3GPP TSG SA WG3 Security — S3#19 **S3-010317**

3 - 6 July, 2001

Newbury, UK

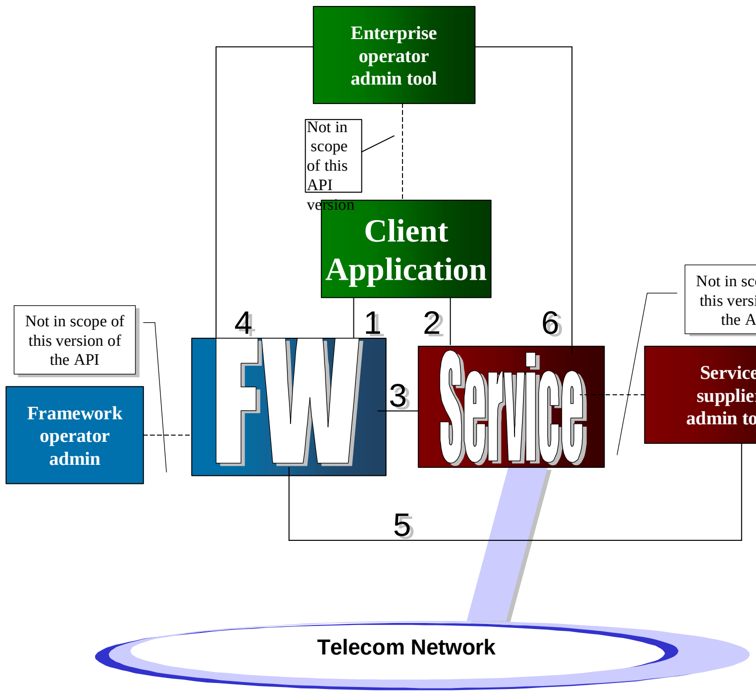
Source: **BT**
Title: **Review of OSA Security - some issues for SA3 to consider**
Document for: **Discussion and Decision**
Agenda Item: **7.9**

Background

The OSA-specifications define an architecture that enables service application developers to make use of network functionality through an open standardised interface, i.e. the OSA APIs. The network functionality is describes as Service Capability Features or Services [note]. The OSA Framework is a general component in support of Services (Service Capabilities) and Applications. The concepts and the functional architecture for the Open Service Access (OSA) are described by 3GPP TS 23.127 [3]. The requirements for OSA are defined in 3GPP TS 22.127 [2].
The OSA API is split into three types of interface classes, Service and Framework.

- Interface classes between the Applications and the Framework, that provide applications with basic mechanisms (e.g. Authentication) that enable them to make use of the service capabilities in the network.
- Interface classes between Applications and Service Capability Features (SCF), which are individual services that may be required by the client to enable the running of third party applications over the interface e.g. Messaging type service.
- Interface classes between the Framework and the Service Capability Features, that provide the mechanisms necessary for multi-vendorship.

These interfaces represent interfaces 1, 2 and 3 of the Figure below. The other interfaces are not yet part of the scope of the work.

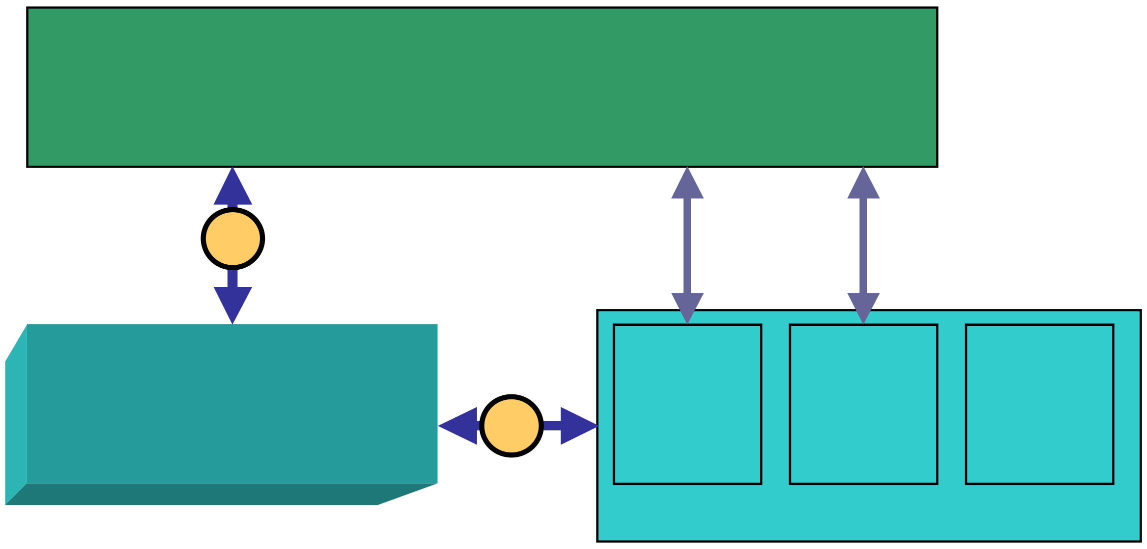

Enterprise operator admin tool
Not in scope of this API version
Client Application
Not in scope of this version of the API
Not in scope of this version of the API
4
1
2
6
3
5
FW
Service
Framework operator admin
Service supplier admin tool
Telecom Network

Within the OSA concept a set of Service Capability Features has been specified. The OSA documentation is structured in parts. The first Part (this document) contains an overview, the second

Part contains common data definitions, the third Part the Framework interfaces. The rest of the Parts contain the description of the SCF's.
NOTE:
The terms 'Service' and 'Service Capability Feature' are used as alternatives for the same concept in this specification. In the OSA API itself the Service Capability Features as identified in the 3GPP requirements and architecture are reflected as 'service', in terms like serviceFactory, serviceDiscovery.

