INTERNATIONAL TELECOMMUNICATION UNION

TELECOMMUNICATION STANDARDIZATION SECTOR

STUDY PERIOD 2017-2020

Q16-13-Jun20-C148

STUDY GROUP 13

Original: English

Question(s): 16/13 eMeeting, 29 June 2020

CONTRIBUTION

Source:	Telecom Engineering Centre (TEC), Ministry of Communications, India
Title:	Draft new Recommendation ITU-T Y.OBF_Trust: "Open Bootstrap Framework enabling trustworthy networking and services for distributed diverse ecosystem" (e-meeting, 29 June 2020)
Purpose:	Proposal

Contact:	Abhay Shanker Verma Telecom Engineering Centre (TEC) India	Tel: + 91 9999554900 E-mail: as.verma@gov.in
Contact:	Vijay Kumar Roy Telecom Engineering Centre (TEC) India	Tel: +91 7011000101 E-mail: vk.roy@gov.in
Contact:	Ranjana Sivaram Telecom Engineering Centre (TEC) India	Tel: +91 9868136990 E-mail: ranjana.sivaram@gov.in
Contact:	Sharad Arora Sensorise Digital Services Pvt Ltd	Tel: +91 9212109999 E-mail: sharad.arora@sensorise.net
Contact:	Jonas Haggard Sensorise Digital Services Pvt Ltd	Tel: +46 702780371 E-mail: jonas.haggard@sensorise.net

Keywords: Y.OBF_Trust; Q16/13; interim; 29 June 2020

Abstract: This document proposes some modifications in the draft Recommendation ITU-T Y.OBF_Trust (TD431) for discussion at interim e-meeting of Q16/13.

1. Introduction

The draft output document of e-meeting dated 8 June 2020 was reviewed in the rapporteur conference call of 10 June 2020, where it was suggested to preferably merge the sections describing the functions and capabilities. The redundant text/ contents to be removed. Scope section also need to be edited to reflect the updated content. Accordingly, some modifications are proposed in order to address the above comments and to improve the information workflows for more clarity and readability.

2. Proposal

It is proposed to make some modifications in the draft Recommendation ITU-T Y.OBF_Trust (**TD431**). The proposed modifications are in track change mode in **Annexure-I**.

3. Reference

[1] SG13-TD431/WP3: Base document for this contribution.

_____

Annexure-I

Draft new Recommendation ITU-T Y.OBF_Trust

Open Bootstrap Framework enabling trusted devices, applications and services for distributed diverse ecosystems

Summary

This Recommendation provides an Open Bootstrap Framework (OBF) for the secure provisioning of trusted services by Application Services Providers (ASPs) that have no existing trust relationship with the users. The OBF ~~provides~~ is a trust framework ~~with a supporting~~described by OBF elements in a reference model and a functional architecture having ~~that includes an~~ OBF client function, ~~an~~ OBF authentication function, ~~an~~ OBF authorization function, OBF application function, ~~and~~ four reference points and OBF security parameters. The recommendation includes requirements of the OBF, and information workflows for the provisioning of ~~security tokens~~keying material that~~,~~ mutually authenticates~~ion between~~ trusted devices, applications and service providers. ~~and also the~~ A mechanism for users to change the service providers is also provided.

This Recommendation is relevant to network operators, IoT service providers and ASPs ~~applications and services providers~~ for deployment of trusted services in the emerging 5G, smart cities, and IoT application/services ~~and services~~ ecosystem.

Keywords

Bootstrapping; IoT Service Provider; OBF; OBF_Token; Open Bootstrap Framework; Trust Framework

Contents

Page

Draft new Recommendation ITU-T Y.OBF_Trust

Open Bootstrap Framework enabling trusted devices, applications and services for distributed diverse ecosystems

1 Scope

This Recommendation ~~specifies~~ proposes an Open Bootstrap Framework (OBF) ~~that facilitates the~~for authentication and authorization of trusted devices, trusted services, service providers and applications.

The scope of this Recommendation includes

- OBF concept ~~and overview~~;
- requirements for~~of~~ the OBF and OBF elements;~~,~~;
- ~~pre-requisites and capabilities of beneficiary devices and applications;~~
- OBF reference model;
- OBF functional architecture; ~~capabilities of the OBF nodes and reference points;~~
- ~~OBF functions;~~ and
- information workflows of the OBF.

~~This Recommendation~~ The recommendation ~~provides~~offers a framework for ~~demonstrates~~ the provisioning of trusted ASP services to the subscribers of network operators who deploy the OBF, by the use of the underlying ~~how existing~~ secure elements and bootstrapping mechanisms ~~deployed by network service providers can be used to provision trusted services by application service providers. to hitherto unknown users and devices.~~

This Recommendation also includes an industry use case~~s~~ in Appendix I, for exemplifying the deployment of the OBF.

2 References

The following ITU-T Recommendations and other references contain provisions which, through reference in this text, constitute provisions of this Recommendation. At the time of publication, the editions indicated were valid. All Recommendations and other references are subject to revision; users of this Recommendation are therefore encouraged to investigate the possibility of applying the most recent edition of the Recommendations and other references listed below. A list of the currently valid ITU-T Recommendations is regularly published. The reference to a document within this Recommendation does not give it, as a stand-alone document, the status of a Recommendation.

[ITU-T X.1113] Recommendation ITU-T X.1113 (2007), *Guideline on user authentication mechanisms for home network services*

[ITU-T X.1124] Recommendation ITU-T X.1124 (2007), *Authentication architecture for mobile end-to-end communication*

[ITU-T X.1158] Recommendation ITU-T X.1158 (2014), *Multi-factor authentication mechanisms using a mobile device*

[ITU-T X.1311] Recommendation ITU-T X.1311 (2011), *Information technology - Security framework for ubiquitous sensor networks*

[ITU-R F.1399] Recommendation ITU-R F.1399 (2001), *Vocabulary of terms for wireless access*

[ITU-T Y.3052] Recommendation ITU-T Y.3052 (2017), *Overview of trust provisioning for information and communication technology infrastructures and services*

3 Definitions

3.1 Terms defined elsewhere

This Recommendation uses the following terms defined elsewhere:

3.1.1. Secure element [ITU-T X.1158 (11/2014)]: A dedicated microprocessor system that contains an operating system, memory, application environment and security protocols intended to be used to store sensitive data and execute sensitive applications.

NOTE – A secure element may reside in a universal subscriber identity module (USIM), a dedicated chip in a phone's motherboard, an external plug in a memory card or as an integrated circuit card.

3.1.2. Security degree [ITU-T X.1124 (11/2007)]: An identifier (e.g., number) that represents a set of security parameters including at least one authentication mechanism, the crypto algorithms and related parameters to reflect the security requirement of a certain service. It is defined to profile the security requirement of each service**.**

3.1.3. Session key [ITU-T X.1113 (11/2007)]: The session key is a temporary key used to encrypt data for the current session only. The use of session keys keeps the secret keys even more secret because they are not used directly to encrypt the data. Secret keys are used to derive the session keys using various methods that combine random numbers from either the client or server or both.

3.1.4. Trust [ITU-T Y.3052 (03/2017)]: Trust is the measurable belief and/or confidence which represents accumulated value from history and the expecting value for future.

Note – Trust is quantitatively and/or qualitatively calculated and measured, which is used to evaluate values of entities, value-chains among multiple stakeholders, and human behaviours including decision making.

3.1.5. User [ITU-R F.1399 (05/2001)]: Any entity external to the network which utilizes connections through the network for communication.

3.2 Terms defined in this Recommendation

This Recommendation defines the following terms:

3.2.1. Bootstrapping: Refers to a process performed in a secure context prior to the deployment of the connected device to establish a security association between the connected devices and application/services that may have been initialized with credentials, enabling a connected device to communicate securely with application/services as well as other connected devices after their deployment. See clause 3.2.2 of [ITU-T X.1311 (02/2011)].

3.2.2. Keying material: The key data which is generated during mutual authentication procedure of the OBF client function and the authentication function and which is used to protect the security communication of the reference point RPDS. The shared keying material parameters is implementation dependent and is negotiated between the OBF client function, the authentication function and the application depending on the type of trusted services and the required security classification. See also clause 3.2.21 of [ITU-T X.1124 (11/2007)]

3.2.3. **Machine KYC:** The Process of establishing a relationship between a machine and its custodian, usually accomplished by the IoT Service Provider by the use of physical or digital verification processes that establish the linkage between the identity of the custodian and the identity of the device owned by the custodian.

3.2.4. **OBF:** A trust framework for provisioning of Trusted Services by extending the security capabilities of a network technology layer to benefit distributed and unrelated Connected Devices and Applications.

3.2.5. **OBF_Token:** A session key, independently generated in the trusted device / user equipment (UE) as well as in the authentication function, based on an agreed security schema between the device and the authentication function for establishing a secure connection between the device and the application.

3.2.6. **Subscription information:** The information that reflects the subscribing relationship among a User of a Connected Device, the ASP and the Operator of the underlying network. See also clause 3.2.22 of [ITU-T X.1124 (11/2007)]

3.2.7. **Trust framework:** A system where a set of verifiable commitments are made by each of the various parties in a transaction to their counter parties, and these commitments necessarily include: (a) controls to help ensure commitments are met and (b) remedies for failure to meet such commitments.

4 Abbreviations and acronyms

This Recommendation uses the following abbreviations and acronyms:

3GPP	3rd Generation Partnership Project
AKA	Authentication and Key Agreement
API	Application Programming Interface
COAP	Constrained Object Authentication Protocol
~~EID~~	~~eUICC-ID~~
FQDN	Fully Qualified Domain Name
GBA	Generic Bootstrapping Architecture
~~GSM~~	~~Global System for Mobile communication~~
HTTP	Hyper Text Transfer Protocol
ICT	Information and Communication Technology
IoT	Internet of Things
IoT SP	IoT Service Provider
IPSec	Internet Protocol Security
KYC	Know Your Customer
M2M	Machine to Machine
M2M SP	M2M Service Provider
MNO	Mobile Network Operator
MQTT	Message Queue Telemetry Transport
MSISDN	Mobile Station International Subscriber Directory Number

~~NSP Network Service Provider, see also clause 3.1.5 - Network Operator~~

OBF Open Bootstrap Framework

PSK Pre-Shared Key

PSK-TLS Pre-Shared Key Cipher suites for Transport Layer Security

SIM Subscriber Identification Module

TLS Transport Layer Security

~~TSP Telecom Service Provider, see also MNO~~

UID Universal Identifier or Public Entity Identifier

5 Conventions

In this Recommendation, requirements are classified as follows:

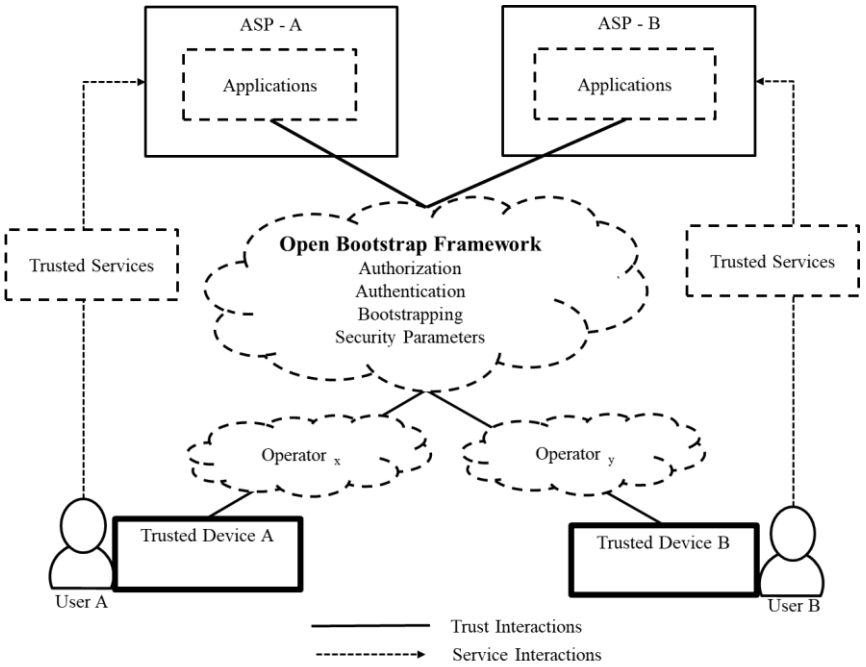
- The keywords "**is required to**" or "**are required to**" indicate a requirement/ requirements, which must be strictly followed and from which no deviation is permitted if conformance to this document is to be claimed;
- The keywords "**is recommended**" indicate a requirement, which is recommended but which is not absolutely required. Thus, such requirements need not be present to claim conformance; and
- The keywords "**optionally**" or "**may**" indicate an optional requirement which is permissible, without implying any sense of being recommended. These terms are not intended to imply that the vendor's implementation must provide the option; it means the vendor may optionally provide the feature and still claim conformance with the specification.

