Project Number:	**IST-2001-32603**
Project Title:	**6NET**
CEC Deliverable Number:	**32603/ULANC/DS/4.2.2/A1**
Contractual Date of Delivery to the CEC:	June 30th 2003
Actual Date of Delivery to the CEC:	August 6th 2003
Title of Deliverable:	Framework for the Support of IPv6 Wireless LANs
Work package contributing to Deliverable:	WP4
Type of Deliverable*:	R
Deliverable Security Class**:	PU
Editors:	Martin Dunmore
Contributors:	Martin Dunmore, John Floroiu, Reinhard Ruppelt, Klaas Wierenga, Renzo Davoli

* Type: P - Prototype, R - Report, D - Demonstrator, O - Other

** Security Class: PU- Public, PP – Restricted to other programme participants (including the Commission), RE – Restricted to a group defined by the consortium (including the Commission), CO – Confidential, only for members of the consortium (including the Commission)

Abstract:

This document aims to provide a framework for the support of wireless LANs (WLANs) in an IPv6-only environment. It is hoped that it will serve as an IPv6 WLAN design and implementation guide for readers that are unfamiliar with IPv6 and/or WLANs.

The original focus for this document was to be on access control solutions but we have now expanded the scope to include all aspects of deploying IPv6 WLANs. While authentication and access control still remain a significant part of this deliverable, we also look at the issues in designing and implementing the WLAN together with IPv6 or MIPv6 specific issues that should be taken into consideration. We have also added a section on hardware and software availability.

We also focus on IEEE 802.11 based WLANs since these are by far the most widely deployed and are certainly beginning to live up to the 'wireless Ethernet' tag.

This is the first version of this deliverable, a second and final version is due in M30 of the project (June 2004).

Keywords:

Wireless LAN, Access, 802.11, IPv6, Mobile IPv6.

Executive Summary

Within activity A4.2, the first deliverable D4.2.1 identified and discussed the important issues pertaining to IPv6 Wireless LAN access. This deliverable expands on these issues and provides a framework document for organisations wishing to deploy and operate Wireless LANs in an IPv6 environment.

In recent years there has been a rather dramatic growth in the WLAN market with 802.11 WLANs now beginning to appear in enterprise networks, public 'hotspots', campuses and small office and home (SOHO) networks. It is now becoming extremely common for business meetings, conferences, trade fairs and exhibitions to provide WLAN access to people attending those events. In the context of IPv6 this growth is somewhat important. Most WLANs are new network deployments, rather than to replace existing wired networks. Therefore, these new WLANs are beginning to eat into the dwindling IPv4 address space. Whilst solutions such as NAT can hide the address shortage, they are really only temporary solutions until the next generation IP, IPv6, becomes widely deployed. It is therefore important to arrive at a framework for the support of WLANs in IPv6-only environments to accommodate future growth in WLAN deployment.

The scope of this deliverable has been expanded to include all aspects of supporting IPv6-only WLANs rather than just access control solutions as was originally intended. We now also look at the issues in designing and implementing the WLAN together with IPv6 or MIPv6 specific issues that should be taken into consideration.

This is the first version of this document, a second and final version is due in M30 of the project (June 2004). It is hoped that it will serve as an IPv6 WLAN design and implementation guide for readers that are unfamiliar with IPv6 and/or WLANs.

Table of Contents

1 Introduction

This document aims to provide a framework for the support of wireless LANs (WLANs) in an IPv6-only environment. Since 6NET is an IPv6 project, the focus of the material in this document concerns IPv6. However, much of the material is also applicable to IPv4. This is not surprising considering that the wireless LAN (WLAN) is a layer 2 technology and is, in theory, independent of the network layer.

This is the first version of this document, a second and final version is due in M30 of the project (June 2004). It is hoped that it will serve as an IPv6 WLAN design and implementation guide for readers that are unfamiliar with IPv6 and/or WLANs.

The original focus for this document was to be on access control solutions but we have now expanded the scope to include all aspects of deploying IPv6 WLANs. While authentication and access control still remain a significant part of this deliverable, we also look at the issues in designing and implementing the WLAN together with IPv6 or MIPv6 specific issues that should be taken into consideration. We also focus on IEEE 802.11 based WLANs since these are by far the most widely deployed and are certainly beginning to live up to the 'wireless Ethernet' tag.

Since the original IEEE 802.11 WLAN specification, there are now have a further three flavours of 802.11 WLAN types (named 'b', 'a' and 'g') available to deploy. In recent years there has been a rather dramatic growth in the WLAN market with 802.11 WLANs now beginning to appear in enterprise networks, public 'hotspots', campuses and small office and home (SOHO) networks. It is now becoming extremely common for business meetings, conferences, trade fairs and exhibitions to provide WLAN access to people attending those events. In the context of IPv6 this growth is somewhat important. Most WLANs are new network deployments, rather than to replace existing wired networks. Therefore, these new WLANs are beginning to eat into the dwindling IPv4 address space. Whilst solutions such as NAT can hide the address shortage, they are really only temporary solutions until the next generation IP, IPv6, becomes widely deployed. It is therefore important to arrive at a framework for the support of WLANs in IPv6-only environments to accommodate future growth in WLAN deployment.

The following chapter gives an overview of the different IEEE 802.11 WLAN types available; their characteristics, typical data rates and their expected coverage range. Chapter three investigates a number of solutions to authentication, access control and data privacy in a WLAN environment. This chapter looks at WEP, Wi-Fi Protected Access (WPA), 802.1x, 802.11i, VPNs, web-based authentication and also a novel IPv6-specific mechanism that utilises packet marking.

Chapter four looks at the design and deployment of WLANs and discusses the site survey, how to choose the 802.11 WLAN, type, how many access points (APs) will be required and the problems of configuring and managing a large number of APs.

Chapter five explores some IPv6/MIPv6 specific issues when deploying a WLAN. These issues include how to configure hosts, how to subnet and address the wireless network and inconsistencies with IPv6 duplicate address detection (DAD). The sixth chapter gives some insight into available hardware and software for deploying WLANs and the final chapter draws some conclusions.

2 IEEE 802.11 Wireless LAN Technologies

The base 802.11 specification consists of the 802.11 MAC (Medium Access Control) and two[1] different PHY (physical) layers: frequency hopping spread spectrum (FHSS) and direct-sequence spread spectrum (DSSS), both of which operate at the 2.4GHz frequency band with a maximum data rate of 1 and 2Mbps resepctively. This original standard ensured interoperability of communications equipment using the same PHY layer types but not between different PHY types. Since then, several standards have been developed under the 802.11 umbrella that allow much higher data rates.

2.1 802.11b

The 802.11b specification (aka Wi-Fi) is a revision to the 802.11 PHY that specifies a high-rate direct-sequence spread spectrum (HR/DSSS) with a maximum data rate of 11Mbps. Like the original specification, 802.11b operates in the 2.4 GHz band and has 13 operating channels in Europe, all 5 MHz wide from 2.412 – 2.472 GHz. Since these channels overlap with each other, it is only possible to choose a maximum of three separate channels within the same coverage area that do not overlap. This has a direct effect on AP density in a coverage area and thus maximum throughput. In other words, a maximum of three APs can be located in the same RF coverage area without causing interference with each other. Although the theoretical maximum data rate for an 802.11b is 11Mbps, actual data rates are more likely to be in the range of 4 to 6Mbps depending on the physical environment. In typical office environments, an 802.11b AP has a maximum range[2] of about 75 metres, but at the highest operating speed this range can be as low as 30 metres.

2.2 802.11a

The 802.11a specification (aka Wi-Fi5) is a revision to the 802.11 PHY that is based on orthogonal frequency division multiplexing (OFDM) operating at the 5GHz frequency band with a maximum data rate of 54Mbps. Actual data rates are likely to be in the range of 25 to 30 Mbps depending on the physical environment. In typical office environments, 802.11a has a maximum range of about 50 metres, but at the highest operating speed this can be as low as 23 metres. 802.11a therefore has less signal range than 802.11b and so will require more APs for the same RF coverage area. However, 802.11a can have more non-overlapping channels than 802.11b, with four, eight or more channels available depending on the country. Since it uses a different frequency band, the 802.11a standard is incompatible with both 802.11b and 802.11g. In other words, an 802.11a device is not able to associate with 802.11b or 802.11g APs.

The recent International Telecommunications Union's World Radiocommunication Conference[3], held in June/July 2003 in Geneva, made significant progress in coordinating global rules for operating 802.11a wireless LANs in the 5 GHz frequency range. The ITU’s decisions increase the number of non-overlapping channels available for WLAN use, which will improve potential throughput and boost overall WLAN scalability. The WRC's delegates, who included industry vendors and national government bodies, agreed to allocate 455 MHz of unlicensed spectrum in the 5 GHz band for global WLAN use. Once the agreement achieves final approval, 100 MHz (5.150-5.250 GHz) will be allocated for indoor WLAN use, and an additional 355 MHz (5.250-5.350 GHz and 5.470-5.725 GHz) will be allocated for mixed indoor/outdoor use.

[1] There is also a third PHY for infrared (IR) although this is largely irrelevant as infrared ports on laptops etc. are predominantly compliant with Infrared Data Association (IrDA) standards instead.
[2] Assuming omnidirectional antennae. Higher ranges can be achieved with high-gain and/or directional antennae.
[3] http://www.itu.int/ITU-R/conferences/wrc/wrc-03/index.asp

This will result in 24 non-overlapping 5 GHz channels in the U.S. with most member nations of the European Community reportedly opening 19 non-overlapping 5-GHz channels for 802.11h use (this is 802.11a with some extensions for avoiding military radar).

2.3 802.11g

The 802.11g specification achieved standard status in July 2003. This standard is another revision to the 802.11 PHY and like 802.11a, is also based on OFDM with a maximum data rate of 54 Mbps. However, 802.11g is designed to be backward compatible with 802.11b. Thus, 802.11b devices can associate with 802.11g APs albeit at the lower data rate of 802.11b. As with 802.11b, 802.11g has a maximum of three non-overlapping channels.

3 Authentication, Security and Privacy

With the advent of the mobile computing devices, the access control and data protection mechanisms have faced new challenges posed by the new nature of the terrestrial wireless communication and ensuing roaming scenarios. In short, by removing the burden of the binding to fixed physical locations, existing threats to the network exploitation and data security have become critical.

A number of elements decisively impact on the scope and shape of the access control and data protection schemes, among them, the ones enumerated below:

1. Physical access of the intruders cannot be limited in any way, which means that eavesdropping becomes trivial, increasing the exposure of the wireless networks to any kind of attacks. Encryption becomes therefore an essential feature wireless access technologies must support;
2. The authentication model must support roaming scenarios, which requires the definition of appropriate trust relationships and authentication schemes between client, access network, visited domain and a third trust party (client's home entity);
3. It is always possible nodes originating from multiple administrative domains may be simultaneously located on the same link, which means that the security mechanisms must have a per-node rather than per-wireless domain or per-access point granularity.
4. When mobile nodes roam within the same domain, support must be provided to enable contexts associated to mobile nodes (including authentication credentials and data protection primitives) migrate from one access router to another without requiring low latency operations;

Wireless LAN technology on the basis of the IEEE 802.11 WLAN standard has become a key aspect of internetworking. The tremendous speed wireless LANs have spread across (still accelerating) at the same time has attracted the focus on the security potentialities of WLANs.

Already in the early stages IEEE's WEP (Wired Equivalent Privacy) security standard which was designed to provide confidentiality for network traffic using the wireless protocol. turned out to be absolutely ineffective. Already in October 2000 Walker was one of the first people to identify several of the problems in WEP [10].

In January 2001 researchers[1] at the University of California at Berkely independently released a paper describing the problems with WEP. Finally, in August 2001, Fluhrer, Mantin, and Shamir described several weaknesses in the key scheduling algorithm of RC4 and proposed attacks for exploiting those weaknesses [11]. The detection of this flaw in the RC4 key setup algorithm theoretically facilitates the total recovery of the secret key.

Shortly after, on August 21, 2001, Stubblefield, Ioannidis, and Rubin were the first to implement this attack. They published a technical report [12] describing the implementation with which it was possible to recover the 128 bit secret key used in a production network, with a passive attack exploiting a WEP design failure where the WEP standard uses RC4 initialization vectors (IV) improperly. They concluded that 802.11 WEP is totally insecure and provided recommendations.

At the same time open source crack tools AirSnort[2,3] and WEPCrack[4], were released as the first publicly available implementations of this attack.

[1] http://www.isaac.cs.berkeley.edu/isaac/wep-faq.html
[2] http://airsnort.shmoo.com/
[3] http://sourceforge.net/projects/airsnort
[4] http://sourceforge.net/projects/wepcrack

3.1 Wired Equivalent Privacy (WEP)

The *Wireless Equivalent Privacy* algorithm (WEP) is IEEE802.11's optional encryption standard [7][1] implemented in the MAC Layer which most radio network interface card- and access point vendors support. WEP has been part of the 802.11 standard since its initial ratification in September 1999. Originally, WEP was designed to provide eavesdrop-proof comparable to that of wired LANs. For this purpose WEP defines functions, based on the RC4 encryption mechanism, which aimed at supporting both data encryption and authentication / data integrity as well where the second function is not an explicit goal in the 802.11 standard.

The specification is based on the following major safety measures:

1. Data encryption using the stream cipher algorithm RC4,
2. Shared secret key,
3. CRC-32 Integrity Check (IC), and a
4. 24 Bit Initialization Vector (IV)

This security framework involves the following problems:

For every stream cipher algorithm it is essential to use a new key for each new data packet to be encrypted[2]. Otherwise an identical data stream would be coded into identical cipher data. In order to achieve this WEP defines variable initialization vectors (IV) which are used to modify the constant WEP key. These IVs are transmitted in the clear text part of the radio message. The problem with the IVs however is that the standard IV length is only 24 bit – too little for efficient data protection: For an access point which is continuously sending data frames at the latest after 5 hours the set of available IVs is exhausted. Worse still, in reality the use of the same effective key for different data frames happens more often[3] considerably facilitating the detection of the constant WEP key by passive monitoring of the radio traffic.

Another approach (*Known-Plaintext* attack) uses pairs of encrypted data and the corresponding original data to record key streams which belong to different IVs. With the knowledge of IVs and the associated key streams it is possible, when later an already known IV is detected, to decrypt the respective stream cipher by simply XOR-ing the data with the key stream formerly registered for this IV.

The use of a CRC-32 integrity check presents another weakness of WEP as it is part of the encrypted payload of the packet, and it is linear, which means that it is possible to compute the bit difference of two CRCs based on the bit difference of the messages over which they are taken. This property can be used to by an skilful attacker to hide modifications made on the encrypted data by adjusting the checksum so that the resulting message appears valid.

Besides the WEP encryption part its authentication procedure is vulnerable as well. The *Shared-Key-Authentication* provides a known-plaintext to a potential attacker since the original and the encrypted data as well can be read from the authentication sequence.[4]

[1] http://grouper.ieee.org/groups/802/11/is the official 802.11 site from the IEEE. Download of specs is now for free.
[2] Note: The IEEE802.11 standard specifies that changing the IV with each packet is optional!
[3] E. g. in case the same key is used by all mobile stations which is widely-used habit.
[4] For further information see also the FAQ concerning the security of the WEP algorithm at http://www.isaac.cs.berkeley.edu/isaac/wep-faq.html

3.2 802.1x and EAP

As described above WEP was intended to provide security between communicating wireless peers by employing symmetric encryption keys. One limitation of WEP as we have seen has been the task of distributing and managing the encryption keys.

The IEEE has proposed the Robust Security Network (RSN) as a long term security architecture for the 802.11 standard.

In order to ease the control of such security scenarios IEEE developed the 802.1x standard (based on developments by Cisco, 3Com, Agere, Compaq und Dell and others), which is used in RSN to provide a means of effectively authenticating and authorizing devices attached to a LAN port.

Specifically the 802.1x supplement[1] defines a mechanism for port-based network access control that makes use of the physical access characteristics of an IEEE 802 LAN infrastructure. It also prevents access to that port in cases in which the authentication and authorization fails.

