INTERNATIONAL TELECOMMUNICATION UNION

TELECOMMUNICATION STANDARDIZATION SECTOR

STUDY PERIOD 2017-2020

Q16-13-Jun20-C127

Study Group 13

English only

Question(s): 16/13 e-Meeting, 8 June 2020

CONTRIBUTION

Source: Telecom Engineering Centre (TEC), Ministry of Communications, India

Title: Draft Recommendation ITU-T Y.OBF_Trust: "Open Bootstrap Framework enabling trusted devices, applications and services for distributed diverse ecosystem"

(e-meeting, 8 June 2020)

Purpose: Proposal

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

Keywords: Y.OBF_Trust; Q16/13; interim; 08 June 2020

Abstract: This document proposes some modifications in the draft Recommendation ITU-T Y.OBF_Trust (TD423) for discussion at interim e-meeting of Q16/13.

1. Introduction

In the Editor's Note in Section 8 of the output document (**TD423**) of e-meeting dated 18-22 May 2020, it is expected that the "*section be reviewed and re-drafted following conventions. Only essential requirements to be kept and the redundant text be removed.* In order to address the above observation and for more clarity and improved readability, some modifications are being proposed.

It is mentioned that although the required references are made in the document, however, it is felt that it may be necessary to take out some clauses or parts from those references and reproduce those with appropriate changes in this document, in order to make it more readable. Such clauses and parts have been proposed in this contribution.

2. Proposal

It is proposed to make some modifications in the draft Recommendation ITU-T Y.OBF_Trust (**TD423**). The proposed modifications are in track change mode in **Annexure-I**.

3. Reference

[1] SG13-TD423/WP3: Base document for this contribution.

Annexure-I

Draft new Recommendation ITU-T Y.OBF_Trust

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

Summary

This Recommendation provides an Open Bootstrap Framework (OBF) for secure provisioning of trusted services by Application Services Providers that have no existing trust relationship with the Users. The OBF provides a trust model with a supporting reference architecture and deployment that includes an OBF Client, an OBF Authentication Server, an OBF Resource Server and four Reference Points. The recommendation includes mechanisms and workflows for the provisioning of security tokens, mutual authentication between Connected Devices, Applications and Service Providers and also the mechanism for Users to change the Service Providers.

This Recommendation describes the concept, trust model and the functional architecture of the OBF. It identifies the requirements, capabilities and functions to support the OBF. In addition, it specifies the mechanisms and workflows for the deployment of the OBF.

This Recommendation is relevant to Network Service Providers, IoT Service Providers and Applications/ Services Providers for deployment of trusted services in the emerging 5G / Smart Cities / IoT Application / Services domain.

Keywords

Bootstrapping; IoT; IoT Service Provider; OBF, OBF Proxy; 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 specifies an Open Bootstrap Framework that facilitates the Authentication and Authorisation of Connected Devices, Connected Services, Service Providers and Applications.

The scope of this Recommendation includes

- overview of the Open Bootstrap Framework
- OBF reference model
- the OBF elements
- requirements and capabilities of OBF elements
- pre-requisites and capabilities of beneficiary devices and applications
- mechanisms and workflows of the OBF

This Recommendation demonstrates how existing secure elements and bootstrapping mechanisms deployed by Network Service Providers can be used to provision trusted services by Application Service Providers to untrusted Users and Connected Devices.

This Recommendation also includes Industry use cases in Appendix I for exemplifying the deployment of the OBF.

2 References

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

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

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

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

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

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

[ITU-T Y.2724] Recommendation ITU-T Y.2724 (2013), *Framework for supporting OAuth and OpenID in next generation networks*

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

[ITU-T Y.4000] Recommendation ITU-T Y.4000/ Y.2060 (2012), *Overview of the Internet of things;* and

[ITU-T Series Y Supplement 53 (12/2018] *ITU-T Y.4000-series – Internet of Things use cases*

[ITU-T Y.4413] Recommendation ITU-T Y.4413/F.748.5 (2015), *Requirements and reference architecture of the machine-to-machine service layer*

[ITU-T Y.4451] Recommendation ITU-T Y.4451 (2016), *Framework of constrained device networking in the IoT environments*

[ITU-T M.1400] Recommendation ITU-T M.1400 (2015), *Designations for interconnections among operators' networks*

[ITU-T M.3208.1] Recommendation ITU-T M.3208.1 (1997), *TMN management services for dedicated and reconfigurable circuits network: Leased circuit services*

[ITU-T M.3320] Recommendation ITU-T M.3320 (1997), *Management requirements framework for the TMN X-Interface*

3 Definitions

3.1 Terms defined elsewhere

This Recommendation uses the following terms defined elsewhere:

3.1.1. Authentication servers [ITU-T X.1113 (11/2007)]: Authentication servers refer to servers that provide authentication services to users or other systems. Authentication is generally used as the basis for authorization (determining whether a privilege will be granted to a particular user or process), privacy (preventing the disclosure of information to non-participants), and non-repudiation (not being able to deny having done something that was authorized to be done based on the authentication).

3.1.2. Constrained Device [ITU-T Y.4451 (09/2016)]: A device that has constraints on characteristics such as limited processing capability, small memory capability, limited battery power, short range and low bit rate.

3.1.3. Internet of Things (IoT) [ITU-T Y.4000/ Y.2060 (06/2012)]: A global infrastructure for the information society, enabling advanced services by interconnecting (physical and virtual) things based on existing and evolving interoperable information and communication technologies.

NOTE 1 – Through the exploitation of identification, data capture, processing and communication capabilities, the IoT makes full use of things to offer services to all kinds of applications, whilst ensuring that security and privacy requirements are fulfilled.

NOTE 2 – From a broader perspective, the IoT can be perceived as a vision with technological and societal implications.

3.1.4. **M2M Service Provider** [ITU-T Terms and Definitions]: Entity (e.g., a company) that provides M2M common services to a M2M application service provider or to the user. See [ITU-T Y.4413/F.748.5 (11/2015)] and [ITU-T Series Y Supplement 53 (12/2018)].

3.1.5. **Network Operator** [ITU-T M.1400 (04/2015)]: An operator that manages a telecommunications network. A Network Operator may be a Service Provider and vice versa. A Network Operator may or may not provide particular telecommunications services. See clause 1.4.2.3 of [ITU-T M.3208.1 (10/97)], and clause 1.4.4 of [ITU-T M.3320 (04/97)]

3.1.6. **Resource server** [ITU-T Y.2724 (11/2013)]: The server hosting the protected resources, capable of accepting and responding to protected resource requests using access tokens.

3.1.7. **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.

3.1.8. **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**.**

3.1.9. **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.

3.1.10. **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.

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

3.2 Terms defined in this Recommendation

This Recommendation defines the following terms:

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

3.2.2. **Connected Device**: A device that has an embedded secure element in itself or its Connectivity Element.

NOTE **-** Though Connected Device may or may not be a Constrained Device; however, in this framework a Constrained Device may also be used as a Connected Device.

3.2.3. **IoT Service Provider:** A Provider of IoT Devices, Communications, Applications and Services.

NOTE - Similar to M2M Service Provider defined in clause 3.1.4.

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

3.2.5. **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.

3.2.6. **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.

3.2.7. **OBF_Token:** A session key, independently generated in the Connected Device / User Equipment (UE) as well as in the Authentication Server, based on an agreed security schema between the Device and the Authentication Server for establishing a secure connection between the Connected Device and the Application.

3.2.8. **Operator Services:** Services provided to the user of a Connected Device, that are offered by and hosted in the network of the Network Service Provider (NSP) e.g. Mobile Network Operator (MNO), Telecom Service Provider (TSP).

3.2.9. **Resource Server:** A Server that holds / hosts the permissions/ restrictions applicable to protected user resources.

3.2.10. **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)]

3.2.11. **Third Party**: An entity other than the Network Service Provider or the IoT Service Provider, which consumes the security capabilities of a network for providing trust for applications and / or services offered to the end users.

