SIP roaming solution amongst different WLAN-based service providers

Alessandro Ordine[(1)], Julián F. Gutiérrez[(1)], Luca Veltri[(2)]
(1) Dpt of Electronic Engineering – University of Rome "Tor Vergata", Italy
(2) Dpt. of Information Engineering – University of Parma, Italy

Abstract-.**Deployment of 802.11 Wireless LANs is increasingly on the rise leading to new service scenarios in which users are connected everywhere – everytime. However the IEEE 802.11 standard was designed for short range wireless data transmissions and does not natively provide any support for roaming amongst different access networks. In the more general case, a mobile user should be expected to be able to roam into a visited domain and gain access to the network on the basis of some credentials shared with his home domain or WISP. There are several mechanisms that can be involved in providing such access control and roaming functionality but no any standard has overcome. In this paper[1] a new SIP based solution is proposed. SIP-based authentication is provided end-to-end between user-to-network and network-to-network. The proposed solution realizes full proxy-to-proxy authentication at SIP level, enabling dynamic and secure WISP-to-WISP interworking. The proposed solution has been also implemented and successfully tested in a demonstrating testbed.**

I. INTRODUCTION

Wireless LANs (based on IEEE 802.11 [8]) in the last years are experimenting widespread deployment in the modern countries' markets. The easiness of deploying and using a wireless network, and the deployment cost (very low), have been critical factors in the success of such technology.

This generates a new and never seen before scenario in which users are connected *everywhere – everytime*. That's a logical consequence of the wireless phenomenon, as it has changed several things in people's lifestyle, such as the way people work (for example connected to Internet or to their office through a WLAN hotspot in an airport or in a train station), or how they live their leisure time (videoconferencing, instant photo or music sharing, network gaming, etc.). Obviously this new scenario represents a new challenge for the professionals of the communication at all levels as wireless technology provides several advantages, but it also represents some new problems never seen before.

The IEEE 802.11 standard was designed for short range wireless data transmissions. Although a network can be deployed by several access points, this restriction converts a wireless network in a LAN, with no possibility to be directly extended to a layer-two MAN or WAN. A chance to achieve such objective could be to consider different WLAN-based access networks belonging to different ISP (WISP) interconnected together via an IP-based backbone (or the big Internet). In order to realize such scenario, a mechanism for inter-provider roaming (as already deployed for cellular networks) is therefore needed. In a inter-provider roaming scenario a user that roams into a visited domain, can gain access to the visited network (i.e. administrated by a foreign WISP) by authenticating with his/her own credentials shared with his/her home domain/WISP.

There are several mechanisms that can be involved in providing such access control and roaming functionality and there is not a prevailing standard way to perform it. Some mechanisms are specific to the wireless access technology (layer-two mechanisms); other mechanisms are implemented at the IP network layer (VPN/IPSec-based mechanisms). However, currently the most common way to offer WiFi "public services", providing authentication and authorization functionality is by using some upper layer (application level) mechanisms. Usually an authentication gateway/access server is placed between the wireless access and the services. Layer-two access is not controlled, but users' terminals are forced to authenticate against the gateway/access server before receiving service grants. 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 very implementation dependent.

Typical implementations of captive portals rely on a web-based approach. The authentication procedure begins when an un-authenticated user starts his/her web browser attempting to browse to a generic 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 acting as authentication server (AS) and 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 authorization or, better, it can act as an AAA client to perform the subscriber's authentication and authorization with a remote AAA server (for example using RADIUS or Diameter).

If the authorization procedure succeeds 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

[1] This work has been supported by the Italian Ministry for University and Research (MIUR) under the PRIN project TWELVE (http://twelve.unitn.it)

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).

Although simple and flexible, this method can sometime suffer of some security limitations. Moreover, since it is based on HTTP interface, it is not very tight to emerging real-time multimedia applications.

In this paper a new SIP based solution is introduced. SIP-based authentication is provided end-to-end between user-to-network and network-to-network (in case of different ISPs are involved). The proposed solution does extend the base SIP authentication procedure by realizing full proxy-to-proxy authentication at SIP level, enabling dynamic and secure WISP-to-WISP interworking. The proposed solution has been also implemented and tested in a demonstrating testbed, within the framework of the TWELVE research project [9]. The implementation has been based on a captive portal roaming solution, named "Uni-Fy" , and used within the same project.