Employing these features the standard pledges to provide a central, scalable and easy to manage solution for the required authentication and access control purposes of WLAN clients attaching to wireless access points. For these purposes 802.1x implements the *Extensible Authentication Protocol* (EAP) [13] and adopts a central RADIUS server [14] for authentication.

The picture below illustrates the flow in an 802.1x authentication

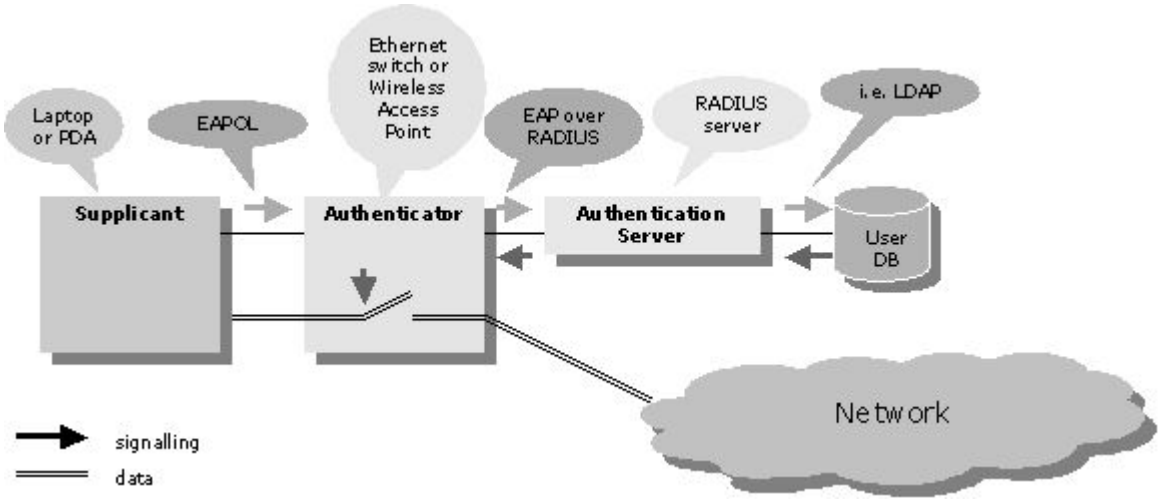

Figure 1 802.1x and EAP

In this flow, the client submits credentials within an EAP on LAN message; the AP transforms this into EAP over RADIUS and sends it to the Authentication Server that validates the credentials. Note that the AP in this scenario is almost transparent; its only task is to transform EAPOL messages into EAP over RADIUS messages.

There are several 802.1x based security solutions on the market today, like:

- Lightweight EAP (LEAP) a proprietary solution from Cisco

[1] ISO/IEC 15802-3:1998 (IEEE Std 802.1D-1998)

- Protected EAP (PEAP) [19] from Microsoft, Cisco, and RSA Security
- EAP TLS[1] [20]
- EAP SKE[2], [21]
- EAP TTLS[3] [22]
- EAP MD5 [23]
- EAP SIM [24]

and others, which are to repair the weaknesses of 802.1x when using less secure EAP-types according to their manufacturers.

The main EAP solutions are compared in Table 1[4].

Topic	EAP MD5	LEAP	EAP TLS	PEAP	EAP TTLS
Security Solution	Standards-based	Proprietary	Standards-based	Standards-based	Standards-based
Certificates – Client	No	n/a	Yes	No	No
Certificates – Server	No	n/a	Yes	Yes	Yes
Credential Security	None	Weak	Strong	Strong	Strong
Supported Authentication Databases	Requires clear-text database	Active Directory, NT Domains	Active Directory	Active Directory, NT Domains, Token Systems, SQL, LDAP etc.	Active Directory, NT Domains, Token Systems, SQL, LDAP
Dynamic Key Exchange	No	Yes	Yes		Yes
Mutual Authentication	No	Yes	Yes		Yes

Table 1 Security features of different EAP solutions

In February 2002, Arbaugh and Mishra published the results of an investigation describing two scenarios showing how an attacker could misuse a successful authentication to gain unauthorized access [15].

In the first scenario (*session hijacking*) an attacker uses an already existent client / access point connection (after authentication has taken place) to take over the connection pretending to be a valid access point (AP). Afterwards the AP terminates the connection to the client by means of a 802.11-MAC-Disassociate message. The client now supposes to be disconnected while the attacker plays the role of the former client using its

[1] EAP Transport Level Security
[2] EAP Shared Key Exchange
[3] EAP Tunneled TLS Authentication Protocol
[4] based on http://www.funk.com/radius/Solns/EAP_type_chart.gif © Funk Software Inc.

MAC address and in so doing it gains network access. This session hijacking attack is possible because 802.11 doesn't require authentication and integrity checks in case of management frames.

The second scenario (*man-in-the-middle*) is facilitated by the circumstance that the AP in fact authenticates the client but not vice versa. This allows an attacker to suitably tamper the message confirming a successful authentication (EAP success) sent by the AP to the WLAN client. From now on an attacker can impersonate himself as client against the AP and as AP against the client allowing him to inspect and consistently modify all passing messages. This manipulation is made possible because the EAP success message does not carry any integrity information.

Arbaugh and Mishra conclude that "the current RSN architecture does not provide strong access control and authentication due to a series of flaws in the composition of protocols that make up RSN."

These scenarios illustrate the importance of selecting an EAP-type that supports dynamic key exchange and client authentication (EAP-MD5 is therefore not recommended). 802.1x was originally conceived as being an asymmetric solution with only client authentication and not AP authentication. However, all popular EAP methods (TLS, TTLS, PEAP) use mutual authentication of both the client (supplicant) and access point (authenticator) and are therefore not vulnerable to this kind of attacks. TTLS and PEAP build on the notion of first establishing a TLS connection to the Authenticator (the so-called Inner Authentication) followed by the actual authentication (the Inner Authentication) for instance PAP, while EAP-TLS uses a TLS connection with mutual authentication of client and server (thus requiring setting up a PKI for client certificates).

By using EAP the 802.1x architecture is easily extensible and can in fact support multiple authentication mechanisms at the same time. Upcoming standards like WPA and 802.11i build upon this framework.

3.3 WPA (Wi-Fi Protected Access)

It is obvious from the previous section that WEP simply does not provide sufficient security in the wireless LAN for most enterprise needs. Within the IEEE there is work on a new standard, 802.11i (see below) which is intended to solve the security flaws inherent in WEP. However, since the IEEE802.11 working group has not yet ratified the 802.11i standard, the Wi-Fi[1] Alliance[2] announced a new security solution on October 31st, 2002, called Wi-Fi Protected Access (WPA) as a temporary replacement to the existing WEP standard.

For enterprise-class products, WPA specifies the following functions and technology components:

- *User authentication and dynamic encryption-key distribution*, two features missing from the original 802.11 standard. These are delivered through support for 802.1x and a choice of Extensible Authentication Protocol (EAP) algorithms. IEEE 802.1x specifies how EAP should be encapsulated in LAN frames. There are many EAP algorithms to choose from, depending on such factors as whether mutual authentication of both the user and the network is required. Some of the EAP flavors that support mutual authentication, albeit with different methods, are EAP-Transport Level Security (TLS), EAP-Tunneled TLS (TTLS) and Protected EAP (PEAP).
- *Encryption*. A Temporal Key Integrity Protocol (TKIP) engine handles dynamic key distribution. In WEP, there was one static encryption key that had to be manually entered. So changing the key across many wireless devices was unwieldy and therefore was often not employed, leaving data vulnerable. TKIP is an interim solution to the major portion of 802.11i that is not required in WPA yet.
- *Message Integrity Check (sometimes called 'Michael')*, a cryptographic checksum that is part of TKIP, to make sure packets have not been altered in transit.

[1] i. e. "Wireless Fidelity", cf. http://www.wi-fi.org/

[2] The Wi-Fi Alliance is a nonprofit organization which promotes the 802.11 wireless LAN standard. Apple and more than 180 other companies are members, and more than 450 products have been certified.

Within an enterprise wireless infrastructure, access points run 802.1x and TKIP. The back-end authentication server runs your choice of EAP algorithm. Client devices run 802.1x, TKIP and an EAP supplicant.

For SOHO infrastructures, WPA specifies the same level of encryption as enterprise-class products, but the authentication process is simplified to what has been termed a pre-shared key (PSK), but which is in practice a simple password mechanism.

For the purpose of repairing WEP, WPA uses selected components of 802.11i, mainly based on the Temporal Key Integrity Protocol (TKIP) [16], [17]:

- An **extended IV**,
  for 802.11, WEP encryption is optional. For WPA, encryption using TKIP is required. The TKIP protocol is part of the IEEE 802.11i encryption standard for wireless LANs. TKIP also provides per-packet key mixing. TKIP replaces WEP with a new encryption algorithm that is stronger than the WEP algorithm. WPA with TKIP, for example, uses 48-bit IVs (instead of the 24-bit WEP initialization vector) that greatly increase the number of possible shared keys and therefore significantly reduce IV reuse.
- **Re-keying**,
  In the 802.1x standard re-keying of unicast encryption keys is optional. Additionally, 802.11 and 802.1x provide no mechanism to change the global encryption key that is used for multicast and broadcast traffic. With WPA, re-keying of both unicast and global encryption keys is required. TKIP changes the unicast encryption key for every frame and each change is synchronized between the wireless client and the wireless AP. For the global encryption key, WPA includes a means for the wireless AP to advertise changes to the connected wireless clients.
- **Message integrity check (aka 'Michael')**
  Michael basically adds a kind of check digit to each message to detect if any data has been altered. With Michael WPA specifies a new algorithm that calculates an 8-byte message integrity code (MIC). The MIC is placed between the data portion of the IEEE 802.11 frame and the 4-byte integrity check value ICV[1]. The MIC field is encrypted together with the frame data and the ICV. Michael also provides replay protection. A new frame counter in the IEEE 802.11 frame is used to prevent replay attacks.

In case of larger networks WPA in addition provides for

- authentication by means of IEEE 802.1x and EAP, which make use of a radius server for user administration.

The main 802.11i elements such as

- secure hand off,
- secure de-authentication and
- improved encryption procedures (AES CCMP) are not supported.

The key distinctions between WPA and WEP are listed in Table 2:

	WPA	WEP
Encryption	Fixes main WEP flaws	Cracked
	RC4	RC4

[1] A mathematical algorithm which uses the plaintext message as input gives an ICV as output. The 802.11 standard specifies the use of the CRC-32 algorithm. CRCs (Cyclic Redundancy Check) had been primarily designed to detect random errors in transmitted messages. It is not cryptographically secure. Attackers can easily change packets and then change the ICVs appropriately.

	128-bit encryption / 64-bit authentication	40/104-bit keys
	48-bit IV	24-bit IV
	Dynamic session keys	Static WEP keys
	Automatic EAP key management	No key management
	Michael to check data integrity	CRC-32 integrity check
	Michael to check header integrity	No check for header integrity
	Replay attack secure by IV sequence	No replay attack protection
	DoS vulnerability	
Authentication	User authentication, utilizing 802.1x authentication with one oft the EAP implementations available (e.g. EAP-TLS[1], EAP-TTLS[2], PEAP[3]), and pre-shared key technology (PSK)	Flawed, WEP key itself was used for authentication

Table 2 WPA vs. WEP

Since WEP does not provide secure authentication or effective integrity checking, WPA was designed to upgrade WEP with respect to these functions by means of the Michael algorithm. Michael's functionality was implemented in software and intended to be integrated into existing WLAN devices. Michael's strength however is limited due to the restricted computing power of existing access points. Besides, Michael puts the entire WLAN adapter to sleep for the protection from brute force attacks for one minute, as soon as it discovers more than one potential attacker packets within one minute in order to discourage aggressors to send quickly successive forged packets. According to [17] "*This level of protection is much too weak to afford much benefit by itself, so TKIP complements Michael with counter-measures. The design goal of the counter-measures is to throttle the utility of forgery attempts, limiting the knowledge the attacker gains about the MIC key. If a TKIP implementation detects two failed forgeries in a second, the design assumes it is under active attack. In this case, the station deletes its keys, disassociates, waits for a minute, and then re-associates. While this disrupts communications, it is necessary to thwart active attack. The countermeasures algorithm thus limits the expected number of undetected forgeries such an active adversary might generate to about one per year per station.*"

According to Niels Ferguson, the designer of Michael, the WPA sleep does not open a substantially larger vulnerability than anyway already existed. In words of Ferguson[4]: "The Michael-countermeasures based DOS attack is not important for the following reason: there are other, more serious, DOS attacks in the basic 802.11 protocol. If we 'fix' the countermeasures to make the DOS attack more difficult the attacker will simply switch to one of the other DOS attack modes. The net gain is zero, so there is no reason not to use the countermeasures as specified."

3.4 802.11i / WPA version 2

To address the WLAN security gaps, the IEEE 802.11 Working Group instituted Task Group i to produce a security upgrade for the 802.11 standard now also known as WPA2. 802.11i is building the standard around 802.1x port-based authentication for user and device authentication. It includes two main developments: Wi-Fi Protected Access (WPA), i.e. WPA is a subset of 802.11i, and Robust Security Network (RSN). With

[1] Transport Layer Security
[2] Tunneled Transport Layer Security
[3] Protected Extensible Authentication Protocol
[4] Re: DOS attack on WPA 802.11?, The Mail Archive, http://www.mail-archive.com/cryptography@wasabisystems.com/msg03078.html

these developments 802.11i promises to plug the security holes in WEP and WPA as a self-contained IEEE standard. The current actual drawback of 802.11i is that it is not yet available as a ratified standard and will not be available presumably for at least another year.

The key distinctions between WPA and WEP are listed in **Error! Reference source not found.**:

	WPA2	WPA
Encryption	Enhanced encryption protocol: the Advanced Encryption Standard (AES-CCMP) with variable key sizes of 128-, 192- or 256-bits	RC4
	CCM[1] integrity check	Michael integrity check
	Hardware change necessary	Software upgrade possible
Authentication	Secure de-authentication and disassociation[2]	
	Secure IBSS[3] and secure fast handoff	
Compatibility	WPA2's mixed mode supports both WPA and WPA2.	WPA's mixed mode supports both WEP and WPA.

Table 3 WPA2 vs WPA

3.5 VPNs

IEEE 802.1x is often regarded as the key technology for the purpose of authenticating distributed WLAN access points. However, a study of the Colorado State University [18] comes to a different conclusion as the authors recognize security lacks which lead to "less than an industrial strength authentication solution". Thus, it was decided not to incorporate 802.1x into CSU's wireless network. Currently at Colorado State University, the preferred means of achieving privacy and authentication on wireless LANs is through a virtual private network (VPN).

Especially for sites with the highest level of security concerns it is recommended to use Internet Protocol Security (IPSec[4]), as available with most VPN solutions.

VPN technology can provide industrial strength authentication and encryption. As 802.1x defines layer 2 authentication, a VPN based on IPsec acts as a layer 3 tunnelling protocol. It should be noted that a layer 2 protocol like 802.1x authenticates the user before he gains network access and therefore prevents attacks against the network infrastructure. IPsec in contrast does not protect the link layer. IPsec was originally developed in the framework of IPv6 and now presents the standard for encrypted communication over the Internet. With its high-level security encryption and per-packet authentication IPsec meets the highest security requirements; and it is suitable for the protection of end-to-end communication.

A VPN is a secure virtual network which is built over using an unsecure network connection. For getting access authentication is required. Subsequent data transfer is done encrypted. Due to the fact that the

[1] Counter with CBC (Cipher Block Chaining) MAC (Message Authentication Code). CCM provides both authentication and encryption in a single key. CCM has been submitted to the National Institute of Standards and Technology (NIST) for consideration as a standard mode for use with the Advanced Encryption Standard (AES).
[2] Disassociation is the process of terminating the association of a client with a given access point in the WLAN whereas authentication denotes the process of verifying the credentials of a client desiring to join a WLAN.
[3] The Independent Basic Service Set (IBSS) enables security between client workstations operating in ad hoc (peer-to-peer) mode.
[4] http://www.ietf.org/html.charters/ipsec-charter.html

communication is based on a secure virtual network a potential attacker with access to the link layer cannot eavesdrop or forge the data unnoticed.