6 OBF Concept ~~and Overview~~

Users of new age devices and applications require secure mechanisms for accessing trusted services. At the same time, providers of trusted applications and services also require mechanisms for a minimum level of authentication of the Users. From time immemorial, the Network Operators have played the role of providing connectivity to the premises of subscribers, undertaking the subscriber verification and then allowing the connectivity to be used for a diverse set of services.

The Open Bootstrap Framework (OBF) ~~This recommendation~~ makes it possible to extend the existing trust relationship between the network operator and its subscribers to enable one to many trust relationships between the many users and the diverse new age service providers.

The OBF can enable secure service interactions between users and ASPs. This may be done by utilizing the inherent security capabilities of the underlying network technology layer such as authentication, bootstrapping and authorization to create trustful interactions between devices and applications.



~~The OBF is a set of requirements, capabilities, functions, security parameters and mechanisms that can open up the security capabilities of the network layer to all types of trusted devices, applications and services. The OBF can be implemented by any Network Operator or IoT Service Provider independent of the underlying network technology. Further, any user of a bootstrapped device can access the applications and services of any ASP by using the OBF functions and mechanisms.~~

~~The OBF can enable secure service interactions between users and ASPs. This may be done by utilizing the inherent security capabilities of the underlying network technology layer such as authentication, bootstrapping and authorization to create trustful interactions between devices and applications.~~

The concept of the OBF is shown in the diagram below:

The OBF is a set of requirements, ~~capabilities,~~ functions, security parameters and mechanisms that can open up the security capabilities of the network layer to all types of trusted devices, applications and services. The OBF can be implemented by any Network Operator or IoT Service Provider independent of the underlying network technology. Further, any user of a bootstrapped device can access the applications and services of any ASP by using the OBF functions and mechanisms.

~~The OBF can enable secure service interactions between users and ASPs. This may be done by utilizing the inherent security capabilities of the underlying network technology layer such as authentication, bootstrapping and authorization to create trustful interactions between devices and applications.~~

The OBF can address the following actors and stakeholders:

1. Users: A person that is a subscriber of the Network Operator, desirous of using trusted services from ASPs. The user provides its credentials to the ASP, whose services it intends to consume, via the Network Operator or IoT Service Provider that holds the verified credentials of the user by virtue of an earlier verification process.

Figure 6-1: OBF Concept

2. Network Operator: An entity that provides network connectivity services and undertakes the physical verification process for the subscriber. It can share the trust to bridge new relationships between providers of trusted services and users of trusted devices by employing appropriate security functions, information flows and mechanisms.
3. Applications & Service Providers (ASP): An entity that develops and offers trusted services and applications, and has a requirement for a minimum level of authentication and authorization prior to the use of its application and services by the users. However, the ASP does not have a direct relationship with the users, unlike the relationship between the Network Operator and its Subscriber. The ASP has an expectation of deriving its trust from the relationship between the Network Operator and its subscriber.

When the stakeholders engage to establish trust and security in their transactions, these are referred to as the trust interactions. In other cases, when the purpose of the engagement is to use the features and functions of the applications, these are referred to as the service interactions.

7 OBF Requirements

7.1 High-level requirements

The OBF is required to:

- identify and expose ~~and onboard~~ Network Operators and the OBF elements ~~who have~~ that have been deployed ~~the OBF~~;
- identify and onboard ASPs whose applications require to be protected from unauthorized usage;
- identify trusted devices that are authenticated by a Network Operator;
- ~~be a trust framework that~~ expose~~s~~ the inherent security capabilities of any underlying network technology for the benefit of ASPs;
- ~~allow any Network operator to enable the trust framework regardless of the underlying network technology;~~enable applications to establish secure association with trusted devices;
- identify and address the clients and the applications by using the identifiers of the underlying ICT layers;
- ~~permit authorization and de-authorization of applications for~~to a set of users;
- ~~enable multiple OBF implementations to exist simultaneously;~~
- be accessible over the public Internet;
- support industry standard protocols for key management;
- support industry standard authentication and authorization protocols;
- ~~have~~ support ~~for~~ existing bootstrapping frameworks, e.g. the 3GPP GBA [b-3GPP TS 33.220]; and
- enable a network technology agnostic identification and addressing of trusted devices.~~; and~~
- ~~protect the privacy of the sensitive user / identification information.~~
-

The OBF is recommended to:

- permit authorization and de-authorization of applications for a set of users;
- protect the privacy of the sensitive user / identification information;

- allow any Network operator to enable the trust framework regardless of the underlying network technology; and
- enable multiple OBF implementations to exist simultaneously.

~~-~~ The OBF is optionally required to

- permit a user to be authenticated by any one of the many network operators of which the user is a subscriber.~~;~~

- ~~allow bootstrapping of trusted devices, authentication of users and authorization of applications based on the trust provided by the Network Operator;~~
- ~~protect the access to trusted applications by permitting only the authenticated and authorized users to access the trusted applications;~~
- ~~enable applications to establish secure association with trusted devices;~~
- ~~identify and address the clients and the applications by using the identifiers of the underlying ICT layers;~~
- ~~permit authorization and de-authorization of applications to a set of users;~~
- ~~enable multiple OBF implementations to exist simultaneously;~~
- ~~be accessible over the public Internet;~~
- ~~support industry standard protocols for key management;~~
- ~~support industry authentication and authorization protocols;~~
- ~~have support for existing bootstrapping frameworks, e.g. the 3GPP GBA [b-3GPP TS 33.220];~~
- ~~enable a network technology agnostic identification and addressing of trusted devices; and~~
- ~~protect the privacy of the sensitive user / identification information.~~

~~8 OBF pre-requisites for the trusted devices and applications~~

~~8~~7.~~1~~2 Pre-requisites for the trusted devices

In order to use the OBF, the trusted devices ~~is~~are required~~have~~ to:

- host the OBF client function in the device or its connectivity element (e.g. SIM card);
- be OBF aware, and initiate the bootstrapping process, ~~if~~ when the OBF application requires it~~indicates the need~~;
- support the application specific protocol over the reference point between the device and the application such as HTTP, MQTT, Web Sockets or COAP;
- support HTTP Digest AKA protocol and optionally others as required by the underlying network technology or application; and
- discover, identify, address and connect to the authentication function relevant to the realm of the trusted device.

~~-~~ The trusted devices may optionally host a secure element to satisfy the security degree of the application.

- ~~support HTTP Digest AKA protocol and optionally others as required by the underlying network technology or application;~~
- ~~discover, identify, address and connect to the authentication function relevant to the realm of the trusted device; and~~

~~optionally host a secure element to satisfy the security degree of the application.~~

~~8~~7.~~2~~3 Pre-requisites for applications

After the bootstrapping is completed, the trusted device and the application can run an application specific protocol, where the authentication of messages will be based on the keying material generated during the mutual authentication between the OBF client function and the authentication function.

In order to use the OBF, the applications ~~is~~are required~~have~~ to:

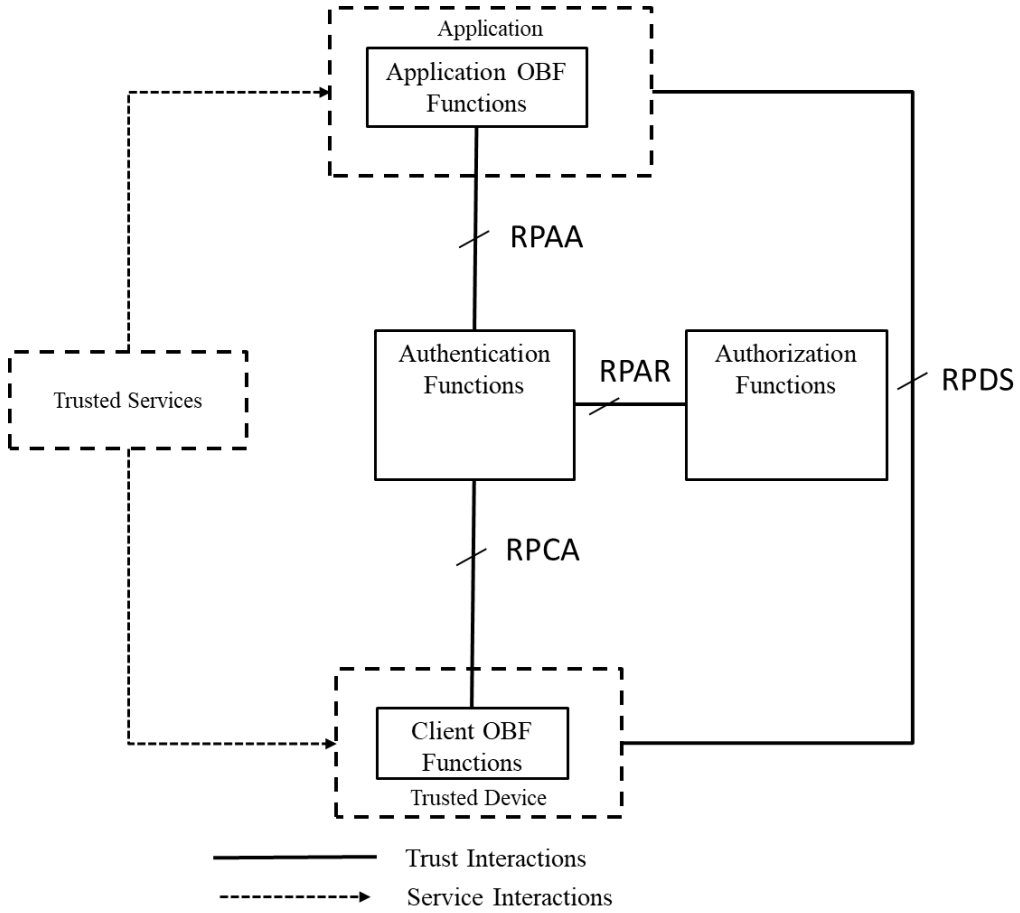
- be OBF aware, and indicate to the device the need for a keying material if it attempts to connect to the application without one;
- be able to locate and communicate securely with the user's authentication function;
- acquire the keying material to secure the interactions with the device;
- implement Diameter / HTTP proxy functionality to act as a proxy towards the authentication function of the realm in which the user is bootstrapped; and
- acquire the user's security parameters from the authorization function via the authentication function.~~; and~~

~~The trusted devices are recommended to~~

- configure the key lifetime and validity settings.

