Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: January 10, 2008

S. Greco Polito
H. Schulzrinne
Columbia University
July 9, 2007

Authentication, Authorization, Accounting and Billing of Roaming Users using SAML
draft-greco-sipping-roaming-01

Status of this Memo

Copyright Notice

Abstract

Roaming services allow users that have a contract with a voice service provider to use access resources owned by other providers known as internet access providers. This draft proposes a token-based Authentication, Authorization, Accounting (AAA) and billing model for roaming users supporting the Session Initiation Protocol (SIP). It also introduces a protocol solution for the proposed model that is based on the Security Assertion Markup Language (SAML) protocol and the Hypertext Transfer Protocol (HTTP).

Table of Contents

1. Introduction

The authentication, authorizzation, accounting (AAA) and billing of roaming VoIP users requires interaction between voice service providers (VSP) contracted by users, and Internet access providers (IAP) that own access resources. We propose an AAA and billing model in which we want to leverage the business, billing and trust relationship that users have with their VSPs to give the customers of VoIP users roaming access to various wireless and wired internet access providers, both traditional carriers, such as 3G providers, as well as local hotspot providers. We propose an AAA and billing solution in which VSPs provide their users with tokens containing all the information that IAPs need for verifying their authorization and activating the accounting and billing procedures. These tokens are issued and signed by a third party, called guarantor, which also provides accounting and billing mediation services. Using the model proposed in this draft, by adding a smaller set of guarantors, IAPs that do not know VSPs, and vice versa, can offer services to customers of VSPs. Figure 1 shows the trust model of the AAA solution proposed in this document.

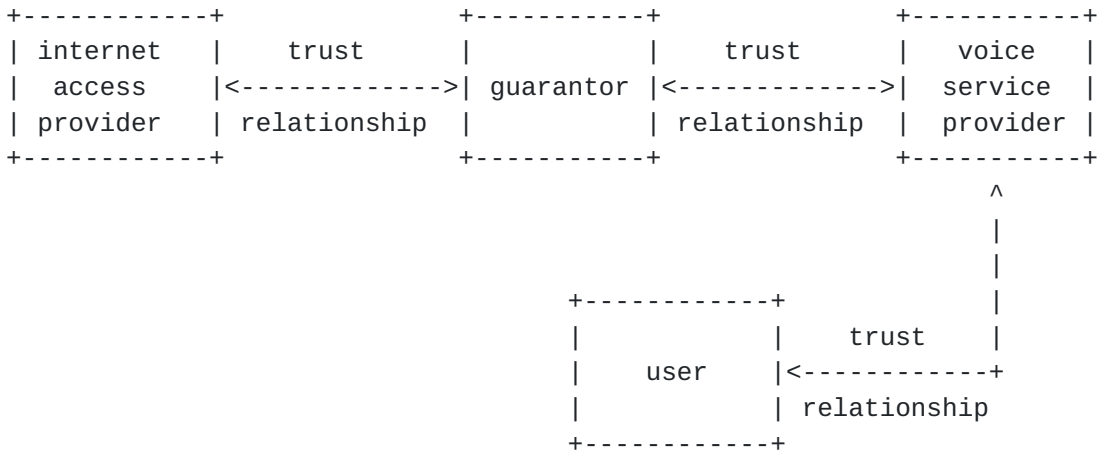

Figure 1: trust model

The role of the guarantor is somewhat similar to that of a clearinghouse for settlement and billing, but it differ in authorization. While clearinghouses are used for authorizing users' calls, the guarantor provides authorization for the use of access network resources. One of the main protocols for clearinghouse-based models is the Open Settlements Protocol (OSP) [OSP]. OSP inroduces a token-based authorization model for interdomain calls in which telephony operators can ask a clearinghouse for tokens proving

the right of their users to place calls toward some destination. In this draft, we extend the concept of tokens introduced by OSP, focusing on the authorization and authentication of roaming users instead of the authorization of calls. We define a token with a long lifetime that can be held by users and allows them to be authenticated and authorized to use access network resources from internet access providers that they do not mantain a business relationship with. We propose an authentication and authorizaion model in which users can provide the IAPs with their tokens. The model does not require interaction between IAPs and VSPs of visiting users for the verification of the user's identity and authorization profile and avoids run home AA (authentication and authorization) messaging that may significaly increase the AA delay if the access network is far from the infrastructure of the VSP where the user's credentials are stored. In the model, security associations between IAPs and VSPs are not required.

The token-based AAA and billing solution proposed in this draft is composed of the procedures that allow users to ask for and obtain tokens from their VSPs, which are called token provisioning procedures, the procedures for getting access authentication and authorization using the token, and the ones for accounting and billing.

1.1. Token provisioning

The procedures that allow users to obtain tokens are called token provisioning procedures and involve users, VSPs and guarantor. Users activate these by a token request to their VSP. VSPs interact with the guarantor in order to build tokens for their users, providing the guarantor with information about users such as their authorization profiles. The guarantor uses this information to issue and sign AAA and billing tokens. The signature guarantees data origin, integrity and non repudiation of the token. The set of procedures performed by VSPs and guarantor for issuing tokens are called token building procedures. In Section 4 , we provide a complete description of the AAA and billing token and the messages used for the communication between VSPs and guarantor based on SAML. SAML [saml-core-2.0-os] is an OASIS protocol for the description and exchange of security information between partners. We define a new SAML profile called roaming profile. This SAML roaming profile describes the set of extensions to the SAML protocol that allows to use it as description protocol for the AAA and billing model of roaming users introduced in this draft.

1.2. Access authentication and authorization

The token-based access authentication and authorization consist of procedures that allow users to provide the access network with their tokens when they ask for access to the Internet and the access network to verify if tokens are valid.

1.3. Accounting and billing

The accounting and billing procedures allow to charge VoIP users for the access network resources used. In this draft, three accounting and charging models are introduced, showing the differences between pre-paid and post-paid billing.