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

4 Abbreviations and acronyms

This Recommendation uses the following abbreviations and acronyms:

3GPP	3rd Generation Partnership Project

AKA	Authentication and Key Agreement
COAP	Constrained Object Authentication Protocol
EID	eUICC-ID
FQDN	Fully Qualified Domain Name
GBA	Generic Bootstrapping Architecture
GSM	Global System for Mobile communication
HTTP	Hyper Text Transfer Protocol
ICT	Information and Communication Technology
IoT	Internet of Things
IoT SP	IoT Service Provider
IPSec	Internet Protocol Security
KYC	Know Your Customer
M2M	Machine to Machine
M2M SP	M2M Service Provider
MNO	Mobile Network Operator
MQTT	Message Queue Telemetry Transport
MSISDN	Mobile Station International Subscriber Directory Number
NSP	Network Service Provider, see also clause 3.1.5 - Network Operator
OBF	Open Bootstrap Framework
PSK	Pre-Shared Key
PSK-TLS	Pre-Shared Key Cipher suites for Transport Layer Security
SIM	Subscriber Identification Module
TLS	Transport Layer Security
TSP	Telecom Service Provider, see also MNO
UID	Universal Identifier or Public Entity Identifier

5 Conventions

In this Recommendation, requirements are classified as follows:

- The keywords "**is required to**" or “**are required to**” indicate a requirement/ requirements, which must be strictly followed and from which no deviation is permitted if conformance to this document is to be claimed;
- The keywords "**is recommended**" indicate a requirement, which is recommended but which is not absolutely required. Thus, such requirements need not be present to claim conformance; and
- The keywords "**optionally**" or "**may**" indicate an optional requirement which is permissible, without implying any sense of being recommended. These terms are not intended to imply

that the vendor's implementation must provide the option; it means the vendor may optionally provide the feature and still claim conformance with the specification.

6 Overview of the Open Bootstrap Framework

6.1 OBF Concept

Users of new age services require mechanisms for trusted access to reliable Services. At the same time, providers of trusted applications and services also require mechanisms for a minimum level of authentication of the Users. Open bootstrap framework enables secure interactions between Users and Connected Devices on the one hand, and the Applications and Service Providers on the other, by utilizing the inherent security capabilities of the underlying network technology layer.

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 OBF recommendation makes it possible to extend the existing trust relationship between the network operator and its subscribers to enable one to many trust relationships between the users and the many new age service providers.

The OBF defines a set of functions, requirements, capabilities and mechanisms that open up the security capabilities of the network layer to all types of connected devices, applications and services. The OBF can be implemented by Network Operators providing any type of network that permits bootstrapping of devices. Further, any user of a bootstrapped device can access the applications and services of any ASP by using the OBF functions and mechanisms.

The OBF is a trust framework, which opens up and extends the existing trust relationship between the user and the Network Operator to be utilized by an ASP for providing trusted services to hitherto unknown user of a connected device.

The OBF addresses the critical need for an inter-operable, network technology agnostic framework for sharing of trust between the Network Operator, ASP and the User by bootstrapping of connected devices, authentication of users and authorization of applications based on the trust provided by the Network Operator.

The Users and their Connected Devices are authenticated based on the bootstrapping provided by the Network Operator that provides the network layer security and access control. ASPs use the Network layer authentication with an added layer of Authorization such as to secure the use of specific Applications to specific Users / Connected Devices.

The OBF provides an interoperable, implementation-independent and scalable approach for multi-party interactions in which a diverse set of connected devices using a variety of network technologies are able to access applications and services securely. The OBF utilizes the identification and authentication carried out for and by underlying technology layers, and extends it towards the ASPs for the provision of trusted services.

The trust framework of the OBF has the following key actors:

1. **Network & IoT Service Provider:** The Network Operator that provides the connectivity services, undertakes the physical verification of the Subscriber and shares that trust with ASPs and Users by the mechanism of the OBF. This reuse of trust, involves on the one hand, issuance and use of pre-shared keys hosted on the Secure Element, and on the other, the issuance and use of secure tokens to the ASPs.
2. **Applications & Service Providers:** The ASP is an entity that has developed applications for providing Trusted Services that benefit Users, and has an expectation of a minimum level of

authentication and authorization for the use of the ASP's application and services by the User. However, the ASP does not have a direct relationship, unlike the one between the Network Operator and the Subscribers. The mechanisms of the OBF provide for a Subscriber of the Network Operator to become an authenticated User of the ASP. The ASP uses the secure token for establishing this trust relationship.

3. **Users:** The Subscribers of the Network Operator benefit from the mechanisms of the OBF, reusing the authentication and authorization for availing the Trusted Services provided by the ASP.

The OBF provides a reference model which can be used to create a functional deployment architecture for any underlying network technology layer.

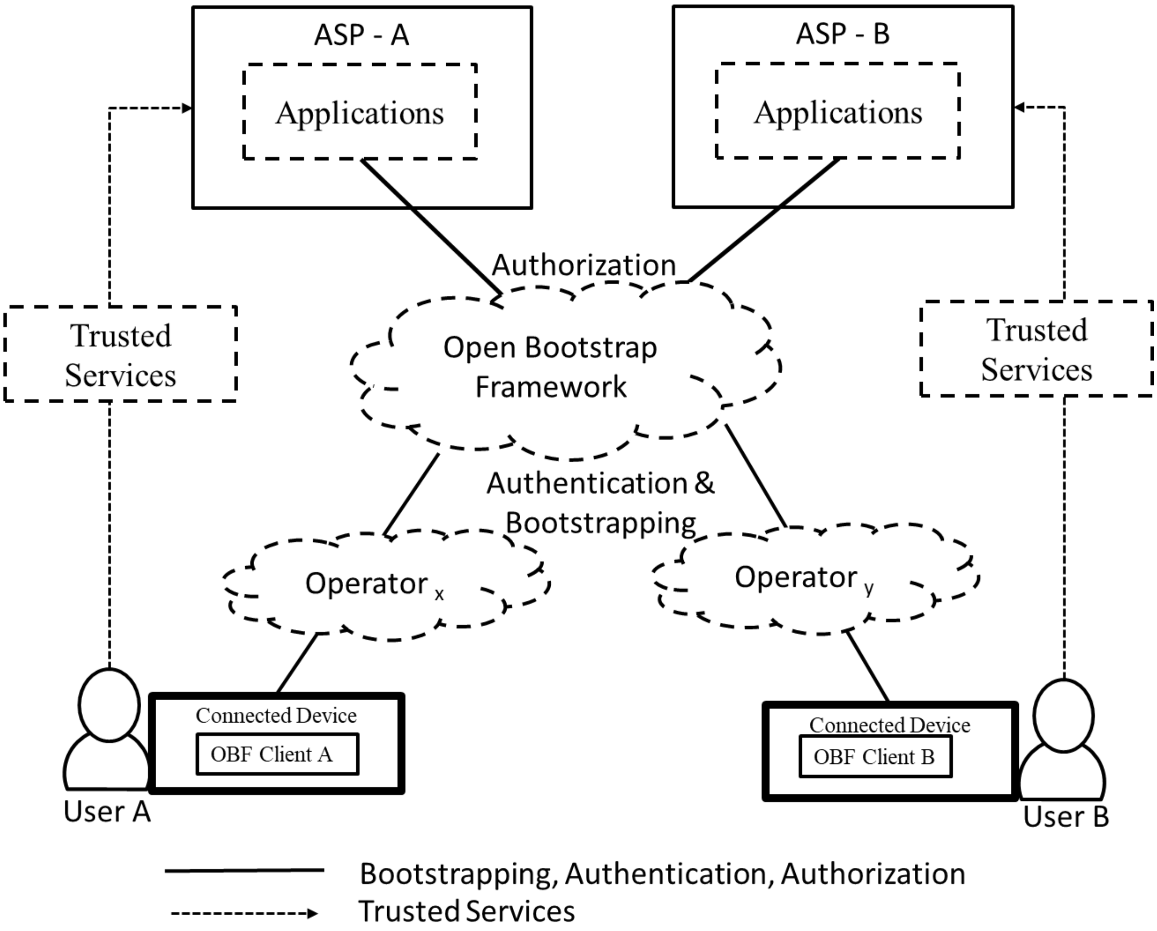

Figure 1: OBF Concept

OBF elements [Nodes, Reference Points, Identifiers] are used to enable application functions in the Application Servers and on the Connected Devices (CD) to establish shared session keys based on an open selectable AKA protocol.

