Network Working Group   J. Vollbrecht
Request for Comments: 2904   Interlink Networks, Inc.
Category: Informational   P. Calhoun
                                              Sun Microsystems, Inc.
                                                        S. Farrell
                                              Baltimore Technologies
                                                        L. Gommans
                                             Enterasys Networks EMEA
                                                          G. Gross
                                                 Lucent Technologies
                                                       B. de Bruijn
                                              Interpay Nederland B.V.
                                                        C. de Laat
                                                  Utrecht University
                                                       M. Holdrege
                                                            ipVerse
                                                         D. Spence
                                           Interlink Networks, Inc.
                                                        August 2000

                      AAA Authorization Framework

Status of this Memo

Copyright Notice

Abstract

   This memo serves as the base requirements for Authorization of
   Internet Resources and Services (AIRS). It presents an architectural
   framework for understanding the authorization of Internet resources
   and services and derives requirements for authorization protocols.

Table of Contents

1. Introduction

    This document is one of a series of three documents under
    consideration by the AAAarch RG dealing with the authorization
    requirements for AAA protocols. The three documents are:

         AAA Authorization Framework (this document)
         AAA Authorization Requirements [2]
         AAA Authorization Application Examples [3]

    There is a demonstrated need for a common scheme which covers all
    Internet services which offer Authorization. This common scheme will
    address various functional architectures which meet the requirements
    of basic services. We attempt to describe these architectures and
    functions as a basis for deriving requirements for an authorization
    protocol [2].

Vollbrecht, et al.   Informational   [Page 2]

These architectures include Policy structures, Certificate
Authorities, Resource Managers, Inter-Domain and Multi-Domain
schemes, and Distributed Services. The requirements are for the
expected use of Authorization services across these architectures.

A representative set of applications that may use this architecture
to support their authorization needs is presented in [3]. The
examples in [3] show how this framework may be used to meet a wide
variety of different authorization needs.

We expect that this work may be extended in the future to a more
comprehensive model and that the scheme described here will be
incorporated into a framework that includes authentication,
accounting and auditing. We have referenced a number of
authorization sources, but also recognize that there may be some that
we have missed and that should be included. Please notify one of the
authors of any such oversight so it can be corrected in a future
revision.

In general, it is assumed that the parties who are participating in
the authorization process have already gone through an authentication
phase. The authentication method used by those parties is outside
the scope of this document except to the extent that it influences
the requirements found in a subsequent authorization process.
Likewise, accounting requirements are outside the scope of this
document other than recording accounting data or establishing trust
relationships during an authorization that will facilitate a
subsequent accounting phase.

The work for this memo was done by a group that originally was the
Authorization subgroup of the AAA Working Group of the IETF. When
the charter of the AAA working group was changed to focus on MobileIP
and NAS requirements, the AAAarch Research Group was chartered within
the IRTF to continue and expand the architectural work started by the
Authorization subgroup. This memo is one of four which were created
by the subgroup. This memo is a starting point for further work
within the AAAarch Research Group. It is still a work in progress
and is published so that the work will be available for the AAAarch
subgroup and others working in this area, not as a definitive
description of architecture or requirements.

This document uses the terms 'MUST', 'SHOULD' and 'MAY', and their
negatives, in the way described in RFC 2119 [4].

2. Authorization Entities and Trust Relationships

   The following framework is being presented in order to provide a
   framework for discussing authorization requirements for a large
   number of applications. The intent is to provide some common
   vocabulary for the discussion. Terminology is introduced for basic
   elements in the authorization transaction and for concepts that
   appear to be common to all (or at least many) authorization
   proposals.

   Figure 1, below, identifies the basic conceptual entities that may
   be participants in an authorization:

   1. A User who wants access to a service or resource.

   2. A User Home Organization (UHO) that has an agreement with the user
      and checks whether the user is allowed to obtain the requested
      service or resource. This entity may carry information required
      to authorize the User, which might not be known to the Service
      Provider (such as a credit limit).

   3. A Service Provider's AAA Server which authorizes a service based
      on an agreement with the User Home Organization without specific
      knowledge about the individual User. This agreement may contain
      elements that are not relevant to an individual user (e.g., the
      total agreed bandwidth between the User Home Organization and the
      Service Provider).

   4. A Service Provider's Service Equipment which provides the service
      itself. This might, for example, be a NAS in dial service, or a
      Router in the QoS service, or a print server in the Internet
      Printing service.

```
        +------+      +------------------------+
        |      |      | User Home Organization |
        |      |      |  +------------------+  |
        |      |      |  |    AAA Server    |  |
        |      |      |  |                  |  |
        |      |      |  +------------------+  |
        |      |      |                        |
        |      |      +------------------------+
        |      |
        |      |
        |      |
        |      |
        | User |      +------------------------+
        |      |      | Service Provider       |
        |      |      |  +------------------+  |
        |      |      |  |    AAA Server    |  |
        |      |      |  |                  |  |
        |      |      |  +------------------+  |
        |      |      |                        |
        |      |      |  +------------------+  |
        |      |      |  |     Service      |  |
        |      |      |  |    Equipment     |  |
        |      |      |  +------------------+  |
        |      |      |                        |
        +------+      +------------------------+
```
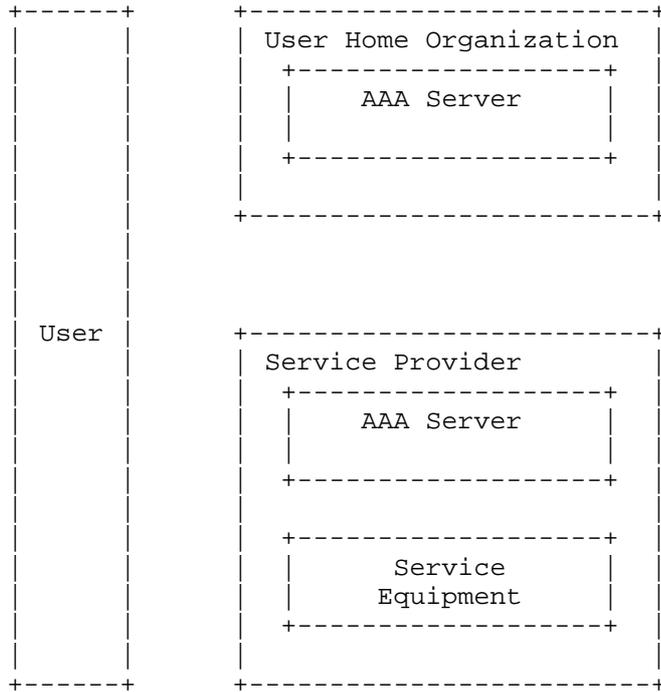
                Fig. 1 -- The Basic Authorization Entities

   These entities will be referenced in the authorization requirements.

   There may be bilateral agreements between pairs of organizations
   involved in an authorization transaction. Agreements between
   organizations may take the form of formal contracts or Service Level
   Agreements. Figure 2 uses double lines to show relationships that
   may exist between the User and the User Home Organization and between
   the User Home Organization and the Service Provider.

```
         +------+      +------------------------+
         |      |      |  User Home Organization  |
         |      |======|  +------------------+  |
         |      |      |  |    AAA Server     |  |
         |      |      |  |                   |  |
         |      |      |  +------------------+  |
         |      |      |                        |
         |      |      +------------------------+
         |      |                  ||
         |      |                  ||
         |      |                  ||
         | User |      +------------------------+
         |      |      | Service Provider       |
         |      |      |  +------------------+  |
         |      |      |  |    AAA Server     |  |
         |      |      |  |                   |  |
         |      |      |  +------------------+  |
         |      |      |                        |
         |      |      |  +------------------+  |
         |      |      |  |     Service       |  |
         |      |      |  |    Equipment      |  |
         |      |      |  +------------------+  |
         |      |      |                        |
         +------+      +------------------------+
```
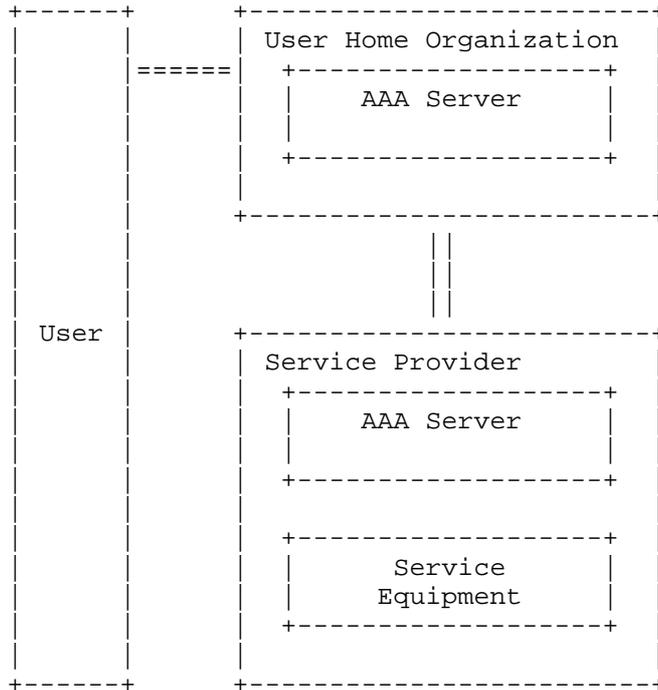
                    Fig. 2 -- Service Agreements

   Authorization is based on these bilateral agreements between
   entities. Agreements may be chained, as shown in figure 2. The User
   has an agreement with the User Home Organization (e.g., the User may
   have access to the service between 9:00 a.m. and 11:00 a.m. daily).
   The User Home Organization has an agreement with the Service Provider
   (e.g., that all requests for access will be granted, except between
   5:00 a.m. and 10:00 a.m. on Sunday). The fulfillment of the User's
   request depends on both agreements being honored.

