DHCP/IPSec-based Security for Wireless Access Networks

A. Molinaro[1], L. Veltri[2]
[1] DIMET - Università degli Studi "Mediterranea" di Reggio Calabria – Italy
[2] Dip. Ingegneria dell'Informazione - Università degli Studi di Parma – Italy
email: antonella.molinaro@ing.unirc.it, luca.veltri@unipr.it

***Abstract*–In this paper we present a simple solution to provide secure access to users/terminals asking for Internet connectivity through a wireless access network. The proposed solution is based on some standard protocols (DHCP, Radius, EAP, and IPSec) and enables user authentication, network configuration parameters personalization, and secure communications on the wireless links. The main strength of our approach is that it can be easily implemented in a wireless local site (WLAN or WPAN) independent of any layer two mechanism and of the manufacturer's implemented features in the wireless user terminal.**

Index Terms— DHCP, IPSec, EAP, Radius

I. INTRODUCTION

Wireless local area networks (WLANs) are gaining popularity as a means for providing IP connectivity in both public and private hot spot environments (i.e., at office, home, university, in airports, hotels, and so on). One of the main challenges for a wireless Internet service provider is to manage the users/terminals access by providing a means to authenticate authorised users and to secure data communication over the wireless links.

Although there are several mechanisms that attempt to secure communication within a WLAN, there is not a standard architecture available yet. In practice, there are as many WLAN architectures as there are available systems.

Generally speaking, a WLAN-based architecture may offer services ranging from those that enable basic IP connectivity up to possibly any application-level services. The essential functionalities of a generic WLAN can be schematically represented as in Fig. 1, where a mobile terminal accesses an IP-based network through a WLAN access network (e.g. WiFi) which is connected to the rest of the Internet through an access router or a gateway. The wireless terminal is normally a laptop computer or a PDA with a built-in or a PCMCIA WLAN card.

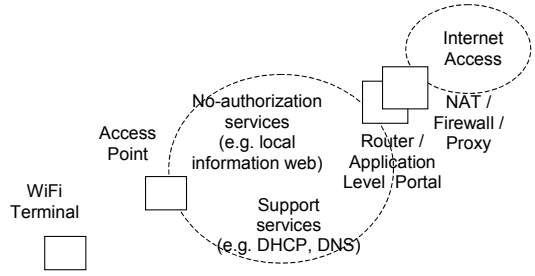

Fig. 1. WLAN based access architecture

The access network is typically formed by wireless and wired segments, allowing direct IP connectivity to a set of *basic services* that could not require authentication/authorisation (e.g. DHCP, DNS, local web browsing and other publicly available services). The connectivity with other services is generally enabled through an access router or an application level gateway. Such a connectivity can be offered directly by the network provider on an intranet (*provider specific services*) or by third parties (*external services*). Examples of such services may be web browsing, messaging and mailing, IP telephony, and many others.

The access to the external network may be open to everybody or restricted only to authorised users. In the latter case, the access router may act as an access control gateway, performing packet filtering based on specific and implementation-dependent security rules. In many cases, the network is connected with the rest of the Internet through a firewall/gateway, eventually performing NAT (Network Address Translation) or proxy functionalities.

Over this access network scenario, no standard user authentication, access control and data securing mechanisms are already defined, although several schemes are currently implemented. Existing solutions operate at the link-layer, which provides secure MAC communications (e.g. in IEEE802.1x and IEEE802.11i), or at the network layer (e.g. IP-sec solutions), or at the application layer (through a secure web interface the user is asked to enter his/her credentials).

Some of the drawbacks of these approaches, for example the link-layer dependency, and the lack of personalization of the configuration parameters as function of the authentication result and user credentials, can be overcome by the secure architecture proposed in this paper. The main strength of our proposal is that it requires the use of standard technologies and protocols (DHCP, RADIUS, EAP, IPSec) and very few changes in the software of the DHCP server to enhance the existing DHCP configuration procedure with authentication and service personalization capability.

