INTERNATIONAL TELECOMMUNICATION UNION

TELECOMMUNICATION STANDARDIZATION SECTOR

STUDY PERIOD 2017-2020

Q16-13-JUL20-C921

STUDY GROUP 13

English only

Question(s): 16/13 Virtual, 20-31 July 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" - for consent

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; Preparation for consent; Q16/13; 20-31 July 2020

Abstract: This contribution proposes for requesting consent to the draft new Recommendation ITU-T Y.OBF_Trust "Open Bootstrap Framework enabling trustworthy networking and services for distributed diverse ecosystems" in the plenary of July 2020 SG13 meeting.

1. Introduction

Draft new Recommendation ITU-T Y.OBF_Trust: "Open Bootstrap Framework enabling trustworthy networking and services for distributed diverse ecosystems" was initiated in March 2019 and since then it has been continuously updated and improved based on the reviews/ observations/ comments in the intervening physical/ virtual meetings. More than 10 iterations for improvement of the document have been carried out so far resulting in the present version, which is the candidate for consent. The output document of virtual interim meeting on 29 June 2020 (**TD-XXX/WP3**) is the base document for this contribution.

Since the work item is targeted for requesting consent in the plenary of July 2020 SG13 meeting, this contribution has been prepared mainly for the purpose of finalizing the same.

2. Proposal

This contribution proposes minor updates towards preparing for consent of Y.OBF_Trust ("Open Bootstrap Framework enabling trustworthy networking and services for distributed diverse ecosystems") at the July 2020 SG13 meeting. The updated draft new Recommendation ITU-T Y.OBF_Trust is annexed (**Annexure-I)** for consideration.

The updates in this contribution are summarized below:

- General English language updates;
- (Summary) Minor edit by adding "related" in referring to "related Security Parameters"
- (clause 3.2.1) Update in the definition of "bootstrapping" (by addition of text: "enabling the device to communicate securely with trusted services").
- (clause 6) Update in the clause describing the "Network operator" (by addition of text: "Network operator refers to any service provider that deploys an OBF realm, including but not limited to Mobile Network Operators, Virtual Network Operator, IoT Service Providers, etc.").
- (Clause 7.1 Requirements) Minor edit to two bullet points
- (Clause 8 OBF Reference Model) Language simplification on the first paragraph
- (Clause 9.2.4 Specifications of reference point RPDS) A minor edit and deletion of a redundant line item
- (Clause 10.2.2 Change of OBF Realm mechanism) Minor edits
- (Figure 10.2) NOTE for clarifying 'opt' as an optional flow

_____

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 recommendation includes the OBF concept, the requirements of the OBF as well as the pre-requisites for the devices and the applications. It also includes a reference model describing the OBF elements and a functional architecture describing four functional groups, four reference points and related security parameters. The information workflows for the bootstrapping, authentication and change of OBF realms ~~is~~ are 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

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 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 authentication and authorization of devices for access to trusted services provisioned by ASPs.

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 by ASPs 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, enabling the device to communicate securely with trusted services. See also clause 3.2.2 of [ITU-T X.1311 (02/2011)].

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 devices and applications.

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

FQDN Fully Qualified Domain Name

GBA Generic Bootstrapping Architecture

HTTP	Hyper Text Transfer Protocol
IoT	Internet of Things
IPSec	Internet Protocol Security
KYC	Know Your Customer
OBF	Open Bootstrap Framework
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**" 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

The Open Bootstrap Framework (OBF) is beneficial for ~~U~~users of new age devices and applications that require a trust framework ~~secure mechanisms~~ for accessing trusted services. At the same time, providers of secure applications and trusted services also require mechanisms for a minimum level of authentication of the ~~Users~~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 bootstrapping, authentication~~,~~ ~~bootstrapping~~ and authorization to create trustful interactions between devices and applications.