The OBF provides for the following functions:

1. **Bootstrapping of Connected Devices**: The bootstrapping function provides a strong mutual authentication of a connected device to the authentication server by the use of the security capabilities of the secure element.

The OBF uses the unique identity and security capabilities of the tamper resilient secure element as the root of trust.

At the completion of the bootstrapping process, a Connected Device is registered to the trust framework of the OBF.

2. **Authentication of the Users:** The Authentication services provide mechanisms for the bootstrapped Connected Devices to be identified by the applications and for the user interactions to be secured using agreed security algorithms.

3. **Authorization of Applications and Services:** The Authorization Services provide mechanisms for the application clients and applications to interact with each other with end to end security enablement using key material and security algorithms provided by the Authentication services of the OBF.

   OBF interactions result in the conversion of a verified Subscriber of a Network into an Authenticated User of the ASP, allowing Service Access to be controlled effectively by the ASP.

7 OBF Elements

The inter-working of the OBF Elements i.e. the three OBF nodes (OBF Client, OBF Authentication Server and OBF Resource Server) and four reference points (RPAA, RPAR, RPCA and RPDS), is shown in the diagram below:

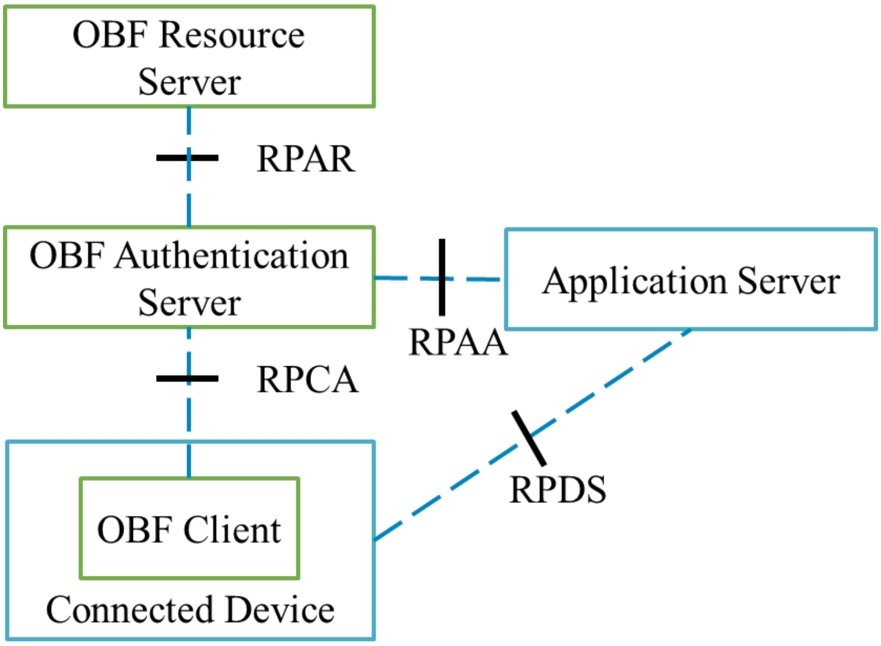

Figure 2: OBF Elements

The Connected Device and the Application Server, are also shown in the diagram as these are the beneficiaries of the trust framework, created by the OBF Elements.

Aside from the three (3) Nodes, four (4) Reference Points, the OBF specifies the required Security Parameters, which are essential for establishing secure sessions. The Security Parameters include Identifiers, subscription information and the OBF_Token. The purpose of the Identifiers is to uniquely identity and address the OBF Clients and the OBF Nodes in an OBF implementation. The purpose of the subscription information is to authenticate and authorise the secure interactions between Users and ASPs via the Network Operator. The OBF_Token is the session key which is used to establish a secure session between the user part and the server part of an Application. The OBF Client binds the User’s identity to the keying material in reference points.

The Connected Device hosts the OBF Client, the user part of the Application offered by the ASP and a part of the subscription information.

The Application Server hosts the server part of the Application and a part of the subscription information required for the delivery of the intended trusted services to the User of the Connected Device. The OBF elements and the Security Parameters are described in the sections below.

7.1 OBF Nodes

The OBF provides three Nodes, each of which is described below:

7.1.1 OBF Client

The OBF Client is an application resident in the Connected 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 side for the bootstrapping of the Connected Device using the Authentication Function. The OBF Client provides the features and functions required for the interaction with the Authentication Server, Resource Server and the Application Server. The OBF Client is specified and provisioned by the IoT Service Provider or the Network Operator that is providing the OBF services.

7.1.2 OBF Resource Server

The OBF Resource Server is a node provisioned by the IoT Service Provider or the Network Operator that provides the key management function and the keying material as per standard AKA protocols. The Resource Server hosts the subscription information of ASPs.

The Resource Server 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 Identifiers.

The Authorisation Function in the Resource Server provides the mechanisms for IoT Service Provider or the Network Operator to authorize ASPs to offer certain services and Users to access the authorized services of the ASP.

7.1.3 OBF Authentication Server

The OBF Authentication Server is a node provisioned by the IoT Service Provider or the Network Operator that identifies and authenticates the OBF Client using the keying material from the OBF Resource Server as per standard AKA protocols and the agreed authentication algorithms.

The Authentication Server generates the OBF_Token, and shares it with the ASPs that are authorized by the Resource Server.

7.2 OBF Reference Points

The OBF specifies four Reference Points, each of which is described below:

7.2.1 RPAA

RPAA is the Reference Point between the Authentication Server and Application Server. It is used by the Application Server to fetch the OBF_Token from the Authentication Server. It is also used to fetch application-specific subscription information of the user from the Authentication Server if requested. The recommended protocol to be used over RPAA is DIAMETER [b-RFC 6733] and [b-RFC 7155].

7.2.2 RPAR

RPAR is the Reference Point between OBF Authentication Server and the OBF Resource Server which uses the DIAMETER protocol [b-RFC 6733] and [b-RFC 7155]. The OBF Authentication Server uses the RPAR to obtain the subscription information regarding the OBF Clients when Users attempt to access certain ASP Applications. The reference points also provides the keying material for the OBF Clients during the bootstrapping mechanism.

7.2.3 RPCA

The Reference Point RPCA is between the OBF Client hosted in the Connected Device and the OBF Authentication Server. The reference point provides the bootstrapping of the OBF Client to the OBF Authentication Server. The required protocol to be used over RPCA is HTTP Digest protocol [b-RFC7616], it may optionally support other protocols as well.

7.2.4 RPDS

The Reference Point RPDS is between the Connected Device and the Application Server. The RPDS supports any protocol as required for the interaction between the user part and the server part of the Application, which is secured using the OBF_Token.

7.3 Security Parameters

The OBF Security parameters include Identifiers, subscription information and the OBF_Token.

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.

7.3.1 Identifiers

The OBF identifiers uniquely identify an OBF Client, a bootstrapped Connected Device to an Authentication Server 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 Node from one of many in multiple technology domains. As an example an Authentication Server 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 Server.

(b) **OBF Client Identifier**:

It is an identifier of the OBF Client or the Connected 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 interface 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.

7.3.2 Subscription Information

Subscription information between a User and its home network contains the User's private entity identifier (e.g., MSISDN), the basic key material (e.g., a shared secret or a public-key certificate) and its lifetime, entity service permission flag (i.e., whether it is allowed to request a specific service), the supported authentication mechanism(s) (e.g., HTTP authentication and key agreement, Diffie-Hellman based authentication mechanisms, a biometric authentication mechanism, etc.), and the authentication inquiring and key generation mechanism (e.g., GBA, Kerberos, Mediation), etc. Subscription information between an ASP and a Network Operator contains the ASP's identity information and public entity identifier (e.g., UID) according to the service, optionally the basic key material (e.g., a shared secret or a public-key certificate) and its lifetime, entity service permission flag (e.g., whether it is allowed to provide a specific service), the supported authentication mechanisms (e.g., certificate-based TLS authentication mechanism, PSK-TLS, IPSec), and the authentication inquiring and key generation mechanism (e.g., GBA, Kerberos, Mediation), etc.

The subscription information related to the user and its authentication server is delivered to the OBF Client from the Resource Server via the Authentication Server 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.

In addition, the subscription information contains a mechanism for key selection, which is used in the OBF Client to mandate the usage of either the Connected Device-based key or the external Secure Element based key or both.

