3GPP TSG SA WG3 Security — S3#29 **S3-030359**
15 - 18 July 2003
San Francisco, USA

Agenda Item: 7.18 (Presence)

Source: Nokia

Title: Rationale of Presence service and Authentication Proxy functionality in detail

Document for: Discussion/Decision

1. Introduction

For Presence and other SIP based services, S3 #28 meeting has discussed the authentication and security of Mt interface which is renamed as Ut interface by S2. This paper is intended to give a comprehensive description of SIP based services so as to depict the nature and the environment of SIP based service. Taking Presence as example, section 2 explains the service working rationale, and how does service utilize the data manipulation that is done previously over Ut interface, next the procedure on how to put or post such a list is given in detail in section 3.

Further this paper presents the design preference based on the description of the service. That is described in section 4 with comparison of the similarity, and the difference between Nokia& Ericsson's proposals and Siemens proposal. Stage 2 and some level stage 3 details are given. Section 5 is the consideration of standardization and schedule. It is shown the objections are minimized based on detail in section 4.

Finally we propose to use Authentication proxy as an optimized solution, which is in section 6.

2. Usage of Resource list through IMS

In Presence service scenario, the UE needs to setup HTTP connection to the home Resource List server over Ut interface, so as to manage the list and authorization control. The list is posted to the server under a URL that is specific for that list. Figure 1 depicts the service architecture with location of the URL.

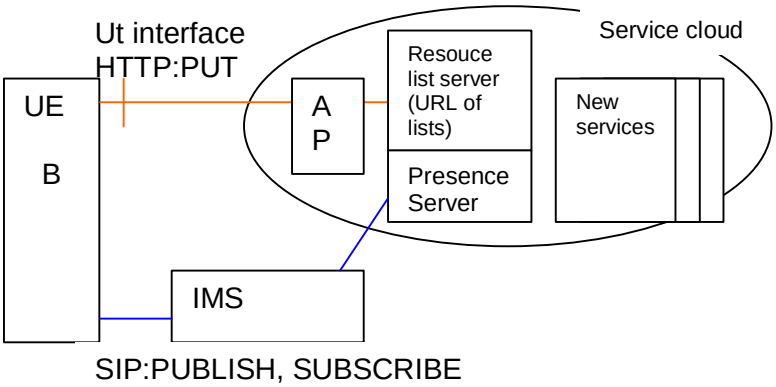


Figure 1: SIP based service through separate interfaces

The UE may act as both Presentity and Watcher. When acting as a presentity, the UE B will publish her own Presence Information (PI) to the Presence Server. This is specified in [24.841] by using the PUBLISH method, and the exact PI format is specified by [draft-ietf-impp-cpim-pidf] and reused in [24.841].

Next, UE B may subscribe to his own list under the URL, so as to see which watcher on his/her own list has subscribed his PI [draft-ietf-simple-winfo-package].

```
SUBSCRIBE sip:B@example.com SIP/2.0
  Via: SIP/2.0/UDP pc34.example.com;branch=z9hG4bKnashds7
  From: sip:B@example.com;tag=123s8a
  To: sip:B@example.com
  Call-ID: 9987@pc34.example.com
  Max-Forwards: 70
  CSeq: 9887 SUBSCRIBE
  Contact: sip:B@pc34.example.com
  Event: presence.winfo
```

Here the event header = Presence.winfo to indicate that this subscribe's his own watcher information. And there is B's own URI in the Request URI of the SUBSCRIBE method. The identity and security are controlled by IMS. Later on, when a watcher subscribes his presence information, a NOTIFY SIP message will be sent from Presence server to UE B via IMS.

Or when the UE acting as a watcher, it subscribers to the Presence information of his collection of presentities using SUBSCRIBE method. This format is specified in [draft-ietf-simple-event-list] as below:

```
SUBSCRIBE sip:adam-buddies@pres.example.com SIP/2.0
  Via: SIP/2.0/TCP terminal.example.com;branch=z9hG4bKwYb6QREiCL
  Max-Forwards: 70
  To:
  From: ;tag=ie4hbb8t
  Call-ID: cdB34qLToC@terminal.example.com
  CSeq: 322723822 SUBSCRIBE
  Contact:
  Event: presence
  Expires: 7200
  Supported: eventlist
  Accept: application/cpim-pidf+xml
  Accept: application/rlmi+xml
  Accept: multipart/related
  Accept: multipart/signed
  Accept: multipart/encrypted
  Content-Length: 0
```

Note here the Event header= Presence, and the resource list is a SIP URI in a server, that maybe part of a Presence server.