As for the organization of the paper, in Section II we present three different SIP authentication methods. The base Uni-Fy gateway is described in Section III, while in Section IV the proposed solution is described. In Section V we discuss some security considerations. Finally in Section VI conclusions and possible future works are presented.

II. SIP AUTHENTICATION OVERVIEW

SIP, the Session Initiation Protocol, is a signaling protocol for Internet conferencing, telephony, presence, events notification and instant messaging. SIP was developed within the IETF MMUSIC (Multiparty Multimedia Session Control) working group, with work proceeding since September 1999 in the IETF SIP working group. It is specified in RFC 3261 [1], and successive related RFCs.

SIP authentication model is inherited from the HTTP digest authentication (RFC 2617 [2]). Originally also the HTTP basic authentication was supported by SIP, but it has been deprecated by the RFC 3261, due to the insecurity on sending the shared secret (i.e. the users' password) in clear within the request messages.

II.1 DIGEST AUTHENTICATION

Digest authentication follows a challenge-response scheme based on a shared secret key (password). In a SIP-based network, the authentication can take place between a user agent and a server (e.g. a registrar server or an intermediate proxy).

A User Agent (UA), in the following also referred as users' terminal, represents one of the communication's endpoints. It can be a hardware or software device with the capability of initiating or receiving a call based on SIP signaling. A terminal can work either as a server (UAS), if it receives the call, or as a client (UAC) if it initiates the call. Authentication is normally requested by the UAS (to be sure on the UAC identity before processing its requests) or also by the UAC (in such case, mutual authentication is provided). SIP Digest Authentication procedure is based on a four-messages exchange. Fig. 1 illustrates a successful authentication procedure.

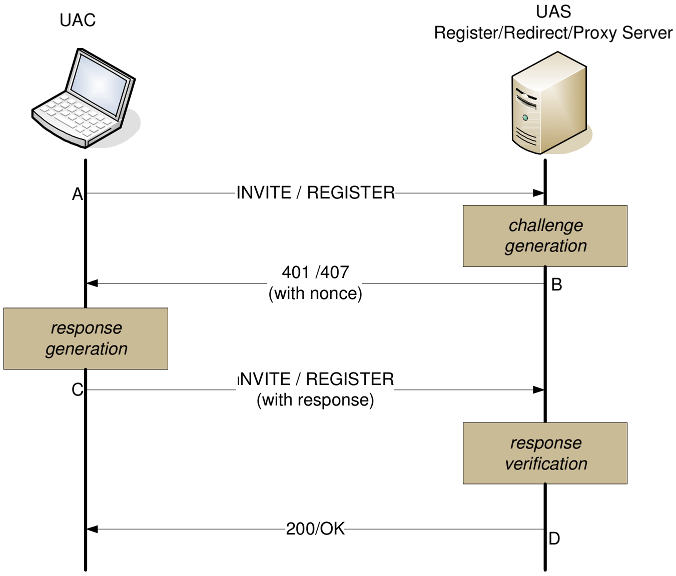

Fig. 1: SIP Digest Authentication Scheme

The first of these messages (A) is sent by the UAC, and it's a SIP request . Usually, this request is an INVITE message (if the UAC is trying to establish a new session) or a REGISTER message (if updating the location information). Often this first request doesn't include any authentication information nor credentials, since the other side has not challenged the UAC yet. The request is received by a UAS, that can be a registrar, a redirect or a proxy server. A registrar is a server that accepts REGISTER requests and uses the information provided by successful registration requests to update location information within a particular domain it is responsible for [1]. A redirect server is a UAS that generates 3xx responses to requests it receives, redirecting the UAC to contact an alternate set of URIs [1]. A proxy server receives SIP requests and forwards them on behalf of the requestor [1]. After generating a challenge for the UAC, these entities send the second message of the process (message B), containing the challenge for the UAC. The challenge is composed by different parameter such as the realm, authentication method, algorithm, and nonce. This response message will be either an *Unauthorized* response message containing a WWW-Authenticate header if sent by UAS, registrar or redirect server (401 response code), or a *Proxy Authentication Required* response message containing a Proxy-Authenticate header if sent by a proxy server (407 response code).

Once the UAC receives the challenge it calculates the response. This response is computed using the algorithm specified within the request (usually the MD5) with parameters such as nonce, the shared secret key, user-name and some others [1]. Once the response to the challenge has been computed, a new SIP request (message C, that is the same method of message A), is sent by the UAC, including now a response parameter, which is actually the response to the challenge received within the previous message.