1.4. Content

The remained of the document is organized as follows. In Section 2 we provide the terminology used in the subsequent sections. In Section 3 we introduce the token-based AAA and billing model and we describe the behavior of users, voice service providers, guarantors and internet access providers for token provisioning and token-based access authentication and authorization. In Section 4 we introduce the SAML roaming profile. Specifically, Section 4.1 defines the rules for the SAML-based description of the messages used by VSPs and the guarantor in the token building phase. Section 4.2 introduces the SAML roaming assertion that is a SAML-based description of the token. Section 5 focus on the issue regarding accounting and billing. It introduces pre-paid and post-paid charging models and the concept of price negotiation between users and IAPs for payment of Internet access resources. In Section 6 we provide our security considerations about the protocol proposed. Section 7 contains details about the extensions to the SAML elements that support the SAML roaming profile and the description of the messages used for price negotiation between IAPs and users.

2. Terminology

Voice Service Provider (VSP): provider of multimedia voice services (VoIP services).

Internet access provider (IAP): provider of access network resources such as wireless local access networks and 3G networks.

Guarantor: provider of mediation services for AAA and billing.

User: customer contracting with a VSP for obtaining VoIP and roaming services.

Security Assertion Markup Language (SAML): set of specifications describing security assertions that are encoded in Extensible Markup Language (XML) [XML], profiles for attaching the assertions to various protocols and frameworks, the request/response protocol used to obtain the assertions, and bindings of this protocol to various transfer protocols [saml-glossary-2.0-os].

SAML assertion: piece of data produced by a SAML authority regarding either an act of authentication performed on a subject, attribute information about the subject, or authorization data applying to the subject with respect to a specified resource [saml-glossary-2.0-os]

AAA and billing token or roaming token: element asserting the user's right to access the network. It contains user authorization information and information needed to the IAP for activating the accounting and billing procedures.

Roaming assertion: SAML-based description of the token.

SAML Simple Object Access Protocol (SOAP) binding: definition on how to use SOAP to send and receive SAML requests and responses. This binding has protocol-independent aspects, but also calls out the use of SOAP over HTTP [saml-binding-2.0-os].

Roaming SAML profile: set of constraints and rules for using the SAML protocol and the SAML assertion capability in the roaming AAA and billing context of use.

For the SIP terminology, see [RFC3261].

For the overall SAML terminology, see [saml-glossary-2.0-os].

3. Protocol description

Figure 2 shows the messages exchanged between user, VSP and guarantor for the token provisioning procedure, and the one between user, IAP, guarantor and VSP for token-based AA. The token provisioning procedure is performed in the following steps:

1. The user starts the procedure sending a token request to its VSP, using an HTTP GET request.

2. When the VSP receives a token request, it asks the user for its authentication and authorization credentials. The user's authentication avoids that malicious subjects obtain tokens impersonating authorized users. In this scenario, the HTTP digest method is used for user authentication and SIP credentials may be reused.

3. The user resends its token request including the information required for its authentication in the HTTP GET message.

4. The VSP verifies the user's identity. If the user authentication is successful, the VSP sends a token building request to the guarantor, asking it to issue and sign a token for its user. This request contains information that the guarantor needs for building the token such as the user's profile and the token length (see Section 4.1 for the description of this request).

5. The guarantor builds and signs the token and returns it to the VSP. The token is described using SAML (see Section 4.2) and inserted in a token building response. Section 4.1 describes the token.

