INTERNATIONAL TELECOMMUNICATION UNION

TELECOMMUNICATION STANDARDIZATION SECTOR

STUDY PERIOD 2017-2020

SG13-TD409/WP3

STUDY GROUP 13

Original: English

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

TD

Source:	Editors
Title:	Draft Recommendation ITU-T Y.OBF_Trust: “Open Bootstrap Framework enabling trustworthy networking and services for distributed diverse ecosystem” (output of e-meeting, 6 April 2020)
Purpose:	Information

Contact:	Abhay Shanker Verma TEC India	Tel: + 91 9999554900 E-mail: as.verma@gov.in
Contact:	Ranjana Sivaram TEC India	Tel: +919868136990 E-mail: ranjana.sivaram@gov.in
Contact:	Sharad Arora Sensorise Digital Services Pvt Ltd India	Tel: +91 9212109999 E-mail: sharad.arora@sensorise.net

Keywords: Bootstrapping; IoT; IoT Service Provider; OBF; OBF Proxy; OBF_Token; Open Bootstrap Framework; Trust Framework

Abstract: This document contains the updated Recommendation ITU-T Y.OBF_trust “Open Bootstrap Framework enabling trustworthy networking and services for distributed diverse ecosystems” based on the discussion at interim e-meeting of Q16/13 based on the contribution C91R2.

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

No.	Source	Contribution title and proposals	Agreements
C91R2	India	Draft Recommendation ITU-T Y.OBF_trust: “Open Bootstrap Framework enabling trustworthy networking and services for	The contribution was agreed.

		distributed diverse ecosystems” Proposals: This document proposes to update the draft Recommendation ITU-T Y.OBF_Trust in accordance with the changes carried out in track mode.	

Draft new Recommendation ITU-T Y.OBF_trust

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

Summary

This Recommendation describes an Open Bootstrap Framework (OBF), which includes an OBF Client, an OBF Authentication Server, an OBF Resource Server and four Reference Points. It unfolds a bootstrapping architecture and a description of the OBF elements mechanisms and workflows for the mutual authentication between Connected Devices, Applications and Service Providers.

The objective of the OBF is to provide security bootstrapping to devices for the purpose of extending trustful services to any Application/ Service Provider by re-using the Secure Element and trustful networking capabilities of the network technology layer.

The Recommendation is relevant to Network Operators, IoT Service Providers and Applications/ Services Providers for deployment of secure 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

Page

Draft new Recommendation ITU-T Y.OBF_trust

Open Bootstrap Framework enabling trustful 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

- A Concept that extends the use of embedded Secure Elements and Keys, originally intended for Operator Services, to be used for creating secure associations for Applications provided by Third Party Service Providers;
- An Open Bootstrap Framework with definitions of Nodes and Reference Points; and
- A set of functions, mechanisms and workflows for securitising the interactions between the stakeholders in the physical space and the services in the cyber space.

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.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-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. **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.

3.1.9. **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.

3.2 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. **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.

3.2.5. **OBF:** A trust framework for extending the security capabilities of a network technology layer to benefit Third-Party Connected Devices and Applications.

3.2.6. **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 Server.

3.2.7. **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 e.g. MNO.

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

3.2.9. **Third Party**: An entity other than the Mobile Network Operator or the M2M Service Provider, which consumes the security capabilities of a network for providing trust for applications and / or services offered to the end users.

3.2.10. **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:

BSF	Bootstrapping Server Function
COAP	Constrained Object Authentication Protocol
eUICC	Embedded UICC
EID	eUICC-ID
HLR	Home Location Register
HTTP	Hyper Text Transfer Protocol
ICT	Information and Communication Technology
IoT	Internet of Things
IoT SP	IoT Service Provider
KEK	Key Encryption Key
KMS	Key Management System
KYC	Know Your Customer
M2M	Machine to Machine
M2M SP	M2M Service Provider
MNO	Mobile Network Operator
MQTT	Message Queue Telemetry Transport
NAF	Network Application Function
OBF	Open Bootstrap Framework
PSK	Pre-shared Key
SE	Secure Element

SIM	Subscriber Identification Module
SLF	Subscriber Locator Function
TEE	Trusted Execution Environment
TLS	Transport Layer Security
TSP	Telecom Service Provider, see also MNO
UICC	Universal Integrated Circuit Card

5 Conventions

In this Recommendation, requirements are classified as follows:

- The keywords "**is required to**" indicate a requirement 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 "**can optionally**" and "**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

The OBF uses a unique identity in a tamper resilient hardware that can act as a root of trust, providing the required identity for authentication of remote and dispersed devices, applications and actors in an ICT enabled business value chain. By adding the required Key Management, Authentication and Authorization functions, a bootstrapping framework is defined that makes it possible for any application and service provider to provide a higher degree of security to the User and Services.