   Note that these agreements may be implemented by hand configuration
   or by evaluation of Policy data stored in a Policy database. The
   point is that there must be a set of known rules in place between
   entities in order for authorization transactions to be executed.

   Trust is necessary to allow each entity to "know" that the policy it
   is authorizing is correct. This is a business issue as well as a
   protocol issue. Trust is often established through third party
   authentication servers (such as Kerberos), via a certificate
   authority, by configuring shared secrets or passwords, or by sharing
   a common facility (such as a connecting wire between processors).
   These "static" trust relationships are necessary for authorization

transactions to take place. Static trust relationships are used in
an authorization sequence to establish a "dynamic" relationship
between the User and the Service Equipment. Several possible
authorization sequences are possible, each of which use the static
trust "chain" to have the user first be approved by the User Home
Organization, and then have the Service Provider accept the request
based on its trust of the User Home Organization.

3. Message Sequences

   In general, the User Home Organization and the Service Provider are
   different entities or different "administrative domains". In the
   simplest case, however, the User Home Organization and the Service
   Provider may be combined as a single entity. This case will be used
   to describe three authorization sequences possible with the simple
   case.

   In following sections these concepts will be applied to more
   complicated cases involving separate User Home Organization and
   Service Provider entities (as in roaming) and multiple Service
   Providers each with their own AAA Servers and Service Equipment (as
   in distributed services).

3.1. Single Domain Case

   This case includes the User, the Service Provider's AAA Server, and
   the Service Provider's Service Equipment. Examples of this case
   include a NAS supported by a standalone RADIUS server, or a QoS
   Router supported by a local bandwidth broker.

   The sequences considered in the following figures are the "agent",
   "pull", and "push" sequences for the single domain case.

3.1.1. The Agent Sequence

   In the agent sequence (see figure 3), the Service Provider AAA Server
   functions as an agent between the User and the service itself. The
   AAA Server receives a request from the User and forwards
   authorization and possibly configuration information to the Service
   Equipment. In this model, the User sends a request to the Service
   Provider's AAA Server (1), which will apply a policy associated with
   the User and the particular service being requested. The AAA Server
   sends a request to the Service Equipment, and the Service Equipment
   sets up whatever is requested (2). The Service Equipment then
   responds to the AAA Server acknowledging that it has set up the
   Service for the user (3). The AAA Server replies to the User telling
   it that the Service is set up (4).

Depending on the nature of the service, further communication may
take place between the User and the Service Equipment. For this to
occur, there needs to be a binding between the User and the
authorized service. This requires further study.

```
                        +------------------------+
       +------+         | Service Provider       |
       |      |    1    |  +------------------+   |
       |      |-------+->|    AAA Server      |   |
       |      |<-----+--|                    |   |
       |      |    4    |  +------------------+   |
       | User |         |      | /|\              |
       |      |         |      |2 |3              |
       |      |         |      \|/  |             |
       |      |         |  +------------------+   |
       |      |         |  |    Service       |   |
       |      |         |  |    Equipment     |   |
       |      |         |  +------------------+   |
       +------+         |                        |
                        +------------------------+
```
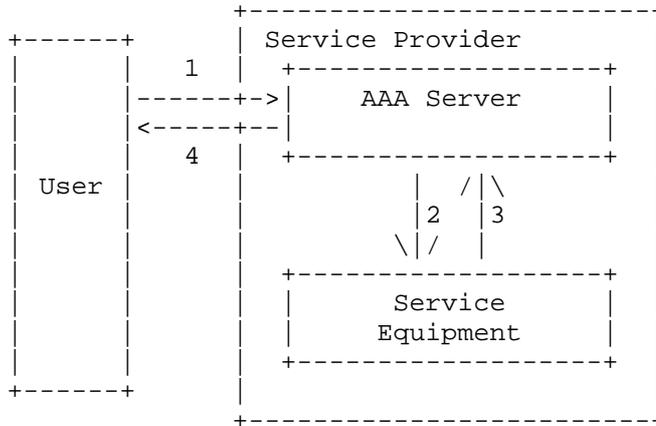
                   Fig. 3 -- Agent Sequence

   Example: A regular user may ask for 1 Mb/s bandwidth (1). The
   bandwidth broker (AAA Server) tells the router (Service Equipment) to
   set this user into the 1Mb/s "queue" (2). The router responds that
   it has done so (3), and the bandwidth broker tells the User the
   bandwidth is set up (4).

3.1.2. The Pull Sequence

   The pull sequence (figure 4) is what is typically used in the Dialin
   application, in the Mobile-IP proposal, and in some QoS proposals.
   The User sends a request to the Service Equipment (1), which forwards
   it to the Service Provider's AAA Server (2), which evaluates the
   request and returns an appropriate response to the Service Equipment
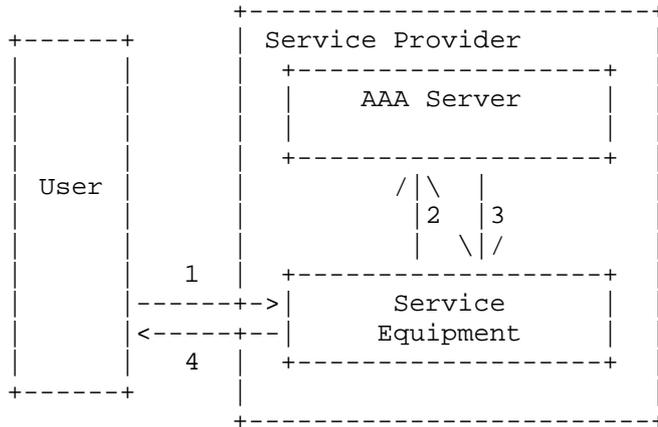   (3), which sets up the service and tells the User it is ready (4).

```
                        +-----------------------+
         +------+       | Service Provider      |
         |      |       |  +------------------+  |
         |      |       |  |   AAA Server     |  |
         |      |       |  |                  |  |
         |      |       |  +------------------+  |
         | User |       |     /|\    |         |
         |      |       |      |2    |3         |
         |      |       |      |   \ |/         |
         |      |   1   |  +------------------+  |
         |      |------+->|   Service         |  |
         |      |<-----+--|   Equipment       |  |
         |      |   4   |  +------------------+  |
         +------+       |                       |
                        +-----------------------+
```

                  Fig. 4 -- Pull Sequence

3.1.3. The Push Sequence

   The push sequence (figure 5) requires that the User get from the
   Service Provider's AAA Server a ticket or certificate verifying that
   it is o.k. for the User to have access to the service (1,2).   The
   User includes the ticket in the request (3) to the Service Equipment.
   The Service Equipment uses the ticket to verify that the request is
   approved by the Service Provider's AAA Server. The Service Equipment
   then sends an o.k. to the User (4).

   The ticket the user gets from the Service Provider's AAA Server will
   typically have some time limit on it. It may contain an indication
   of service location, and in some applications, it might be used for
   more than one request.