7.3.3 OBF_Token

The OBF_Token is a session key, independently generated in the OBF Client as well as in the Authentication Server.

The OBF_Token is generated by the OBF Client using the security parameters negotiated as part of the bootstrapping process.

The OBF_Token is used for establishing a secure association between the Connected Device and the Application.

The timestamp of the OBF_Token is synchronised and controlled by the Authentication Server.

8 Requirements

8.1 General Requirements for the OBF

The following general requirements are applicable to the OBF:

- OBF Authentication Server is required to be accessible over the public Internet.
- OBF is required to enable identification and authentication of all connected device regardless of the network technology the device is connected to.
- OBF is required to enable the provisioning of access to applications offered by the ASP.
- OBF is inter-operable and independent of underlying network technologies, and is required to:
    - expose the inherent security capabilities of any underlying network technology;
    - work in situations where there are multiple heterogenous networks simultaneously available; and
    - support the numbering and identifiers related to underlying network technologies.
- OBF is required to work for all types of providers and users, and supports:
    - the on-boarding of various application developers and service providers; and
    - all types of devices regardless of uses cases and operating systems.
- OBF is required to protect the privacy of the user identification information.

8.2 Requirements on the OBF Nodes

8.2.1 General Requirements on the OBF Nodes

All the OBF Nodes are required to:

- support transferability such that a user is free to choose services from any Network Operator, IoT SP or Application Service Provider; and
- enable identification and authentication of trusted nodes and clients.

8.2.2 Requirements on the OBF Client

The OBF Client is required to:

- interact with the secure element of the connected 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 servers, allowing services of only one authentication server at a given point in time;
- generate and / or retrieve the OBF identifier as per the selected authentication server;
- 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; and
- protect the use of the network subscriber identity against discovery and misuse.

8.2.3 Requirements for the OBF Resource Server

The OBF Resource Server is required to:

- Manage the keying material used in the bootstrapping mechanism;
- Provide methods for provisioning of the applications; and

- Support the protocols required over the reference point RPAA.

8.2.4 Requirements for the OBF Authentication Server

The OBF Authentication Server is required to:

- Support standard AKA protocols;
- Manage the lifecycle of keys as per the agreed AKA protocol;
- Establish the OBF Identifier and communicate that to the OBF Client;
- Establish the OBF subscription information in conjunction with the Resource Server and communicate it to the OBF Client;
- generate the OBF_Token as per the subscription information negotiated with an application;
- limit the applicability of the OBF_Token to a specific application;
- maintain the list of authorized applications and the related subscription parameters; and
- protect the use of the network subscriber identity against discovery and misuse.

8.3 Requirements for the Reference Points

8.3.1 Requirements for the RPAA

The Reference Point RPAA is required to:

- secure the communication between the Application Server and the Authentication Server;
- allow the transfer of OBF_Token; and
- allow the transfer of user’s subscription information to enable access control policies between connected devices and applications.

8.3.2 Requirements for the RPAR

The Reference Point RPAR is required to:

- permit identification and mutual authentication between the Authentication Server and Resource Server;
- allow the transfer of security parameter required for bootstrapping; and
- allow the transfer of subscription information to establish the access control policies between connected devices and applications.

8.3.3 Requirements for the RPCA

The Reference Point RPCA is required to:

- identify and mutually authenticate the connected device, the OBF Client and the Authentication Server;
- support the bootstrapping process between the OBF Client and the Authentication Server;
- transfer the identification of the OBF Client using the OBF Identifier; and
- transfer the Security Parameters to the OBF Client.

8.3.4 Requirements for the RPDS

The Reference Point RPDS is required to:

- support the application-specific protocol between the connected device and the application;

- send the indication from the application to the connected device that OBF_Token is required prior to connecting to the application; and
- support the use of the OBF_Token for creating the secure association between the connected device and the application.

8.4 Other Requirements

The OBF_Token is required to:

- bind the user identity to the keying material used in the Reference Points;
- be the globally unique identifier of realm of the OBF in which it is issued;
- create the secure association between the connected device and the application; and
- support any underlying network technology.

9 Pre-requisites for the Connected Devices and Application Servers

9.1 Pre-requisites for the Connected Devices

In order to use the OBF, the connected devices have to:

- host a secure element;
- host the OBF Client in the device or its connectivity element (e.g. SIM card);
- support the application specific protocol over the reference point RPDS such as HTTP, MQTT, Web Sockets or COAP;
- support HTTP Digest AKA protocol and optionally others as required by the underlying network technology or application;
- initiate the bootstrapping process when the application indicates the requirement; and
- discover, identify, address and connect to the authentication server relevant to the realm of the connected device.

9.2 Pre-requisites for the Application

After the bootstrapping has been completed, the connected device and the application server can run an application specific protocol, where the authentication of messages will be based on the OBF_Token generated during the mutual authentication between the OBF Client and the Authentication server.

In order to use the OBF, the applications have to:

- indicate to the connected device the need for OBF_Token if it attempts to connect to the application without one;
- be able to locate and communicate securely with the user's authentication server;
- acquire the OBF_Token to secure the interactions with the connected device;
- implement Diameter/HTTP proxy functionality to act as a proxy towards the Authentication Server of the realm in which the user is bootstrapped;

- acquire the user’s security parameters from the Resource Server via the Authentication Server; and
- configure the key lifetime and validity settings.

10 Capabilities of the OBF

10.1 General Capabilities of the OBF

The OBF has the capability to:

- expose the inherent security capabilities of any underlying network technology to allow bootstrapping of clients and devices for secure access to trusted services;
- use in the context of any underlying network technology layer, by any connected device via any network operator;
- support the numbering and identifiers related to underlying network technologies;
- work in situations where there are multiple heterogeneous networks simultaneously available; and
- provision access to applications offered by ASPs.

10.2 Capabilities of the OBF Nodes

10.2.1 General capabilities of the OBF Nodes

All the OBF Nodes are capable of identification and authentication of trusted nodes and clients within an OBF realm.

10.2.2 Capabilities of the OBF Client

The OBF Client has the capability to:

- interface with the connected device and its secure element;
- access the keys / key stores in the secure element of the connected device or the connectivity element;
- support the required application protocol and AKA in the reference point RPCA;
- support the application protocol in the reference point RPDS and initiate the bootstrapping process if indicated by the application;
- support the discovery of the OBF Authentication Server;
- generate and / or retrieve the OBF identifier as per the selected authentication server;
- generate the OBF_Token as per security parameters negotiated during the bootstrapping process;
- securely store the identifiers, subscription information and the OBF_Token; and
- provide user’s identity only upon user’s explicit consent / established intent.

10.2.3 Capabilities of the Resource Server

The OBF Resource Server has the capability to:

- store the pre-shared keys or certificates corresponding to the connected device;

- manage the keys and lifecycle of the keying material as per the agreed AKA protocol;
- provision the users and applications with the required application security parameters;
- respond to the authentication server over DIAMETER with the authentication vector and user's security parameters such as the key lifetime and user identities;
- enable addition / deletion of authorized connected devices / users through standardized API or user interfaces;
- enable delegation / revocation of access control rights to authorised OBF Clients through standardized API or User Interfaces;
- enable addition / deletion of authorized application providers / applications through standardized API or user interfaces; and
- enable provisioning and de-provisioning of authorized users of application through standardized API or user interfaces.

10.2.4 Capabilities of the Authentication Server

The Authentication Server has the capability to:

- handle AKA protocols such that it can support the one used by the underlying network technology layer;
- manage the lifecycle of keys as per the agreed AKA protocol;
- configure and communicate the format of the OBF Identifier to the OBF Client;
- configure the OBF subscription information in conjunction with the Resource Server and communicate that to the OBF Client;
- generate the OBF_Token as per the subscription information specific to an application, and limit the applicability of the OBF_Token to a specific application; and
- maintain the list of users, authorized applications and the corresponding OBF_Tokens.

10.3 Capabilities of the Reference Points

10.3.1 Capabilities of the RPCA

The Reference Point RPCA has the capability to:

- to establish the identity of the OBF Client of a connected device to the Authentication Server;
- use the agreed AKA for authentication between authentication server and the OBF Client;
- transfer the OBF Identifier from the Authentication Server to the OBF Client;
- transfer the OBF_Token from the authentication server to the OBF Client; and
- establish shared keys and the key lifecycle management process between the OBF Client and Authentication Server.