A reference model for such an Open Bootstrap Framework (OBF) is defined below.

6.1 OBF Reference Architecture

The elements of the proposed OBF reference model are shown in the diagram below.

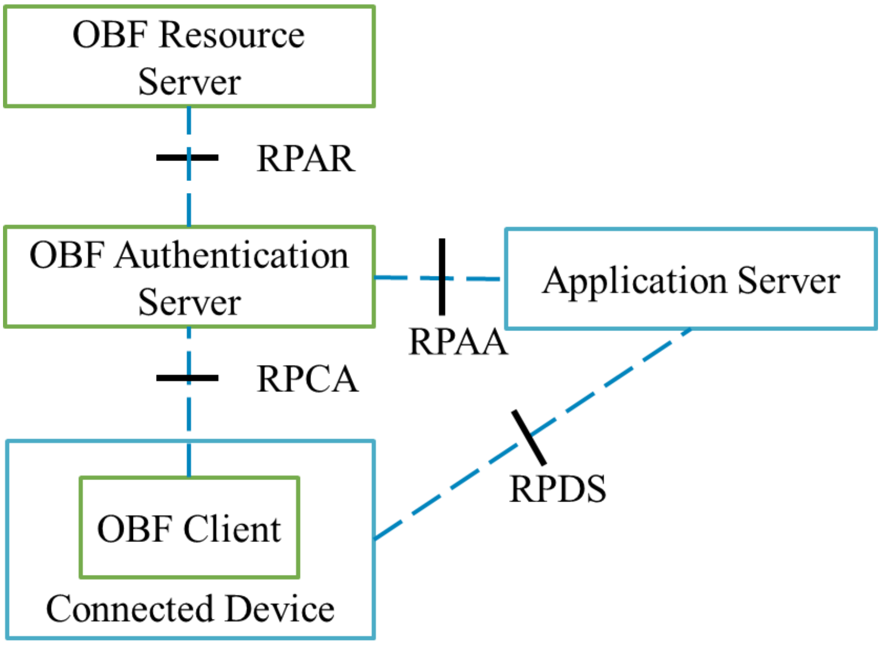

Figure 1: OBF Reference Architecture

The elements of the reference architecture consist of three nodes and four reference points. The Connected Device and the Application are the beneficiaries of the OBF, but not a part of the OBF. The software elements, namely, the OBF Client, OBF Authentication Server and the OBF Authorisation Server are the nodes of the reference model. The nodes interact with each other using four reference points, namely, RPAA, RPAR, RPDS and RPCA.

When the elements of the reference architecture work together with the beneficiary Connected Devices and Applications as per the mechanisms and workflows defined for the OBF, they create a trust framework which is described below.

6.2 OBF Trust Framework

The OBF trust framework is a set of relationships and interactions between actors in the Physical and Cyber space, who use the elements of the OBF, and a set of defined mechanisms and workflows, to achieve the objective of enhanced trust and security.

The concept of the trust framework created by the OBF is shown in Figure 2. The framework shows two domains, namely, the Operator Domain, the Third-Party Service Provider Domain. The trust framework has two operating spaces – the Physical and the Cyber space. The Actors in the OBF trust framework are the Network Service Providers such as the MNOs and IoT SPs; Applications and Services Providers that provision ICT-enabled Services and the User community that buys and uses the ICT-enabled Devices, Applications and Services.

By following the OBF recommendations, the actors in the Physical space are able to derive a trustful relationship between themselves, the Connected Devices and the ICT-Enabled Applications by reusing the security enablement from underlying network technology layer.

The Figure 2 shows the interactions between the elements of the OBF, and the Actors in the Physical and the Cyber Space. The trust framework enables identification, authentication and authorization for the use of Connected Devices and Applications, using mechanisms and workflows which are more fully described in the sections below.

