INTERNATIONAL TELECOMMUNICATION UNION

TELECOMMUNICATION STANDARDIZATION SECTOR

STUDY PERIOD 2017-2020

SG13-TD444/WP3

STUDY GROUP 13

Original: English

Question(s): 16/13 Virtual, 20-31 July 2020

TD

Source: Editors

Title: Draft new Recommendation ITU-T Y.OBF_Trust: “Open Bootstrap Framework enabling trustworthy networking and services for distributed diverse ecosystem” (output of e-meeting, 29 June 2020)

Purpose: Information

Contact: Abhay Shanker Verma
Telecom Engineering Centre (TEC)
India
Tel: + 91 9999554900
E-mail: as.verma@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

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

Abstract: This document contains the updated draft Recommendation ITU-T Y.OBF_Trust “Open Bootstrap Framework enabling trustworthy networking and services for distributed diverse ecosystems” based on the discussion at interim e-meeting of Q16/13 on 29 June 2020.

This document is the revised baseline text of draft Recommendation ITU-T Y.OBF_Trust: “Open Bootstrap Framework enabling trustworthy networking and services for distributed diverse ecosystems”. This document is based on the following contribution.

Base document: TD431/WP3

C148	TEC, India	Y.OBF_Trust: Proposal to make mainly modifications in the requirements clause by including the pre-requisites clause, rewriting the reference model clause, reorganising the OBF Functional architecture clause and changes in the information workflow, in order to make it more readable.	Q16/13

- Proposal of contribution

Based on the latest output document (TD431/WP3) of draft Recommendation Y.OBF_Trust (“Open Bootstrap Framework enabling trusted devices, applications and services for distributed diverse ecosystems”), this contribution proposes to update the draft Recommendation ITU-T Y.OBF_Trust in accordance with the changes carried out in track mode.

- Meeting result
  - The proposed modifications were discussed in detail and after the drafting changes the contribution C148 has been accepted.

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 recommendation includes the OBF concept, the requirements of the OBF as well as the pre-requisites for the devices and the application. It also includes a~~The OBF is a trust framework described by OBF elements are described in a~~ reference model describing the OBF elements and a functional architecture describing four functional groups, four reference points and security parameters. ~~. The details of the functional groups and the specifications of the reference points are included and a functional architecture having OBF client function, OBF authentication function, OBF authorization function, OBF application function, four reference points and OBF security parameters. The recommendation includes requirements of the OBF, and~~ The information workflows for the bootstrapping, authentication ~~provisioning of keying material that mutually authenticates trusted devices, applications and service providers. A mechanism for users~~and ~~to~~ change ~~the service providers~~of OBF realm is also provided.

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

Keywords

Bootstrapping; IoT service provider; OBF; OBF_Token; Open Bootstrap Framework; Trust Framework

Contents

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 proposes an Open Bootstrap Framework (OBF) for secure provisioning of trusted services by Application Services Providers (ASPs) that have no existing trust relationship with the users. OBF can be deployed by the network operators or IoT service providers to enable ~~for~~ authentication and authorization of ~~trusted~~ devices for access to~~,~~ trusted services provisioned by ASPs.~~, service providers and applications.~~

The scope of this Recommendation includes

- OBF concept;
- requirements for the OBF and OBF elements;
- OBF reference model;
- OBF functional architecture; and
- information workflows of the OBF.

The recommendation offers a framework for the provisioning of trusted ASP services to the subscribers of network operators who deploy the OBF, by the use of the underlying secure elements and bootstrapping mechanisms.

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 cryptographic process of binding the user's identity to the keying material provisioned in the secure element of the user's device. ~~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 also 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 of the communication of the reference point the between device and applicationRPDS. 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)]~~

~~NOTE – 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.~~

~~**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.~~3.2.2. **open bootstrap framework (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	3^{rd} Generation Partnership Project
AKA	Authentication and Key Agreement
API	Application Programming Interface
~~COAP~~	~~Constrained Object Authentication Protocol~~
FQDN	Fully Qualified Domain Name
GBA	Generic Bootstrapping Architecture
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~~
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
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