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

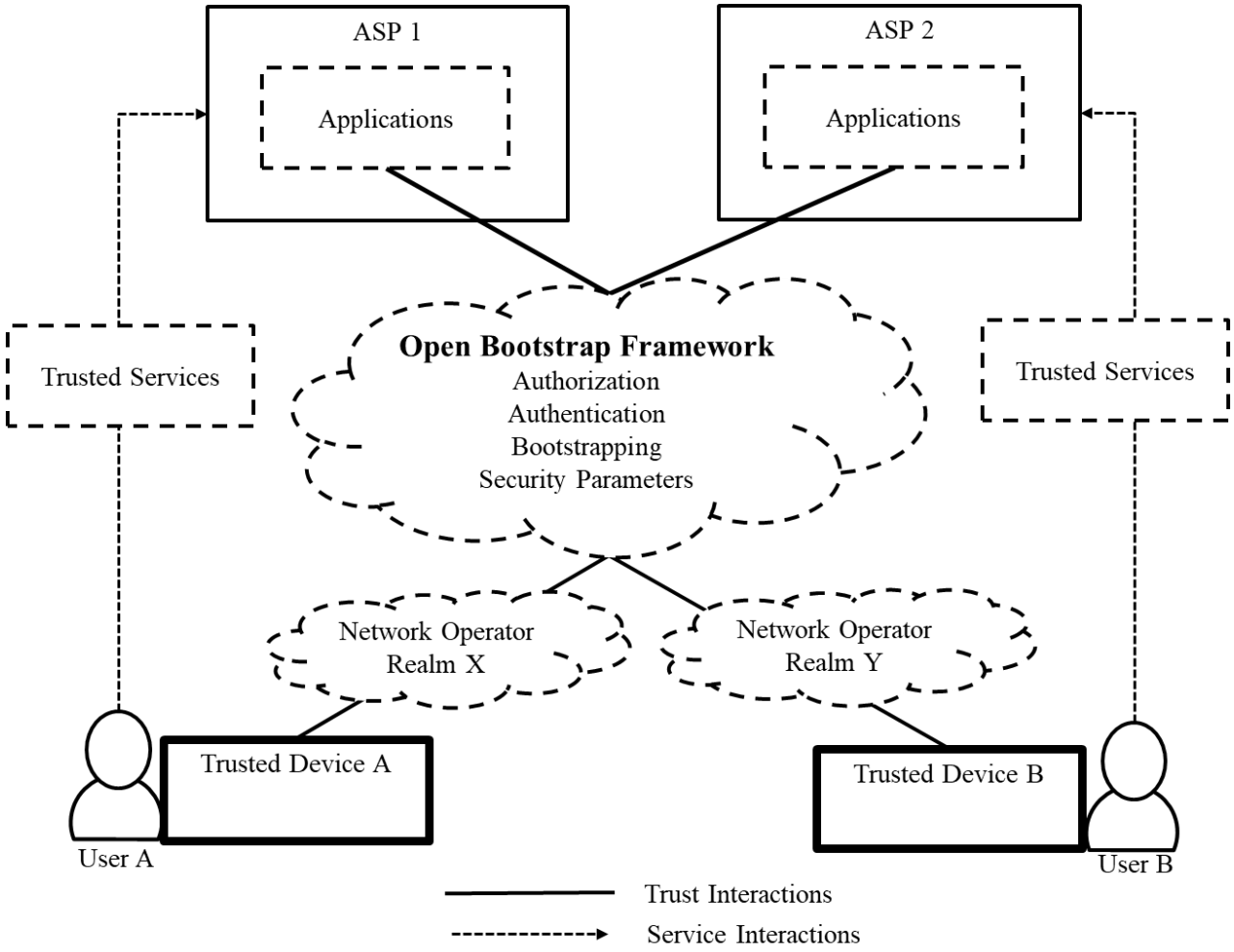


Figure 6-1: OBF concept

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 ASP by using the security capabilities of the OBF.

An OBF realm can address the following actors and stakeholders:

1. **User~~s~~:** 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~~of the subscriber and the device. It can share ~~the~~ this trust to bridge new relationships between providers of trusted services and users of trusted devices by deploying an OBF realm. Network operator refers to any service provider that deploys an OBF realm, including but not limited to Mobile Network Operators, Virtual Network Operator, IoT Service Providers, etc.

