3GPP TSG SA WG3 Security — S3#31 **S3-030731**
18 – 21 November 2003
Munich, Germany

Source:	**Nokia**
Title:	**Proxy and various HTTP services**
Document for:	**Discussion and Decision**
Agenda Item:	**GAA**

1. AP USING TLS WITH SHARED SECRETS

This section describes how GAA can use AP using shared key TLS. In this scenario, the BSF and AP-NAF can be either co-located or separate network elements. Figure 1 depicts the latter case.

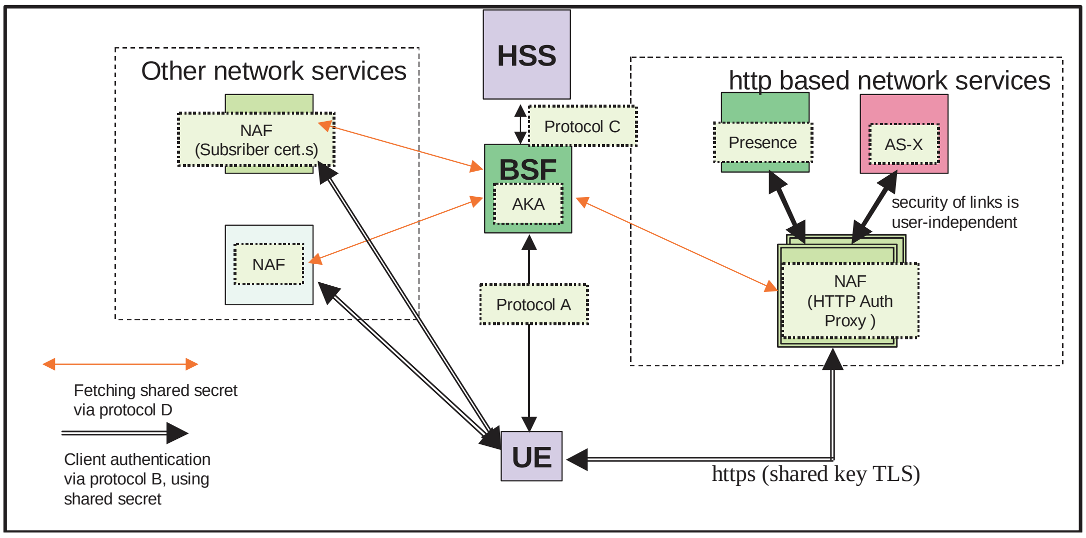
Other network services
HSS
http based network services
NAF (Subsriber cert.s)
Protocol C
BSF
AKA
Presence
AS-X
security of links is user-independent
NAF
Protocol A
NAF (HTTP Auth Proxy)
Fetching shared secret via protocol D
Client authentication via protocol B, using shared secret
UE
https (shared key TLS)

Figure 1: AP-NAF using shared key TLS.

At the left side of Figure 1, the UE contacts NAF directly; while at the right side, the UE contacts the Presence and other IMS based SIP services via Authentication Proxy. It was agreed in SA3 meeting that, for one particular service, there should be only one network configuration as solution. This is benefitial, not only for UE to reach service in clear manner, but also it saves the investment to the service. Therefore the principle is reasonable and should be sustained.

2. CONFIGURATION OF PROXY TO A TERMINAL

There is no problem to use forward proxy. It seems like http stacks do allow the calling application (service in terminal) to specify a proxy on a per-server or even a per-URL basis. So it should be pretty straight forward for the Presence client (or a generic Ut client) to maintain. On the other hand, when a new service does not require Proxy configuration in the network, the new application over the HTTP stack does not need to be the same with Presence service. The terminal just needs to maintain the autoconfiguration file; the semantics would be like:

```
Function FindProxyForURL (url, host)
{ if (isHostName (host) ||
!dnsDomainIs (host, “Presence.operator.com”)||
!dnsDomainIs (host, “OtherServer.operator.com”))
return “PROXY a.b.c.d”
else
return “DIRECT”
}
```

Today’s browsers do support this type of autoconfiguration.

In contrast, a reverse proxy handling security function on behalf of a group of SIP servers may have problems mainly in configuration for service in UE and in network side.

1. The UE does not see Proxy in the middle, thus UE may not understand that the established TLS session can be re-used for accessing to another Application server.

2. All SIP servers sharing the same DNS name will make ambuguility for S-CSCF to reach the proper server.

3. A higher-level DNS name stored in a certificate would improve the server identity problem, but we doubt whether any external CA would like to issue many wildcard certificate for 3GPP operators. Or 3GPP operator must run CA function themselves, to generate such self-signed certificate.

It is worth of noting, if the shared-key TLS is used, it does not matter proxy type, whether forward or reverse one. Basically the (reverse) proxy will acquire the DNS name of all servers behind, and ‘hijack’ on the connections intended to servers directly.

Conclusion: We see that a forward proxy would reflect the network topology clearly to the terminal configuration. Then in case shared-key TLS is chosen, the type of proxy really does not matter.

3. AP AND AS INTERFACE

S3-030540 suggests AP and AS to insert cookies, we feel there are few drawbacks on standardisation and implementation aspects:

1. Current HTTP implementations have own API defined already (for setting and fetching/comparing a cookie), and having a "special" cookie value would mean an extra processing step when receiving the cookie. This might also interfere with normal AS session handling using HTTP cookies.

2. It is requiring ASs to understand a particular token syntax for the cookie, so I don't see there being much benefit compared to a new (X-something) header, or indeed having the proxy do URL-rewriting, etc. How about a new 3GPP specific header? E.g., X-HTTP-Asserted-Identity? This would be totally transparent to the terminal.

3. The sentence seems to suggest that only one IMPU is used: “AS can assume that the AP has authenticated the client with this identity.” If so, it does not fulfil the requirement that AS would be contacted with any of the IMS IMPUs.

4. Also, an AS that doesn't care about identity could then ignore this header and do as it pleases. An out-of-the box AS could not do this if we used cookies, since it would have to know at least the "special" cookie in order to ignore it (and not think that a cookie is invalid etc.)

Another contribution from Nokia S3-030555 proposed an alternative, where AP can VERIFY the HTTP message without further modification. AP contacts BSF based on TID or session identifier from UE, for the UE identities as well as the session secret Ks_NAF. Here the procedure can be enhanced, so that only the public identities are fetched over Zn interface. Thus the UE does not need to insert own private identity into the HTTP message. This is because the Bootstrapping procedure already authenticates the UE, it is sufficient for it proving its possession of keys over Ua interface. The AP only needs to check that the public IDs populated in the URL is allowed (see bold part in below), and drop it if not.

```
PUT http://Presence.example.com/services/Presence-lists/users/user1_public1/fr.xml HTTP/1.1
Host: Presence.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>
```

Note that since mutual authentication is guaranteed by shared key TLS connection, and additional AKA-based authentication in application is not needed any more, which simplifies the procedure significantly. Once the TLS connection is established, the UE can right away send user data in HTTP message.

Note two, in case of TLS 1.0 for Ua interface, the digest would be required for AP to authentication the client. Yet it is independent of the choice of the two approaches.

4. CONCLUSION AND PROPOSAL

This meeting is proposed to endorse the S3-030555 solution as working assumption.