   In the push sequence the communication between the AAA Server and the
   Service Equipment is relayed through the User rather than directly
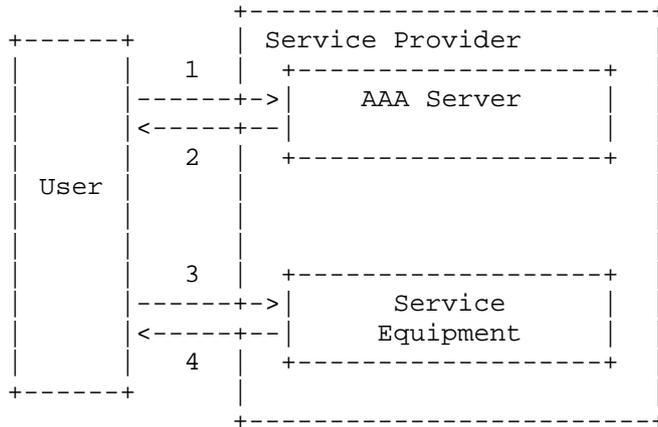   between themselves.

```
                   +------------------------+
     +------+      | Service Provider       |
     |      |  1   | +------------------+   |
     |      |------+->|    AAA Server    |   |
     |      |<-----+--|                  |   |
     |      |  2   | +------------------+   |
     | User |      |                        |
     |      |      |                        |
     |      |      |                        |
     |      |  3   | +------------------+   |
     |      |------+->|     Service      |   |
     |      |<-----+--|    Equipment     |   |
     |      |  4   | +------------------+   |
     +------+      |                        |
                   |                        |
                   +------------------------+
```

                    Fig. 5 -- Push Sequence

3.2. Roaming -- the User Home Organization is not the Service Provider

   In many interesting situations, the organization that authorizes and
   authenticates the User is different from the organization providing
   the service. This situation has been explored in the Roaming
   Operations (roamops) Working Group. For purposes of this discussion,
   any situation in which the User Home Organization is different from
   the Service Provider is considered to be roaming.

   Examples of roaming include an ISP selling dialin ports to other
   organizations or a Mobile-IP provider allowing access to a user from
   another domain.

   The same agent, pull and push sequences are possible with roaming.
   If the Service Provider's AAA Server and the Service Equipment are
   grouped as a logical entity for purposes of description, then the
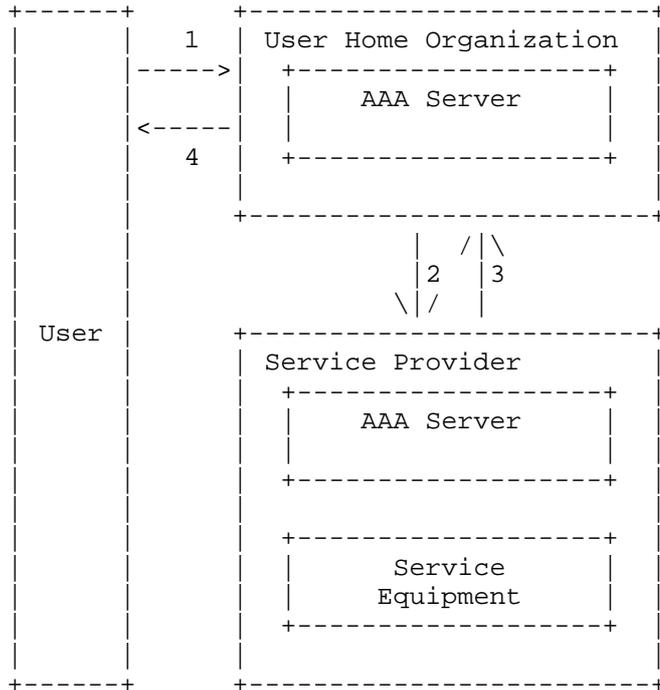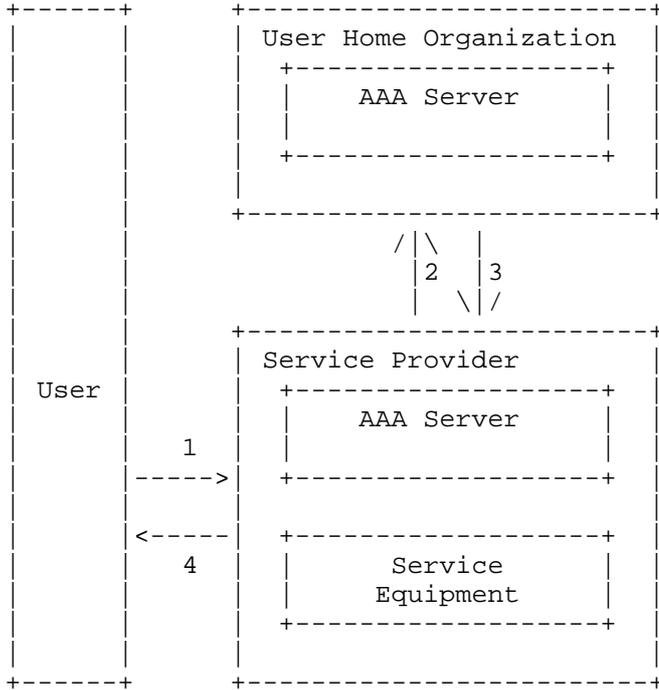   following figures illustrate these cases.

```
        +------+   +------------------------+
        |      |  1 | User Home Organization |
        |      |----->|  +------------------+  |
        |      |      |  |    AAA Server    |  |
        |      |<-----|  |                  |  |
        |      |   4  |  +------------------+  |
        |      |      |                        |
        |      |      +------------------------+
        |      |             |   /|\
        |      |             |2  |3
        |      |             \|/  |
        | User |      +------------------------+
        |      |      |  Service Provider      |
        |      |      |  +------------------+  |
        |      |      |  |    AAA Server    |  |
        |      |      |  |                  |  |
        |      |      |  +------------------+  |
        |      |      |                        |
        |      |      |  +------------------+  |
        |      |      |  |     Service      |  |
        |      |      |  |    Equipment     |  |
        |      |      |  +------------------+  |
        |      |      |                        |
        +------+      +------------------------+
```

                Fig. 6 -- Roaming Agent Sequence

```
        +------+       +------------------------+
        |      |       | User Home Organization |
        |      |       |  +------------------+  |
        |      |       |  |    AAA Server    |  |
        |      |       |  |                  |  |
        |      |       |  +------------------+  |
        |      |       |                        |
        |      |       +------------------------+
        |      |            /|\   |
        |      |            |2   |3
        |      |            |  \|/
        |      |       +------------------------+
        |      |       | Service Provider       |
        | User |       |  +------------------+  |
        |      |       |  |    AAA Server    |  |
        |      |    1  |  |                  |  |
        |      |       |  +------------------+  |
        |      |------>|                        |
        |      |       |  +------------------+  |
        |      |<------|  |                  |  |
        |      |    4  |  |     Service      |  |
        |      |       |  |    Equipment     |  |
        |      |       |  +------------------+  |
        |      |       |                        |
        +------+       +------------------------+
```

           Fig. 7 -- Roaming Pull Sequence

```
        +------+    +------------------------+
        |      |  1 | User Home Organization |
        |      |----->|  +-------------------+  |
        |      |    |  |     AAA Server    |  |
        |      |<-----|  |                   |  |
        |      |  2 |  +-------------------+  |
        |      |    |                        |
        |      |    +------------------------+
        |      |
        |      |
        |      |
        | User |    +------------------------+
        |      |    |  Service Provider      |
        |      |    |  +-------------------+  |
        |      |    |  |     AAA Server    |  |
        |      |  3 |  |                   |  |
        |      |----->|  +-------------------+  |
        |      |    |                        |
        |      |<-----|  +-------------------+  |
        |      |  4 |  |     Service       |  |
        |      |    |  |     Equipment     |  |
        |      |    |  +-------------------+  |
        |      |    |                        |
        +------+    +------------------------+
```
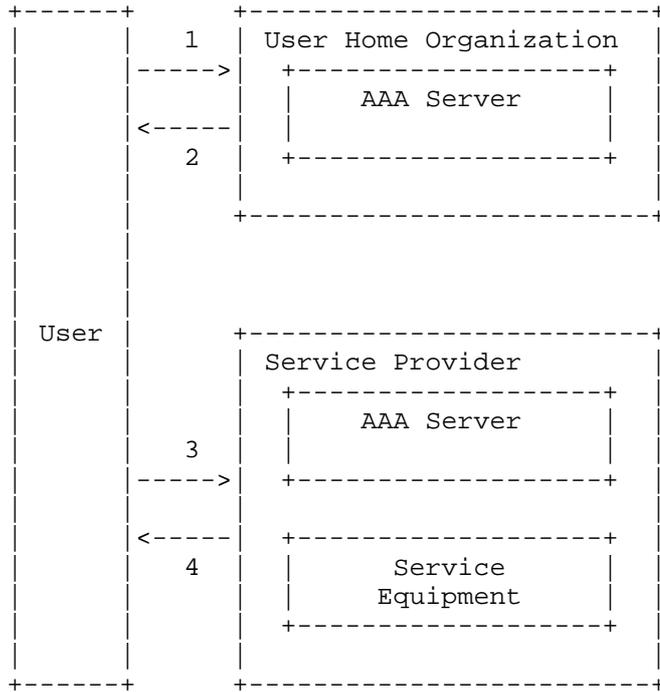
Fig. 8 -- Roaming Push Sequence