6. The VSP, extracts the token from the token building response, and sends it to the user in an HTTP 200 OK response.

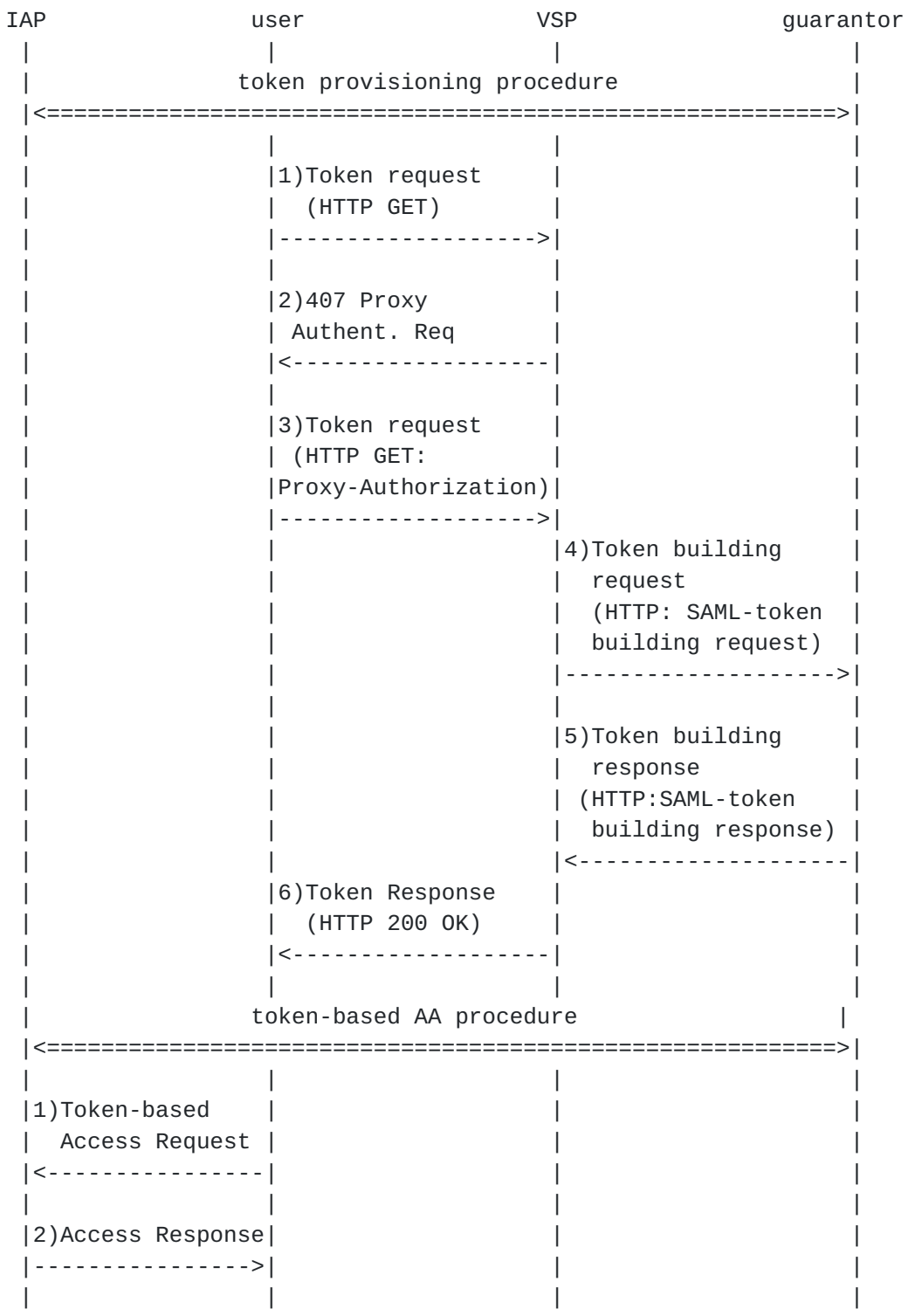

Figure 2: messaging for token provisioning and token-based AAA

The user stores the token received and uses it to obtain AA from IAPs visited late. The token-based AA of roaming users requires interaction between the user and the IAP:

1. The user sends a token-based access request containing the token to the IAP asking for access to the Internet. The token has to be sent using a secure transport channel to prevent malicious subjects from intercepting the flow between user and access network and make copies of the token.

2. The IAP verifies the validity of the token and returns a response containing the authorization verification result to the user.

The protocols used to provide the access network with the user's token may depend on the access network technology. In all cases, a secure channel has to be build between user and access network to provide the access network with the token. For example, if the access network technology supports EAP (Extensible Authentication Protocol) [RFC3748], the EAP-TTLS [draft-funk-eap-ttls-v1-01] method may be used to constract the secure channel.

3.1. User behavior

The user supports HTTP and asks the VSP for the token using an HTTP GET message. The URL of the GET message points to a location where the VSP stores the user's token. The token request causes the activation of the HTTP digest authentication procedure between user and VSP. If the authentication is successful, the VSP provides the user with the token in a HTTP 200 OK message. The user will use this token until its expiration time to ask IAPs for access to the Internet.

3.2. VSP behavior

The VSP uses HTTP to communicate with the user and the guarantor. Each time the VSP receives a token request, it verifies the user identity and, if the authentication is successful, it verifies if the last token issued for the user is expired (it is assumed that the VSP maintains a local copy of the token) and, if the last issued token is expired, the VSP activates the token building procedure asking the guarantor for a new signed token with a token building request. The VSP is the entity that maintains a contract with the user and knows his authorization profile. The VSP includes the user identity, the user's authorization profile, and the token lifetime in the token building request. It structures this request in the format of SAML request using the set of extensions defined in Section 4.2. When the VSP receives the token from the guarantor, it sends it to the user in the body of the 200 OK response to the HTTP GET request.

3.3. Guarantor behavior

The guarantor builds and signs tokens for users that are customers of VSPs. When the guarantor receives the token building request, it builds the token, inserting the information contained in the request along with its identifier, the domain name of the VSP, the signature and its certificate. The guarantor signs the token using its private key. The signature guarantees integrity, data origin and non repudiation of the token. The guarantor composes the token in the form of a SAML assertion as described in Section 4.2 and returns it to the VSP. It uses the SAML assertion response specifications for describing the token building response (see Section 4.1 for details). The guarantor is responsible for paying the IAPs that authorize users on the basis of its signature on the tokens. The guarantor is reimbursed by the VSPs which are the only entities that hold the users' credentials and can charge them (see Section 5 for details about accounting and billing).

3.4. IAP behavior

After receiving a token from a visitng user, the IAP verifies the token integrity. The IAP uses the guarantor's public key carried in the token for the verification. If the token is not corrupted, the IAP verifies its expiration time. If the token has not expired, the IAP allows the user to access the network. The digital signature guarantees non repudiation of the token and gives the right to ask the guarantor for the payment of the access resources used by visiting users to the IAPs. IAPs are not interested in knowing the identity of the VSP of visiting users because the guarantor provides the payment for the access resources instead of the VSP.

4. Roaming SAML profile

SAML defines a framework for the exchange of security information about a subject between partners called requesting, asserting and relying parties. The asserting party is the entity that produces an authentication and authorization assertion about a subject when required by the requesting party, while the relying party uses the assertion for authorizing the subject.

This draft defines a new SAML profile, called roaming SAML profile. It defines a set of specifications that allows to use SAML for the description of the token and the token building request and response introduced above.

In the SAML roaming profile, the VSPs assume the role of SAML requesting parties, the guarantor the one of asserting party, and IAPs the one of relying parties. Below, we will call the token described using SAML "roaming assertion" and we will assume that user and VSP support SIP (Session Initiation Protocol) [RFC3261].

4.1. Token building request and response

The token building request is the message used by the VSP for asking the guarantor for the token (see step 3 of the token provisioning procedure of Figure 2). It is structured as a SAML request and is described using the following elements:

- The Issuer. This SAML element contains the identifier of the entity generating the request message. Here, it contains the VSP's domain name.

- The Subject. It contains the identifier of the subject of the assertion and it is set to the SIP AoR of the user in the token building request.

- The Conditions. This element allows the SAML requesting party to impose conditions limiting the validity and/or use of the assertion. Here, it is used by the VSP to provide the guarantor with information about the assertion lifetime and the user profile. For the description of the token lifetime we use the SAML NotBefore and NotOnOrAfter attributes of the assertion element. The first is the start time of the token, the second the expiration time of the token. For the description of the user profile and its inclusion in the Conditions element, we propose an extension of the SAML ConditionAbstractType described in Section 7.1.