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

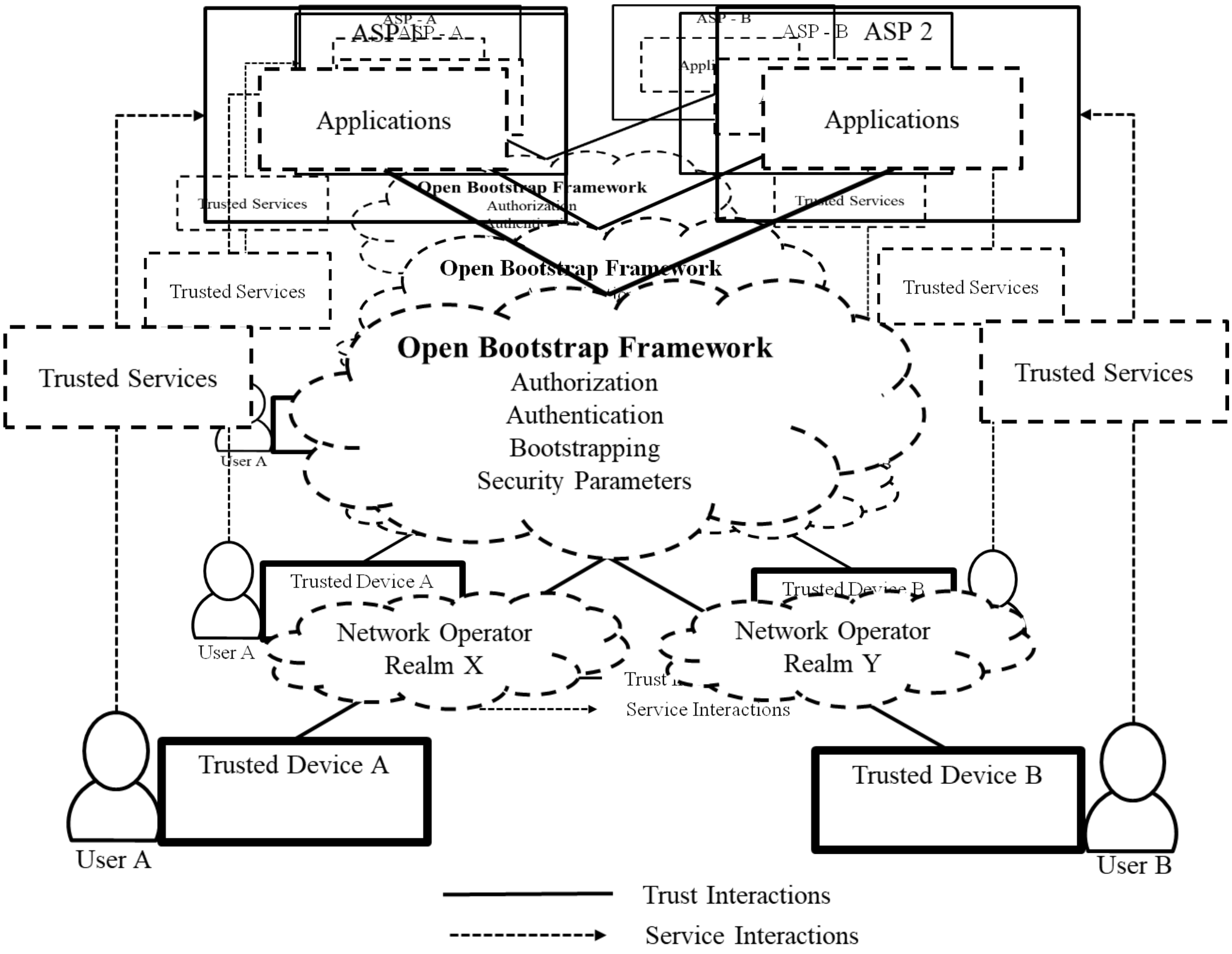


Figure 6-1: OBF concept

The OBF is a set of requirements, 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. An implementation of the OBF is referred to as an OBF realm. Further, any user of a bootstrapped device can access the applications and services of an~~y~~ ASP by using the security capabilities of the OBF~~ functions and mechanisms~~.

~~The~~ An OBF realm 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.
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~~ deploying an OBF realm~~appropriate security functions, information flows and mechanisms~~.

3. **Application 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 network operators and the OBF elements that have been deployed;
- identify and onboard ASPs whose applications require to be protected from unauthorized usage;
- identify trusted devices that are authenticated by a network operator;
- expose the inherent security capabilities of any underlying network technology for the benefit of ASPs;
- enable applications to establish secure association with trusted devices;
- identify and address the clients and the applications by using the identifiers of the underlying Information and Communication Technology (ICT) layers;
- be accessible over the public Internet;
- support industry standard protocols for key management;
- support industry standard authentication and authorization protocols;
- support 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.

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

7.2 Pre-requisites for the trusted devices

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

- have an implementation of secure clients ~~ost the OBF client function~~ in the device or its connectivity element (e.g. SIM card);
- have configurations that make the device ~~be~~ OBF aware, and initiate the bootstrapping process, when the OBF application requires it;
- support the application specific protocol over the reference point between the device and the application such as HTTP, Message Queue Telemetry Transport (MQTT), Web Sockets or Constrained Object Authentication Protocol (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 OBF realm~~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.

It is recommended that the ~~The~~ trusted devices ~~are recommended~~ have the capability to configure the lifetime and check the validity before using the keying material.

~~to configure the key lifetime and validity settings.~~

7.3 Pre-requisites for the 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~~The OBF~~, the~~ applications are required to:

- be OBF aware, and be able to indicate to the device the protocol and ~~need for a~~ keying material ~~if it attempts~~required to connect to the application~~n without one~~;
- implement Diameter / HTTP proxy functionality to act as a proxy towards the OBF realm in which the user is bootstrapped;~~ and~~
- be able to locate the user's OBF realm and communicate securely with ~~the user's~~ the OBF ~~authentication~~ functions;
- ~~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 OBF realm~~authorization function~~; and
- implement the security parameters in its security protocol used for creating secure associations between the device and the application~~ via the authentication function~~.
- ~~The trusted devices are recommended to configure the key lifetime and validity settings.~~

8 OBF reference model

The OBF reference model describes the key elements and the reference points 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 OBF reference model is shown in the diagram below:

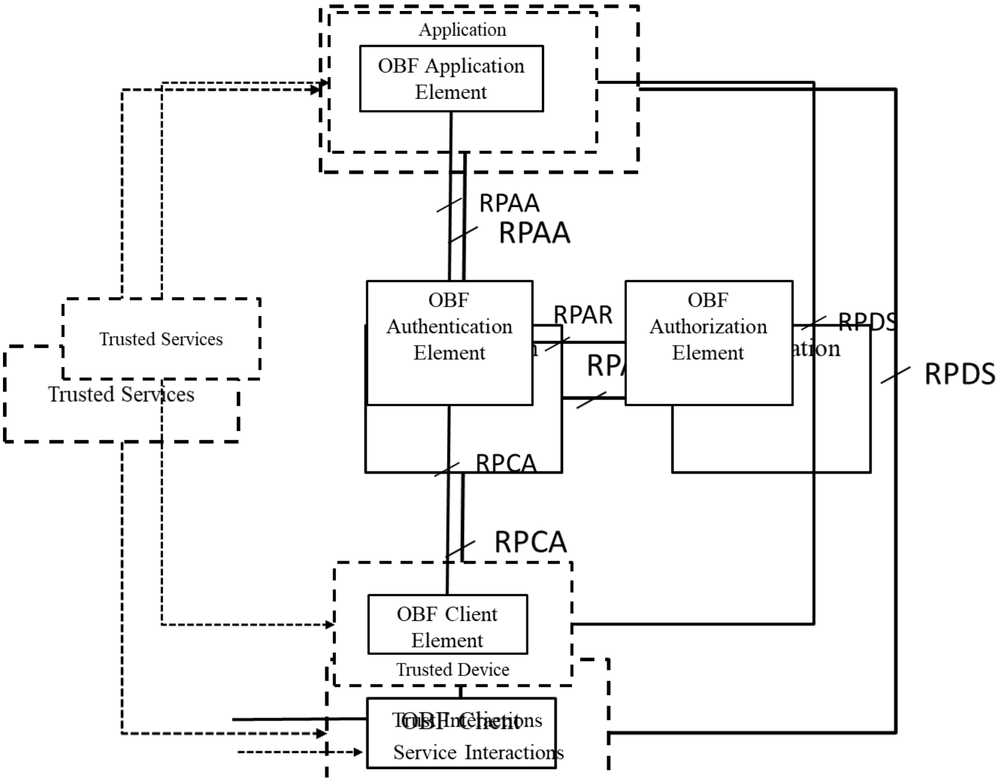


Figure 8-1: OBF reference model

Figure 8-1: OBF reference model

8.1 OBF e~~e~~lements

The OBF Elements enable two types of interaction between the device and the application. The trust interactions establish the required security between the user of the connected device and the application. The Service interactions allow the user to benefit from the use the application which required the secure association.

The elements of the OBF enable theses interactions, each of which is described below.