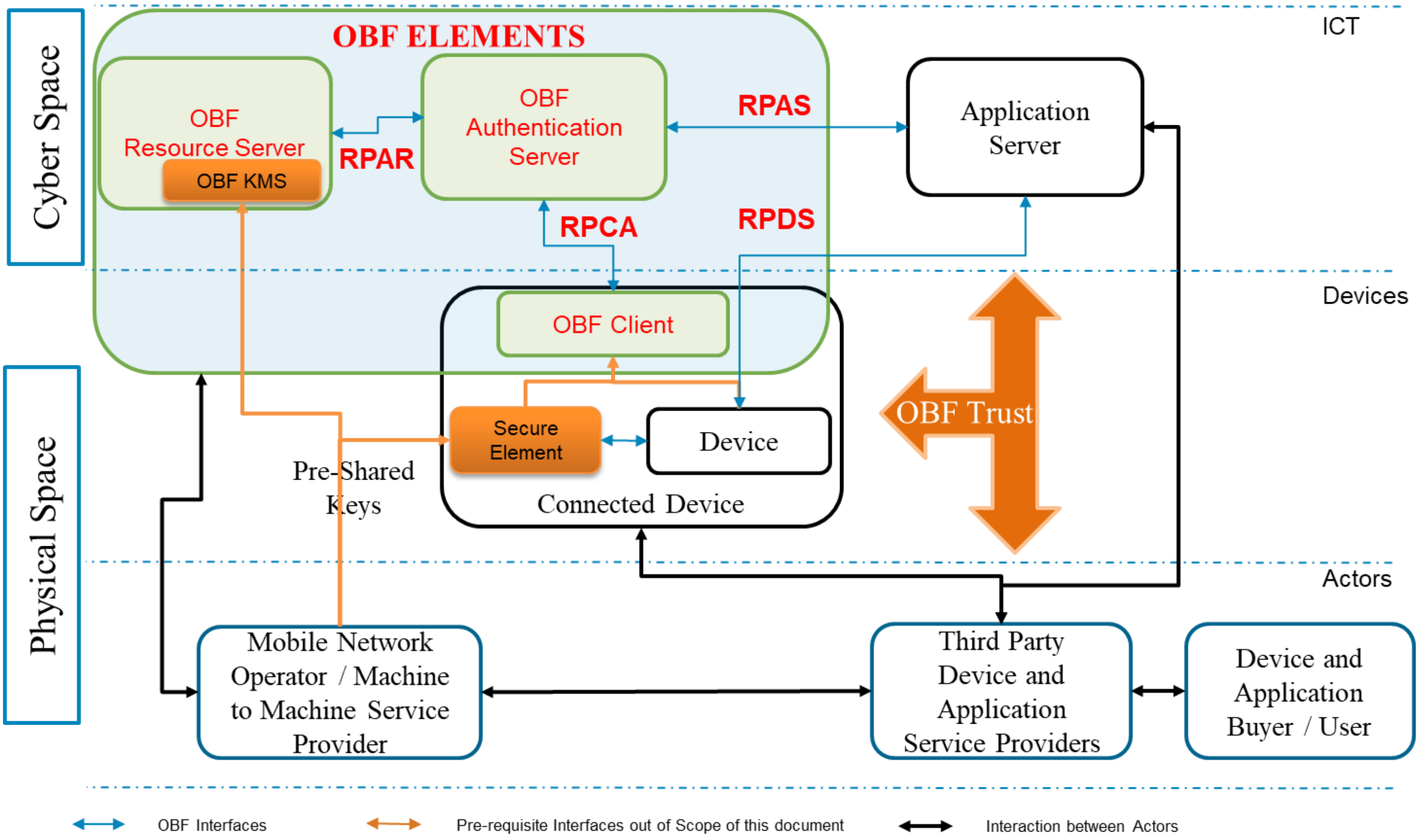
Cyber Space
Physical Space
OBF ELEMENTS
ICT
Devices
Actors
OBF Resource Server
OBF KMS
RPAR
OBF Authentication Server
RPAS
Application Server
RPCA
RPDS
OBF Client
Secure Element
Device
Connected Device
OBF Trust
Pre-Shared Keys
Mobile Network Operator / Machine to Machine Service Provider
Third Party Device and Application Service Providers
Device and Application Buyer / User
OBF Interfaces
Pre-requisite Interfaces out of Scope of this document
Interaction between Actors

Figure 2: Trust Framework using OBF Reference Model

It is not in the scope of this document to specify the processes such as Trusted Application Provisioning as these are controlled by policies and governance mechanisms on the related market, actors and ecosystems.