3.3. Distributed Services

   To provide a complete service to a user, offerings from several
   service providers may need to be combined. An example would be a
   user who requires a QoS service for a session that crosses multiple
   ISPs. Any service that is provided by more than one Service Provider
   acting in concert is a distributed service. Figure 9 illustrates
   distributed services.

```
                +------------------+   +------------------+
+------+        | Org1             |   | Org2             |
|      |        |  +-------------+  |   |  +-------------+  |
|      |        |  | AAA Server  |  |   |  | AAA Server  |  |
|      |        |  |             |  |   |  |             |  |
|      |        |  +-------------+  |   |  +-------------+  |
| User |=======|                   |======|                  |
|      |        |  +-------------+  |   |  +-------------+  |
|      |        |  |  Service    |  |   |  |  Service    |  |
|      |        |  | Equipment   |  |   |  | Equipment   |  |
|      |        |  +-------------+  |   |  +-------------+  |
+------+        |                   |   |                  |
                +------------------+   +------------------+
```
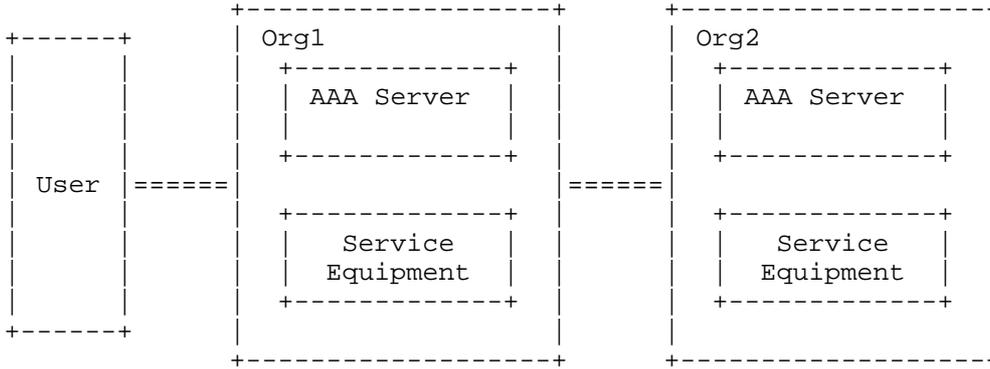
                  Fig. 9 -- Distributed Services

The agreements between entities in figure 9 imply that the request
from the User will be authenticated and authorized by the first
organization, then forwarded to the second organization. Note that
the sequence between User and Org1 may be different than between Org1
and Org2. The first might use a pull sequence, and the second might
use an agent sequence. This example is illustrated in figure 10.

```
                +------------------+   +------------------+
+------+        | Org1             |   | Org2             |
|      |        |  +-------------+  | 3 |  +-------------+  |
|      |        |  | AAA Server  |--+------+->| AAA Server  |  |
|      |        |  |             |<-+------+--|             |  |
|      |        |  +-------------+  | 6 |  +-------------+  |
| User |        |      /|\        |   |        |  /|\      |
|      |        |       |2  |7     |   |        |4  |5      |
|      |        |       |  \|/     |   |        \|/  |      |
|      |      1 |  +-------------+  |   |  +-------------+  |
|      |------+->|  Service    |  |   |  |  Service    |  |
|      |<-----+--|  Equipment  |  |   |  |  Equipment  |  |
|      |      8 |  +-------------+  |   |  +-------------+  |
+------+        |                   |   |                  |
                +------------------+   +------------------+
```
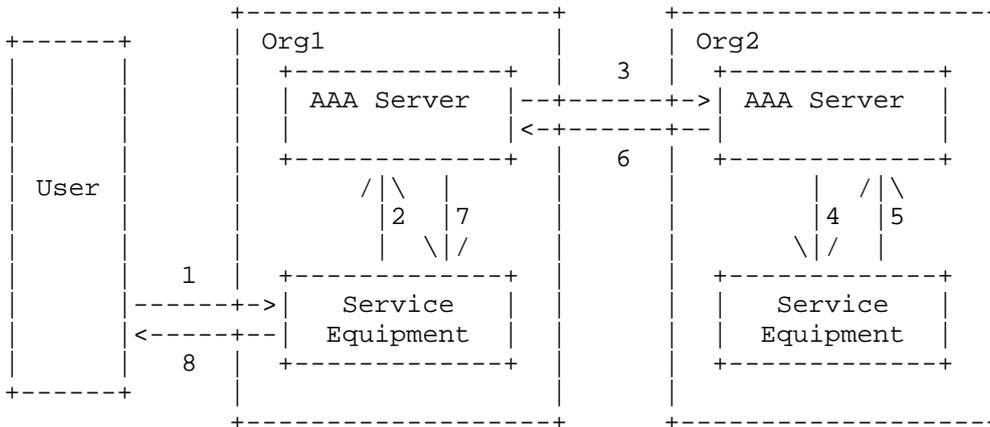
               Fig. 10 -- A Possible Distributed Sequence

There are a number of other ways that authorization sequences for
distributed services can be set up. For example, it is possible
that, in order to reduce delay time in setting up a session, Org1
could send a response to the user before receiving a verification
that Org2 has authorized the service. In that case Org1 would need
to be able to revoke the authorization sent earlier if Org2 does not
send the authorization in some amount of time.

3.4. Combining Roaming and Distributed Services

   Figure 11 shows a combination of Roaming and Distributed Services.
   Contract and trust relationships may be set up in number of ways,
   depending on a variety of factors, especially the business model.

```
+------+     +------------------+     +------------------+
|      |     | User Home Org    |     | SuperOrg         |
|      |     | +------------+   |     | +------------+   |
|      |     | | AAA Server |   |     | | AAA Server |   |
|      |     | |            |   |     | |            |   |
|      |     | |            |   |     | |            |   |
|      |     | +------------+   |     | +------------+   |
|      |     |                  |     |                  |
|      |     +------------------+     +------------------+
|      |
|      |
|      |     +------------------+     +------------------+
| User |     | Org1             |     | Org2             |
|      |     | +------------+   |     | +------------+   |
|      |     | | AAA Server |   |     | | AAA Server |   |
|      |     | |            |   |     | |            |   |
|      |     | +------------+   |     | +------------+   |
|      |     |                  |     |                  |
|      |     | +------------+   |     | +------------+   |
|      |     | |  Service   |   |     | |  Service   |   |
|      |     | | Equipment  |   |     | | Equipment  |   |
|      |     | +------------+   |     | +------------+   |
|      |     |                  |     |                  |
+------+     +------------------+     +------------------+
```
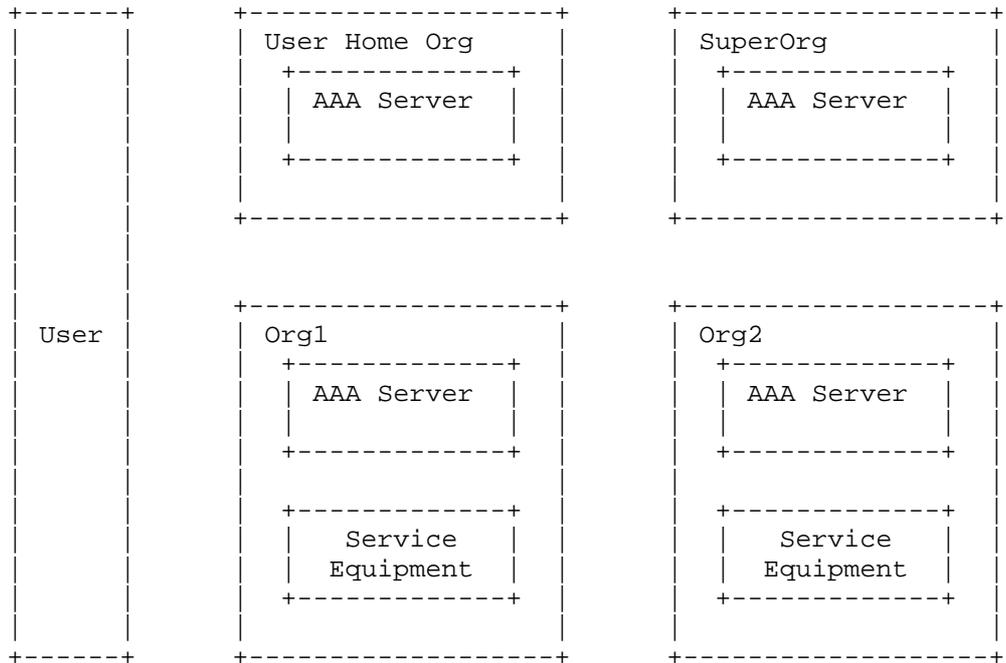
           Fig. 11 -- Roaming and Distributed Services

   New entities that combine or add capabilities can be created to meet
   business needs.   In figure 11, one such possibility, a SuperOrg
   entity is shown. The idea is that this entity would provide
   authentication and authorization for organizations that are providing
   services to end-users. It could be considered to be a wholesaler or
   broker. While not all authorization will require having a broker,
   authorization protocols should allow such entities to be created to
   meet legitimate requirements.