8.1.1 OBF c~~c~~lient element

The OBF client element 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~~. ~~The OBF client has the features and functions required for the interaction with the authentication,~~

Figure 8-1: OBF reference model

~~authorization and the application.~~ The OBF client is specified and provisioned by the network operator that is providing the OBF realm and the associated trust services.

8.1.2 OBF a~~a~~uthorization element

The OBF authorization element ~~is a node provisioned by the network operator that~~ carries out the key management and provides the keying material as per standard ~~AKA~~ security protocols.~~ The authorization 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.~~

8.1.3 OBF a~~a~~uthentication element

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

~~The authentication generates the OBF_Token, and shares it with the authorized ASPs.~~

8.1.4 OBF a~~A~~pplication element

The OBF Application element ~~is a node, which receives OBF_Token from the authentication upon successful bootstrap. It stores the OBF_Token and uses the same in~~ sets~~ting~~ up the secure connections between the ~~application client~~device and the applications using the security enablement from the other OBF elements.

8.2 OBF r~~r~~eference p~~p~~oints

The OBF specifies four reference points, namely, RPAA - the reference point between the authentication function and the application, RPAR - the reference point between OBF authentication function and the OBF authorization function, RPCA - the reference point between the OBF client function hosted in the trusted device and the OBF authentication function, and RPDS - the reference point between the trusted device and the application.

~~The OBF specifies four reference points as below:~~

~~8.2.1 RPAA~~

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

~~8.2.2 RPAR~~

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

~~8.2.3 RPCA~~

~~RPCA is the reference point between the OBF client function hosted in the trusted device and the OBF authentication function.~~

~~8.2.4 RPDS~~

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

9 OBF functional architecture

~~Based upon the OBF reference model, and the detailed analysis of the requirements, a~~The functional architecture diagram below describes the functionalities of the OBF. The OBF Elements are further detailed into various functions along with the specifications of the reference points. ~~for the OBF is~~