3. **Application service~~s~~ provider~~s~~ (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~~ address the OBF elements in an OBF realm deployed by a network operator~~s and the OBF elements that have been deployed~~;
- identify and on-board ASPs whose applications require to be protected from unauthorized usage;
- identify and on-board ~~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 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 in the device or its connectivity element (e.g. SIM card);

- have configurations that make the device 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.

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

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

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.

The OBF applications are required to:

- be OBF aware, and be able to indicate to the device the protocol and keying material required to connect to the application;
- implement Diameter / HTTP proxy functionality to act as a proxy towards the OBF realm in which the user is bootstrapped;
- be able to locate the user's OBF realm and communicate securely with the OBF functions;
- acquire the user's security parameters from the OBF realm; and
- implement the security parameters in its security protocol used for creating secure associations between the device and the application.

8 OBF reference model

The OBF reference model describes the ~~key~~ interactions between the OBF 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, namely, the devices and the secure applications.

The OBF reference model is shown in the diagram below:

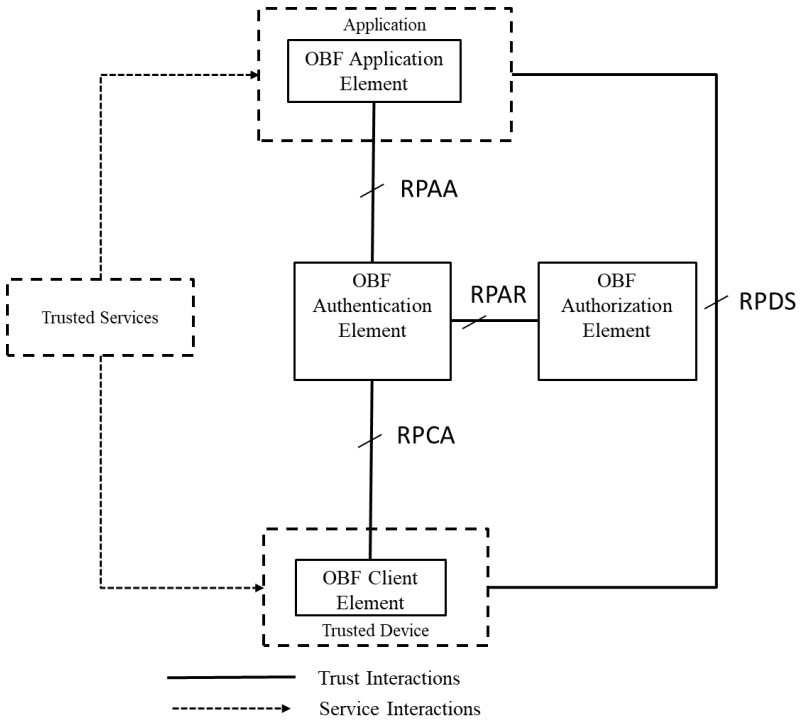


Figure 8-1: OBF reference model

8.1 OBF elements

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 client 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 device. 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 authorization element

The OBF authorization element carries out the key management and provides the keying material as per standard security protocols.

8.1.3 OBF authentication element

The OBF authentication element identifies and authenticates the OBF client element using the keying material from the OBF authorization element.

8.1.4 OBF application element

The OBF Application element sets up the secure connections between the device and the applications using the security enablement from the other OBF elements.

8.2 OBF reference points

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.

9 OBF functional architecture

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.