Conclusion: As we see in both cases, the resource list must be available prior to SUBSCRIBE sent by the UE. In other words, Ut interface data manipulation may take place before IMS registration in real life.

3. UE PUT/POST Presence information (PI) in detail

This section describes how a UE uses put/post resource list into a server, taking presence as example. Figure 2 depicts the stage 3 message flow chart with authentication handled by Authentication Proxy (AP). Currently SIMPLE working group has defined the XACAP (based on ACAP and XML) encoded body to encode the list [draft-ietf-simple-xcap].

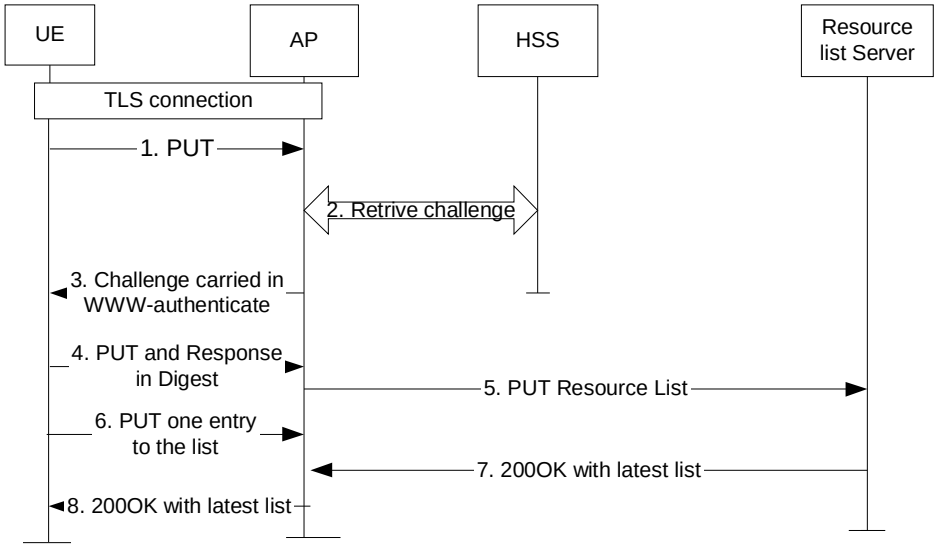


Figure 2: Stage 3 message flow chart of post a list, with authentication handled by Authentication Proxy

1. First, a user user1_public1 creates a new Presence-list, initially with no users in it:

```
PUT http://Presence.example.com/services/Presence-lists/users/user1_public1/fr.xml
HTTP/1.1
Host: Presence.example.com
Authorization: username="user1_private@example.com"
Content-Type:application/Presence-lists+xml

<?xml version="1.0" encoding="UTF-8"?>
  <Presence-lists xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <list name="friends" uri="sip:friends@example.com" subscribable="true">
    </list>
  </Presence-lists>
```

2. AP shall fetch the challenge from HSS, check that private identity is associated with the present public identity.

3. The AP shall authenticate the user based on AKA authentication. (Note, here are two ways to proceed: based on on-line bootstrapped secret in Digest response or based on enhanced AKAv1. Example below shows example of AKAv2.)

```
HTTP/1.1 401 Unauthorized
Host: Presence.example.com
WWW-Authenticate: Digest
```

realm="Presence.example.com",
nonce=base64(RAND + AUTN + server specific data),
qop=" auth-int" ,
opaque="5ccc069c403ebaf9f0171e9517f40e41",
algorithm=AKAv2-MD5

4. The UE responds the challenge

```
PUT http://Presence.example.com/services/Presence-lists/users/user1_public1/fr.xml
HTTP/1.1
Host: Presence.example.com
Authorization: Digest
        username="user1_private@example.com",
        realm="Presence.example.com",
        nonce=base64(RAND + AUTN + server specific data),
        qop=" auth-int" ,
        opaque="5ccc069c403ebaf9f0171e9517f40e41",
        algorithm=AKAv2-MD5,
        uri=" /?username=user1_public1",
        response="6629fae49393a05397450978507c4ef1"
  Content-Type:application/Presence-lists+xml

<?xml version="1.0" encoding="UTF-8"?>
  <Presence-lists xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
   <list name="friends" uri="sip:friends@example.com" subscribable="true">
   </list>
  </Presence-lists>
```

5. AP removes the Authorization and forwards the rest of the message to the Presence server. Note AP does not to understand the xml body. In case there is problem in the xml contents, the Resource List server shall return an HTTP error message.