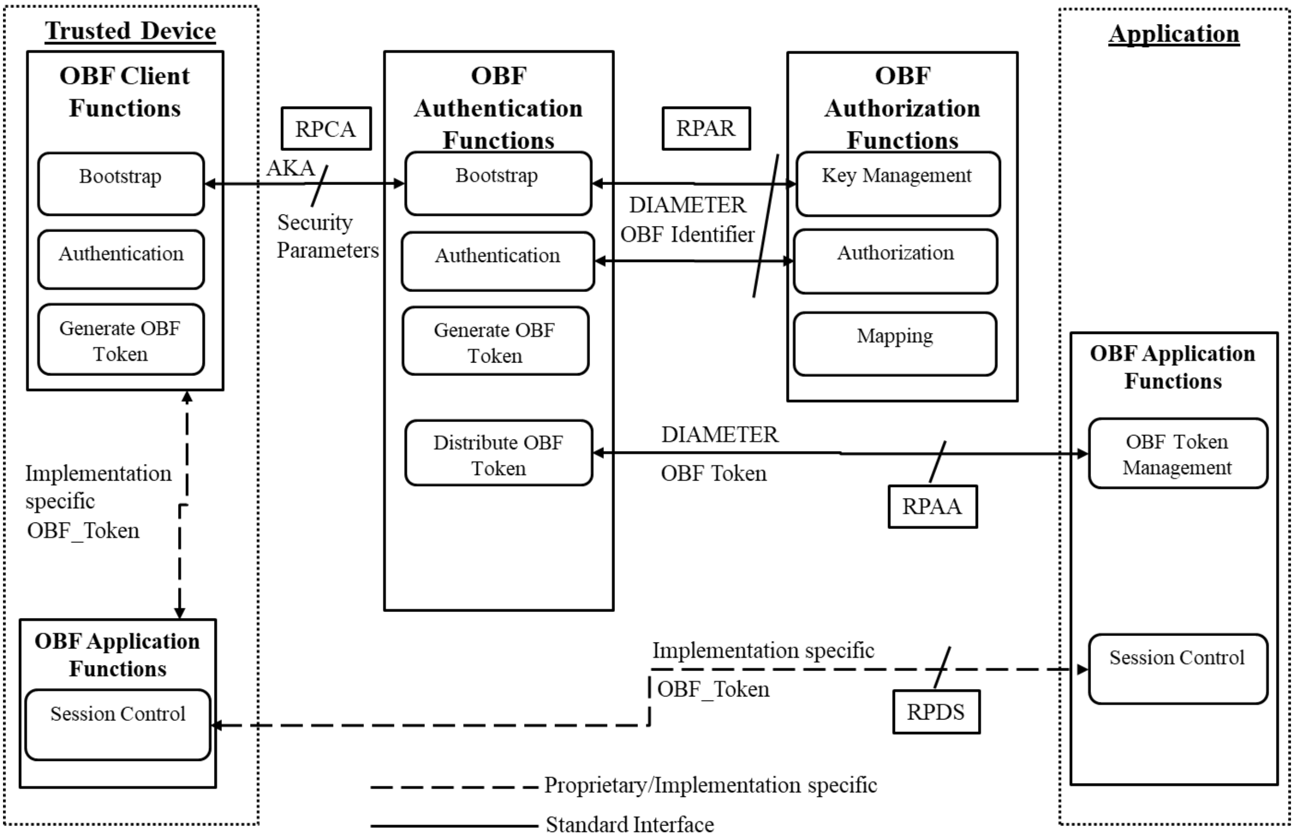
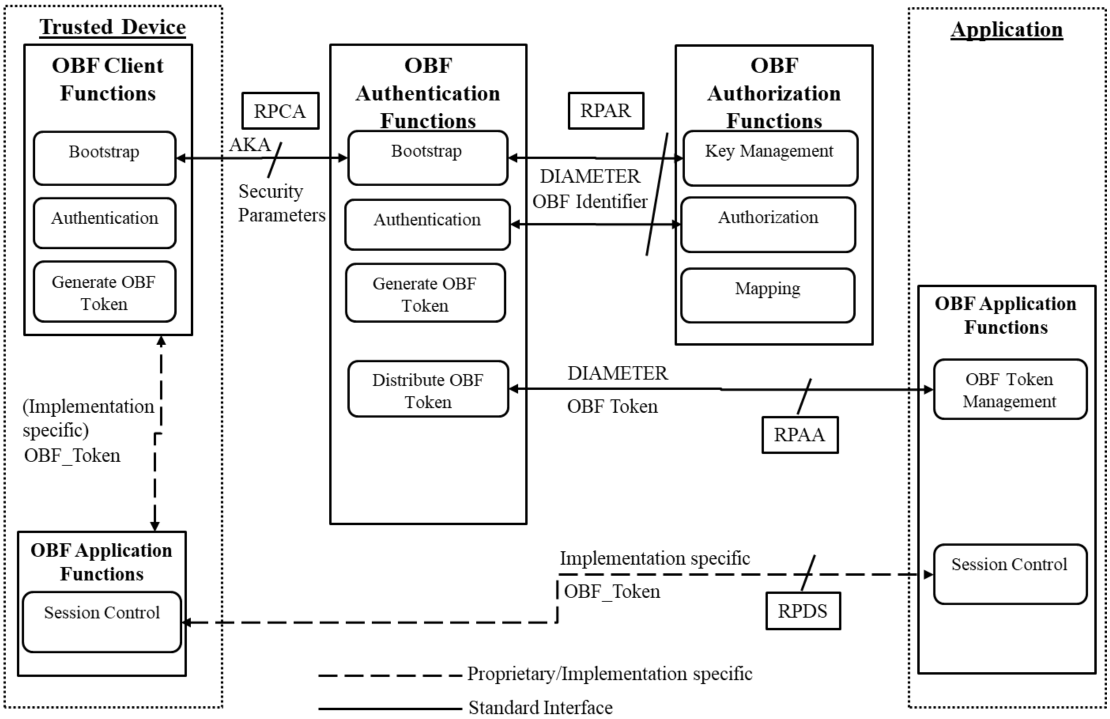
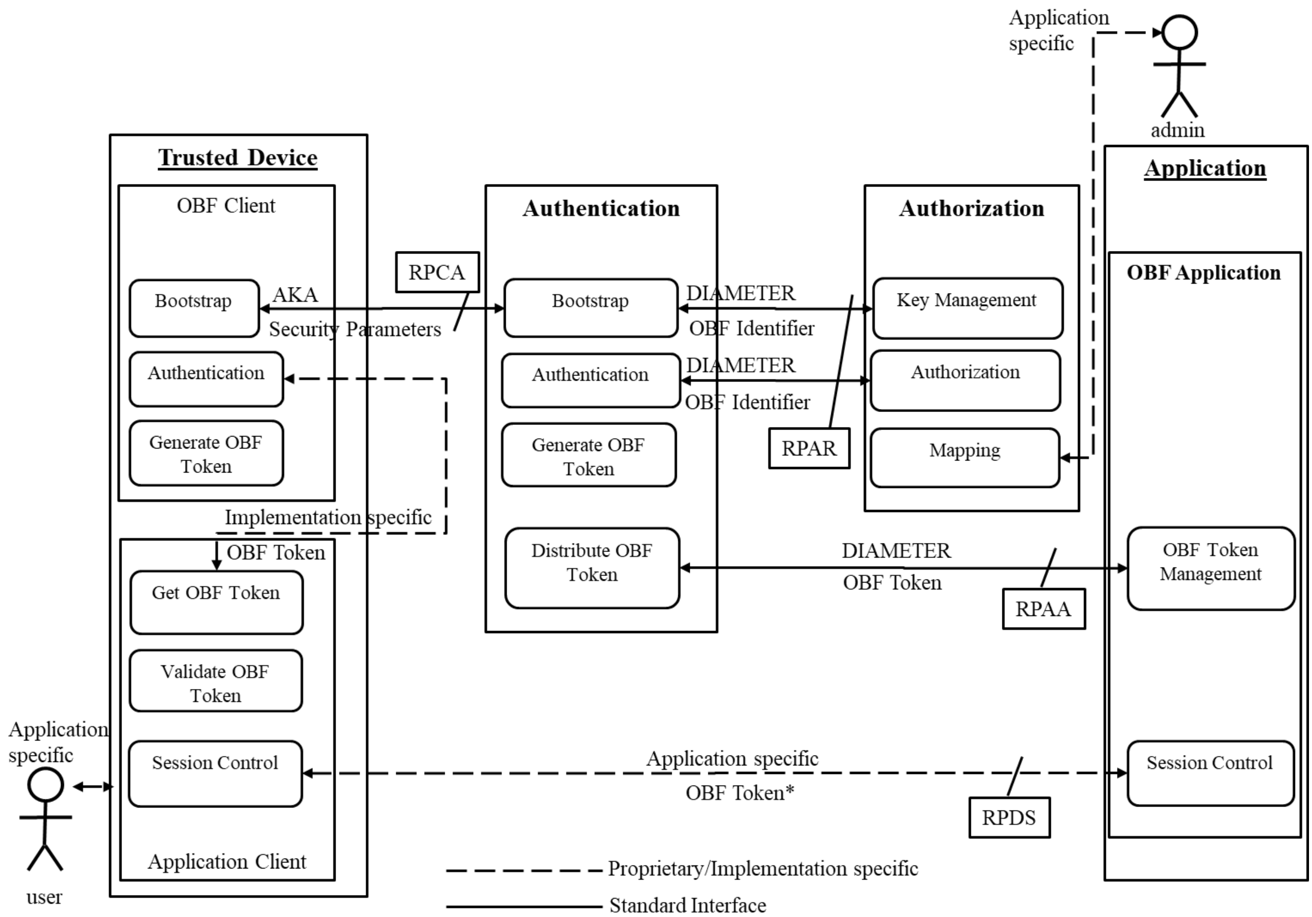


Figure 9-1: OBF functional architecture

~~presented in the diagram below:~~





The functional architecture ~~presents~~consists of the following:~~:~~

- the OBF functions;

- the reference points; and
- the security parameters used within the OBF elements.
- The functions, reference points and security parameters are described below.

9.1 OBF functions

NOTE - ~~ote: ~~When the OBF is deployed in the network~~ of a network operator~~, thate context is referred to as a realm. The instantiated functions within the realm are referred as nodes. As an example, an ~~a~~Auth~~oris~~entication function, when instantiated in the network~~ of a Network Operator~~, ~~is referred as~~will be called the authentication node in the realm of that OBF deployment.~~e operator.~~

The following functionalities are supported by a~~A~~ll the OBF ~~nodes~~functions:

- ~~implement ~~the OBF numbering;
- identification~~y~~ and authentication~~e~~ of each other within the OBF realm(s);
- identification~~y~~ and authentication~~e~~ of OBF clients; and
- ~~support ~~transferability ~~such that a user is free to choose services from~~between~~ any~~ network operators~~ operator or ASP~~.

Each of the functions ~~The OBF functions / sub-functions ~~are described below.

9.1.1 OBF authentication functions

The OBF Authentication functions are a group of four functions that enable the bootstrapping of the trusted device. Each of the functions are described below.

(a) ~~9.1.1~~ Authentication function

This~~e~~ function mutually authenticates the OBF client and the authentication ~~function~~node, 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.