In section 2 a brief overview of the most common securing mechanisms for WLAN-based architectures is given. In section 3 the proposed secure architecture is presented with a description of DHCP extensions to authentication and authorisation functionality and the use of the IP security mechanism to trust communications. In section 4 an implementation of the authentication and security mechanism is described. Conclusive remarks are summarised in section 5.

II. CURRENT WLAN SECURING MECHANISMS

A. *Link level authentication and data confidentiality*

In this scenario security mechanisms are provided by the WLAN at the data link (MAC) layer and are used for users and mobile terminals (MTs) authentication/authorisation and for data security. We focus on the mechanisms supported by IEEE 802.11 that is today the "de facto" WLAN standard [1].

IEEE 802.11 provides two authentication methods: *open* system and *shared key* authentication. While the former is actually a non-authentication scheme letting anyone requesting authentication be accepted, the latter is based on a challenge-response mechanism with a shared secret.

In the shared key authentication the MT sends an authentication request to the wireless Access Point (AP). The AP sends a chosen plaintext string to the terminal that, on its turn, replies with the WEP-encrypted string. If the string is correctly encrypted the AP sends a message to the MT to indicate that the authentication was successful. All communication between MT and AP is then encrypted with the shared secret key.

The weakness of this approach is mainly due to the fact that authentication is not mutual since: i) only the MT is authenticated, and ii) the WEP (Wired Equivalent Privacy) security mechanism is very weak [2].

The IEEE 802.11i Task Group (TGi) is working on the enhancement of the basic IEEE 802.11 with more secure mechanisms [3]. Regarding authentication and access control 802.11i uses the IEEE 802.1X framework, which in turn uses the Extensible Authentication Protocol (EAP) allowing for end-to-end mutual authentication between the MT and an Authentication Server (AS). Thus, even though 802.11i still performs access control on layer 2, the authentication message exchange is not restricted to the MAC layer but uses other IEEE as well as IETF standards.

IEEE 802.1X is a standard for port-based access control designed for filtering frames from/to non-authenticated terminals. The authenticator node (e.g. AP) acts as intermediate between MT and AS. Only when the EAP-based authentication procedure successes the authenticator node starts relaying MT frames from/to the rest of the network.

Normally the AS is implemented through a separate RADIUS or Diameter server. EAP messages between the authenticator node (the AP) and the AS are exchanged through the RADIUS (or Diameter) protocol (see Fig. 2). Data communications is then secured with Advanced Encryption Standard (AES)-based encryption relying on a fresh secret key that is periodically re-generated.

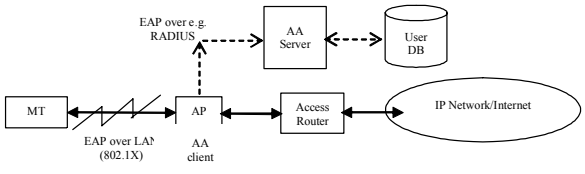

Fig. 2. Example of end-to-end authentication using EAP

B. *Network level security*

In this case security mechanisms are implemented at level three, i.e. at the IP layer. A typical implementation uses standard IP security (IPSec) extension and Virtual Private Network (VPN) technologies to ensure both data authentication and confidentiality. IPSec [4] is the standard IP extension to provide per-packet authentication and encryption services based respectively on Authentication Header (AH) and Encapsulating Security Payload (ESP) extensions. Such securing services use standard cryptographic algorithms based on secret material that can be statically configured or dynamically generated based on a secret symmetric key, a private/public keys pair, or X.509 digital certificates based on a common CA (Certification Authority) or a PKI (Public Key Infrastructure). An IPSec tunnel can be configured between end user terminals or between any intermediate node. Particularly, in a WLAN-based public IP access, an IPSec tunnel is normally setup between the MT and the access router in order to provide authentication, access control, and data confidentiality over the wireless links, regardless the underling WLAN technology. The main disadvantage of this solution is the difficulty to configure the MT with a valid static private key or to configure both the MT and the access router with valid certificates due to the lack of a real PKI implementation.

C. *Upper level authentication: Captive Portals*

In this case authentication and authorisation are performed at higher layers, by placing an authentication gateway in front of the wireless network to control network and services connectivity, and to force users authenticating against it before using the network. The gateway/access server is responsible for opening and closing firewall rules, thus allowing users to reach the services provided by the network. These services will typically include access to the global Internet.