6. UE with user1_public1adds an entry to the list:

```
PUT http://Presence.example.com/services/Presence-lists/users/bill/fr.xml?
  Presence-lists/list[@name="friends"] HTTP/1.1
  Content-Type:text/plain
  Content-Length: nnnn

 <entry name="Bill" uri="sip:bill@example.com">
   <display-name>Bill Doe</display-name>
  </entry>
```

7. Maybe the Presence server should give a response to UE, to indicate the current list by

```
HTTP/1.1 200 OK
Content-Type:text/plain
 <entry name="Bill" uri="sip:bill@example.com">
   <display-name>Bill Doe</display-name>
 </entry>
```

8. AP shall forward the message 6 to the UE. And the User interface indicates the user that the list is loaded properly. When the UE would like manage data on other server, such as conference list, the UE can reuse the connection, with another public id if required. AP can verify the association with IMPI, this is discussed in detail in section 4.1.

4. AP functionality in detail

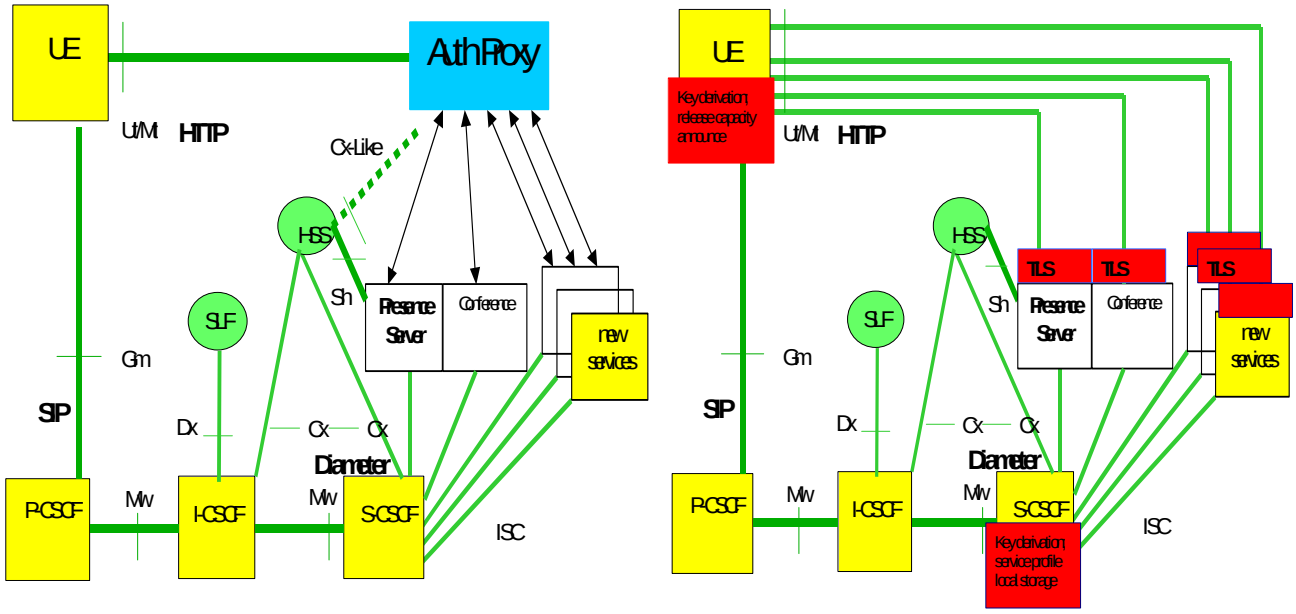
Ut interface is established for user to manipulate any kind of SIP application specific data on an Application Server, such as presetity list, watcher list, chat groups, conference attendee list et cetera. Re-use the interface would request a good extension model. This is also the intension in IETF standardization work [draft-ietf-simple-xcap].

