this particular part of the standard done correctly and quickly were enormous. If 802.11 could not ”get security right”, the expected wide adoption of the technology would never occur.

The terms *Robust Secure Networks (RSN)* and *Safe Secure Networks (SSN)* have both been used publicly by 802.11i to describe its general goal, although the finally ratified amendment uses RSN. The 802.11i amendment defines RSN procedures that occur during the association phase; these allow the client and AP to determine the security context of their particular association. The security context, at the most basic level, will establish whether the *personal* or the *enterprise security* mode is to be used.

The personal security mode is called *PreShared Key (PSK)* mode in the 802.11i standard parlance and its intended to provide easy-to-use, yet adequate for the SOHO[1] markets.

The enterprise security mode uses the IEEE 802.1X standard for authentication and key distribution. This mode is a much more robust security technology than personal mode, but a much deeper knowledge of security technologies is required in order to deploy enterprise security growth effectively. A more complex authentication infrastructure is also required. This complexity is the reason why the term ”enterprise” is used.

2.4.2 802.11i Communication Security

Regardless of whether personal security mode or enterprise security mode is used, and regardless of the cipher used, 802.11i security needs two keys: a *Pairwise Transient Key (PTK)* and a *Group Transient Key (GTK)*. The PTK is used to protect unicast traffic both from the STA to the AP and from the AP to the STA. The GTK is used by the AP to encrypt broadcast/multicast traffic sent to all STAs currently in the BSS, and the STAs need this key in order to decrypt the traffic. The basic operation is as follows:

1. The STA associates and then negotiates the security parameters used with the association.
2. The AP authenticates the user in enterprise security mode. This step does not exist in personal security mode.
3. A four-way key validation and distribution protocol is executed such that the PTK becomes available in the STA and on the AP.
4. The agreed-upon temporal keys are installed, using the negotiated cipher, and subsequent frames are protected.

In the final version of the 802.11i standard, corresponding to WPA2, the encrypted GTK is sent with the message three of the four-way handshake. In the interim version that corresponded to WPA, the GTK was communicated in a two-way handshake dedicated to communicating the broadcast key. When the GTK is updated by the AP for all STAs in the BSS, the two-way handshake is still used by itself, both in WPA and WPA2.

Step 3 is identical in both personal and enterprise modes in that the four-way handshake is used to derive a PTK from a *Pairwise Master Key (PMK)* in both cases. The two modes differ with respect to the source of the PMK. For personal mode the PMK is already present in both the STA and the AP before step 3. This situation is in fact the case because the PMK is part of the PreShared Key, which is a static pass-phrase encoded on the AP and many STAs expecting to associate with it. In the case of enterprise security mode, the PMK is dynamically derived through the authentication process that

[1] Small Office, Home Office

occurs in step 2. This method adds greater degree of entropy to the enterprise mode process since the PMK itself is fresh for each session. The figure 2.9 shows the 802.1X key hierarchy.

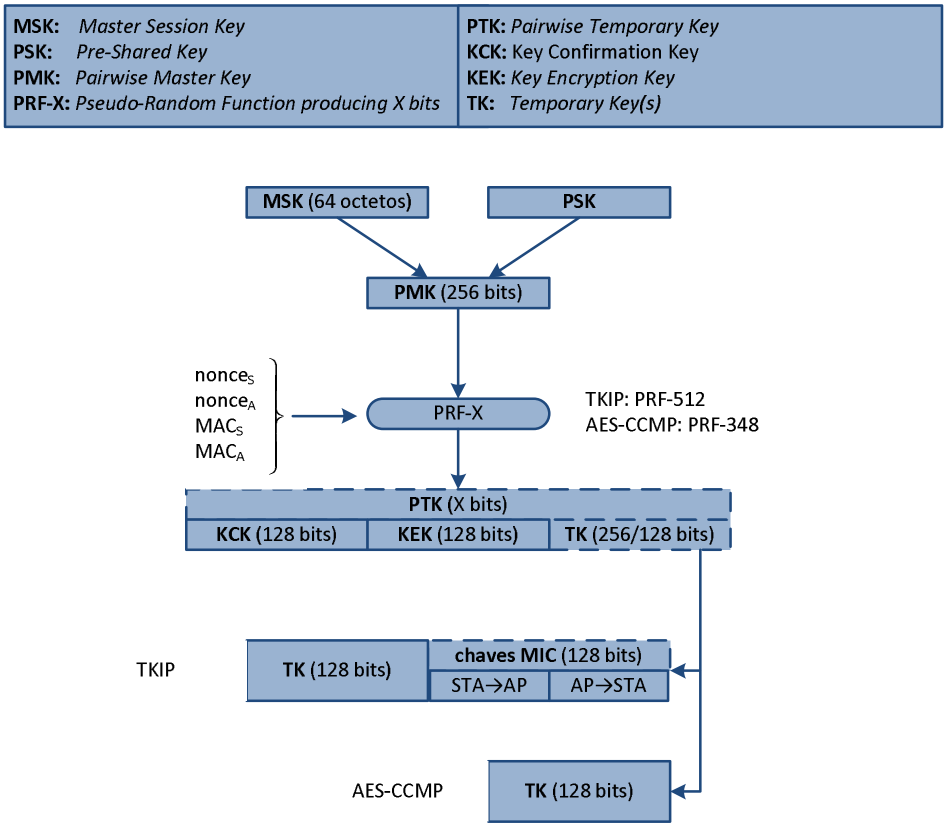


Figure 2.9: 802.1X key hierarchy.

2.4.3 Preauthentication in 802.11i

The finally ratified 802.11i standard reflected the work group i best effort to allow fast secure roaming within the existing constraints of 802.11. One of the last major features added to 802.11i was *preauthentication*. This feature is highly relevant to the discussion on roaming because the security gained from the task group i work is achieved at the expense of additional complexity. This complexity manifests itself in terms of latency between associating and being able to use 802.11 for effective user communications. If this lag occurs only once at the beginning of a long session, the additional delay, ranging from hundreds of microseconds to a few seconds, may be acceptable. In a mobile roaming environment, a brute-force implementation of 802.11i security would result in this penalty being paid at the start of each and every roam. Task group i recognized this solution as unacceptable for a technology that is increasingly being considered as a medium for the mobile VoIP user. Such user cannot be subjected to multisecond interruptions in phone calls while wandering down the corridor.

Preauthentication lessens latency by caching some of the keying material derived during the authentication in neighboring APs to which the user is likely to roam. This advanced work is accomplished by performing the authentication for the candidate APs through the current AP and distribution service, usually a wired LAN infrastructure, connecting it to the candidate AP. This technique permits control dialog between the STA and roaming candidate APs without disruption of the current data flow, as would be the case in basic 802.11, which presumes that the STA needs to associate with the candidate AP in order to communicate. This exchange would in fact be necessary were the candidate APs are not connected via the same DS, but this situation is rarely the case. Despite the decreased latency, even preauthentication really falls short of being able to provide sufficiently low latency to permit acceptable roaming.

A presentation by IEEE [8] focus on the problem of Pre-Authentication and what an STA can and cannot do while in the Authenticated/Associated state with an AP and trying to talk to another AP with which the STA is the Unauthenticated/Unassociated state.

When a STA is not associated with an AP it can only send management frames. That's why some protocols pre-authenticate the STA via the current AP. In this case a STA sends 802.1X frames encapsulated in data frames so that the current AP can forward then to the new AP. This creates complexity in the network because for security purposes security associations need to exist between neighbor APs. So neighbor graphs need to be created an new entities need to be added to the network infrastructure to manage these neighbor graphs and manage a whole new set ok keys. For this is also needed some kind of centralized architecture, like CAPWAP, for inter access point protocol that will create some overhead of management frames on the network.

Steps Involved in 802.11i Preauthentication

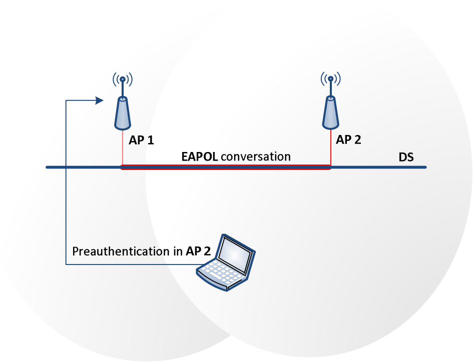


Figure 2.10: An example of a STA currently associated to AP 1 and wishing to preauthenticate with AP 2.

The client STA first associates with AP 1 and then performs normal 802.11i authentication procedures. At this point, preauthentication diverges from the mandatory 802.11i steps. First, the STA becomes aware of neighboring candidate APs through the scanning procedures. These APs are placed on the current driver scan list. These candidates will normally be other APs in the same ESS with sufficiently strong *Received Signal Strength (RSS)* to be considered candidates. The STA then initiates a normal EAPOL conversation with each of the selected candidates via the current AP.

This is a fundamental step in preauthentication, as it solves the problem of not being able to communicate with candidate APs without breaking the association with the current AP. (Recall that 802.11 infrastructure mode requires that an STA must be associated with an AP in order ro send data frames to it and that an STA can only associate with one AP at a time.) By interacting with the candidate APs through the current AP and hence through the wired network, the STA can carry out the normal EAPOL conversation that is necessary to preauthenticate with another AP according to the 802.11i specification.

The EAP conversation carried in the EAPOL frames must be relayed to some AAA server, but this portion can be done as it normally is, using RADIUS packets as the transport over the wired network.

Each of these preauthentications will derive a unique PMK, one for each pairing of a STA with an AP. The STA must keep track of which PMK belongs to each of the APs with which it has preauthenticated. Both STA and AP maintain an identifier called the PMK Identity (PMKID) that enables both to confirm that they are using the correct PMK when they begin their four-way exchange.

2.4.4 802.11i Denial of Service

One of the problems of 802.11i is that management frames are not secured. This situation is suitable for the occurrence of, very simple, though effective DoS attacks.

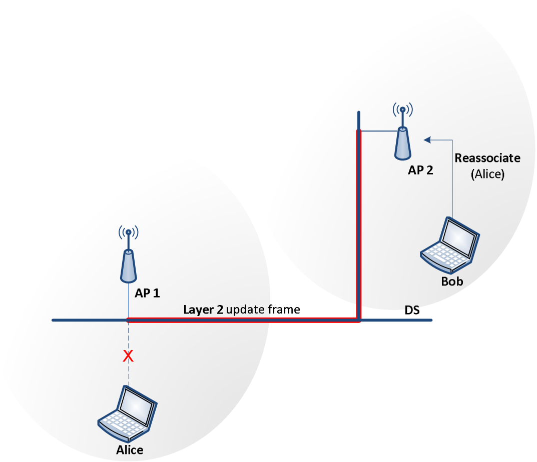


Figure 2.11: DoS attack trough a reassociation frame (a).

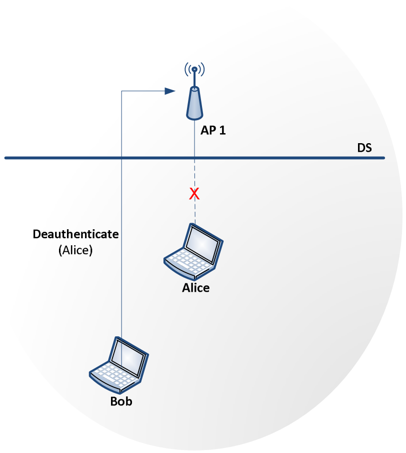


Figure 2.12: Dos attack trough a deauthentication frame (b).

In the example of figure 2.11 the attacker Eve sends a reassociation frame to AP 2 on behalf of Alice. After receiving the reassociation frame, AP 2 will send a layer 2 update frame to the wired network so that AP 1 can deauthenticate Alice, and all layer 2 equipments can update their bridging tables forwarding the packets targeted to Alice to AP 2. In this case Eve does not need to be in the same BSS of Alice. The attack can be done in any AP of the same ESS, because the layer 2 update frame is a broadcast frame.

In the example of figure 2.12 the attack is only possible in the same BSS. Eve impersonates Alice and sends a deauthentication frame to the AP.

These attacks are possible because in the 802.11i protocol management frames are not authenticated.

As we will see later on 4, our protocol deals with this problems in a simple way. First all the frames exchanged by our protocol are simple 802.11 management frames with some extra payload fields, second all the frames are secure so that mutual authentication can occur between all entities in our protocol.

2.4.5 IEEE 802.11i: Fast handoff issues

The 802.11i architecture and its authentication components (802.1X), in an attempt to provide strong mutual authentication, ended up with a time consuming protocol with several messages exchanges, and, therefore almost impracticable when our goal is to achieve fast handovers. The question is: ”Why should we have to run a full 802.1X authentication with the AS every time an handover occurs, if the AS already knows the STA from the initial 802.1X authentication?” The answer is simple: ”We don't.” During the handover the only mutual authentication that as to exist is between the STA and the new AP. This is only trust relationship that does not exist prior to the handover. Another issue that is still to be addressed is denial of service vulnerabilities as seen in 2.4.4.

In [9] a retrospective of 802.11i is made outlining what was accomplished and what is missing in 802.11i. The author states that denial of service vulnerabilities are only partially addressed and there is no mandatory-to-implement authentication or key management scheme. Denial of service attacks can be easily performed do to the fact that 802.11i does not secure management frames. The author also points the fact that there are two many exchanges in the negotiation of new key material when a Association/Reassociation occurs, and proposes a new modified 802.11 state machine (see figure 2.13.

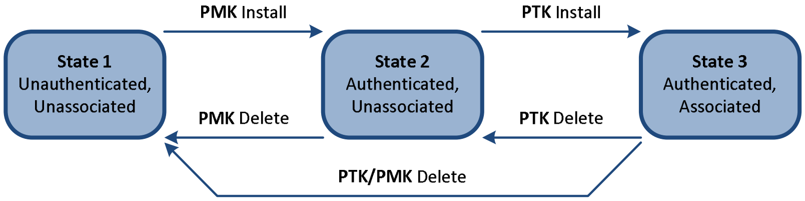


Figure 2.13: New 802.11 state machine.

We addressed this issues by creating a new key hierarchy. The root key of this hierarchy is the EMSK that is derived from the initial 802.1X authentication. The new keys are then pre-distributed to the entities involved in the handover, allowing mutual authentication between the STA and the new AP. Doing this we eliminate the need of running a 802.1X authentication, and we authenticate all messages exchanged, thus, preventing denial of service attacks.

2.5 IEEE 802.1X

IEEE 802.1X is an IEEE standard for port-based Network Access Control; it is part of the IEEE 802 (802.1) group of protocols. It provides authentication to an STA willing to connect to a WLAN. Its function is to establish a point-to-point connection or preventing access from that port if authentication fails. 802.1X is also used to create and distribute fresh session keys both to the STA and AP so that they can secure the messages exchanged via the wireless medium.

The standard IEEE 802.11 admission-control model includes three components:

- A client based 802.1X supplicant
- An AP based 802.1X authenticator
- A back-end Authentication Server (AS), typically a AAA RADIUS server

These three items, in combination, provide a robust, and scalable admission-control system that works over many media types, including 802.11. Although it is not obvious that the authentication model is of direct importance to the roaming topic, the reader requires a basic understanding of this paradigm in order to absorb the material on fast secure roaming.

2.5.1 The Extensible Authentication Protocol

The early applications of RADIUS technology relayed user credentials to an AAA server via protocols such as CHAP and PAP. Over time, it was necessary to modify and to extend these and similar protocols in order to accommodate new aspects of authentication. These modifications and extensions forced changes in different network components, some of which were not directly involved in the details of authentication. This natural evolution was the genesis of the EAP standard RFC 2284 [10], which was intended to allow different and extensible authentication methodologies to be encapsulated within the EAP transport protocol. This encapsulation permits the authentication methods to change and to mature over time without forcing unnecessary modifications to different network components. In its simplest form, EAP-encapsulated user credentials may be sent to the AAA server for validation much as they are for CHAP and PAP.

2.5.2 The Authentication Server (AS)

The role of the AS server is to make authentication and authorization decisions for the network. The server processes the credentials transmitted from the client requesting admission and the server grants or denies access based on those credentials and the access being requested. When access is permitted, the granting may be accompanied by the authorization of different kinds of access. The AS server may also start and stop the recording of accounting information related to that user's session.

2.5.3 User Authentication with 802.1X in 802.11

The full process of authentication in a WLAN using the 802.1X can be divided in three major stages (see figure 2.14).

