Authentication and Authorization of mobile clients in public data networks

Prakash Reddy, Venky Krishnan, Kan Zhang, Devaraj Das
Mobile and Media Systems Laboratory
HP Laboratories Palo Alto
HPL-2002-213
August 2nd , 2002*

security, local wireless, nomadic

We present a protocol that enables mobile clients to be authenticated and authorized in data networks that are deployed in public places otherwise referred to as hotspots! The three key elements of a hotspot network are the mobile client, the hotspot server and the service provider. A mobile client is any device that can be used to access the internet. The hotspot server is a node in the data network that is a bridge between wireless clients and wired broadband network. The service provider is an entity that has an existing service relationship with the client and the hotspot server. The protocol discussed in this paper shows how three parties: Client, hotspot server and the service provider come together in a mutually un-trusted environment, authenticate each other and upon authentication exchange authorization tokens that are used in subsequent service requests. The most common use of this protocol is for clients to gain internet connectivity in public places, specifically in hotspots. The hotspot server provides the equivalent of cellular network roaming functionality. The service provider allows added features to its clients.

* Internal Accession Date Only
Approved for External Publication

Authentication and Authorization of mobile clients in public data networks

Prakash Reddy, Venky Krishnan, Kan Zhang, Devaraj Das

HP Labs
1501 Page Mill Road, Palo Alto Ca USA

Abstract. We present a protocol that enables mobile clients to be authenticated and authorized in data networks that are deployed in public places otherwise referred to as hotspots! The three key elements of a hotspot network are the mobile client, the hotspot server and the service provider. A mobile client is any device that can be used to access the internet. The hotspot server is a node in the data network that is a bridge between wireless clients and wired broadband network. The service provider is an entity that has an existing service relationship with the client and the hotspot server. The protocol discussed in this paper shows how three parties: Client, hotspot server and the service provider come together in a mutually un-trusted environment, authenticate each other and upon authentication exchange authorization tokens that are used in subsequent service requests. The most common use of this protocol is for clients to gain internet connectivity in public places, specifically in hotspots. The hotspot server provides the equivalent of cellular network roaming functionality. The service provider allows added features to its clients.

1 Introduction

Mobile access to the web is rapidly growing in popularity. You see people accessing the web at airports, coffee shops, malls etc. Internet connections are being made available in more and more public places. There are several reasons for the sudden growth of wireless access to the web in public places. One of the reasons is the standardization of the wireless local area network (WLAN) around the 802.11b. This standard referred to as Wi-Fi operates at speeds up to 11 Mega Bits per second. As a result of this standardization there has been an explosion of wireless devices equipped with WLAN access cards. The cost of wireless networks is coming down and they are easier to set up than a wired network.

By year-end 2007, 60 percent of the population in the U.S. and Europe will carry wireless devices, according to a recent Gartner report. By 2010, the percentage will leap to 75 percent.

Even though the public wireless networks are seeing rapid growth, they are not in the mainstream yet. The public wireless networks are mushrooming along different lines a) An individual deploying their own access point and making it available for public use b) Companies trying to build their own infrastructure to provide access for fees

and c) A few other companies are trying to build a virtual network by aggregating existing public wireless networks d) A true roaming model. If we examine the evolution of cellular networks, each carrier started out by building their own network and servicing their own customers. If the customer were to travel outside the providers network, the service was not available. The carriers soon realized that they could make their network appear wider and generate more revenue by setting up service level agreements (SLA) to allow customers to access each other's networks leading to the birth of roaming. The carriers had to figure out how to authenticate, authorize and bill each other's customers.

We see a similar model developing with perhaps one major difference. As the cost of deploying a public wireless network infrastructure is considerably smaller than when compared to deploying a cellular infrastructure, we anticipate lot more infrastructure providers in this space.

Given this model, we need an authentication and authorization (AA) model that can deal with the three parties that are involved in connecting a mobile node to the public wireless network.

Current schemes do not span trust boundaries, because WLAN/LAN infrastructure providers from unrelated enterprises have not established a mechanism for authentication and authorizing a mobile node. This paper defines an AA protocol which can be deployed in a public wireless networks.

In section 2 we will introduce the terminology and we will discuss the AA schemes currently being used. Section 3 will describe the network model and walk thru the steps of authentication and authorization and describe in detail the message sequence. We will walk thru the implementation details of the protocol in section 4. We will conclude by discussing the results and identifying future work.