~~9~~8 OBF Reference Model

~~This recommendation envisages the need for a~~The OBF reference model ~~which includes~~describes ~~certain~~ the key function~~s~~s and the reference points ~~to address~~over which the functions interact with



each other. The trusted device and the application are also shown in the diagram as these are the beneficiaries of the trust framework. ~~the requirements identified in the previous sections~~

~~.~~ The OBF reference model ~~provides the main functions and the reference points which are~~is shown

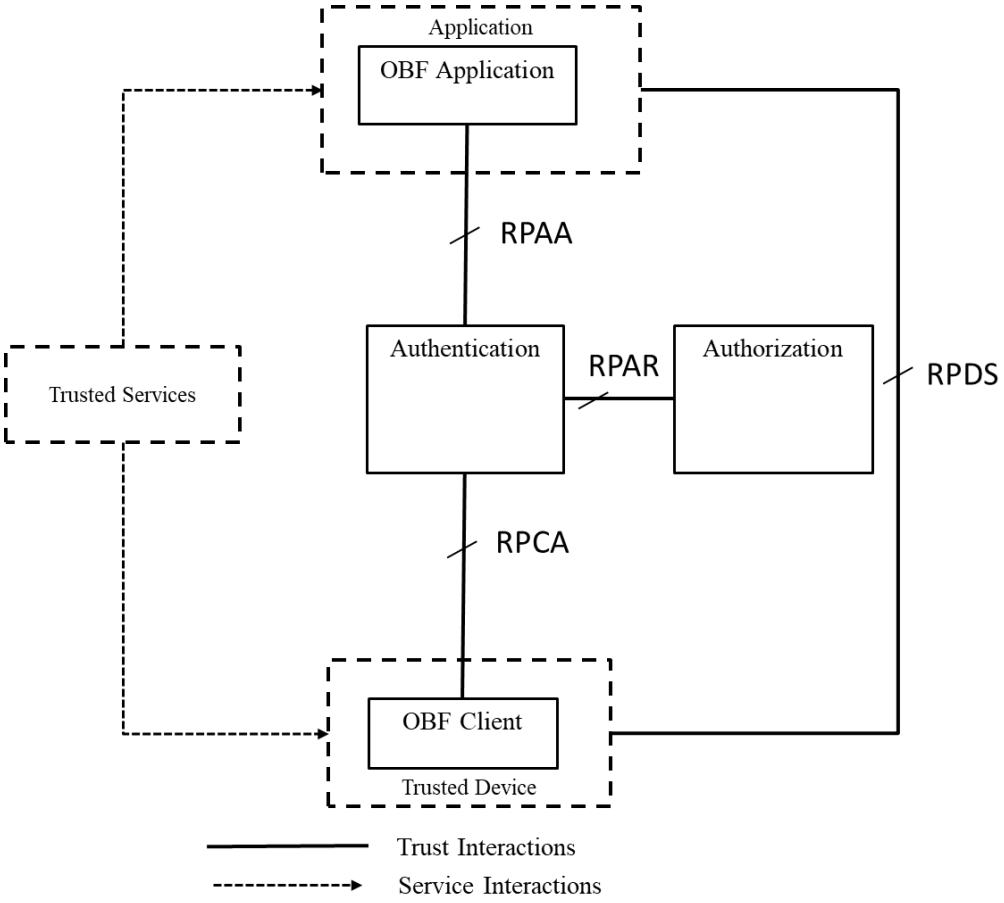


Figure ~~8~~9-1: OBF reference model

in the diagram below:

~~The trusted device and the application are also shown in the diagram as these are the beneficiaries of the trust framework.~~

~~9~~8.1 OBF elements ~~OBF functions~~

~~9~~8.1.1 OBF client ~~function~~

The OBF client ~~function~~ is an application resident in the trusted device or its associated connectivity element (e.g. the SIM or the authentication element) that provides the bootstrapping application and the keying material on the device for the bootstrapping of the trusted device using the authentication ~~function~~. The OBF client ~~function provides~~has the features and functions required for the interaction with the authentication ~~function~~, authorization ~~function~~ and the application. The OBF client ~~function~~ is specified and provisioned by the network operator that is providing the OBF services.

~~9~~8.1.2 OBF authorization ~~function~~

The OBF authorization ~~function~~ is a node provisioned by the network operator that ~~provides~~ carries out the key management ~~function~~ and provides the keying material as per standard AKA protocols. The authorization ~~function~~ hosts the subscription information of ASPs.

The authorization function is the repository of the UIDs of ASPs that are authorized to provide services. It also hosts the mapping between applications registered by ASPs and the access rights provided to the users as a list of OBF client function identifiers.

The authorization function provides the mechanisms for the network operator to authorize ASPs to offer certain services and users to access the authorized services of the ASP.

98.1.3 OBF authentication function

The OBF authentication function is a node provisioned by the Network Operator that identifies and authenticates the OBF client function using the keying material from the OBF authorization function as per standard AKA protocols and the agreed authentication algorithms.

The authentication function generates the OBF_Token, and shares it with the authorized ASPs that are authorized by the authorization function.

98.1.4 OBF OBF application function (Application OBF functions)

The OBF Application OBF application function is a node, which receives OBF_Token from the authentication function upon successful bootstrap. The functionIt stores the OBF_Token and uses it the same in setting up secure connections between the application client and the application.

98.2 OBF reference points

The OBF specifies four reference points as , each of which is described below:

89.2.1 RPAA

RPAA is the reference point between the authentication function and the application.

It is used by the application to fetch the OBF_Token from the authentication function. It is also used to fetch application-specific subscription information of the user from the authentication function if requested.

The reference point RPAA has the capability to:

- allow the transfer of user's subscription information to enforce access control policies between trusted devices and the applications;
- support the DIAMETER [b-RFC 6733] and [b-RFC 7155] protocol;
- enable secure communication between the authentication function and the application;
- allow the application to send its address (e.g. FQDN), public entity identity (e.g., UID), basic key material (e.g., a shared secret or a public key certificate), entity service permission flag, supported authentication mechanisms and the authentication inquiring and key generation mechanism to the authentication function;
- allow the authentication function to verify that the application is authorized to obtain the identifiers, key material and subscription information for a user;
- allow the application to indicate to the authentication function the single application or several applications for which it requires user identity and security parameters;
- allow the application to obtain a selected set of application-specific user security parameters;
- allow the transfer of the OBF_Token from the authentication function to the application; and
- allow the application to indicate to the authentication function the protocol identifier of the RPDS security protocol for which it requires the keying material.

98.2.2 RPAR

RPAR is the reference point between OBF authentication function and the OBF authorization function.

The OBF authentication function uses the RPAR to obtain the subscription information regarding the OBF client functions when users attempt to access certain ASP applications. The reference point also provides the keying material for the OBF client functions during the bootstrapping mechanism.

The reference point RPAR has the capability to:

- allow identification and mutual authentication between the authentication function and authorization function on supported DIAMETER [b-RFC 6733] and [b-RFC 7155] protocol;
- allow the transfer of security parameter required for bootstrapping;
- allow the transfer of subscription information to establish the access control policies between trusted devices and the applications;
- allow the authentication function to request bootstrapping information for specific users; and
- allow the authorization function to send the user's security parameters to the authentication function.

98.2.3 RPCA

RPCA is the reference point between the OBF client function hosted in the trusted device and the OBF authentication function. The reference point provides the bootstrapping of the OBF client function to the OBF authentication function.

The reference point RPCA has the capability to:

to establish the identity of the OBF client function of a trusted device to the authentication function;

support the HTTP Digest protocol [b-RFC7616], it may optionally support other protocols as well

use the agreed AKA for authentication between authentication function and the OBF client function;

transfer the identification of the OBF client function using the OBF identifier;

support the bootstrapping process between the OBF client function and the authentication function;

identify and mutually authenticate the trusted device and the application using the OBF client function and the authentication function; and

establish the OBF_Token between the authentication function and the OBF client function.

89.2.4 RPDS

RPDS is the reference point between the trusted device and the application.

The reference point supports any protocol as required for the interaction between the application client function and the application, which is secured using the OBF_Token.

The reference point RPDS has the capability to:

- support the application-specific protocol between the trusted device and the application;
- send the indication from the application to the trusted device that a valid OBF_Token is required prior to connecting to the application;
- support the use of the OBF_Token for creating the secure association between the trusted device and the application;
- allow the application to indicate to the application client function, the invalid OBF_Token for the required authentication;

~~— enable the negotiation and selection of the key between the client function and the application;~~

~~— use a security protocol identifier as required by the underlying network technology layer;~~

~~— allow the application to signal to the application client function regarding lifecycle management of keys; and~~

~~— enable the use of the OBF_Token for securing the association between the application client function and the application.~~

~~10~~9 OBF functional architecture

Based upon the OBF reference model, and the detailed analysis of the requirements, a functional architecture for the OBF is presented in the diagram below:

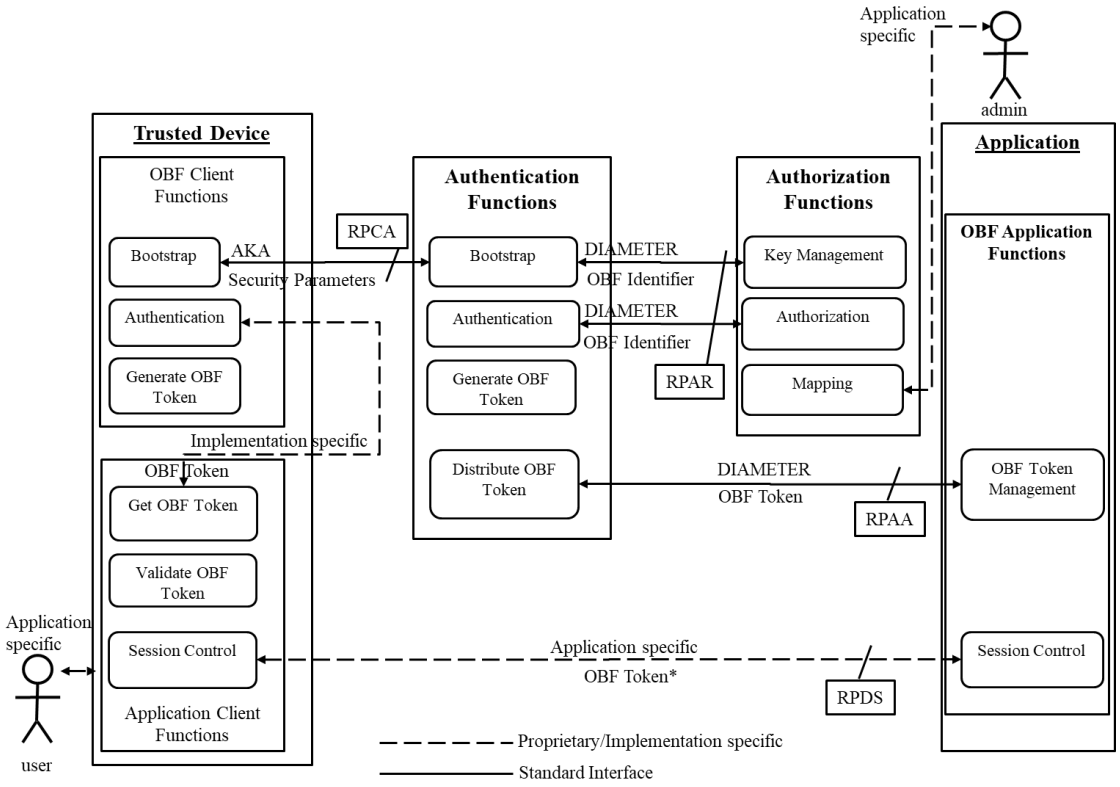
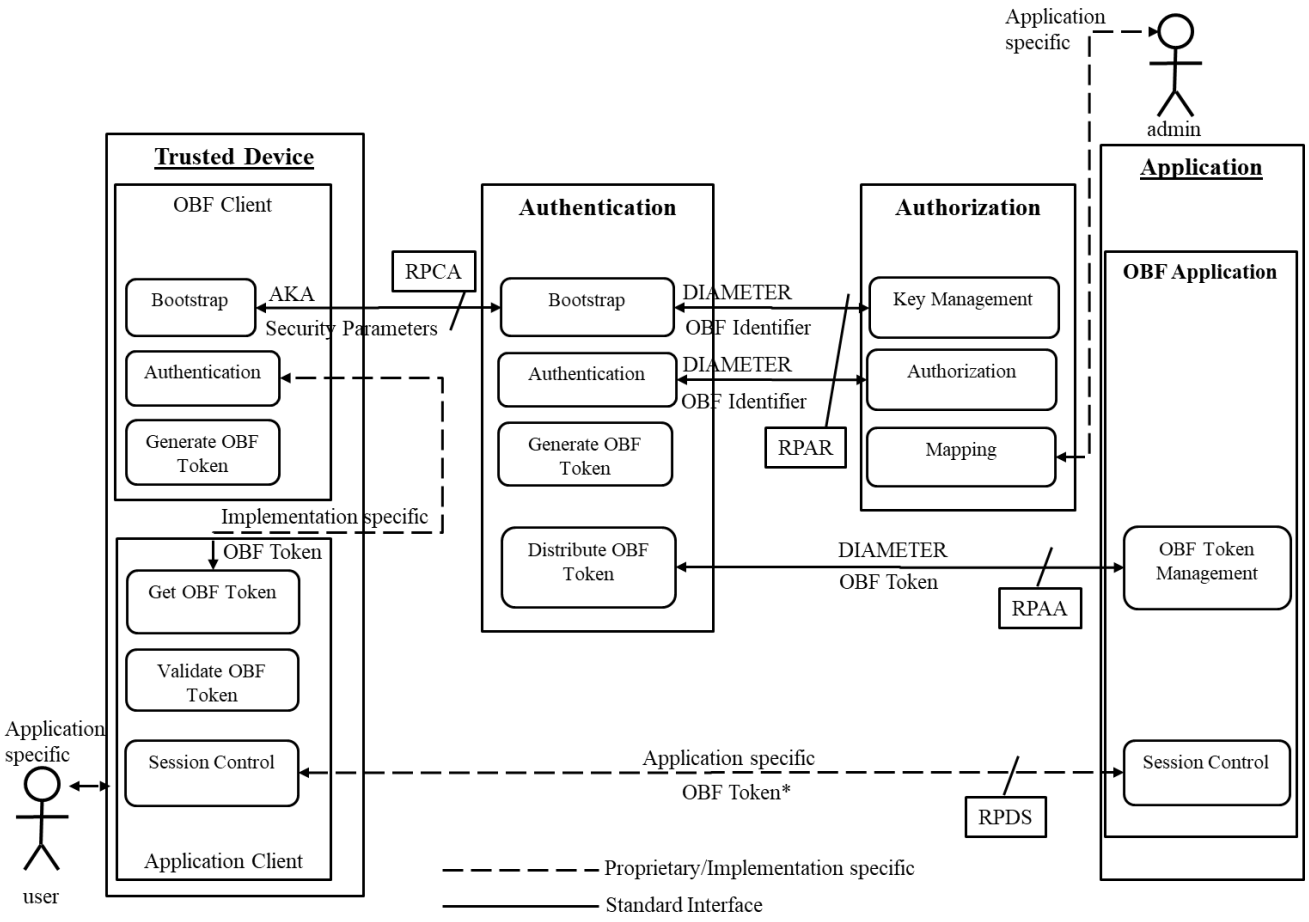


Figure ~~109~~-1: OBF Functional Architecture

Figure ~~109~~-1: OBF Functional Architecture



~~In addition to the actors, main functions and the~~The functional architecture ~~reference points describe~~presents:

- ~~d in the reference model, the~~ the OBF functions;~~al architecture provides for -~~
- the reference points; and
- ~~the security parameters used within detailed sub-functions of the main functions;~~
- ~~the security parameters of the OBF; and~~
- the OBF elements~~the information workflows~~.

~~All the OBF functions al nodes are have the capability to:~~

- ~~identify and authenticate the trusted nodes and clients within an OBF realm;~~
- ~~support transferability such that a user is free to choose services from any network operator or ASP; and~~
- ~~enable identification and authentication of trusted nodes and clients.~~

9.1 OBF functions

Note: When the OBF is deployed in the network of a network operator, the context is referred as a realm. The instantiated functions within the realm are referred as nodes. As an example, an authorisation function, when instantiated in the network of a Network Operator, is referred as the Authentication Node in the realm of the operator.

All the OBF nodes:

- implement the OBF numbering;
- identify and authenticate each other within the OBF realm(s);
- identify and authenticate OBF clients; and
- support transferability such that a user is free to choose services from any network operator or ASP.

The OBF functions / sub-functions are described below.

~~10.1 Authentication functions~~

~~10~~9.1.1 Authentication function

The function mutually authenticates the OBF Client and the authentication function, as an enabling step in the process towards generation of long-term keying material within the bootstrapping function. The function is executed over the reference point RPCA.

~~The capabilities of~~ OBF authentication function ~~has the capability to~~are~~is~~:

- ~~to~~ maintains the list of users, authorized applications and the related subscription parameters;
- ~~to~~ protects the use of the network subscriber identity against discovery and misuse;

- supportsto handle AKA protocols such that it can support the one used by the underlying network technology layer;
- to manages the lifecycle of keys as per the agreed AKA protocol;
- to configures and communicates the format of the OBF identifier to the OBF client function; and
- to configures the OBF subscription information in conjunction with the authorization function and communicate that to the OBF client function.
-

109.1.2 Bootstrapping function

The This function, hosted in the authentication function as well as in the OBF client function, which creates a new registration for the trusted device by way of establishment of new long-term secret keys for secure communication.

109.1.3 Generate OBF_Token function

This function is responsible to generate the OBF_-Token after bootstrapping has successfully been completed by using the agreed keying material and algorithms. The new association provides for mutual authentication of the devices and applications hitherto unknown to each other. The OBF_Token is generated as per the subscription information specific to an application, and its applicability is limited to a specific application.

109.1.4 Distribute OBF_Token function

This function securely transfers the OBF_Token to the application, so it can be used by the session functions in the application. When a device is bootstrapped a new OBF_Token is generated, this token must be distributed to the correct application. This function securely transfers the OBF_Token to the application, so it can be used by the session functions in the application.

10.2 Authorization functions

109.21.15 Key management function

The function provides the mechanisms for management and association of keys and algorithms between the authorization function and the OBF client function. It stores the pre-shared keys or certificates corresponding to the trusted devices and manages the keys and lifecycle of the keying material as per the agreed AKA protocol.

109.21.26 Authorization function

Thise function validates if the OBF client function has the right to use the authentication for the requested application. The function hosts the repository of registered applications that can be permitted for use by the device / user, and also the mapping of the specific applications that are allowed to be used by a user / OBF client function.

The The capabilities of OBF authorization function has the capability tois:

- to supports the protocols required over the reference point RPAA;
- to provisions the users and applications with the required application security parameters; and
- to responds to the authentication function over DIAMETER with the authentication vector and user's security parameters such as the key lifetime and user identities.;
-

~~10~~9.~~2~~1.~~3~~ 7 Mapping function

The mapping function is an administrative function to map users, trusted devices and permitted applications. This can be done on an individual level, or based on the agreement between the user and the OBF provider~~.~~

~~The~~ The mapping ~~administrative~~ function enables ~~includesis~~:

- ~~to enable~~ addition / deletion of authorized devices / users through standardized API or user interfaces;~~;~~
- ~~to enable~~ delegation / revocation of access control rights to author~~ised~~ized OBF client functions through standardized API or user interfaces;~~;~~
- ~~to enable~~ addition / deletion of authorized application providers / applications through standardized API or user interfaces and enables provisioning; and
- ~~to and~~ de-provisioning of authorized users of application through standardized API or user interfaces.

~~10~~9.1.~~8~~ 3 OBF client function~~s~~

The OBF client function ~~function is~~has ~~the capability to~~:

- ~~to~~ interacts with the secure element of the trusted device or the connectivity element;
- ~~to~~ supports the required AKA protocol;
- ~~to~~ stores the keying material and select from one amongst several keys for security enablement;
- ~~to~~ selects from one amongst several available authentication functions, allowing services of only one authentication function at a given point in time;
- ~~to~~ generates and / or retrieve the OBF identifier as per the selected authentication function;
- ~~to~~ securely stores the security parameters including identifiers, subscription information and the OBF_Token;
- ~~to~~ generates the OBF_Token as per security parameters negotiated during the bootstrapping process;
- ~~to~~ protects the use of the network subscriber identity against discovery and misuse; and
- ~~to~~ supports the application protocol in the reference point RPDS and initiate the bootstrapping process if indicated by the application.

~~10.3.1 Authentication function~~

~~See 10.1.1~~

~~10.3.2 Bootstrapping function~~

~~See 10.1.2~~

~~10.3.3 Generate OBF_Token function~~

~~See 10.1.3~~

~~10.4 Application client functions~~

~~10~~9.~~4~~1.~~1~~9 Validate OBF_Token function

This function validates the lifetime of the OBF_Token. When the application client function is started, or required to ~~wants to communicate~~initiate ~~securely with~~ the interaction with the application, the ~~current~~ OBF_Token is validated to ensure the lifetime of the token has not expired. If the lifetime has expired or if no current OBF_Token is available or when indicated by the application, the application client function will use the get OBF_Token function to obtain a new OBF_Token.

~~10~~9.~~4~~1.~~2~~10 Get OBF_Token function

This function is used to initiate the bootstrapping of the trusted device by calling the OBF client function, leading to the device obtaining a new OBF_Token.

~~10~~9.~~4~~1.~~3~~11 Session control function

This function is~~e session control function is~~ application specific, it~~but~~ utilizes the OBF_Token to initiate and maintain a secure session towards the application. The ~~session control~~ function is~~can~~ be implemented within an industry standard session control such as TLS PSK, Kerberos, IPSec~~, etc.~~.

~~10.5 Application functions~~

~~10~~9.~~5~~1.1~~2~~ OBF_Token management function

When an OBF_Token has been generated it must be distributed to the application, the OBF_Token management function shall receive and store the OBF_Token for future sessions.