1st Stage: 802.11 discovery and association

In this stage o supplicant (STA) connects to the WLAN. The STA starts a normal discovery process, followed by the authentication and association of the STA with the target AP. Because this first authentication is not important once we are using 802.1X, it's normal to use the OSA (*Open System Authentication*) in this stage. At the end of this stage the STA is authenticated and associated with the

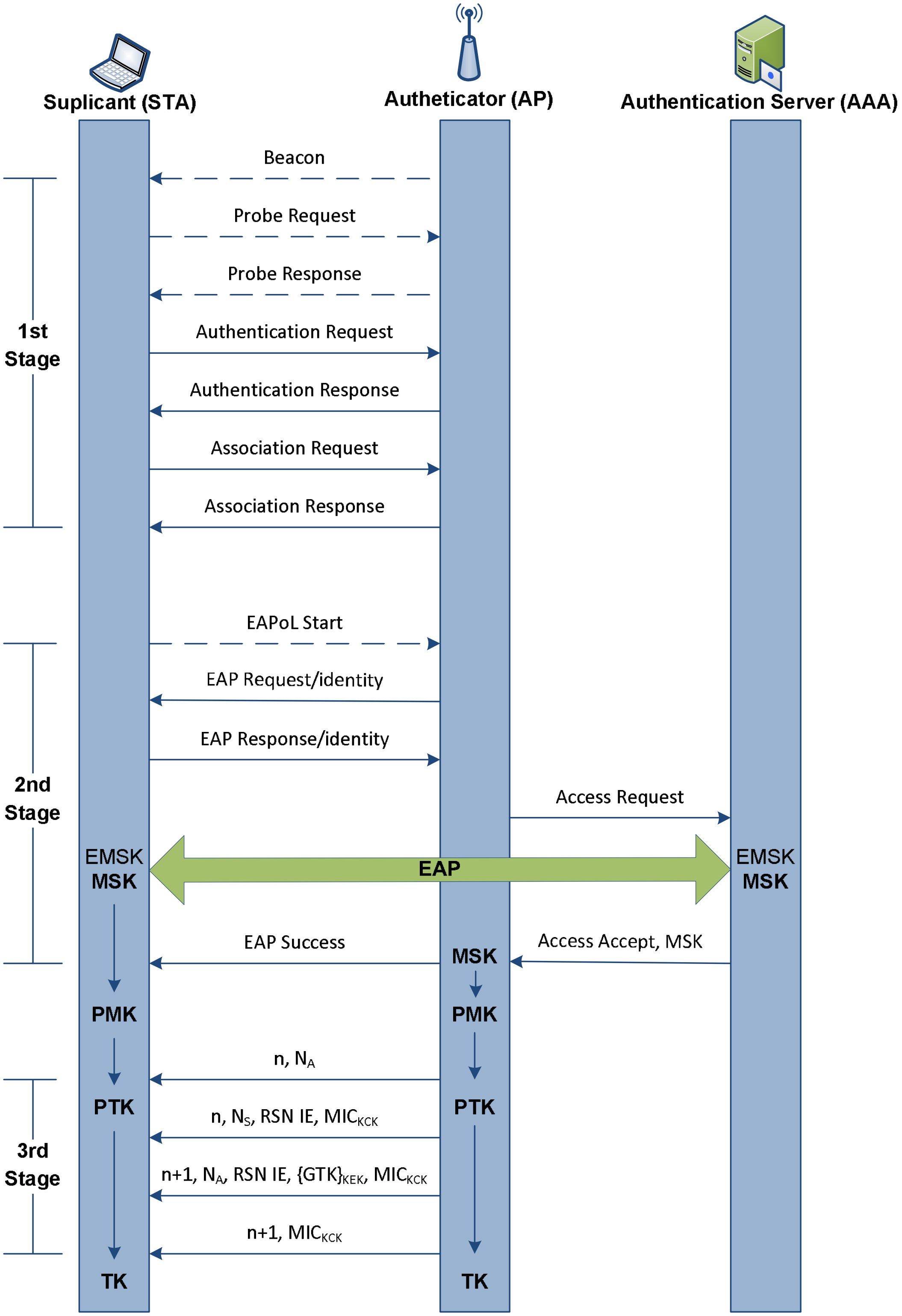


Figure 2.14: Full 802.1X Authentication.

target AP.

2nd Stage: EAP authentication

1. Upon detecting the 802.11 association, either the supplicant or the authenticator may send an EAPOL Start message.
2. The authenticator opens the uncontrolled port for the 802.1X authentication session, leaving all non 802.11X traffic blocked at the controlled port.
3. The authenticator sends an EAP Request/Identity message.
4. The supplicant's EAP Response packet with the user identity is passed to the authentication server in the first RADIUS Access Request message.
5. The authentication server challenges the supplicant to prove itself, and the server may send its credentials to prove itself to the supplicant (if the supplicant requires mutual authentication). This information is encoded in an EAP message and sent to the AP in the payload of a RADIUS Access Challenge message, which is forwarded to the supplicant by the AP in an EAPOL message
6. The supplicant sends its credentials to the server in order to allow the server to verify the client's identity. The client's credentials are embedded in an EAP message and transported to the AP in a EAPol message and relayed by the AP to the authentication server in the second RADIUS Access Request message.

3rd Stage: four-way handshake

The authentication process leaves two considerations: the authenticator still needs to authenticate itself to the supplicant (STA), and keys to encrypt the traffic need to be derived. The earlier EAP exchange has provided a shared secret key PMK (*Pairwise Master Key*). This key is however designed to last the entire session and should be exposed as little as possible. Therefore, the four-way handshake is used to establish another key, called the PTK (*Pairwise Transient Key*). The PTK is generated by concatenating the following attributes: PMK, AP nonce (ANonce), STA nonce (SNonce), AP MAC address and STA MAC address. The result is then put through a cryptographic hash function.

The handshake also yields the GTK (Group Temporal Key), used to decrypt multicast and broadcast traffic. The actual messages exchanged during the handshake are depicted in the figure 2.14 and explained below:

1. The AP sends a nonce to the STA (*ANonce*). The STA now has all the attributes to construct the PTK.
2. The STA sends its own nonce (*SNonce*) to the AP together with a MIC (*Message Integrity Code*), for message authentication. The AP now as all the attributes to construct the PTK.
3. The AP sends the GTK and a sequence number together with another MIC. The sequence number is the sequence number that will be used in the next multicast or broadcast frame, so that the receiving STA can perform basic replay detection.
4. The STA sends a confirmation to the AP, also authenticated with a MIC

2.6 Opportunistic Key Caching

Above we presented the IEEE 802.11i preauthentication. The purpose of this complex mechanism is to place keying material in roaming candidate APs in advance of the roam itself. If we free ourselves from the bounds of the IEEE 802.11i standard and its laudable goal of wide interoperability, there are mechanisms that go beyond the standard to address this problem, which are simpler and sometimes more powerful than preauthentication. In particular, if a network implements a central management device controlling a *mobility domain* of a set of APs, it is possible for that central manager to make the 802.11i PMK available through each of the APs in that mobility domain. Such a centralized manager can be implemented via traditional *autonomous* APs using vendor specific inter-AP protocol, but it is the centralized wireless switch manufacturers that have been quickest to offer key caching solutions. This solution is natural because with these arquitectures, the PMK is readily transferable to the *lightweight* APs under the control of the wireless switch. Indeed, in some centralized switch arquithectures, the PMK need never be sent to the AP at all as the encryption is performed entirely whitin the switch itself rather than in the AP. It should be noted that the term ”fat” AP is sometimes used to denote an autonomous AP, and the term ”thin” AP is often used to denote a lightweight AP.

It is worth nothing that both the inter-AP protocol as well as the AP-to-wireless switch protocols could be standarized, but, to date, they have not been. The LWAP and CAPWAP efforts are extant examples of such efforts. This fact is most likely because this area is where individual manufacturers believe that they can add value over and above the 802.11 standard and, hence, are reluctant to give away this advantage.

In whatever manner this key caching occurs, in order to benefit from it the association exchange with the AP must denote that the AP possesses the PMK, which was obtained via one of these vendor specific means, and thus the associating client understands that it can forego full authentication and that it may proceed directly to the four-way handshake which derives the PTK. This process is called *Opportunistic Key Caching (OKC)* as the process does not prescribe how the PMK reaches the target AP, just the manner in which the STA will *opportunistically* take advantage of that situation.

OKC relies in a new 802.11i information element, PMKID, that was defined to support preauthentication. The 802.11i standard does mandate that the PMKID information element must be understood and correctly processed by compliant APs and STAs. The original intent in 802.11i was that the PMKID would identify a PMK that had arrives at the target AP, as a result of an earlier preauthentication. OKC is a matter of the client ”guessing” that it is associated with a wireless switch, and therefore, instead od performing preauthentication with possible roaming candidate APs, it assumes on a roam that the target AP has knowledge of the current PMK. This information is encoded in the PMKID, which is sent in the Reassociation Request as an information element. If the PMK is known by the new AP, then the authentication reduces to just the four-way handshake and the roam is fast; if the PMK is not known, then a full EAP reauthentication must take place. A wireless switch configured to do OKC will not advertise preauthentication capability; the absence of preauthentication stimulates the client to try OKC. The worst that can happen is that OKC attempt will fail and full authentication will take place. In practice, this situation does not happen in an OKC-enabled mobility domain.

While OKC’s performances equals that of preauthentication in terms of the reduced frame exhange at the time of the roam, it is far superior to preauthentication in terms of the authentication overhead on the AAA server. The preauthentication client anticipates potential roams to a number of neighboring APs and completes full 802.1X style authentications with each of them, including some to which the client may never roam. A common criticism of preauthentication is that it can actually increase the total authentication load on the AAA server as compared to performing a single full 802.11i authentication with each actual roam. OKC is superior to both of these in that there is a single 802.11i authentication when a client enters a mobility domain, and no further authentications as long as it roams within the domain. The only thing that would force a full reauthentication within that mobility domain would be the expiration of an AAA session timeout, and these intervals are configurable by the IT manager and are typically quite long.

2.7 802.11r Fast BSS Transitions

The original 802.11 standard does incorporated roaming in a simple form. However, the limited speed and security conferred by the original standard makes it essentially of no use for secure voice applications. The 802.11r task group delas with the standarizing aspects of 802.11 that will reduce the transition time during the roam, referred to as a BSS transition in 802.11r.

With regard to the principal 802.11 amendments required for fast secure roaming for VoIP applications, the 802.11r standard is less mature than the others. By less mature, we mean that the other standards are either already ratified or closer to ratification than 802.11r.

The 802.11r standard defines the term *mobility domain* as the set of fast-transition-capable APs to which an STA can roam at the moment. All the APs in a mobility domain are interconnected via a single DS. In order to describe 802.11r capable STAs and APs, the amendment introduces the terms *Fast Transition Enabled STA (TSTA)* and *Fast Transition Enabled AP (TAP)*.

2.7.1 Basic Service Set Transition Pre-802.11r

Based on the pre-802.1r techniques, a BSS transition for a secure QoS sensitive application required the following five stages:

1. Scanning for target APs.
2. Open 802.11 authentication.
3. Reassociation.
4. PTK derivation. The complexity of this step varies depending on whether key caching, preauthentication, or new complete 802.1X reauthentication is involved in providing the PMK at the new AP. Even in the abbreviated cases of key caching and preauthentication, a four-way handshake is required to derive the PTK.
5. Qos admission control with the new AP.

onsidering these five serialized stages, even if we assume the shorter latency of key caching or preauthentication, the total time for the BSS transition will likely be measured in the many tens of miliseconds or even more, which will disrupt a voice conversation. In addition, the fact that the QoS admission occurs only at the end of this leghty process implies that admission may fail and the roaming STA will have to start over and explore another potential candidate.

2.7.2 Overview of the 802.11r Standard

The phrase ”Fast BSS Transition” implies that the 802.11r standard attempts to define procedures and protocols that result in an 802.11 roam. In reality, 802.11r will be able to produce a *faster* security and QoS-sensitive BSS transition than would be possible without 802.11r. The standard does this by addressing the part of the delays in the handoff procedures that lie in the areas of 802.11i security and 802.11e QoS.

The delays associated with detecting the need to roam and the selection of a target AP (Discovery Delay) as well as the resumption of downstream application data flows (Application Resumption Delay and Infrastructure Routing Delay) are all beyond the scope of 802.11r, yet contribute significantly to the roaming delays experienced by the user. As more time may be spent on the activities that is spent establishing the QoS and security context during the roam, the net roaming experience, even with full compliance with 802.11r, may still fall short of users expectations.

The 802.11r standard defines many terms in an attempt to impose order on the chaos in the earlier discussin on 802.11 that centered around fast secure roaming. Two of the most fundamental new terms are Mobility Domain (MD) and *first contact.* First contact for a given end-user station is defined to be the initial association of the STA with an AP in that MD. The procedures for this initial association are different from that for STA's subsequent association in the MD. These subsequent associations are the Fast BSS Transitions, which are the focus of 802.11r.

The bulk of 802.11r relates to reducing the security overhead during BSS transitions. The two most obvious security-related benefits are clarification of how the opportunistic key caching is accomplished and the elimination of the four-way handshake that traditionally followed reassociation. With regard to the postassociation four-way exchange, this elimination is acomplished by overloading the four-frame Authentication-Association exchange with new IEs that contain this information. In the case of OKC, the 802.11i standard did not stipulate the means by which the target AP obtained the keying information that the STA and AP developed during first-contact procedures. Instead, the term *opportunistic* was used because the first-contact keying information somehow ”magically” reaches the target AP.

Opportunistic key caching assumed that the necessary keying information would be made available to the target AP either by some vedor-specific inter-AP protocol or, in the case of a wireless switch architecture, due to the fact that the four-way exchange was actually centrally controlled for all the APs in the ESS and that centralized controller was in possession of the PMK all along. The fact that the process was not dictated vy the standard resulted in an inconsistent application of OKC. The 802.11r standard addresses this problem by defining a new key hierarchy and the concept of MDs as dsicussed subsequently.

A group of APs jointly form a single MD and in doing so gain access to a common key hierarchy. This can occur easily if the MD is designed to consist of a set of APs connected to a centralized controller or a set of controllers from the same manufacturer, but specification of these mechanisms is beyond the scope of 802.11r. Such a centralized controller is sometimes referred to as the *Mobility Domain Controller (MDC)* in the 802.11r context. A centralized wireless switch fits naturally into this role.

During a roam, if the target AP advertises membership in the desired MD, the client may associate with it with the hope that the needed key hierarchy is present. It should be noted that 802.11r does not guarantee that the needed key hierarchy be present when a roam occurs, in which case unexpected latency may occur wherein the new AP needs to collect this information from its holder. When the STA performs its initial association and full authentication in that MD, it gains access to the 802.11r key hierarchy. This key hierarchy contains PMK-R0 material that is related to the 802.11i PMK. From this information a PMK-R1 key is derived, which is specific to that AP-STA pairing. The important fact is that a target AP in the same MD can independently derive a new PMK-R1 equal to a new PMK-R1 derived by the client attempting to associate with the AP based on the client's BSSID and possession of the key hierarchy. This process can occur without resorting to the RADIUS server or any other external device, presuming that the AP has the key hierarchy.

The exact manner in which that target AP obtains the keying information is implementation dependent. As of this writing, no one anticipates the IEEE standardizing the inter-AP protocol. Such an effort specify how this information could be exchanged betwen different manufacturers' APs. Indeed, the already formalized IEEE 802.11f work, targeted to standardize inter-AP protocols, has been *rescinded*. Because of this, it seems unlikely that IEEE standardization of such inter-AP protocols will be realized in the near future. It seems more likely that any related standardization will occur trough the work of the IETF CAPWAP group, which focuses on communication between APs and the infrastructure switches that connect them.

2.8 Centralized Wireless Switch Architectures

The classical 802.11 deployment consists of a wired 802.3 switch with a number of intelligent APs connected to the switch, providing the 802.11 radio access to the network infrastructure. In this traditional architecture, 802.11 intelligence is distributed among intelligent APs. These APs are responsible for generating and processing 802.11 management frames, for maintaining and reporting frame statistics, and for handling security features, including authentication and encryption.