In principle, the VPN consists of a gateway giving way to the local network and the WLAN clients. The access point(s) in this scenario only presents a component of the physical medium and does not make up a constituent part of the VPN. Accordingly, access points are not directly connected to the local network but only via the VPN gateway; which acts as a firewall giving access only to VPN secured traffic. Figure 2depicts how secure communication is provided by VPN tunnels, which connect the WLAN clients with the trusted VPN gateway residing in the LAN.

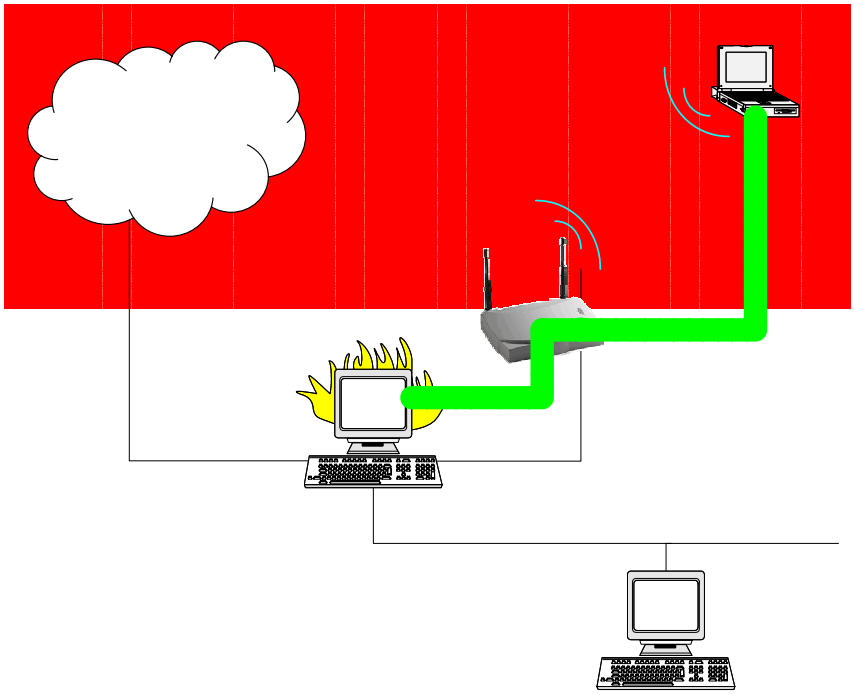

Figure 2 VPN for Secure WLAN Access

3.5.1 Advantages of VPN-based Solutions

VPN solutions typically operate in the ISAKMP/IKE framework and therefore provide the following advantages over other access control schemes:

- Derive strong cryptographic material by:
  - Using Diffie-Hellman exchange;
  - Providing a master SA (IKE SA) that protects actual SA negotiations (child SAs).

Access technology independent;

3.5.2 Shortcomings of existing Access Control Mechanisms (EAP-based)

- Protocols like EAP-TLS and PEAP establish a TLS session that protects the subsequent authentication exchange between the client and the authentication server. This introduces the overhead of interacting with a PKI infrastructure (and the associated scalability issues, especially in the case of EAP-TLS). Moreover the client is exposed to computational DoS attacks (by malicious nodes posing as authentication servers and providing faked certificates for validation) while the

authentication server itself may be victim of connection depletion attacks (by malicious nodes flooding it with fake authentication requests);

- EAP-SKE does not use TLS and therefore eliminates the associated overhead. The message exchange is instead authenticated using a HMAC computed with a symmetric key the supplicant and the access point share. In this case, however, perfect forward secrecy is not supported: once the master key is compromised, all deriving session keys are compromised as well and past traffic may be decrypted.
- The key management schemes supported by the current proposals are rather inflexible: they do not provide support for negotiating a number of parameters such as transforms, key strength and key lifetime between the peers.
- The "unicast WEP key" derived by the client and the authentication server upon successful client authentication is delivered by the authentication server to the access point possibly through a number of intermediate proxies. (The unicast WEP key is further used by the access point to encrypt the "broadcast WEP key" which is the key the traffic will be encrypted with over the wireless link). The transfer of a shared key otherwise than through an end-to-end secure channel is however insecure. On the other hand, the attempt to set up such end-to-end secure channels between access points and authentication servers easily runs into scalability problems, especially in an inter-domain scenario.
- Finally, a fast handover mechanism would is missing and consequently the authentication process must be repeated each time the supplicant attaches to another access point. This is of course not the case when the link layer encryption key is the same throughout the entire wireless domain. Such a scenario however is not taken into consideration as it contradicts the high level requirement mentioned under point 3.

3.5.3 EAP Integration with VPN-based Solutions

EAP seems to receive wider and wider acceptance as a generic link access protocol. IKEv2 has also introduced EAP as an alternative authentication mechanism to the already existent certificate- and shared secret-based solutions. Similar to the TLS based versions of EAP, the IKEv2's IKE SA protects the actual EAP exchange.

3.5.4 WAVEsec – A Linux VPN

WAVEsec[1] is a way to authenticate and encrypt 802.11 traffic on its vulnerable hop between a WLAN client and a wired gateway using either KAME or Linux FreeS/WAN[2] IPSec implementations. The novel thing about WAVEsec is how it arranges the trust required between the client notebook and the WAVEsec gateway. WAVEsec does it by exchanging public keys during the DHCP address assignment. The client can therefore be completely configured just by plugging an 802.11 interface in.

The client provides its forward hostname and public key in a DHCP request. The DHCP server then inserts both into the DNS server for the reverse zone (mapping of IP addresses to names) using Dynamic DNS update. The DHCP server responds giving the client needed information via three new DHCP options:

- a WAVEsec gateway address
- the gateway's mode (inline or appendix)
- the gateway's public key

[1] http://www.wavesec.org/
[2] FreeS/WAN (http://www.freeswan.org/) is an opensource project and software package implementing IPsec / IKE VPNs for Linux machines.

To support this functionality the DHCP server must be capable of taking a custom DHCP option containing a raw RSA key in DNS format and installing it into a DNS server using dynamic DNS updates. To set up a WAVEsec client, the following is required:

- to run Linux
- to install an extended DHCP client (*dhclient*), and configure it for WAVEsec
- to install and configure Linux FreeS/WAN

There are 5 logical entities involved in a WAVEsec scenario. They are:

- the WLAN client
- the WAVEsec access point
- the DHCP server
- the DNS server
- the Internet connected router

For SOHO networks and other small networks, all server- and routing functions might be combined into a single machine.

3.6 Simple VPN-based Access Control Procedure

This section is devoted to defining an access control protocol that aims at achieving a high level of flexibility, scalability and robustness. This includes in principle the ability to operate over various access technologies, to negotiate strong security schemes, to operate in an inter-domain environment and to resist to a whole range of attacks. We will start by analysing existent approaches in order to identify the best way these high level requirements can be achieved.

3.6.1 Evaluation of Existing Technologies

The following shortcomings have been observed at the EAP-based access control mechanisms:

- Protocols like EAP-TLS and PEAP establish a TLS session that protects the subsequent authentication exchange between the client and the authentication server. This introduces the overhead of interacting with a PKI infrastructure (and the associated scalability issues, especially in the case of EAP-TLS). Moreover the client is exposed to computational DoS attacks (by malicious nodes posing as authentication servers and providing faked certificates for validation) while the authentication server itself may be victim of connection depletion attacks (by malicious nodes flooding it with fake authentication requests);
- EAP-SKE does not use TLS and therefore eliminates the associated overhead. The message exchange is instead authenticated using a HMAC computed with a symmetric key the supplicant and the access point share. In this case, however, perfect forward secrecy is not supported: once the master key is compromised, all deriving session keys are compromised as well and past traffic may be decrypted.
- The key management schemes supported by the current proposals are rather inflexible: they do not provide support for negotiating a number of parameters such as transforms, key strength and key lifetime between the peers.
- The “unicast WEP key” derived by the client and the authentication server upon successful client authentication is delivered by the authentication server to the access point possibly through a number of intermediate proxies. (The unicast WEP key is further used by the access point to encrypt the

"broadcast WEP key" which is the key the traffic will be encrypted with over the wireless link). However, the transfer of a shared key through anything other than an end-to-end secure channel is insecure. On the other hand, the attempt to set up such end-to-end secure channels between access points and authentication servers easily runs into scalability problems, especially in an inter-domain scenario.

- The scalability itself comes into question, as the authentication process must be repeated each time the supplicant attaches to a different access point. This is of course not the case when the link layer encryption key is the same throughout the entire wireless domain. Such a scenario however is not taken into consideration as it contradicts the high level requirement mentioned under point 3. Repeating the authentication procedure each time

On the other hand, VPN solutions, which typically operate in the ISAKMP/IKE framework, provide the following advantages:

- Derive strong cryptographic material by:
  - Employing Diffie-Hellman exchanges;
  - Providing a master SA (IKE SA) that protects actual SA negotiations (child SAs).
- Access technology independent;

Finally, it must be remarked that the AAA infrastructures receive growing acceptance as an inter-domain authentication and authorization framework.

3.6.2 Defining the Solution

It appears therefore that extending the ISAKMP with an AAA-based authentication scheme is likely to yield the optimal solution. The protocol presented here describes one possible way to achieve this. It reuses ISAKMP/IKE and runs in two phases. The first phase is a typical AAA authentication request/reply that involves the mobile user, the attendant (NAS/VPN concentrator) and the user's home AAA server entity (AAAH). The AAA exchange is also used to perform an authenticated exchange of IKE phase 1 messages in the form of an shortened aggressive mode exchange.

The second phase consists of a shortened IKE quick mode exchange (identical to the QM exchange in IKEv2) that take place between the attendant and the mobile terminal. The SA negotiated during this phase (typically ESP in tunnel mode) will protect all subsequent data traffic.

The first phase exchange must take into account that because the nomadic node and the attendant do not share any pre-existent trust relationship, the impact of a number of attacks is exacerbated by the easiness an attacker can access the transmission medium. The proposed protocol guards against the following potential attacks:

- Computational DoS attacks – computationally expensive operations (asymmetric cryptographic operations including public Diffie-Hellman key generation and shared Diffie-Hellman key derivation) must be avoided as long as the nomadic node is not yet authenticated. Initial authentication would ideally be based on shared secrets;
- Resource depletion attacks – intermediate entities must avoid creating states as long as the nomadic node is not yet authenticated.
- Reply protection – replied control messages may have significant impact because they may:
  - Produce alteration of the states in intermediate entities that are relevant to the victim in such a way that the victim cannot receive the service;
  - Facilitate computational DoS attacks because the replied messages pass the first barrier of authentication.

Man-in-the-middle attacks – intermediate entities substitute the security information (such as public keys) with the purpose of breaching the security mechanisms or stealing the secure connection set up on behalf of a legitimate nomadic node. Such attacks can be mounted on the access link by malicious visitors that poison the neighbour caches as well as by malicious AAA entities located on the bath between the access domain and home domain.

3.6.3 Terminology

Nomadic node — Nomadic node denotes an IP host fitted with a WLAN network adapter, which attaches to different access points located in administrative domains (subsequently denoted as "visited domains") that may be different from the administrative domain the "nomadic node" originates (subsequently denoted as "home domain").

Attendant — Attendant denotes an IP router that controls the access of nomadic nodes to the network. Attendants terminate the IPsec tunnels that protect the nomadic nodes' traffic over the wireless link, and filters out all unauthorized traffic. They also implement the AAA client function.

Access protocol (AP) — The access protocol defines the control message exchange between the nomadic node and the attendant. It essentially exchanges AVP[1]-packed data that is used to build the AAA commands, which are further transferred between the involved AAA entities.

3.6.4 General Description

Figure 3 illustrates the message exchange. The local AAA server (AAAL) has not been depicted, its functionality is reduced to forwarding AAA commands between the attendant and the home AAA server (AAAH).

It is assumed that the nomadic node and the AAAH share a secret denoted as SK_{NH}. For location detection the nomadic node relies on the periodic router advertisement messages sent by the attendant. Additionally the nomadic node may explicitly solicit router advertisements.

[1] Attribute value pair

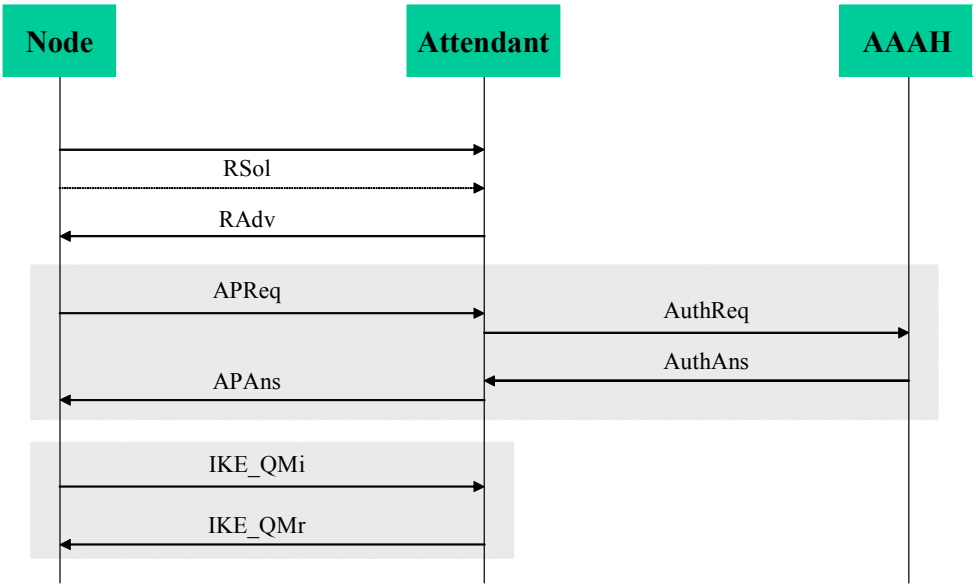

Figure 3 The message exchange

The protocol has the following properties:

- It substantially relies on ISAKMP/IKE exchanges, which enables the nomadic node and the attendant to negotiate strong security schemes and supports fast re-keying;
- No state is required to be maintained on the attendant until the authentication request is validated by the appropriate AAAH. This property has however a limited impact because in a typical configuration the attendant plays the role of an AAA client, and the Diameter protocol requires clients to maintain state. Nevertheless, the proposed protocol does not pose additional burden, should a resource depletion attacks occur;
- An end-to-end security association between the attendant and the AAAH is not required, while the protocol ensures robustness against man-in-the-middle attacks from malicious AAA entities located on the attendant-AAAH chain;

Timestamps are used in order to protect against replay attacks. A mechanism similar to the one used in Mobile IPv4 is used to synchronize the timers on the nomadic node and AAAH.

For details of the message contents, please refer to the appendix.

3.7 Web-based Authentication and Access Control

One relatively straightforward way of authenticating WLAN users is via a web-based authentication system. It does not require any special functionality to be present at the APs or client devices. An access control device (generally a router) is located between the fixed network and the WLAN. This device will deny any traffic from/to unauthorised devices. For a device to be allowed to communicate over the fixed network, the user must be authenticated first. User authentication is achieved through a simple web interface.

When a device enters the WLAN, it receives an initial IPv6 address via DHCPv6; but this address is not allowed to send/receive traffic over the fixed network. The user of the device must launch their web browser so that authentication can take place. The access control device redirects HTTP requests to the authentication web server so that the user does not have to know the URL for the authentication service. The authentication service returns a web page to the user consisting of a form for the user to enter his or her credentials (e.g. username and password). Based on these credentials the authentication server will either grant (by modifying

the access control list on the access control device) or deny access to the IP address assigned to the user device.

The de-registration of a user may be performed via a 'logout' button on a pop-up window or more suitably (the user may not log-off or have pop-up windows disabled) via the timeout of ICMP echo requests sent by the access control device.

The level of security for this authentication mechanism depends on the particular implementation. User credentials should never be sent as clear text via HTTP. Instead, HTTPS (secure HTTP) should be used between the web browser and the authentication server. The authentication server could employ the use of RADIUS or LDAP services at the backend.