   Having considered the basic players and how they interact, we will
   now consider different ways that authorization data may be stored in
   the network.

4. Relationship of Authorization and Policy

   The Policy Framework (policy) Working Group is seeking to provide a
   framework to represent, manage, and share policies and policy
   information in a vendor-independent, interoperable, scalable manner.
   [5],[6],[7]. This section explores the relationship of policy and
   authorization and sets the stage for defining protocol requirements
   for supporting policy when included as part of multi-domain
   authorization. The work presented here builds on the policy
   framework, extending it to support policy across multiple domains.

   One view of an authorization is that it is the result of evaluating
   policies of each organization that has an interest in the
   authorization decision. In this document the assumption is that each
   administration may have policies which may be indexed by user, by
   service, or by other attributes of the request. The policies of each
   administration are defined independently of other administrations.

   Each independent policy must be 1) retrieved, 2) evaluated, and 3)
   enforced.

4.1. Policy Retrieval

   Policy definitions are maintained and stored in a policy repository
   [5] by (or on behalf of) the organization that requires them. The
   Policy Framework WG is working on a way to describe policy [7].
   Other implementations describe policy as a set of ACL lists. Policy
   definitions must be retrieved in order to be evaluated and enforced.
   Policy Definitions can be indexed by requester, by service attribute,
   or by some other key. The organization requiring the policy is also
   responsible for determining which policy is to be applied to a
   specific authorization request.

   Policy retrieval is typically done by the administration that defines
   the policy or by an agent acting for that administration. Thus a
   policy defining the times of day that a particular User is allowed to
   connect to the network is maintained and retrieved by the User
   Organization. A policy defining a time that ports will be unusable
   because of maintenance is maintained and retrieved by the Service
   Provider.

   Note that some implementation may choose to have the Service Provider
   retrieve a policy from the User Home Organization using a distributed
   directory access protocol. This may be appropriate in some cases,
   but is not a general solution. To understand why, suppose the remote
   administration and the home administration communicate via a broker
   which proxies their communications. In this case the remote and home

   administrations have no prior relationship, and therefore the home
   administration directory is unlikely to be open for access by the
   remote administration and vice versa.

4.2. Policy Evaluation

   Evaluation of policy requires access to information referenced by the
   policy. Often the information required is not available in the
   administration where the policy is retrieved. For example, checking
   that a user is allowed to login at the current time can readily be
   done by the User Home Organization because the User Home Organization
   has access to current time. But authorizing a user requiring a 2Mb/s
   path with less than 4 hops requires information available at a
   Service Provider and not directly available to the UHO, so the UHO
   must either 1) have a way to query a remote administration for the
   needed information or 2) forward the policy to the remote
   administration and have the remote administration do the actual
   evaluation or 3) attempt somehow to "shadow" the authoritative source
   of the information (e.g by having the Service Provider send updates
   to the UHO).

   Applications might support either 1) or 2), and a general
   authorization protocol must allow both. Case 3) is not considered
   further as shadowing seems too "expensive" to be supported by an AAA
   protocol.

   An example of case 1 is when a Service Provider forwards a request to
   a UHO which includes a query for the clearance code of the User. The
   Service Provider policy includes reference to the clearance code and
   the information in the reply is used as input to that policy.

   An example of case 2 is when the UHO approves an authorization
   conditional on the Service Provider confirming that there is
   currently a specific resource available for its use. The UHO
   includes the "policy" along with a conditional authorization to the
   Service Provider.

4.3. Policy Enforcement

   Policy Enforcement is typically done by the Service Provider on the
   Service Equipment. The Service Equipment is equivalent to the Policy
   Target described in the Policy Framework [5]. Thus a NAS may enforce
   destination IP address limits via "filters" and a Router may enforce
   QoS restrictions on incoming packets. The protocol that sends the
   information between the Service Equipment and the Service Provider
   AAA Server may be specific to the Service Equipment, but it seems
   that the requirements are not different in kind from what is required
   between other AAA servers.

In particular, an AAA Server could send a "policy" to the Service
Equipment stating what the equipment should do under various
situations. The Service equipment should either set up to "enforce"
the policy or reject the request.

The AAA Server could also send a query to the Service Equipment for
information it requires to evaluate a policy.

4.4. Distributed Policy

A policy is retrieved by a Policy Retrieval Point (PRP) from a Policy
Repository, evaluated at a Policy Decision Point (PDP) or Policy
Consumer, and enforced at a Policy Enforcement Point (PEP) or Policy
Target [5].

Generally, any of the AAA Servers involved in an authorization
transaction may retrieve a policy or evaluate a policy, and any of
the Service Equipment may enforce a policy. Policy Repositories may
reside on any of the AAA Servers or be located elsewhere in the
network.

Information against which policy conditions are evaluated (such as
resource status, session state, or time of day) are accessible at
Policy Information Points (PIPs) and might be accessed using Policy
Information Blocks (PIBs). An interesting question in any
authorization application that uses policy is where are the PDPs,
PRPs, PIPs and PEPs?

Figure 12 shows which policy elements may be available at different
points in the model. In distributed services, there may be multiple
Service Providers involved in the authorization transaction, and each
may act as the policy elements shown below.

Note that the User (or requester) may also be a PRP (e.g. use policy
description to specify what service is being requested), a PIP (have
information needed by other entities to evaluate their policy), and a
PDP (decide if it will accept a service with specific parameters).

```
     +------+     +-----------------------------+
     |      |     | User Home Organization      |
     |      |     | +-------------------+  PRP  |
     |      |     | |    AAA Server     |  PIP  |
     |      |     | |                   |  PDP  |
     |      |     | +-------------------+       |
     |      |     |                             |
     |      |     +-----------------------------+
     |      |
     |      |
     |      |     +-----------------------------+
     | User |     | Service Provider            |
     |      |     | +-------------------+  PRP  |
     |      |     | |    AAA Server     |  PIP  |
     | PRP  |     | |                   |  PDP  |
     | PIP  |     | +-------------------+       |
     | PDP  |     |                             |
     |      |     | +-------------------+       |
     |      |     | |      Service      |  PIP  |
     |      |     | |      Equipment    |  PEP  |
     |      |     | +-------------------+       |
     |      |     |                             |
     +------+     +-----------------------------+
```
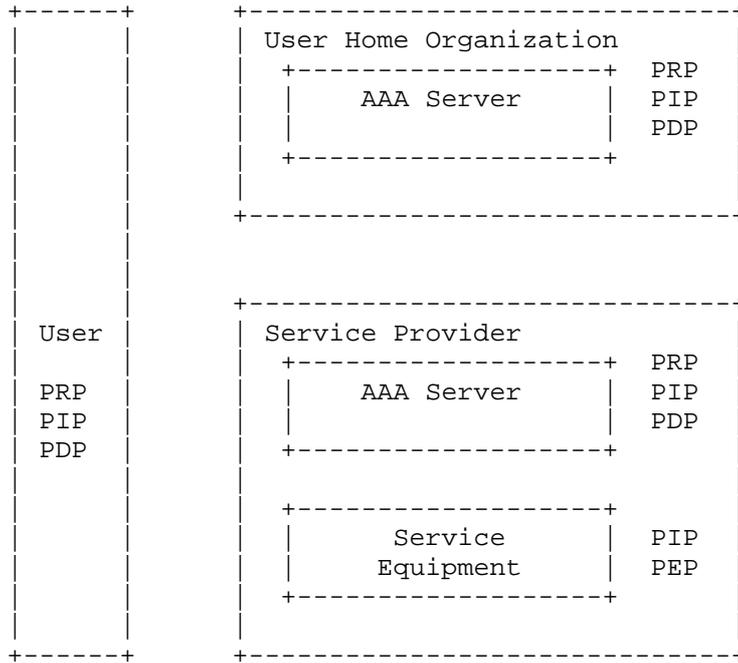
            PRP = Policy Retrieval Point
            PIP = Policy Information Point
            PDP = Policy Decision Point
            PEP = Policy Enforcement Point

        Fig. 12 -- Where Different Policy Elements May be Located

   An AAA protocol must be able to transport both policy definitions and
   the information needed to evaluate policies. It must also support
   queries for policy information.

5. Use of Attribute Certificates to Store Authorization Data

   This section outlines another mechanism that could be used for
   securely transporting the attributes on which an authorization
   decision is to be made.   Work on X.509 Attribute Certificates is
   currently being undertaken in the Public Key Infrastructure (PKIX)
   Working Group [8]. This proposal is largely based on that work.

   When considering authorization using certificate-based mechanisms,
   one is often less interested in the identity of the entity than in
   some other attributes, (e.g. roles, account limits etc.), which
   should be used to make an authorization decision.