Such access mechanisms are often referred to as Captive Portals, and are becoming a popular way for WiFi communities and hotspot operators to provide user authentication and IP flow management (e.g., basically traffic shaping and bandwidth control) without a special client application. Often, Captive Portals allow a user to exploit a traditional Internet browser as a simple and secure authentication device [5].

Typical implementations of Captive Portals rely on an Authentication, Authorization, and Accounting (AAA) server for verifying identities of subscribers, and on a user database for storing subscribers' credentials and users' profiles. All authentication mechanisms are implemented at network and application layers, and they are very implementation dependent. No completely standard mechanism is currently defined, but a rather common approach is to use simple web-based interfaces between the user (MT) and the access system.

In some cases, the authentication procedure begins when an un-authenticated user starts his/her web browser attempting to browse to any web page. At this point, the HTTP request is redirected to a new HTTPS URL, corresponding to a web page on a remote web server asking for the user authentication. Through a secure web interface the user is asked to enter his/her credentials (e.g. login name and password) corresponding to a known authentication realm/domain. After the

credentials submission, usually via HTTPS/TLS with certificate provided by the server itself, the server can directly perform authentication and authorisation or, better, it can act as an AAA client to perform the subscriber's authentication and authorisation with a remote AAA server (for example using RADIUS or Diameter).

If the authorisation procedure successes the Captive Portal opens and configures appropriate firewall rules, activating some user's privileges, such as network connectivity or access to particular services (for example to specific machines and ports), or advanced network services (such as more bandwidth, quality of service, traffic encryption, etc.). The firewall rules are completely implementation dependent, however a common approach is filtering packets based on layer two and layer three users' identifiers (MAC and IP addresses).

Figure 3 shows the basic scheme for a Captive Portal.

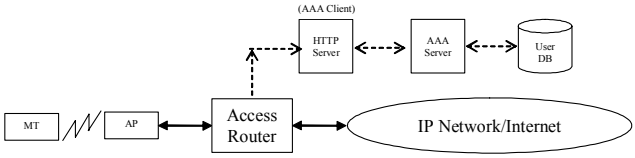

Fig. 3. Authentication architecture for AA with captive portals

The firewall rules on the access router normally remain active for the authenticated user until he/she disconnects from the network or based on specific policies. Normally Captive Portals periodically check validity of the authentication of currently allowed users, for example by reissuing the authentication request. In some cases, an active script (e.g. JavaScript) is left open on the user terminal in order to automatically refresh the login credentials and keep the connection open.

The main advantages of the Captive Portals approach are:

-- it is *plug&play*, it can be used with standard terminals;

-- it is *simple*, it is completely software implemented and no specific hardware is required; moreover, it can be based on well known application level technologies;

-- it is *flexible*, it can implement a wide range of authentication-required and non-authentication-required services.

On the other hand, its main disadvantages are:

-- it is not a completely standard mechanism; several implementations exist;

--the confidentiality of user traffic is rarely guaranteed;

-- it does not permit interaction with the terminal configuration and does not allow the personalization of the host configuration parameters as function of the authentication result and user credentials.

III. Proposed Mechanism for Secure WLAN-Based Access

In this section an *open* architecture for secure authentication and data transfer in a WLAN-based access scenario is proposed. The proposal addresses the generic scenario in which a user needs to be authenticated in a wireless (LAN or PAN) access network, and a secure data communication needs to be established between the user terminal and the network.

The proposed solution is based on a standard AAA infrastructure, simple DHCP (Dynamic Host Configuration Protocol) dynamic functionality, and IPSec securing communications. It is independent of any layer-two mechanism, since it enforces confidentiality and access control directly at the IP layer through IPSec security and packet filtering functions. Hence it can be used with any wireless LAN or PAN access technology (IEEE 802.11, Bluetooth, or whichever new incoming technology). Moreover the IPSec encapsulation and security mechanisms easily allow the extension to new encryption algorithms and the selection by the wireless access provider of the most suitable one, without changing the access network hardware infrastructure.