~~10.5.2 Session control function~~

~~See 10.4.3~~

9.2 OBF reference points

The OBF specifies four reference points, each of which is described below:

9.2.1 RPAA

RPAA is the reference point between the authentication function and the application. It is used by the application to fetch the OBF_Token from the authentication function. It is also used to fetch application-specific subscription information of the user from the authentication function if requested.

The reference point RPAA ~~has the capability~~is ~~to~~:

- allows the transfer of user's subscription information to enforce access control policies between trusted devices and the applications;
- supports the DIAMETER [b-RFC 6733] and [b-RFC 7155] protocol;
- enables secure communication between the authentication function and the application;
- allows the application to send its address (e.g. FQDN), public entity identity (e.g., UID), basic key material (e.g., a shared secret or a public-key certificate), entity service permission flag, supported authentication mechanisms and the authentication inquiring and key generation mechanism to the authentication function;
- allows the authentication function to verify that the application is authorized to obtain the identifiers, key material and subscription information for a user;

- allows the application to indicate to the authentication function the single application or several applications for which it requires user identity and security parameters;
- allows the application to obtain a selected set of application-specific user security parameters;
- allows the transfer of the OBF_Token from the authentication function to the application; and
- allows the application to indicate to the authentication function the protocol identifier of the RPDS security protocol for which it requires the keying material.

9.2.2 RPAR

RPAR is the reference point between OBF authentication function and the OBF authorization function. The OBF authentication function uses the RPAR to obtain the subscription information regarding the OBF client functions when users attempt to access certain ASP applications. The reference point also provides the keying material for the OBF client functions during the bootstrapping mechanism.

The reference point RPAR allows ~~has the capability~~is ~~to~~:

- ~~allow~~ identification and mutual authentication between the authentication function and authorization function on supported DIAMETER [b-RFC 6733] and [b-RFC 7155] protocol;
- ~~allow t~~the transfer of security parameter required for bootstrapping;
- ~~allow~~ the transfer of subscription information to establish the access control policies between trusted devices and the applications;
- ~~allow~~ the authentication function to request bootstrapping information for specific users; and
- ~~allow~~ the authorization function to send the user's security parameters to the authentication function.

9.2.3 RPCA

RPCA is the reference point between the OBF client function hosted in the trusted device and the OBF authentication function. The reference point provides the bootstrapping of the OBF client function to the OBF authentication function.

The reference point RPCA ~~has the capability~~is ~~to~~:

- ~~to~~ establishes the identity of the OBF client function of a trusted device to the authentication function;
- supports the HTTP Digest protocol [b-RFC7616], it may optionally support other protocols as well
- uses the agreed AKA for authentication between authentication function and the OBF client function;
- transfers the identification of the OBF client function using the OBF identifier;
- supports the bootstrapping process between the OBF client function and the authentication function;
- identifiesy and mutually authenticates the trusted device and the application using the OBF client function and the authentication function; and
- establishes the OBF_Token between the authentication function and the OBF client function.

9.2.4 RPDS

RPDS is the reference point between the trusted device and the application. The reference point supports any protocol as required for the interaction between the application client function and the application, which is secured using the OBF_Token.

The reference point RPDS ~~has the capability~~is ~~to~~:

- supports the application-specific protocol between the trusted device and the application;
- sends the indication from the application to the trusted device that a valid OBF_Token is required prior to connecting to the application;
- supports the use of the OBF_Token for creating the secure association between the trusted device and the application;
- allows the application to indicate to the application client function, the invalid OBF_Token for the required authentication;
- enables the negotiation and selection of the key between the client function and the application;
- uses a security protocol identifier as required by the underlying network technology layer;
- allows the application to signal to the application client function regarding lifecycle management of keys; and
- enables the use of the OBF_Token for securing the association between the application client function and the application.

~~11. XXX~~

~~10~~19.~~6~~13 Security Parameters

The security parameters include identifiers, subscription information and the keying material i.e. OBF_Token. The purpose of the identifiers is to uniquely identity and address the OBF client functions and the OBF nodes in an OBF implementation. The purpose of the subscription information is to authenticate and authoriz~~s~~e the secure interactions between users and ASPs via the network operator.

The security parameters are implementation specific, and can change significantly from one deployment to another. They are determined by several factors, including but not limited to, the OBF deployment model, the underlying network technology, the AKA protocol, the numbering/ identification mechanism of the network and internet layer, by the service type and the security degree required for the use case, etc.

~~10~~9.~~6~~3.1 Identifiers

The OBF identifiers uniquely identify an OBF client function, a bootstrapped trusted device to an authentication function and the application. The OBF provides for the following identifiers:

a. OBF Node Identifier;
b. ~~OBF Client~~OBF client Identifier;
c. OBF Security Protocol Identifier

The description of the various identifiers is provided below.

(a) **OBF Node Identifier**:

The OBF node identifier comprises such minimum connection and security attributes that can uniquely address and fully support the OBF authentication function from one of many in

multiple technology domains. As an example, an authentication function will require the node's FQDN and the Global Title Address and the associated AKA to fully qualify the requirement of the OBF node identifier, when such a node is deployed in a GSM network. The OBF node identifier provides an implementation dependent address, connection and security information of the authentication function.

(b) ~~OBF Client~~OBF client **Identifier**:
It is an identifier of the OBF client function or the trusted device, which includes at least a network technology identifier, underlying network layer identifier of the device, and IP layer identifier of the device.

(c) **OBF Security Protocol Identifier**:
It is an identifier, which is associated with a security protocol over reference point RPDS. The OBF security protocol identifier is a string of five octets. The first octet denotes the organization, which specifies the security protocol. The remaining four octets denote a specific security protocol as per Annex-H of [b-3GPP TS 33.220] within the responsibility of the organization.

~~10~~9.~~6~~3.2 Subscription Information

Subscription information [ITU-T X.1124 (11/2007)] between a user and its home network contains the user's private entity identifier (e.g., MSISDN), the basic key material (e.g., a shared secret or a public-key certificate) and its lifetime, entity service permission flag (i.e., whether it is allowed to request a specific service), the supported authentication mechanism(s) (e.g., HTTP authentication and key agreement, Diffie-Hellman based authentication mechanisms, a biometric authentication mechanism, etc.), and the authentication inquiring and key generation mechanism (e.g., GBA, Kerberos, Mediation), etc. Subscription information between an ASP and a network operator contains the ASP's identity information and public entity identifier (e.g., UID) according to the service, optionally the basic key material (e.g., a shared secret or a public-key certificate) and its lifetime, entity service permission flag (e.g., whether it is allowed to provide a specific service), the supported authentication mechanisms (e.g., certificate-based TLS authentication mechanism, PSK-TLS, IPSec), and the authentication inquiring and key generation mechanism (e.g., GBA, Kerberos, Mediation), etc.

The subscription information related to the user and its authentication function is delivered to the OBF client function from the authorization function via the authentication function during the bootstrapping process. The subscription information related to the application (e.g. access to application allowed, type of certificates which may be issued) is sent to the OBF client function.

In addition, the subscription information contains a mechanism for key selection, which is used in the OBF client function to mandate the usage of either the trusted device-based key or the external secure element based key or both.

~~10~~19.~~7~~24 OBF_Token

The OBF_Token binds the user's identity to the keying material in the reference points. The OBF_Token is a session key, independently generated in the OBF client function as well as in the authentication function. The OBF_Token is generated by using the security parameters negotiated as part of the bootstrapping process. It is used for establishing a secure session between the trusted device and the application. The timestamp of the OBF_Token is synchron~~ised~~ized and controlled by the authentication function.

The OBF_Token is the session key, which is used to establish a secure session between the application client and the application. The OBF client function binds the user's identity to the keying material in reference points.

The OBF_Token ~~is to~~:

- binds the user identity to the keying material used in the reference points;
- is ~~be~~ the globally unique identifier of realm of the OBF in which it is issued;
- supports any underlying network technology;
- identifies~~y~~ the realm of the OBF in which it is issued;
- serves as a temporary identifier of the user;
- is ~~be used as~~ a key identifier in protocols used in reference point RPCA and RPDS;
- enables the application to detect and address the authentication function that has sponsored the OBF_Token; and
- ~~be in a f~~has a format that is usable by the underlying network technology layer bootstrapping capabilities.

~~11 Capabilities of OBF nodes and reference points~~

~~11.1 Capabilities on the OBF nodes~~

~~11.2.1 Common capabilities of the OBF nodes~~

~~All the OBF Nodes have the capability to:~~

- ~~identify and authenticate the trusted nodes and clients within an OBF realm;~~
- ~~support transferability such that a user is free to choose services from any network operator or ASP; and~~
- ~~enable identification and authentication of trusted nodes and clients.~~

~~11.2.2 Capabilities of the OBF client function~~

~~The OBF client function has the capability to:~~

- ~~interact with the secure element of the trusted device or the connectivity element;~~
- ~~support the required AKA protocol;~~
- ~~store the keying material and select from one amongst several keys for security enablement;~~
- ~~select from one amongst several available authentication functions, allowing services of only one authentication function at a given point in time;~~
- ~~generate and / or retrieve the OBF identifier as per the selected authentication function;~~
- ~~securely store the security parameters including identifiers, subscription information and the OBF_Token;~~
- ~~generate the OBF_Token as per security parameters negotiated during the bootstrapping process;~~
- ~~protect the use of the network subscriber identity against discovery and misuse; and~~
- ~~support the application protocol in the reference point RPDS and initiate the bootstrapping process if indicated by the application.~~

11.2.3 Capabilities of the OBF authorization function

The OBF authorization function has the capability to:

- support the protocols required over the reference point RPAA;
- store the pre-shared keys or certificates corresponding to the trusted devices;
- manage the keys and lifecycle of the keying material as per the agreed AKA protocol;
- provision the users and applications with the required application security parameters;
- respond to the authentication function over DIAMETER with the authentication vector and user's security parameters such as the key lifetime and user identities;
- enable addition / deletion of authorized devices / users through standardized API or user interfaces;
- enable delegation / revocation of access control rights to authorised OBF client functions through standardized API or user interfaces;
- enable addition / deletion of authorized application providers / applications through standardized API or user interfaces; and
- enable provisioning and de-provisioning of authorized users of application through standardized API or user interfaces.

11.2.4 Capabilities of the OBF authentication function

The OBF authentication function has the capability to:

- maintain the list of users, authorized applications and the related subscription parameters;
- protect the use of the network subscriber identity against discovery and misuse;
- handle AKA protocols such that it can support the one used by the underlying network technology layer;
- manage the lifecycle of keys as per the agreed AKA protocol;
- configure and communicate the format of the OBF identifier to the OBF client function;
- configure the OBF subscription information in conjunction with the authorization function and communicate that to the OBF client function; and
- generate the OBF_Token as per the subscription information specific to an application, and limit the applicability of the OBF_Token to a specific application.