10.3.2 Capabilities of the RPDS

The Reference Point RPDS has the capability to:

- support the application protocol agreed between the connected device and the application;
- allow the application to indicate to the client the necessity for the OBF authentication;

- enable the negotiation and selection of the key between the client and server part of the Application;
- use a security protocol identifier as required by the underlying network technology layer;
- allow the application to signal to the client regarding lifecycle management of keys; and
- enable the use of the OBF_Token for securing the association between the client and server part of the Application.

10.3.3 Capabilities of the RPAA

The Reference Point RPAA has the capability to:

- support the DIAMETER [b-RFC 6733] and [b-RFC 7155] protocol;
- enable secure communication between the Authentication Server and the Application Server;
- allow the Application Server 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 Server;
- allow the Authentication Server to verify that the Application Server is authorized to obtain the identifiers, key material and subscription information for a user;
- allow the Application server to indicate to the Authentication Server the single application or several applications for which it requires user identity and security parameters;
- allow the Application server to obtain a selected set of application-specific user security parameters;
- allow the transfer of the OBF_Token from the Authentication Server to the Application Server; and
- allow the Application to indicate to the Authentication Server the protocol identifier of the RPDS security protocol for which it requires the key material.

10.3.4 Capabilities of the RPAR

The Reference Point RPAR has the capability to:

- support the DIAMETER [b-RFC 6733] and [b-RFC 7155] protocol for mutual authentication of the Authentication Server and Resource Server;
- allow the Authentication Server to request bootstrapping information for specific users; and
- allow the Resource Server to send the user's security parameters to the Authentication Server.

10.4 Other Capabilities

The OBF_Token, used for binding the user identity to the keying material used in the Reference Points, is globally unique and has the capability to:

- identify the realm of the OBF in which it is issued;
- serve as a temporary identifier of the user;
- be used as a key identifier in protocols used in reference point RPCA and RPDS;
- enable the application server to detect and address the authentication server that has sponsored the OBF_Token; and

- be in a format that is usable by the underlying network technology layer bootstrapping capabilities.

11 OBF Functions

The Functions implemented in the Secure Element, Device and the Servers, which are involved in the Authentication process, are shown in the diagram below.

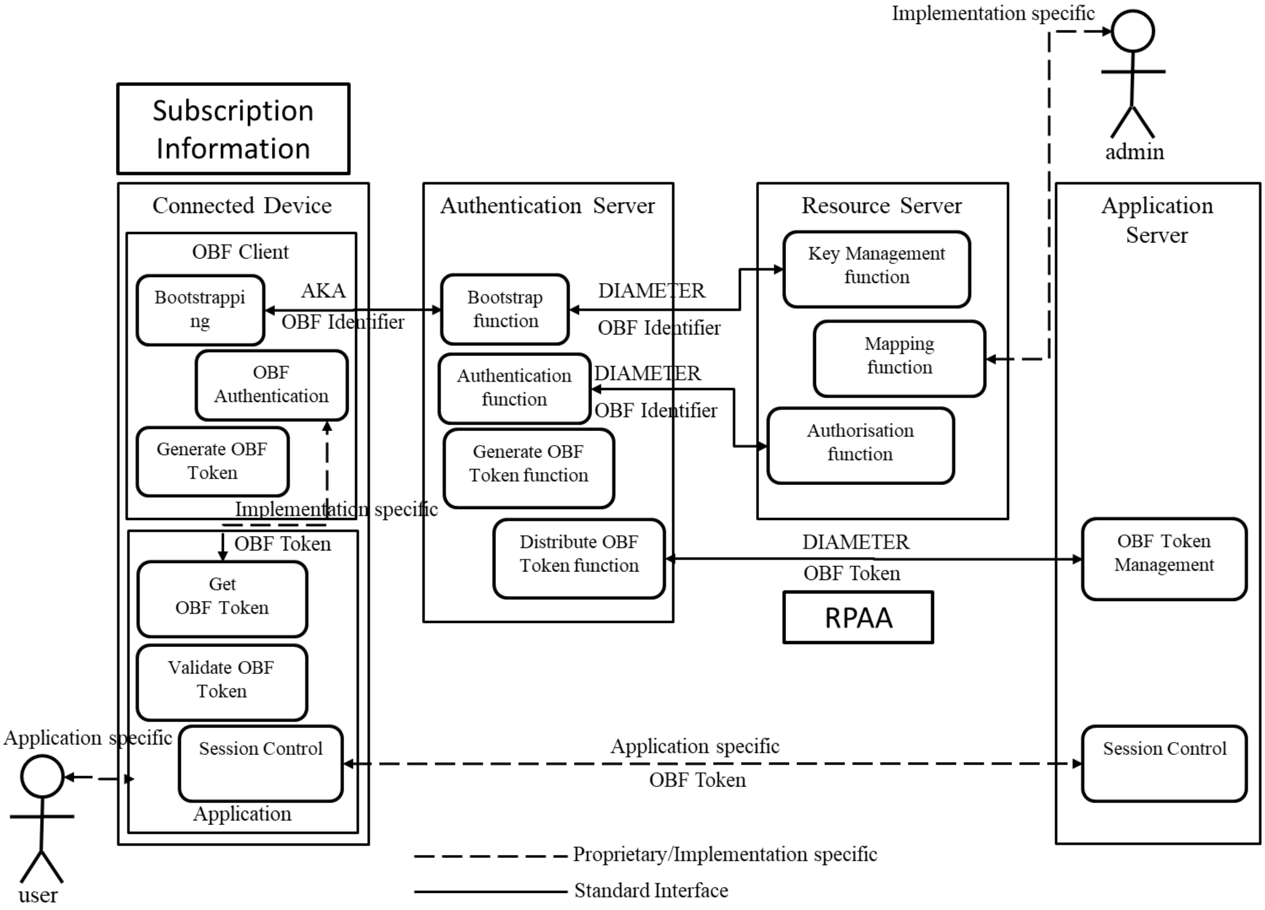

Figure 3: OBF Functions

The OBF functions are described below.

11.1 Authentication Server functions

11.1.1 Authentication Function

This function is hosted in the network of the Network Operator/IoT SP under the control of the issuer of the Secure Element. The Authentication Server, Resource Server, and Secure Element participate in Authentication procedure in which a shared secret is established between the Authentication Server and the OBF Client hosted in the Secure Element by running the bootstrapping procedure over the reference point RPCA.

The Authentication Function resides in the Authentication server and initiates authentication process of the Connected Device (or Client?) based on authentication information available from the underlying network technology.

11.1.2 Bootstrapping Function

This function is hosted in the Authentication Server of the Network Operator/ IoT SP and in the OBF Client of the Connected Device. The Authentication Server creates a new registration for the Connected Device by way of establishment of secret keys for secure communication. The new association provides for mutual authentication of Connected Device and Applications hitherto unknown to each other.

11.1.3 Generate OBF_Token Function

This function is responsible to generate the OBF token after bootstrapping has successfully been completed by using the keying material and algorithms agreed on.

11.1.4 Distribute OBF_Token Function

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

11.2 Resource Server functions

11.2.1 Key Management Function

This function resides in the OBF Client and in the Resource Server. The function provides the mechanisms for management and association of keys and algorithms between the Resource Server and the OBF Client.

11.2.2 Authorisation Function

The Authorisation Function resides in the Resource Server and validates if the OBF Client has the right to use the authentication for the requested application/service. The Authorisation Function hosts the repository of registered Third Party applications that can be permitted for use by the Device / User. The Authorisation Server maps the Application identities to the OBF_Token issued to the User by the Authentication Function.

11.2.3 Mapping Function

The mapping function is an administrative function to map users, Connected Devices, and Application together. This can be done on an individual level, or based on the agreement between the user and the OBF provider.

NOTE: The details of this function is not in the scope of this recommendation.

11.3 OBF Client functions

11.3.1 Bootstrapping Function

See 11.2

11.3.2 Authentication Function

See 11.1

11.3.3 Generate OBF_Token Function

See 11.1.3

11.4 Application Client functions

11.4.1 Validate OBF_Token

When the Application Client is started, or wants to communicate securely with the application server, the current OBF_Token is validated to ensure the lifetime of the token has not expired. If the lifetime has expired, or if no current OBF_Token is available the Application Client will use the Get OBF_Token Function to get a new OBF_Token.