The only potential disadvantage of the proposed approach consists in the increase of processing capability required in the access router implementing IPSec functions. However, the increase can be easily managed through the enhanced CPU processing capability of current high-speed routers and the hardware implementation of routing and security functions (e.g. by using hardware accelerator cards).

The basic idea behind our proposed architecture is quite simple. To control the access of wireless terminals into a local site and to the Internet, a MT starts a normal DHCP configuration procedure with a local DHCP server. The DHCP server, acting as an authenticator, issues an EAP-based authentication and authorisation procedure through which i) both MT and the network can be mutually authenticated, ii) MT can be authorised and iii) a new security association can be established between MT and the network consisting in fresh keying material used to encrypt and authenticate data exchanged between MT and the access router.

DHCP [6] is an UDP-based client/server protocol used to dynamically assign IP addresses and to bind several network parameters by local DHCP servers to fixed or mobile client hosts for some periods of time. DHCP operates with a four-way handshaking procedure that uses the following messages: i) DHCP DISCOVER sent by MT in broadcast on its local physical subnet looking for a local DHCP server, ii) DHCP OFFER sent by one or more servers offering a new configuration valid in the visited network, iii) DHCP REQUEST sent by MT selecting the desired server and configuration, iv) DHCP ACK sent by the selected server containing the configuration parameters for the requesting client. If the selected server is unable to satisfy the DHCP REQUEST the server responds with a DHCP NACK message.

In the proposed architecture, the authenticator (i.e. the DHCP server) uses a backend authentication server (AS), for generating new per-session keys, and for maintaining accounting information. Hence, the authenticator acts as a pass-through agent forwarding EAP authentication messages back and forth between the peers (MT and the backend AS). Such exchanged messages are encapsulated within the proper AAA protocol, such as RADIUS or Diameter. Separation of the authenticator from the backend AS simplifies credentials management and policy decision making.

When the MT is correctly authenticated and authorised, the authenticator properly configures the access router with the session key and filtering rules. Per-session keys are derived from the authentication material and independently generated

by the MT and the AS (hence no keying material is exchanged between MT and the network).

Fig. 4 shows the main components of our reference scenario.

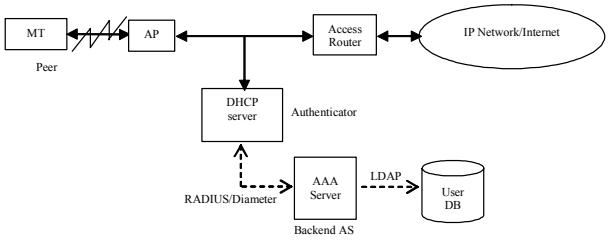

Fig. 4. Proposed reference scenario

As authentication method the Authentication and Key Agreement (AKA) protocol can be used [7]. AKA is a challenge-response mechanism for mutual authentication and session keys generation defined by 3GPP (Third Generation Partnership Project) for 3G mobile networks, and used both for radio network authentication and IP multimedia service authentication purposes.

Fig. 5 shows the basic AKA authentication procedure.

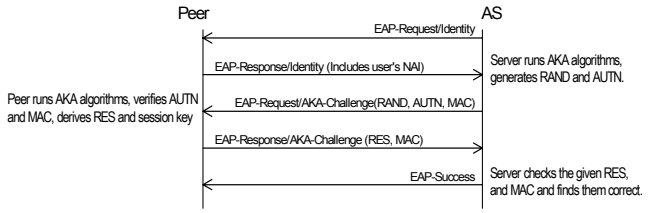

Fig. 5. AKA authentication procedure

The configuration and authentication procedure is summarised in Fig. 6. The procedure starts when the MT tries to dynamically configure itself in a new visited network. This could happen, for example, when the user attempts to browse to any web page, to send an e-mail message or to transfer a file from/to a remote site. At that time, the DHCP client in the MT starts a standard DHCP session by broadcasting a DHCP DISCOVER message. On receiving the message, the DHCP server starts the authentication procedure by sending a request for identity encapsulated in a *EAP Request* message (*EAP-Request/Identity*) to the MT. An authentication agent running on the MT receives the EAP-Request/Identity message and replies with an *EAP-Response/Identity* message containing the requested user identifier. This identifier can be the MAC address of the user's terminal, or a NAI (Network Access Identifier) [8]. This initial identity request/response message exchange can be omitted in case the user identifier is derived directly from the DHCP DISCOVER message (for example, in case the MAC address or a NAI DHCP option is used). After obtaining the user identifier the DHCP server sends it to the backend AS though a proper *AAA Request* message.