After receiving this new request, the UAS checks if the received challenge response equals the expected value (i.e. the value obtained by calculating response by means of the locally stored user's shared secret). If it succeeds, the UAC is successfully authenticated and the proper response message corresponding to the UAC request is sent by the UAS (message D).

II.2 AKA AUTHENTICATION

The previous procedure such as defined in the RFC 3261 is referred as SIP Digest authentication procedure.

The authentication framework has been also extended by the Third Generation Partnership Project (3GPP) in order to be use and interoperate with the 3G systems. The authentication mechanism used in 3G networks is the Authentication and Key Agreement procedure (AKA). AKA is basically a challenge/response authentication mechanism that naturally provides mutual authentication (user-to-network and network-to-user), and roaming facilities. In order to let the SIP-based signaling platform to inter-operate with 3G systems the SIP AKA [5] authentication method has been defined. In our proposal we, as it will be described in the following sections, we use AKA as base authentication mechanism also for WLAN-based access systems. The main benefit of such approach is obviously the reusability of the 3GPP infrastructure and the compatibility between the new WLAN access networks and already development 3G systems.

Authentication in 3G lays on two different keys called K_U and K_I that are stored both in the client's SIM card and in the Home Subscriber Server (HSS). The SIM card consists of two virtually-different modules: USIM (UMTS-SIM), containing K_U, and ISIM (IP multimedia SIM), containing K_I.

As already explained above, AKA is a challenge-response mechanism that provides both mutual authentication and session keys generation. AKA is used both to authenticate the radio network as well as the IP services associated to 3G (IP Multimedia Subsystem - IMS) [4]. The authentication procedure in 3G is represented in Fig. 2.

When a node on behalf of the 3G network wants to authenticate a mobile user it use a set of authentication information called Authentication Vector (AV). An AV consists of a set of parameters required to achieve a successful authentication procedure, and is provided by the HSS based on the secret key and on a sequence number SQN.

An AV is composed of: RAND, a random challenge called nonce when talking about the Digest procedure; AUTN, a token used to authenticate the network towards the client; XRES, a expected result (the expected response to the challenge from the client); IK, a session key for integrity check; and CK, a key used for encryption that differs each time an authentication procedure is required.

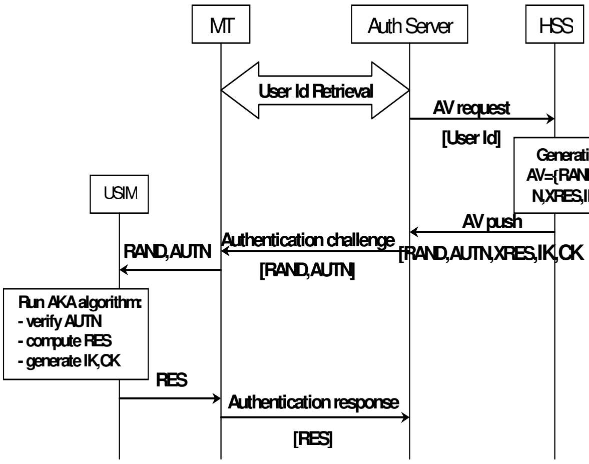

Fig. 2: AKA successful procedure

When requested, a new AV is passed to the authentication entity/node (the authenticator) in the network and is used to challenge a new supplicant MT. Once the authentication request arrives to the USIM module of the MT, the USIM uses the secret key K and the sequence number SQN to verify the AUTN. If both AUTN and SQN are valid, the network has been authenticated and the user-to-network authentication procedure proceeds. The USIM module calculates the keys IK and CK (that will remain local, this means they will not be sent to the network) as well as the authentication result (RES). RES is sent to the authenticator to be compared with XRES. If they match, the MT/user is authenticated and further communications will be protected with IK and CK keys.

II.3 DIGEST – AKA AUTHENTICATION

This AKA mechanism can be combined with the Digest one, originating the *Digest AKA authentication* scheme [3]. One of the main feature of this mechanism is that it lets the UAC check the identity of the network; this is done bye UAC extracting the AKA RAND and AUTN parameter values from the nonce parameter of the proper authenticate header field.

This parameter is formed as base64 encoding of RAND, AUTN, and server-data, i.e

nonce=base64(RAND,AUTN,server-data)

where server-data is some optional data provided by the UAS. As already explained, the supplicant (i.e. the 3G MT and/or the SIP UAC) uses those parameters to authenticate the network.