3.7.1 Example: IST Mobile and Wireless Summit

One recent deployment example of web-based access control is that of the 2003 IST Mobile and Wireless Communications Summit in Aveiro, Portugal. To prevent unauthorised access, each user had to enter login credentials (username and password) to an authentication web page that users were directed to automatically upon starting their web browser. Conference delegates had to obtain 'access cards' from the registration desk. Each card contained a username and password that was valid for 5 hours from the time of the first login. In this way, user authentication was achieved visually, by staff at the registration desk who checked that the requesting user was registered for the conference. Once the 5 hour time limit had expired, delegates had to return to the registration desk to obtain a new (with different username and password) access card.

This method worked well up to a point but it was rather inflexible, especially the 5 hour limit. This meant that users had to obtain an access card at least twice for each day of the conference. This was especially irritating when the time limit expired while in the middle user checking email or downloading papers, presentations etc. All application connections were severed when the time limit expired and could not be re-established until the user logged in with a new access card. Thus, it was often many minutes before a user was able obtain a new access card from the registration desk and re-establish his application connections. Soon, many delegates were obtaining access cards in advance of their current one expiring to minimise the nuisance value. However, it was still not possible to move seamlessly from an existing user account to a new one without breaking existing application connections.

In hindsight, the time limit value could have been chosen in a more thoughtful way. For example, a 24-hour limit (so users only had to obtain a card once each morning) or even no limit at all (i.e. the duration of the conference).

3.8 Lancaster University Access Control Architecture

The Mobile IPv6 testbed at Lancaster University and the city of Lancaster can employ a novel access control architecture designed specifically for an IPv6-based wireless architecture. Since the deployment of the wireless components of the testbed are in public places, there is a danger of unauthorised users gaining access to the network. Therefore, it is important to restrict access to the network only to authorised systems and users. A secure user authentication and authorisation and a reliable access control mechanism is vital – especially for the wireless LAN segments.

3.8.1 Requirements

The requirements for the access control architecture are based primarily on the objectives of the Mobile IPv6 testbed:

Mobility – Without doubt mobility is the focal aspect of the research within the Mobile IPv6 Testbed. It is therefore crucial to design the access control architecture for a highly mobile network environment, where users frequently roam between wireless cells and networks.

Security – As the wireless network infrastructure is publicly available throughout large parts of the city centre, a secure access control mechanism is required to restrict services to authorised users only. In addition, the access control architecture must protect the network against internal and external security threats (for example, denial-of-service attacks).

Flexibility – One of the key requirements is flexibility. As we cannot foresee yet what services will be developed within the course of the Mobile IPv6 Testbed research, we require a maximum flexibility for the access control approach (for example, to support different granularities of control and a broad spectrum of access policies).

Extensibility – The access control architecture must be extensible to enable the integration of additional functionality or interaction with value-added services (such as accounting, QoS, etc.) in the future.

Transparency – The access control approach must be fully transparent to correspondent nodes (i.e., hosts to which a mobile node is conversing) external to our mobile network in order to ensure full interoperability with the standard Internet.

Usability – User access to the public network infrastructure should be as easy as possible (i.e., simple installation of software at the beginning and continued ease of use).

Scalability – The size of a public network spanning the city centre of Lancaster demands a scalable access control architecture in terms of number of users and end-terminals.

Manageability – In order to facilitate the manageability of user accounts and access policies, a comprehensive management system is required. The access control architecture should also be fairly universal (i.e., independent from the underlying technologies) to provide a uniform solution across the whole network (avoiding the need for administration of multiple systems).

3.8.2 Network Infrastructure

The Mobile IPv6 Testbed is constructed along the wireless overlay network concept [5], [6] whereby a number of different wireless technologies (such as HSCSD, GPRS and Bluetooth) are used in combination with IEEE 802.11 wireless LANs, in order to provide the coverage and network performance required for future network services. This approach was chosen for two reasons. Firstly, it ensures that mobile users maintain access to network resources wherever and whenever possible, as the most appropriate connection can be chosen at any given time. Secondly, it is likely to more accurately emulate the network topology of future public access wireless networks, thus providing us with a more realistic test environment.

Although many of the wireless technologies that will likely make up future overlay networks already have some form of access control (for example, WEP in 802.11 [7] or RLC/MAC in GPRS [8]), those access control mechanisms are often quite distinct from each other. As the Testbed is formed from a range of such layer 2 network technologies, the access control mechanism must therefore be independent of those underlying network types. The Testbed addresses this problem by adopting a layer 3 approach to access control.

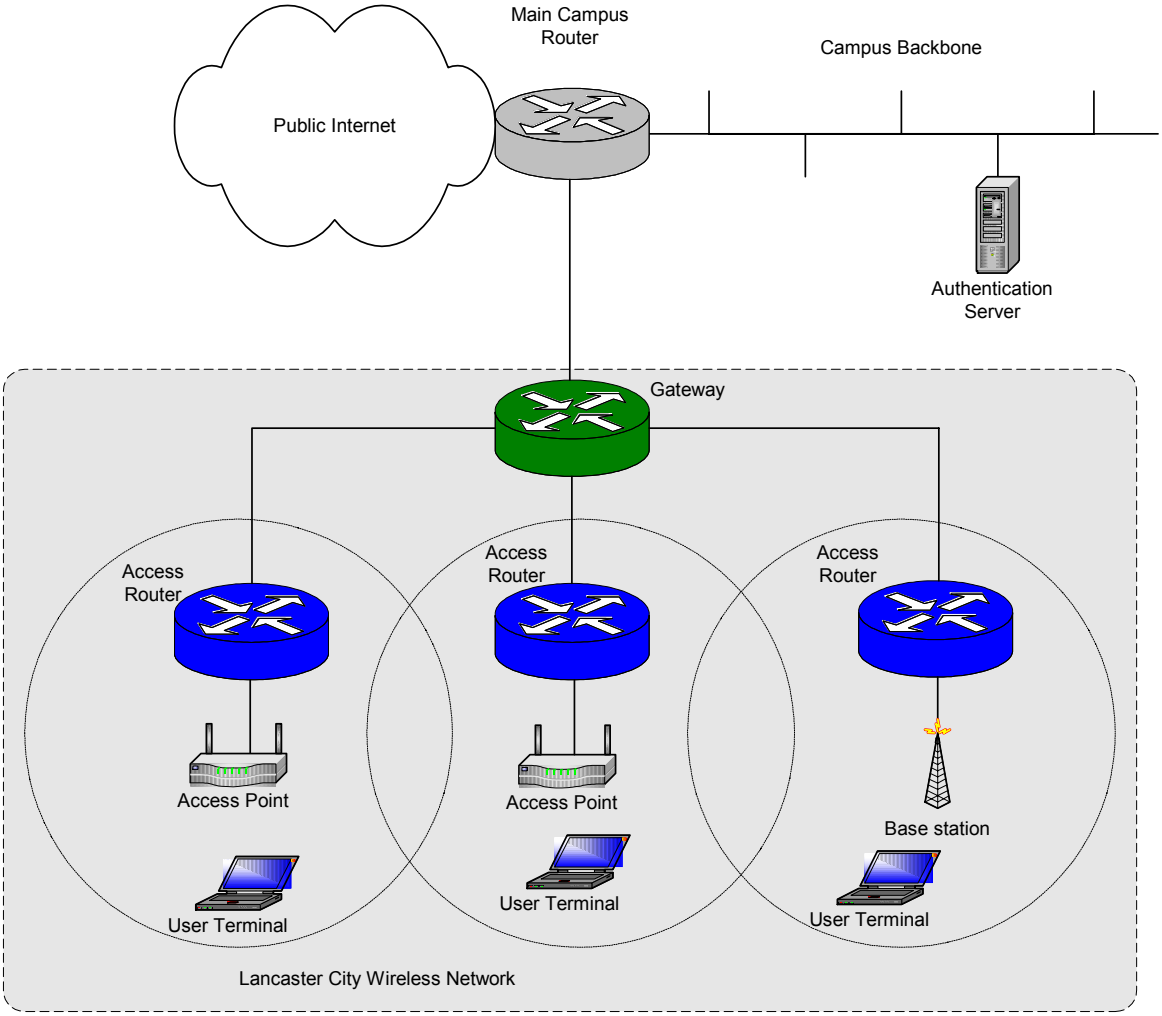

Figure 4 Network Infrastructure (simplified)

The logical network infrastructure for our wireless network is illustrated in Figure 4. As can be seen from the diagram, the network is formed from a number of over-lapping wireless cells, consisting of a variety of technology 'flavours'. Adjacent cells of the same flavour can be merged together using layer two bridging in order to form a cell with a larger footprint. Although bridging is an effective means to interconnect a small number of homogenous cells, it is well understood that such architectures do not scale. In addition, when the case of a mobile device crossing between different flavours of network is considered, it becomes apparent that bridging provides little support. In order to address these issues, the Testbed network places layer 3 administrative boundaries between cells of different flavours, and optionally between cells of the same flavour. These boundaries separate logical areas of administrative control, called districts.

Each district within the Testbed consists of one IPv6 (sub) network. As can be seen from Figure 1, each district within the Testbed is served by an IPv6 access router. This access router is directly responsible for the management of that district, such as IPv6 routing, access control and billing. Access routers are interconnected and linked back to the campus backbone via a wired infrastructure, using SDH, DSL or over point-to-point microwave links.

Scalability

The use of individually managed IPv6 based cells gives many scalability advantages. Primarily, as there is a vast expanse of available IPv6 address space, it is perfectly feasible to uniquely address each cell within the

Testbed as a separate IPv6 network, and still maintain scalability. This would be difficult to achieve with IPv4, even through the use of network address translators (NATs). Additionally, the auto-configuration support offered by IPv6 negates the need for services with higher administration costs, such as DHCP.

The architecture also provides scalability for larger networks. As the access control is enforced by the access router (at the first hop of the wireless network), this distributes the load of access control throughout the network.

Finally, the fact that the wireless cells are routed rather than bridged results in less broadcast traffic on those cells, thus improving network utilisation. In turn, this also improves the security of the network, as it makes it far more difficult for users to snoop packets or masquerade as other network nodes (in the case of a fully routed network, both the attacker and target must be co-located within the same cell, making the attacker much easier to discover and track).

Host Mobility

Although there are clearly many advantages to be gained from having a fully routed network, it does add more complexity to the mobility management subsystem of the network. Consider the case of a mobile device roaming between various flavours of network within the Testbed. As the mobile node crosses an administrative boundary between districts, it sees a change in its IPv6 point of attachment. We envision that these administrative boundaries will be highly commonplace, separating areas with differing access control settings. They could be as often as different offices/ laboratories within a building, and different shops within a city. It is therefore vital that the transition between districts is handled smoothly and quickly by the infrastructure. We use Mobile IPv6 [9] to enable this roaming between cells, which provides us with the necessary location independence and transparency, a distributed mobility management architecture and reasonable handoff performance[1]. However, this adds an additional requirement onto the access control architecture – any authentication or access control mechanisms must not interfere with the performance of the handoff between districts.

3.8.3 Access Control Mechanism

The access control mechanism proposed for the Mobile IPv6 testbed is based on the principles of packet marking and packet filtering. Data packets are tagged on the client terminal through an extension to the network stack before they leave the node. Based on presence and credentials associated with the packet marking, access to the trusted network (i.e., public Internet or value-added service networks) is granted or denied.

The key components of our access control architecture are described here. Figure 4 illustrates how they are situated within our network infrastructure.

- The *Authentication Server* (AS) is responsible for the authentication and authorisation of clients on the access network. Upon successful authentication and authorisation of a user, the AS issues a limited lifetime access token to the user.
- User *end-terminals* (i.e., handheld devices, laptops, etc.) request the authorisation of the node on behalf of the current user and perform the packet marking for outbound traffic. A valid access token is obtained from the AS upon successful authorisation of the user. The access token, in turn, provides the basis for the packet marking.
- *Access Routers* (ARs) control the access to the protected network. They block traffic originating from or sent to unauthorised end-terminals based on network-level packet filtering. Co-locating the ARs directly with the base stations enables highly flexible access control close to the user (thus minimizing the area which can be targeted by an unauthorized attacker).

[1] Handoff times with 'normal' Mobile IPv6 are not good enough for real-time applications. Further mechanisms to support faster handoffs for real-time applications will be studied in D4.1.3

- The *Gateway* connects the access network with the public Internet or a private Intranet (i.e., Campus network). It is concerned with external security threats from arbitrary nodes on the public network and is an extension of the firewall concept.

Account Creation

In order to access a publicly accessible network guided with our access control system, a user's end-terminal requires our Mobile IPv6 stack extension. The user downloads and installs the extension at account setup time[1]. The user's secret credentials (i.e., username and password) are created and registered with the authentication server. The user is also assigned a group, which defines the level of service granted to the user. For example, groups have individual access profiles in terms of which cells they can access (i.e., at what times, and how long or frequently).

In future, we plan to support service differentiation in terms of QoS (i.e., different priority levels, data rate limitations, payload volume restrictions) based on the group affiliation.

Session Initialisation

Before a user can access the network, the end-terminal requires a valid access token in order to tag packets for transit via the access routers. As a result, the user will be prompted to enter his username and password during session initialisation (e.g. when the device is turned on in a cell or at initial network entrance). This process occurs only once per login session as the credentials are cached for the duration of a session.

User Authentication

User authentication is carried out between the user's end terminal (e.g. PDA or laptop) and the authentication server. The client software takes care of authentication of the user currently logged in. The authentication request sent from an end terminal to the authentication server includes the user's username and password, the node's MAC Address and IPv6 Address, and a secret session key. While the username and password are required to authenticate the user, the MAC and IP addresses are needed to authorise the client node on the access routers. As further discussed below, the session key is needed for the encryption of the access tokens, to avoid address spoofing attacks with the network.

In order to prohibit malicious users from spoofing the secret credentials of other users or a session key, the authentication request message must be encrypted. We use public key encryption based on the RSA algorithm [27] and the standard IPsec encryption header [28] in order to avoid the need for a secure key exchange mechanism and a special protocol extension. Public key encryption is advantageous as the client must know only the public key of the authentication server (which can be statically configured) and not vice versa.

Token Generation

Access tokens are the secret credentials that grant packets from authorised end terminals access to the protected network. The tokens are issued to particular users upon successful authentication and authorisation. However, passing the access tokens in clear text to the client would allow malicious users in the same cell to snoop valid tokens. Therefore, to secure the access control mechanism even against MAC address spoofing, the access token requires encryption. The shared session key passed within the authentication request is used for the encryption of the authentication response message.

In order to avoid brute force attacks on access tokens, we chose to restrict the lifetime of the access tokens to a configurable time interval, referred to as the expiration time of a token. Beyond this interval the extended protocol stack must refresh the node's authorisation based on the cached user credentials to request a new token.

[1] It should be noted here that our extension does not impact the "normal" use of the Mobile IPv6 stack in conventional networks.