In the centralized architecture, some or all of these functions are moved back into a centralized wireless switch. Control of the association, authentication, and encryption processes may all be performed by the switch rather than by the APs. The fact that the intelligence of scores of APs is concentrated in one switch is an enormous lever that can be used to the best advantage for any problem requiring coordination among APs. To be fair, such concentration can also present scaling challenges, but successful centralized switch manufacturers have proven that these are surmountable.

One of the best examples of such a problem requiring the coordination between APs, is of course, the hand-off of an STA from one lightweight AP to another AP during the course of a roam. This centralized architecture is attractive enough that a number of companies now offer 802.11 products based on this architecture. Indeed, the companies that make the boldest claims about fast-roaming support are those companies espousing this architecture; this greater speed follows directly from the simpler inter-AP coordination that the architecture allows. These products generally conform to the 802.11 standards on the radio side, although vendor-specific protocols are the norm for communication between the APs and the switch.

2.8.1 MAC Processing

Within the general centralized architecture family, there are very divergent architectures. One such area of divergence is in how much of the MAC layer is actually implemented on the AP.

This can range from all of the MAC layer, in the case of the traditional autonomous AP, to none of it, in the case of pure-radio lightweight AP. When no MAC layer processing occurs on the AP, we call this a *local MAC* implementation. All packet processing, including QoS management and encryption related functions, occurs only in the controller. As the APs are not involved in encryption at all, there simply is no need for the AP to possess the PMK ever, making OKC a natural fit for the local MAC architecture. Aruba is a prime example of a wireless switch manufacturer that uses local MAC processing.

2.8.2 LWAPP, CAPWAP, and SLAPP

An early IETF effort to standarize a communication protocol between a wireless switch and a lightweight AP was led by the *Light Weight Access Point Protocol (LWAPP)* working group. LWAPP referred to the wireless switch as an *Access Router (AR)*. While some vendors have built implementations around drafts of LWAPP, it was never ratified, and no real multi-vendor interoperability has resulted from this work. A more recent activity in the IETF to standarize such protocol is active under the auspices of the *Control And Provisioning of Wireless Access Points (CAPWAP) Working Group*. In this group, the wireless switch is referred to as the *Access Controller (AC)*. CAPWAP is in its early stages and may face resistance from some of the manufacturers which have used the vendor-specific nature of that communication as a means of differentiating their products. CAPWAP has selected LWAPP as the basis for its standarization efforts.

Another effort, the *Simple Access Point Protocol (SLAPP)* was intended to be less ambitious and thus avoid some of the resistance faced by its predecessors, but this effort appears to have lost momentum, as of this writing. One of the reasons that these standarization efforts have been run in the IETF rather than the IEEE is the notion that the communication should be at layer three, so that the APs may be located more than a single hop from the AC. The features covered by CAPWAP will offer real improvements in pratical roaming performance across multiple vendors, so considerable importance is attached to this standardization effort.

2.9 Pre-Keying

One of the problems of the 802.11i is the long latency that occurs every time the STA as to re(authenticate) and (re)associate when a handover occurs. For instance, VoIP requires 802.11 transition times, including 802.11 security setup, below 50 ms. This is one of the problem stated in [11].

Our work as we will see in chapter 4 also meets the design goals proposed in this document:

- Make 802.11i keys available before association.
- Reuse 802.11i framework to make keys available: do not redesign 802.11i infrastructure and minimize amount of new invention.

Our protocol do make keys available before association, by authenticating the STA before the handover occurs trough what we call an *Reauthentication Server (RS)* that derives from IETF HOKEY server refhokey-key-mgm-01 that will be discussed in the next section.

We also reuse key material from the first 802.1X authentication and there no need to redesign the the network infrastructure, and the only new entity that we add to the network is the RS, that can coexist in the AAA server.

We also solve most of the issues present in this document [11]. We prevent that the same PTK is used across two association. Our protocol derives a new PMK for every STA-AP pair and during reassociation it derives fresh temporary keys including a new GTK.

One of the major benefits of our protocol compared to other protocols that use pre-keying is that we only derive the the temporary keys during reassociation (at the cost of two messages and not four like it happens in 802.1X with the four-way handshake), so that we can eliminate the problem that happens when the some key expires before the reassociation, for example the GTK updates.

Our protocol also specifies where do the keys come from or how are they derived and it also defines the mechanism by which this keys are distributed.

2.10 HOKEY Server

Our protocol uses a variety of design goals specified by the *IETF Handover Keying (hokey) Working Group*. To support our protocol we introduced a *Reauthentication Server (RS)* that must exist in the domain. This RS follows the work made by the IETF HOKEY WG. The HOKEY WG tries to achieve fast handovers by eliminating unnecessary re-execution of EAP authentication every time a handover occurs, once that the same EAP authentication server is used. The idea is to use unexpired keying material from a full EAP authentication to avoid a full EAP exchange with the AAA server to extend a session. The HOKEY WG uses the EMSK as the root of a cryptographic key hierarchy that are used to provided the needed security services. Guidelines are also specified for deriving the child keys, to ensure that all the levels of the hierarchy are cryptographically separated [2]. Different uses for keys derived from the EMSK have been proposed. Some examples include handoff across APs in various mobile technologies, mobile IP authentication and higher layer application authentication. The goal is to have security properties of one usage have minimal or no effect on the security properties of other usages.

The solutions specified by the HOKEY WG fall into several categories, based on timing and mechanism. The authentication and key management may occur before handoff, when latency is much less critical. Solutions should reduce or eliminate the number of referrals to AAA servers, and solutions should avoid re-executing lengthy EAP method exchanges.This may be accomplished by providing new mechanisms for cryptographic keying material in combination with a protocol for the timely delivery of appropriate keys to the appropriate entities.

Our RS follows the HOKEY WG guidelines, and is used as a key holder of the domain to eliminate the need of communication with the AAA server that may not reside in the domain and may be several hops away. This way we not only eliminate latency introduced with the round trip delays of the communications with the AAA server, but we also confine all the extra management needed to support or protocol to the domain.

2.11 Mobile IP and its Role in 802.11 Roaming

When an STA roams from an ESS to another ESS, it is normal to be assigned a new IP address belonging to the target subnet. This results in the applications having to make a new TCP/UDP connection with the new IP address. To the extent that the STA is ”known” to other network entities by its IP address, if the IP address changes every time the STA roams to a new location, this readdressing disrupts those entities. *Mobile IP*[2] attempts to address this issue. With Mobile IP, the STA retains the same static IP address even though it connects to the Internet from different geographic locations.

When a source computer wants to send a packet to a destination computer, the source does not know or care where the destination is located, it just wants its packets to be delivered to the proper recipient. This is the function of the *network layer*, Layer 3 of the Open Systems Interconnection (OSI) Model. The network layer is responsible for dynamically selecting a path from the original source of a packet to its ultimate destination. In the Internet, the network layer protocol is named *Internet Protocol (IP).*

IP accomplishes very little by itself. Typically, one or more routing protocols are necessary to move packets around a complex network. Routing protocols are used by routers to exchange information about the location of the various destinations and links which comprise the Internet. Examples of routing protocols include *Open Shortest Path First (OSPF)* [12], the *Routing Information Protocol (RIP)*, and the *Border Gateway Protocol (BGP)* [13].

Mobile IP is a network layer solution to node mobility in the Internet. By this we mean that Mobile IP accomplishes its task by setting up the routing tables in appropriate nodes, such that IP

[2] The name Mobile IP comes from the *IP Routing for Wireless/Mobile Hosts (mobileip)* working group of the Internet Engineering Task Force (IETF)

packets can be sent to mobile nodes not connected to their home link.

2.11.1 Mobile IP Terminology

Mobile IP defines three functional entities where its mobility protocols must be implemented:

1. **Mobile Node**: a node which can change its point of attachment to the Internet from one *link* to another while maintaining any ongoing communications and using only its IP *home address*.

2. **Home agent**: a router with an interface on the mobile node's *home link* which:

   (i) the mobile node keeps informed of its current location, represented by its *care-of-address*, as the mobile node moves from link to link;

   (ii) in some cases, *advertises reachability* to the *network-prefix* of the mobile node's *home address*, thereby attracting IP packets that are destined to the mobile node's *home address*;

   (iii) intercepts packets destined to the mobile node's *home address* and *tunnels* them to the mobile node's current location; i.e, to the *care-of-address*.

3. *Foreign Agent*: a router on a mobile node's *foreign link* which:

   (i) assists the mobile node in informing its home agent of its current *care-of-address*;

   (ii) in some cases, provides a *care-of-address* and *de-tunnels* packets for the mobile node that have been *tunneled* by its home agent;

   (iii) serves as a default *router* for packets generated by the mobile node while connected to this *foreign link*.

Home Address, Home Link and Home Agent

A mobile node's *home address* is an IP address assigned to the mobile node ”permanently”, i.e., for as long as an address would be assigned to any stationary host or router. The home address *does not* change as a mobile node moves from link to link. Rather, a mobile node's home address would change only for the same reasons, and under the same circumstances, as the address of a stationary host or router would change (e.g., if the network required renumbering).

The mobile node's home address is closely related to the mobile node's home agent and therefore its home link. Specifically, the network-prefix of the mobile node's home address *defines* its home link. That is, a mobile node's *home link* is that link which has been assigned the same network-prefix as the network-prefix of the mobile node's home address. A mobile node's *home agent* is a router that has at least one interface on the mobile node's home link.

With very few exceptions, a mobile node communicates with all other nodes using only its home address. That is, a mobile node's home address is the IP Source Address of all packets sent by the mobile node and the IP Destination Address of all packets sent to the mobile node. This requires a mobile node's home address to be placed in the ”IP address” field of its entry in the Domain Name System, such that other nodes looking up the mobile node's hostname will find the mobile node's home address.

Finally, a mobile node's home link need not be a physical link composed of some physical medium. Rather, the home link can be a virtual link that exists only as software within the mobile node's home agent. The home agent can be considered to have a virtual interface through which it attaches to this virtual home link. Such a mobile node can never ”connect” to its virtual home link and therefore will never be ”at home”. Instead, the mobile node can physically connect only to foreign links and will always be ”away from home”.

Care-of Address, Foreign Link, and Foreign Agent

A care-of address is an IP address associated with a mobile node that is visiting a foreign link. It as the following properties:

- A care-of address is specific to the foreign link currently being visited by a mobile node.
- A mobile node’s care-of address generally changes every time the mobile node moves from one foreign link to another.
- Packets destined to a care-of address can be delivered using exiting Internet routing mechanisms; i.e., no Mobile IP-specific procedures are needed in order to deliver packets to a care-of address.
- A care-of address is used as the exit-point of a tunnel from the home agent towards the mobile node.
- A care-of address is almost never used as the IP Source or Destination Address in a mobile node’s conversations with other nodes. Specifically, the care-of address will never be returned by a Domain Name Server when another node looks up the mobile node’s hostname.

There are two conceptual types of care-of addresses:

1. A *foreign agent care-of address* is an IP address of a foreign agent which has an interface on the foreign link being visited by a mobile node. A foreign agent care-of address can be any one of the foreign agent’s IP addresses, so long as the foreign agent has at least one interface on the foreign link. Thus, the network-prefix of a foreign agent care-of address need not equal the network-prefix that as been assigned to the foreign link. A foreign agent care-of address can be shared by many mobile nodes simultaneously.

2. A *collocated care-of address* is an IP address temporarily assigned to an interface of the mobile node itself. The network-prefix of a collocated care-of address must equal the network-prefix that has been assigned to the foreign link being visited by a mobile node. This type of care-of address might be used by a mobile node in situations where no foreign agents are available on a foreign link. A collocated care-of address can be used by only one mobile node at a time.

Summarizing, a care-of address is an IP address that is close to a mobile node’s visited, foreign link. By *close* we mean that the care-of address is at most one network ”hop” away from the visited foreign link. The care-of address is either and address of a foreign agent with an interface on this foreign link or an address that as been assigned temporarily to an interface of the mobile node itself. The care-of address is used by the home agent to deliver packets to a mobile node while it is visiting a foreign link; specifically, the care-of address is an exit-point of an IP tunnel from the home agent toward the mobile node.

2.11.2 Mobile IP Operation

1. Home agents and foreign agents advertise their presence on any attached links by periodically multicasting or broadcasting special Mobile IP messages called *Agent Advertisements*.

2. Mobile nodes listen to these *Agent Advertisements* and examine their contents to determine whether they are connected to their home link or a foreign link. *While connected to their respective home links, mobile nodes act just like stationary node, that is, they make use of no other Mobile IP functionality*. The rest of the steps which follow, therefore, assume that a mobile node has discovered that is connected to a foreign link.

3. A mobile node connected to a foreign link acquires a care-of address. A foreign agent care-of address can be read from one of the fields within the foreign agent's *Agent Advertisement*. A collocated care-of address must be acquired by some assignment procedure, such as the *Dynamic Host Configuration Protocol*, the *Point-to-Point Protocol's IP Control Protocol*, or manual configuration.

4. The mobile node *Registers* the care-of address acquired in step 3 with its home agent, using a message-exchange defined by Mobile IP. In the registration procedure, the mobile node asks for service from a foreign agent, if one is present on the link. In order to prevent remote denial-of-service attacks, the registration messages need to be authenticated.

5. The home agent, or some other router on the home link, advertises reachability to the network-prefix of the mobile node's home address, thus attracting packets that are destined to the mobile node's home address. The home agent intercepts these packets, possibly by using proxy ARP[3], and tunnels them to the care-of address that the mobile node registered in step 4.

6. At the care-of address, at either the foreign agent or one of the interfaces of the mobile node itself, the original packet is extracted from the tunnel and then delivered to the mobile node.

7. In the reverse direction, packets sent by the mobile node are routed directly to their destination, without any need for tunneling. The foreign agent serves as a router for all packets generated by a visiting mobile node.

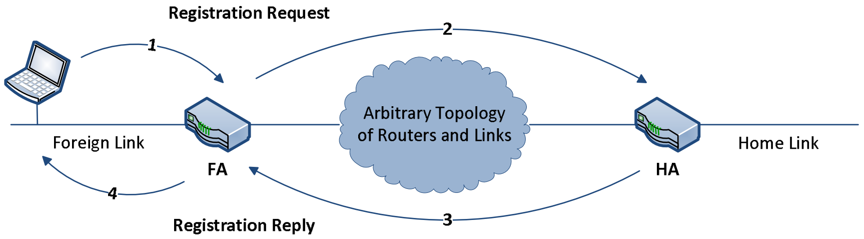


Figure 2.15: Mobile IP registration on a foreign link using a foreign agent's care-of address.

A packet destined to a mobile node's home address cannot be delivered to the mobile node when it is connected to a foreign link, unless all the routers along the path from the home link to the foreign

[3] Sometimes it is useful to have some designated node answer an *ARP Request* on behalf of another node that is currently or permanently unable to send *ARP Replies* itself. An example is when a mobile computer moves to a new link and can no longer receive *ARP Requests* sent by nodes on the previous link. Such an *ARP Reply* is called a *proxy ARP*, and the node sending it is said to be *proxy ARPing* for the other node.

link are given host-specific routes. In the absence of such host-specific routes, some other mechanism is required in order to deliver packets to a mobile node that is visiting a foreign link. Mobile IP uses tunneling as this delivery mechanism.

Thus, Mobile IP requires an address which is ”near” a visiting mobile node which can serve as the exit-point of the tunnel. Also, the address of this exit-point should be reachable via existing routing mechanisms. That is, it must not require any Mobile IP-specific functionality to deliver tunneled packets to the exit-point; otherwise a circular dependency results.

The care-of address provides this tunnel exit-point and must be at most one network ”hop” away from the mobiles node’s foreign link. In the case of a foreign agent care-of address, the care-of address is exactly one hop away; in the case of a collocated care-of address, the care-of address is zero hops away.

Since every tunnel must have an entry-point in addition to an exit-point, a node must be identified which can serve as this entry-point. A router connected to the mobile node’s home link, the home agent, is this tunnel entry-point. In order for the home agent to tunnel packets destined to a mobile node, though, the home agent must be able to intercept such packets. To do this, the home agent or some other router on the home link advertises reachability to the network-prefix of the mobile node’s home address.

With the home agent able to intercept packets destined to the mobile node’s home address, the home agent must also know the exit-point of the tunnel. This is the function performed by Mobile IP’s authenticated Registration procedure, wherein the mobile node informs its home agent of its current care-of address. First the mobile node must determine that it is connected to a foreign link and second it must acquire a care-of address, before it can register this care-of address with its home agent.