7 OBF Elements

The OBF specifies three (3) Nodes, four (4) Reference Points and the OBF_Token. Each of these elements is described in the section below.

7.1 OBF Nodes

The OBF specifies 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 the Connected Device Connectivity Element that provides the bootstrapping application and the key 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 and Application Server. The OBF Client is specified and provisioned by the IoT Service Provider or the Mobile Network Operator that is providing the OBF services.

7.1.2 OBF Resource Server

The OBF Resource Server is a network node that provides the key material on the Service Provider side for the bootstrapping service provided by the Authentication Server. The OBF Resource server hosts the required Key Management Systems.

The OBF Resource Server is specified and provisioned by the IoT Service Provider or the Mobile Network Operator that is providing the OBF services.

7.1.3 OBF Authentication Server

The OBF Authentication Server is a network node that mutually authenticates the OBF Client towards the OBF Resource Server, generating in the process, a set of algorithms and keys that are then used for the security of the transactions between the Connected Device and the Application Server that is hosting the Connected Services.

7.2 OBF Reference Points

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

7.2.1 RPAA

The Reference Point between Authentication Server and Application Server. It is used by the Application Server to fetch key material from the Authentication Server. It is also used to fetch application-specific user security settings from the Authentication Server if requested. The recommended protocol to be used over RPAA is RADIUS [b-RFC 2865] with the addition on TLS [b-RFC6614].

7.2.2 RPAR

The Reference Point between OBF Authentication Server and OBF Resource Server. Here the OBF Authentication Server can get the resource rights for a certain Connected Device. The recommended protocol to be used over RPAR is RADIUS [b-RFC 2865].

7.2.3 RPCA

The Reference Point is between the OBF Client hosted in the Secure Element and the OBF Authentication Server. The Reference point provides mutual authentication between the OBF Client in the Secure Element and OBF Authentication Server. It allows the OBF Client in the Secure Element to bootstrap the Connected Device and the Connected Service using session keys. The recommended protocol to be used over RPCA is HTTP Digest protocol [b-RFC7616], the interface between the Connected Device and the Secure Element is as per the specifications of the underlying Network Technology.

7.2.4 RPDS

The Reference Point is between the Connected Device and the Application Server. It carries the application protocol, which is secured using the keys material agreed between OBF Client hosted in the Secure Element and the OBF Authentication Server. The communication protocol between the Connected Device and the Application Server is not in the scope of this recommendation.

7.3 OBF_Token

A session key independently generated in the Connected Device / User Equipment (UE) as well as in the Authentication Server, by the Authentication Function on the Device and Server, respectively, for establishing a secure association between the Connected Device and the Application provided by the Third-Party Service Provider or IoT Service Provider.

OBF_Token shall be derived either from the device or secure element by using device identification, Secure key material, connectivity information, and time stamp/ counters.

8 OBF Functions

The Functions implemented in the Secure Element, Device and the Servers, which are involved in the Authentication process, are as follows:

8.1 The Authentication Function

This function is hosted in the network of the MNO/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 as described in the OBF Authorisation Function below.

8.2 OBF Client Function

A function of the OBF Client hosted in the Connected Device that executes the bootstrapping procedure with the Authentication Server and provides the Connected Device with security association to run bootstrapping procedure.

8.2.1 Connected Device Function

An Application calls this function over the reference point RPDS when an application server requires a bootstrapped security association.

8.2.2 OBF Authorisation Function

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

9 Capabilities of the OBF Nodes

9.1 OBF Nodes

All the OBF Nodes are required to have the capability to support :

- transfer of Connected Devices between Authentication Service Providers such as MNOs, and IoT SPs; and

- functions and workflows as specified herein below.

In addition, the capabilities of individual OBF Nodes are described below.

9.1.1 OBF Client

OBF Client has the capability to initiate the bootstrapping process to create a repository of trusted Connected Devices and the corresponding Authentication Servers.