11.4.2 Get OBF_Token Function

If the Application Client does not have a currently valid OBF_Token this function is used to initiate the bootstrapping of the device by calling the OBF Client, and after completion a new OBF_Token is returned.

11.4.3 Session Control Function

The Session Function is application specific, but utilizes the OBF_Token to initiate a secure session towards the Application Server. The session can be TLS PSK, Kerberos, IPSEC etc.

11.5 Application Server functions

11.5.1 OBF_Token Management Function

When an OBF_Token has been generated it must be distributed to the Application Server, the OBF_Token Management Function shall receive and store the OBF_Token for future sessions.

11.5.2 Session Control Function

See 11.4.3

12 Mechanisms and Workflows

In the OBF bootstrapping and authentication process using pre-shared keys, the same long-term master key is stored in the secure element of the connected device and in the Resource Server (generation and loading of the long-term master key are implementation specific and proprietary).

When requested by Authentication Server for bootstrapping an OBF Client, the Resource Server generates a time-limited master key which is shared between the Authentication Server and the OBF Client of the connected device. Optionally, the OBF Client can generate the time-limited master key in the connected device and store it in the secure element of the connected device.

When requested by an Application, the Authentication Server and the OBF Client generate an application specific session key using the time-limited master key called the OBF_Token.

The Authentication Server shares the OBF_Token with the Application Server.

The Authentication Server also informs the Application Server how to invoke the OBF Client resident on the connected device.

The OBF_Token is used in the mutual authentication between the client part of the application on the connected device and the server part of the application.

12.1 Authentication Workflow

The Authentication Workflow is meant for a User that would like to use a Service or an Application that can benefit from the OBF Authentication.

When a User requires to access an Application from the Connected Device, or the Application requires to exchange data with the Connected Device, it signals to the OBF Client the requirement to use the bootstrap framework for authentication. This process in accomplished in the following steps:

1. Bootstrapping is initiated, if it has not been executed previously. Please see section 12.2 below;
2. The User request towards the Application server is executed and the application uses a challenge-response mechanism to identify the User and the user responds to the challenge-response mechanism used by the Application; and
3. The OBF Client uses the OBF_Token, which is used to set up a secure connection using TLS for any data exchange between the Connected Device application and the Application Server.

NOTE – The mechanism to invoke the OBF Client 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 4):

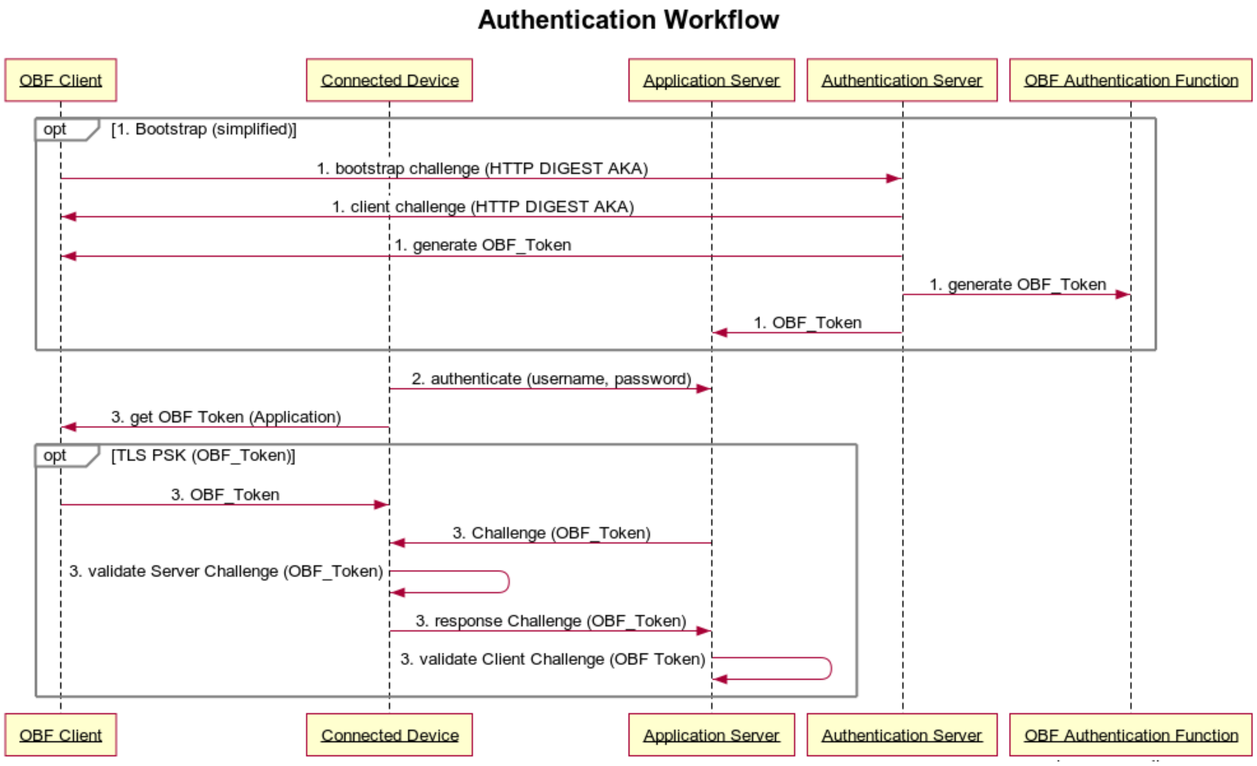
Authentication Workflow
OBF Client
Connected Device
Application Server
Authentication Server
OBF Authentication Function
opt
[1. Bootstrap (simplified)]
1. bootstrap challenge (HTTP DIGEST AKA)
1. client challenge (HTTP DIGEST AKA)
1. generate OBF_Token
1. generate OBF_Token
1. OBF_Token
2. authenticate (username, password)
3. get OBF Token (Application)
opt
[TLS PSK (OBF_Token)]
3. OBF_Token
3. Challenge (OBF_Token)
3. validate Server Challenge (OBF_Token)
3. response Challenge (OBF_Token)
3. validate Client Challenge (OBF Token)
OBF Client
Connected Device
Application Server
Authentication Server
OBF Authentication Function

Figure 4: Authentication Workflow

12.2 Key Management during bootstrap Flow

The shared key that exists on both the Secure Element, and in the Key Management Function of the Resource Server, is used to authenticate the OBF Client with the Authentication Server. Session Keys are used for securing the communication between the Connected Device and an Application. This process in accomplished in the following steps:

1. The Authentication Server will validate the client in the bootstrapping stage;
2. The Authentication Server and the OBF Client will mutually challenge each other to validate credentials;

3. The Resource Server validates if the User has the right to use the authentication for the given Application;
4. When the mutual authentication has completed the OBF Client and Authentication Server agree on the OBF_Token; and
5. The OBF_Token is provided to the Application Server 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 5):