In many such cases, it is better to separate this information from
the identity for management, security, interoperability or other
reasons. However, this authorization information may also need to be
protected in a fashion similar to a public key certificate. The name
used here for such a structure is an Attribute Certificate (AC) which
is a digitally signed (certified) set of attributes.

An AC is a structure that is similar to an X.509 public key
certificate [9] with the main difference being that it contains no
public key. The AC typically contains group membership, role,
clearance and other access control information associated with the AC
owner. A syntax for ACs is also defined in the X.509 standard.

When making an access decision based on an AC, an access decision
function (in a PEP, PDP or elsewhere) may need to ensure that the
appropriate AC owner is the entity that has requested access. The
linkage between the request and the AC can be achieved if the AC has
a "pointer" to a Public Key Certificate (PKC) for the requester and
that the PKC has been used to authenticate the request. Other forms
of linkage can be defined which work with other authentication
schemes.

As there is often confusion about the difference between public key
certificates (PKC's) and attribute certificates (ACs), an analogy may
help. A PKC can be considered to be like a passport: it identifies
the owner, it tends to be valid for a long period, it is difficult to
forge, and it has a strong authentication process to establish the
owner's identity. An AC is more like an entry visa in that it is
typically issued by a different authority than the passport issuing
authority, and it doesn't have as long a validity period as a
passport. Acquiring an entry visa typically requires presenting a
passport that authenticates that owner's identity. As a consequence,
acquiring the entry visa becomes a simpler procedure. The entry visa
will refer to the passport as a part of how that visa specifies the
terms under which the passport owner is authorized to enter a
country.

In conjunction with authentication services, ACs provide a means to
transport authorization information securely to applications.
However, there are a number of possible communication paths that an
AC may take.

In some environments, it is suitable for a client to "push" an AC to
a server. This means that no new connections between the client and
server domains are required. It also means that no search burden is
imposed on servers, which improves performance.

In other cases, it is more suitable for a client simply to
authenticate to the server and for the server to request the client's
AC from an AC issuer or a repository. A major benefit of the this
model is that it can be implemented without changes to the client and
client/server protocol. It is also more suitable for some inter-
domain cases where the client's rights should be assigned within the
server's domain, rather than within the client's "home" domain.

There are a number of possible exchanges that can occur, and there
are three entities involved: client, server, and AC issuer. In
addition the use of a directory service as a repository for AC
retrieval may be supported.

Figure 13 shows an abstract view of the exchanges that may involve
ACs. Note that the lines in the diagram represent protocols which
must be defined, not data flows. The PKIX working group will define
the required acquisition protocols. One candidate for the lookup
protocols is LDAP (once an LDAP schema exists which states where an
AC is to be found).

```
    +--------------+                        +--------------+
    | AAA Server/  |                        |              |
    | AC Issuer    +----+                   | Directory    |
    |              |    |                    |              |
    +--+-----------+    | Server            +-------+------+
       |                | Acquisition               |
       |Client          |                           |Server
       |Acquisition     +--------------------+       |Lookup
       |                                     |       |
    +--+-----------+                        +--+----+------+
    |              |    AC in application   | Service      |
    |    User      +----------------------+ Equipment/     |
    |              |       protocol        | AAA Server    |
    +--+-----------+                        +--------------+
       |
       | Client Lookup
    +--+-----------+
    |              |
    | Directory    |
    |              |
    +--------------+
```
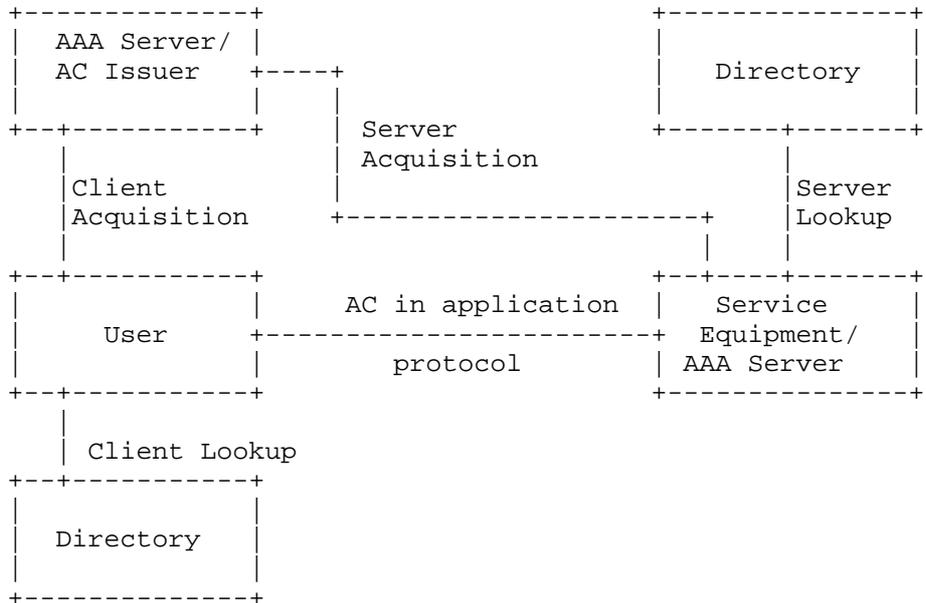
                    Fig. 13 -- AC Exchanges

Figure 14 shows the data flows which may occur in one particular
case, that termed "push" above (section 2.1.3).

```
    +--------------+
    |  AAA Server/ |
    |  AC Issuer   |
    |              |
    +--+----------+
       |
       |Client
       |Acquisition (1)
       |
    +--+----------+                      +--------------+
    |             |   AC in application  |   Service    |
    |     User    +----------------------+   Equipment/ |
    |             |      protocol (2)     |  AAA Server  |
    +-------------+                      +--------------+
```
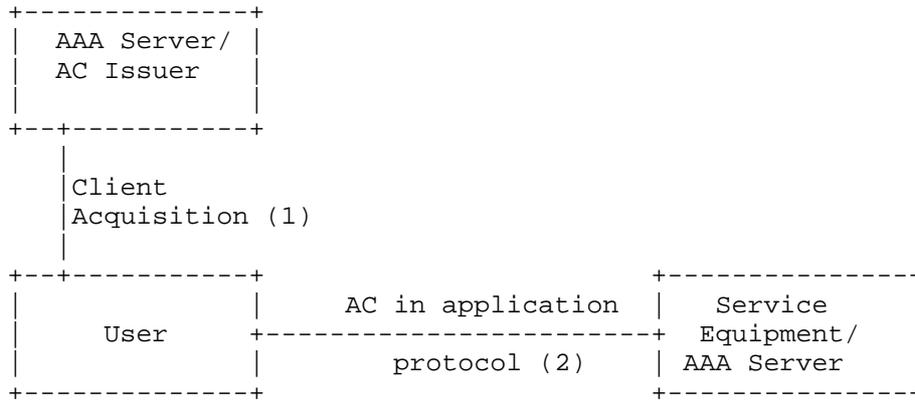
              Fig. 14 -- One example of an AC exchange

In the diagram, the user first contacts the AC Issuer and then
incorporates the AC into the application protocol. The Service
Equipment must then validate the AC and use it as the basis for the
access decision (this functionality may be distributed between a PEP
and PDP).

6. Resource Management

   Authorization requests may be chained through a set of servers, as
   described in previous sections. Each of the servers may have a
   contractual relationship with servers on either side of it in the
   chain. In many of the applications being considered, the
   authorization results in establishing of an ongoing service which we
   call a session. Each of the servers involved in the authorization
   may also want to keep track of the state of the session, and be able
   to effect changes to the session if required. To make it simple to
   discuss this capability, we assume that each AAA Server MAY have a
   Resource Manager component. Resource Managers tracking the same
   session need to be able to initiate changes to the session, and to
   inform other Resource Managers when changes occur. Communication
   between Resource Managers creates requirements for an authorization
   protocol.

   An example of the use of resource management might be a user which
   sets up a QoS path through two ISPs, and while this path is active,
   one of the ISPs gets a request for more bandwidth from a higher
   priority user. The ISP may need to take some bandwidth from a the
   lower priority user's previously allocated session and give it to the

new request. To do this, each of the administrations in the
authorization path must be informed and agree to the change (this
could be considered to be "authorizing the new value").

6.1. Session Management and State Synchronization

When an AAA Server grants authorization of some resource to an AAA
requester (either a User or another AAA Server), the server may need
to maintain session state information. This is used to make
decisions about new sessions based on the state of current sessions,
and to allow monitoring of sessions by all interested AAA Servers.

Each session is identified by a session identifier, which must be
unique within each AAA Server. Communication between AAA Servers
must include the session identifier. It is desirable that the
session identifier is the same across all AAA servers, otherwise each
server will have to map identifiers from other servers to its own
identifiers. A single session identifier significantly simplifies
auditing and session control functions.