2 Authentication and Authorization

Authentication and authorization are partly independent. Authentication is necessary before any authorization can happen. Authenticating clients in un-trusted domains requires adopting a scheme/protocol that will allow verifying the identity of the involved parties without revealing shared secrets or keys.

2.1 Definitions

Authentication is the act of verifying a claimed identity. It should be possible for the receiver of a message to determine its origin and an intruder should not be able to assume someone else's identity.

Authentication is a technique to ensure that the stated identity of the user is correct. As a first step, the entity interested in authenticating itself with some service will introduce itself and declare its identity. The service provider should be able to verify that the contacting party (sender) is the one it claims to be. The initiating party has to present some verification to prove its identity. The initiating party on the other hand would want to ensure the validity of the contacted party. The contacted party has to present some identification about itself to the initiator.

Upon successful authentication, the service provider (contacted party) is assured that the service is available only to users who have a right to access the service and the user can be sure that the correct service provider is being used.

Authorization is the process of determining if the presenter of certain credentials is authorized to access a resource or make use of a service. Typically a user is authenticated by the service and an authorization token is supplied to the user, which is used for any future service requests. The authorization token could be as simple as a random number, or could be encoded with other information like expiration time, users identity etc. In order for authorizations to be effective, they should be carefully constructed, and protected from intruders when services are requested. One can easily weaken a sound authentication scheme with a weak authorization mechanism. Authorization tokens should not be easily reproducible, they should be protected when making service requests and finally should minimize the time for which they are valid so as to prevent them from being used in replay attacks.

2.2 Evaluation criteria

We will be examining some of the AA schemes currently in use in public wireless networks. Before describing and analyzing these solutions, this section considers the various requirements that a viable solution needs to address.

- First and foremost, the authentication and authorization scheme needs to work at a practical level. From a users perspective it should be easy to use, perhaps non-intrusive.
- It needs to scale.
- The solution should allow for new service providers and infrastructure providers to join and participate in the emerging public network.
- The solution should facilitate access to services minimizing redundant interactions for authentication and provide a single-sign on regardless of the users primary service provider.
- It must support the need to access the service independent of the users physical location.
- Clients should be allowed to use multiple client devices to gain access.

From a service provider point of view, their customers should be able to access public networks that are their own or belong to those of its partners. This implies that once they have implemented the server side of the scheme, they would be able to expand/participate in the public network by either deploying their own infrastructure or by establishing SLAs with existing service providers.

The primary responsibility of the infrastructure providers is to verify the authenticity of the authorizations issued by their service provider partners prior to allowing the users access to services. In addition, they can keep track of usage statistics and present them to the users service provider for accounting/billing purposes.

2.3 Related work

In this section we will examine some of the AA schemes used in the public wireless networks. Any AA scheme has to be evaluated within the context of its operation. The context is defined to be the number of parties involved and the interaction needed to authenticate and authorize a user.

The first model we will examine is where an entity has deployed a WLAN in a public place. Any person with a mobile node entering the place has the ability to access the network provided the user has established a relationship with the owner of the network. For example this relationship could be in the form of registering with the provider or giving out the credit card at the time of access. It is quite possible to establish a relationship at the time of access by providing the proper credentials. Once a relationship exists, the user is required to access a particular web page using their browser. This page requires the users to authenticate themselves by entering their user id and the associated password. Any access to the network resources is prevented until the user has been authenticated. The authentication step needs to happen at each location the user wants to access the network. Lets examine this model by first breaking it into onetime steps and steps that are to be repeated each time.

Registration is a one time process and involves the following steps:

- The user must go thru a registration process with the provider and establish a user id and password. This may involve user modifying some settings on the mobile device.
- The user must know how to reach the web page that allows them to authenticate themselves.
- Authentication is a process that has to be repeated every time users enter a physical location where they would like to access the public WLAN. The process is made up of the following steps:
- When the user is interested in accessing the public WLAN, he or she must direct the web browser to the authentication page of the provider.
- Enter the user id and the associated password. Which would be sent to the provider's infrastructure and verified.
- Upon successful authentication, the user is allowed to use the network.

Table 1. shows the messages that are exchanged between the user and the network owner as part of the authentication process.