~~OBF ~~The authentication function provides the following functionalities:

- ~~o~~ maintains the list of users, authorized applications and the related subscription parameters;
- ~~o~~ protects the use of the network subscriber identity against discovery and misuse;
- ~~o~~ supports AKA protocols such that it can support the one used by the underlying network technology layer;
- ~~o~~ manages the lifecycle of keys as per the agreed AKA protocol;
- ~~o~~ configures and communicates the format of the OBF identifier to the OBF ~~OBF ~~client functions ~~function~~; and
- ~~o~~ configures the OBF ~~subscription information~~security parameters in conjunction with the OBF authorization functions and communicates that to the OBF client functions~~ function~~.

(b) ~~9.1.2~~ Bootstrapping function

This function~~, hosted in the authentication function~~ ~~as well as in the OBF client function, creates~~ provides the functionality for a~~a~~ new registration ~~for ~~of ~~the ~~a trusted device by way of establishment of new long-term secret key(s) for secure communication.

(c) ~~9.1.3~~ Generate OBF_Token function

This function ~~is responsible to~~ generates the OBF_Token, after bootstrapping has successfully been completed, by using the agreed ~~keying material~~OBF security parameters~~ 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~~is specific to the subscription information and the application for which it is generated~~ specific to an application, and its applicability is limited to a specific application~~.

NOTE – The lifetime of the OBF_Token may vary significantly across various use cases. When the application client function is invoked, or required to initiate the interaction with the application, the OBF_Token may be 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 generate OBF_Token function to obtain a new OBF_Token.

(d) ~~9.1.4~~ 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.

9.1.2 OBF authorization functions

The OBF Authorization functions are a group of three functions that work together to ensure that applications can be mapped to devices and the security parameters can be agreed between devices and the applications.

The authorization functions are the repository of the UIDs of ASPs that are authorized to provide services. It holds the mapping information 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.

(a) ~~9.1.5~~ Key management function

~~The~~ This 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.

(b) ~~9.1.6~~ Authorization function

This function validates if the device can access the application based on the OBF_Token ~~client function has the right to use the~~ sent in the authentication request~~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 functions of a device.

The ~~OBF~~ authorization function provides the following functionalities:

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

(c) ~~9.1.7~~ 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 mapping function provides the following functionalities~~enables~~:

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

9.1.3~~1.8~~ OBF ~~client~~ application functions

The OBF Application Functions are deployed in the device of the user and the applications of the ASP. This group of functions enable the session security between the device and the application, each of which is described below.

(a) Session control function

This function is application specific. It utilizes the OBF_Token to initiate and maintain a secure session towards the application. The function is implemented within an industry standard session control such as TLS, PSK-TLS, Kerberos, IPSec.

(b) OBF_Token management function

The OBF_Token management function receives and stores the OBF_Token within the Application for securing the future sessions between the device and the application.

9.1.4 OBF client functions

The three OBF client functions, namely bootstrapping function, authentication function and OBF_Token generation function, correspond to the OBF Authentication Functions with the same functionality as described in ~~clause~~section 9.1.1.

Together, the three functions enable the OBF client to:~~:~~

—

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

~~o~~- support~~s~~ the application protocol in the reference point RPDS and initiate the bootstrapping process if indicated by the application.

~~9.1.9 Validate OBF_Token function~~

~~**This function validates the lifetime of the OBF_Token. When the application client function is started, or required to initiate the interaction with the application, the 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.**~~

~~9.1.10 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.**~~

~~9.1.11 Session control function~~

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

~~9.1.12 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.**~~

9.2 Specifications of OBF reference points

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

9.2.1 ~~Specifications of Reference Point~~ RPAA

The reference point 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 provides the following functionalities:

- 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 ~~Specifications of Reference Point~~ RPAR

The reference point RPAR ~~is the reference point between OBF authentication function and the OBF authorization function. The OBF authentication function uses the RPAR to obtain~~provides 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 provides the following functionalities~~allows~~:

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

9.2.3 ~~Specifications of Reference Point~~ RPCA

The reference point 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 functions to the OBF authentication functions.

- The reference point RPCA provides the following functionalities:

- 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;
- identifies 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 ~~Specifications of Reference Point~~ RPDS

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

The reference point RPDS provides the following functionalities:

- 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.

9.3 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 realm. The purpose of the subscription information is to authenticate and authorize 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.

9.3.1 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 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 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.

9.3.2 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., Mobile Station International Subscriber Directory Number (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.

9.3.3 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 of the device/ user equipment (UE) as well as in the authentication function~~, ,~~ based on an agreed security schema between the device and 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 synchronized 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 characteristics of the OBF_Token are as follows:

-(a) It binds the user identity to the keying material used in the reference points;

-(b) It is the globally unique identifier of realm of the OBF in which it is issued;

-(c) It supports any underlying network technology;

-(d) It identifies the realm of the OBF in which it is issued;

-(e) It serves as a temporary identifier of the user;

-(f) It is a key identifier in protocols used in reference point RPCA and RPDS;

-(g) It enables the application to detect and address the authentication function that has sponsored the OBF_Token; and

-(h) It has a format that is usable by the underlying network technology layer bootstrapping capabilities.

10 Information workflows

This clause specifies important procedures for the trust and service type interactions in accordance with the functional architecture ~~defined~~outlined in the ~~clause~~section 9. Four major flows are described, two for bootstrapping and authentication, and another two for changing the OBF realm whilst using symmetric or asymmetric keys.

The details of the four information workflows are described in the sections below.

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

10.1 Bootstrapping & authentication workflow

The bootstrapping and authentication workflows are meant for bootstrapping a device to the OBF realm, and authorizing it for using a particular trusted application. Two types of information workflows are provided: (i) Bootstrapping workflow, and (ii) Authentication workflow.