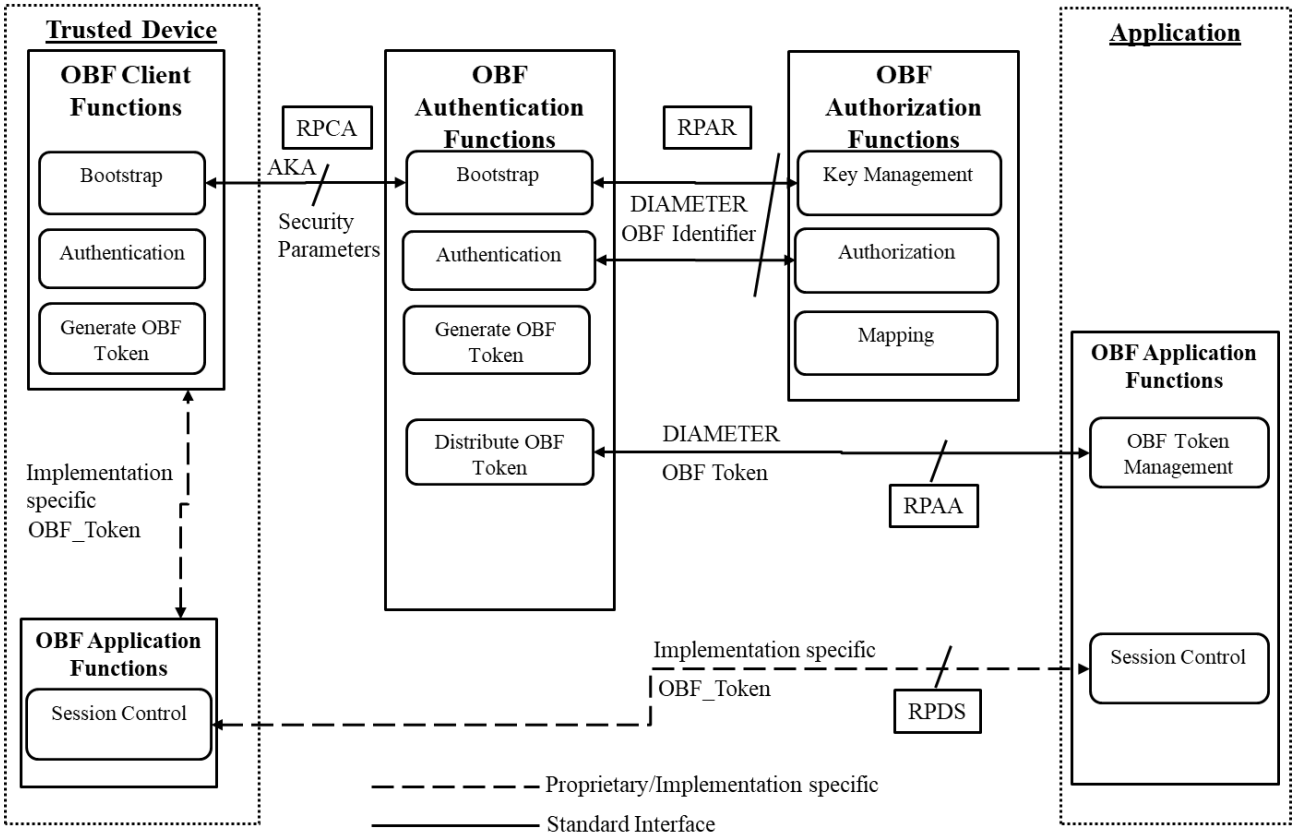


Figure 9-1: OBF functional architecture

The functional architecture 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 - When the OBF is deployed in the network, that context is referred to as a realm. The instantiated functions within the realm are referred as nodes. As an example, an Authentication

function, when instantiated in the network, will be called the authentication node in the realm of that OBF deployment.

The following functionalities are supported by all the OBF functions:

- the OBF numbering;
- identification and authentication of each other within the OBF realm(s);
- identification and authentication of OBF clients; and
- transferability between network operators.

Each of the 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) Authentication function

This function mutually authenticates the OBF client and the authentication 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.

The authentication function provides the following functionalities:

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

(b) Bootstrapping function

This function provides the functionality for a new registration of a trusted device by way of establishment of new long-term secret key(s) for secure communication.

(c) Generate OBF_Token function

This function generates the OBF_Token, after bootstrapping has successfully been completed, by using the agreed OBF security parameters. The OBF_Token is specific to the subscription information and the application for which it is generated.