Basically the AP has two functions:

a. To reuse the TLS connections to be only one, that improves the reuse rate. AP is the termination point of TLS connection and the HTTP session will terminate in each application server.

b. To authenticate UE on behalf of all application servers.

Figure 3 depictures the contrast of different proposals. In left side is Nokia&Ericsson proposal. The Authentication Proxy handles the security connection as a dedicated server. Traffic between the AP and the application servers are inside the same security domain, protection is not necessary.



Nokia&Ericsson proposal (S3-030256) Siemens proposal (S3-030223, 224)

Figure 3: the comparison and contrast of two proposals

The functionality a) has benefits that we believe important:
1. It can save network resources and the investment on purchase. If AP is absent, the functionality a) has to be duplicated into many elements, i.e. every server, as the red elements in Figure 3. It is more

optimized to invest only once in AP. AP is a logical entity; physically it could be co-exist with other application.

2. It optimizes the user interaction. Full TLS Connection setup is quite complex and time consuming, requiring at least two round trips and expensive public-key cryptographic operations, e.g. the 1024 modular exponentiation required by RSA. Thus the cache usage is specified in RFC 2246 to re-use pre-master key.

Now due to the usage of Ut interface, the user turns to manipulate own data either at a time for many integrated SIP services, or periodically to each server. In both cases, pre-master key agreed in last session is stored in AP, thus it can be shared and the session can be resumed, thus reuse the earlier RSA operation result. This largely reduces session establishment load as shown in many research works, for example [Coarfa2002]. In a distributed TLS model as shown in right side, the re-use of TLS can only be applied when a user contacts one server in a short period.

[Coarfa2002] also shows that CPU costs have more impact on TLS server throughput than data exchange. Due to the reusability, the system performance can largely improved by using a dedicated high performance server.

Regarding to function b), whether it should be combined with IMS registration, we have investigated the rationale of service on several aspects:

1. Support of IMS for the service:
   S2 stage 2 specification [23.141] of Presence has also specified how does a Presentity updates own presence information without support of IMS in Annex A.2.3.1. It is to us clear that the service does not only depend on IMS registration.

2. CN1 opinion:
   SA #28 meeting in Berlin sent a LS to CN1 asking their opinion to feasibility of Siemens proposal. CN1 provided their view in S3-030325. Though stating the solution looks feasible, it also points out the drawbacks if relying on authentication to IMS registration:

- it causes backward compatibility problems;
- It puts additional processing load on the S-CSCF which is multiplied by the number of application servers involved.

Explicitly, CN1 points out that registration to IMS should be used exclusively for authentication of the UE to the IMS. In other words, Ut interface is independent from IMS registration and authentication.

3. 3rd party services:
   To expand the service scope the IMS enables 3rd party application. Other than service delivery, security is another consideration. When AP is present, it is fully trusted by operator, thus the authentication, and corresponding charging information are guaranteed, and can be verified with CDR provided by external servers.

If we dismiss AP, the function a) has to be duplicated into many elements, i.e. every server; then function b) would require adding the key derivation function into S-CSCF, service profile storage in S-CSCF, and last key storage and release capability announcement into UE, it is basically the rationale of Siemens proposal (key lifetime management and interaction with IMS registration needs further consideration).

Conclusion: We believe the functionalities of AP should be present, and the central processing is more optimized design in cost and more efficient in system performance.

4.1 AP and HSS interface (Cx-Like)

The two functions listed in present paper p.6, requests information from HSS. One is authentication against the IMPI, and one is retrieval of all IMPUs belong to the user to access to all servers regardless of IMS registration status. Both of the functions are available through Cx interface.

The authentication commands [29.229], Multimedia-Auth-Request (MAR) and Multimedia-Auth-Answer (MAA), are necessary. Either the specific application id or adding a new AVP indicating Authentication Proxy functions required.