After extracting the network authentication token, AUTN, and checking that it could only come from a valid network, the UAC proceeds to calculate the answer to the challenge (actually the RAND value) based on the shared secret (K). Another main benefits of the AKA mechanism is that it generates a sort of password for every authentication procedure. This password is the RES value and it is provided by the HHS within the AV and computed separately by the UAC and by the HHS based on RAND and K values (RES=f(RAND,K)).

This RES parameter is used to form the digest response parameter. Such response parameter can be calculated for example as MD5 of a set of authentication parameters as

$$digest\text{-}resp=MD5(MD5(username:realm:f(RAND,K)):\\ :base64(RAND,AUTN,sd):MD5(A2))$$

where the operator ":" just indicates a concatenation; f(RAND,K) is the unique password for each request; sd is the server data described above and A2 is a string whose value is depends on the value of a parameter called qop (refer to [1] for further details).

The whole authentication process is shown in Fig. 3.

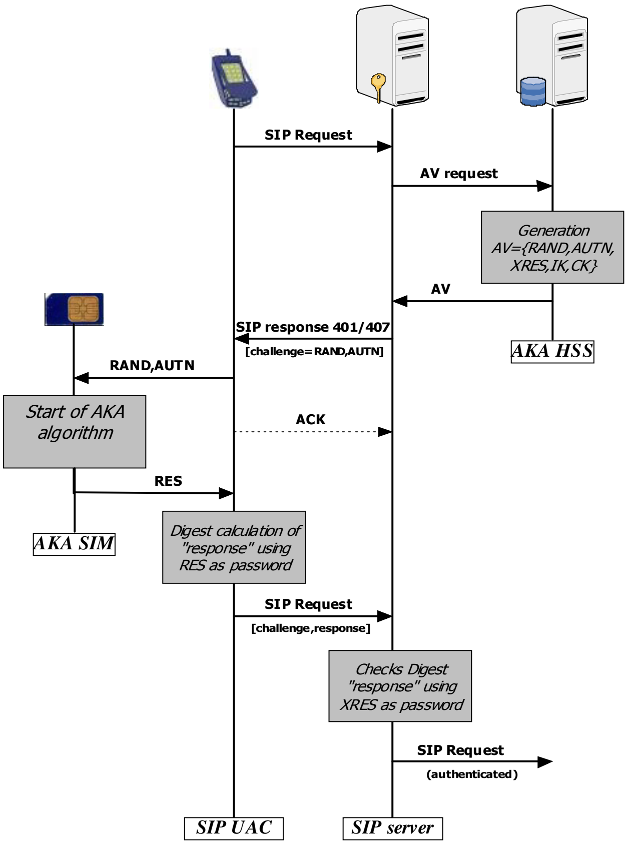

Fig. 3: AKA Digest Authentication Scheme

III. UNI-FY OVERVIEW

The proposed solution, described in Section IV, is based on a distributed access control system used within the TWELVE research project, and called Uni-Fy [6]. In this section a simple overview of the Uni-Fy system is provided.

Uni-Fy is a Wireless LAN/HotSpot management system with distributed authentication, access and policy control, and other capabilities. Authentication and authorization functions are implemented at application layer, while access control is applied at IP layer by means of firewalling capability. The overall scheme can be view as a captive portal implementation. Fig. 4 shows the Uni-Fy access model.

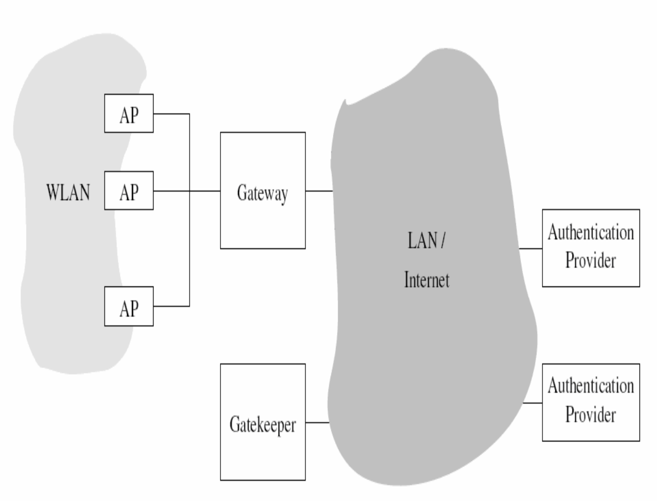

Fig. 4 Uni-Fy basic Scheme

An Uni-Fy access domain comprises four main entities:

- an access network through which mobile users can attach and gain connectivity toward the rest of the IP network (e.g. Internet) after being successfully authenticated with the system; such network is normally formed by one or more IEEE 802.11 WLAN, but can also be formed by wireless network providing connectivity via Bluetooth (PAN) or WiMAX (MAN) technologies, or/and by an Ethernet wired LAN;
- one Gateway, that acts as access router for the access network; it is the point of attachment of the access network to the rest of the IP network, and it is connected with the various wireless access points directly or by means of a wired/wireless LAN infrastructure; it basically implement packet filtering functionality; its main goal is to enforce the policy rules dynamically setup by the Gatekeeper; hence the Gateway is the Uni-Fy Policy Enforcement Point (PEP);
- one Gatekeeper, that is the system that, together with the Gateway, enforce authentication procedure before granting access to roaming users; it works at application level redirecting specific application sessions to a proper authentication server; together with the Gateway (that works at lower level) is the core of the captive portal system;
- one or more Authentication Provider, directly or indirectly trusted by the Gatekeeper, and to which the Gatekeeper redirects application sessions in order to force a proper authentication procedure; their implementation strictly depend on the specific application supported for authentication purpose; it can be formed by a application server (HTTP server, SMTP server, SIP registrar, others) and optionally by a backend authentication server (an AAA server such as a RADIUS or Diameter server) and an LDAP or DB repository; the result of the authentication and authorization functions is communicated back to the Gatekeeper in a application-dependent manner and it is used by the Gatekeeper (i.e.

the Policy Decision Point - PDP) to grant access to the roaming users through the Gateway.

The Gateway and the Gatekeeper can be co-located or can be implemented on different nodes. In the former case all traffic redirected by the Gateway to the Gatekeeper is passed back to the Gateway before being relayed toward the proper Authentication Provider. Fig. 5 shows the functional architecture of the Gateway- Gatekeeper regardless the are co-located on the same node or split into two different systems. All traffic from/to the access network is handled by the router and packet filter modules of the Gatekeeper. Packets are classified , which identifies them based on their MAC and IP addresses and transport protocol, according to per-user access control lists (ACLs) dynamically configured by the Gatekeeper.

Only traffic from/to authenticated users have a proper ACL while all other traffic incoming from the access network is redirected to the Gatekeeper.

To process the different packet types that it receives, the Gatekeeper uses different plugins. These plugins are just modules contacted via a dispatcher (element that receives the incoming packet and searches for a suitable plugin to process it). Plugins act as local Application Level Gateways (ALGs) processing incoming packets according to known protocol types.

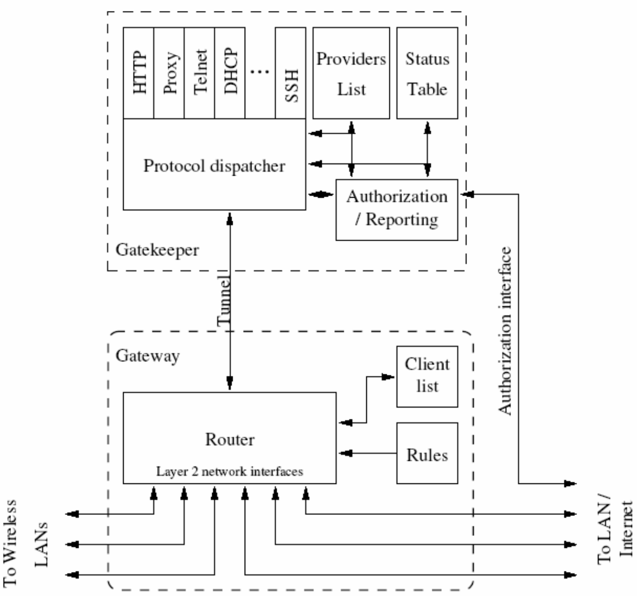

Fig. 5: Uni-Fy's functional architecture

According with the system architecture just described above, the entire access control system can be split up into two main components: i) the Access Provider that is composed by a local access domain, formed by the wireless/wired access network, and by the pair Gateway-Gatekeeper, ii) the Authentication Provider, composed by one or more application-specific authentication servers and optionally by a backend AAA server.