10.1.1 Bootstrapping workflow~~Bootstrapping with symmetric keys~~

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

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

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 OBF client functions will send a challenge request to the OBF Authentication functions. The OBF authentication function will validate the credentials of the OBF client based on the

keys/algorithms used in the HTTP Digest/AKA~~The OBF authentication functions will validate the OBF client functions in the bootstrapping stage~~;

2. The OBF authentication function will send back a challenge back to the OBF client functions; The OBF client functions will validate the OBF Authentication functions based on the keys/algorithms used in the HTTP Digest/AKA~~The OBF authentication functions and the OBF client functions will mutually engage in a challenge-response mechanism to validate credentials~~;
3. After the successful mutual authentication in steps 1 and 2, the OBF authentication functions will check if the given device is authorized to use OBF for trusted services for the given application~~The OBF authorization functions validates, if the user has the right to use the bootstrapping for the given application~~;
4. When the authorization has been approved, the OBF client functions and the OBF authentication functions generate an OBF_Token as per the agreed AKA protocol~~When the mutual authentication has been completed, the OBF client functions and the OBF authentication functions 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 - ~~ote:~~ 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 10-1) in which the numbering of the steps in the diagram follows the numbering of steps in the paragraph above:

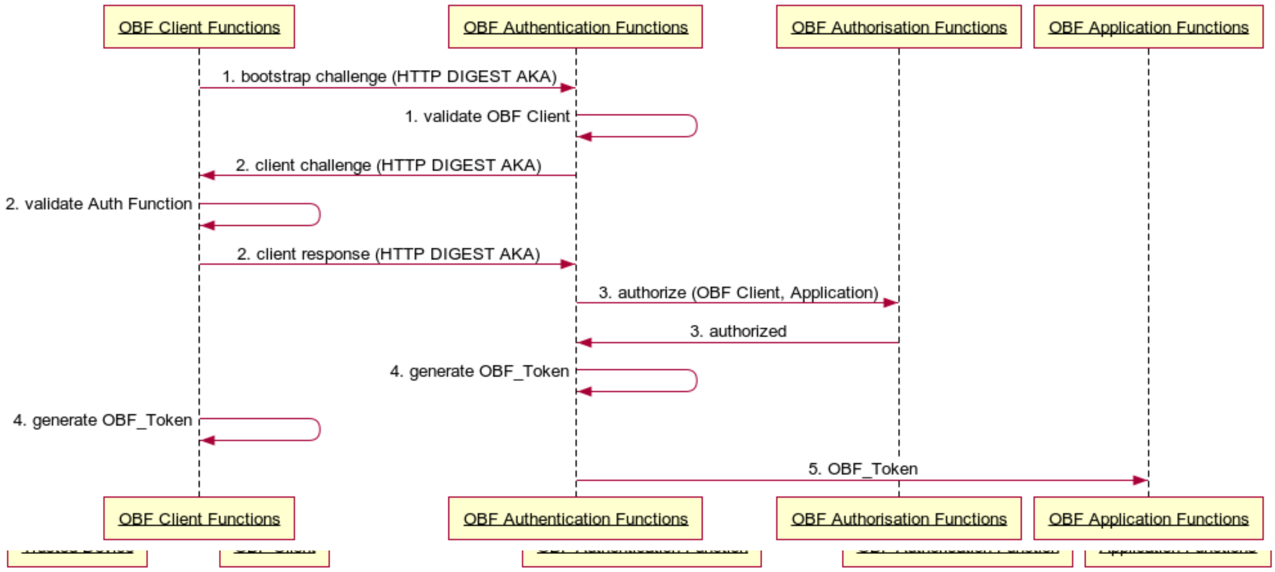

OBF Client Functions
OBF Authentication Functions
OBF Authorisation Functions
OBF Application Functions
1. bootstrap challenge (HTTP DIGEST AKA)
1. validate OBF Client
2. client challenge (HTTP DIGEST AKA)
2. validate Auth Function
2. client response (HTTP DIGEST AKA)
3. authorize (OBF Client, Application)
3. authorized
4. generate OBF_Token
4. generate OBF_Token
5. OBF_Token
OBF Client Functions
OBF Authentication Functions
OBF Authorisation Functions
OBF Application Functions

Figure 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.

10.1.2 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 functions to use the bootstrap framework for authentication. This process is accomplished in the following steps, provided that the bootstrapping has been completed as per 10.1:

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 optionally requests the OBF client functions to get a new OBF_Token if no previous is available, or has expired~~The user request towards the application is executed and the OBF application functions 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
2. The OBF application functions use the OBF_Token to send a challenge to the device. Upon success, the OBF_Token and the session control function are used to secure the data exchange between the device and the application.~~The OBF client functions 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:

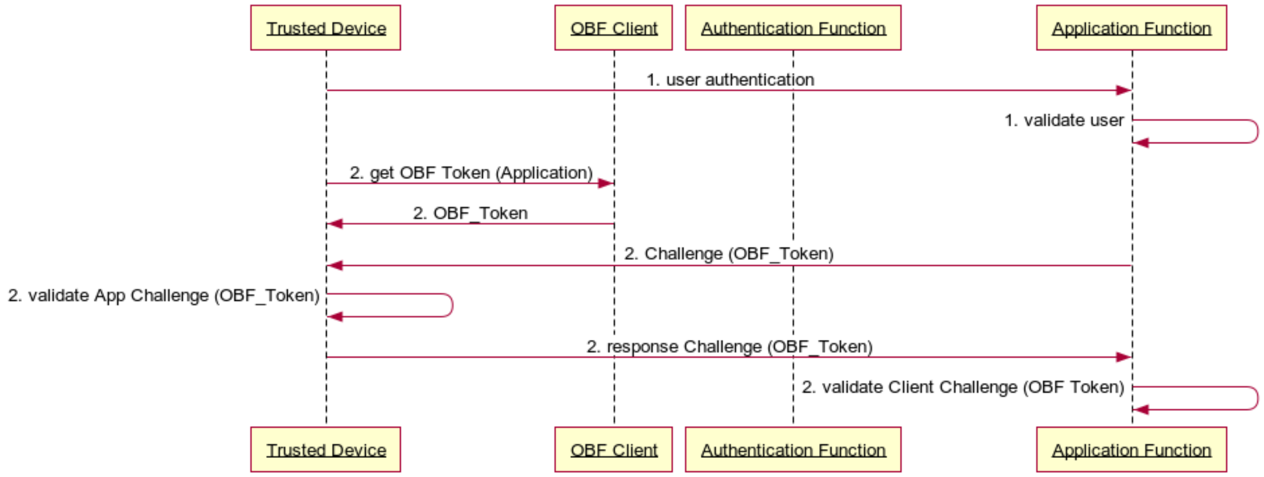


Figure 10-2: Authentication workflow

10.2 Workflow for changes in OBF realm

10.2.1~~3~~ Change of OBF realm ~~ing of authentication provider flow~~ (symmetric keys)

A user that is beneficiary of the OBF enabled trusted services provided by a network operator may require to change the network operator, but still may want to continue the use of trusted services, which were supported by the OBF authentication function.

The changing of the OBF realm is enabled by the OBF mechanism as per the mechanism defined below.

The user of the service has to approach the ~~new service provider~~next network operator or IoT service provider~~-~~, referred as the new OBF realm, for enabling the use of the ~~authentication~~ trusted services for his device.