Mobile IP Agent Discovery aids mobile nodes in performing these last two functions. It is accomplished by having both home agents and foreign agents periodically multicast or broadcast Mobile IP *Agent Advertisements*, which mobile nodes can receive and inspect to determine their current link and any agents available on that link.

If the mobile node discovers that it is connected to its home link, it behaves just like any fixed node in that it makes use of no additional Mobile IP functionality. In the case of a foreign link, the mobile node reads a care-of address from one of the fields within a foreign agent’s *Agent Advertisement*; registers the care-of address with its home agent; and begins receiving packets from the foreign agent via a tunnel from the home agent to the care-of address.

2.11.3 Mobile IP Registration

A mobile nodes *registers* whenever it detects that its point-of-attachment to the network has changed from one link to another. Also, because these registrations are valid only for a specific *Lifetime*, a mobile node registers when it has not moved but when its existing registration is due to expire. Mobile IP Registration is the process by which a mobile node:

- requests routing services from a foreign agent on a foreign link;
- informs its home agent of its current care-of address;
- renews a registration which is due to expire;
- deregisters when it returns to its home link.

There are also more subtle capabilities of Registration which allow a mobile node to:

- have multiple, simultaneous care-of addresses registered with its home agent, wherein the home agent tunnels a copy of packets destined to the mobile node's home address to each of the multiple care-of address;

- deregister a specific care-of address while retaining others;

- dynamically ascertain the address of a potential home agent, if the mobile node has no prior knowledge of its home agent(s).

2.11.4 Mobile Node Denial-of-Service

One of the primary purposes of Mobile IP Registration is for the mobile node to inform its home agent of its current care-of address, the address to which the home agent will subsequently tunnel all packets destined to the mobile node's home address. Consider what would happen if an attacker were to generate a bogus *Registration Request*, specifying his own IP address as the care-of address for a mobile node (see figure 2.16). Then, all packets sent by correspondent nodes would be tunneled by the mobile node's home agent to the attacker. There are two obvious problems with this:

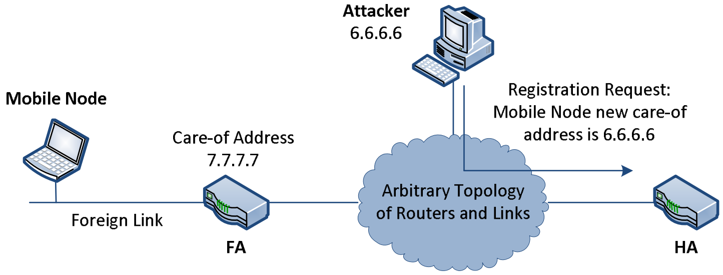


Figure 2.16: Example of an attacker lying about a mobile node's care-of address.

1. The attacker gets to see a copy of every packet destined to the mobile node.

2. The mobile node has been cut off from all receptions, because it cannot receive any packets.

Preventing Denial-of-Service

The solution to this threat is to require cryptographically strong authentication[4] in all registration messages exchanged by a mobile node and its home agent.

Mobile IP allows a mobile node and home agent to use whichever authentication algorithm(s) they choose. However, all implementations must support the default algorithm of ”Keyed MD5”. This authentication method uses the *MD5 Message-Digest Algorithm* [14] to provide secret-key authentication and integrity checking.

Mobile IP authentication using Keyed MD5 works as follows. A mobile node generates a *Registration Request*, filling in all the fields of the request and extension except for the Authenticator field, which is left blank. Then the mobile node computes an MD5 message digest over a sequence of bytes that include:

[4] Strong authentication means that it is next-to-impossible for a attacker to generate a bogus *Registration Request* without the home agents being able to recognize it as a forgery

- the shared, secret key which is known only to the mobile node and its home agent;
- the fixed-length portion of the *Registration Request* message;
- all extensions, up to and including the fields of the Mobile-Home Authentication Extension but not including the Authenticator field itself;
- the shared, secret key again.

$$\text{K-MD5} = MD5\left(\text{Key}, \text{Registration Request Message}, \text{Registration Request Extensions}, \text{Key}\right)$$

The output of the MD5 computation is a 16-byte message digest that the mobile node places within the Authenticator field of the Mobile-Home Authentication Extension. This completes the assembly of the *Registration Request* message, which the mobile node then sends to its home agent.

When the message arrives at the home agent, it performs largely the same processing as the mobile node performed in assembling the message. The home agent computes its own message digest using the shared, secret key and the fields of the received *Registration Request*. It then compares the computed message digest with the one received within the Authenticator field from the mobile node. If they are equal, then the home agent knows that the mobile node indeed sent the *Registration Request* and that the message had not been modified in transit. Thus, the Mobile IP authentication extension provide both authentication and integrity checking.

The exact inverse of the procedures described above happens when the home agent returns a *Registration Reply* to the mobile node. The home agent computes a message digest of the *Registration Reply* and the secret key and includes this message digest within the Authenticator field of the *Registration Reply*. The mobile node verifies the message digest to authenticate the home agent and to check the integrity of the reply.

2.11.5 Mobile IP Replay Attacks

The Mobile-Home Authentication Extension prevents the denial-of-service attack described above. However, it is not enough by itself, because a attacker could obtain a copy of a valid *Registration Request*, store it, and then ”replay” it at a later time, thereby registering a bogus care-of address for the mobile node. To prevent this replay attack from happening, the mobile node generates a unique value for the Identification field in each successive attempted registration. The Identification field is generated in such a way as to allow the home agent to determine what the next value should be. In this way, if the attacker attempts a replay attack, the Identification field in his stored *Registration Request* will be recognized as being out-of-date by the home agent.

Mobile IP specifies two ways in which the Identification field can be chosen on order to prevent these replay attacks. The first uses *timestamps*, wherein the mobile node uses its current estimate for the date-and-time-of-day in the Identification field. If this estimate is not sufficiently close to the home agent’s estimate of the current time, then it rejects the mobile node’s registration and at the same time provides the mobile node with enough information to synchronize its clock to the home agent’s clock.

The other method uses *nonces*. In this method, the mobile node specifies to the home agent the value that the home agent must place in the lower half of the Identification field in the next *Registration Reply* that it sends to the mobile node. Similarly, the home agent specifies to the mobile node the next value it must use in the upper half of the Identification field in its next *Registration Request*. If either node receives a registration message in which the Identification field does not match this next, expected value, then the message is rejected in the case of the home agent or ignored in the case of the mobile node. The rejection mechanism allows the mobile node to synchronize to the home agent in case it has stale information about which value to use in the Identification field and vice versa.

Chapter 3

Related Work

The scope of our work is to achieve fast intra-domain handover by avoiding the full 802.1X authentication each time a handover occurs. There are two main approaches for tackling this problem: (i) proactive security context transfers between APs and (ii) fast recreation of new security contexts in new serving APs; our contribution follows this last one. Some hybrid approaches exist as well.

Using security context transfer between APs for fast handover is appealing because it reduces the reassociation workload required to STAs [15, 16, 17] or makes it completely disappear [18]. Some proposed architectures manage APs as physical terminals of a "central AP" managed by the network, where an STA's security contexts can be migrated to the AP, or APs, closest to the STA [18, 17].

However, context transfers have also several disadvantages. The first disadvantage is that the network of APs must have some management infrastructure for handling the inter-AP secure migration of STA's security contexts: [15] uses a Neighbour Graph of APs and requires the establishment of security associations between arbitrary APs; [16] uses the AS and AP neighbourhood information; [18] uses a centralized Access Controller, which takes full control of handover decisions; [17] uses a centralized CAPWAP architecture [19], which involves extra network entities and requires extra facilities for managing switches' tables. These management infrastructures raise several security issues, which are described in [20].

The second disadvantage is that, according to 802.11i [21], key hierarchies used in each AP for a given STA (starting in PMK) must be different, which means that STAs and APs must nevertheless run a 4WHP after a reassociation [15, 16, 17]. In [16], though, is proposed an alternative approach for postponing the 4WHP, by temporarily reusing a PTK distributed by the previously serving AP.

Running this protocol is also mandatory if RSN IE elements provided by different APs of the same network are different. On the other way, in systems were STAs do not notice when being served by different APs, as in [18], it is very hard, if not impossible, to deploy APs with different operational capabilities (e.g. different data protection cipher suits).

The third disadvantage is that APs must proactively exchange contexts before the actual occurrence of reassociations [15, 16, 17], possibly wasting time and resources for tackling a problem that may never exist. Moreover, all this effort may not be enough; in [15, 16] the STA must run a complete 802.1X authentication whenever associating to an AP outside the Neighbour Graph of the previous AP.

Finally, the fourth disadvantage is that enabling STAs to be transparently served by different APs, as in [18], complicates the management of the access network, namely the management of link layer routing and address translation tables.

When security contexts are recreated in new serving APs, for implementing fast handovers some optimizations must occur to reduce the delay imposed by post-reassociation 802.1X phases.

In [22] is proposed an architecture where an STA may communicate after being reassociated and before running the remaining 802.1X authentication phases (2 and 3). The communication is tunnelled to the previously serving AP through Dynamic Secure Tunnels. These tunnels are created on demand during reassociations, with the help of the AS, and stay afterwards for serving other STAs. However, this approach is likely to complicate APs, since they have to decrypt and validate layer 2 frames coming from the radio link and from security tunnels and, vice-versa, they have to encrypt layer 2 frames to be sent by radio or through a security tunnel. Spoofed reassociation requests may also lead APs to initiate the creation of useless tunnels.

In [23] is proposed an architecture based on a secure 3-party key distribution protocol, using a local HOKEY server [24] besides the usual 802.1X entities, Authenticator an AS. The HOKEY server reduces the duration of the second 802.1X phase, by replacing the full EAP authentication by a local ERP (EAP Reauthentication Protocol [25]) authentication between the STA and the HOKEY server. The HOKEY server uses derived EAP keying material for ERP, uploaded by the local AS, this way eliminating roundtrips to a remote AS. However, we still have the same phases of 802.1X: phase 2, now with ERP, and phase 3 (4WHP).

In [26] is proposed an hybrid approach, where there is some partial context migration between APs and some partial context recreation between the STA and the new serving AP. APs are dynamically organized in clusters using Neighbour Graphs [27], and on each cluster there is a cluster roaming key (CRK) per visiting STA. This CRK is computed from the initial STA PMK and from the (current) list of cluster members. This CRK is used by the STA and serving AP to derive their own, local PMK without message exchanges. Using a per-AP PMK, an STA performs an equivalent 4WHP using only the two 802.11 Reassociation Request and Response frames (which are authenticated). However, using only two messages requires a "self nonce generation": the STA must guess the nonce the AP will use for computing PTK from PMK. This guessing may fail, requiring two more synchronization frames.

This proposal is very similar to ours; however there are some fundamental differences. Similar aspects are the usage of 802.11 protocols for 802.1X-like authentication and the authentication of reassociations. The management of PMKs, however, is completely different: they have to manage clusters of APs and to provide clustering information to STAs, which require it for computing CRKs, while we do not care about AP clustering. Furthermore, they do not provide fast reassociation when the STA moves between APs belonging to different clusters. Finally, a compromised AP in a cluster is able to derive the STA's PMK used by all other APs in the same cluster (this problem also exists in [16], but with lower risk), while in our proposal the PMK used in each AP cannot be used to compute the PMK used in all other APs.

The 802.11r standard initiative [28] aims at reducing the security overhead during AP transitions. The two most obvious security-related contributions are the clarification of opportunistic key caching in APs and the elimination of the 4WHP that in the current 802.1X follows reassociations.

The 802.11r defines a new EMSK-based key hierarchy and the concept of Mobility Domains. A group of APs jointly forms a single Mobility Domain and in doing so gain access to a common, EMSK-based key hierarchy for an STA. Optimistically, an STA assumes that such key hierarchy exists in the target AP when roaming occurs (opportunistic key caching). If it does not exist, then an unexpected latency may occur while the target AP collects this information from its holder. However, the 802.11r does not specifies how or when this key hierarchy is uploaded in advance to the APs of the same Mobility Domain and how APs fetch key hierarchies on demand. All these key management actions are implementation dependent.

Our proposal is inline with 802.11r concerning the elimination of the 4WHP after a reassociation.

But our proposal goes further ahead, specifying how target APs obtain the keying material, thus making or proposal implementation independent. The key material is distributed without requiring architectural changes in the network elements to support inter-AP protocols, either vendor-specific or other standard like CAPWAP [17] from IETF. Furthermore, the APs of the same Mobility Domain obtain new key material, namely PMKs, that are unique for each AP-STA pair and not shared by all APs of the same Mobility Domain, as in 802.11r. This way, the compromise of an AP of a Mobility Domain does not compromise the security of past and future communications of STAs with other APs of the same Mobility Domain. Finally, we provide authentication for reassociation protocols, while 802.11r does not.

The papers discussed in this chapter did not proposed a layer 3 solution, being that the presented handover solutions would only work inside the ESS. Proactive security context transfers between APs, and neighbor graph management leads to a layer 3 solution almost impossible or at least very complex.

Chapter 4

Contribution - Layer 2 Fast, Secure Handover

Our goal is to perform fast 802.1X reauthentications to minimize reassociation handovers, while not using the 802.1X approach for implementing them. In other words, in reauthentications we want to achieve exactly the same results of 802.1X, described in Section 2.5, but with the shortest possible set of frame exchanges after initiating a reassociation with another AP.

Our proposal for implementing fast and secure handovers is to use two existing 802.11 protocols, authentication and reassociation, with extensions capable of providing all the functionalities of an 802.1X reauthentication. By doing so, we are able to completely skip phases 2 and 3 of 802.1X, using only its first phase for reauthentication, key distribution and reassociation (c.f. Fig. 2.14) .

We assume that 802.1X reauthentications can only occur after an ordinary 802.1X authentication. This previous authentication is responsible for producing and distributing secret material that will be used to carry on one or more reauthentications. This approach is inline with the fast reauthentication goals presented in [29].

According to Section 2.5, we have the following requirements for the new 802.11 authentication and reassociation protocols:

1. Install a fresh, secret PMK in both STA and AP;
2. Use PMK and two nonces to produce a new, fresh session key PTK;
3. Confirm a common knowledge of PTK;
4. Exchange authenticated RSN IE capabilities;
5. Send a confidential GTK from the AP to the STA.

We handle the two first requirements in a new 802.11 authentication protocol and the two last requirements in a new 802.11 reassociation protocol (see Figure 4.2). The third requirement, mutual knowledge of PTK, will happen along both protocols.

Just to clarify the reader, we will use a modified 802.11 authentication protocol to implement the new, fast 802.1X reauthentication protocol. This is a completely new approach, since currently 802.11 and 802.1X authentications are totally independent from each other. Therefore, in the text we may use either the expressions "*802.1X reauthentication*" or "*new 802.11 authentication*" to refer to this protocol, depending on the context.

4.1 Reauthentication Service

The Reauthentication Service (RS) is a new service that we use for handling fast reauthentications. Following the terminology of 802.11r, the RS is the enabler of a Mobility Domain, which is formed by all APs that know how to reach and securely interact with the RS for handling STA's reauthentication requests.

The RS replaces the AS during reauthentications. Therefore, it can be implemented in two different ways: (i) as part of the local AS; or (ii) as an independent, HOKEY server [1]. In any case, we consider that for a given domain served by one AS there is exactly one RS.

For performing reauthentications, the RS receives secret material from the local AS. We assume that the RS is able to authenticate AS messages and these can only be understood (decrypted) by the RS. Note that if RS is part of the local AS, and not a separate service/host, these assumptions are automatically true.

The APs use the RS instead of the AS for handling STA´s reauthentication requests. We assume that there are security associations between all APs and the RS, similar to the ones that exist between APs and the local AS. These security associations are fundamental to (i) authenticate RS messages to APs and to (ii) enforce the confidentiality of keys provided by the RS to the APs. Using the terminology of 802.11r, the Mobility Domain of an STA is the set of APs that have a security association with the local RS. Again, if the RS is part of the local AS, and not a separate service/host, the 802.1X architecture guaranties these assumptions, because it uses a security association between each AP and the local AS.

4.2 Initial 802.1X authentication

As we previous stated, for the new reauthentication protocol we assume that some secret material was produced on behalf of a previous, ordinary 802.1X authentication. Such material is a key and a related unique identifier: **Reauthentication Key** (**RK**) and **STA Digital Pseudonym** (**SDP**).