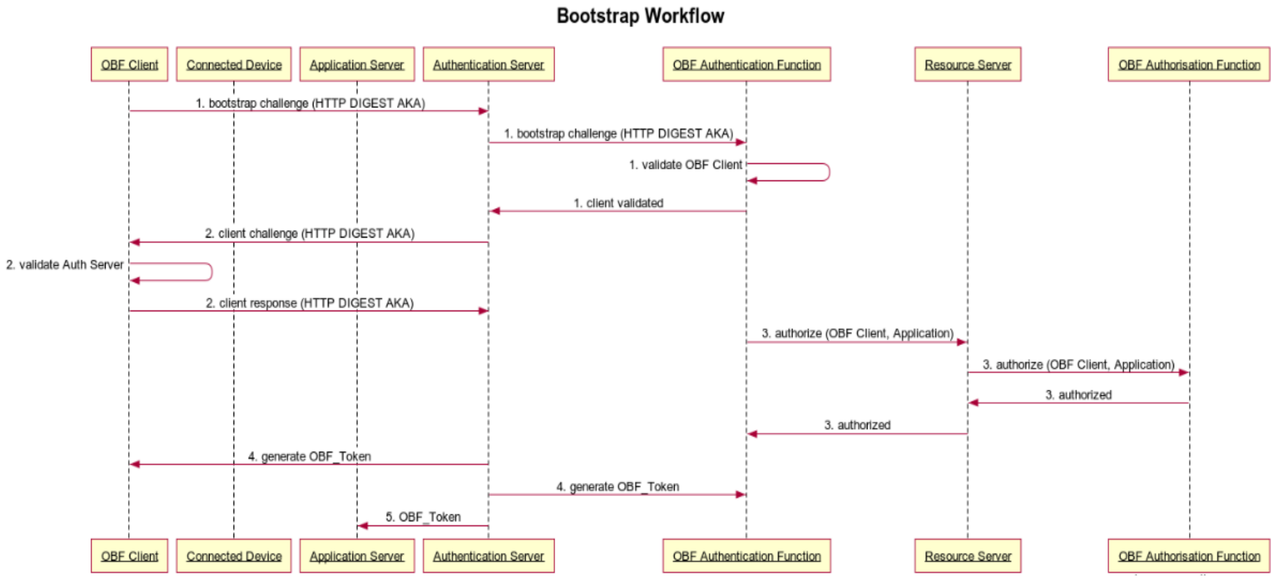
Bootstrap Workflow
OBF Client
Connected Device
Application Server
Authentication Server
OBF Authentication Function
Resource Server
OBF Authorisation Function
1. bootstrap challenge (HTTP DIGEST AKA)
1. bootstrap challenge (HTTP DIGEST AKA)
1. validate OBF Client
1. client validated
2. client challenge (HTTP DIGEST AKA)
2. validate Auth Server
2. client response (HTTP DIGEST AKA)
3. authorize (OBF Client, Application)
3. authorize (OBF Client, Application)
3. authorized
3. authorized
4. generate OBF_Token
4. generate OBF_Token
5. OBF_Token
OBF Client
Connected Device
Application Server
Authentication Server
OBF Authentication Function
Resource Server
OBF Authorisation Function

Figure 5: Bootstrapping Workflow

12.3 Changing of Authentication Provider Flow (Asymmetric keys)

A User may change the Connectivity Provider, but still may want to continue the use of Services which are supported by the OBF Authentication. The Authentication Provider may be changed as per the mechanism defined below:

1. User requests new Authentication Services Provider for its services;
2. The new Authentication Services Provider completes the Machine KYC;
3. The new Authentication Service Provider provides its Public Key to the old Authentication Service Provider with a request to transfer the User’s Account to the new Authentication Service Provider;
4. The old Authentication Services Provider uses its Private Key to update the Secure Element of the User with the Public Key of the New Authentication Services Provider;
5. Upon successful confirmation of the transfer the new Authentication Services Provider informs the Application Services Providers about the change in the OBF_Token for a User; and

6. The Application Service Provider uses the new OBF_Token along with embedded connectivity identity to verify the User.

The Process is described in the diagram below (Figure 6):

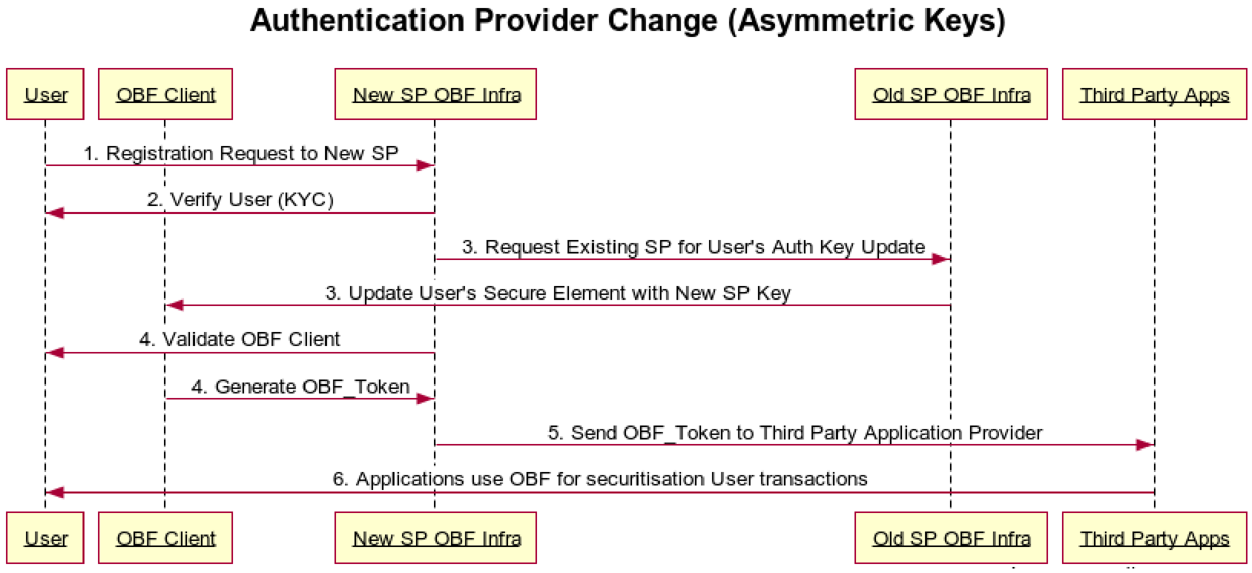
Authentication Provider Change (Asymmetric Keys)
User
OBF Client
New SP OBF Infra
Old SP OBF Infra
Third Party Apps
1. Registration Request to New SP
2. Verify User (KYC)
3. Request Existing SP for User's Auth Key Update
3. Update User's Secure Element with New SP Key
4. Validate OBF Client
4. Generate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

Figure 6: Authentication Provider Change (Asymmetric keys)

12.4 Changing of Authentication Provider Flow (Symmetric keys)

The User of the service has to approach the new IoT Service Provider / Mobile Operator for enabling the use of the Authentication Services. The Steps for such a transfer are described below:

1. User requests new Authentication Services Provider for its services;
2. The new Authentication Service Provider requests existing Authentication Service Provider for User’s Shared Keys;
3. The new Authentication Services Provider uses the old key to update the Secure Element with a new key following the Machine KYC;
4. The new Authentication Services Provider informs the User and the old Authentication Services provider of the successful confirmation of the transfer to the new Authentication Services Provider;
5. Upon successful confirmation of the transfer the new Authentication Services Provider informs the Application Services Providers about the change in the OBF_Token for a User;
6. The Application Service Provider uses the new OBF_Token along with embedded connectivity identity to verify the User.

The Process is described in the diagram below (Figure 7):

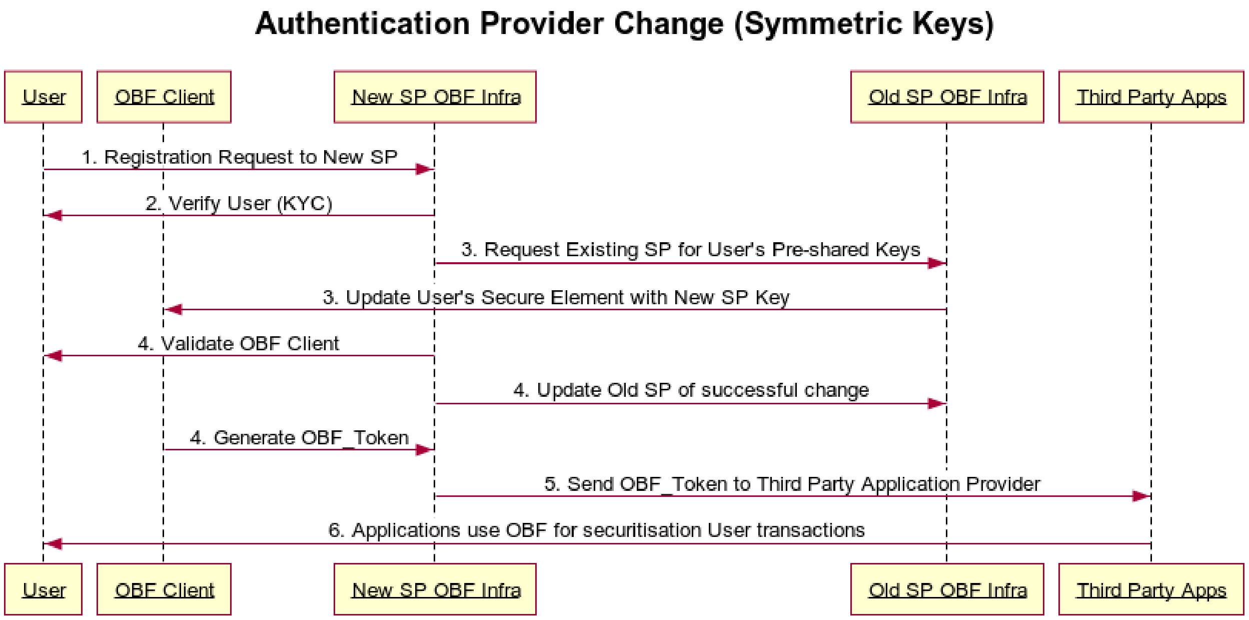
Authentication Provider Change (Symmetric Keys)
User
OBF Client
New SP OBF Infra
Old SP OBF Infra
Third Party Apps
1. Registration Request to New SP
2. Verify User (KYC)
3. Request Existing SP for User's Pre-shared Keys
3. Update User's Secure Element with New SP Key
4. Validate OBF Client
4. Update 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 7: Authentication Provider Change (Symmetric Keys)