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) 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) Key management function

This function provides the 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) Authorization function

This function validates if the device can access the application based on the OBF_Token sent in the authentication request. The function hosts the repository of registered applications that can be permitted for use by the device, and also the mapping of the specific applications that are allowed to be used by OBF client functions of a device.

The authorization function provides the following functionalities:

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

(c) 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:

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

9.1.3 OBF 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 section 9.1.1.

Together, the three functions enable the OBF client 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.

9.2 Specifications of OBF reference points

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

9.2.1 RPAA

The reference point RPAA is 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 RPAR

The reference point RPAR 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:

- 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 RPCA

The reference point RPCA 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 RPDS

The reference point RPDS supports the protocol required for the secure interaction between the device and the application.

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 or new 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 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, 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 device and the application. The timestamp of the OBF_Token is synchronized and controlled by the authentication function.

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 outlined in the 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.

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

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;
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;
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;
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; 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 10-1) in which the numbering of the steps in the diagram follows the numbering of steps in the paragraph above:

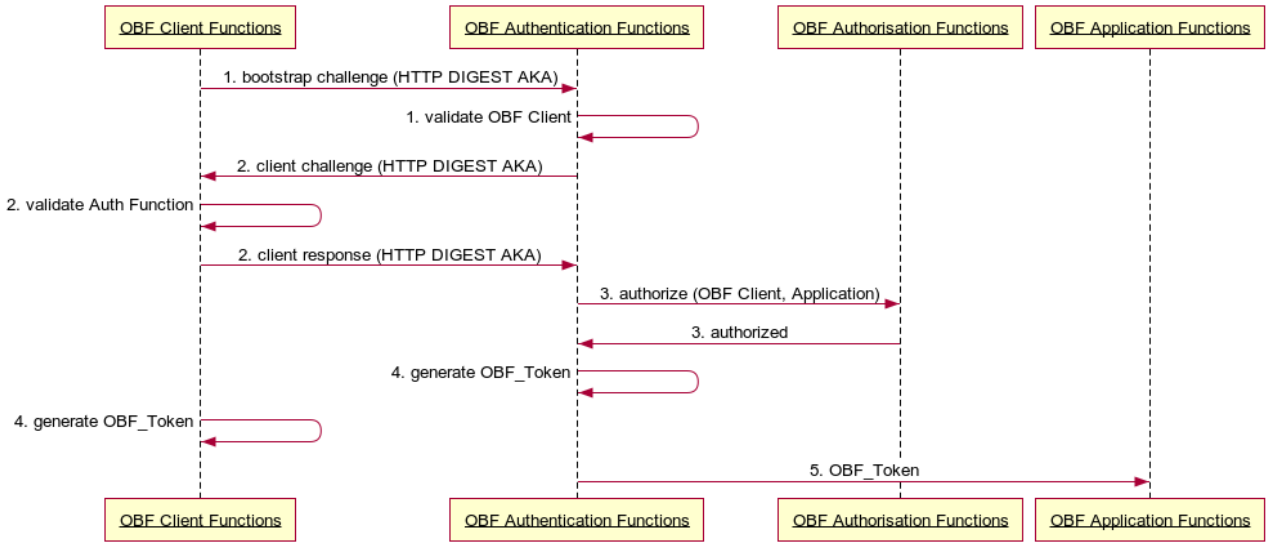


Figure 10-1: Bootstrapping workflow

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.