Maintaining session state across AAA administrative boundaries
increases the complexity of the problem, especially if each AAA
Server in the trust chain must keep state as well. This can be
viewed as an interdomain database replication problem. The protocol
must include tools to help manage replicated state. Some of the
problems to be addressed are:

a) Service Equipment must be able to notify its Resource Manager when
   a session terminates or changes state in some other way. The
   Resource Manager must inform other Resource Managers which keep
   state for this session.

b) The Resource Manager will need to set a time limit for each
   session which must be refreshed by having the Resource Manager
   query for authoritative status or by having the authoritative
   source send periodic keep alive messages that are forwarded to all
   Resource Managers in the authorization chain. Determining the
   appropriate session lifetime may be application specific and
   depends on the acceptable level of risk. If the service being
   offered is billed based on time, the session lifetime may need to
   be relatively small; if the service is billed on usage, the
   lifetime may be relatively large.

c) Any Resource Manager in the chain must have the ability to
   terminate a session. This requires the Resource Manager to have
   knowledge of at least the adjacent AAA Servers in the
   authorization chain.

An example of how resource management can be used is in the PPP
dialin application. A home ISP may wish to restrict the number of
concurrent sessions that a user can have at any given time. This is
particularly important when service providers give all-you-can-eat
Internet access. The possibility for fraud is quite large, since a
user could provide his or her username/password to many people,
causing a loss of revenue. Resource management would allow the home
ISP AAA server to identify when a user is active and to reject any
authorization request for the user until termination indication is
received from the NAS or until the session expires.

6.2. The Resource Manager

   This section describes the functions of the Resource Manager in more
   detail.

   The Resource Manager is the component which tracks the state of
   sessions associated with an AAA Server or Service Equipment. It also
   may allocate resources to a session (e.g. IP addresses) and may track
   use of resources allocated by peer resource managers to a session
   (e.g. bandwidth in a foreign administrative domain). The resource
   manager also provides interfaces to allow the User to acquire or
   release authorized sessions.

   The Resource Manager maintains all session specific AAA state
   information required by the AAA Server. That state information may
   include pointers to peer Resource Managers in other administrative
   domains that possess additional AAA state information that refers to
   the same session. The Resource Manager is the anchor point in the
   AAA Server from which a session can be controlled, monitored, and
   coordinated even if that session is consuming network resources or
   services across multiple Service Provider administrative domains.

   The Resource Manager has several important functions:

   a) It allows a Service Provider operations staff to inspect the
      status of any of the allocated resources and services including
      resources that span foreign Service Provider administrative
      boundaries. The peer Resource Managers will cooperatively share
      only the state information subset that is required to assist in
      diagnosing cross-domain trouble tickets. The network operator may
      also modify or altogether cancel one of the User's active
      authorizations.

   b) It is the process contacted by other Resource Managers to inform
      the AAA Server that a specific session has been cancelled. This
      information is relayed to the other peer Resource Managers that
      also know about that session and hence must cancel it.

   c) The Resource Manager conceals the identity and location of its
      private internal AAA components from other administrative domains
      and from the User, while at the same time facilitating cooperation
      between those domains.

   d) The Resource Manager cooperates with "policy servers" or Policy
      Decision Points (PDPs). The Resource Manager maintains internal
      state information, possibly complex cross-administrative domain
      information, supported by dialogues with its peer Resource
      Managers. A policy server can use the state information when
      evaluating a particular policy.

   e) The separation of the Resource Manager and the policy server into
      two distinct architectural components allows a single session to
      span multiple administrative domains, where each administrative
      domain has one or more policy server cooperating with its Resource
      Manager.

   AAA resource managers will normally use the same trust relationships
   needed for authorization sequences. It is possible for independent
   relationships to be established, but that is discouraged.

7. AAA Message Forwarding and Delivery

   An AAA Server is responsible for securely forwarding AAA messages to
   the correct destination system or process in the AAA infrastructure.
   Two well known examples are forwarding AAA messages for a roaming AAA
   service, and forwarding AAA messages for a distributed AAA service.
   The same principle can also be applied to intra-domain
   communications. The message forwarding is done in one of two modes.

   The first mode is when an AAA server needs to forward a message to a
   peer AAA server that has a known "logical destination address" that
   must be resolved by an application-specific procedure into its actual
   network address. Typically the forwarding procedure indexes into a
   database by an application-specific identifier to discover the peer's
   network address. For example, in the roaming dialin application, the
   application-specific identifier may be an NAI. A bandwidth brokerage
   application would use other search indices unique to its problem
   domain to select the addressed peer AAA server. After the address
   resolution procedure has completed successfully, then the AAA server
   transmits the message to its peer over the connection associated with
   that destination network address.

   The second mode is when the AAA server already has an established
   session representing an authorization. The session's state contains
   the addressing and context used to direct the message to its
   destination peer AAA server, PDP, PEP, or User. The message is sent

over the AAA server's connection to that destination peer,
multiplexed with other session's messages. The message must be
qualified by a session identifier that is understood by both end
points. The AAA message's destination may be either intra-
administrative domain, or inter-administrative domain. In the former
case, the destination process may reside on the same system as the
AAA server.

In addition to the above message forwarding processing, the
underlying message delivery service must meet the following
requirements:

- Unicast capability -- An end system can send a message to any
  other end system with minimal latency of session setup/disconnect
  overhead messages, and no end system overhead of keeping state
  information about every potential peer.

- Data integrity and error detection -- This data transport protocol
  assumes an underlying datagram network layer service that includes
  packet discard on error detection, and data integrity protection
  against third party modifications.

- Reliable data transport assurance -- When an end system
  successfully receives a message marked receipt requested, it must
  acknowledge that message to the sending system by either
  piggybacking the acknowledgement on an application-specific reply
  message, or else as a standalone acknowledgement message. The
  sending system maintains a retry timer; when the timer expires,
  the sending system retransmits a copy of its original message. It
  gives up after a configurable number of unsuccessful retries.

- Sequenced data delivery -- If multiple messages are sent between a
  pair of end systems, those messages are delivered to the addressed
  application in the same order as they were transmitted.
  Duplicates are silently suppressed.

- Responsive to network congestion feedback -- When the network
  enters into congestion, the end systems must detect that
  condition, and they must back off their transmission rate until
  the congestion subsides. The back off and recovery algorithms
  must avoid oscillations.

8. End-to-End Security

   When AAA servers communicate through intermediate AAA servers, such
   as brokers, it may be necessary that a part of the payload be secure
   between the originator and the target AAA server. The security
   requirement may consist of one or more of the following: end-to-end

message integrity, confidentiality, replay protection, and
nonrepudiation. Furthermore, it is a requirement that intermediate
AAA servers be able to append information such as local policy to a
message before forwarding the message to its intended destination.
It may also be required that an intermediate AAA Server sign such
appended information.

This requirement has been clearly documented in [10], which describes
many current weaknesses of the RADIUS protocol [11] in roaming
networks since RADIUS does not provide such functionality. One
well-known attack is the ability for the intermediate nodes to modify
critical accounting information, such as a session time.

Most popular security protocols (e.g. IPSec, SSL, etc.) do not
provide the ability to secure a portion of the payload. Therefore, it
may be necessary for the AAA protocol to implement its own security
extensions to provide end-to-end security.

9.   Streamlined Authorization Process

The techniques described above allow for great flexibility in
distributing the components required for authentication and
authorization. However, working groups such as Roamops and MobileIP
have identified requirements to minimize Internet traversals in order
to reduce latency. To support these requirements, data fields
necessary for both authentication and authorization SHOULD be able to
be carried in a single message set. This is especially important
when there are intermediate servers (such as Brokers) in the AAA
chain.

Furthermore, it should be possible for the Brokers to allow end-to-
end (direct) authentication and authorization. This can be done as
follows. The User Home Organization generates a ticket which is
signed using the UHO's private key. The ticket is carried in the
accounting messages. The accounting messages must flow through the
Broker since the Broker is acting as the settlement agent and
requires this information. There are Brokers that will require to be
in the authentication and authorization path as well since they will
use this information to detect fraudulent activity, so the above
should be optional.

In order for end-to-end authentication and authorization to occur, it
may be necessary for the Broker to act as a certificate authority.
All members of the roaming consortium would be able to trust each
other (to an extent) using the certificates. A Service Provider's
AAA server that sends a request to the Broker should be able to
receive a redirect message which would allow the two peers (Service
Provider and UHO) to interact directly. The redirect message from

the Broker should include the UHO's certificate, which eliminates the
Service Provider from accessing the certificate archive. The request
from the Service Provider could include its own certificate, and a
token from the Broker's redirect message that is timestamped and
guarantees that the Service Provider is in good standing with the
Broker. This eliminates the home domain from accessing the
Certificate Revocation List (CRL).