11.3 Capabilities of the Reference Points

11.3.1 Capabilities of the RPAA

The reference point RPAA has the capability to:

- allow the transfer of user's subscription information to enforce access control policies between trusted devices and the applications;
- support the DIAMETER [b-RFC 6733] and [b-RFC 7155] protocol;
- enable secure communication between the authentication function and the application;
- allow the application to send its address (e.g. FQDN), public entity identity (e.g., UID), basic key material (e.g., a shared secret or a public-key certificate), entity service permission flag, supported authentication mechanisms and the authentication inquiring and key generation mechanism to the authentication function;
- allow the authentication function to verify that the application is authorized to obtain the identifiers, key material and subscription information for a user;
- allow the application to indicate to the authentication function the single application or several applications for which it requires user identity and security parameters;
- allow the application to obtain a selected set of application-specific user security parameters;
- allow the transfer of the OBF_Token from the authentication function to the application; and
- allow the application to indicate to the authentication function the protocol identifier of the RPDS security protocol for which it requires the keying material.

11.3.2 Capabilities of the RPAR

The reference point RPAR has the capability to:

- allow identification and mutual authentication between the authentication function and authorization function on supported DIAMETER [b-RFC 6733] and [b-RFC 7155] protocol;
- allow the transfer of security parameter required for bootstrapping;
- allow the transfer of subscription information to establish the access control policies between trusted devices and the applications;
- allow the authentication function to request bootstrapping information for specific users; and
- allow the authorization function to send the user's security parameters to the authentication function.

11.3.3 Capabilities of the RPCA

The reference point RPCA has the capability to:

- to establish the identity of the OBF client function of a trusted device to the authentication function;
- support the HTTP Digest protocol [b-RFC7616], it may optionally support other protocols as well
- use the agreed AKA for authentication between authentication function and the OBF client function;
- transfer the identification of the OBF client function using the OBF identifier;
- support the bootstrapping process between the OBF client function and the authentication function;

- identify and mutually authenticate the trusted device and the application using the OBF client function and the authentication function; and
- establish the OBF_Token between the authentication function and the OBF client function;

11.3.4 Capabilities of the RPDS

The reference point RPDS has the capability to:

- support the application-specific protocol between the trusted device and the application;
- send the indication from the application to the trusted device that a valid OBF_Token is required prior to connecting to the application;
- support the use of the OBF_Token for creating the secure association between the trusted device and the application;
- allow the application to indicate to the application client function, the invalid OBF_Token for the required authentication;
- enable the negotiation and selection of the key between the client function and the application;
- use a security protocol identifier as required by the underlying network technology layer;
- allow the application to signal to the application client function regarding lifecycle management of keys; and
- enable the use of the OBF_Token for securing the association between the application client function and the application.

11.4 Other Capabilities

The OBF_Token has the capability to:

- bind the user identity to the keying material used in the reference points;
- be the globally unique identifier of realm of the OBF in which it is issued;
- support any underlying network technology;
- identify the realm of the OBF in which it is issued;
- serve as a temporary identifier of the user;
- be used as a key identifier in protocols used in reference point RPCA and RPDS;
- enable the application to detect and address the authentication function that has sponsored the OBF_Token; and
- be in a format that is usable by the underlying network technology layer bootstrapping capabilities.

12111**0** Information Workflows

The detailed workflows showing the service and trust interactions are described below.

12111**0**.1 Bootstrapping with symmetric keys

Prior to using the authentication services of the OBF, the OBF client function of the device performs a bootstrapping workflow with the authentication function.

The bootstrapping function uses the symmetric (pre-shared) keys, which exist on, both, the secure element of the device and in the authorization function. These keys are used to mutually authenticate the OBF client function and the authentication function.

After the mutual authentication, the session keys are generated which are used for securing the communication between the trusted device and an application. This process is accomplished in the following steps:

1. The authentication function will validate the OBF client function in the bootstrapping stage;
2. The authentication function and the OBF client function will mutually engage in a challenge-response mechanism to validate credentials;
3. The authorization function validates, if the user has the right to use the bootstrapping for the given application;
4. When the mutual authentication has been completed, the OBF client function and the authentication function generate an OBF_Token as per the agreed AKA protocol; and
5. The OBF_Token is provided to the application for use in subsequent security associations.

Note: The steps 1, 2, 3 are a part of the digest access authentication AKA.

The bootstrapping and the session key management process is described in the diagram below (Figure ~~11~~10-1) in which the numbering of the steps in the diagram follows the numbering of steps in the paragraph above:

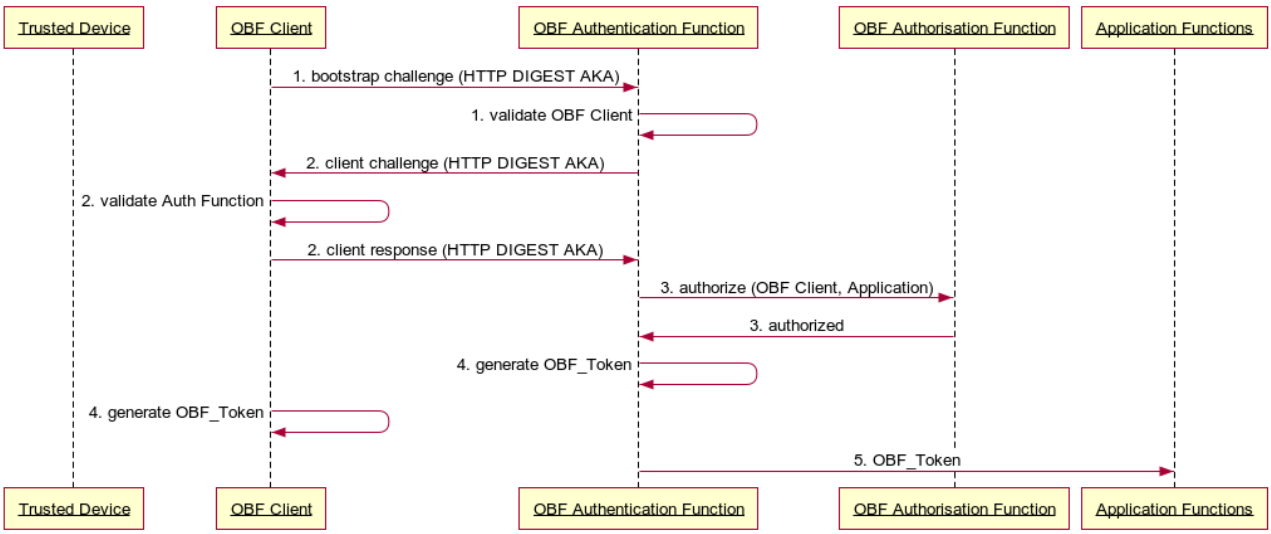


Figure ~~11~~10-1: Bootstrapping workflow

NOTE: The workflow for bootstrapping using asymmetric keys is similar, with the exception that in place of pre-shared keys the public keys are used for authentication.

~~11~~10.2 Authentication workflow

~~**Authentication workflow**~~

When a User requires to access an application from the trusted device, or the application requires to exchange data with the trusted device, it signals to the OBF client function to use the bootstrap framework for authentication. This process i~~s~~n accomplished in the following steps, provided that the bootstrapping has been completed as per ~~11~~10.1:

~~1. Bootstrapping is initiated, if it has not been executed previously [please see 12.2 below];~~

~~2.~~1. The user request towards the application is executed and the application uses a challenge-response mechanism to identify and authenticate the user and the user responds to the challenge-response mechanism used by the application; and

~~3.~~2. The OBF client function uses the OBF_Token, which is used to set up a secure connection using TLS for any data exchange between the application client function on the trusted device and the application.

NOTE – The mechanism to invoke the OBF client function for initiating the bootstrap procedure is left to the implementation and not covered in the scope of this recommendation.

The Authentication workflow is described in the diagram below:

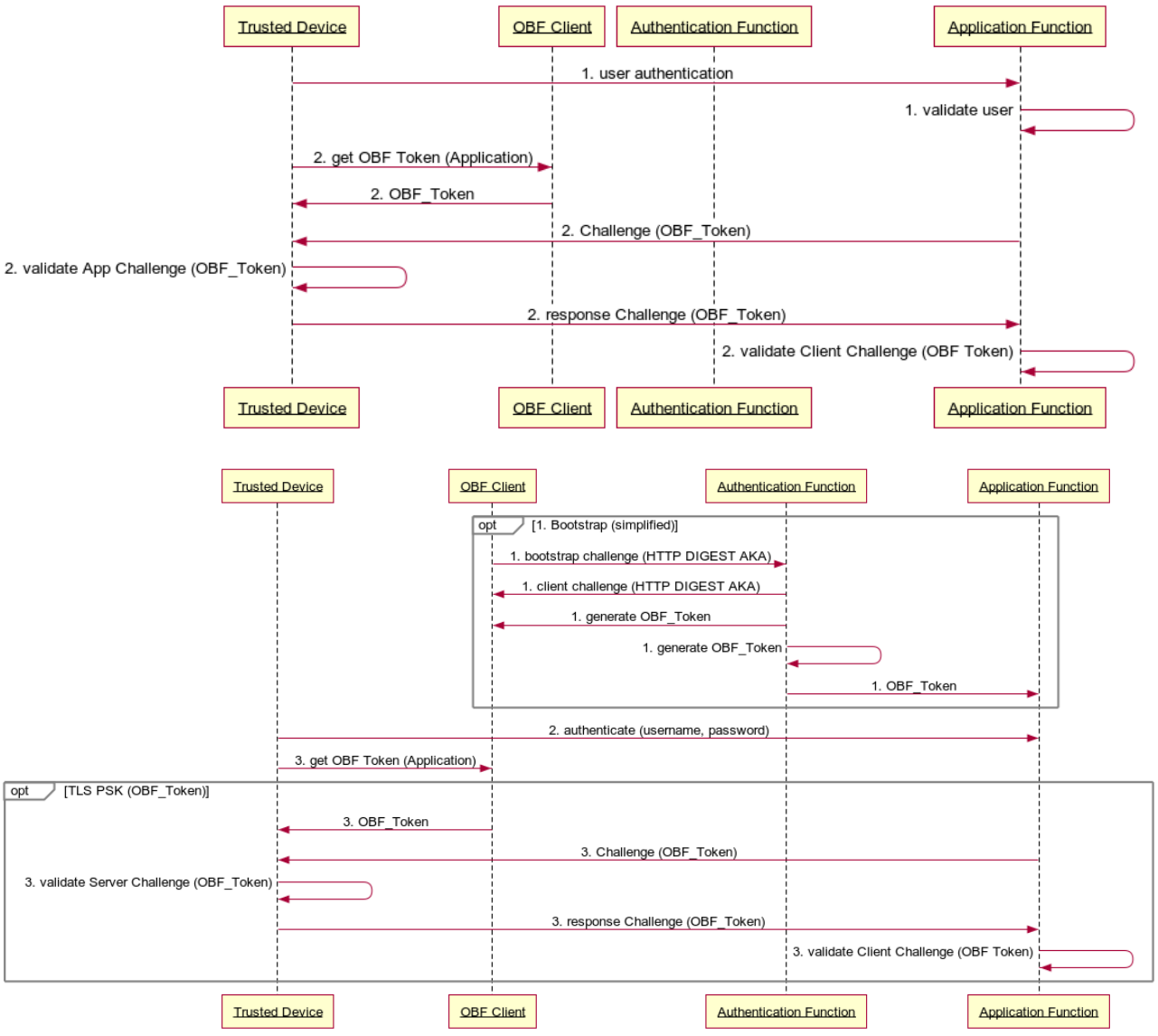


Figure ~~12~~1110-~~21~~: Authentication workflow

~~1211.2 Bootstrapping with symmetric keys~~

~~The symmetric (pre-shared) keys exist on both the secure element and in the authorization function. These are used to authenticate the OBF client function with the authentication function. Session keys are used for securing the communication between the trusted device and an application. This process in accomplished in the following steps:~~