These two values are computed by both the Supplicant and the AS during an ordinary 802.1X authentication and uploaded by the latter to the RS (see Figure 4.1). The RK will be used to generate a fresh PMK for each reauthentication request tagged with SDP.

After an 802.1X authentication, a Supplicant shares a secret EMSK with the AS that authenticated it. We will use this key to derive the values of RK and SDP as follows:

$$\begin{aligned}\text{RK} &= \text{PRF-X}\left(\text{EMSK}, \text{“802.11 authentication”}\right)\\ \text{SDP} &= \text{PRF-X}\left(\text{RK}, \text{ID}\right)\end{aligned}$$

where the PRF-X(Y) represents the first X bits computed over Y by a pseudo-random function for key expansion defined in [30] and where ID is the identification of the Supplicant provided to the AS during its authentication. According with [2], RK is a Domain Specific Root Key and SDP is a Domain Specific Usage Specific Root Key.

We use the SDP for uniquely identifying an authenticated session of an STA instead of its MAC address. The reason for doing so is that an SDP cannot be spoofed by an attacker, as it derives from EMSK, while a MAC address can be spoofed. This way, an attacker cannot use MAC spoofing attacks to install in the RS a new RK for a victim STA.

Given the security assumptions of Section 4.1, the upload of SDP and RK from the AS to the RS is as protected, in terms of secrecy and integrity control, as the upload of MSK from the same AS to an AP (Authenticator).

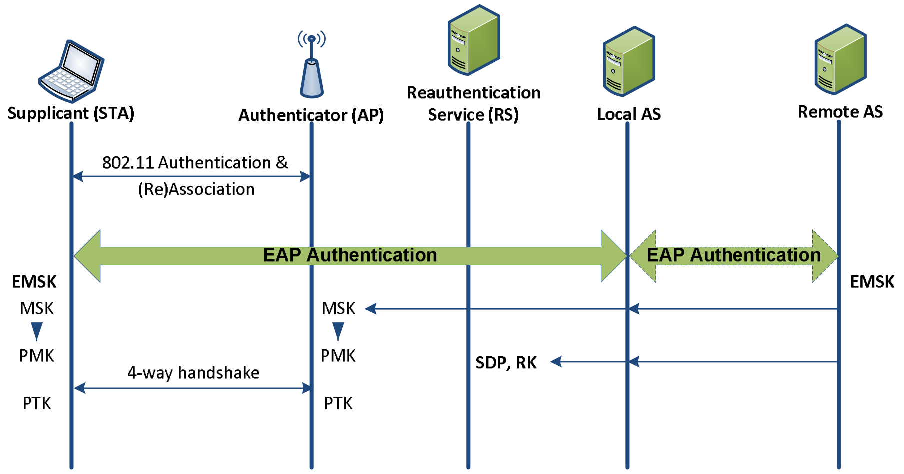


Figure 4.1: Reauthentication Service: integration with the 802.1X architecture and secret material uploaded to it (SDP and RK) by the local AS upon a successful EAP-based protocol executed on behalf of an 802.1X authentication.

STA→AP	Auth. Req., SDP, $\{K\}_{RK}$, N1, MIC (K)
AP→RS	SDP, $\{K\}_{RK}$, N1, MIC (K)
AP←RS	SDP, N3, PMK
STA←AP	Auth. Resp., N2, N3, ΔT, MIC (KCK)

Table 4.1: New 802.1X reauthentication message exchange

4.3 New 802.1X reauthentication protocol

The new 802.1X reauthentication protocol uses the basic structure of 802.11 authentication protocols, while carrying extra data in frames' payload. We use only two 802.11 frames, one Authentication Request and one Authentication Response, for performing an 802.1X reauthentication on the wireless medium, though the complete protocol involves two extra messages between the AP and the RS (see Figure 4.2):

where $\text{PMK} = \textit{hash}\,(\text{K}, \text{N3})$ and KCK is computed from PMK, N1 and N2 as in 802.1X (cf. Section 2.5).

First, the STA generates a nonce N1 and a random key K. Then it sends an 802.11 Authentication Request to the AP, containing its SDP, K encrypted with RK, N1 and a MIC computed with K. The AP forwards all these values to the RS, which uses SDP to find the STA's RK. Once knowing RK, it extracts K, validates the MIC and, if valid, generates a nonce N3, hashes it with K for producing PMK and sends the PMK to the AP, together with N3. Along these steps, the STA is identified with its pseudonym, SDP. The final 802.11 Authenticate Response message contains two nonces, N2 and N3, the first generated by the AP and the second by the RS, and a MIC computed with KCK. The STA uses N3 to compute PMK, and this key, together with N1 and N2, to compute the shared PTK (and its components KCK, KEK and TK) as in 802.1X. The KCK component of PTK is then used to authenticate the received message.

In the described reauthentication protocol, 802.11 Authentication Requests can be replayed. This fact enables attackers to install new, fresh PMK and PTK keys in APs on behalf of spoofed STAs,

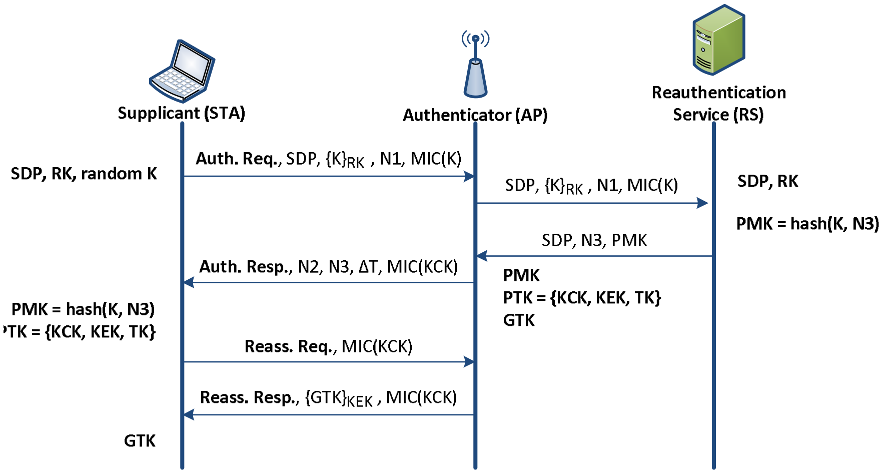


Figure 4.2: New 802.1X reauthentication and 802.11 reassociation protocols and interoperation with the Reauthentication Service (RS).

creating two different DoS problems: (i) APs may get flooded by old, useless security associations for STAs that may no longer be around and (ii) attackers may interfere with security associations that are currently being used by STAs, by installing new keys. Fortunately, both problems can be solved very easily by using a monotonic, sequence counter for producing the nonce N1. With this strategy, the RS can detect and refuse replays of reauthentication requests for a given SDP; all it has to do is to keep and check the previous, valid value of N1 used in the last successful reauthentication request of that SDP. This way, N1 is used both for detecting replays of reauthentication protocol runs and for producing new, fresh PTK keys.

At the end of this reauthentication protocol, the STA believes that the AP knows the new keys, PMK and PTK, because the 802.11 Authentication Response is authenticated with KCK, a part of PTK. Therefore, STA can be sure the AP is genuine (belongs to the target network), otherwise it could not have received the key PMK from the RS. On the contrary, the AP is not yet convinced that the STA knows PMK and PTK. This proof will be provided latter, during an association or reassociation. Nevertheless, the AP may consider the STA authenticated, because of the positive reply from the RS. Regardless of the freshness of K and N1 provided by the STA, the freshness of PTK is ensured by N3, provided by RS.

The ΔT field of the 802.11 Authentication Response indicates the amount of time the AP will keep the security context (authenticated state and secret keys PMK and PTK) negotiated with the STA. If the STA does not get associated to the AP before ΔT, the AP removes the security context without any warning. This simple mechanism prevents APs from getting flooded with security contexts, while giving STAs some feedback about the lifetime of their security contexts' caching in APs. Furthermore, APs are free to manage independently their security contexts, namely they can use different caching periods.

To reduce the outage while roaming, a mobile STA should use this reauthentication protocol with all neighbour APs as soon as they become suitable targets for roaming. This can take place during AP scanning/probing phases, when the STA looks for APs more interesting that the one actually serving it [31, 32, 33]. Note, however, that while scanning/probing phases may occur frequently to better decide when to roam to another AP, authentication with each neighbour AP needs to be performed once for some time, which depends on the AP security contexts' caching period (conveyed to the STA

STA→AP	Reass. Req., MIC(KCK)
STA←AP	Reass. Resp., $\{\text{GTK}\}_{\text{KEK}}$, MIC(KCK)

Table 4.2: New 802.11 reassociation message exchange

by the ΔT field in the 802.11 Authentication Response).

4.4 New 802.11 reassociation protocol

The new 802.11 reassociation protocol has two differences regarding the current one: (i) authenticates exchanged RSN IE values and (ii) sends a group key GTK from the AP to the STA. This is achieved with some extra fields in the 802.11 Reassociation Request and Response frames (see Figure 4.2):

The MIC values in both request and response authenticate the RSN IE values exchanged in the payload of the frames. Furthermore, the MIC in the 802.11 Reassociation Request proves to the AP that the STA effectively knows PMK and PTK.

These modifications could be applied as well to the 802.11 association protocol. The only difference between them is that a Reassociation Request frame contains an extra field with the MAC address of the AP formerly serving the STA.

4.5 New AP 802.11/802.1X state diagram

The new 802.1X reauthentication and 802.11 reassociation protocols need to be handled in the context of a new state machine within an AP. Figure 4.3 shows two state diagrams, one for the current 802.11/802.1X authentication and association (left) and another for our new protocols (right).

In the state diagram on the right, we can see that after a reauthentication with the new 802.11 authentication protocol, both AP and STA are already authenticated in terms of 802.11 and 802.1X. Then, after the new 802.11 reassociation protocol, the STA becomes associated to the AP and the 802.1X port in the AP can immediately be open, because 802.1X authentication is already completed. This fact enables both AP and STA to start exchanging data frames without further delays[1]. Therefore, after the reassociation, no security-related delays contribute to handover outage delays.

This new state diagram differs slightly from the one proposed by B. Aboba in [9] (see Figure 4.4), where the PMK is installed when the STA becomes authenticated and the PTK is installed when the STA becomes associated. In fact, there is no problem in anticipating the installation of PTK and there are even some benefices, such as allowing the authentication of management frames, namely the authentication of reassociation protocol runs with KCK.

[1] Nevertheless, the AP may not immediately start sending data frames because of delays in the update of switching tables.

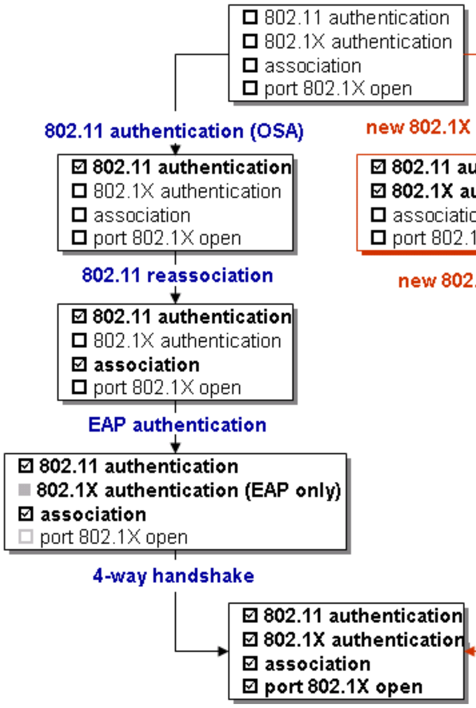


Figure 4.3: AP state diagram concerning 802.11 and 802.1X authentication, 802.11 association and 802.1X port state. On the left side, we have the state diagram of an actual 802.11/802.1X authentication and association. On the right side, we have the state diagram for handling our new 802.1X fast reauthentication and 802.11 reassociation protocols.

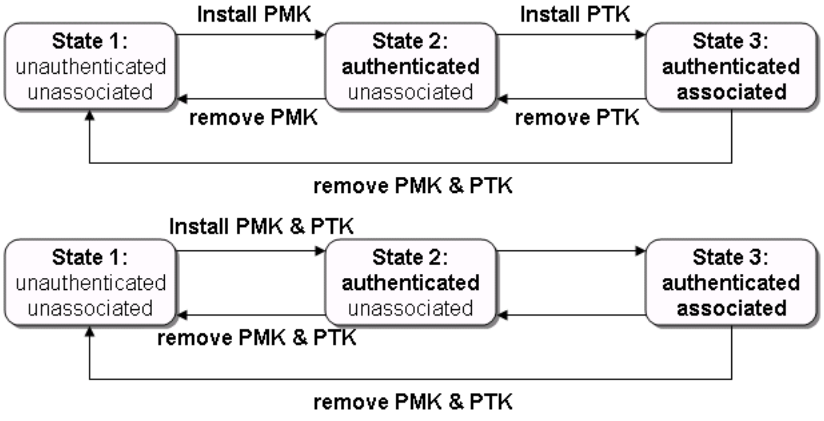

Install PMK
Install PTK
State 1: unauthenticated unassociated
State 2: authenticated unassociated
State 3: authenticated associated
remove PMK
remove PTK
remove PMK & PTK
Install PMK & PTK
State 1: unauthenticated unassociated
State 2: authenticated unassociated
State 3: authenticated associated
remove PMK & PTK
remove PMK & PTK

Figure 4.4: State diagram proposed by B. Aboba (top) and our state diagram (bottom).

Chapter 5

Contribution - Layer 3, Fast, Secure Handover

Our goal was to integrate Mobile IP with our previous layer 2 fast handover protocol so that we can extend secure, fast handovers to layer 3, when a STA performs handovers between ESS.

Our design goals were the following:

1. Reduce as much as possible the latency of layer 3 handovers;
2. Use Mobile IP without adding more complexity to an already complex protocol;
3. Define and distribute keys to authenticate messages exchanged during the Mobile IP registration process;
4. Inform the STA in advance, about which FA/HA is serving the current ESS, so that the STA can determine if it is about to roam to a different subnet, and if so, what is the IP of the FA/HA of that subnet;
5. Make this layer 3 improvement as independent as possible of our previous layer 2 fast handover protocol.

The RS is the most important entity of our architecture and it as the responsibility to derive and distribute any keys that are needed. Network administrators can take advantage of it if they want, for instance, to create secure tunnels between HA and FA using IPsec. The RS could supply the necessary keying material for this procedure.

Our proposal for integrating this secure layer 3 improvement with our protocol is to use the RS to supply the HA with the keys needed to authenticate the messages from an STA an vice-versa. For this, we assume there are security associations between the RS and all the HA/FA.

To authenticate mobility-related messages, a new key, Mobile Key (MK), was added to our key hierarchy. This key derives from the previous RK and SDP derived during the initial 802.1X authentication (see 4.2) as follows:

$$\mathrm{MK(i)} \quad = \quad \mathrm{PRF\text{-}X}\left(\mathrm{RK}, \mathrm{SDP}, \mathrm{i}\right)$$

We use the key MK to authenticate the *Registration Request* and *Registration Reply* messages. The *i* field used to compute MK(i) is a counter maintained by the RS and the STA so that a fresh key is derived every time a new registration occurs. No change is made to the *Registration Request* and

Registration Reply messages. We only make use of the extension provided, namely, the Mobile-Home Authentication Extension [4]. This takes care of item 2 and 3 of our design goals.

To provide the STA with the information it needs for knowing when it is about to roam to a different subnet, we have several options: (i) add an extra field to a *Beacon* and *Probe Response* with the IP address and network-prefix of the HA/FA of the ESS that the target AP serves (ii) add the previous information in the *Authentication Response* message (iii) downloadable network configuration. With this information, during the initial authentication the STA will learn the IP address of its HA when it receives a *Beacon*, a *Probe Response* or a *Authentication Response* message from a target AP e will then compare the new HA/FA IP address with the IP address of its HA. The result of this comparison will tell the STA: (i) if the target AP is serving the same ESS or (ii) if the AP is serving a different ESS. If the second case occurs the STA will know the IP address of the FA of the ESS served by that target AP. This information will then be used by the STA if later he decides to connect to that AP. There is still another situation that could happen. If the AP does not advertises the HA/FA of its subnet, the STA can assume that there is no FA on that subnet. In this case, the STA will need to acquire a co-located care-of address when he decides to roam. This takes care of item 3 of our design goals.