Mobile Client	Service provider
1. User access providers web page	1. Provider sends a authentication form
2. User enters user id and password	2. Provider verifies the presented credentials against its user database and approves or rejects
3. User is approved and is allowed to access the network resources.	3. Provider keeps track of usage

Let us examine to see how well this scheme matches the criteria we have laid out. The scheme outlined above seems to be practical and scalable from the provider's point of view. In this case the network and service is provided by the same entity. The network provider can expand the network and automatically allow all its existing users to access the expanded network. From the users point of view, they can use the network where their provider has a deployed infrastructure. The users have the responsibility to authenticate themselves every time they enter a physical location. Entering user id and password information using a resource-constrained device becomes an issue, however a different implementation of this scheme could over come this limitation.

The user id and the password are passed in the clear or can be encrypted. If they are passed in the clear it is very easy for sniffers to steal the credentials. Encryption scheme would require additional software to be installed on the client device and all authentication requests have to be handled by this software. A more serious issue with this scheme is if another rogue provider pretends to be the provider known to the user, they can easily steal the users credentials and any other information being accessed by the user. This scheme does not allow the user to verify the authenticity of the provider.

Privacy in this scheme is limited to what is agreed upon between the user and the provider at the time of signup. The provider has the ability to keep information about all of the services being accessed on a per-user, per physical location.

The virtual public WLAN uses a similar scheme as above. The main difference is that network is made up of several different infrastructure providers and a coalesced together and made to appear as a single network by the service provider. The user signs up with the service provider and is allowed to access any of the public WLANs deployed by the service provider partners. The major difference in this scheme from the user perspective is the availability of the expanded network.

Table 2. shows the three entities involved in a public network and the messages that are exchanged between them.

Mobile Client	Infrastructure	Service provider
1.User connects to infrastructure	1. Detects user, notifies service provider. Returns the form sent by provider to user.	1. Provider sends a authentication form
2. User enters id and password	2. Does minimal validation and forwards it to service provider. Saves the authorization token and accepts the user.	2. Provider verifies the presented credentials against its user database and approves by sending an authorization token.
3. User is approved is able to use the service.	3. Keeps track of users usage. Sends usage info to service provider.	3. Receives user specific usage data.

In terms of protecting the user credentials and the privacy this scheme suffers from similar limitations as identified above. Even though the user has a better coverage, the user is still required to authenticate at each new physical location.

The service provider has to deal with wide range of infrastructure owners and enter into SLAs with each of them. The service provider cannot guarantee the performance or the reliability of the network as each infrastructure owner may have distinct standards.

The infrastructure provider by signing up with a service provider may be forced to restrict the use of the infrastructure to just the clients of the service provider. They may be limited or restricted from offering local services and charging for it as they have no way to authenticate or bill the client directly.

3 Our protocol

We use the following authentication and authorization protocol for service access at hotspot servers. In our system, a nomadic user/client gets network services by subscribing to an Internet Service Provider (ISP). There can be many ISPs each running its own authentication/authorization server (AS) for user authentication and authorization. There also exist wireless-based hotspot servers (HS), which provide connectivity as well as local services to the nomadic user. Hotspot servers could be deployed either by ISPs or by third parties.

Our protocol has two phases, the client authentication phase and the access setup phase. In the client authentication phase, the client and his/her ISP (AS server)

mutually authenticate each other and derive a shared session key. In the access setup phase, the AS server sends client authorization information to the HS server. Additionally, the AS server chooses a service access key and sends it to the HS server and the client in a secure way. The client can then use the service access key to access HS server securely. Before describing the protocol, we first outline some of the assumptions we make about the environment.

3.1 Assumptions

Here are some of the assumptions we make while describing the protocol.