The AS, based on the user identity and on a user sequence number (SN), generates a new authentication vector *(RAND, AUTN, XRES, CK, IK)* formed by: a random number RAND, a network authentication token AUTN, an expected result XRES, a session key for integrity check IK, and a session key for encryption CK. At this point the AS starts the AKA protocol by sending an *EAP-Request/AKA-Challenge* message encapsulated in a *AAA Response* to the DHCP server. EAP-AKA packets encapsulate parameters as attributes, encoded in a <*Type*, *Length*, *Value*> format. The *EAP-Request/AKA-Challenge* message contains the attributes RAND and AUTN, and a message integrity Check (*MIC*).

On receiving the *EAP-Request* the DHCP server simply relays the message to the MT. The MT runs the AKA algorithm and verifies the AUTN value by using the secret key K and the sequence number SN. If this is successful, the peer is talking to a legitimate AS and can proceed sending the *EAP-Response/AKA-Challenge* message. This message contains a result parameter RES, allowing the AS in turn to authenticate the MT, and a *MIC* attribute to protect integrity of the EAP message.

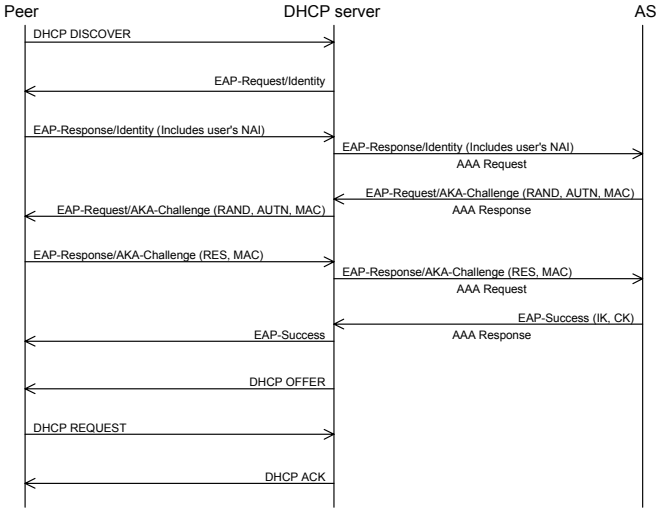

Fig. 6. Proposed authentication procedure

The message is relayed by the DHCP server to the AS that verifies the correctness of RES and MIC values. In case of success, the AS sends the *EAP-Success* packet to the DHCP server, including derived keying material in the message. The DHCP server finally sends *EAP-Success* to the MT confirming the authentication success. Note that the DHCP server does not include any keying material along with the EAP-Success message since the MT can autonomously derive it from RAND, SN, and secret key information.

As soon as the MT is authenticated, the DHCP procedure can continue with the normal message exchange (OFFER, REQUEST, ACK). When the MT is authenticated and correctly configured in the new network, the DHCP server can remotely or locally configure the access router with the proper integrity and encryption keys, and filtering rules for the new MT. Furthermore, the DHCP server may optionally setup on the access router quality of service (QoS) configuration rules allowing a per-user traffic handling policy.

A. EAP over UDP

As already described, all authentication messages exchanged between MT and the authenticator, and between the authenticator and the backend AS, are carried by EAP [9]. The advantage of using EAP as a transport mechanism is its flexibility, since it can support multiple authentication protocols

(e.g. CHAP, OTP, TLS, AKA, etc.). Moreover, EAP natively enables the use of the backend AS.

EAP uses four different types of messages: EAP-Request, EAP-Response, EAP-Success, and EAP-Failure. EAP messages can be encapsulated into different layer-two protocols, such as PPP, IEEE 802.11 (EAP over LAN), etc., or into different application-level protocols, such us RADIUS or Diameter.