9.1.2 OBF Resource Server

OBF Resource Server has the capability to:

- create and upload Keys through OBF Key Management System to the OBF Resource Server and the OBF Client, in cases where the underlying Network Technology system requires the creation of keys by an external element;
- ingest keys through OBF Key Management System, where the underlying Network Technology creates the keys;
- register the Resource Servers and the Resource Server Providers (MNOs and IoT Service Providers);
- to register the Application Servers and the Third Party Application Service Providers;
- provision Third Party Application Service provider applications towards Connected Devices;

9.1.3 OBF Authentication Server

OBF Authentication Server has the capability to initiate the bootstrapping process to create a repository of trusted Connected Devices and the corresponding Authentication Servers.

10 Requirements

The OBF may be deployed by an MNO or an IoT SP and used by Third Party Application providers. The requirements for the Open Bootstrap Framework are identified in the clauses below:

10.1 Requirements for the OBF Nodes

All the OBF Nodes are required to have support for:

- Published addressability, access and registration processes for Connected Devices and Applications offered by MNOs, IoT SPs or Third Party Service Providers;
- Inter-operability and transferability such as to provide freedom for the end user or buyer to choose services from any MNO, IoT SP or Third Party Application Service Providers without affecting the Authentication Services offered by the OBF; and
- Compatibility with various underlying Networking Technologies, in order to provide the Authentication and Authorization Services using the global identities, key material and crypto algorithm as per the underlying Network Technology layer.

Apart from the above requirements pertaining to all the nodes, additional requirements of the OBF Client, OBF Resource Server and the OBF Authentication Server are as below.

10.1.1 Requirements for the OBF Client

The OBF Client is required to be capable of interacting with the Secure Element, which may be a part of the Connected Device or the Connectivity Element.

10.1.2 Requirements for the OBF Resource Server

(a) The OBF Resource Server implementation is required to conform to the following:

- Store the identities and credentials of the Connected Devices and the Applications
- Store the mapping of the stakeholders and custodians with the Connected Devices and the Applications; and
- Provide methods for provisioning of the Applications permitted to be accessed by Connected Devices.

(b) The Resource Server Key Management Function and the Secure Element must support commonly used security algorithms.

10.1.3 Requirements for the OBF Authentication Server

The OBF Authentication Server implementation is required to support:

- The use of global identities as per the underlying Network Technology layer without any change; and
- The use of Pre-Shared Keys or Public Key Infrastructure, either as part of the Network Technology layer authentication service or as a standalone OBF Authentication Service provided by an IoT SP.

10.2 Requirements for the Reference Points

10.2.1 Requirements for the RPAA

The OBF RPAA interface requires that the Authentication Server and Application Server implement mechanisms that:

- secure the communication between the Application Server and the Authentication Server; and
- ensure transfer of the OBF_Token from the Authentication Server to the Application Server.

10.2.2 Requirements for the RPAR

The OBF RPAR interface requires that the Resource Server provide the Authentication Server with relevant data to be shared with an Application Server.

10.2.3 Requirements for the RPCA

The OBF RPCA interface requires that:

- the identification of the OBF Client (Secure Element), and the Connected Device that the Secure Element is attached to, is possible to be undertaken by the Authentication server;
- the mechanism for mutual authentication between the Authentication Server and OBF Client (Secure Element) is implemented by the Authentication Server and the OBF Client; and
- the mechanism for transfer of the OBF_Token from the Authentication Server to the Application Server is implemented by both sides.

10.2.4 Requirements for the RPDS

The OBF RPDS interface requires that:

- The OBF Client and the Authentication Server support the HTTP Digest protocol [b-RFC7616];
- The OBF Client has an implementation that allows the OBF Client to communicate with the Secure Element;
- The Third-Party application running on the Connected Device signals to the OBF Client (Secure Element) when it requires to use the OBF; and
- The Application Server and the Connected Device application use the OBF_Token to create new sessions (TLS PSK).

10.3 Requirements for the OBF_Token

The OBF_Token is required to be:

- globally unique;
- usable as a key identifier in protocols used in Reference point RPAA; and
- able to provide adequate information to the OBF Authentication Server to make it capable of detecting the domain and the OBF Resource Server of the Connected Device.