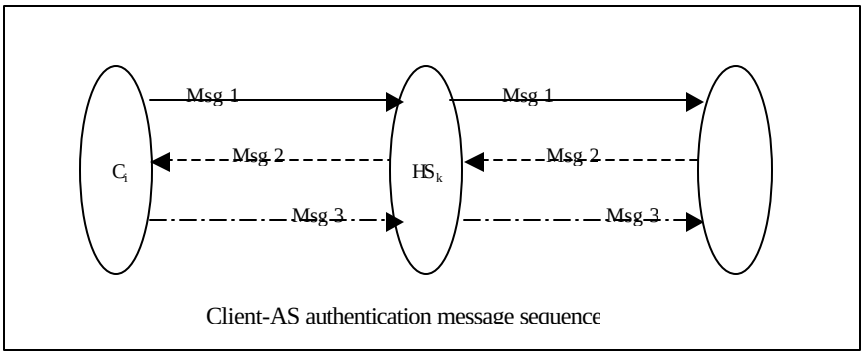
a. There are many Hotspot (HS) servers. $HS_{[1...l]}$
b. There are many ISPs that run Authentication/Authorization Servers (AS) – $AS_{[1...m]}$
c. There are several nomadic users – referred as Client (C) – $C_{[1...n]}$
d. The number of clients are much greater than the hotspot servers, which in turn are more than authorization servers.
e. Any given nomadic client C_i has a Service Level Agreement (SLA) with one or more ISPs (ASs).
f. A nomadic client C_i will request service access through HS_k (where k is based on C_i's location.
g. In-order for hotspot providers to be viable they will have SLAs with ASs. When C_i requests a service from HS_k, HS_k needs to ensure that it [HS_k] has an SLA with C_i's AS_j.
h. Anonymity of C_i should be maintained wherever possible. Thus HS_k does not need to know C_i's identity. C_i's identity is verified only by AS_j.
i. There is a Discovery protocol/mechanism that enables the C_i to establish a communication channel to HS_k.
j. The User & the personal client device are viewed as one (C_i). Ci gets access to HS_k using wireless local area network interfaces like 802.11, C_i is uniquely referred by an identifier (like the Wireless LAN card's MAC address).
k. The Client Device can range from a Laptop to a single functionality device with minimal UI.
l. We assume that each pair of AS and HS servers have a secure communication channel between them. These secure communication channels can be set up using a PKI infrastructure or other existing methods. Since AS and HS servers are expected to online, we think this is a reasonable assumption.

3.2 Client authentication phase

Before a mobile client can access any service offered by a HS, they must be authenticated. The pre-requisite for client authentication is that the client has established a relationship with an AS and the HS also has an SLA in place. We assume that the client shares a secret authentication key with each AS that he/she has an SLA with.

There are three messages exchanged between an authentication server AS and client C_i in this phase. All three messages are sent via a hotspot server HS. The role of HS is simply forwarding messages between the client and the appropriate AS.

Fig. 1. Shows the three messages that are exchanged between the client (C_i) and authorization service (AS_j) to establish mutual authentication. Hotspot (HS_k) simply forwards the messages.



- **Msg 1 (C_i -> AS_j): C_i_id, AS_j_id, $Nonce_i$, HS_k_id, Request specific data**
  where $Nonce_i$ is a number randomly chosen by C_i and sent to AS_j as a challenge. C_i_id is how the client is identified by AS_j. AS_j_id and HS_k_id are identifiers of the authentication server and the hotspot server, respectively.

Upon receiving Msg 1, AS_j verifies that the request is from a valid client (by looking up its client database). If the verification fails, AS_j aborts the protocol.

- **Msg 2 (AS_j -> C_i): C_i_id, AS_j_id, $Nonce_i$, $Nonce_j$, Response specific data, MAC_j**
  where $Nonce_j$ is a number randomly chosen by AS_j and sent to C_i as a challenge, and MAC_j is the message authentication code computed on the whole massage using the authentication key K_{ij} shared between C_i and AS_j. MAC_j serves as AS_j's response to C_i's challenge in Msg 1.

Upon receiving Msg 2, C_i verifies that MAC_j is correctly computed using key K_{ij}. If the verification fails, C_i aborts the protocol.

- **Msg 3 (C_i -> AS_j): C_i_id, AS_j_id, $Nonce_i$, $Nonce_j$, MAC_j, Response specific data, MAC_i**
  where MAC_i is the message authentication code computed on the whole massage using the authentication key K_{ij} shared between C_i and AS_j. MAC_i serves as C_i's response to AS_j's challenge in Msg 2. MAC_j is included in this message as a way of integrating the previously exchanged data.

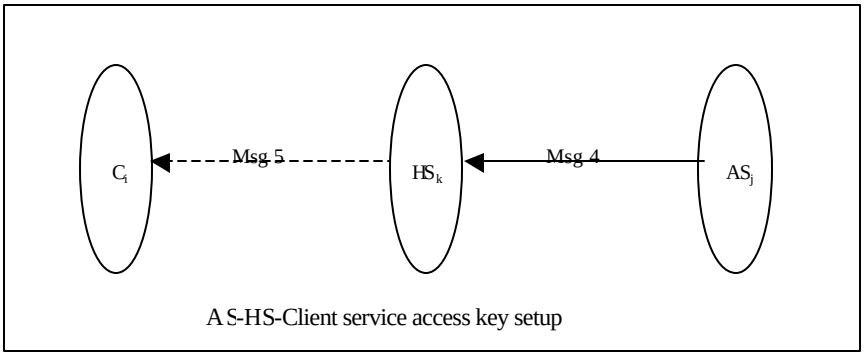
Upon receiving Msg 3, AS_j verifies that MAC_i is correctly computed using key K_{ij}. If the verification fails, AS_j aborts the protocol.