Appendix I

Explanation of the use case example

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

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

I.1 Background and Diversified multi-stakeholder eco system

The Ecosystem comprises of the following Actors

a. Network Operator or IoT SP: Supplier of the SIM and Secure Element

b. Device Manufacturer – manufacturer of the Device with the embedded SIM / Secure Element

c. Vehicle Manufacturer – manufactures of the vehicle with the embedded device, SIM and Secure Element

d. Buyer – the entity or person that pays for the Vehicle

e. Application Provider – the entity that provides the Application for registration, tracking and transfer of the vehicle

f. Certifying Agency – the entity that Certifies the Device and the Application

g. Trust Centre – the Agency responsible for the registration and enforcement of Vehicle rules, typically a State actor

I.1.1 Background

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

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

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

I.1.2 Diversified multi-stakeholder eco system

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

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

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

I.2 Use case

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

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

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

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

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

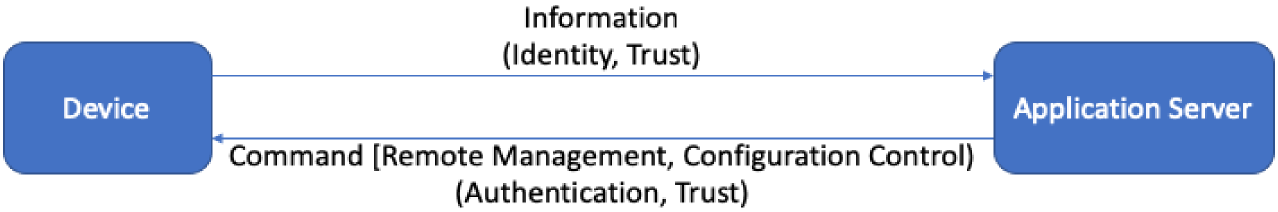

Figure I.1: Device-Application Server Communication

I.2.1 Important consideration for security

Following are important consideration for security implementation:

1. The tamper proof identity of the SIM / Secure Element (IccID / EID) is used as the primary identifier for the connected device

2. Appropriate mechanisms are followed for the generation and sharing of Security key between the SIM / Secure Element and the Authentication Server
3. The Network Application Function (NAF) and the OBF interact securely following the standards prescribed by 3GPP GAA.

I.2.2 Functions required

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

I.2.2.1 Device Functions

(a) Validate Checksum Function

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

(b) Decrypt Encrypted Server Data Function

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

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

(c) Encrypted Device Data Function

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

I.2.2.2 Secure Element Functions

(a) Decrypt Data Function

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

(b) Encrypt Device Data Function

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

I.2.2.3 Application Server Functions

(a) Key Import Function

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

(b) Decrypt Device Data Function

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

(c) Encrypt Server Data Function

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

I.2.3 Application Server to Device flow (Sample)

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

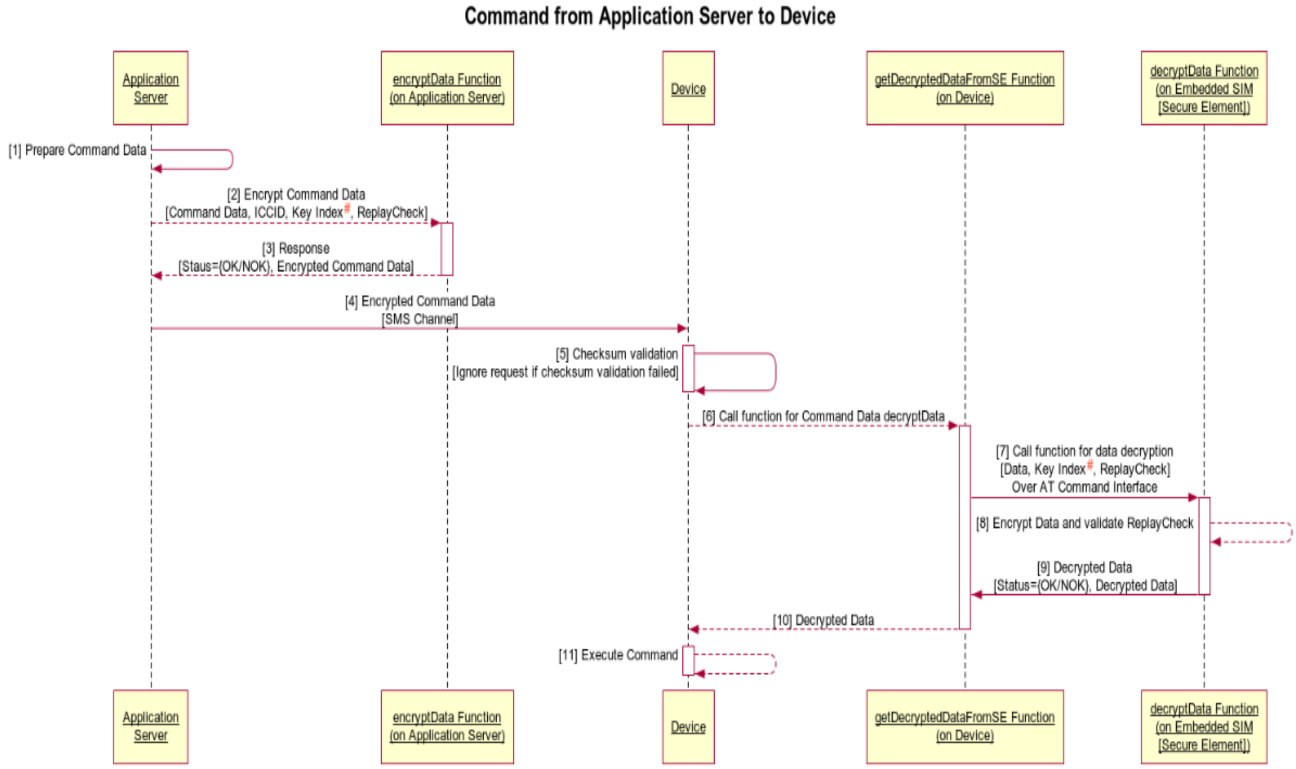
Command from Application Server to Device
Application Server
encryptData Function (on Application Server)
Device
getDecryptedDataFromSE Function (on Device)
decryptData Function (on Embedded SIM [Secure Element])
[1] Prepare Command Data
[2] Encrypt Command Data [Command Data, ICCID, Key Index#, ReplayCheck]
[3] Response [Staus=(OK/NOK), Encrypted Command Data]
[4] Encrypted Command Data [SMS Channel]
[5] Checksum validation [Ignore request if checksum validation failed]
[6] Call function for Command Data decryptData
[7] Call function for data decryption [Data, Key Index#, ReplayCheck] Over AT Command Interface
[8] Encrypt Data and validate ReplayCheck
[9] Decrypted Data [Status=(OK/NOK), Decrypted Data]
[10] Decrypted Data
[11] Execute Command
Application Server
encryptData Function (on Application Server)
Device
getDecryptedDataFromSE Function (on Device)
decryptData Function (on Embedded SIM [Secure Element])

Figure I.2: Application Server to Device Communication Flow

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

Bibliography

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

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

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

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