Goal item 1 is the most critical, as network configuration truly adds to the handover latency (see table 2.1). Our proposal is to run the care-of address registration procedure before reassociation, thus eliminating completely the network configuration latency of our fast handover protocol, as seen in 1.2. This can be done using one special registration capabilities: have multiple, simultaneous care-of addresses registered with the home agent, wherein the home agent tunnels a copy of packets destined to the mobile node's home address to each of the multiple care-of address. This way, we can, for a short period of time have the packets sent to the previous care-of address and the new care-of address, for the duration of the reassociation. After the reassociation the STA can deregister its old care-of address.

As seen in this description, no changes needed to be made to our previous layer 2 fast handover protocol, making it independent of the layer 3 handover. Thus it can still work in a scenario where Mobile IP is not present.

5.1 Protocol Description

This protocol description assumes that there are security associations between the RS and all the HA/FA. Also the STA knows, before the handover decision: (i) if its going to roam to a different subnet, (ii) if the new subnet as a FA, (iii) the IP address of the FA.

Assuming that the STA is going to roam to a different subnet, there are two possible scenarios: (i) the target subnet as a FA and the STA as to obtain a care-of address, or (ii) there is no FA and the STA has to obtain a co-located care-of address.

Figure 5.1 shows the message exchange that happens when the STA roams to a subnet where a FA exists. The protocol works as follows:

1. After the preauthentication the STA knows that the target AP serves a different subnet, and also knows the IP address of the FA.

2. When the STA decides to roam to that subnet it first sends an *Agent Solicitation* message. This solicitation differs from an normal *Agent Solicitation*, because in a normal solicitation the Destination Address is a broadcast address. This can not be done in our protocol because we want this message to be sent before the association with the new AP, and, at this point, the STA

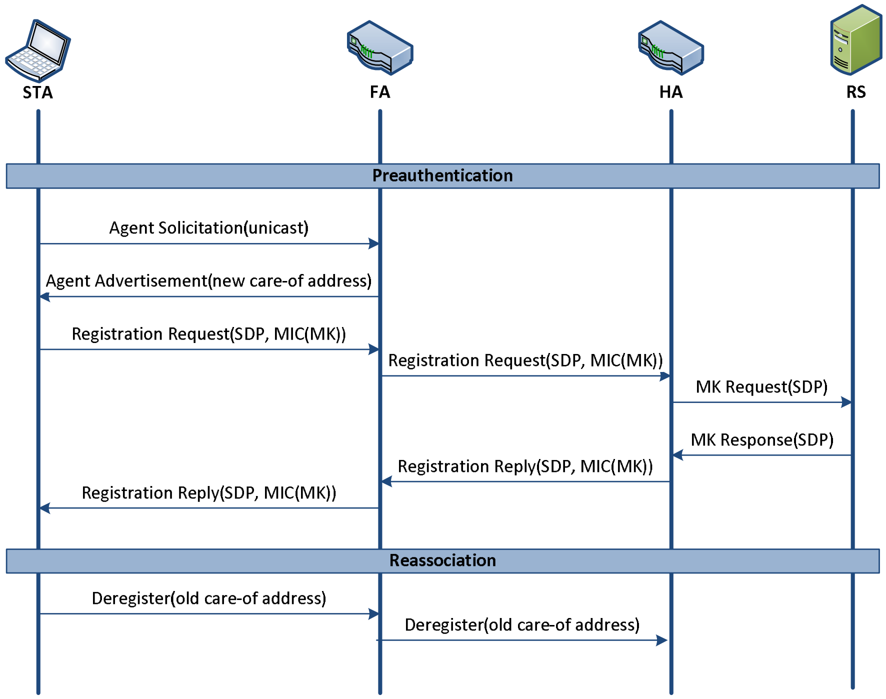

STA
FA
HA
RS
Preauthentication
Agent Solicitation(unicast)
Agent Advertisement(new care-of address)
Registration Request(SDP, MIC(MK))
Registration Request(SDP, MIC(MK))
MK Request(SDP)
MK Response(SDP)
Registration Reply(SDP, MIC(MK))
Registration Reply(SDP, MIC(MK))
Reassociation
Deregister(old care-of address)
Deregister(old care-of address)

Figure 5.1: Protocol description of an STA roaming to a subnet with a FA.

does not belong to the same subnet as the target AP. So the STA will fill the Destination Address of the *Agent Solicitation* message with the IP address of FA that it learned before.

3. Upon receiving an *Agent Solicitation* message from the STA, the FA will send a *Agent Advertisement*, and once more with a slight difference from a normal *Agent Advertisement* message, that is, the Destination Address field of this message will contain the STA home address.

4. After receiving the *Agent Advertisement* message, the STA will start the registration process with it's home agent. This process is a normal Mobile IP registration process were we use the extensions to provide to the HA the SDP. So, the STA sends a *Registration Request* message to its HA with its SDP and the Mobile-Home Authentication Extension calculated with the key MK.

5. The FA processes the *Registration Request* message, and, if correct, it relays the message to the HA.

6. Upon receiving the *Registration Request* the HA asks the RS for the MK associated to that SDP. After receiving the MK, the HA authenticates and processes the message. If correct, it sends a *Registration Reply* authenticated with MK.

7. The FA processes the *Registration Reply* and, if correct, it relays the message to the STA. The STA will then authenticate the message with MK. At this point the HA tunnels the packets to both the old and the new care-of address of the STA.

8. After the registration process as been completed the STA reassociates with the target AP and deregisters its old care-of address.

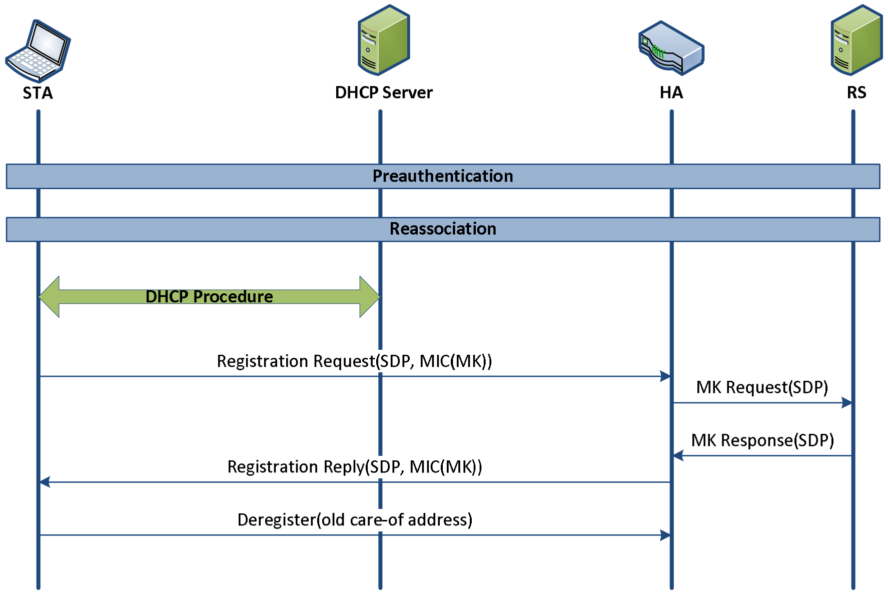

STA
DHCP Server
HA
RS
Preauthentication
Reassociation
DHCP Procedure
Registration Request(SDP, MIC(MK))
MK Request(SDP)
MK Response(SDP)
Registration Reply(SDP, MIC(MK))
Deregister(old care-of address)

Figure 5.2: Protocol description of an STA roaming to a subnet without a FA.

Figure 5.2 shows the message exchange when the STA roams to a subnet where no FA exists. In this case the STA as to run the DHCP protocol to obtain a co-located care-of address, because there is no FA present in the target subnet. The DHCP, being a layer 2 protocol, as to be done after the reassociation, because the STA as no way of running a layer 2 protocol while being associated to a different subnet.

When no FA is available in the target subnet the latency of handover will increase substantially, because of the four messages exchanged when running the DHCP protocol. In this case there is no improvement in the layer 3 handover latency, however the overall handover latency is still improved because of the fast reassociation of our protocol.

Chapter 6

Evaluation

In this chapter we evaluate both the functionalities introduced by our contributions and their security.

6.1 Evaluation of Layer 2 Contributions

The list below presents the most common problems and open issues that exist today regarding mobility using the 802.11i architecture:

1. There are no countermeasures against DoS attacks using spoofed management frames (Association, Reassociation, Disassociation, Deauthentication, etc.);
2. No standardization of information exchange over the 802.11i pre-authentication channels;
3. No standard solution for distributing a PMK of an STA to multiple APs;
4. No standard solution for replay attacks due to the reuse of the same PMK across two associations;
5. An STA cannot be informed of GTK updates that occur between a pre-authentication and an effective association.

Concerning item 1, our handover protocol takes a small step forward, since reassociation requests are authenticated. This authentication prevents attackers from creating DoS scenarios with spoofed reassociation frames.

This authentication issue started to be addressed by IEEE due to the development of 802.11k, that defines and exposes information to facilitate the management and maintenance of a WLAN. The problem is that 802.11k messages require protection prior to the STA transition from one AP to another. However, this is difficult because 802.11i keys are not available until after reassociation. But with our 802.1X reauthentication protocol we are able to distribute PMK keys to specific APs, therefore taking a first step towards the required key distribution for authenticating 802.11k messages.

Concerning item 2, we do not use the pre-authentication approach defined by 802.11i, through the currently associated AP. Instead, we recovered the original 802.11 approach: we authenticate first in all APs that are candidate for roaming using the wireless medium and we only negotiate a new PTK when we effectively reassociate to an AP.

The 802.11i assumptions for RSN environments state that the STA and the AP generate a different, fresh PTK for each session between the pair [21, §8.1.4 g)]. Since the exact meaning of the term *session* is not clarified in the standard, we assumed that a session may span several associations with the same AP[1]. Under this assumption, we provide a different, fresh PMK for each session and a fresh, derived PTK. The freshness of PMK is ensured by the STA and by the RS, with K and N3, respectively; the freshness of PTK is ensured by the freshness of PMK and the two nonces, N1 and N2, produced by the STA and AP, respectively.

Thus, concerning items 3 and 4, we are able to distribute different PMK keys to different APs; each key is distributed by an 802.1X reauthentication. Therefore, we not only define a key distribution architecture for PMK keys, but we also prevent the reuse of the same PMK across different sessions (associations).

A TGi fast roaming goal, also stated in [20], is that compromise of one AP must not compromise past or future key material. Our proposal goes towards meeting this goal, as the PMK in each AP cannot be used to compute the PMKs used in other APs (assuming that hashing functions used to compute PMK are one-way). For fully meeting this goal, we also have to ensure that secret associations between APs and RS are independent, in the sense that compromise of one AP does not help to compromise security associations established by other APs with the RS.

Concerning item 5, the distribution of GTK keys is a long-standing problem of 802.11i pre-authentications. In fact, running a 802.11i pre-authentication protocol between an STA and an AP distributes a GTK to the STA, but the key may change until the effective association of the STA to AP. We solve this problem in a very simple way, as GTK keys are distributed only when reassociations happen.

To conclude, we present in the list below some of the goals achieved with our protocol, some of them still been discussed by the IEEE 802.11 committees.

Summing up, we reused the available 802.11i keys framework and, with a minimum amount of invention, we were able to make 802.11i keys available before (re)associations, but only to the APs of interest for an STA. Such keys cannot be used across associations and GTK keys are distributed in the right occasion, i.e., when reassociations take place. Resistance against DoS attempts was increased by authenticating reassociation requests.

Concluding, we successfully reduced the roaming latency due to security requirements and we solved many of the 802.11i problems related with roaming, most of them still been debated by IEEE standardization committees.

6.1.1 Practical Considerations

Unlike solutions that migrate security contexts, we negotiate a security context with each candidate AP. Therefore, we are able to tolerate differences in security capabilities of APs, as long as they implement our two new reauthentication and reassociation protocols. Natural differences between APs are the set of supported cipher suites. For example, an STA may which to use AES CCMP whenever possible and TKIP otherwise, and not all APs may support AES CCMP. In this case, our solution gracefully adapts to the resources provided by each STA.

Our reauthentication protocol follows the recommendations of [29] concerning backward compatibility and impact to existing deployments. In terms of backward compatibility, the protocol allows (i) a peer not supporting fast reauthentication to still function in a network supporting fast reauthentication, and allows (ii) a peer supporting fast reauthentication to still function in a network not supporting fast authentication. Note that an STA can detect if a network or STA does not support fast reauthentication by using Information Elements in 802.11 Beacon or Probe Response frames. In terms of

[1] Otherwise, they would have used the term *association* instead of *session*.

impact in existing deployments, the reauthentication protocol makes changes in STAs, authenticators (APs), and AS to meet the fast, secure roaming goal, but the changes are reduced and do not sacrifice performance.

Note, however, that a network supporting both reassociation protocols, the current one (unauthenticated) and the new (authenticated) reassociation protocol, is not able to avoid DoS attacks using the current protocol. This happens because an AP is not able to confirm if the STA issuing the old reassociation request is able or not to use the new protocol, therefore it must default to authorize the request.

6.1.2 Security Evaluation with Avispa

We used the Avispa tool [34] to analyse our new protocol. Namely, we evaluated the reachability of the following goals by legitimate players: secrecy of K and GTK and authentication of received N2, N3 and RSN IE values. We found no problems in reaching these goals and we did not find attacks against the protocol.