When all the verifications are successful, client C_i and authentication service AS_j have mutually authenticated each other. They can now privately compute the shared session key K_s from the message authentication code computed on (MAC_i, MAC_j) using the shared authentication key K_{ij}, i.e., $K_s = MAC(MAC_i, MAC_j)$.

3.3 Access setup phase

After successfully authenticating client C_i, AS_j obtains authorization information for C_i and send it to HS_k using the secure channel between AS_j and HS_k. Additionally, AS_j choose a service access key K_a and send it to both C_i and HS_k so that C_i can use it for secure service access with HS_k. The following two messages are exchanged in this phase.

Fig. 2. Shows the messages exchanged by the client, authorization service and hotspot in the access setup phase.



- **Msg 4 (AS_j -> HS_k): Authorization Data, K_a, $E_{Ks}[K_a]$, AS_j_id, MAC_s**
  where $E_{Ks}[K_a]$ denotes the encryption of K_a using K_s and MAC_s is the message authentication code computed on data ($E_{Ks}[K_a]$, AS_j_id) using K_s. This message is sent over the secure channel between AS_j and HS_k.

After receiving Msg 4, HS_k gets the necessary authorization information and the service access key K_a. HS_k then forwards ($E_{Ks}[K_a]$, AS_j_id, MAC_s) to the client C_i.

- **Msg 5 (HS_k -> C_i): $E_{Ks}[K_a]$, AS_j_id, MAC_s**

After receiving Msg 5, client C_i verifies that MAC_s is correctly computed using key K_s associated with the received AS_j_id. When successful, client C_i decrypts $E_{Ks}[K_a]$ using K_s to obtain K_a. Client C_i can then use the shared service access key K_a to

access HS_k securely.

3.4 Discussion

We have described a protocol that allows clients to authenticate themselves with authorization services in public un-trusted environments. The protocol does not require the client to send its secrets over the network and also allows it to verify the credentials of the authorization service. Several of the popular protocols either require the client to send its secrets over the network or provide no way to authenticate the credentials of the service provider. In public un-trusted networks it is critical that client not be required to send its keys over the network. For example in user-name, password based protocol, it is trivial for a rogue hotspot to save the credentials of users and later use them.

Kerberos[11] is an application-level security and authentication system that was developed as part of MIT's Project Athena is another protocol that is quite popular, however it requires that all clients and services be part of a single kerberos authentication database.

4. Implementation

We will briefly discuss the various alternatives we considered and give an overview of our implementation

4.1 General discussion

The hotspot AA protocol's intent is to establish a secure environment where there is mutual distrust between the three parties involved, i.e., the Client, the hotSpot server and the Authentication Server. NASREQ & Mobile IP extensions to Diameter[1][1] have similar goals, i.e. secure authenticated remote access, but address them in different domains. Therefore some of their features do not satisfy the requirements defined for our domain as defined in the introduction section.

Currently, Internet access is the only application these extensions support. The hotspot AA protocol also supports AA for application services that may be potentially distributed.

The ***NASREQ extension*** is tuned to provide IP access, logging-in service etc. It does not support a 3-party authentication scheme. The ***Mobile IP extension*** is very similar to hotspot AA protocol conceptually. This is because the three entities - mobile node

[1] The Diameter protocol, an IETF Internet Draft based on RADIUS, is defined to provide a base protocol that can be extended in order to provide AAA services to new access technologies. The base protocol is not stand-alone & is designed to be extended with a Diameter application. Two Diameter applications, currently defined in the IETF drafts, of interest are: the Mobile IP & NASREQ extensions.

(MN), foreign agents and the home agents share a security association. New session keys are generated for each combination MN-Home, MN-Foreign, Foreign-Home whenever a user tries to access a home agent. The hotspot-AA protocol deals with mutual authentication between User (MN) and AS (Home agent) while in the mobile IP case the MN authenticates with the Home Agent; Home Agent authentication (by the user) is not required.