In the proposed architecture, EAP messages exchanged between the authenticator and AS are encapsulated within the proper AAA protocol (e.g. RADIUS), whereas EAP messages between the authenticator and MT can use either a layer-two or a transport/application layer encapsulation. As described in the next section, in our implementation we preferred using the latter approach and defining a new EAP over UDP (EAPoU) encapsulation mechanism. The main advantages of an UDP-based transport are the flexibility of using different layer two technologies, and the implementation simplicity both in the authenticator (i.e. the DHCP server) and the MT.

The EAPoU messages, encapsulated into UDP datagrams, are formed by the EAP packet leaded by an EAPoU header. The new EAPoU header is composed of two fields: *Client-Identifier* (CID) (variable size), and *CID length* (1 byte). CID is an unique identifier used by the authenticator to address the supplicant (i.e. the MT), whereas CID length indicates the identifier size (see Fig. 7). In our implementation CID is the client hardware address copied by the *chaddr* DHCP field [6].

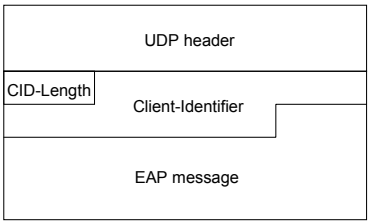

Fig. 7. EAP over UDP encapsulation.

The authentication procedure applies the first time the MT connects to the network and each time it tries to extend the lease time. In the former case, the MT communicates with the authenticator before the DHCP procedure is completed and when no IP address has been configured. In this case, the authenticator uses IP broadcast address (255.255.255.255) as the destination for all EAP Request and Success/Failure messages (with TTL equal to 1), while the MT uses address 0.0.0.0 as source for all EAP Response messages. In all cases, the MT exploits the CID value to properly recognise Request addressed to it.

IV. Testbed Implementation

The proposed architecture has been implemented in a demonstration testbed composed of an IEEE 802.11b wireless network directly connected to a Linux-based access router (AR) also hosting the DHCP server. For the sake of simplicity, also the AS has been co-located with the DHCP server, so no external AAA message exchange has been implemented.

Packet filtering in the access router is implemented through the standard Linux filtering support. A laptop PC with Linux OS is used as the MT. It is equipped with a WLAN card and hosts an AKA authentication client agent (AKA-AC), i.e. the authentication supplicant. The default DHCP client is used for the MT.

Fig. 8 shows the overall testbed layout.

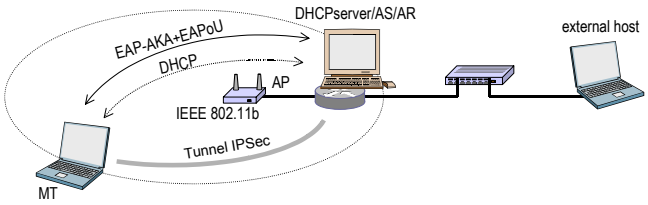

Fig. 8. Implementation testbed layout

According to the proposed architecture, each time the MT roams into the wireless network it issues a DHCP DISCOVER message. The DHCP server retrieves the user identity and passes it to the AS. The AS picks up a new authentication vector (RAND, XRES, CK, IK, AUTN) and sends it in a EAP-Request/AKA-Challenge message to the MT, so the authentication procedure continues. If the MT is correctly authenticated and authorised, a fresh keying material is independently generated (IK and CK) by the AS and MT, and the underling IPSec layer is configured. From this time on, all the IP-based communications are IPSec encapsulated providing both robust data authentication and confidentiality.

Fig. 9 shows the implementation architecture of the MT and AR/DHCP/AS. The entire authentication layer (AKA-AC and AS systems) and the DHCP server have been implemented in Java. IPSec functionality has been activated in both the MT and the AR, and the network layer is based on open source products (FreeSWan for IPSec and Linux *netfilter* for the AR packet filtering).