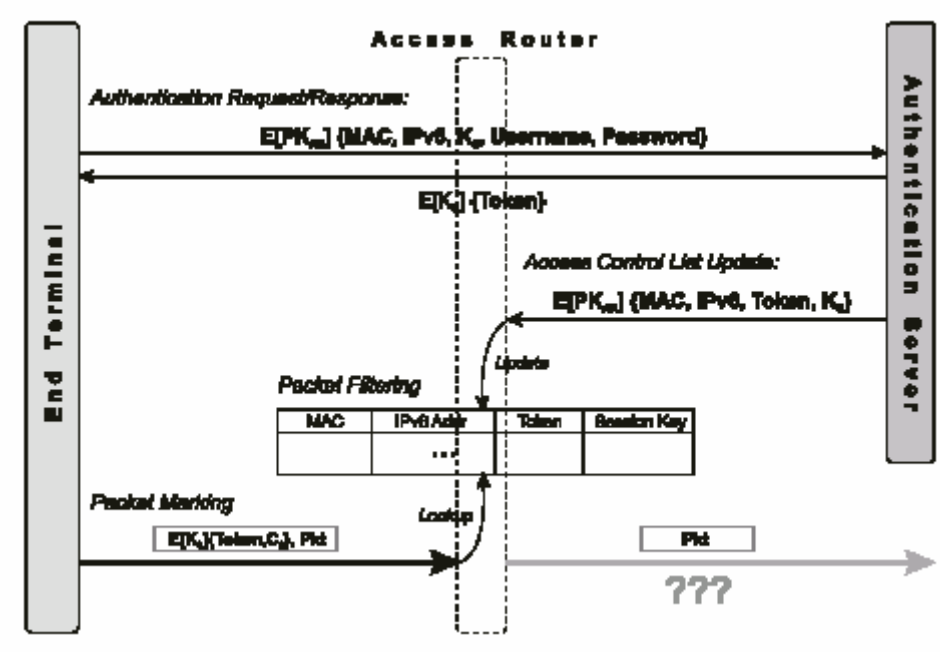

Figure 5 Access Control Protocol

Since the access token is simply a pseudo-random value that is large enough to make it hard to guess or discover by a brute force search within the lifetime of the token, the expiration time must reflect the size of the access token[1]. The refresh time of the authentication protocol must be sufficiently smaller than the expiration time[2].

The main advantage of those short-lived access tokens is that they provide extra security and robustness. The fact that they change so frequently make them hard to crack.

Packet Marking

End terminals use packet marking as a technique to indicate authorised packets to the access routers. When the Mobile IPv6 stack forwards a packet, it includes our access control extension within the Hop-by-Hop Option extension header of the IPv6 packet as illustrated in Figure 6. This header contains the access token and a checksum besides usual housekeeping information (i.e., protocol version and encryption type). The token and checksum are both encrypted using the session key associated with the access token. The checksum is required as a measure against replay attacks. It prevents a potential attacker from simply snooping the extension header and adding it to their own data.

Since the extension header must be attached to all data packets, a very lightweight encryption mechanism is required. We therefore use a symmetric cipher in order to avoid the performance overhead of asymmetric cryptographic algorithms.

[1] Since we encrypt the 32-bit access token together with a 96-bit checksum, we currently use a 10 minute expiration time.

[2] We suggest a refresh time that equals to: Trefres = min{Texpiration – 2 X avg(Tauthentication), 2/3 X Texpiration}

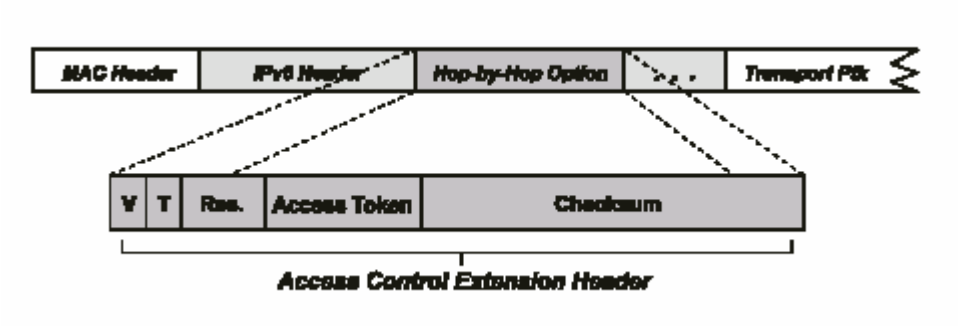

Figure 6 Access Control Extension Header

Packet Filtering

The access control mechanism described so far is based on network-level packet filtering. Access routers check packets sent to and from the end terminals for authorisation. While packets sent to the wireless network must have the destination address of an authorised end terminal, packets sent from a client node must carry a valid access token. For this, each access router maintains an access control list (ACL) that accommodates all the filter information required to identify 'authorised' packets, namely the MAC address, IPv6 address, access token, and session key for each authorised end terminal. The access routers receive this information from the authentication server when a user of the cell successfully authenticates with the network.

When a packet is received from the wireless network, the access router looks up the MAC address in the ACL. If an entry for the end-terminal exists, the access router verifies the IPv6 source address. In the case of a match, it decrypts the access token and checksum using the session key (held within the ACL) and validates its content against the ACL. When successful, the Hop-by-Hop Option containing the access control extension header is stripped off and the packet is passed on. Packets that fail any of those tests (e.g. due to an unknown MAC address, a wrong IPv6 address match, or an invalid or expired token) are dropped. One exception to this rule is that when a client is first seen in a cell, it is allowed to contact certain well-known IPv6 addresses; this allows nodes to initially communicate with the authentication server.

To quickly recover from missing ACL entries due to packet loss or a router crash, the access router indicates failure immediately, such that the client can re-authenticate right away. To prevent malicious users from trying to gain unauthorised access to the network, we plan to add a mechanism to black list malicious users who repeatedly send packets with invalid IP addresses or access tokens (e.g. example, through link-layer access restriction).

The soft-state authentication protocol facilitates fine-grained access control with respect to time and location. It allows end-terminals to be restricted to certain districts based on time. Furthermore, the soft-state approach eases the withdrawal of access privileges. For example, a user who is caught using the network in an inappropriate way or who runs out of online time credit can be denied access to the network by simply refusing further access tokens refreshes.

Roaming Support

In networks such as the Testbed, support for roaming users that frequently move between microcellular networks is crucial. From a network-level point of view, roaming support is provided through the Mobile IPv6 protocol. With respect to our access control architecture, we therefore focus on minimising the impact of access control on handoff performance. When a mobile node moves into a new network cell, it acquires a new care-of-address (CoA) to reflect its new physical network location[1]. As a result of the network handoff, the network access will also be controlled by a different access router, which may have no knowledge of

[1] This can be achieved either through DHCPv6 or the autoconfiguration mechanisms of IPv6.

previous authorisations for the mobile node. Unfortunately, it could take up to several minutes (i.e., until the next authentication refresh is carried out) before the mobile node would obtain access to the network again. Since service disruptions of this order are clearly not acceptable for networks such as the Mobile IPv6 Testbed, we introduced three special measures:

1. Mobile nodes immediately initiate a fresh authentication cycle for the node's new IPv6 address immediately after a network handoff.
2. The authentication server sends the periodic ACL updates not only to the client's access router, but also to the neighbouring access routers in order to 'preheat' their access control lists with authorisations for potential roaming clients. Note that with the emergence of the context transfer protocol [29] currently being discussed within the IETF, we consider using this 'proactive' means to transfer ACL state from previous to new access routers.
3. Access routers grant a short reprieve time for roaming nodes entering a cell, before they block traffic from the node. This technique preserves safety by granting access to packets based on a node's previous authorisation. Due to the preheating of the neighbouring access routers, a node's new access router will already have an entry in its access control list when the node moves into its coverage area. This allows packets with a valid MAC address and access token to pass for the period of the reprieve time. However, if the router does not receive a fresh access list update for the node's new IPv6 address before the reprieve time expires, traffic will be blocked.

These extensions have the advantage that they do not interfere with or slow down network-level handoffs. The initial user authentication required when entering a new district is simply delayed (i.e. carried out in the background) to avoid extra latency. The reprieve time must be chosen carefully. On the one hand, the interval should be minimal as it gives provisional access to users based on their previous authorisation while, on the other hand, it must be long enough to complete a whole authorisation cycle[1].

Core Network Protection

The design of our access control architecture assumes that the core network can be trusted. This assumption seems reasonable, since the access routers and the authentication server typically belong to the same administrative domain. In our network, for example, the access routers are physically protected by locked cabinets in buildings not open to the general public, and the physical links from the access routers back to the campus network are hard to intercept. However, in case we identify fraudulent misuse within the core network or in network segments that are not trusted, we fall back to use end-to-end encryption for communication between the authentication server and the access routers.

For this, standard public key encryption as supported by IPsec is recommended. In addition, we can also use the gateway router as a firewall for the core network. For security reasons and to avoid denial-of-service attacks, it can block remote traffic (sourced from the public network) directly sent to the access routers. To minimise the risk of denial-of-service attacks on the clients, the gateway could potentially rate-control transmissions to end-terminals.

Enhanced Security

In cases where users demand a high level of security (e.g. full privacy), the architecture supports an additional layer of protection based on full encryption of the payload on the wireless link (i.e., between the access router and client device). This offers an alternative to the IEEE 802.11 wired equivalent privacy (WEP) protocol, which has been shown to be vulnerable to attack [30]. We plan to allow the user to freely choose the level of security depending on the network use and the end-terminal at hand, as full payload encryption can be very heavyweight for low-performance mobile devices.

Finally, it is worth noting that standard IPsec authentication and encryption are entirely complementary to our access control architecture. They can be used in addition to achieve secure end-to-end communication.

[1] We recommend a reprieve time of approximately 2-5 seconds depending on the network performance and authentication server.

3.8.4 Implementation

This section outlines the implementation of the key components of our access control architecture for the wireless network around Lancaster. Due to the lack of space, we provide only a brief description here.

Client Software

According to our architecture the client software is responsible for user authentication and packet marking. User authentication is performed by a system service executed on the end-terminal. In order to gain access to the network, the user must first provide its username and password to the terminal. The credentials are then stored locally such that the actual authentication (and periodic re-authentication) with the authentication server can be performed by the service without user intervention. The authentication protocol used for client authentication with the AS is based on a lightweight request/response protocol. UDP is used for transport. Standard IPsec encryption is applied for end-to-end encryption of the authentication request[1]. The clients use RSA public-key encryption based on the public key of the authentication server (which can be pre-configured at the client to avoid the need for a key distribution service) to encrypt the authentication request.

As described earlier, the authentication request includes a new session key for the authentication server to establish a secure communication channel back to the client. For the encryption of the authentication response, we use symmetric encryption based on the shared session key. This is accomplished via the tiny encryption algorithm (TEA) [31].

In response to an authentication request, the server replies either with a new access token for the client or an error message. In the case of success, the client service decrypts the authentication response using the current session key and passes the token to the protocol stack for the packet marking.

As highlighted earlier, our extended protocol stack also performs the packet marking, including the most recent access token into every packet (see Figure 3) to indicate the client's access credentials to the access routers. To prevent MAC address spoofing and replay attacks based on a spied access token, we encrypt the token along with a packet checksum (using the shared session key). The 96-bit checksum is computed from frequently changing protocol fields of the IPv6 header (namely the source and destination address, flow id, and, payload length), the transport protocol header (namely the source and destination port, and checksum), and limited data of the payload using MD5 [32]. In order to minimise the latency due to encryption of the access credentials, we use the fast block cipher TEA. The lightweight algorithm has no known cryptanalysis and is claimed to be at least as secure as the well-known IDEA cipher.

We have chosen to use the Mobile IPv6 protocol stack as a starting point to add the extra functionality required for packet marking because of the experience we have gained with this protocol stack in recent years. In particular, we support the protocol stacks for Microsoft Windows 2000/XP and Linux, since we have implemented both stacks within previous projects at Lancaster.

Authentication Server

The authentication server runs a user-level application or service responsible for managing the user accounts, and the authentication and authorisation of the users and their terminals.

Upon receipt of an authentication request (on the well known server port), the authentication application tries to authenticate the user. On success, it sends the authentication response message including the access token back to the client and triggers the dissemination of the access control list update to the respective active router(s).

The authentication server uses a standard Mobile IPv6 stack with support for IPsec in order to provide the cryptographic means for the encrypted communication channel to the end-terminals (and potentially the access routers). A dynamic mechanism to add and remove IPsec security associations will be provided to

[1] For this to work in the absence of a global public key infrastructure, we statically configure the security association (SA) for the authentication server on the client. Based on this SA, IPv6 knows how to encrypt/decrypt data packets send to or received from the authentication server.

allow the authentication service to flexibly define how authentication messages are encrypted (and decrypted respectively)[1].

For the transport of our application-level authentication protocol we use UDP. Since the authentication protocol is fairly lightweight (i.e. only small amounts of data are exchanged), it does not justify the overhead of establishing separate TCP sessions for each authentication cycle. Reliability is achieved by means of a simple client driven retransmission strategy.

Although the use of UDP for the transport benefits scalability of the authentication server, the bottleneck in our architecture is still the centralised server. To overcome this limitation, we plan to exploit the new IPv6 feature anycast. This novel addressing scheme enables replication of servers behind a single anycast address to increase availability and redundancy.

Access Router and Gateway

The access routers and the gateway firewall are based on the LARA++ active router architecture [10]. LARA++ is a component-based active router platform that supports dynamic extensibility of the router functionality through remote loading and on-the-fly instantiation of active components. A sophisticated composition framework enables flexible integration of these components into the packet processing chain on the router, where they provide additional functionality.

The packet filtering and access control list management are implemented as active LARA++ components. The ACL management component listens for ACL updates (i.e., UDP datagrams sent to a well-known port on the router) and updates its access control list accordingly. The packet filter component in comparison intercepts inbound traffic (originating from the end-terminals) to verify their access credentials. If valid, the filter component removes the Hop-by-Hop Option including the access control extension header and forwards the packet; otherwise it drops the packets. Outbound traffic, in contrast, is intercepted to check whether or not it is destined to authorised clients. The gateway router will include a number of active components that attempt to secure the access network from malicious external nodes. These components will try to detect denial-of-service attacks (e.g. ping floods) by external nodes based on packet analysis (i.e., packet type, source address, data rate, etc.).

Comparisons with Other Approaches

The design of our architecture has drawn on the experience of earlier public access control research. In particular, we have combined a range of existing ideas with our own expertise in active and mobile networking and protocol design to develop a flexible, lightweight, scalable and secure access control solution with special support for mobile environments.

Two early access control systems, namely Carnegie Mellon's NetBar system [34] and the public access system developed at UC Berkeley [35], use specialised hardware (i.e. hubs, switches) to control network access on a port basis. Both solutions dynamically enable or disable linklayer access to network ports based on user authentication. While CMU's NetBar system is based on a remote configurable VLAN switch, Berkeley's solution relies on an intelligent hub. Despite the fact that both solutions require expensive specialized hardware, they are not practical for wireless networks, where many end-terminals share the same base station and hence the same network port on the switch/hub.

A more promising hardware-centric approach was recently announced by the IEEE 802.1x standardisation body. The port-based network access control [36] performs layer 2 authentication of the host to obtain access to a switch LAN by means of the extensible authentication protocol (EAP). This approach provides per-port access control at the first point of attachment (the edge). The fact that our infrastructure is based on microcellular layer 3 networks, which can exactly correspond to the link-layer cells, allows our solution to support access control at the same granularity than port-based network access control.

[1] More specifically, a global SA for datagrams received from any of the end-terminals is needed to instruct IPsec to decrypt the messageusing the server's private key, whereas individual SAs for every client are required to define outbound message to be encrypted based on the current session key (shared between the client and AS).

The systems described above are all limited to address a single aspect, namely access control. Our architecture in comparison is provisioned to address supplementary aspects of a public access infrastructure, such as accounting, quality of service, monitoring, or detection of security attacks. Especially the use of dynamically extensible active routers inside the access network provides great flexibility for future integration of additional functionality and services as they are needed or being developed.

However, the major difference to the access control approach introduced so far is probably that our architecture performs access control at the network-layer rather than the link-layer. The advantage of layer 3 access control is that it can be used as a uniform mechanism across many link-layer technologies. Current link-layer access control solutions are still predominantly based on the idiosyncrasies of the technology at hand, although standardisation efforts are under way.

Two further network-level access control systems we are aware of are Standford's SPINACH system [37] and Microsoft's CHOICE [38]. Both have been fully deployed in a real environment. The early SPINACH system controls network access simply based on the address pair (IP, MAC) of successfully authenticated users. This approach cleverly reuses the existing infrastructure without the need for additional hardware or specialised client software, but at the cost of inferior security (i.e., no measures against MAC address spoofing). The more recent CHOICE system in comparison accomplishes a high-level of security through the concept packet marking and packet filtering. Successfully authenticated and authorised users receive a token at session initialisation time. A custom network device driver on the client attaches the tag to every outbound data packet to indicate its authorisations.

The architecture involves separate authoriser and verifier gateways in addition to a central authentication server. While the authoriser gateway enables restricted access to the authenticator only, the verifier gateway grants full access to the network based on the packet tags.