10. Summary of the Authorization Framework

   The above has introduced the basic players in an authorization
   transaction as User, User Home Organization, Service Provider's AAA
   Server, and Service Equipment. It has discussed relationships
   between entities based on agreements or contracts, and on "trust".
   Examples of authorization sequences have been given.

   Concepts of roaming and distributed services have been briefly
   described. Combination of roaming and distributed services was also
   considered and the concept of a "wholesaler" or Broker was
   introduced. We have considered the use of policies and attribute
   certificates to store and transmit authorization data. We discussed
   the problem of managing the resources to which access has been
   authorized including the problem of tracking state information for
   session-oriented services, and we defined the Resource Manager
   component of a AAA Server. We considered the problem of forwarding
   AAA messages among servers in possibly different administrative
   domains. We considered the need for end-to-end security of portions
   of the payload of authorization messages that pass through
   intermediate AAA Servers. Finally we stressed the need for support
   of a streamlined authorization process that minimizes delay for
   latency-sensitive applications.

   The intent is that this will provide support for discussing and
   understanding requirements of specific applications that need
   authorization services.

11. Security Considerations

   Authorization is itself a security mechanism. As such, it is
   important that authorization protocols cannot easily be abused to
   circumvent the protection they are intended to ensure. It is the
   responsibility of protocol designers to design their protocols to be
   resilient against well-known types of attacks. The following are
   some considerations that may guide protocol designers in the
   development of authorization protocols.

   Authorization protocols must not be susceptible to replay attacks.
   If authentication data is carried with the authorization data, for
   example, the authentication protocol used must either be impervious
   to replay or else the confidentiality of the authentication data must
   be protected.

   If proxying is required, the authorization protocol must not be
   susceptible to man-in-the-middle attacks.

   If the push model is used, the confidentiality of the authorization
   data must be ensured so that it may not be hijacked by third parties
   and used to obtain a service fraudulently.

   If the agent model is used, the binding between the authorization and
   the service itself must be protected to prevent service authorized to
   one party from being fraudulently received by another.

   In addition to guarding against circumvention, authorization
   protocols designed according to this framework will have some
   intrinsic security requirements. These are included among the
   requirements in [2] and summarized briefly below.

   Among the intrinsic security needs is the fact that authorization
   protocols may carry sensitive information. It is necessary to
   protect such information from disclosure to unauthorized parties
   including (as discussed in section 8) even certain parties involved
   in the authorization decision.

   We have discussed the use of multi-party trust chains involving
   relaying of authorization data through brokers or other parties. In
   such cases, the integrity of the chain must be maintained. It may be
   necessary to protect the data exchanged between parties using such
   mechanisms as encryption and digital signatures.

   Finally, because authorization will be necessary to gain access to
   many Internet services, a denial of service attack against an
   authorization server can be just as effective as a denial of service
   attack against the service equipment itself in preventing access to
   Internet services.

Glossary

   Attribute Certificate -- structure containing authorization
      attributes which is digitally signed using public key
      cryptography.

Contract Relationship -- a relation established between two or more
    business entities where terms and conditions determine the
    exchange of goods or services.

Distributed Service -- a service that is provided by more than one
    Service Provider acting in concert.

Dynamic Trust Relationship -- a secure relationship which is
    dynamically created between two entities who may never have had
    any prior relationship. This relationship can be created if the
    involved entities have a mutually trusted third party. Example: A
    merchant trusts a cardholder at the time of a payment transaction
    because they both are known by a credit card organization.

Policy Decision Point (PDP) -- The point where policy decisions are
    made.

Policy Enforcement Point (PEP) -- The point where the policy
    decisions are actually enforced.

Resource Manager -- the component of an AAA Server which tracks the
    state of sessions associated with the AAA Server or its associated
    Service Equipment and provides an anchor point from which a
    session can be controlled, monitored, and coordinated.

Roaming -- An authorization transaction in which the Service Provider
    and the User Home Organization are two different organizations.
    (Note that the dialin application is one for which roaming has
    been actively considered, but this definition encompasses other
    applications as well.)

Security Association -- a collection of security contexts, between a
    pair of nodes, which may be applied to protocol messages exchanged
    between them. Each context indicates an authentication algorithm
    and mode, a secret (a shared key, or appropriate public/private
    key pair), and a style of replay protection in use. [12]

Service Equipment -- the equipment which provides a service.

Service Provider -- an organization which provides a service.

Static Trust Relationship -- a pre-established secure relationship
    between two entities created by a trusted party. This
    relationship facilitates the exchange of AAA messages with a
    certain level of security and traceability. Example: A network
    operator (trusted party) who has access to the wiring closet

creates a connection between a user's wall outlet and a particular
network port. The user is thereafter trusted -- to a certain
level -- to be connected to this particular network port.

User -- the entity seeking authorization to use a resource or a
service.

User Home Organization (UHO) -- An organization with whom the User
has a contractual relationship which can authenticate the User and
may be able to authorize access to resources or services.

References

[1] Bradner, S., "The Internet Standards Process -- Revision 3", BCP
     9, RFC 2026, October 1996.

[2] Farrell, S., Vollbrecht, J., Calhoun, P., Gommans, L., Gross,
     G., de Bruijn, B., de Laat, C., Holdrege, M. and D. Spence, "AAA
     Authorization Requirements", RFC 2906, August 2000.

[3] Vollbrecht, J., Calhoun, P., Farrell, S., Gommans, L., Gross,
     G., de Bruijn, B., de Laat, C., Holdrege, M. and D. Spence, "AAA
     Authorization Application Examples", RFC 2905, August 2000.

[4] Bradner, S., "Key words for use in RFCs to Indicate Requirement
     Levels", BCP 14, RFC 2119, March 1997.

[5] Stevens, M., "Policy Framework", Work in Progress.

[6] Strassner, John, Ed Ellesson, and Bob Moore, "Policy Core
     Information Model -- Version 1 Specification", Work in Progress.

[7] Strassner, John, et al, "Policy Framework LDAP Core Schema",
     Work in Progress.

[8] Farrell, Stephen and Russell Housley, "An Internet Attribute
     Certificate Profile for Authorization", Work in Progress.

[9] Housley, R., Ford, W., Polk, W. and D. Solo, "Internet X.509
     Public Key Infrastructure -- Certificate and CRL Profile", RFC
     2459, January 1999.

[10] Aboba, B. and J. Vollbrecht, "Proxy Chaining and Policy
     Implementation in Roaming", RFC 2607, June 1999.

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

   [12] Perkins, C., "IP Mobility Support", RFC 2002, October 1996.

   [13] Yavatkar, R., Pendarakis, D. and R. Guerin, "A Framework for
        Policy-based Admission Control", RFC 2753, January 2000.

Authors' Addresses

   John R. Vollbrecht
   Interlink Networks, Inc.
   775 Technology Drive, Suite 200
   Ann Arbor, MI 48108
   USA

   Phone: +1 734 821 1205
   Fax: +1 734 821 1235
   Mail: jrv@interlinknetworks.com

   Pat R. Calhoun
   Network and Security Research Center, Sun Labs
   Sun Microsystems, Inc.
   15 Network Circle
   Menlo Park, California, 94025
   USA

   Phone: +1 650 786 7733
   Fax: +1 650 786 6445
   EMail: pcalhoun@eng.sun.com

   Stephen Farrell
   Baltimore Technologies
   61 Fitzwilliam Lane
   Dublin 2
   Ireland

   Phone: +353 1 647 7406
   Fax: +353 1 647 7499
   EMail: stephen.farrell@baltimore.ie

Leon Gommans
Enterasys Networks EMEA
Kerkplein 24
2841 XM Moordrecht
The Netherlands

Phone: +31 182 379279
email: gommans@cabletron.com
       or at University of Utrecht:
       l.h.m.gommans@phys.uu.nl

George M. Gross
Lucent Technologies
184 Liberty Corner Road, m.s. LC2N-D13
Warren, NJ 07059
USA

Phone: +1 908 580 4589
Fax: +1 908-580-4991
Email: gmgross@lucent.com

Betty de Bruijn
Interpay Nederland B.V.
Eendrachtlaan 315
3526 LB Utrecht
The Netherlands

Phone: +31 30 2835104
EMail: betty@euronet.nl

Cees T.A.M. de Laat
Physics and Astronomy dept.
Utrecht University
Pincetonplein 5,
3584CC Utrecht
Netherlands

Phone: +31 30 2534585
Phone: +31 30 2537555
EMail: delaat@phys.uu.nl

   Matt Holdrege
   ipVerse
   223 Ximeno Ave.
   Long Beach, CA 90803

   EMail: matt@ipverse.com

   David W. Spence
   Interlink Networks, Inc.
   775 Technology Drive, Suite 200
   Ann Arbor, MI   48108
   USA

   Phone: +1 734 821 1203
   Fax: +1 734 821 1235
   EMail: dspence@interlinknetworks.com

Full Copyright Statement

Acknowledgement