The Access Provider and the Authentication Provider can be also administrated by two different entity. Moreover, according with the Uni-Fy architecture, the same Access Provider can trust and rely on different Authentication Providers (probably one local Authentication Provider and zero or more trusted remote Authentication Providers).

IV. SIP-BASED AUTHENTICATION METHOD

As already said, one of the most common approach for handling with roaming amongst different wireless ISPs is the use *captive portal* mechanisms. In this section we describe a new proposal of a captive portal like mechanism based on the SIP authentication procedure (Section II) and on an access control scheme based on the Uni-Fy architecture (presented in Section III).

In terms of inter-ISP roaming, the relation between the ISP providing network access (the Access Provider) and the ISP providing authentication functionality (the Authentication Provider) can be one of the following:

1. The Authentication Provider can be a different administratively entity, separate from the Access Provider, with a strict trust relationship with it; different Access Providers can relay one the same centralized Authentication Provider; such scenario can be further extended by a centralized hierarchical authentication architecture;
2. Each Access Provider implements a local Authentication Provider that shares with the other trusted Access Providers; in this case the various ISP (acting as both Access Provider and Authentication Provider) will form a kind of web of trusted-ISP; the users when accessing through an Access Provider can choose their own Authentication Provider with which try to authenticate.

The idea proposed in this paper is to realize such web of trusted-ISP by means of the same signaling platform used for multimedia real-time service and used by 3G mobile networks.

When a mobile user roams into a new visited network it try to register with his/her own SIP registrar server (acting as home registrar or *Home Authentication Provider*). Such procedure is intercepted by the local Gatekeeper (the access controller) administrated by the visited ISP and redirected to the Home Authentication Provider opportunely modified with ISP-to-ISP authentication and authorization capabilities, according to the architecture described in the previous section.

In order to assure ISP-to-ISP authentication and correct authorization information retrieval from the Home Authentication Provider (i.e. the remote SIP registrar server), an extension of the standard UAC-to-UAS SIP authentication procedure is proposed and has been implemented.

Two new header fields allowing authentication between two intermediate SIP entities are here defined: Proxy-To-Proxy-Authenticate (shortly referred in the following as *pp-*

authenticate) and Proxy-To-Proxy-Authorization (shortly referred in the following as *pp-authorization*).

According with the standard SIP authentication procedure, the *pp-authenticate* header is used to carry authentication request information, while *pp-authorization* header is used to carry authentication response information.

The *pp-authenticate* header is used by a generic intermediate proxy to authenticate a next-hop proxy or next-hop UAS, in order to correctly trust information sent as response from such next hop entity. The *pp-authenticate* header is inserted by the proxy within a proxing SIP request message, while the *pp-authorization* is inserted in a SIP response message by the next hop entity in response to the *pp-authenticate* request.

The authentication method used with the *pp-authenticate* and *pp-authorization* can be anyone of the already proposed authentication methods in SIP, without any restriction, and is selected by the intermediate node that starts the proxy-to-proxy authentication procedure.

Let us now consider how the new SIP extension applies to a SIP-based access control scenario. Let us consider a mobile user that roams into a new visited network and tries to register his/her UA with his/her own home registrar server within the home network/ISP. Let us consider the case in which there is a trust relationship (with proper authorization and roaming policy) already established between the home ISP (acting as Authentication Provider) and the visited ISP (acting as Access Provider). The register request sent by the UA is then intercepted by the gatekeeper in the visited network and forwarded to the user's registrar server.

Fig. 6 shows the complete registration and authentication procedure exchanged between the user and the Access Provider and between the Access Provider and the Authentication Provider.

When the registrar server receives the first register request sent by the UA it starts standard UAC-to-UAS authentication procedure, by sending a 401 Unauthorized response message containing a WWW-Authenticate header with the authentication method (AKA is expected to be used) and the challenge, as described in section II.1 and II.3. The message is transparently forwarded to the UA.

When receiving this response the UA sends a new register request populated with an Authorization header with the proper authentication challenge response. When intercepting this authenticated register request, the gatekeeper starts a new proxy-to-proxy authentication procedure attempting to challenge the remote registrar.In the register request a new *pp-authenticate* header field is added with realm, algorithm, username, nonce, method, uri and other parameters according to the selected authentication method used for the proxy-to-proxy authentication. Although any authentication method can be used, in the rest of this section a Digest authentication is supposed to be used.