~~1. The authentication function will validate the OBF client function in the bootstrapping stage;~~

~~2. The authentication function and the OBF client function will mutually challenge each other to validate credentials;~~

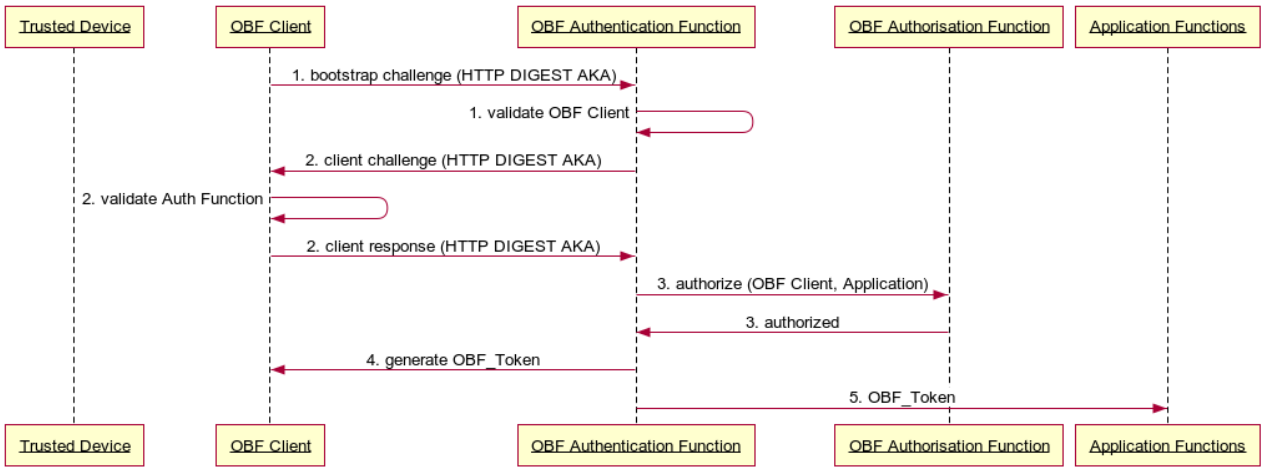
~~3. The authorization function validates if the user has the right to use the authentication for the given application;~~

~~4. When the mutual authentication has completed the OBF client function and authentication function agree on the OBF_Token; and~~

~~5. The OBF_Token is provided to the application for use in subsequent security associations.~~

~~Note: The steps 1, 2, 3 are a part of the digest access authentication AKA.~~

~~The bootstrapping and the session key management process is described in the diagram below (Figure 12-2):~~



~~Figure 1211-2: Bootstrapping workflow~~

~~NOTE: The workflow for bootstrapping using asymmetric keys is similar, with the exception that in place of pre-shared keys the public keys are used for authentication.~~

~~1211~~10.3 Changing of authentication provider flow (symmetric keys)

The user of the service has to approach the new service provider for enabling the use of the authentication services. The Steps for such a transfer are described below:

1. User requests new authentication services provider for its services;
2. The new authentication service provider requests existing authentication service provider for user's shared keys;
3. The new authentication services provider uses the old key to update the secure element with a new key following the machine KYC;
4. The new authentication services provider informs the user and the old authentication services provider of the successful confirmation of the transfer to the new authentication services provider;
5. Upon successful confirmation of the transfer the new authentication services provider informs the application services providers about the change in the OBF_Token for a user;
6. The application service provider uses the new OBF_Token along with embedded connectivity identity to verify the user.

The process is described in the diagram below (Figure ~~11~~10-3):

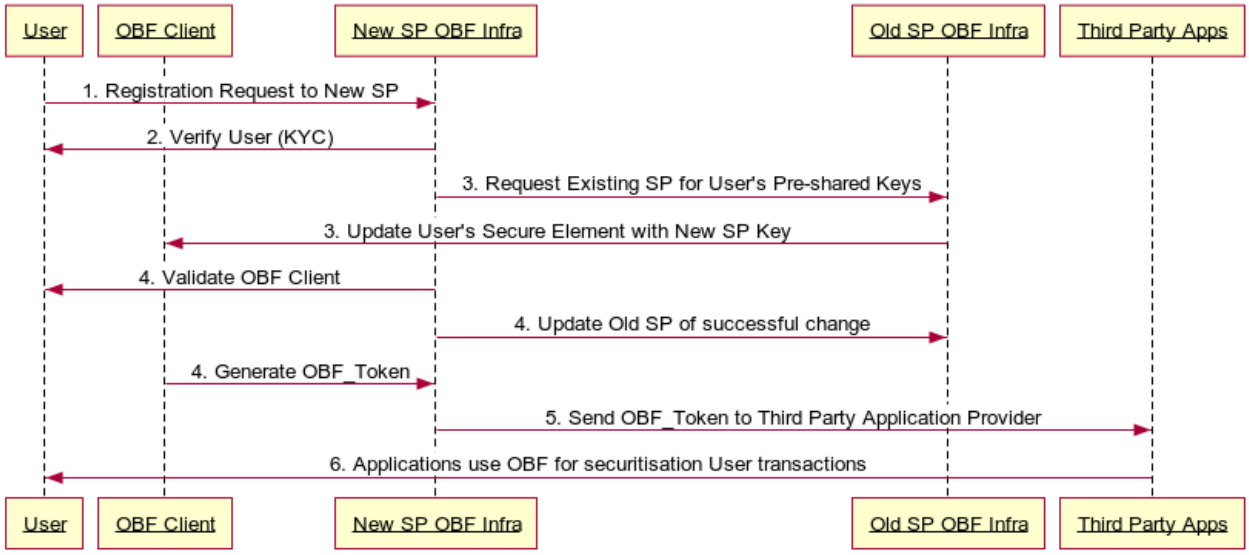


Figure ~~11~~10-3: Authentication provider change (symmetric keys)

~~11~~10.4 Changing of authentication provider flow (asymmetric keys)

~~**Changing of authentication provider flow (asymmetric keys)**~~

A user may change the connectivity provider, but still may want to continue the use of services, which are supported by the OBF authentication function. The authentication provider may be changed as per the mechanism defined below:

1. User requests new authentication services provider for its services;
2. The new authentication services provider completes the machine KYC;
3. The new authentication service provider provides its public key to the old authentication service provider with a request to transfer the user's account to the new authentication service provider;

4. The old authentication services provider uses its private key to update the secure element of the user with the public key of the new authentication services provider;
5. Upon successful confirmation of the transfer the new authentication services provider informs the application services providers about the change in the OBF_Token for a user; and
6. The application service provider uses the new OBF_Token along with embedded connectivity identity to verify the user.

The Process is described in the diagram below (Figure ~~12-11~~10-~~43~~):

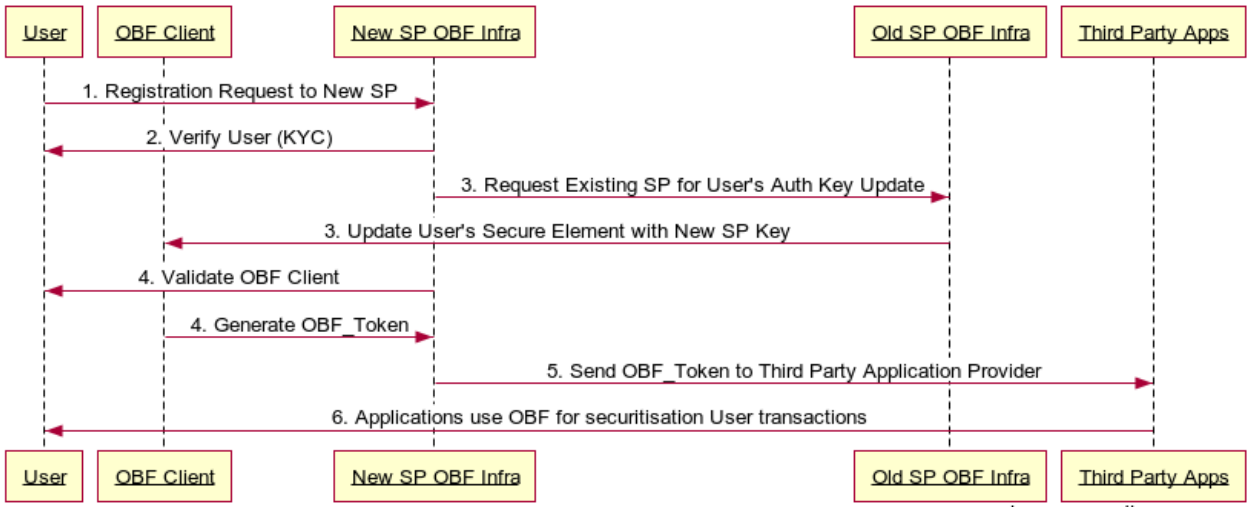


Figure ~~12~~11~~11~~10-~~43~~: Authentication provider change (asymmetric keys)

~~1211.4 Changing of authentication provider flow (symmetric keys)~~

~~The user of the service has to approach the new service provider for enabling the use of the authentication services. The Steps for such a transfer are described below:~~

~~1. User requests new authentication services provider for its services;~~

~~2. The new authentication service provider requests existing authentication service provider for user's shared keys;~~

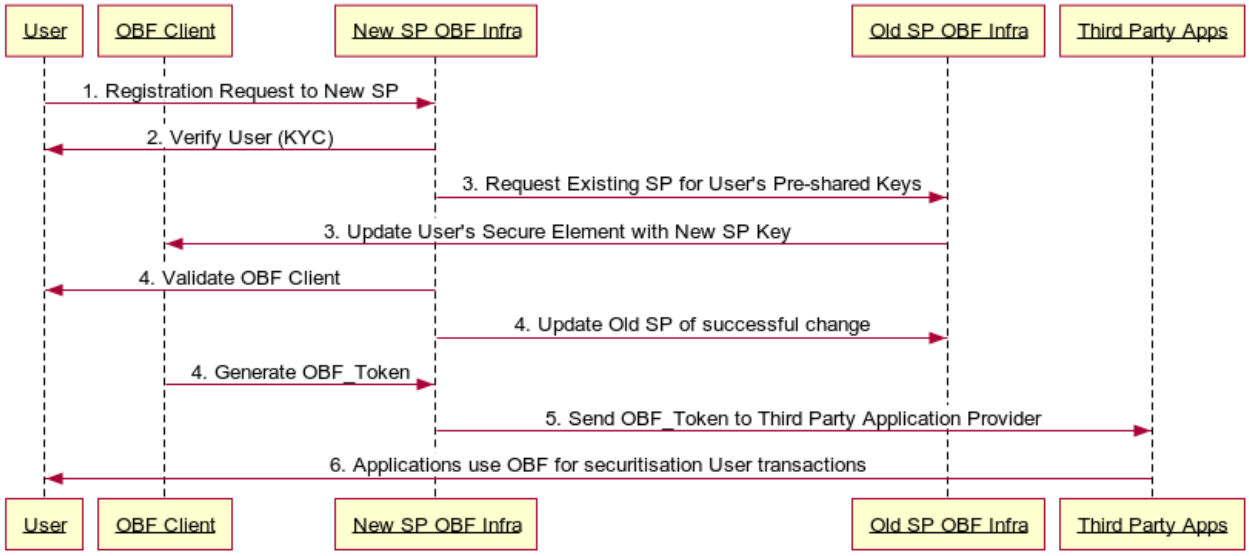
~~3. The new authentication services provider uses the old key to update the secure element with a new key following the machine KYC;~~

~~4. The new authentication services provider informs the user and the old authentication services provider of the successful confirmation of the transfer to the new authentication services provider;~~

~~5. Upon successful confirmation of the transfer the new authentication services provider informs the application services providers about the change in the OBF_Token for a user;~~

~~6. The application service provider uses the new OBF_Token along with embedded connectivity identity to verify the user.~~

~~The process is described in the diagram below (Figure 12-4):~~



~~Figure 1211-4: Authentication provider change (symmetric keys)~~

Appendix I

Explanation of the use case example

(This appendix does not form an integral part of this Recommendation.)