The steps for such a transfer of realm, in the case when symmetric keys are used for authentication, are described below:

1. User requests n~~n~~ew ~~authentication services provider~~OBF realm for its services;
2. The n~~n~~ew ~~authentication service provider~~OBF realm undertakes the verification of the user and the device (machine KYC) and upon successful verification, requests ~~existing authentication service provider~~the old OBF realm for the user's shared keys;
3. The n~~n~~ew ~~authentication services provider~~OBF realm uses the o~~o~~ld OBF realm's key(s) to update the secure element with the ~~a new~~ key(s) ~~following the machine KYC~~of the new OBF realm;
4. The new OBF realm authenticates the OBF client functions ~~-~~using its keys, and upon success, ~~The new authentication services provider~~ informs the user and the o~~o~~ld ~~authentication services provider~~OBF realm of the successful ~~confirmation of the~~ transfer of the user to the n~~n~~ew ~~authentication services provider~~OBF realm; the new OBF realm and the user generate a new OBF_Token for use in the new OBF realm.

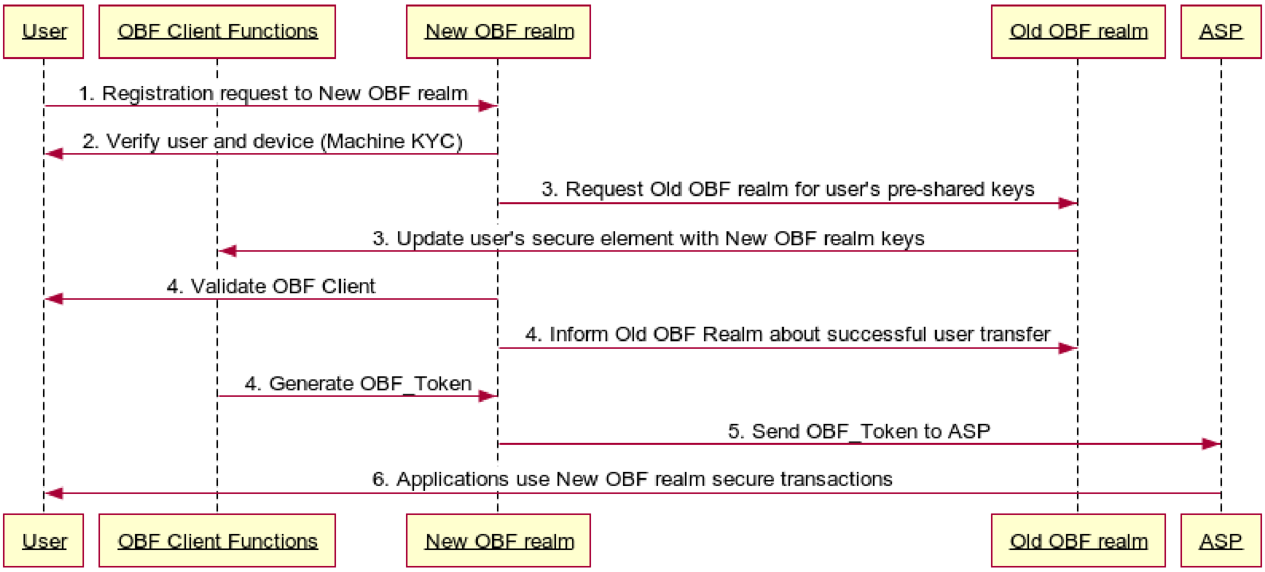
5. The new OBF realm ~~Upon successful confirmation of the~~ transfers the ~~new authentication services provider informs~~user's OBF_Token to the ASP~~application services providers about the change in the OBF_Token for a user~~; and

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

~~6.~~

NOTE ~~–~~: Machine KYC is 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.

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


User
OBF Client Functions
New OBF realm
Old OBF realm
ASP
1. Registration request to New OBF realm
2. Verify user and device (Machine KYC)
3. Request Old OBF realm for user's pre-shared keys
3. Update user's secure element with New OBF realm keys
4. Validate OBF Client
4. Inform Old OBF Realm about successful user transfer
4. Generate OBF_Token
5. Send OBF_Token to ASP
6. Applications use New OBF realm secure transactions
User
OBF Client Functions
New OBF realm
Old OBF realm
ASP

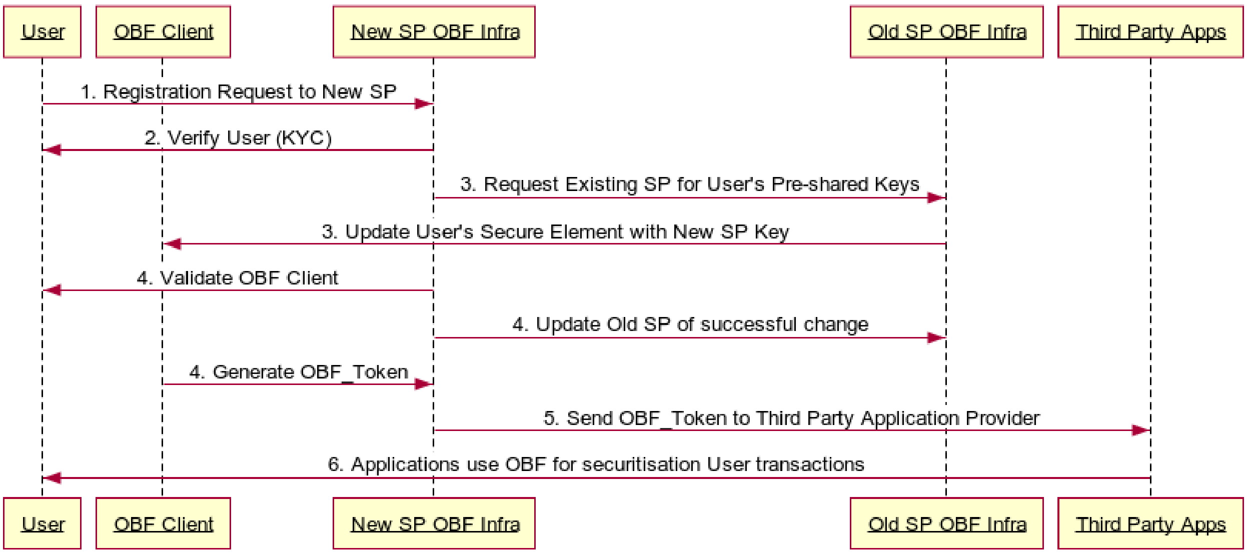

User
OBF Client
New SP OBF Infra
Old SP OBF Infra
Third Party Apps
1. Registration Request to New SP
2. Verify User (KYC)
3. Request Existing SP for User's Pre-shared Keys
3. Update User's Secure Element with New SP Key
4. Validate OBF Client
4. Update Old SP of successful change
4. Generate OBF_Token
5. Send OBF_Token to Third Party Application Provider
6. Applications use OBF for securitisation User transactions
User
OBF Client
New SP OBF Infra
Old SP OBF Infra
Third Party Apps