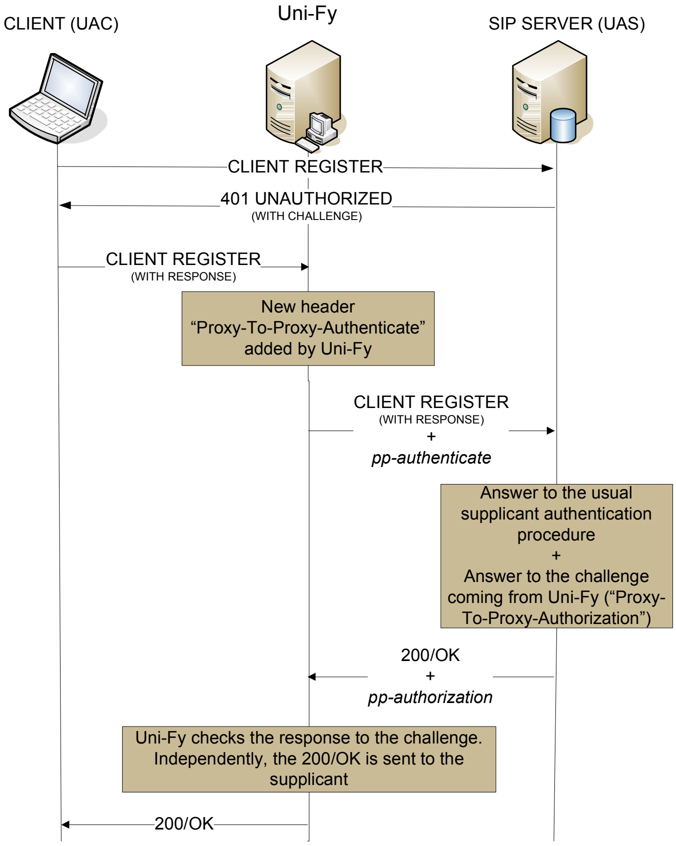
CLIENT (UAC)
Uni-Fy
SIP SERVER (UAS)
CLIENT REGISTER
401 UNAUTHORIZED
(WITH CHALLENGE)
CLIENT REGISTER
(WITH RESPONSE)
New header "Proxy-To-Proxy-Authenticate" added by Uni-Fy
CLIENT REGISTER
(WITH RESPONSE)
+
pp-authenticate
Answer to the usual supplicant authentication procedure + Answer to the challenge coming from Uni-Fy ("Proxy-To-Proxy-Authorization")
200/OK
+
pp-authorization
Uni-Fy checks the response to the challenge. Independently, the 200/OK is sent to the supplicant
200/OK

Fig. 6: Proposed Authentication Scheme

The receiving registrar server (the UAS), according to this procedure processes both the *Authorization* and the *pp-authentication* header fields (the former for user authentication, the latter for proxy-to-proxy authentication). For the latter, a new *pp-authorization* header field is added into the registration response generated after the user authentication has been performed (a 200 OK response is sent in case the authentication process succeeded).

This *pp-authorization* header field should include, at least, the computed response to the challenge sent, together with the other parameters sent with the *pp-authenticate* header field.

Once the gatekeeper receives such message with the *pp-authorization* header (the fourth message) it checks if the new response match with expected result that is locally calculated based on the secret shared between the Access Provider and the remote Authentication Provider (i.e. the registrar server). If it succeeds, and if the response code sent to the UAC from the UAS is a 200/OK code, the Gatekeeper updates its authorization table, changing the status of the new user/terminal to "AUTHORIZED". If the authentication verification fails, the status is changed to "FORBIDDEN". Fig. 7 represents this operation.

The whole authentication and authorization scenario has been implemented in a testbed, based on the Uni-Fy access control mechanism described in the previous section. The Gateway/Gatekeeper nodes have been realized based on the

Uni-Fy implementation provided in the TWELVE project framework [5]. The Gatekeeper plugin for SIP has been developed in C++ based on the reSIProcate C++ SIP stack library [5].

Finally, the registrar server, opportunely extended with proxy-to-proxy authentication has been implemented in Java, based on the mjsip SIP stack library [9].