This appendix provides explanation of the use case examples of OBF. In this use case, the background, the device functions and the sample data flow has been described.

I.1 Background and diversified multi-stakeholder eco system

The Ecosystem comprises of the following Actors

a. Network Operator or IoT SP: Supplier of the SIM and Secure Element
b. Device Manufacturer – manufacturer of the Device with the embedded SIM / Secure Element
c. Vehicle Manufacturer – manufactures of the vehicle with the embedded device, SIM and Secure Element
d. Buyer – the entity or person that pays for the Vehicle
e. Application Provider – the entity that provides the Application for registration, tracking and transfer of the vehicle
f. Certifying Agency – the entity that Certifies the Device and the Application
g. Trust Centre – the Agency responsible for the registration and enforcement of Vehicle rules, typically a State actor

I.1.1 Background

Indian automotive standard body has laid down a Standard (Automotive Indian Standard AIS140) for the registration and tracking of public service vehicles, including the communication between Vehicle Tracking Device (VTS) and a Vehicle Tracking and Alarms Management Server (VTAMS)

As per this standard, the VTS device sends various data packets to the VTAMS server like Position-Velocity-Time Data, Panic Alarm, Safety Alerts, Health Data, Diagnostics etc. VTAM Server controls the devices by sending various commands to VTS device; like get device diagnosis, configuration command, Panic Alarm Acknowledgement, Panic Alarm Closure etc. Communication from device to server and server to device is taking place over SMS and TCP/IP channel.

Given the mission critical nature of the service, the VTAMS server is having mechanisms to establish the Integrity, Identity, Authenticity and Trust to ensure the secure and trustful implementation of public safety for the citizens.

I.1.2 Diversified multi-stakeholder eco system

In continuation of background, it is also important to describe the diversified eco system which will enable the AIS140 standard in India.

1. There are more than 40 VTS device manufacturer who are supplying the VTS devices for Public Transport Vehicles

2. Few device manufacturers are designing and manufacturing the devices from ground up and few are assembling the devices and controlling the firmware only. Many devices are constrained devices and are designed for specific purpose only.
3. There are 4 major MNOs (Network Operators) providing the communication channel.
4. There are multiple IoT SPs, providing the end to end services
5. There are multiple SIM manufacturer, supplying the SIM cards to IoT SP or OEM directly
6. There are more than 30 States that will implement their own Application Servers at the State Data Centres
7. There are dozens of Application Service Providers who will license the Tracking and Alarms Management Systems to individual States

I.2 Use case

This use case is for Remote Manageable basic vehicle tracking devices (without crypto functionality) with embedded SIM (secure element). In this use case, device is sending health, diagnosis and other data to national backend system (Application Server). Device is also receiving configuration change command (like application server IP change) from National Backend System (Application Server).

When device is sending data to National Backend System (Application Server), then:

1. Application server is able to identify the device correctly
2. Application server is able to check the data integrity which means no one in between have changed the data
3. Application server is be able to identify replay attack from a malicious entity
4. No one in between device and application server should be able to read the data being sent by device

Similarly, when National Backend System (Application Server) is sending command, like application server address change, to device:

1. Device is able to identify that this request is coming from authorized application server
2. Device is able to check the data integrity which means no one in between have changed the data
3. Device is able to identify replay attack from a malicious entity
4. No one in between application server and device should be able to read the data being sent

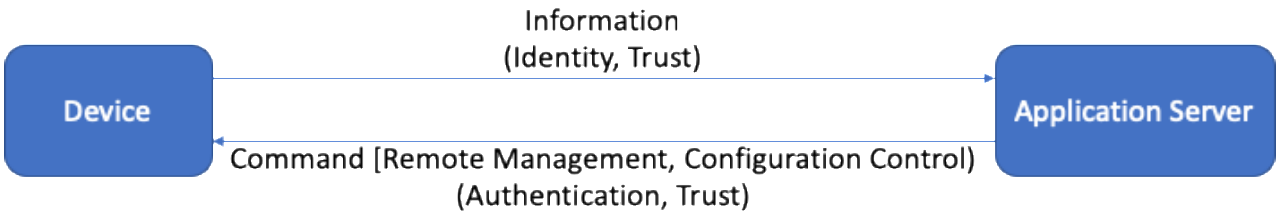


Figure I.1: Device-Application Server Communication

I.2.1 Important consideration for security

Following are important consideration for security implementation:

1. The tamper proof identity of the SIM / Secure Element (IccID / EID) is used as the primary identifier for the connected device
2. Appropriate mechanisms are followed for the generation and sharing of Security key between the SIM / Secure Element and the Authentication Server

3. The Network Application Function (NAF) and the OBF interact securely following the standards prescribed by 3GPP GAA.

I.2.2 Functions required

Following functions are required on device, secure element and application server to meet the mentioned security requirement ″see clause I.2.1″:

I.2.2.1 Device functions

(a) Validate checksum function

This function is used by device to validate the checksum of the incoming data. This will ensure the **Data Integrity**. If checksum is not matched, then device will not process the data further and ignore it.

(b) Decrypt encrypted server data function

When Device receives data from an application server (like configuration change command), it will first establish the data integrity. Once the data integrity is established, the device will send the data to Secure Element for decryption.

The purpose of the function is to authenticate the Application Server to the Device and protect the communication from man in the middle / replay attacks.

(c) Encrypted device data function

This function is used by Device when device is sending any data [like Health Packet or Diagnosis Data or PVT (Position, Velocity, Time) data] to an Application Server.

I.2.2.2 Secure element functions

(a) Decrypt data function

This function is called by device and responded by the Secure Element with the result that the Secure Element decrypts the Server Encrypted Data by the use of a key from a specified key index.

(b) Encrypt device data function

This function is called by device and responded by the Secure Element with the result that the Secure Element encrypts the Device Data by the use of a key from a specified key index.

I.2.2.3 Application server functions

(a) Key import function

This function is used by Application Server to import encryption/decryption keys for the SE (Secure Element) from a trusted source. Establishing trusted source is out of scope of this explanation.

(b) Decrypt device data function

This is function is used by Application Server to request the decryption of incoming data from the device. Application server establishes ′Identity′ and ′Authenticity′ of the incoming Device Data request using this function.

(c) Encrypt server data function

This is function is used by Application Server to request the encryption of data intended to be sent to a device (e.g. a command, like configuration change). When called, this function adds TRUST data which is used by device to establish mutual authentication with the server.

I.2.3 Application Server to Device flow (Sample)

Following is a sample data flow for ′Command (Remote Management, Configuration Control)′ sent from Application Server to Device.

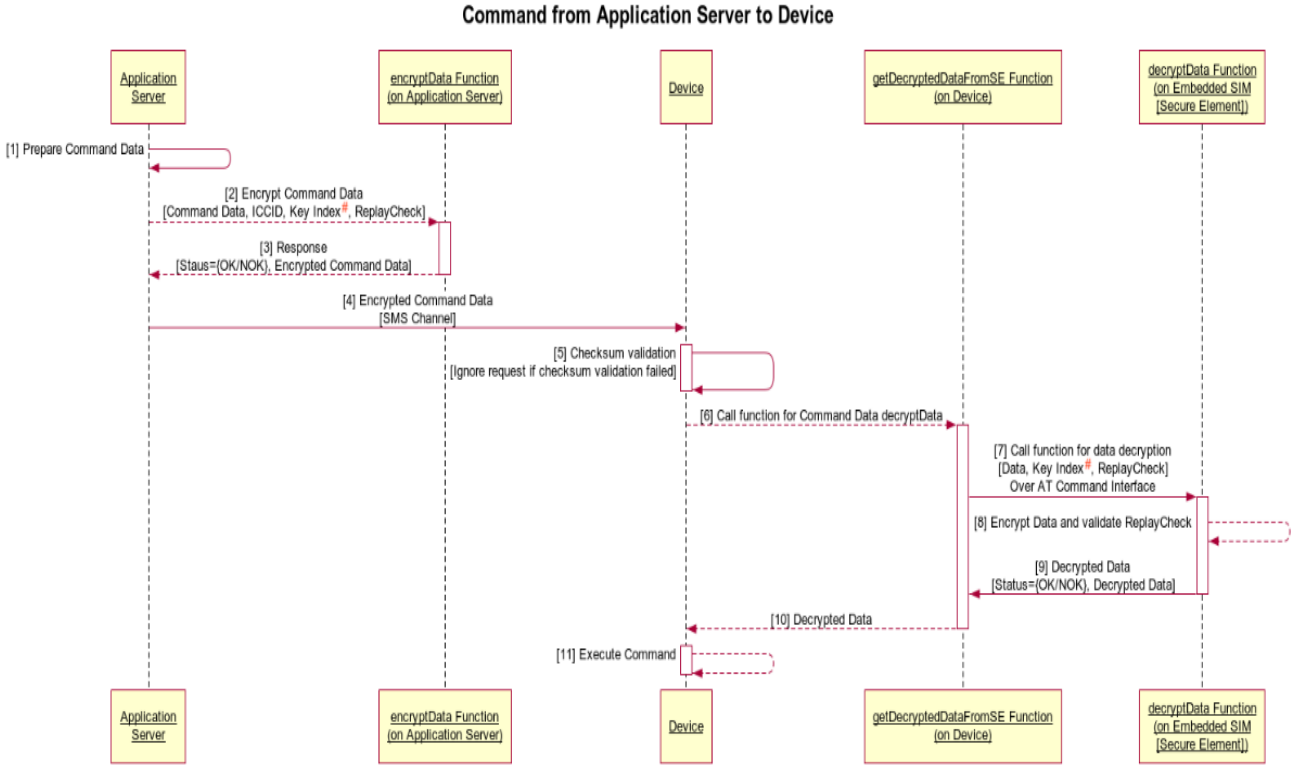


Figure I.2: Application server to Device communication flow

NOTE 1 – # In future, one-time session key, shared using public/private key and crypto challenge could be used instead of fixed keys

Bibliography

[b-RFC 6733] IETF, Request for Comments: 6733 (October 2012), *Diameter Base Protocol*

[b-RFC 7155] IETF, Request for Comments: 7155 (April 2014), *Diameter Network Access Server Application*

[b-RFC 7616] IETF, Request for Comments: 7616 (September 2015), *HTTP Digest Access Authentication*.

[b-3GPP TS 33.220] 3GPP TS 33.220 V16.0.0 (2019-09), *Generic Authentication Architecture (GAA); Generic Bootstrapping Architecture (Release 16).* <https://www.3gpp.org/ftp/Specs/archive/33_series/33.220/>

_____