4.2 Hotspot-AA protocol implementation

Section has 2 sub-sections: the first part discusses the implementation details of the AA protocol & sub-section 2 is about the encryption options between client & HS.

4.2.1 AA Protocol

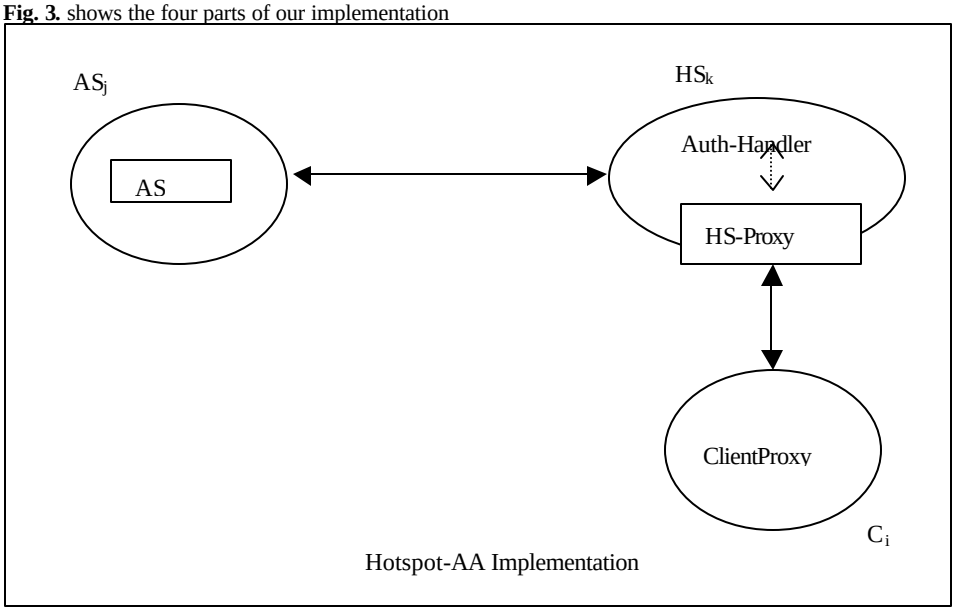
Our implementation currently supports authenticated access for HTTP-based access & services. The protocol has been implemented over both Diameter (as an Diameter application) and HTTP [13]. For the Diameter based implementation, the publicly available implementation[2] of the Diameter base protocol from Sun [8] has been used (which also implements the Diameter protocol API specification [9]). The hotspot-AA protocol messages are transported in the format as defined in the EAP specification (in the NASREQ extension). The protocol was implemented over HTTP to also cater to entities not supporting the Diameter release from Sun. It also enabled access to systems across firewalls (by exploiting the presence of http proxies).

The implementation of the hotspot-AA protocol is a proxy-based solution and hence the execution of the protocol is transparent to the user being authenticated. The implementation was done keeping Internet Access AA in mind mostly because one of the primary functionality of the hotspot server is providing access to the Internet at the hotspots.

We also provide implicit authorization for local web services deployed at the hotspots. For example, certain services should not be accessible to a certain class of users whereas certain services should be accessible to only administrators. These kinds of Access Control can be specified at the HS.

[2] Only binaries are made available.

Fig. 3. shows the four parts of our implementation



4.2.1.1 Client side protocol handler (ClientProxy)

The client side protocol handler is implemented as a proxy (referred to as ClientProxy). The user is expected to set his browser's HTTP/HTTPS proxy to the ClientProxy. ClientProxy modifies each request in a way such that the entities at the HS can verify that the client is authentic. Specifically, the ClientProxy signs the request URLs with the authorization key and the signature is appended to the URL request before sending it to the HS. Apart from the signature, the Client-ID is also appended.

The first URL request triggers the hotspot-AA protocol execution after the ClientProxy discovers that it does not possess the authorization key for accessing the entities at the HS. The ClientProxy sends and receives the hotspot-AA protocol messages as HTTP POST messages and responses respectively. The protocol message exchanges happens with the Auth Handler (described in the next section) at the HS.

4.2.1.2 HS side protocol handler (Auth-Handler & HS-Proxy)

The HS side has two components:

- Auth-Handler

The implementation of the HS side protocol handler (referred to as *Auth Handler*) is implemented as a web service. There are two variations of this:

– Diameter based Auth Handler

This implements a subset of the NAS side of the Diameter NASREQ extension. Simply put, the Auth Handler converts the POST HTTP messages from ClientProxy to corresponding Diameter messages. These messages are then sent to the AS and the responses are converted to HTTP POST response and sent to the client.