Avispa Code:

```
%
% S - STA
% A - AP
% R - RS
%
% S -> A : SDP, N1, {K}_Krk, MIC(K)
% A -> R : SDP, N1, {K}_Krk, MIC(K)
% R -> A : N3, PMK
% A -> S : N2, N3, MIC(KCK)

role sta(STA, AP, RS : agent,
 Hash: hash_func,
 SDP : message,
 Krk : symmetric_key,
 SND, RCV : channel (dy))
played_by STA def=

  local State: nat,
   K, GTK : message,
   PMK : symmetric_key,
KEK : message,
KCK : message,
MIC : message,
N1, N2, N3 : text,
RSNIE_sta, RSNIE_ap : message
  init
    State := 0



  transition

  1. State = 0 /\ RCV(start) =|>
```

```
     State’ := 2 /\ K’ := new()
       /\ N1’ := new()
 /\ SND(SDP.N1’.{K’}_Krk.Hash(SDP.N1’.K’))

  2. State = 2 /\ RCV(N2’.N3’.Hash(N2’.N3’.Hash(2.N1.N2’.Hash(N3’.K)))) =|>
     State’ := 4 /\ PMK’ := Hash(N3’.K)
                 /\ KEK’ := Hash(1.N1.N2’.PMK’)
                 /\ KCK’ := Hash(2.N1.N2’.PMK’)
       /\ RSNIE_sta’ := new()
 /\ SND(RSNIE_sta’.Hash(RSNIE_sta’.Hash(2.N1.N2’.Hash(N3’.K))))
 /\ witness (STA,AP,sta_ap_rsnie,RSNIE_sta’)
 /\ secret(KCK’,kck1,{STA,AP,RS})
 /\ secret(KEK’,kek1,{STA,AP,RS})

  3. State = 4 /\ RCV(RSNIE_ap’.{GTK’}_KEK.Hash(RSNIE_ap’.GTK’.Hash(2.N1.N2.Hash(N3.K)))) =|>
     State’ := 6 /\ request(STA,AP,ap_sta_rsnie,RSNIE_ap’)
 /\ secret(K’,k,{STA,RS})

end role

role ap (STA, AP, RS : agent,
Hash : hash_func,
Kar : symmetric_key,
SND_STA, RCV_STA, SND_RS, RCV_RS : channel (dy))
played_by AP def=

  local State: nat,
SDP : message,
RSNIE_sta, RSNIE_ap : message,
GTK : message,
PMK : message,
KEK : message,
KCK : message,
MIC : message,
N1, N2, N3 : text,
EK : {message}_symmetric_key
  init
    State := 1

  transition

  1. State = 1 /\ RCV_STA(SDP’.N1’.EK’.MIC’) =|>
     State’ := 3 /\ SND_RS(SDP’.N1’.EK’.MIC’.Hash(SDP’.N1’.MIC’.Kar))

  2. State = 3 /\ RCV_RS(SDP.N3’.{PMK’}_Kar.Hash(SDP.N3’.PMK’.Kar)) =|>
     State’ := 5 /\ N2’ := new()
            /\ KEK’ := Hash(1.N1.N2’.PMK’)
            /\ KCK’ := Hash(2.N1.N2’.PMK’)
       /\ SND_STA(N2’.N3’.Hash(N2’.N3’.Hash(2.N1.N2’.PMK’)))
 /\ secret(KCK’,kck2,{STA,AP,RS})
 /\ secret(KEK’,kek2,{STA,AP,RS})

  3. State = 5 /\ RCV_STA(RSNIE_sta’.Hash(RSNIE_sta’.KCK)) =|>
```

```
     State’ := 1 /\ RSNIE_ap’ := new()
       /\ GTK’ := new()
       /\ SND_STA(RSNIE_ap’.{GTK’}_KEK.Hash(RSNIE_ap’.GTK’.KCK))
       /\ request(AP,STA,sta_ap_rsnie,RSNIE_sta’)
       /\ witness(AP,STA,ap_sta_rsnie,RSNIE_ap’)

end role

role rs (STA, AP, RS : agent,
 Hash: hash_func,
 SDP : message,
 Krk, Kar : symmetric_key,
 SND, RCV : channel (dy))
played_by RS def=

  local State : nat,
   K : message,
PMK : message,
   N1, N3 : text

  init
    State := 0

  transition

  1. State = 0 /\ RCV(SDP.N1’.{K’}_Krk.Hash(SDP.N1’.K’).Hash(SDP.N1’.Hash(SDP.N1’.K’).Kar)) =|>
     N3’ := new() /\ PMK’ := Hash(N3’.K’)
         /\ SND(SDP.N3’.{PMK’}_Kar.Hash(SDP.N3’.PMK’.Kar))

end role

role session(S,A,R : agent,
     SDP : message,
     Krk, Kar : symmetric_key,
     Hash: hash_func )
def=

  local S_STA_AP, R_STA_AP, S_AP_RS, R_AP_RS, S_RS_AP, R_RS_AP, S_AP_STA, R_AP_STA: channel(dy)

composition
    sta (S, A, R, Hash, SDP, Krk, S_STA_AP, R_AP_STA)
/\  ap  (S, A, R, Hash, Kar, S_AP_STA, R_STA_AP, S_AP_RS, R_RS_AP)
/\  rs  (S, A, R, Hash, SDP, Krk, Kar, S_RS_AP, R_AP_RS)

end role

role environment()
def=

  const s, a, r : agent,
        kun, ksr, kar, kir, kis : symmetric_key,
sdp : message,
h : hash_func,
k, n1, n2, n3, pmk, kek1, kek2, kck1, kck2, gtk : protocol_id,
sta_ap_rsnie, ap_sta_rsnie : protocol_id

  intruder_knowledge = {s,a,r,kis,kir,h}

  composition
       session(s,a,r,sdp,ksr,kar,h)
    /\ session(i,a,r,sdp,kun,kar,h)
    /\ session(s,i,r,sdp,ksr,kun,h)

end role

goal
    secrecy_of k
    secrecy_of pmk
    secrecy_of kek1, kek2
    secrecy_of kck1, kck2
    secrecy_of gtk

    authentication_on sta_ap_rsnie
    authentication_on ap_sta_rsnie
end goal

environment ()
```

6.2 Evaluation of Layer 3 Contributions

One of the security problems of Mobile IP is vulnerability to DoS attacks due to spoofed registration requests. To prevent this attacks, we need to enforce mutual authentication between an STA and its HA, and between an HA and an FA.

Mutual authentication between the STA and the HA is achieved at the cost of of one more key associated to the SDP, stored in the RS, that the HA will ask when a *Registration Request* message arrives. This key is used to authenticate both *Registration Requests* and *Registration Responses*, this way ensuring the mutual authentication between an STA and its HA.

Concerning the mutual authentication between Mobile IP agents (HA and FA), we considered that this issue is solved by the network provider. For our security requirements all that matters is that there is a security association between each Mobile IP agent and the RS, to ensure a proper distribution of authentication keys, identified by and SDP, to HAs.

By default, an STA trusts the information provided by a genuine AP. The question now is to distinguish a genuine from a rogue AP. Concerning the distribution of trusted Mobile IP advertising information (network mask, FA IP address, etc.), this can be achieve along authenticated elements of our 802.1X reauthentication protocol.

Chapter 7

Practical considerations

A long-standing problem of 802.11i pre-authentications is the distribution of GTK keys. In fact, running a 802.11i pre-authentication protocol between an STA and an AP distributes a GTK to the STA, but the key may change until the effective association of the STA to AP. We solve this problem in a very simple way, as GTK keys are distributed only when reassociations happen.

Unlike solutions that migrate security contexts, we negotiate a security context with each candidate AP. Therefore, we are able to tolerate differences in security capabilities of APs, as long as they implement our two new reauthentication and reassociation protocols. Natural differences between APs are the set of supported cipher suites. For example, an STA may which to use AES CCMP whenever possible and TKIP otherwise, and not all APs may support AES CCMP. In this case, our solution gracefully adapts to the resources provided by each STA.

Our reauthentication protocol follows the recommendations of [29] concerning backward compatibility and impact to existing deployments. In terms of backward compatibility, the protocol allows (i) a peer not supporting fast reauthentication to still function in a network supporting fast reauthentication, and allows (ii) a peer supporting fast reauthentication to still function in a network not supporting fast authentication. Note that an STA can detect if a network or STA does not support fast reauthentication by using Information Elements in 802.11 Beacon or Probe Response frames. In terms of impact in existing deployments, the reauthentication protocol makes changes in STAs, authenticators (APs), and AS to meet the fast, secure roaming goal, but the changes are reduced and do not sacrifice performance.

Note, however, that a network supporting both reassociation protocols, the current one (unauthenticated) and the new (authenticated) reassociation protocol, is not able to avoid DoS attacks using the current protocol. This happens because an AP is not able to confirm if the STA issuing the old reassociation request is able or not to use the new protocol, therefore it must default to authorize the request.

Chapter 8

Implementation

We implemented a prototype of our reauthentication and reassociation protocols using Linux (Fedora Core 8) and the MadWifi v0.9.4 driver and tools. This driver was updated to implement the protocols in both STAs and APs. For the STA we used `wpa_supplicant`, and for the AP we used `hostapd`

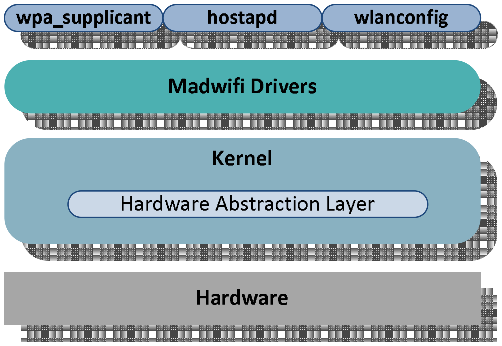


Figure 8.1: Structure our prototype implementation.

The *Multiband Atheros Driver for WiFi (Madwifi)*[1] supports wireless cards based on the Atheros chipsets. The Madwifi drivers are unique in that they support multiple interfaces on the same wireless card known as *Virtual Access Points (VAPs).*

Madwifi comes with a whole host of private commands (accessible through `iwpriv`) which allow getting and setting driver features/details.

The main goal of this prototype was to evaluate the handover latency using our new protocols (see Section 6). As previously explained, when a reauthentication is realized by an STA in advance, i.e. while being associated to an AP, this latency depends solely on the time to perform a reassociation to another AP. Consequently, we did not implement the Reauthentication Service as a separate service, but as part of the AP driver. External applications, namely the `iwpriv` command, are used to provide SDP-RK pairs for the driver. The driver running on a STA keeps only one pair; the driver running on a AP keeps all provided key pairs, overwriting existing ones.

The driver was extended to send, receive and process the new authentication and association messages, both on the STA and the AP side. We also defined a new private command accessible trough `iwpriv`, to initiate reauthentications in arbitrary APs using the previously downloaded SDP-RK pair. After a successful reauthentication, the drivers of both STA and AP keep internally an association

[1] This driver is not fully open source, it requires a hardware abstraction layer (HAL) closed source binary.

between the MAC address of the peer and a PTK key. For the reauthentication protocol we chose a new protocol number, 2 (0 is used for OSA and 1 for Shared Key Authentication).

The main modification in the code was made in the functions `ieee80211_recv_mgmt` and `ieee80211_send_mgmt` for the files `ieee80211_input.c` and `ieee80211_output.c` respectively.

The driver was extended as well to allow external applications of an STA to perform reassociations. Upon a reassociation request for an AP, given its MAC address, the driver looks for a local MAC-PTK association and, if present, starts the new reassociation protocol using PTK. After a successful reassociation, the drivers of both STA and AP update association information, namely the keys for protecting data exchanges, derived from the TK portion of PTK.

In this prototype implementation, we used AES to encrypt K and GTK, HMAC SHA-1 to compute the MICs with K or KCK and SHA-256 for computing PMK from K and N3. For the AES we used 128 bits for both keys and data blocks.

Chapter 9

Performance Evaluation

For evaluating the performance of the secure handover, we forced a reassociation between an STA and two APs using 802.11g. For both APs and STA we used Linux hosts with the same operating system version (Fedora Core 8) and wireless network interface (Netgear WG511T, with the Atheros chipset). For the AP hosts we used two equal 3.2 GHz Pentium 4 desktop hosts; for the STA we used a 1.7 GHz Pentium M laptop.

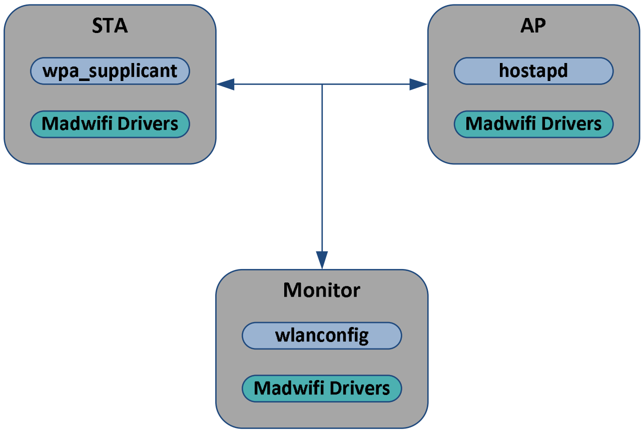


Figure 9.1: Model of our practical implementation.

9.1 Setting the AP

- This was the procedure used to configure the Madwifi AP in 802.1g using auto channel select, and WPA-PSK via *hostapd*. The user space program *hostapd* requires a configuration file. The AP will have an IP address of 192.168.0.20.

  The configuration file (named `/etc/hostapd.conf`) is shown below.

  ```
  interface=ath0 bridge=br0
  driver=madwifi
  logger_syslog=0
  logger_syslog_level=0
  logger_stdout=0
  ```

```
logger_stdout_level=0
debug=0
eapol_key_index_workaround=0
dump_file=/tmp/hostapd.dump.0.0
ssid=Atheros Wireless Network
wpa=1
wpa_passphrase=mypassphrase
wpa_key_mgmt=WPA-PSK
wpa_pairwise=TKIP CCMP
wpa_group_rekey=600
```

Now, the following commands will create the AP.

```
myprompt# wlanconfig ath create wlandev wifi0 wlanmode ap
ath0
myprompt# iwconfig ath0 essid "Atheros Wireless Network"
myprompt# iwpriv ath0 mode 11g
myprompt# brctl addbr br0
myprompt# brctl addif br0 eth0
myprompt# brctl addif ath0
 myprompt# brctl setfd br0 1
myprompt# ifconfig ath0 up
myprompt# ifconfig eth0 up
myprompt# ifconfig br0 192.168.0.20 up
myprompt# hostapd -dd /etc/hostapd.conf
```

- For configuring the AP using only the madwifi drivers with Shared Key Authentication:

  ```
  myprompt# wlanconfig ath create wlandev wifi0 wlanmode ap
  ath0
  myprompt# iwconfig ath0 essid "Atheros Wireless Network"
  myprompt# iwpriv ath0 authmode 2
  myprompt# iwconfig ath0 key "s:mypassphrase"
  myprompt# ifconfig ath0 up
  ```

9.2 Setting the STA

- To configure the driver to be a station attempting to associate with the WPA-PSK AP Above. The station will have an IP address of 192.168.0.100.

  The user space program `wpa_supplicant` requires a configuration file. The file used in this example is shown below and named `/tmp/my_psk.conf`.

  ```
  network={
       ssid="Atheros Wireless Network"
       scan_ssid=1
       key_mgmt=WPA-PSK
       psk="mypassphrase"
  }
  ```

Now, the following commands will create the station which will scan for the AP with an SSID of `Atheros Wireless Network`.

```
mypromt# wlanconfig ath create wlandev wifi0 wlanmode sta
ath0
mypromt# iwconfig ath0 essid "Atheros Wireless Network"
mypromt# ifconfig ath0 192.168.0.100 up
mypromt# wpa_supplicant -iath0 -c /tmp/my_psk.conf -d
```

- For configuring the driver to be a station to associate with Shared Key Authentication:

```
mypromt# wlanconfig ath0 destroy
mypromt# wlanconfig ath create wlandev wifi0 wlanmode sta
mypromt# iwconfig ath0 essid "Atheros Wireless Network"
mypromt# iwpriv ath0 authmode 2
mypromt# iwpriv ath0 mode 11g
mypromt# iwconfig ath0 key "s:mypass"
mypromt# iwconfig ath0 channel 2
mypromt# ifconfig ath0 up
```

9.3 Setting the Monitor

The *ath0* interface named for the Atheros wireless chipset (*ath*) which is created by default in managed mode. In order to configure an interface in monitor mode, we must delete or ”destroy” this interface using the *wlanconfig* utility:

```
mypromt# wlanconfig ath0 destroy
mypromt# wlanconfig aht0 create wlandev wifi0 wlanmode monitor
mypromt# ifconfig ath0 up
mypromt# iwconfig ath0 channel X
```

The *X* is the channel were we want to capture wireless traffic. This channel as to be the same as the AP and the STA. Recall that the wireless cards can only capture traffic on a single time channel at any given time.

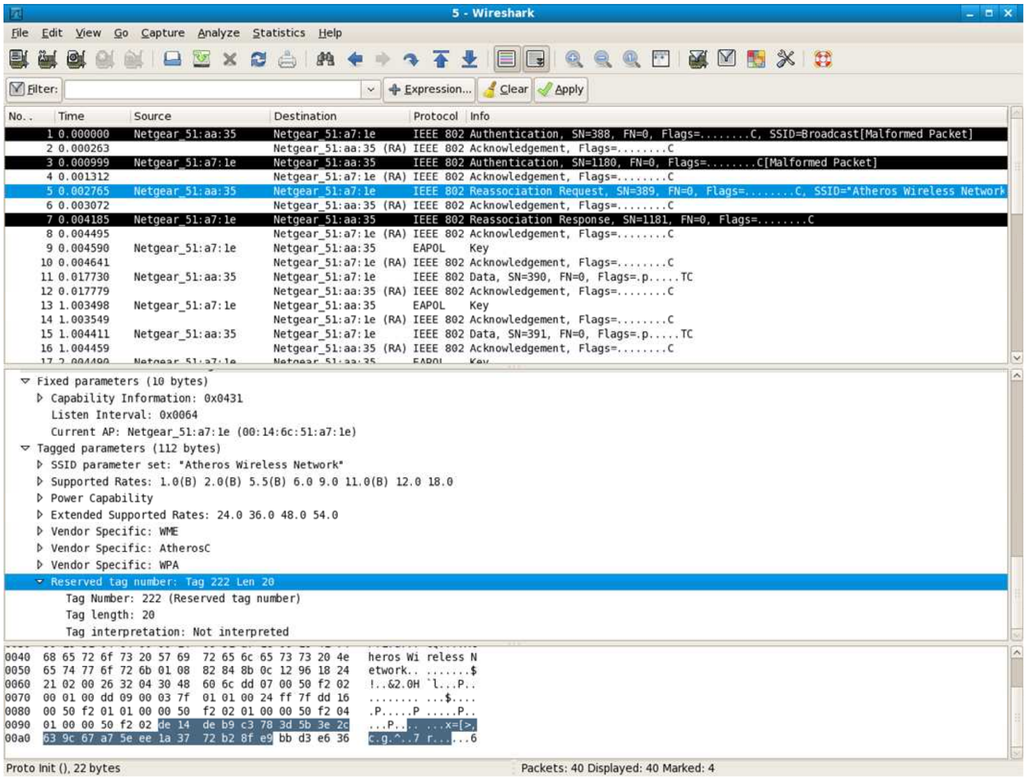
9.4 Results

Once the wireless card in linux have been placed in monitor mode, we started `wireshark` and defined a filter to only capture the packets of the AP and the STA.