11 Pre-requisites for the Devices and Application Servers

11.1 Pre-requisites for the Devices

Following pre-requisites are required to be fulfilled by the Connected Devices that make use of the OBF:

- Host a Secure Element and have an implementation of the OBF Client in the Connected Device or its Connectivity Element;
- Support for interface between the Connected Device and the Secure Element as per the specifications of the underlying Network Technology; and
- Support for one or more protocols - HTTP, MQTT, Web Sockets or COAP.

11.2 Pre-requisites for the Application Servers

Following pre-requisites are required to be fulfilled by the Application Servers that make use of the OBF:

- Support for one or more protocols - HTTP, MQTT, Web Sockets or COAP, which are used by the Devices in the ecosystem; and
- Have the ability to set local validity conditions of the shared key material according to the local policy; and
- Have the ability to honour lifetime and local validity condition of the shared key material.

NOTE - It is recommended that support for new protocols are added as and when released within the relevant ecosystem.

12 Operations and Mechanisms

The following Operational Workflows are defined for the OBF. However, in the workflows, the details / aspects of Numbering, Identity and Machine KYC management, the Challenge-Response Mechanism adopted for establishment of trust, and the method of session key generation are not covered and are outside the scope of this Recommendation.

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 Application Server, the application 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 TLS secure connection for any data exchange between the Connected Device application and the Application Server.

The Authentication workflow is described in the diagram below (Figure 3 & 4):