Figure 3 shows an example of the token building request described

using the elements introduced above. See Section 7.2 for the description of the XML schema of this request.

```
<?xml version="1.0"?>
<!-- token building request compiled by the VSP of the user
bob@example.com. The request contains the ID attribute with the
identifier of the request, the IssueInstant attribute with the
time instant of issue of the request expressed in UTC form, the
Issuer element with the domain name of the service provider, the
Subject element with the identifier of the user, the conditions
element with the start time and the expiration time of the token
in the NotBefore and NotOnOrAfter attributes in UTC form, the user
QoS class -->
<req:token_building_request
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion"
 xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"
 xmlns:cond="http://www.tti.unipa.it/~silvana/tokencondition"
 xmlns:pt="http://www.tti.unipa.it/~silvana/"
 xmlns:req="http://www.tti.unipa.it/~silvana/requesttype"
 xsi:schemaLocation="http://www.tti.unipa.it/~silvana/requesttype
 token_building_request.xsd"
 ID="_a75adf55-01d7-40cc-929f-dbd8372ebdfc" Version="2.0"
 IssueInstant="2006-02-01T00:46:02Z">
 <saml:Issuer>serviceprovider.example.com</saml:Issuer>
 <saml:Subject>
  <saml:NameID
  Format="urn:oasis:names:tc:SAML:1.1:nameid-format:emailAddress">
   bob@example.com
  </saml:NameID>
 </saml:Subject>
 <saml:Conditions NotBefore="2006-02-01T00:55:02Z"
  NotOnOrAfter="2006-03-01T00:55:02Z">
  <saml:Condition xsi:type="cond:condition_profileType">
   <cond:UserProfile>
    <cond:UserClass>Gold</cond:UserClass>
   </cond:UserProfile>
  </saml:Condition>
 </saml:Conditions>
</req:token_building_request>
```

Figure 3: XML token building request example

The guarantor delivers the token to the VSP into the token building response (see step 4 of the assertion provisioning procedure of Figure 2). The guarantor uses the SAML description rules for a

response to a generic SAML query message [saml-core-2.0-os] to contruct the token building response, mapping this response in the SAML Response element. The SOAP SAML binding described in [saml-binding-2.0-os] is used for the transport of the token building request and response. It defines how to use HTTP and SOAP to send and receive SAML requests and responses.

4.2. SAML roaming assertion

We call the token described using the SAML assertion [saml-core-2.0-os] specifications and the set of extensions provided in this section "roaming assertion". The token is mapped in the SAML Assertion element and its content is described using the following SAML elements:

- The Issuer. In a generic SAML assertion, this element contains the identifier of the identity that is making the claim in the assertion and it is set with the guarantor identifier in a roaming-assertion.

- The Signature. This element contains the signature provided by the guarantor. Following the OASIS specifications [saml-core-2.0-os] [saml-sec-consider-2.0-os] for this element, the guarantor computes it using the XML signature specifications [xmldsig].

- The Subject. It contains the SIP AoR of the user which is the subject of the statement in the roaming assertion.

- The Conditions. In a generic SAML assertion, this element contains information that must be evaluated by the relying party for the verification and use of the assertion. Here, the NotBefore and NOtOnOrAfter attributes of this element are used to describe the start and expiration time of the token.

- The Statement. It is an SAML extension point defined for allowing the use of SAML in new contexts. This draft defines an extension of the SAML type of the Statement element in order to include the domain name of the VSP and information about the user profile in this element . The XML schema of the extension to the type of Statement element is provided in Section 7.3.

Figure 4 shows an example of the roaming assertion.

```
<?xml version="1.0"?>
<!DOCTYPE saml:Assertion>
<!--  Roaming assertion example. The assertion element contains the
value of the assertion identifier in its ID attribute and the time
```

```
   instant of issue of the assertion in the IssueInstant attribute.
   The assertion contains the Issuer element with the guarantor
   identifier; the Signature with information about the method and the
   algorithms used, the digest value, the signature value and the
   certificate of the guarantor; the Subject with the user identifier;
   the Conditions that stores the token lifetime information; the
   Statement with the VSP domain name and the user profile -->
   <saml:Assertion xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion"
   xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"
   xmlns:tk="http://www.tti.unipa.it/~silvana/"
   xmlns:tkc="http://www.tti.unipa.it/~silvana/tokencondition"
   ID="_a75adf55-01d7-40cc-929f-dbd8372ebdfc"
   IssueInstant="2006-02-01T00:50:02Z" Version="2.0"
   xsi:schemaLocation="http://www.tti.unipa.it/~silvana/
   roaming_statementType.xsd">
     <saml:Issuer>guarantor.example.com</saml:Issuer>
     <Signature xmlns="http://www.w3.org/2000/09/xmldsig#">
       <SignedInfo>
        <CanonicalizationMethod
        Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#"/>
        <SignatureMethod
        Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1"/>
        <Reference>
           <Transforms>
             <Transform
   Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature"/>
           </Transforms>
           <DigestMethod
           Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
           <DigestValue>UmhxeI9DkkQU0iVs4FfiXYvTkMQ=</DigestValue>
        </Reference>
       </SignedInfo>
       <SignatureValue>jNFui.........
       </SignatureValue>
       <KeyInfo>
        <X509Data>
         <X509Certificate>MIIE...........
         </X509Certificate>
        </X509Data>
       </KeyInfo>
     </Signature>
     <saml:Subject>
       <saml:NameID
       Format="urn:oasis:names:tc:SAML:1.1:nameid-format:emailAddress">
        bob@example.com
       </saml:NameID>
     </saml:Subject>
```

```
    <saml:Conditions NotBefore="2006-02-01T00:55:02Z"
    NotOnOrAfter="2006-03-01T00:55:02Z">
    </saml:Conditions>
    <saml:Statement xsi:type="tk:roaming_statementType">
      <tk:ServiceProviderID>
      serviceprovider.example.com
      </tk:ServiceProviderID>
      <tk:policy_info>
          <tkc:UserProfile>
            <tkc:UserClass>Gold</tkc:UserClass>
          </tkc:UserProfile>
      </tk:policy_info>
    </saml:Statement>
  </saml:Assertion>
```

Figure 4: XML roaming assertion example

5. Accounting and billing

The accounting and billing model proposed in this document delegates to the guarantor the provisioning of accounting and billing intermediation services between IAPs and VSPs. The guarantor is responsible for paying the IAPs that provide roaming users with their services, while the guarantor is reimbursed by the VSPs of the users. The digital signature on the token guarantees non-repudiation of the token. Signed tokens give the IAPs the right to ask the guarantor for the payment of the resources provided to visiting users.