After the STA being successfully associated with the AP we used our private command to force a preauthentication and a reassociation using the following command:

```
mypromt# roam_auth 00:14:6C:51:A7:23
```

From network captures with WireShark we observed that the new 802.11 reassociation protocol takes about 1.5 ms. The reassociation delay is the the interval between the observation of the 802.11 Reassociation Request and the arrival of the 802.11 Reassociation Response frame as we can see in



```
No. .  Time      Source             Destination             Protocol Info
     1 0.000000  Netgear_51:aa:35   Netgear_51:a7:1e        IEEE 802 Authentication, SN=388, FN=0, Flags=........C, SSID=Broadcast[Malformed Packet]
     2 0.000263                     Netgear_51:aa:35 (RA)   IEEE 802 Acknowledgement, Flags=........C
     3 0.000999  Netgear_51:a7:1e   Netgear_51:aa:35        IEEE 802 Authentication, SN=1180, FN=0, Flags=........C[Malformed Packet]
     4 0.001312                     Netgear_51:a7:1e (RA)   IEEE 802 Acknowledgement, Flags=........C
     5 0.002765  Netgear_51:aa:35   Netgear_51:a7:1e        IEEE 802 Reassociation Request, SN=389, FN=0, Flags=........C, SSID="Atheros Wireless Network
     6 0.003072                     Netgear_51:aa:35 (RA)   IEEE 802 Acknowledgement, Flags=........C
     7 0.004185  Netgear_51:a7:1e   Netgear_51:aa:35        IEEE 802 Reassociation Response, SN=1181, FN=0, Flags=........C
     8 0.004495                     Netgear_51:a7:1e (RA)   IEEE 802 Acknowledgement, Flags=........C
     9 0.004590  Netgear_51:a7:1e   Netgear_51:aa:35        EAPOL    Key
    10 0.004641                     Netgear_51:a7:1e (RA)   IEEE 802 Acknowledgement, Flags=........C
    11 0.017730  Netgear_51:aa:35   Netgear_51:a7:1e        IEEE 802 Data, SN=390, FN=0, Flags=.p.....TC
    12 0.017779                     Netgear_51:aa:35 (RA)   IEEE 802 Acknowledgement, Flags=........C
    13 1.003498  Netgear_51:a7:1e   Netgear_51:aa:35        EAPOL    Key
    14 1.003549                     Netgear_51:a7:1e (RA)   IEEE 802 Acknowledgement, Flags=........C
    15 1.004411  Netgear_51:aa:35   Netgear_51:a7:1e        IEEE 802 Data, SN=391, FN=0, Flags=.p.....TC
    16 1.004459                     Netgear_51:aa:35 (RA)   IEEE 802 Acknowledgement, Flags=........C
    17 2.004490  Netgear_51:a7:1e   Netgear_51:aa:35        EAPOL    Key
```

```
▽ Fixed parameters (10 bytes)
  ▷ Capability Information: 0x0431
    Listen Interval: 0x0064
    Current AP: Netgear_51:a7:1e (00:14:6c:51:a7:1e)
▽ Tagged parameters (112 bytes)
  ▷ SSID parameter set: "Atheros Wireless Network"
  ▷ Supported Rates: 1.0(B) 2.0(B) 5.5(B) 6.0 9.0 11.0(B) 12.0 18.0
  ▷ Power Capability
  ▷ Extended Supported Rates: 24.0 36.0 48.0 54.0
  ▷ Vendor Specific: WME
  ▷ Vendor Specific: AtherosC
  ▷ Vendor Specific: WPA
  ▽ Reserved tag number: Tag 222 Len 20
      Tag Number: 222 (Reserved tag number)
      Tag length: 20
      Tag interpretation: Not interpreted
```

```
0040  68 65 72 6f 73 20 57 69  72 65 6c 65 73 73 20 4e   heros Wi reless N
0050  65 74 77 6f 72 6b 01 08  82 84 8b 0c 12 96 18 24   etwork.. .......$
0060  21 02 00 26 32 04 30 48  60 6c dd 07 00 50 f2 02   !..&2.0H `l...P..
0070  00 01 00 dd 09 00 03 7f  01 01 00 24 ff 7f dd 16   ........ ...$....
0080  00 50 f2 01 01 00 00 50  f2 02 01 00 00 50 f2 04   .P.....P .....P..
0090  01 00 00 50 f2 02 de 14  de b9 c3 78 3d 5b 3e 2c   ...P.... ...x=[>,
00a0  63 9c 67 a7 5e ee 1a 37  72 b2 8f e9 bb d3 e6 36   c.g.^..7 r......6
```

Proto Init (), 22 bytes

Packets: 40 Displayed: 40 Marked: 4

Figure 9.2: Capture with Wireshark.

figure 9.2. As we can see in the packet details pane, we added a new element ID of length 20 bytes, and a unused tag number 222, to carry the payload introduced by our protocol.

The delay of 802.1X authentications was evaluated in [35] and the minimum value (150 ms) was obtained with 802.1X fast resume; the minimum value obtained for the 4WHP was 10 ms. Thus, our reassociation delay is at least two orders of magnitude smaller than the fastest reassociation with 802.1X and at least one order of magnitude smaller than the 4WHP. This happens because we use less frames, only two, while 802.1X uses at least eight and 4WHP uses four, and the AP does not need to contact other network entities to perform an authenticated reassociation. Instead, it only uses local data: the key PTK previously installed using the new reauthentication protocol.

Comparing our performance results with the ones presented by similar contributions (the ones that totally or partially recreate security contexts in new serving APs), we are clearly ahead. In [22] an STA may have to wait for the establishment of a secure tunnel between the current AP and the previously serving AP, with the help of an AS, after a reassociation. The delay of such procedure was not evaluated but surely it is much higher then 1.5 ms, as it uses 6 messages. In [23] the authentication and key distribution with ERP and an HOKEY server after a reassociation takes more than 184 ms. In [26], which partially uses secure context migration, the handover delay is higher than 8 ms (value obtained with a simulator).

Chapter 10

Conclusions

Many approaches have been presented using context transfer based protocols in an effort to eliminate the need for reassociation procedures. These approaches add complexity to the network because of the need of new entities to manage the context transfer operations and keying material. Some of them use the same keying material when a handover occurs, others create new key material, thus needing to perform a four-way handshake to ensure the freshness of the keys. They also need to ensure the update of the routing tables.

Other approaches try the modify the normal 802.1X in an effort to reduce roundtrip delays and the number of messages. They fail by not authenticating reassociation requests and most of them are not able to eliminate the four-way handshake, that adds a considerably latency to the handover process.

In some published words it's not clear how certain requirements are met, like the fact that an STA can have only a single association at any given time or how mutual authentication and fresh key derivation at each AP happens.

We presented a new approach for achieving fast, 802.1X-like authentications during reassociations. Our approach recovers the original 802.11 model, using authentications prior to reassociations. This way, we are able to remove reauthentication delays from handover outage delays, since they can be executed before reassociations (see Figures 1.1 and 1.2). Furthermore, we are able to authenticate reassociation protocols, which is useful against DoS attacks.

The proposed architecture uses a key hierarchy starting in EMSK and a Reauthentication Service for dealing with reauthentications of STAs. This new service can be implemented by an AS or an HOKEY server. Finally, we do not require special configuration facilities from network equipments, no topological information regarding the actual deployment of APs and no homogeneity among facilities provided by APs (e.g. homogeneous cipher suits).

We also proposed a solution to integrate Mobile IP in our architecture. This was a simple matter once that we had already a well defined key hierarchy and a RS that distribute the needed keys to the HA. With these keys we were able to perform mutual authentication and therefore preventing denial-of-service attacks. Most important of all we presented a solution were it was possible to completely eliminate the network configuration delay, by doing it prior to the handover.

With a prototype implementation we measured handover times around 1.5 ms. Although our work did not include QoS, there is a lot of room for manoeuver being that the ITU-T [1] considers acceptable a latency for a VoIP network below 150 ms, and that VoIP requires 802.11 transition times, including 802.11 security setup below 50 ms.

[1] The ITU Telecommunication Standardization Sector (ITU-T) coordinates standards for telecommunications on behalf of the International Telecommunication Union (ITU)

In conclusion our protocol achieves fast, secure 802.1x-like (re)authentication, and (re)associations fully compliant with the IEEE 802.11 standard. It added a security enhancement by securing the channel prior to (re)association, enabling the secure exchange of control and management frames, a long discussed issue due to the appearing of new amendments to the IEEE 802.11 standard like for instance the 802.11k. We also proposed a extension of our protocol to the layer 3 using Mobile IP, eliminating completely the network configuration outage from the critical handover time.

Our protocol requires very little from the environment, as it only adds a RS to the network architecture. It clearly defines how the new key hierarchy is derived and, most important, it defines how and when the new keys are distributed, eliminating the need for proprietary enhancements to fulfill this task, as opposed to the IEEE 802.11r that fails in defining how keys are distributed.

Bibliography

[1] M. Nakhjiri and Y. Ohba, ”, IETF HOKEY WG Internet-Draft, Nov. 2007, draft-ietf-hokey-key-mgm-01.

[2] J. Salowey, L. Dondeti, V. Narayanan, and M. Nakhjiri, “Specification for the Derivation of Root Keys from an Extended Master Session Key (EMSK)”, IETF HOKEY WG Internet-Draft, Nov. 2007, draft-ietf-hokey-emsk-hierarchy-02.

[3] James D. Solomon, *Mobile IP: the Internet unplugged*, Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1998.

[4] C. Perkins, “IP Mobility Support”, RFC 2002, IETF, Oct. 1996.

[5] “Ieee standard for 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”, *IEEE Std 802.11-2007 (Revision of IEEE Std 802.11-1999)*, pp. C1–1184, 12 2007.

[6] Bernard Aboba, “Fast Handoff Issues”, IEEE 802.11-04/827r0, July 2004.

[7] Ishwar Ramani and Stefan Savage, “SyncScan: Practical Fast Handoff for 802.11 Infrastructure Networks”, in *INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE*, Mar. 2005, vol. 1, pp. 675–684.

[8] Haixing He and Darwin Engwer, “Authenticated State and IEEE 802.11 Pre-Authentication”, IEEE 802.11-04/827r0, July 2004.

[9] Bernard Aboba, “IEEE 802.11i: A Retrospective”, Mar. 2004.

[10] L. Blunk and J. Vollbrecht, “PPP Extensible Authentication Protocol (EAP)”, RFC 2284, IETF, Mar. 1998.

[11] Jesse Walker and Emily Qi, “Pre-Keying”, IEEE 802.11-04/0476r0, May 2004.

[12] J. Moy, “OSPF specification”, RFC 1131, IETF, Oct. 1989.

[13] Y. Rekhter, T. Li, and Eds., “A Border Gateway Protocol 4 (BGP-4)”, RFC 1654, IETF, July 1994.

[14] R. Rivest, “The MD5 Message-Digest Algorithm”, RFC 1321, IETF, Apr. 1992.

[15] A. R. Prasad and H. Wang, “Roaming key based fast handover in WLANs”, in *IEEE Wireless Communications and Networking Conf. (WCNC 2005)*, Mar. 2005, vol. 3, pp. 1570–1576.

[16] M. Kassab, A. Belghith, J. Bonnin, and S. Sassi, “Fast Pre-Authentication Based on Proactive Key Distribution for 802.11 Infrastructure Networks”, in *1st ACM Works. on Wireless Multimedia Networking and Performance Modelling (WMuNeP’05)*, Montreal, Canada, Oct. 2005.

[17] B. Sarikaya and X. Zheng, “CAPWAP Handover Protocol”, in *IEEE Int. Conf. on Communications (ICC’06)*, June 2006, vol. 4, pp. 1933–1938.

[18] L. Zan, J. Wang, and L. Bao, “Personal AP Protocol for Mobility Management in IEEE 802.11 Systems”, in *Proc. of the 2nd Ann. Int. Conf. on Mobile and Ubiquitous Systems: Networking and Services (MOBIQUITOUS’05)*, Washington, DC, USA, 2005, pp. 418–425, IEEE Computer Society.

[19] S. Govindan, H. Cheng, Z. H. Yao, W. H. Zhou, and L. Yang, “Objectives for Control and Provisioning of Wireless Access Points (CAPWAP)”, RFC 4564, IETF, July 2006.

[20] M. Nakhjiri and Y. Ohba, ”, IETF HOKEY WG Internet-Draft, Nov. 2007, draft-ietf-hokey-key-mgm-01.

[21] LAN/MAN Standards Committee of the IEEE Computer Society, “Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, Amendment 6: Medium Access Control (MAC) Security Enhancements”, IEEE Std 802.11i, July 2004.

[22] J. Chen, Y. Tseng, and H. Lee, “A Seamless Handoff Mechanism for DHCP-Based IEEE 802.11 WLANs”, *IEEE Communications Letters*, vol. 11, no. 8, pp. 665–667, Aug. 2007.

[23] R. Marin, P. J. Fernandez, and A. F. Gomez, “3-Party Approach for Fast Handover in EAP-Based Wireless Networks”, in *Proc. of OTM Conferences, 2nd Int. Symp. on Information Security (IS’07)*, Vilamoura, Portugal, Nov. 2007, pp. 1734–1751, Springer, LNCS 4804.

[24] T. Clancy, M. Nakhjiri, V. Narayanan, and L. Dondeti, “Handover Key Management and Re-authentication Problem Statement ”, IETF HOKEY WG Internet-Draft, Nov. 2007, draft-ietf-hokey-reauth-ps-07.

[25] V. Narayanan and L. Dondeti, “EAP Extensions for EAP Re-authentication Protocol (ERP)”, IETF HOKEY WG Internet-Draft, Nov. 2007, draft-ietf-hokey-erx-08.

[26] Chung-Ming Huang and Jian-Wei Li, “An IEEE 802.11 Fast Reassociation and Pairwise Transient Key establishment Based on the Dynamic Cluster Method”, in *Works. of Computer Networks and Wireless Communications, Int. Computer Symp. (ICS 2006)*, Taipei, Taiwan, 2006.

[27] A. Mishra, Min Ho Shin, Jr. N. L. Petroni, T. C. Clancy, and W. A. Arbaugh, “Proactive key distribution using neighbor graphs”, *IEEE Wireless Communications*, vol. 11, no. 1, pp. 26–36, Feb 2004.

[28] Raymond Greenlaw and Paul Goransson, *Secure Roaming in 802.11 Networks*, Elsevier, 2007, ISBN-13 978-0-7506-8211-4.

[29] T. Clancy, M. Nakhjiri, V. Narayanan, and L. Dondeti, “Handover Key Management and Re-Authentication Problem Statement”, RFC 5169 (Informational), Mar. 2008.

[30] C. Kaufman, “Internet Key Exchange (IKEv2) Protocol”, RFC 4306, IETF, Dec. 2005.

[31] S. Pack and Y. Choi, “Fast Inter-AP Handoff using Predictive-Authentication Scheme in a Public Wireless LAN”, in *IEEE Networks Conf. (Joint IEEE ICN 2002 and IEEE ICWLHN)*, Aug. 2002.

[32] Arunesh Mishra, Minho Shin, and William A. Arbaugh, “An empirical analysis of the ieee 802.11 mac layer handoff process”, *Computer Communication Review*, vol. 33, no. 2, pp. 93–102, 2003.

[33] H. Velayos and G. Karlsson, “Techniques to reduce IEEE 802.11b MAC layer handover time”, Tech. Rep. TRITA-IMIT-LCN R 03:02, Kungl. Tekniska Hogskolen, Stockholm, Sweden, Apr. 2003.

[34] The Avispa Team, *Automated Validation of Internet Security Protocols and Applications (AVISPA) v1.1 User Manual*, June 2006.

[35] A. Alimian and B. Aboba, “Analysis of Roaming Techniques”, IEEE 802.11 WG document 802.11-04/0377r1, 2004.