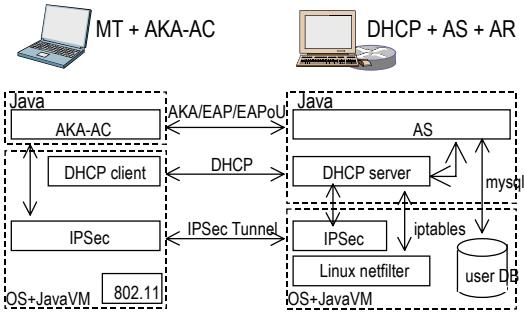

Fig. 9. Implementation architecture

V. Conclusions

We presented a quite simple solution to control and secure the Internet access of mobile wireless terminals roaming in a WLAN access network. The main design goal was to give the local provider simple mechanisms to authenticate the roaming users and provide secure communications in order to enable IP connectivity and application-level services.

Authentication is performed by the DHCP server in a link and application layer independent manner. This required adding some functionality in the software of the DHCP server, while keeping unchanged the DHCP client side in the wireless terminal population.

The main strength of our proposal is that it can be imple-

mented in each wireless provider site by only requiring the use of standard technologies and protocols (DHCP, RADIUS, EAP, IPSec) and very few changes in the software of the DHCP server. In addition, our authentication mechanism is transparent to either the specific link layer protocol (e.g. IEEE 802.11) or the design options of the wireless terminal's manufacturer. The only concern for the local site administrator is to provide the authorised user with an authentication application to run in the user MT as the supplicant side of the authentication procedure.

This approach has the advantages to be manufacturer-independent and to scale with the number of authorised users/terminals.

Another advantage of our architecture is that it does not require the un-authenticated user to start a specific application session to gain IP connectivity (browser pages/applets or pop-up windows). The authentication process is, in fact, triggered by any request for an IP address originated from the wireless terminal.

The main distinguishing feature of our architecture is the possibility for the DHCP server to *personalize* the configuration parameters according to the users' credentials and profiles stored in the user database. This possibility is not available in solution like captive portals or when the MT directly communicates with the access router. The server may perform the service personalization at various levels; for example:

-- *type of lease*: some time attributes could count for the time-of-day in which the access can be granted according to the user class (certain user can be allowed accessing only at a certain time of the day, for example);

-- *IP address*: according to the user credentials the network could assign private or public IP addresses to *allow* for limited or global IP connectivity;

-- *Default router and static routes*: they could be *set* according to the user class and privileges;

-- *Security level*: the *network* could personalize a per-user security level;

-- *Type of Service*: the access router could use this information to establish different QoS control mechanisms over the data packets.

-- *IP Telephony*: the network can configure the mobile terminal with a default outbound SIP *proxy* server and/or with a default media gateway, or the access router (acting as media gateway) can enable the media relay to/from external ISDN/PSTN/IP networks.

REFERENCES

[1] LAN MAN Standards of IEEE Comp. Soc., "Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specification," 1999

[2] W. A. Arbaugh, N. Shankar, Y. C. J. Wan, K. Zhang, "Your 802.11 Wireless Network has no Clothes", IEEE Wireless Communications, December 2002, pp.44-51

[3] IEEE 802.11i, work in progress 2003: "Draft Supplement to Standard for Telecommunications and Information Exchange Between Systems - LAN/MAN Specific Requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: Specification for Enhanced Security".

[4] Stephen Kent, Randall Atkinson, "Security Architecture for the Internet Protocol", Standards Track RFC 2401, IETF, November 1998.

[5] Captive Portals, NoCat, http://nocat.net,
Opengate, http://www.cc.saga-u.ac.jp/opengate/index-e.html,
OpenSplash, http://opensplash.qalab.com/

[6] R. Droms - "Dynamic Host Configuration Protocol", IETF Request For Comments, RFC 2131, March 1997.

[7] 3GPP TS 33.102, "Security architecture", Third Generation Partnership Project, Technical Specification, Release 6.

[8] B. Aboba, M. Beadles - "The Network Access Identifier", IETF Request For Comments, RFC 2486, January 1999.

[9] B. Aboba, L. Blunk, J. Vollbrecht, J. Carlson, H. Levkowetz, "Extensible Authentication Protocol (EAP)", IETF Request For Comments, RFC 3748, June 2004