IAPs have to issue accounting records with information about the amount of resources for which a user has to be charged in order to ask for and receive payment. In the following three billing models are proposed. The first two models (see Section 5.1) are based on the assumption that VSPs, IAPs and guarantor define the price of the roaming service in advance and that users are provided with information about this price when they contract with their VSPs. The third model introduced (see Section 5.2) allows a per-session negotiation of the price of the service between IAPs and visited users. This method allows each IAP to offer its own pricing plan to visiting users.

5.1. Billing without price negotiation

The billing without negotiation of the price of the roaming service, is based on the assumpition that users know this price when they asks IAPs for access to the Internet. This avoids any exchange between IAPs and users for the negotiation of the price of the service at the connection time. The following subsections describe two models for post-paid and pre-paid billing without price negotiation, respectively.

5.1.1. Post-paid billing without price negotiation

In post-paid billing models, the charging procedures are activated after the use of resources or services. In the roaming context, if a post-paid billing method is used, users are allowed to access the Internet without controlling in advance if they can pay for the services used, i.e., without controlling the availability of money from the user before providing him with access network resources. The main advantage of post-paid billing methods is that they allow to charge users for the real amount of resources used on the basis of accounting records provided by the IAPs.

The main drawback of post-paid methods is the risk of insolvency of users. It is possible implement per-timeslot charging methods to reduce this risk. When per-timeslot methods are used, the session is

divided in slots having a pre-defined length and the accounting and charging procedures are activated at the end of each timeslot. This way, the risk of insolvency of the user depends on the length of the timeslot and may be reduced using short timeslots. On the other hands, short timeslots increase the accounting and charging signaling and processing overhead.

The token-based access model introduced in the previous sections support both per-session and per-timeslot post-paid methods. In the per-session method, IAPs issue accounting records for the total length of the connection session of visiting users and send them to the guarantor. The guarantor provides these records to the VPSs of the users that activates the charging procedures. Similarly, if a per-timeslot method is used, IAPs issue accounting records for each timeslot of connection and provide these records to the guarantor at the end of each timeslot.

5.1.2. Pre-paid billing without price negotiation

In pre-paid billing the user is charged before using the resources. Pre-paid billing solutions have the advantage that they guarantee payment from users for the resources used. They suffer from the disadvanage that users may be charged for more than the resource really used because the billing is performed before having information about the amount of resources used on the basis of a priori agreements.

In pre-paid billing, the connection sessions are usually divided into timeslots (as in the per-timeslot post-paid solution) and the user is charged for each timeslot of connection in advance. In the pre-paid billing, IAPs issue a request for user charging at the beginning of each session slot. This request is sent to the guarantor which forwards it to the user's VSP. The user is charged for each session slot in advance and the system does not assume risks for user insolvency. The procedures for the payment of the IAPs from the guarantor and the one of the guarantor from the VSPs are independent on the procedure for charging the user and may be performed at any time depending on settlements between IAPs, VSPs and guarantor.

Figure 5 shows the messaging between IAP, guarantor and VSP for both the pre-paid and post-paid billing models introduced in this section. The messaging is composed of a charging request issued by the IAP and sent to the guarantor which forwards it to the VSP. In the case of post-paid billing IAPs include in the charging request the accounting records describing the amount of resources used by the user with information such as the amount of data exchanged by the user and the length of the connection. In the case of pre-paid billing, the charging request is sent at the beginning of the connection or at the

beginning of each time slot and the user is charged based on the length of the connection slot.

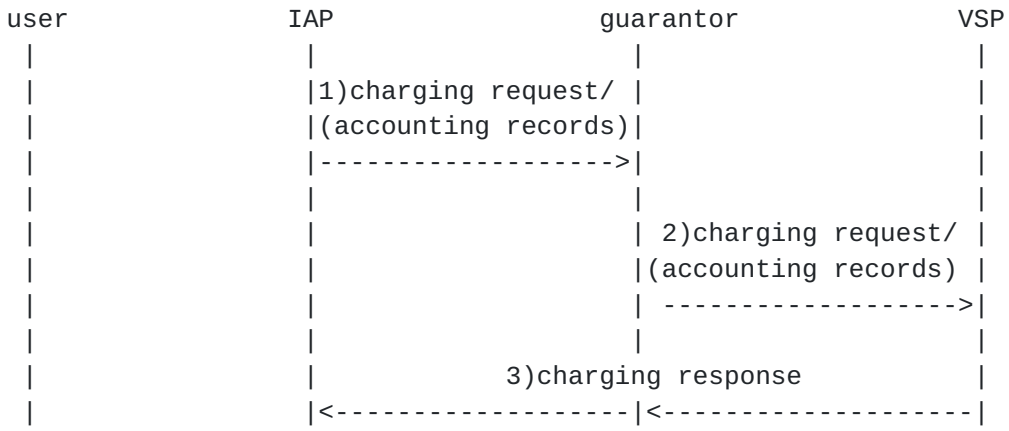

Figure 5: Messaging for pre-paid and post-paid billing without price negotiation

5.2. Pre-paid billing with price negotiation

The pre-paid and post-paid solutions introduced above assume that VSP, IAP and guarantor have defined the price of the roaming service and that users know this price before asking a IAP for access to the Internet. This section describes a method that allows IAPs to have independent price plans and inform visiting users about their price plans at the connection time following the model proposed in [draft-jennings-sipping-pay-05] for billing of multi-provider VoIP calls. The charging method proposed requires that users are provided with a pair of public and private key, and that VSPs are provided with a public certificate issued by the guarantor. The model consists of the following interactions between user, IAP and VSP:

- The IAP sends a contract offer to the user after receiving an access request with a token and verifying the validity of this token. This offer contains the description of the price plan of the IAP, the identifier of the IAP and a timestamp element with the issuing instant of the offer.