Figure 10-3: Change of OBF realm~~Authentication provider change~~ (symmetric keys)

10.2.24 Change~~ing~~ of ~~authentication provider flow~~OBF realm (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.~~ It is possible that the new OBF realm is using asymmetric keys for authentication. The Steps for transfer of the OBF realm, in the case when asymmetric keys are used for authentication, are described below:~~The authentication provider may be changed as per the mechanism defined below:~~

1. User requests n~~n~~ew ~~authentication services provider~~OBF realm for its services;
2. The ~~new authentication services~~new OBF realm ~~provider~~ completes the m~~m~~achine KYC;
3. The new OBF realm ~~new authentication service provider~~ provides its public key to the old OBF realm ~~old aAuthentication Sservice Pprovider (ASP)~~ with a request to transfer the user's account to the new OBF realm~~new authentication service provider~~;
4. The ~~o~~old ~~authentication services provider~~OBF realm uses its private key to update the secure element of the user with the public key of the ~~new~~ new ~~authentication services provider~~OBF realm;
5. Upon successful confirmation of the transfer the ~~new~~ new ~~authentication services provider~~OBF realm informs the ~~application services providers~~ASP about the change in the OBF_Token for a user; and
6. The ~~application service provider~~ASP uses the new OBF_Token ~~along with embedded connectivity identity~~ to ~~verify~~ authenticate the user.

The Process is described in the diagram below (Figure 10-4):

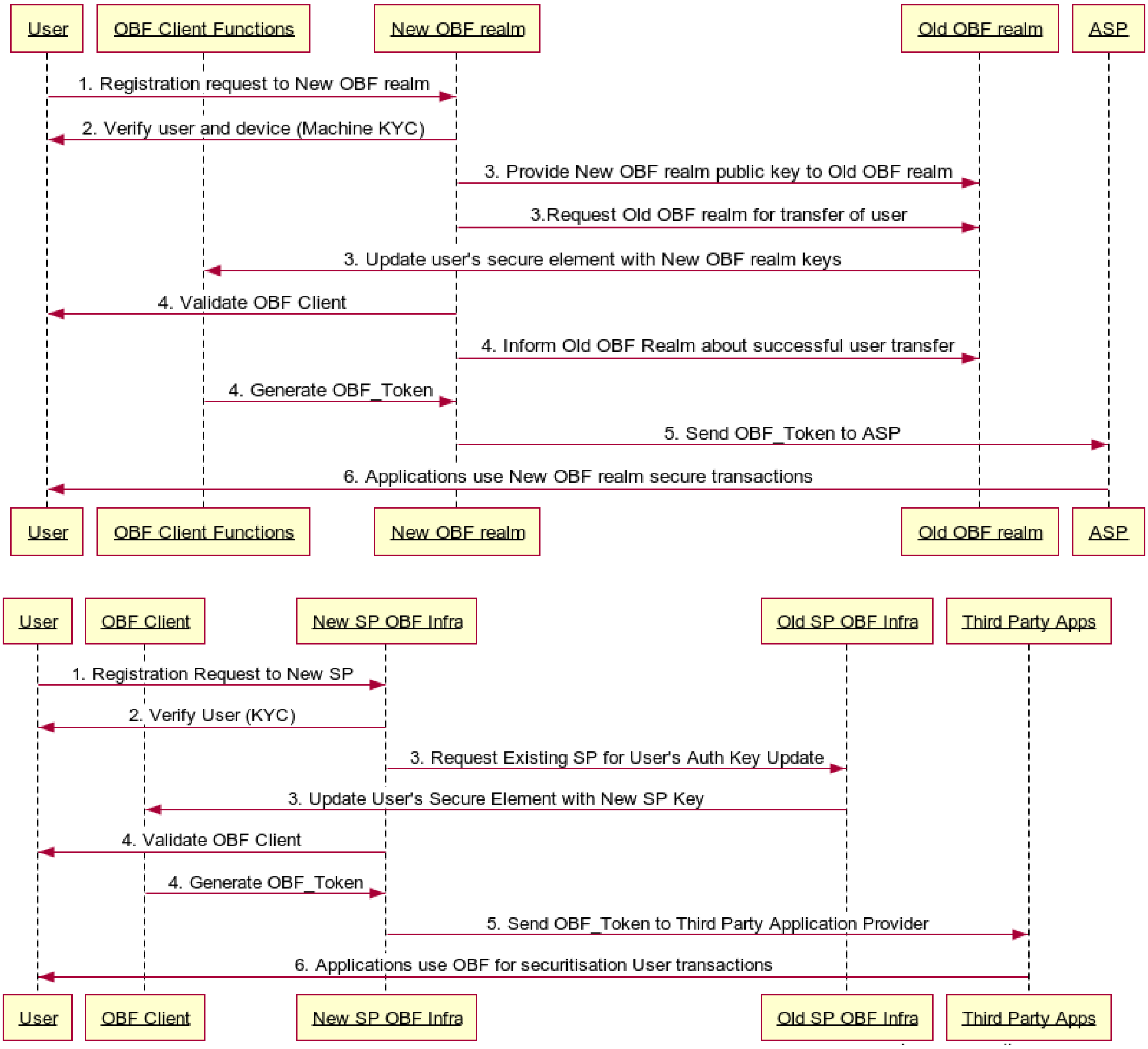

User
OBF Client Functions
New OBF realm
Old OBF realm
ASP
1. Registration request to New OBF realm
2. Verify user and device (Machine KYC)
3. Provide New OBF realm public key to Old OBF realm
3.Request Old OBF realm for transfer of user
3. Update user's secure element with New OBF realm keys
4. Validate OBF Client
4. Inform Old OBF Realm about successful user transfer
4. Generate OBF_Token
5. Send OBF_Token to ASP
6. Applications use New OBF realm secure transactions
User
OBF Client Functions
New OBF realm
Old OBF realm
ASP
User
OBF Client
New SP OBF Infra
Old SP OBF Infra
Third Party Apps
1. Registration Request to New SP
2. Verify User (KYC)
3. Request Existing SP for User's Auth Key Update
3. Update User's Secure Element with New SP Key
4. Validate OBF Client
4. Generate OBF_Token
5. Send OBF_Token to Third Party Application Provider
6. Applications use OBF for securitisation User transactions
User
OBF Client
New SP OBF Infra
Old SP OBF Infra
Third Party Apps

Figure 10-4: ~~Authentication provider change~~Change of OBF realm (asymmetric 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).*