Although based on the same access control principles, our approach distinguishes itself from CHOICE in a number of ways. Three key differences are:

1. We introduce the concepts of short-lived access tokens and session keys, and a soft-state authentication protocol to enhance robustness and security. The fact that the user's security credentials (tokens and keys) are frequently renewed enables the use of lighter weight crypto systems without sacrificing security.
2. Our access control architecture accounts for smooth handoffs between layer 3 networks. Our approach is therefore not restricted to link layer handoffs and a single layer 3 network, which makes our architecture more scalable than CHOICE.
3. We introduce the concept of microcellular administrations (referred to as districts) to enable fine-grained access control, accounting and monitoring, which considerably improves flexibility (for example, a wide range of access policies and accounting models can be implemented).

Furthermore, unlike CHOICE and SPINACH, our architecture does not rely on the availability of other higher level services, such as DHCP for auto-configuration of the client terminals and HTTP (and SSL) for Web-based user authentication. Instead, our clients use the standard IPv6 auto-configuration mechanism to obtain a network address, IPsec encryption to secure the authentication protocol, an extension to the Mobile IPv6 stack to accomplish packet tagging, and a lightweight request/response authentication protocol (based on UDP).

3.9 Vite and dbind

Vite (the Italian word for 'vine' or 'grapevine') is an authorization service IPv4/IPv6 compatible. It has been used for years (IPv4) and at least 8 months (IPv6) at the Department of Computer Science, University of Bologna.

All the students' laptop networks and wireless networks use Vite to authenticate users. Vite in its current implementation uses NIS, ssh and a specific tool (fsh). When a user joins the network, plugs his/her laptop in a public LAN socket or get connected to an access point he/she is on a local LAN with no route to the Internet. The user gets an address through a DHCP service (IPv4) or by auto-reconfiguration (IPv6, router advertisements are sent on the net). If the user wants to be connected to the internet he opens an ssh connection to the vite server.

On vite the NIS service authenticates the user: fsh is started instead the standard user's shell. fsh uses a script to log the MAC address, to open the routing (by updating the iptables) and it waits until the TCP transport connection terminates.

Route path is deleted upon TCP end or abend.

There are several pros of Vite approach:

- It uses only well known standards
- The security of the authentication phase gets inherited from ssh and NIS
- Heartbeat is managed by ssh and TCP keep-alive signals
- It is lighter than VPN

Current implementation:

- Has been interfaced to our DDNS service named dbind
- Provides independent IPv4 and IPv6 authentication and routing (a user can decide weather to have V4only V6only or both routed to the Departmental network and the Internet. This is useful when testing one or the other protocol stack.)

dbind is a DDNS service. It is useful both for permanent workstations and for temporary address assignments.

Dbind has two main goals:

1. create and update DNS tables of permanent machines with no pain (a boot init.d script does all the work). When a workstation changes its address it is sufficient either to run the script (/etc/init.d/dbind restart) or to reboot.
2. manage the database of DHCP and dynamic reconfiguration/vite assigned address. When a new client joins the net and get authenticated its address is automatically updated on the DNS server.

The DNS server must be configured to have an account named dbind and an account for each permanent machine (for simplicity we have put the hostname as the username on the DNS server).

Each account has a home directory just for ssh dsa authentication and uses a specific bindsh program as login shell. When a remote machine is authorized to access as dbind it can add and remove entries from several domains, when it is authorized to login as a specific host it can just update the DNS data for that specific host.

The former is used for temporary addresses the latter for permanent machines. ssh is used to secure the remote invocation of the bindsh shell. bindsh invocations are idempotent (several invocations with the same data have the same effect as a single one).

The bindsh shell has very simple commands (add inet, add inet6, rm inet, rm inet6) and the DNS get updated using the callee address. The dbind script at the remote site executes:

```
ssh $DBIND_SERVER add inet
ssh $DBIND_SERVER add inet6
```

on start, and

```
ssh $DBIND_SERVER rm inet
ssh $DBIND_SERVER rm inet6
```

on stop. When a remote machine has access as the user dbind it can specify also the name, the domain and the address of the DNS entry to be added.

Vite uses the dbind service.

4 WLAN Network Design and Deployment

4.1 Conducting a site survey

Conducting a site survey before installing a wireless network is a tedious but necessary process. In its simplest form, a network administrator installs an AP and walks around the coverage area with a WLAN-enabled laptop or PDA taking RF signal measurements at various points within the area. From the analysis of the signal measurements, it is possible to determine the optimum location of the AP according to the desired coverage area and data rate desired. However, there is a direct trade-off between coverage area and achieved data rates. Furthermore the entire process is somewhat cumbersome and very much trial and error. This hit-or-miss approach is worsens as the larger the network. Once the WLAN contains many, APs serving hundreds of users over a large area covering multiple floors and walls, the site survey becomes ever more complicated. To help with this, some WLAN vendors bundle basic site-survey tools with their APs and NICs. More sophisticated site survey software packages are available but they are often expensive and geared more towards the design of cellular networks.

It pays to follow a structured approach to the design of the WLAN network. First, the network designer must select the type of WLAN network most suitable for the requirements. From this, the number of APs needed and where they should be located can be approximated. Next, the correct channel selections for each AP should be determined to provide the maximum throughput possible with minimal co-channel interference. Finally, testing of the APs and user feedback should be used to fine-tune the location and configuration of the APs

4.2 Choosing the 802.11 Network Type

For owners of existing 802.11b Wi-Fi equipment, 802.11g promises to provide a smooth migration path to higher data rates. Since it uses the same 2.4GHz band and is backward-compatible with 802.11b, it has several benefits:

- Many vendors will be able to offer firmware and software upgrades to their existing devices
- Users with ‘legacy’ 802.11b cards will be able to use new 802.11g APs
- A new site-survey is not necessary

Thus, network managers can install new 802.11g access points in their existing 802.11b-based networks without affecting existing users. However, there are a couple of issues worth noting. The first is that, like 802.11b, 802.11g can use only three non-overlapping channels, making it difficult to avoid interference for medium to large WLANs. The second issue is that when an 802.11b device associates with an 802.11g AP the throughput for all the 802.11g clients also associated with that AP will be reduced. This is due to the relatively longer transmission times between the AP and the 802.11b client compared to that of the 802.11g clients. Once an 802.11b station associates with an 802.11g AP, realistic 802.11g transmission speeds fall from about 23Mbps to about 14Mbps - even if the 802.11b station is not transmitting. 802.11g networks slow down because they are designed to automatically go into *protected mode* in the presence of 802.11b stations. A special CTS (clear-to-send) header is appended to 802.11g frames so that 802.11b stations can detect and avoid collisions with 802.11g traffic; but this clutters up the network with additional overhead.

Alternatively, 802.11a offers higher data rates and more non-overlapping channels than 802.11b thus, giving potential for fewer APs for a given level of throughput. For network designers, this means that it will be easier to design 802.11a WLANs that avoid interference because at least 19 non-overlapping 5 GHz channels have been freed up for global use.However, RF coverage range is less than with 802.11b so this advantage is

alleviated slightly (i.e. more APs are needed for a given amount of RF coverage desired). Perhaps more importantly, the different ranges and frequency band that is used will require a site survey to determine the optimal number and positioning of APs according to network requirements. For an entirely new wireless network deployment, this is not really a problem as a site survey for whichever 802.11 type would be necessary anyway. However, for existing 802.11b based wireless networks, upgrading to 802.11a will almost certainly require existing (upgraded) APs to be moved and possibly new APs to be added if anywhere near optimal deployment is desired.

802.11a devices are generally more expensive than 802.11b but this will most likely change as more products are shipped. The incompatibility feature can be alleviated somewhat by deploying dual a/b devices and APs that many vendors are making available. Users with dual a/b cards will be able to associate with either type of AP depending on what is available. Similarly, dual APs can be configured to offer whichever of 802.11a/b according to the nature of the user base. Furthermore, products designed for use in North America and Canada (under FCC regulations) will eventually work in Europe and other parts of the world. Until the recent decisions from the WRC, the global 5 GHz spectrum allocation has been diverse, making consistency of 802.11a product design and usage inherently difficult.

The decision as to which WLAN type is to be used is based on the network requirements of the organisation and roughly comprises: area coverage, data capacity, and any existing network/user infrastructure. Of course, a/b/g WLAN types can be used to complement each other. For example, 802.11b can be used for blanket coverage with 802.11a used in areas where higher throughput is required.

4.3 Calculating the Number of Access Points

Depending on the requirements of the wireless network, a designer can plan for either maximum RF coverage or for maximum data capacity (throughput). Obviously, planning for maximum throughput will require a greater number of access points. If the required data capacity of the network is anticipated to be low compared to the achievable data rate of the network type (e.g. <5Mbps in 802.11b environment), the network designer can simply use a single AP to cover each cell. Conversely, maximising data capacity will require multiple APs in each 'cell' according to how many non-overlapping channels are available.

Designing simply for coverage one can simply set the radio on each AP to maximum to widen the signal's reach. Yet, designing simply for coverage will often not provide users with an acceptable level of throughput (due to low signal quality) except in the most basic of circumstances. Instead, it is more beneficial to have smaller cell sizes (microcells) by reducing the power of the AP RF transmissions, although more APs are needed for a given coverage area. Such microcells increase overall network throughput because they share more bandwidth amongst relatively fewer stations. However, the network designer must be careful to plan the channel assignments so the there is no co-channel interference.

4.4 Management of Access Points

Most inter-AP management protocols are proprietary in that they will only work with APs from the same vendor. Currently, there is no standard inter-AP management protocol that will allow the management of multiple APs from arbitrary vendors. This means that a network designer must either 1) deploy APs from the same vendor and use a proprietary AP management protocol or 2) deploy heterogeneous APs and have to manually configure and manage them according to the network's requirements. In light of this, recent effort in the IEEE (the 802.11f task force) has concentrated on the standardisation of a protocol to achieve multi-vendor interoperability of APs across a distribution system.

In addition, at the recent IETF 57 meeting in Vienna, there was a proposal to form an AP management working group called capwap (Control And Provisioning of Wireless Access Points) which would take the 'Lightweight Access Point Protocol' (LWAPP) from the seamoby working group as a starting point. The majority of the attendees voted in favour of forming the working group.

4.4.1 802.11f / IAPP

The IEEE P802.11 specification details the MAC and PHY layers of a Wireless LAN system and includes the basic architecture of these systems, including the concepts of Access Points and distribution systems. Since there are many ways to create a WLAN network, the precise implementation of APs and distribution systems was not included in the specification. Furthermore, many of these implementation possibilities require concepts from higher layer entities and, as such, were considered out of scope for 802.11.

The drawback with this is that AP devices from different vendors are unlikely to inter-operate across a distribution system because of the different approaches taken to the design of the distribution system. Although 802.11 based systems have grown in popularity, this limitation is a direct impediment to the design and implementation of medium and large scale WLAN networks.

To address this problem, the IEEE 802.11f task group came up with the Inter-Access Point Protocol (IAPP) [41], which specifies the necessary information that needs to be exchanged between APs to support the 802.11 distribution system functions. The IAPP has been developed primarily for the environment of a distribution system consisting of IEEE 802 LAN components that are running IP; although other environments can be supported.

The IAPP specification is a "recommended practice" document that aims to achieve radio AP interoperability within a multi-vendor WLAN network. The specification defines the registration of APs within a network and also the interchange of information between APs points when a user/mobile station moves from one AP to another (handoff). Thus IAPP will help with fast hand-off from AP to AP with a distribution system.

IAPP packets can be carried in TCP or UDP and the Remote Authentication Dial-in User Service (RADIUS) is used so APs can obtain information about one another. In particular, an IAPP entity must be able to use a RADIUS server to discover the IP address of another AP in the ESS when given its BSSIDs. Also, an IAPP entity will use a RADIUS server to obtain security information to protect the content of certain IAPP packets. The IAPP is most likely to be implemented by APs, although other network devices such as bridges and switches may also be affected.

The APs function much the same as 802.1D bridges; the IAPP also support the following functions:

- distribution system services (as defined in ISO/IEC 8802-11:1999)
- mapping of AP MAC addresses to network layer (IP) addresses
- formation of a distribution system
- maintenance of a distribution system
- enforcement of the restriction that a station may only associate with one AP at any given time
- transfer of station context information between APs

4.4.2 LWAPP and CAPWAP

As part of the seamboy (seamless mobility) IETF working group, there is work in progress to design a protocol that allows a router or switch to control and manage a group of APs in an interoperable fashion [41]. Although the intention is for this protocol to be layer 2 independent, the main focus is on an 802.11 binding. This Light Weight Access Point Protocol (LWAPP) is meant to provide a common way for wireless WLAN switches to communicate with any vendor's radio infrastructure. The aim is to allow for the decoupling of radio systems (i.e. the PHY and MAC functionality of an 802.11 AP) from their management systems. This will provide the flexibility to choose 802.11 radios from multiple vendors while being able to choose a centralised wireless configuration/security/management/monitoring system from another vendor. The protocol is being backed by several WLAN switch vendors such as Airespace, DoCoMo USA Labs and Legra Systems. In theory, different brands of WLAN switches and APs will be able to communicate over the

LAN making it possible to select any combination of 802.11 APs and switches while still being able to manage and control the network.

In such architecture, the APs are 'lightweight' in that they contain little or nothing else other than the 802.11 functionality. One analogy would be that 802.11 APs become like "disposable light bulbs" in that they can be swapped in and out without changing the operation of the wireless network.

Although it does not define specific management features, the LWAPP specification describes a common way for the two network entities (radio and management) to communicate. This effort has emerged, in part, because the thin access point (AP)/wireless switch architecture (which moves some, if not all, smarts from the infrastructure radios into a centralized controller) didn't exist at the time that 802.11 standards were created. Similarly, several of the functions (such as SNMP and DHCP) specified in LWAPP encroach on the Layer 3 services area; thus, the involvement of the IETF rather than the IEEE.

Although LWAPP is focused on 802.11 WLANs, a superset of the effort dubbed the 'Control and Access Provisioning of Wireless Access Points' (aka 'capwap') can be applied to any wireless network such as short-range, high-speed ultrawideband wireless networks and the emerging 802.16a and 802.16e standards for fixed and mobile metro-area wireless networking. At the 57th IETF meeting in Vienna, members voted in favour of creating a working group called *capwap* to address this very issue.

The main LWAPP and CAPWAP features include:

- independent of the wireless link type
- acquisition of APs by the management entity
- automatic discovery of the management entity
- configuration and management of the wireless link by the management entity
- control of AP host load
- security for LWAPP/CAPWAP signalling
- no changes to the 802.11 MAC

The AP and switch vendors that are adding wireless support into their existing product lines, should eventually include code for the LWAPP protocol in their products. Although the IETF ratification process could last well into 2004 (the target is to standardise within 12 months), vendors may begin to introduce early versions of LWAPP, or parts of it much earlier.

5 Working with IPv6 / Mobile IPv6

5.1 Host Configuration

When introducing an IPv6 host into a wireless network, the host must be able to obtain the necessary information to configure itself for network operation. In theory, this should not be too different in a wireless LAN than it would be in a wired LAN. If the WLAN APs are implemented as simple layer 2 devices, they take on a role analogous to Ethernet switches in a wired network.

In this way, IPv6 hosts can configure themselves via the usual methods of stateful address autoconfiguration using DHCPv6 [42] or stateless address autoconfiguration using Router Advertisements (RAs) and Neighbour Discovery [43],[44]). Both the stateless and stateful autoconfiguration methods provide a way for a fixed or mobile IPv6 host to autoconfigure itself with one or more IPv6 addresses, default gateway and other parameters.

Although present in DHCPv6, there is currently no DNS option in RAs. Since the availability of a DNS server is crucial for the support of Internet services such web, email and file transfer in all realistic network environments, RAs must be supplemented with DHCPv6 or static configuration in order for IPv6 hosts to obtain the address for their local DNS server.