- The user may accept the offer or reject it. If he accepts the offer, he signs the offer received with his private key and sends the signed offer to its VSP. The payment offer signed by the user assumes the meaning of charging request from the user to its VSP. The user uses this message to ask its VSP to charge him following the charging rules described in the message (amount of anticipated

cost, per-unit cost, etc..) See Figure 13 for the description of the contract offer.

- The VSP that receives a charging request charges the user on the basis of the content of the offer and returns a charging receipt to the IAP. The charging receipt is build adding a digital signature and the certificate of the VSP to the original contract offer. The VSP extracts the contract offer from the charging request message.

- When the IAP receives the receipt, he can verify its freshness through the timestamp of the contract offer encapsulated in it. The certificate carried in the message guarantees that the VSP is trusted by the guarantor and allows to verify data origin and integrity of the receipt. If the verification of the receipt is successful, the IAP allows the user to access to the network.

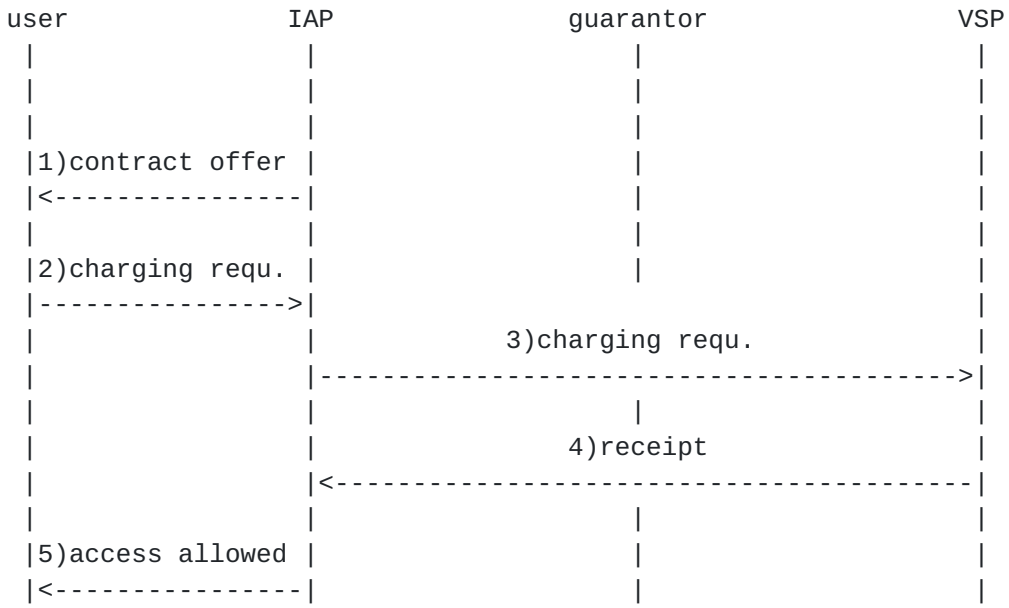

Figure 6: Messaging for charging with cost negotiation

The contract offer contains a timestamp element, the identifier of the IAP and its price plan. The timestamp is used for avoiding replay attacks and for the verification of the freshness of the receipt issued by the VSP. The IAP identifier allows the VSP to identify the IAP to which sending the receipt. The cost plan describes the cost of the service. XML examples of a contract offer, a charging request and a receipt are provided in Figure 7 (see

Figure 13 for the XML schema of the contract offer), Figure 8 and Figure 9, respectively.

```
<?xml version="1.0" encoding="UTF-8"?>
<ContractOffer xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
        <IAP_ID>IAP.example.com</IAP_ID>
        <timestamp>2007-02-28T23:20:50.52Z</timestamp>
        <cost costPerUnitTime="6" initialCost="250" timeUnitSize="6000">
                <currency currency="USD" currencyDivisor="1000" namespace="ISO.
4217"/>
        </cost>
</ContractOffer>
```

Figure 7: Contract offer XML example

```
<?xml version="1.0" encoding="UTF-8"?>
<ChargingAuthorization xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
        <ContractOffer>
                <IAP_ID>IAP.example.com</IAP_ID>
                <timestamp>2007-02-28T23:20:50.52Z</timestamp>
                <cost costPerUnitTime="6" initialCost="250"
timeUnitSize="6000">
                        <currency currency="USD" currencyDivisor="1000"
namespace="ISO.4217"/>
                </cost>
        </ContractOffer>
        <UserSignature>.....</UserSignature>
</ChargingAuthorization>
```

Figure 8: Charging request XML example

```
<Receipt xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
        <ContractOffer>
                <IAP_ID>IAP.example.com</IAP_ID>
                <timestamp>2007-02-28T23:20:50.52Z</timestamp>
                <cost costPerUnitTime="6" initialCost="250"
timeUnitSize="6000">
                        <currency currency="USD" currencyDivisor="1000"
namespace="ISO.4217"/>
                </cost>
        </ContractOffer>
        <VSPSignature>.....</VSPSignature>
        <VSPCertificate>....</VSPCertificate>
</Receipt>
```

Figure 9: Receipt XML example

One of the main advantage of the model introduced above is that the charging is authorized by the user directly.