– HTTP based Auth Handler

In this case, the hotspot-AA HTTP POST messages from the ClientProxy are forwarded to the intended AS over HTTP.

Access to the Auth Handler for hotspot-AAA protocol execution itself does not require user authentication. In both the above cases, the HS and the AS gets mutually authenticated (and a session key is generated) before the client messages are forwarded. The HS and the AS also undergoes the shared key based authentication. The shared key is established as a result of the SLA. (There are other alternates possible for the HS $\leftrightarrow$ AS authentication, using PKI, for example.) The mutual authentication between any two principals is limited to a definite time interval after which they have to undergo the authentication again. The Authorization key that is distributed by the AS (Message #4 in the protocol description) is stored.

- HS-Proxy

For providing Internet access to clients the HS hosts a proxy. In order to provide access to only authenticated users, the proxy needs to have some hooks to query the Auth Handler, etc. In other words, the proxy has to invoke the AA infrastructure for each client request.

Every client URL request contains a signature and the client-id (refer the section on Client side protocol handler). HS-Proxy invokes the Auth-Handler service with the arguments - URL, signature and client-id. The Auth-Handler validates the signature and responds back with the authorization status.

The Authorization key given by the Auth-Handler is used by the HS-Proxy to do encryption/decryption.

4.3 AS side protocol handler

The AS has also been implemented over both Diameter and HTTP. The Diameter based implementation is a standalone Diameter server (implementing the Diameter base protocol) which dynamically loads the AS side of the hotspot-AA protocol (implemented as a library). The library registers a set of callbacks to process the

Diameter messages dealing with hotspot-AAA. On receipt of a message, the Diameter server parses the message and invokes the registered callbacks.
The HTTP based implementation is a web-service. The AS maintains two databases – one for the clients and one for the HSs that it has SLAs with.

4,4 Encryption options

Data encryption is optional and is left to the client to decide whether he wants encrypted data transfer or not. Encryption is provided only between the ClientProxy and the HS-Proxy. The HS-Proxy decrypts the data before forwarding it to the next network hop. Encryption/decryption algorithms use the Authorization Key (that the AS hands over to the dient and the HS) as the key. The ClientProxy and the HS-Proxy does the encryption on only a part of the data (bodies of the HTTP messages).
Note that this encryption is over and above the encryption due to data transfer with a secure web-site (SSL communication).

5. Summary

We have outlined several AA schemes that are in practice today and shown that none of these adequately address the three party mutually un-trusted network model. We propose a protocol that can be used to authenticate and authorize clients, infrastructure and service providers that operate in an un-trusted environment. This protocol can play a significant role in the development of public data networks that support true roaming. The current implementation supports only HTTP, we would like to extend the implementation to support other transport mechanisms like FTP and RTSP. We also would like to investigate how this protocol can co-exist with mobile-ip.

6. References

[1] Diameter Base Protocol
http://www.interlinknetworks.com/references/technical_materials/docs/draft-ietf-aaa-diameter-mobileip-07.txt
[2] Diameter Mobile IPv4 Application
http://www.interlinknetworks.com/references/technical_materials/docs/draft-ietf-aaa-diameter-mobileip-07.txt
[3] Diameter NASREQ Application
http://www.interlinknetworks.com/references/technical_materials/docs/draft-ietf-aaa-diameter-nasreq-07.txt

[4] PPP Extensible Authentication Protocol (EAP)
http://www.faqs.org/rfcs/rfc2284.html
[5] Diameter implementation
http://playground.sun.com/diameter
[6] The Diameter API
http://www.interlinknetworks.com/references/technical_materials/docs/draft-ietf-aaa-diameter-api-01.txt
[7] cbserver- A fast and tiny Coolbase Appliance Server
http://www.cooltown.com/dev/reference/coolbase/cbserver/cbserverPaper.pdf
[8] OpenSSL
http://www.openssl.org/
[9] Hypertext Transfer Protocol
http://www.w3.org/Protocols/rfc2616/rfc2616.html
[10] tinyproxy
http://tinyproxy.sourceforge.net/
[11] B. Clifford Neuman and Theodore Ts'o. Kerberos: An Authentication Service for Computer Networks, *IEEE Communications*, 32(9):33-38. September 1994