Framework API

The Framework API contains interfaces between the Application Server and the Framework, and between Network Service Capability Server (SCS) and the Framework (these interfaces are represented by the yellow circles in the diagram below). The description of the Framework in this document separates the interfaces into these two distinct sets: Framework to Application interfaces and Framework to Service interfaces.



Some of the mechanisms are applied only once (e.g. establishment of service agreement), others are applied each time a user subscription is made to an application (e.g. enabling the call attempt event for a new user).
Basic mechanisms between Application and Framework:

- **Authentication:** Once an off-line service agreement exists, the application can access the authentication interface. The authentication model of OSA is a peer-to-peer model. The application must authenticate the framework and vice versa. The application must be authenticated before it is allowed to use any other OSA interface.
- **Authorisation:** Authorisation is distinguished from authentication in that authorisation is the action of determining what a previously authenticated application is allowed to do. Authentication must precede authorisation. Once authenticated, an application is authorised to access certain service capability features.
- **Discovery of framework and network service capability features:** After successful authentication, applications can obtain available framework interfaces and use the discovery interface to obtain information on authorised network service capability features. The Discovery interface can be used at any time after successful authentication.
- **Establishment of service agreement:** Before any application can interact with a network service capability feature, a service agreement must be established. A service agreement may consist of an off-line (e.g. by physically exchanging documents) and an on-line part. The application has to sign the on-line part of the service agreement before it is allowed to access any network service capability feature.
- **Access to network service capability features:** The framework must provide access control functions to authorise the access to service capability features or service data

for any API method from an application, with the specified security level, context, domain, etc.

Some issues for SA3 to consider

1. The clientAppID received in the initiateAuthentication() can be used by the Framework to reference the correct public key for the client application (the key management system is currently outside of the scope of the OSA APIs). **Should SA3 standardise the key management reusing the mechanisms developed for Network Domain Security. Of course, OSA is intended to allow third party service providers to connect, not roaming partners, so this may not be an appropriate use of NDS security associations. Siemens provided the following paper, but there seemed to be a general unease at the concept of allowing an unknown third party access to your network purely on the basis that they were known to someone with whom you had a roaming agreement.**

S1O01004
(Temporary-Service-Ag...

2. This method is used by the client application to authenticate the Framework using the mechanism indicated in prescribedMethod. The Framework shall respond with the correct responses to the challenges presented by the client application. The number of exchanges and the order of the exchanges is dependent on the prescribedMethod. challenge : in TpString The challenge presented by the client application to be responded to by the Framework. The challenge mechanism used will be in accordance with the IETF PPP Authentication Protocols - Challenge Handshake Authentication Protocol RFC 1994, August1996 [4]. The challenge will be encrypted with the mechanism prescribed by selectEncryptionMethod(). **Is this acceptable to SA3 or do we want to suggest that we use a challenge response based on 3GPP AKA with sequence numbers etc**

3. The authType parameter identifies the type of authentication mechanism requested by the client. It provides operators and clients with the opportunity to use an alternative to the API level Authentication interface, e.g. an implementation specific authentication mechanism like CORBA Security, using the Authentication interface, or Operator specific Authentication interfaces. OSA API level Authentication is the default authentication mechanism (P_OSA_AUTHENTICATION). If P_OSA_AUTHENTICATION is selected, then the appDomain and fwDomain authInterface parameters are references to interfaces of type Ip(App)APILevelAuthentication. If P_AUTHENTICATION is selected, the authInterface parameters are references to interfaces of type Ip(App)Authentication which is used when an underlying distribution technology authentication mechanism is used**. Should SA3 provide any guidelines on the choice of operator specific authentication?**

4. Algorithm definitions TpAuthCapability This data type is identical to a TpString, and is defined as a string of characters that identify the authentication capabilities that could be supported by the OSA. Other Network operator specific capabilities may also be used, but should be preceded by the string "SP_". Capabilities may be concatenated, using commas (,) as the separation character. The following values are defined.

String Value	Description
NULL	An empty (NULL) string indicates no client capabilities.
P_DES_56	A simple transfer of secret information that is shared between the client application and the Framework with protection against interception on the link provided by the DES algorithm with a 56-bit shared secret key.
P_DES_128	A simple transfer of secret information that is shared between the client entity and the Framework with protection against interception on the link provided by the DES algorithm with a 128-bit shared secret key.
P_RSA_512	A public-key cryptography system providing authentication without prior exchange of secrets using 512-bit keys.
P_RSA_1024	A public-key cryptography system providing authentication without prior exchange of secrets using 1024-bit keys.

TpSigningAlgorithm This data type is identical to a TpString, and is defined as a string of characters that identify the signing algorithm that shall be used. Other Network operator specific capabilities may also be used, but should be preceded by the string "SP_". The following values are defined.

String Value	Description
NULL	An empty (NULL) string indicates no signing algorithm is required
P_MD5_RSA_512	MD5 takes an input message of arbitrary length and produces as output a 128-bit message digest of the input. This is then encrypted with the private key under the RSA public-key cryptography system using a 512-bit key.
P_MD5_RSA_1024	MD5 takes an input message of arbitrary length and produces as output a 128-bit message digest of the input. This is then encrypted with the private key under the RSA public- key cryptography system using a 1024-bit key

Other Network operator specific capabilities may also be used. **Should SA3 provide details of other algorithms that could be used?**

5. Security relevant sections of 29198-03/

Contact: Colin Blanchard
BTexact Technologies
MLB1 PP8
Adastral Park
Ipswich
IP5 5RE
Phone +44 1473 605353
Mobile +44 07711 191835
Fax +44 1473 623910
colin.blanchard@bt.com