6. Security Considerations

The security properties of the proposed protocol depend on the security features of HTTP and SAML. VSPs use the HTTP authentication features and authenticate users before accepting their token requests to avoid that malicious subjects impersonating them obtain assertions. For the same reason the guarantor authenticates VSPs before accepting their token building requests. VSPs have to authenticate the guarantor before sending token building requests because these contain private information about users. VSPs and the guarantor may use the SAML authentication features described in [saml-sec-consider-2.0-os]for their mutual authentication. Moreover they must guarantee confidential transport of the assertion encrypting the SOAP messages as specified for the SOAP SAML binding in [saml-sec-consider-2.0-os]. This avoids that users eavesdropping the conversation can make copies of the assertion. For the same reason VSPs encrypt the bodies of the 200 OK responses carrying roaming assertions. The parties involved may use the Transport Layer Security (TLS) protocol [RFC2246] which allows building secure communication channels between them.

7. XML schemas

In this section we provide the XML schemas of the elements and types defined in this draft to support the SAML roaming profile.

7.1. XML schema of the Condition element

As introduced in Section 4.1, the SAML Condition element is used for describing the token lifetime and the user profile. For the description of the token lifetime we use the NotBefore and NotOnOrAfter attributes of this element defined in [saml-core-2.0-os]. For the description of the user profile, we define the condition_profileType. It extends the ConditionAbstractType of the Condition element defined in [saml-core-2.0-os] and adds the UserProfile element to it. We use the quality of service class for describing the user profile. The quality of service class is described using the values Gold, Bronze, Silver. Figure 10 shows the xml schema of the condition_profileType.

```
<?xml version="1.0"?>
<!-- definition of the condition_profileType -->
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 targetNamespace="http://www.tti.unipa.it/~silvana/tokencondition"
 xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion"
 elementFormDefault="qualified">
 <xsd:import namespace="urn:oasis:names:tc:SAML:2.0:assertion"
  schemaLocation="saml-schema-assertion-2.0.xsd"/>
<xsd:complexType name="condition_profileType">
  <xsd:complexContent>
   <!-- extension of the ConditionAbstractType with the
    UserProfile element which contains the UserClass element -->
   <xsd:extension base="saml:ConditionAbstractType">
    <xsd:sequence>
     <xsd:element name="UserProfile">
      <xsd:complexType>
       <xsd:sequence>
        <xsd:element name="UserClass">
         <xsd:simpleType>
          <xsd:restriction base="xsd:string">
           <xsd:enumeration value="Gold"/>
           <xsd:enumeration value="Bronze"/>
           <xsd:enumeration value="Silver"/>
          </xsd:restriction>
         </xsd:simpleType>
        </xsd:element>
       </xsd:sequence>
      </xsd:complexType>
     </xsd:element>
    </xsd:sequence>
   </xsd:extension>
  </xsd:complexContent>
 </xsd:complexType>
</xsd:schema>
```

Figure 10: XML schema of the condition_profileType

7.2. XML schema of the token building request

The token building request is described extending the SAML type RequestAstractType [saml-core-2.0-os] of a generic SAML request. As described in the XML schema of Figure 11, the extension that we propose adds the Subject and the Conditions element to the ones of the RequestAbstractType. The Subject element is used for including the SIP URI in the request. The Conditions element allows to introduce the token lifetime and the user profile in the token building request.

```
<?xml version="1.0"?>
<!-- definition of the token_building_request element that describes
the token building request -->
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 targetNamespace="http://www.tti.unipa.it/~silvana/requesttype"
 xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion"
 xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"
 xmlns:cond="http://www.tti.unipa.it/~silvana/tokencondition"
 xmlns="http://www.tti.unipa.it/~silvana/requesttype"
 elementFormDefault="qualified">
 <xsd:import namespace="urn:oasis:names:tc:SAML:2.0:assertion"
  schemaLocation="saml-schema-assertion-2.0.xsd"/>
 <xsd:import namespace="urn:oasis:names:tc:SAML:2.0:protocol"
  schemaLocation="saml-schema-protocol-2.0.xsd"/>
 <xsd:import
  namespace="http://www.tti.unipa.it/~silvana/tokencondition"
  schemaLocation="condition_profileType.xsd"/>
 <xsd:element
 name="token_building_request" type="token_buildingReqType"/>
  <xsd:complexType name="token_buildingReqType">
   <xsd:complexContent>
   <!-- Extension of the SAML RequestAbstractType with the
    Subject element and the Conditions element -->
    <xsd:extension base="samlp:RequestAbstractType">
     <xsd:sequence>
      <xsd:element ref="saml:Subject"/>
      <xsd:element ref="saml:Conditions"/>
     </xsd:sequence>
    </xsd:extension>
   </xsd:complexContent>
  </xsd:complexType>
</xsd:schema>
```

Figure 11: XML schema of the token building request

7.3. XML schema of the Statement element

Figure 12 shows the XML schema of the type of the Statement element of the roaming assertion. It is obtained as extension of the StatementAbstractType defined in [saml-core-2.0-os]. It allows to include in the Statement element of the roaming assertion the domain name of the VSP, and the policy_info element. This carries information about the user profile and has the type defined in Section 7.1.

```
<?xml version="1.0"?>
<!-- Definition of the roaming_statementType  -->
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 targetNamespace="http://www.tti.unipa.it/~silvana/"
 elementFormDefault="qualified"
 xmlns="tokenaccess"
 xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion"
 xmlns:pre="http://www.tti.unipa.it/~silvana/tokencondition">
 <xsd:import namespace="urn:oasis:names:tc:SAML:2.0:assertion"
  schemaLocation="saml-schema-assertion-2.0.xsd"/>
 <xsd:import namespace="http://www.tti.unipa.it/~silvana/tokencondition"
  schemaLocation="condition_profileType.xsd"/>
 <xsd:complexType name="roaming_statementType">
  <xsd:complexContent>
   <xsd:extension base="saml:StatementAbstractType">
    <xsd:sequence>
     <!-- Definition of the ServiceProviderID element having
      saml:NameIDType type  -->
     <xsd:element name="ServiceProviderID" type="saml:NameIDType"/>
     <!-- Definition of the policy_info element having
      condition_profileType type -->
     <xsd:element name="policy_info" type="pre:condition_profileType"/>
    </xsd:sequence>
   </xsd:extension>
  </xsd:complexContent>
 </xsd:complexType>
</xsd:schema>
```