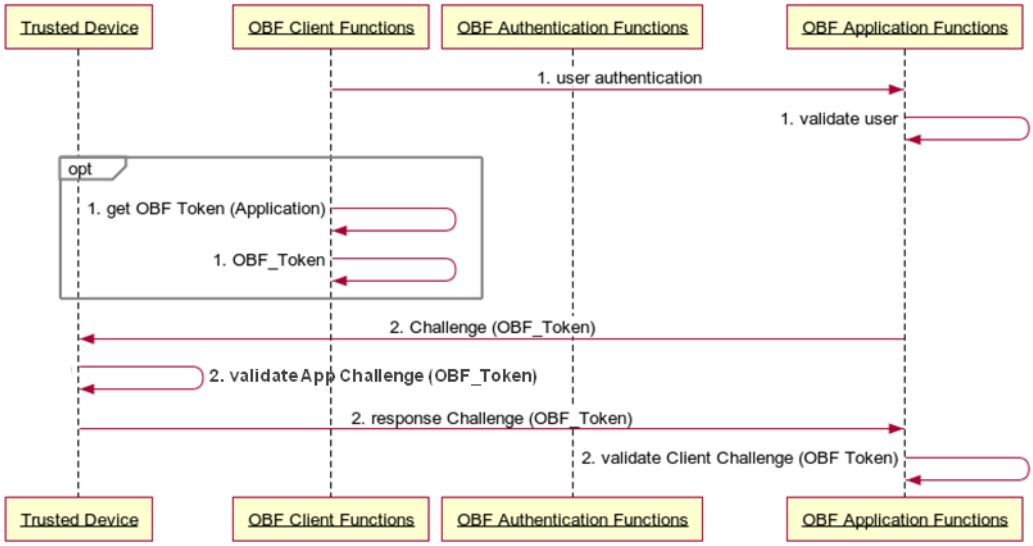
10.1.2 Authentication workflow

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; 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.

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:



NOTE – 'opt' refers to an optional flow

Figure 10-2: Authentication workflow

10.2 Workflow for changes in OBF realm

10.2.1 Change of OBF realm (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 next network operator or IoT service provider, referred as the new OBF realm, for enabling the use of the 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 new OBF realm for its services;
2. The new OBF realm undertakes the verification of the user and the device (machine KYC) and upon successful verification, requests the old OBF realm for the user's shared keys;
3. The new OBF realm uses the old OBF realm's key(s) to update the secure element with the key(s) of the new OBF realm;
4. The new OBF realm authenticates the OBF client functions using its keys, and upon success, informs the user and the old OBF realm of the successful transfer of the user to the new 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 transfers the user's OBF_Token to the ASP; and
6. The ASP uses the new OBF_Token to provide trusted services to the user.

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):

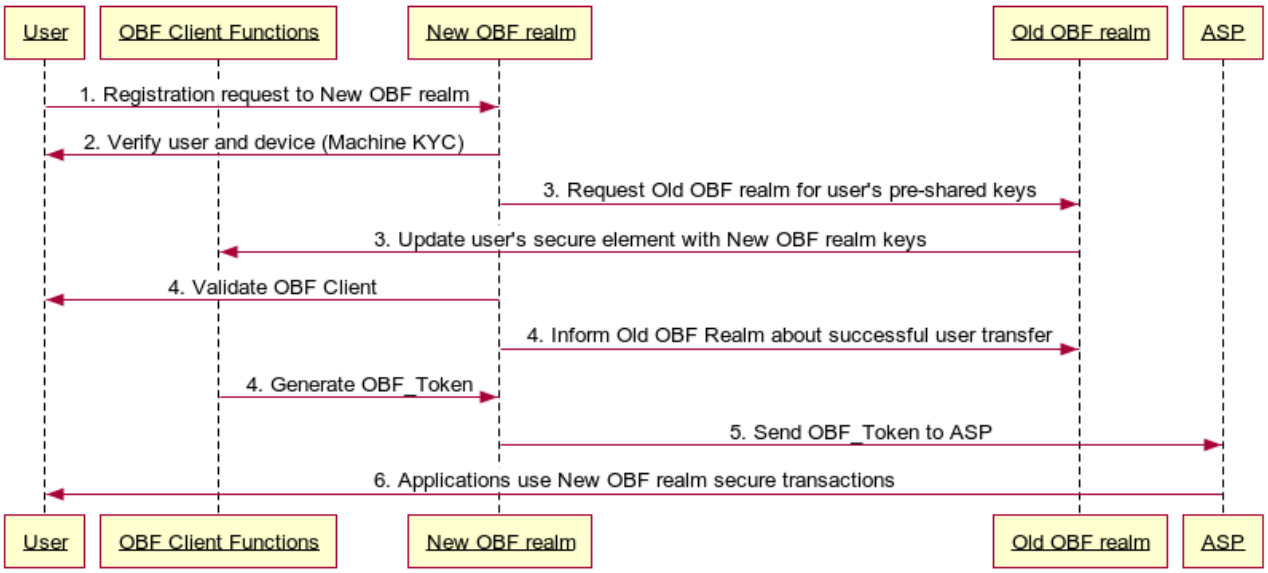


Figure 10-3: Change of OBF realm (symmetric keys)