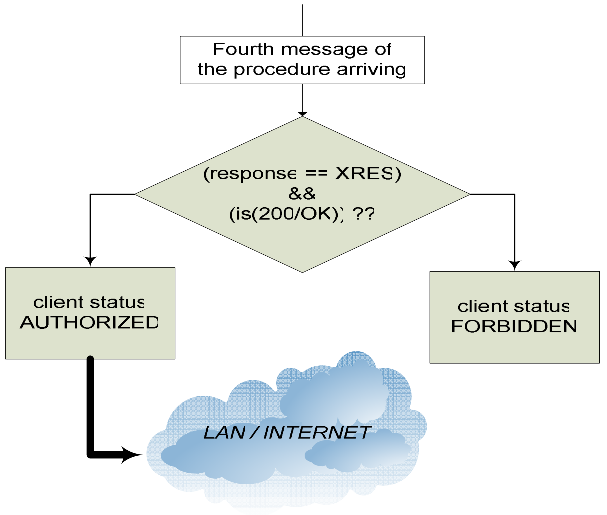
Fourth message of the procedure arriving
(response == XRES) && (is(200/OK)) ??
client status AUTHORIZED
client status FORBIDDEN
LAN / INTERNET

Fig. 7: Client's state transaction in Uni-Fy

V. Security considerations

A deep analysis of all possible security threats on the proposed access solution is out of the scope of this work and will be the objective of further works. However some considerations are hereafter discussed.

As far as a strong security mechanism is not implemented within the access network at layer 2 (e.g. 802.11i with AES) or at layer 3 with IPSec, different attacks can be mounted against both user data confidentiality and access control. For example, as far as access control at data plane is performed on the basis MAC/IP address matching, such service can be attacked by mounting a DoS attack toward a legitimated terminal and by reusing the same MAC/IP addresses. The use of short re-authentication timeouts aims to mitigate such type of attacks.

Regarding the proposed SIP authentication procedure based on the new Proxy-to-Proxy-authentication, the same considerations and limits stood out for the standard UAC-to-UAS authentication and described in RFC 2631 (Section 23) are still valid.

VI. Conclusions And Future Work

In this paper we addressed the inter provider roaming problem and we proposed a simple authentication procedure based on SIP. Our solution can coexist with the captive portal technique, providing a simple and powerful roaming mechanism for SIP-compatible and 3G users.

With the proposed solution, a client will be able to access several WLANs based on the use of standard VoIP or instant messaging UA. The authentication and authorization procedure is simply based on some credentials (normally a username and a password) shared with his/her home network/ISP, regardless the specific access network he/she is roaming to. As Digest AKA authentication scheme can be used, the authentication and roaming procedure is fully compatible with present and future 3G systems. This has a twofold advantage: first, any UA, capable to access to the user's USIM, is then able to access to any IEEE 802.11 administrated by its 3G operator, without any additional roaming procedure; second, the same UA can access any network trusted by his ISP/operator reusing his/her own credentials. Note that the overall access control procedure is independent from the specific access technology and can apply for IEEE 802.11 WLAN just as for Bluetooth or WiMAX based access networks.

Finally, the proposed solution has also been successfully implemented in a demonstrating testbed, covering all authentication, authorization, and policy enforcement functionalities.

References

[1] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M. and E. Schooler, "SIP: Session Initiation Protocol", IETF RFC 3261, June 2002.

[2] Franks, J., Hallam-Baker, P., Hostetler, J., Lawrence, S., Leach, P., Luotonen, A. and L. Stewart, "HTTP authentication: Basic and Digest Access Authentication", IETF RFC 2617, June 1999.

[3] Salsano, S., Veltri, L., and G. Martiniello, "Wireless LAN-3G Integration: Unified Mechanisms for Secure Authentication based on SIP", IEEE international Conference on Communications - ICC 2006, June 2006.

[4] 3GPP TS 24.229: "IP Multimedia Call Control Protocol based on Session Initiation Protocol (SIP) and Session Description Protocol (SDP)".

[5] 3GPP TS 33.102: "Security architecture".

[6] Lo Cigno R. et al, "TWELVE Test Bed and Demonstration Planning", http://dit.unitn.it/twelve/docs/TN-3.pdf

[7] "SIPfoundry reSIProcate: an rfc3261 sip stack", http://www.sipfoundry.org/reSIProcate/

[8] "IEEE 802.11, The Working Group Setting the Standards for Wireless LANs", http://grouper.ieee.org/groups/802/11/

[9] "TWELVE project homepage", http://dit.unitn.it/twelve/

[10] MjSIP – GPL open source SIP stack Java implementation, http://www.mjsip.org