Currently, there is some effort in the IETF to standardise a DNS discovery mechanism based on RAs which will remove this requirement [45]. This DNS discovery mechanism introduces two new RA options in Neighbour Discovery: the DNS Server option and the DNS Zone Suffix option. The DNS Server option contains the IPv6 address of the (recursive) DNS server and the DNS Zone Suffix option contains the suffix of the DNS zone where the subnet is located.

There is a general tradeoff between ‘lightweight’ and ‘heavyweight’ APs. In lightweight APs, there is little more than the 802.11 functionality inside the AP. Therefore, all higher layer functionality such as IPv6 routing, DHCPv6, Home Agent, RADIUS, SNMP etc. must be deployed elsewhere in the network, generally by the Access Router and the fixed network behind it. Conversely, heavyweight APs may contain some or all of these higher layer functions thus reducing the need for separate dedicated devices in the network. At the time of writing there are few APs available that contain such higher layer IPv6 features like DHCPv6 or MIPv6 Home Agent functionality. As mentioned earlier, deploying such heavyweight APs can make network configuration and management extremely laborious if they are from heterogeneous vendors. Hence, the trend towards lightweight APs using a common protocol to talk with a centralised management entity that controls the higher layer functions.

5.2 Subnetting and Addressing

A significant part of designing an IPv6 network (wired or wireless) is the subnetting and addressing plan. In wireless networks, IPv6 routing is rarely, if ever, available on the APs. Therefore, the ratio of APs per Access Router (AR) will greatly influence the subnetting and addressing plan (and vice versa). One extreme is to have an AR for every AP in the WLAN (or possibly using separate interfaces of the same router). Each WLAN cell will comprise a separate subnet and will receive separate IPv6 prefixes in RAs than other WLAN cells. This has the advantage that because the wireless cells are routed rather than bridged, there is considerably less broadcast traffic cluttering up the wireless link. This may be an important consideration when maximising available bandwidth is paramount. There is also an argument that if access control is enforced at each AR, this distributes the load of access control throughout the network.

However, this method is probably too costly in terms of router deployment for the general case. Of course, the other extreme is to have the entire WLAN hanging off one interface of the AR. Thus, all the WLAN cells belong to the same IPv6 subnet and receive the same IPv6 prefix in RAs. Depending on the size of the WLAN, this may or may not be a reasonable solution. Certainly, for small WLANs the overhead of broadcast traffic is not too severe and it is generally manageable; but for larger WLANs or where there are logical separations in the WLAN (e.g. different departments needing their own IPv6 subnets), separate routers (or separate interfaces on the same router) will be necessary. Note than in IPv6, broadcast traffic can be quite significant in large WLANs due to Neighbour Discovery and Router Advertisements.

Common practice would seem to be to assign a /64 prefix to each WLAN regardless of the ratio of APs to the AR. Unless some of the APs are 'heavyweight' and can participate in IPv6 routing, all the APs served by an AR interface should be seen as a logical link from the IPv6 point of view and thus should not be further subnetted (hence, no less than a /64). Thus, it is wise for network managers not to assign specific IPv6 subnets for a 'WLAN testbed' since this does not allow for the wireless network to cover departmental boundaries and the addressing can quickly become unsuitable as the wireless infrastructure grows. Instead, subnet and addressing policies should be governed by the needs of the organisation, not the link type.

5.3 DAD considerations

As part of the autoconfiguration process, each IPv6 node must perform duplicate address detection (DAD) when generating its IPv6 address. In general, a host's IPv6 address is generated by concatenating a 64 bit interface identifier (based on its 48/64 bit MAC address) onto a 64 bit network prefix. Therefore, address duplication is most likely to occur when two hosts on the same link both own the same MAC address[1].

If an IPv6 node detects address duplication when receiving a Neighbour Advertisement message from another node, the duplicated address must not be used for this IPv6 node and the node must generate its own address by another mechanism such as random Interface ID generation.

However, in 802.11 networks some inconsistency with DAD has been noted in [46]. If an IPv6 node in an 802.11 WLAN has the same MAC address as another node, it will not be detected using normal DAD.

This is due to the 802.11 MAC protocol, which stipulates that broadcast frames should be discarded by the originating station. If an IPv6 node in an 802.11 WLAN receives a broadcast frame whose source MAC address matches its own MAC address, the 802.11 MAC logic will discard the frame without it ever reaching the IPv6 layer inside the node. Thus, IPv6 DAD will not be able to detect a duplicate address in an 802.11 WLAN because the broadcast frame (which is what Neighbour Discovery uses) from the duplicated node will be discarded by the receiving node's 802.11 MAC logic.

Therefore, IPv6 DAD will always succeed in an 802.11 WLAN even though there may be two hosts owning the same MAC address.

Note that this is inconsistent with the RFC on IPv6 Stateless Address Autoconfiguration.

Appendix A of RFC 2462 [44]:

"*To perform Duplicate Address Detection correctly in the case where two interfaces are using the same link-layer address, an implementation MUST have a good understanding of the interface's multicast loopback semantics, and the interface cannot discard received packets simply because the source link-layer address is the same as the interfaces.*

This particular problem can be avoided by temporarily disabling the software suppression of loopbacks while a node performs Duplicate Address Detection."

[1] In theory this should not occur since but it is possible for MAC addresses to be 'spoofed' or configured incorrectly.

Two possible solutions are described in [46], but they involve either the filtering or ignoring of broadcast frames at the 802.11 MAC. However, these solutions are possible with many of today's 802.11 client NICs and APs.

6 Available Hardware and Software

Over the last few years, hardware and software for the deployment of 802.11 WLANs has become readily available. The 802.11b market is already competitive enough that 802.11b WLAN deployment has become cost-effective enough to be a serious alternative to wired networks for both enterprise and SOHO networks.

The recent introduction of the high-rate 802.11a and 802.11g standards has not only seen the proliferation of associated single-mode devices, but also the emergence multi-mode devices that can operate at more than one 802.11 mode. The chipmaker Atheros Communications[1] has recently introduced tri-mode chipsets with combined 802.11a/b/g support coupled with Advanced Encryption Standard (AES) hardware support for compliance with the forthcoming 802.11i security standard.

Cavium Networks[2] has announced a new range of security processors for Wireless LAN applications and protocols. Cavium's NITROX Wireless Security Processors are designed to make it easier for equipment manufacturers to introduce support for emerging 802.11 security standards, such as Wi-Fi Protected Access (WPA) and 802.11i. For 802.11i, WLAN APs and NICs will benefit from security acceleration in hardware to achieve required performance, which is what Cavium's NITROX Wireless Processors aim to achieve. The WPA security specification is a cut-down version of what will become 802.11i and is beginning to make its way into 802.11 products. 802.11i is due to be ratified as a standard by the IEEE some time in 2004. Aruba Wireless Networks, a manufacturer of high-performance WLAN switching systems for enterprises and wireless ISPs , is the first company to publicly commit to Cavium's wireless technology.

There are now numerous available multimode 802.11 APs and client NICs available. Some are available as dualmode[3] (e.g. 802.11b/a or 802.11b/g) or tri-mode (802.11b/a/g). Some examples of multimode 802.11 client NICs and their associated features are listed in Table 4. Table 5 lists some available multimode 802.11 APs and their associated features.

Product	802.11 modes	OS support	Driver/client features	Security features
D-Link AirXpert ABG DWL-AG520 and DWL-AG650	b, a, g	Windows 95/98/2000/ XP	Site survey tool, profile management, AP scanning, link status meter, NIC diagnostics, performance measurement utility	40 and 128-bit WEP, 802.1x, AES hardware capable
D-Link AirPlus Xtreme G DWL-G520 and DWL-G650	b, g	Windows 95/98/2000/ XP	Site survey tool, profile management, AP scanning, link status meter, NIC diagnostics, performance measurement utility	40 and 128-bit WEP, 802.1x
Enterasys RoamAbout R2	b, a	Windows 95/98/2000/ XP, Linux, Mac, Pocket PC, Palm OS 4.x or 5.x	Site survey tool, profile management, firmware upgrade, AP scanning, link status meter, NIC diagnostics,	40 and 128-bit WEP, 802.1x, AES hardware capable

[1] http://www.atheros.com
[2] http://www.cavium.com
[3] Dual-mode is not necessarily the same as 'dual-band', which refers to a device capable of operating on two different wavebands (e.g. a device capable of operating at 2.4GHz and 5Ghz would be dual-band).

			performance measurement utility	
Linksys Wireless-G WMP54G	b, g	Windows 98/ME/2000/XP	Site survey tool, profile management, firmware upgrade, link status meter, radio transmit power control	40 and 128-bit WEP, AES hardware-capable
Linksys Dual-Band Wireless A+G WMP55AG	b, a, g	Windows 98/ME/2000/XP	Site survey tool, profile management, firmware upgrade, link status meter, radio transmit power control	40 and 128-bit WEP, AES hardware-capable
Netgear WAG511 802.11a+g Dual-Band Wireless PC Card	b, a, g	Windows 95/98/ME/2000/ XP	Link status meter, NIC diagnostics	40 and 128-bit WEP
Netgear WAB501 802.11a+b Dual-Band Wireless Adapter	b, a	Windows 95/98/ME/2000/ XP	Site survey tool, link status meter, radio transmit power control	40 and 128-bit WEP
Proxim Orinoco 11a/b/g ComboCard	b, a, g	Windows 95/98/ME/2000/ XP	Site survey tool, profile management, AP scanning, link status meter, radio transmit power control, NIC diagnostics, performance measurement utility	40 and 128-bit WEP, 802.1x, Cisco LEAP, AES hardware-capable
Proxim Orinoco 11a/b ComboCard	b, a	Windows 95/98/ME/2000/ XP	Profile management, AP scanning, link status meter, radio transmit power control, NIC diagnostics, performance measurement utility	40 and 128-bit WEP, 802.1x, Cisco LEAP, AES hardware-capable
SMC2336W-AG	b, a, g	Windows 95/98/2000/ XP	Site survey tool, profile management, AP scanning, link status meter, radio transmit power control, NIC diagnostics, performance measurement utility	40, 128 and 152-bit WEP, 802.1x, AES hardware-capable

Table 4 Multimode NICs and associated features

Product	802.11 modes	Network Features	Config and Management Features	Security Features
3Com Wireless LAN AP 8200	b, a	DHCP server, Power over Ethernet, radio transmit power control.	HTTP, SNMP, TFTP	40 and 128-bit WEP, MAC address filtering, 802.1x
3Com Wireless LAN AP 8500 and AP 8700	b, a, a*	DHCP server, Power over Ethernet, VLAN tagging, Mobile IP, radio transmit power control.	serial port, HTTP, SNMP, TFTP	40, 128 and 154-bit WEP, MAC address filtering, 802.1x
D-Link AirPlus Xtreme G DWL-2000AP High-Speed 2.4GHz Wireless AP	b, g	DHCP server, PoE, radio transmit power control	HTTP, central management console for multiple APs	40 and 128-bit WEP, MAC address filtering
D-Link AirPro DWL-6000AP	b, a	DHCP server, PoE	HTTP, central management console for multiple APs	40 and 128-bit WEP, MAC address filtering, 802.1x
D-Link AirXpert DWL-7000AP	b, a, g	DHCP server, PoE, radio transmit power control	HTTP, central management console for multiple APs	40 and 128-bit WEP, MAC address filtering, 802.1x
Enterasys RoamAbout R2	b, a, g	DHCP server, broadcast and multicast filters, protocol filtering, PoE, Class of Service, Mobile IP	serial port, Telnet, HTTP, SNMP, TFTP, central management console for multiple APs	40 and 128-bit WEP, MAC address filtering, 802.1x, integrated VPN termination
Intermec MobileLAN access WA21 and WA22	b, a	DHCP server, broadcast and multicast filters, protocol filtering, PoE, VLAN tagging, Class of Service	Telnet, HTTP, TFTP, central management console for multiple APs	40 and 128-bit WEP, MAC address filtering, 802.1x
Netgear WAB102	b, a	DHCP server	HTTP	40 and 128-bit WEP, MAC address filtering
Proxim Orinoco AP-600, AP-2000 and AP-2500	b, a, g	DHCP server, broadcast and multicast filtering, protocol filtering, PoE, VLAN tagging, Class of Service, radio transmit power control	serial port, Telnet, HTTP, SNMP, TFTP, central management console for multiple APs	40 and 128-bit WEP, MAC address filtering, 802.1x

Table 5 Multimode APs and associated features

* = 802.11a Turbo Mode, a proprietary scheme with theoretical data rates of 108Mbps

7 Conclusions

It is clear that the popularity of 802.11 WLANs, coupled with well-documented security weaknesses in WEP, has lead to much thought on the problem of authentication, security and privacy in 802.11 WLANs.

At the time of writing, it is debatable whether there is a definite solution that can meet all requirements. Much depends on the nature of the WLAN, who the users are, the movement of the users, data sensitivity and other network conditions. From the material presented in chapter three, one can see that there are many security solutions to choose from, but there is no commonly accepted solution; at least, not yet. The 802.11i task force promises to bring about such a common solution although in some sense this can be misleading as it is really the use of EAP that makes it so flexible. Probably the most fool-proof security architecture will involve 802.1x and AES under the 802.11i umbrella.

A 'one size fits all solution' security solution may not be possible yet (if ever), although it *is* possible for any WLAN to be made secure provided that the security solution is designed to closely match the network requirements. Of course, there is a trade-off between security and management overhead. In theory combining authentication, encryption and certificates into a security solution should provide all the protection required. However, creating, distributing and maintaining certificates for users is not always a practical way of operating a network. Furthermore, there is a direct trade-off with security mechanisms and network performance. Encrypting and decrypting packets is a costly operation for network devices and usually requires hardware support for commercial networks. It is also a direct hindrance to user roaming in a real-time context. Having to authenticate via whichever security solution is implemented will use up valuable milliseconds, if not seconds, resulting in unacceptable delay and lost packets.

Therefore, it is more prudent to apply security solutions to fit the network usage. What is the main use WLAN? For example, a relatively open and insensitive WLAN can implement no or simple authentication, coupled with no encryption, or encryption via WEP. This might be reasonable for a free and public WLAN hotspot where there is little interest in authenticating users and if data sensitivity is required, it is up to the user to implement their own solution (e.g. an IPSec VPN connection). However, this would not suffice for an enterprise or commercial WLAN where authenticating users is imperative and sensitive data needs to be protected.

Thus, the applicability of security mechanisms to the wireless network in question must be considered. E.g.:

- Enterprise WLAN for corporate users
    - this type has the most stringent requirements
    - need secure access control and authentication of users
        - should use certificates
        - no unauthorised access
    - privacy of data essential
        - need strong encryption (not standard WEP) for all users
    - could use 'temporary' certificates or accounts for trusted visitors
- Campus based WLAN for students and staff.
    - need authorisation and access control for registered students and staff
        - PKI is a possibility
        - but must be simple for users
    - need to cater for visiting students and staff

        - roaming agreements?
        - temporary accounts?
        - restricted open access (web and email)?
    - privacy of data could be important for many users
        - strong encryption according to user type
    - accounting usually not needed but could be by some institutions
- Commercial Public WLAN
    - public access but need to authenticate paying users
        - don't care who as long as they pay (have paid)?
        - must be simple for users
    - will need to grant access to 'visitors' from roaming agreements
    - data sensitivity unknown
        - provider could offer default encryption level (e.g. 128 bit WEP)
        - users with sensitive data should use their own solution
    - accounting backend linked to authentication servers and access routers
- Community Public WLAN
    - free public access
        - no authorisation or access control?
        - depends on usage policy
    - could have default WEP encryption
        - users with sensitive data should use the own solution
- Small Office / Home Network
    - probably little need for access control
        - MAC address filtering may suffice
    - encryption needs depends on use
        - office: default WEP or stronger
        - home: nothing or default WEP

References

[1] Geir, Jim, “Wireless LANs”, Second Edition, SAMS Publishing 2002.

[2] Schiller, Jochen, “Mobile Communicatons”, Addison Wesley, 2000.

[3] Stallings, William, “Wireless Comunications and Networks”, Prentice Hall, 2002.

[4] Intersil, “Wireless and Related Network Standards and Organizations”, 2001.