Figure 12: XML schema of the roaming Statement element type

7.4. XML schema of the contract offer

Figure 13 describes the components of the contract offer introduced in Section 5.2. This offer has been defined following the model introduced in [draft-jennings-sipping-pay-05]. The elements of the offer are the IAP_ID for the description of the IAP identifier, the timestamp containing the issuing time of the offer, and the cost element with the description of the cost plan of the IAP. The description of the cost element is derived from the description of the homonymous element introduced in [draft-jennings-sipping-pay-05] (see it for additional details).

```
<?xml version="1.0" encoding="UTF-8"?>
<schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
 <element name="ContractOffer">
        <complexType>
          <sequence>
                <xs:element name="IAP_ID"></xs:element>
                <xs:element name="timestamp"></xs:element>
                <element name="cost">
                <complexType>
                 <attribute default="0" name="initialCost" type="unsignedLong"
use="optional"/>
                 <attribute default="0" name="costPerUnitTime"
type="unsignedLong" use="optional"/>
                 <attribute default="0" name="timeUnitSize" type="unsignedLong"
use="optional"/>
                 <attribute default="0" name="costPerUnitData"
type="unsignedLong" use="optional"/>
                 <attribute default="0" name="dataUnitSize" type="unsignedLong"
use="optional"/>
            </complexType>
                </element>
                <element name="currency">
                <complexType>
                 <attribute name="namespace" type="string" use="required"/>
                 <attribute name="currency" type="string" use="required"/>
                 <attribute name="currencyDivisor" type="unsignedLong"
use="required"/>
                </complexType>
                </element>
          </sequence>
        </complexType>
 </element>
</schema>
```

Figure 13: XML schema of the contract offer

8. Acknowledgements

The authors thank Hannes Tschofenig for his comments.

9. References

9.1. Normative References

[OSP] European Telecommunications Standards Institute, "Telecommunications and Internet Protocol Harmonization Over Networks (TIPHON) Release 4; Open Settlement Protocol (OSP) for Inter-Domain pricing, authorization and usage exchange", ETSI Technical Specification ETSI TS 101 321 V4.1.1 (2003-11), 2003.

[RFC2246] Dierks, T. and C. C. Allen, "The TLS Protocol Version 1.0", RFC 2246, January 1999.

[RFC2616] Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P., and T. Berners-Lee, "Hypertext Transfer Protocol HTTP 1/1", RFC 2616, June 1999.

[RFC3261] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., and J. Peterson, "Session Initiation Protocol", RFC 3261, June 2002.

[RFC3748] Fielding, R., Aboba, B., Blunk, L., Vollbrecht, J., Carlson, J., and H. Levkowetz, "Extensible Authentication Protocol (EAP)", RFC 3748, June 2004.

[RFC4187] J. Arkko, J. and H. H. Haverinen, "Extensible Authentication Protocol Method for 3rd Generation Authentication and Key Agreement (EAP-AKA)", RFC 4187, January 2006.

[XML] World Wide Web Consortium, "Extensible Markup Language (XML) 1.0", W3C XML, February 1998.

[a802.1x] IEEE, "Port-based network access control", IEEE standard 802.1x, 2001.

[applicationsamlassertionxml] OASIS Security Services Technical Committee, "application/ samlassertion+xml MIME Media Type Registration", IANA MIME Media Types application/samlassertion+xml, December 2004.

[saml-binding-2.0-os] Cantor, S., Hirsch, F., Philpott, R., and E. Maler, "Binding for the OASIS Security Assertion Markup Language (SAML) V2.0", OASIS Standard saml-binding-2.0-os, March 2005.

[saml-core-2.0-os]
Cantor, S., Kemp, J., Philpott, R., and E. Maler, "Assertions and Protocol for the OASIS Security Assertion Markup Language (SAML) V2.0", OASIS Standard saml-core-2.0-os, March 2005.

[saml-glossary-2.0-os]
Hodges, J., Philpott, R., and E. Maler, "Glossary for Security Assertion Markup Language (SAML) V2.0", OASIS Standard saml-glossary-2.0-os, March 2005.

[saml-profiles-2.0-os]
Hughes, J., Cantor, S., Hodges, J., Hirsch, F., Mishra, P., Philpott, R., and E. Maler, "Profiles for the OASIS Security Assertion Markup Language (SAML) V2.0", OASIS Standard saml-profiles-2.0-os, March 2005.

[saml-sec-consider-2.0-os]
Hirsch, F., Philpott, R., and E. Maler, "Security and Privacy Considerations for the OASIS Security Markup Language (SAML) V2.0", OASIS saml-sec-consider-2.0-os, March 2005.

[xmldsig] World Wide Web Consortium, "XML-Signature Syntax and Processing", W3C Recommendation xmldsig, October 2000.

9.2. Informative References

[draft-funk-eap-ttls-v1-01]
Funk, P. and S. Blake-Wilson, "EAP Tunneled TLS Authentication Protocol", draft draft-funk-eap-ttls-v1-01, March 2006.

[draft-ietf-sip-serverfeatures-02]
Rosenberg, J. and H. Schulzrinne, "The SIP Supported Header", draft draft-ietf-sip-serverfeatures-02, September 2000.

[draft-jennings-sipping-pay-05]
Jennings, C., Fischl, J., Tschofenig, H., and G. Jun, "Payment for Services in Session Initiation Protocol (SIP)", draft draft-jennings-sipping-pay-05, October 2006.

Authors' Addresses

Silvana Greco Polito
Columbia University
Viale delle Scienze
Palermo, Sicily 90100
Italy

Email: silvana.greco@tti.unipa.it, silvana@cs.columbia.edu

Henning Schulzrinne
Columbia University
450 Computer Science Building
New York, NY 10027
US

Email: hgs@cs.columbia.edu

Full Copyright Statement

Intellectual Property

Acknowledgment

Funding for the RFC Editor function is provided by the IETF Administrative Support Activity (IASA).