10.2.2 Change of OBF realm (asymmetric keys)

It is possible that the new OBF realm ~~is using~~uses asymmetric keys for authentication. The ~~Steps~~ steps for transfer of the OBF realm, in the case when asymmetric keys are used for authentication, are described below:

1. User requests new OBF realm for its services;
2. The new OBF realm completes the machine KYC;
3. The new OBF realm provides its public key to the old OBF realm with a request to transfer the user's account to the new OBF realm;
4. The old OBF realm uses its private key to update the secure element of the user with the public key of the new OBF realm;
5. Upon successful confirmation of the transfer the new OBF realm informs the ASP about the change in the OBF_Token for a user; and
6. The ASP uses the new OBF_Token to authenticate the user.

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

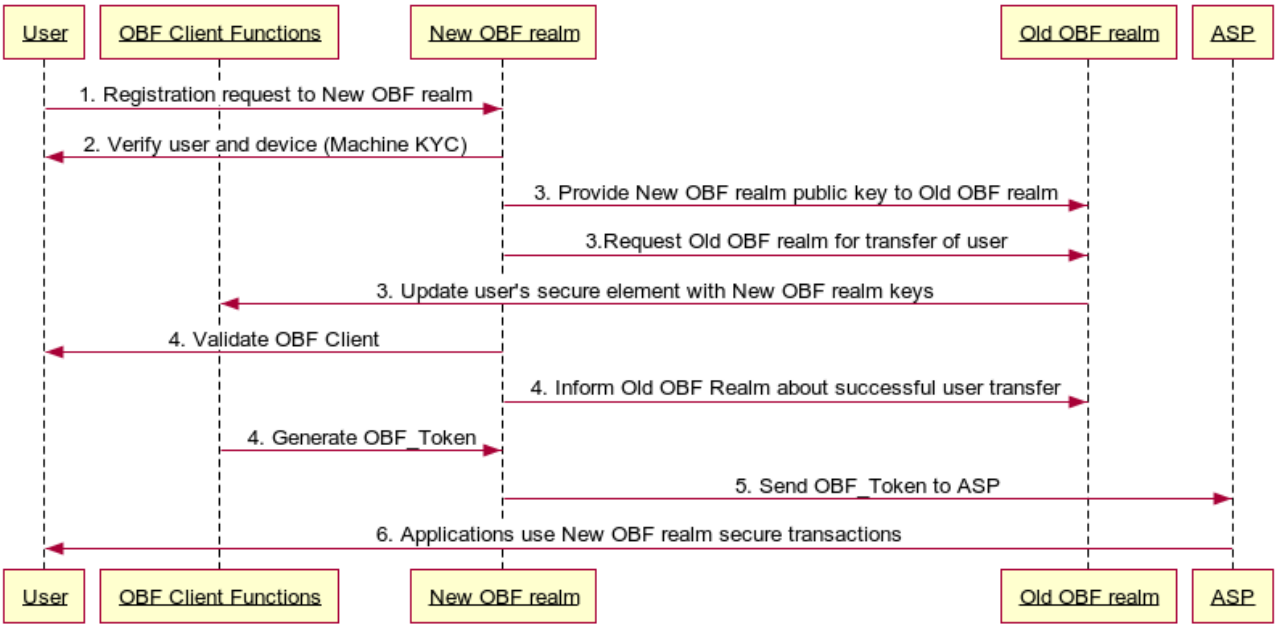


Figure 10-4: Change of OBF realm (asymmetric keys)

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).*

_____