[5] M. Stemm, R.H. Katz, “Vertical Handoffs in Wireless Overlay Networks”, ACM Mobile Networking (MONET), 1997.

[6] T.S. Rappaport, “Wireless Communications – Principles and Practice”, Prentice Hall Publishings, 1996.

[7] LAN MAN Standards Committee of the IEEE Computer Society, “Wireless LAN medium access control (MAC) and physical layer (PHY) specifications”, IEEE Standard 802.11, 1999.

[8] Nokia, “General Packet Radio Service – GPRS – Nokia’s vision for a service platform supporting packet switched applications”, White Paper 1998.

[9] D. Johnson, C. Perkins, J. Arkko, “Mobility Support in IPv6” draft-ietf-mobileip-ipv6-24.txt, IETF Internet Draft, July 2003, work in progress.

[10] J. R. Walker, “Unsafe at any key size; An analysis of the WEP encapsulation”, IEEE document 802.11-00/362, October 27, 2000.

[11] S. Fluhrer, I. Mantin, and A. Shamir, “Weaknesses in the Key Scheduling Algorithm of RC4”, (undated, published in August 2001), presented at the Eighth Annual Workshop on Selected Areas in Cryptography (SAC 2001), Toronto, Canada, (August 16-17, 2001).

[12] A. Stubblefield, J. Ioannidis, and A. D. Rubin, “Using the Fluhrer, Mantin, and Shamir Attack to Break WEP”, AT&T Labs Technical Report TD-4ZCPZZ, Revision 2, August 21, 2001.

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

[14] C. Rigney, A. Rubens, W. Simpson, and S. Willens, “Remote Authentication Dial In User Service (RADIUS)”, IETF RFC 2138, April 1997.

[15] W. A. Arbaugh and A. Mishra, “An Initial Security Analysis of the IEEE 802.1x Standard“, CS-TR-4328, UMIACS-TR-2002-10, 6 February 2002.

[16] J. Walker, “Key Management for WEP and TKIP”, Intel, 802.11 Key Management Series: Part I (http://cedar.intel.com/media/pdf/wireless/80211_1.pdf).

[17] J. Walker, “The Temporal Key Integrity Protocol (TKIP)”, Intel, 802.11 Security Series: Part II (http://cedar.intel.com/media/pdf/security/80211_part2.pdf).

[18] S. Baily, “Is IEEE 802.1x Ready for General Deployment?”, GSEC Version 1.3, April 7, 2002.

[19] A. Palekar, D. Simon, G. Zorn, S. Josefsson, “Proptected EAP Protocol (PEAP)”, IETF Internet Draft draft-josefsson-pppext-eap-tls-eap-06.txt, March 2003.

[20] B. Aboba, D. Simon, “PPP EAP TLS Authentication Protocol”, IETF RFC 2716, October 1999.

[21] L. Salgarelli, Editor, “EAP SKE authentication and key exchange protocol”, IETF draft-salgarelli-pppext-eap-ske-02, November 1, 2002.

[22] P. Funk, S. Blake-Wilson, "EAP Tunneled TLS Authentication Protocol (EAP-TTLS)", IETF draft-ietf-pppext-eap-ttls-02, November 2002.

[23] P. Funk , "The EAP MD5-Tunneled Authentication Protocol (EAP-MD5-Tunneled)", IETF draft-funk-eap-md5-tunneled-00, March 2003.

[24] H. Haverinen (editor), J. Salowey (editor), "EAP SIM ", IETF draft-haverinen-pppext-eap-sim-10, February 2003.

[25] B. Aboba, "The Network Access Identifier", IETF RFC 2486, January 1999

[26] H. Krawczyk, M. Bellare, R. Canetti, "HMAC: Keyed-Hashing for Message Authentication", IETF RFC 2104, February 1997.

[27] B. Kaliski, J. Staddon, "PKCS #1: RSA Crytogrpahy Specifications", IETF RFC 2437, October 1998.

[28] S. Kent, R. Atkinson, "Security Architecture for the Internet Protocol", IETF RFC 2401.

[29] J. Loughney, M. Nakhjiri, C. Perkins, R. Koodli, "Context Transfer Protocol", IETF Internet Draft draft-ietf-seamoby-ctp-03.txt, June 2003

[30] W. Arbaug, N. Shankar, Y.C.J. Wan, "Your 802.11 Wireless Network has no Clothes", Technical Report, Department of Computer Science, University of Maryland, March 2001.

[31] D. Wheeler, R. Needham, "TEA: a Tiny Encryption Algorithm", Technical Report, Computer Laboratory, Cambridge University, UK.

[32] R. Rivest, "The MD5 Message-Digest Algorithm", IETF RFC 1321, April 1992.

[33] S. Schmid, J. Finney, A.C. Scott, W.D. Shepherd, "Component-based Active Network Architecture", in Proceedings of 6th IEEE Symposium on Computers and Communications (ISCC '01), Hammamet, Tunisia, July 2001.

[34] E. A. Napjus, "NetBar – Carnegie Mellon's Solutions to Authenticated Access for Mobile Machines", CMU White Paper, http://www.net.cmu.edu/docs/netbar.html

[35] D. L. Wasley, "Authenticating Aperiodic Connections to the Campus Network", June 1996, http://www.ucop.edu/irc/wp/wpReports/wpr005/wp005_Wasley.html

[36] LAN MAN Standards Committee of the IEEE Computer Society, "Port Based Network Access Control", IEEE Standard 802.1x, June 2001.

[37] E. Poger, M. Baker, "Secure Public Internet Access Handlers (SPINACH)", in Proceedings of the USENIX Symposium on Internet Technologies and Systems, 1997.

[38] A. Miu, P. Bahl, "Dynamic Host Configuration for Managing Mobility between Public and Private Networks", in Proceedings of the 3rd Usenix Internet Technical Symposium, San Francisco, March 2001.

[39] J. Finney, G. O'Shea, "Mobile 4-in-6: A Novel Mechanism for IPv4/v6 Transitioning", in Proceedings of Interactive Distributed Multimedia Systems (IDMS '01), Lancaster, UK, September 2001.

[40] LAN MAN Standards Committee of the IEEE Computers Society, "Recommended Practice for Multi-Vendor Access Point Interoperability via an Inter-Access Point Protocol Across Distribution Systems Supporting IEEE 802.11 Operation", IEEE 802.11f, 2003.

[41] P. Calhoun, B. O'Hara, S. Kelly, R. Suri, D. Funato, M. Vakulenko, "Light Weight Access Point Protocol (LWAPP)", IETF Internet Draft draft-calhoun-seamoby-lwapp-03.txt, June 2003.

[42] R. Droms, J. Bound, B. Volz, T. Lemon, C.Perkins, M. Carney, "Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", IETF RFC 3315, July 2003

[43] T. Narten, E. Nordmark and W. Simpson, "Neighbour Discovery for IP version 6", IETF RFC 2461, December 1998.

[44] S. Thomson and T. Narten, “IPv6 Stateless Address Autoconfiguration”, IETF RFC 2462, December 1998.
[45] J. Jeong, S. Park, L. Beloeil, S. Madanapalli, “IPv6 DNS Discovery based on Router Advertisement”, IETF Internet Draft draft-jeong-dnsop-ipv6-dns-discovery-00.txt, July 2003.
[46] S. Park, S. Madanapalli, O.L.N. Rao, “IPv6 DAD Consideration for 802.11 Environment”, IETF Internet Draft, draft-park-ipv6-dad-problem-wlan-00.txt, July 2003.

Glossary

AAA	Authentication, Authorization and Accounting
ATM	Asynchronous Transfer Mode
AVP	Attribute Value Pairs
CCK	Complementary Code Keying
CHAP	Challenge Handshake Authentication Protocol
DFS/TPC	Dynamic Frequency Selection and Transmit Power Control
DHCP	Dynamic Host Configuration Protocol
DSSS	Direct Sequence Spread Spectrum
EAP	Extensible Authentication Protocol
ETSI	European Telecommunications Standardization Institute
FCC	Federal Communications Commission
FHSS	Frequency Hopping Spread Spectrum
FPK	Fast Packet Keying
FTP	File Transfer Protocol
IEEE	Institute of Electrical and Electronic Engineers
HTTP	Hyper Text Transfer Protocol
ICV	Integrity Check Value
IPsec	IP Security Protocol
IR	Infrared
ITU	International Telecommunications Union
IV	Initialisation Vector
LAN	Local Area Network
LMDS	Local Multipoint Distribution Service
MAC	Medium Access Control
MD5	Message Digest 5
MIC	Message Integrity Check
MMAC-PC	Multimedia Mobile Access Communications Systems Promotion Council
MMDS	Multichannel Multipoint Distribution Service
NAS	Network Access Server
NASREQ	Network Access Server Requirements
NIC	Network Interface Card
OFDM	Orthogonal Frequency Division Multiplexing
PAN	Personal Area Network

PAP	Password Authentication Protocol
PBCC	Packet Binary Convolution Coding
PHY	Physical Layer
POP	Post Office Protocol
PPP	Point-to-point Protocol
QoS	Quality of Service
RADIUS	Remote Authentication Dial-In User Service
RF	Radio Frequency
ROAMOPS	Roaming Operations
SCTP	Stream Control Transmission Protocol
SLIP	Serial Line IP
SSH	Secure Shell
SSID	Service Set Identifiers
SSL	Secure Socket Layer
TCP	Transmission Control Protocol
TDM	Time Division Multiplexing
TLS	Transport Layer Security
UDP	User Datagram Protocol
UNII	Unlicensed National Information Infrastructure
VPN	Virtual Private Network
WAN	Wide Area Network
WECA	Wireless Ethernet Compatibility Alliance
WEP	Wireline Equivalent Privacy
WEP2	The first enhancement to the WEP protocol
Wi-Fi	Wireless Fidelity
WISP	Wireless Internet Service Provider
WLAN	Wireless Local Area Network
WLANA	Wireless Local Area Networking Association
WPAN	Wireless Personal Area Network

Appendix A

A1. Data Exchange in the Simple VPN-based Access Control Procedure

The simple VPN-based access control protocol consists of the message exchanges depicted below where the following notations are used:

Code_fail / Code_succ:	Code indicating failure / success
Encr{key}(data)	Cipher representing "data" encrypted with "key"
HMAC:	Keyed-Hashing for Message Authentication [26]: Keyed hash computed with "key" over "data"
ID_{ii}:	Node identity as defined in the IKE protocol
Indices $_{I, r}$	Denote initiator and responder, respectively, as defined in the IKE protocol
IP:	Internet Protocol address
ISAKMPi_AVP:	AVP containing ISAKMP phase 1 message
K_A:	Private key
KE:	Public Diffie-Hellman key as defined in the IKE protocol
MAC:	Media Access Control (layer 2) address
NAI:	Network Access Identifier [25]
N_N:	A nonce value
SA:	Parameters associated to an IKE or CHILD security association, as defined in the IKE protocol
SK_{XY}:	Key shared by X and Y where X and Y are one of N (nomadic node), A (attendant)or H (AAAH)
SPI:	Security Parameter Index
T_N:	A timestamp value

The message exchange takes place between peers, as depicted in Figure 3. APReq, AuthReq, AuthAns, and APAns denote the control messages exchanged by Node, Attendant and AAAH which are typically composed of a sequence of AVPs containing control information relevant to the AAA protocols. The set of existing AVPs is extended with a number of AVPs that encapsulate control information relevant to the SA and key management (a set of IKE payloads).

Node → Attendant

APReq: NAI, T_N, N_N, IP, MAC, ISAKMPi_AVP, HMAC{SK'_{NH}}(NAI, T_N, N_N, IP, MAC, ISAKMP_AVP),

with ISAKMPi_AVP = SPI_i, SA_i, KE_i, N_i, ID_{ii} and SK'_{NH} = MAC(SK_{NH} | N_N)

The message contains the nomadic node's NAI, which is used by AAAL to identify the appropriate AAAH to send the corresponding AAA authentication request.

A timestamp T_N is used to protect against reply attacks and a nonce N_N serves to derive the nomadic node-AAAH authentication key.

The IP address the nomadic node has acquired (typically through auto-configuration) in the visited network is stored within the message so that the attendant could learn to which address the corresponding reply must be returned without having to maintain states.

Similarly, the MAC address is also recorded in order to avoid having the attendant to maintain states or rely on neighbour discovery, as malicious nodes can easily misuse this protocol.

The nomadic node also includes an AVP containing an ISAKMP message and finally an authenticator that affords integrity protection to the whole message. The ISAKMP payload contains the SPI of the incoming IKE SA, the SA proposal, a public Diffie-Hellman key, a nonce and an identifier.

Attendant → AAAH

AuthReq: APReq, ISAKMPr_AVP, HMAC{K_A}(T_A, IP, MAC, ISAKMPi_AVP, ISAKMPr_AVP),

with ISAKMPr_AVP = SPI_r, SA_r, KE_r, N_r, ID_{rr}

The attendant produces a response to the ISAKMP initiation message received from the nomadic node. This operation must involve minimal computational resources therefore the processing is limited to:

Checking the nomadic node's SA proposal and compiling a response (SA_r) indicating the accepted transforms, Oakley group and lifetime;

Generating a nonce;

Generating an incoming SPI for the IKE SA.

Public Diffie-Hellman keys may be reused over multiple sessions therefore they may be periodically generated off-line, as they are time consuming operations.

Additionally the attendant computes, using key known only to itself, a keyed hash over the ISAKMP messages, the IP and MAC address of the supplicant and a local timestamp. The keyed hash must be echoed by the AAAH and it enables the attendant to check that the authentication answers it receives correspond to previous authentication requests.

AAAH → Attendant

AuthAns: Code_succ, T_N, T_H, N_H, ISAKMPr_AVP, HMAC{SK''_{NH}}(Code_succ, T_N, T_H, N_H, ISAKMPr_AVP), IP, MAC, ISAKMPi_AVP, HMAC{K_A}(T_A, IP, MAC, ISAKMPi_AVP, ISAKMPr_AVP),

with SK''_{NH} = MAC(SK_{NH} | N_H)

The AAAH checks the authenticator provided by the nomadic node, and if the supplicant is authentic, the timestamp provided in the message is compared to the local time. If the two time values are not shifted beyond a pre-configured accepted value, then the AAAH generates a positive answer.

The AAAH provides a success code, its own timestamp, a nonce used to derive the authentication key from SK_{NH} and authenticates the ISAKMP response message generated by the attendant.

Malicious AAA entities may attempt to fake authentication answers. However, the security scheme can only be breached when such an AAA entity conspires with malicious visitors. Otherwise an authentication answer cannot be replied nor can the public Diffie-Hellman key provided by a legitimate user be replaced. Such a collusion situation is however not typical to the proposed protocol and appropriate security measures must be taken to protect the AAA infrastructure against such attempts. In particular the AAA relay and proxy agents must be replaced with AAA redirect agents whenever this is possible, otherwise AAA servers that use their services must setup end-to-end security associations.

Attendant → Node

APAns: Code_succ, T_N, T_H, N_H, ISAKMPr_AVP, HMAC{SK''_{NH}}(Code_succ, T_N, T_H, N_H, ISAKMPr_AVP)

The attendant checks the keyed hash computed with its private key and if the message proves to be authentic and timely the attendant creates a state, derives the shared secret from the public Diffie-Hellman exchange, forwards the relevant data to the nomadic node and waits for it to initiate the IKE quick mode exchange.

Timer Synchronisation

AAAH → Attendant

AuthAns: Code_fail, T_N, T_H, N_H, HMAC{SK″$_{NH}$}(Code_fail, T_N, T_H, N_H)

with SK″$_{NH}$ = MAC(SK$_{NH}$ | N_H)

Attendant → Node

APAns: Code_fail, T_N, T_H, N_H, HMAC{SK″$_{NH}$}(Code_fail, T_N, T_H, N_H)

In this case the attendant does not create any new state and does not modify any existing state. Upon receiving such a message, the nomadic node checks the authenticator and the timestamp before proceeding with the adjustment of the local timer. This ensures that a malicious AAA entity that replies such a message or creates a faked one cannot disturb in any way the service.