In current Cx specification 29.228, the IMPUs of the UE are given in SAR/SAA message (see table 6.2.2.1: User Profile Update request). Since AP does not care the other detail of service profile itself, a simple way to proceed is to add a new AVP containing all IMPUs of the UE regardless of her service profiles. So the change to Cx specification is minimized. As per request from AP, the HSS shall initiate only MAA command. Other commands are not needed such as Registration-Termination-Request/Answer, Push-Profile-Request/Answer because 'SIP registration state' for S-CSCF is not applicable to AP.

4.1.1 The feasibility commented by CN4

S3 has sent a LS to CN4 regarding to this topic. CN4 LS [S3-030337] suggested to S3 to consider security requirements for inter domain usage of Cx protocol as well as synchronization problem of authentication vectors.

The inter domain usage is not a problem for Cx-Like interface, since UE always contacts to home domain where AP is located. So the HSS-AP interface is in same operator's network domain. For the re-synchronization problem, CN4 informs that HSS needs to acquire the source of the request in two ways:

- Adding an AVP to the Cx protocol or as a vendor specific AVP in Diameter Base Protocol
- Each application having separate protocols based on Cx interface protocol

Though listed pros and cons, CN4 stated both ways are feasible: "If no major impacts are anticipated to the Cx functionality a new Diameter application based on the Cx interface can be specified or the requested functionality can be added to the existing Cx application in Rel 6 timescale."

Recommendation has given in 33.102 regarding to Sequence number management in different domain. Since in the present design the connection is re-used for all servers, one fetch per time for Cx-like interface seems a good approach to mitigate this problem.

4.2 Sh interface

Sh interface is for Application server to retrieve user service related data. Current specification does not contain such functionality. The Sh interface contains commands to retrieve different user data from HSS and to maintain service specific data (transparent to HSS) in HSS. In addition the Sh interface contains a subscribe-notify mechanism to inform the AS on changes in user data in HSS.

5. Feasibility in standardization

In the present contribution we have present the detail of AP functionality. The other concerns are discussed, and counter-solutions are recommended.

Nokia sees both approaches feasible: one is by using bootstrapping function, the other one is to use an enhanced AKAv1 to mitigate tunnelling problem. Approach one depends on BSF specification, it also introduces many benefits as pointed out in S3#28 meeting:

- Specification work is in hand of S3 largely, so it is possible to finish before R6 deadline;
- Sequence number management is easier;

- BSF is meant for usage of service such as the topic in question

On the other hand, approach two's drawbacks on Cx-Like interface can be largely mitigated as discussed in section 4.1. The interface can be handled as indicated by CN4 LS, on how to solve the requestor. Consumption of AVs and SQN number management can both be mitigated by using fetching one AV and re-use the connection for all services.

If the proposal is agreed, the specifications may be affected are S3's Presence security, CN1 Presence stage 3, and CN4 Cx specification.

S2 stage 2 for IMS and for SIP based service specifications are not affected.

6. Proposal

Based on detail of AP functionality, we propose to use Authentication proxy as an optimized solution. It is shown that objections are minimized, counter-solutions are recommended.

7. REFERENCE

[23.141] Presence Service: Architecture and Functional Description, V6.1.0, 3GPP. December, 2002.

[24.841] Presence service based on Session Initiation Protocol (SIP), v1.0.0, 3GPP. May 2003.

[29.229] Cx and Dx interfaces based on the Diameter protocol, V5.4.0, 3GPP. June, 2003.

[draft-ietf-impp-cpim-pidf] Presence Information Data Format (PIDF), draft-ietf-impp-cpim-pidf-08.txt, May 2003.

[draft-ietf-simple-xcap] The Extensible Markup Language (XML) Configuration Access Protocol (XCAP), draft-ietf-simple-xcap-00, June 23, 2003.

[draft-ietf-simple-winfo-package] A Watcher Information Event Template-Package for the Session Initiation Protocol (SIP), draft-ietf-simple-winfo-package-05.txt. January 31, 2003.

[draft-ietf-simple-event-list] A Session Initiation Protocol (SIP) Event Notification Extension for Resource Lists, draft-ietf-simple-event-list-04. June 13, 2003.

[S3-030337] LS on adapting Cx interface protocols for security purposes, CN4 wg.

[Coarfa2002] Performance Analysis of TLS Web Servers, Network and Distributed Systems Security Symposium '02, San Diego, California, February 2002.