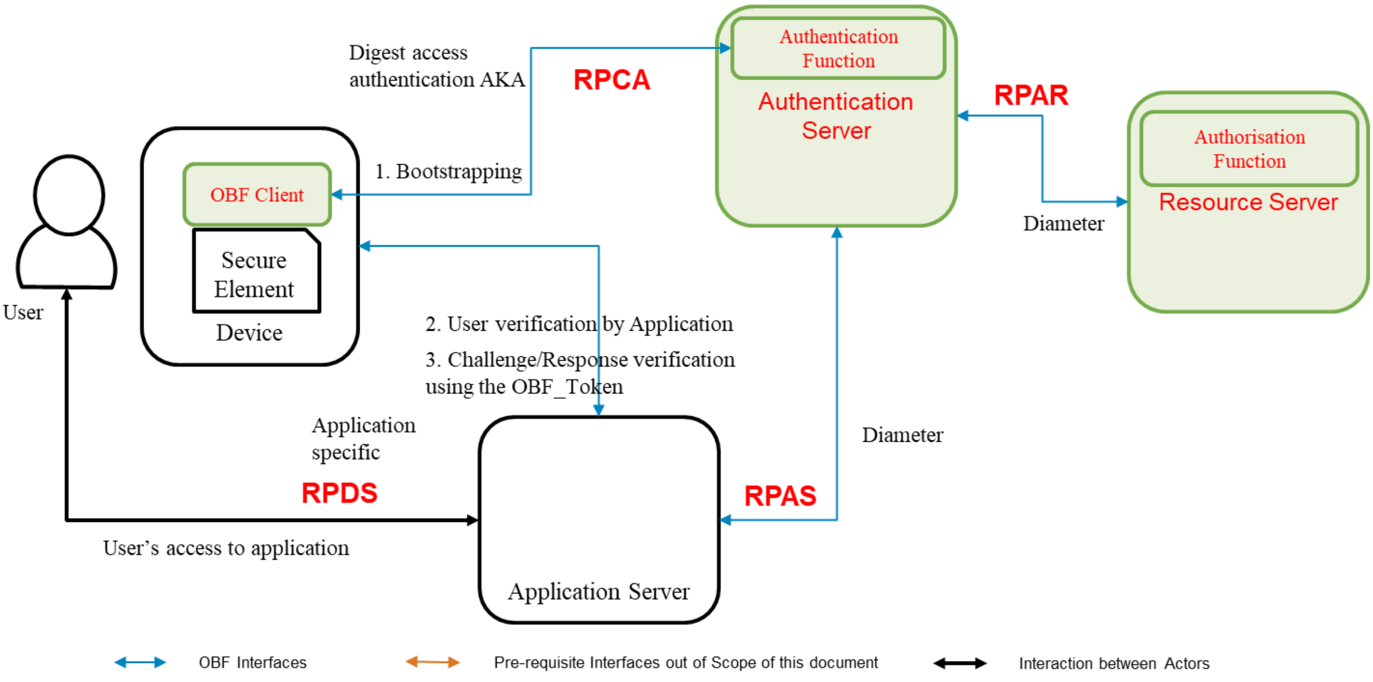

Figure 3: Authentication Workflow

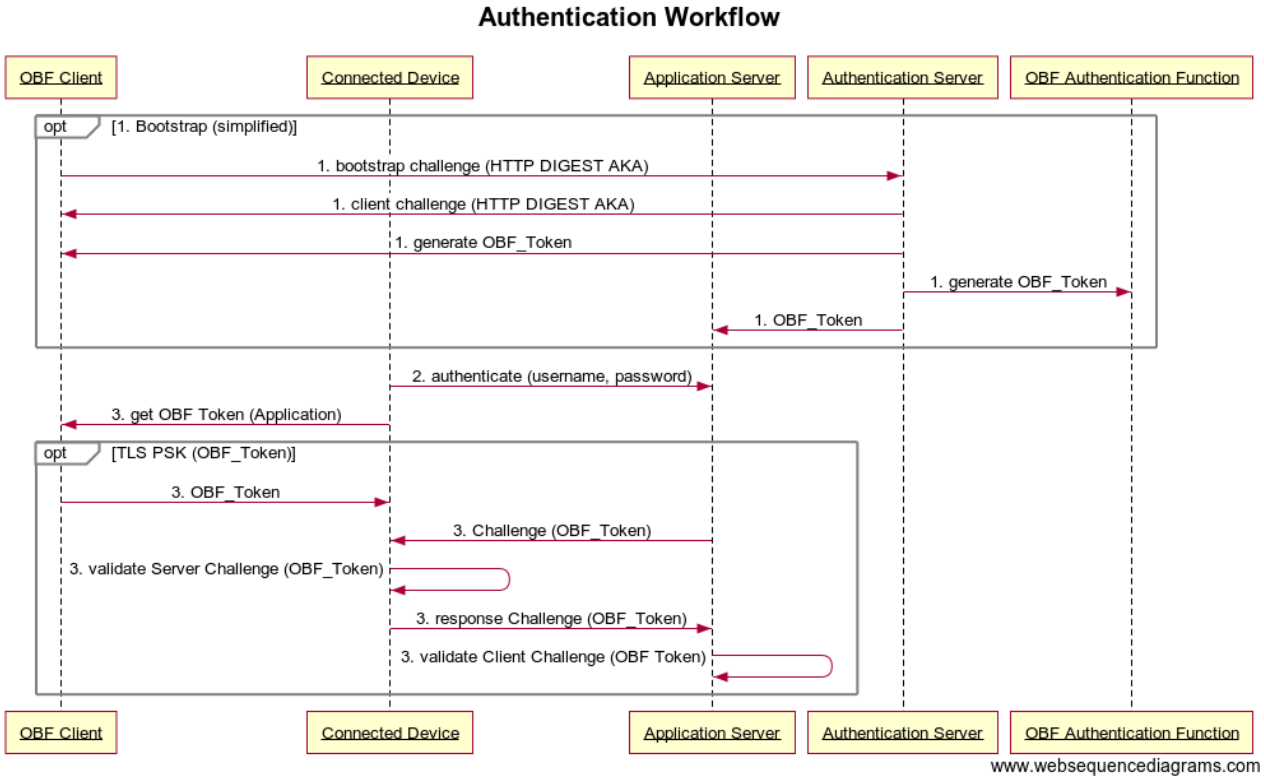
Authentication Workflow
OBF Client
Connected Device
Application Server
Authentication Server
OBF Authentication Function
opt
[1. Bootstrap (simplified)]
1. bootstrap challenge (HTTP DIGEST AKA)
1. client challenge (HTTP DIGEST AKA)
1. generate OBF_Token
1. generate OBF_Token
1. OBF_Token
2. authenticate (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
www.websequencediagrams.com

Figure 4: Authentication Sequence Diagram

12.2 Key Management during bootstrap Flow

The shared key that exists on both the Secure Element, and in the Key Management System of the Authentication server, is used to authenticate the OBF Client with the Authentication Server. Session Keys are used for securing the communication between the device and an Application Server. 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 & 6):

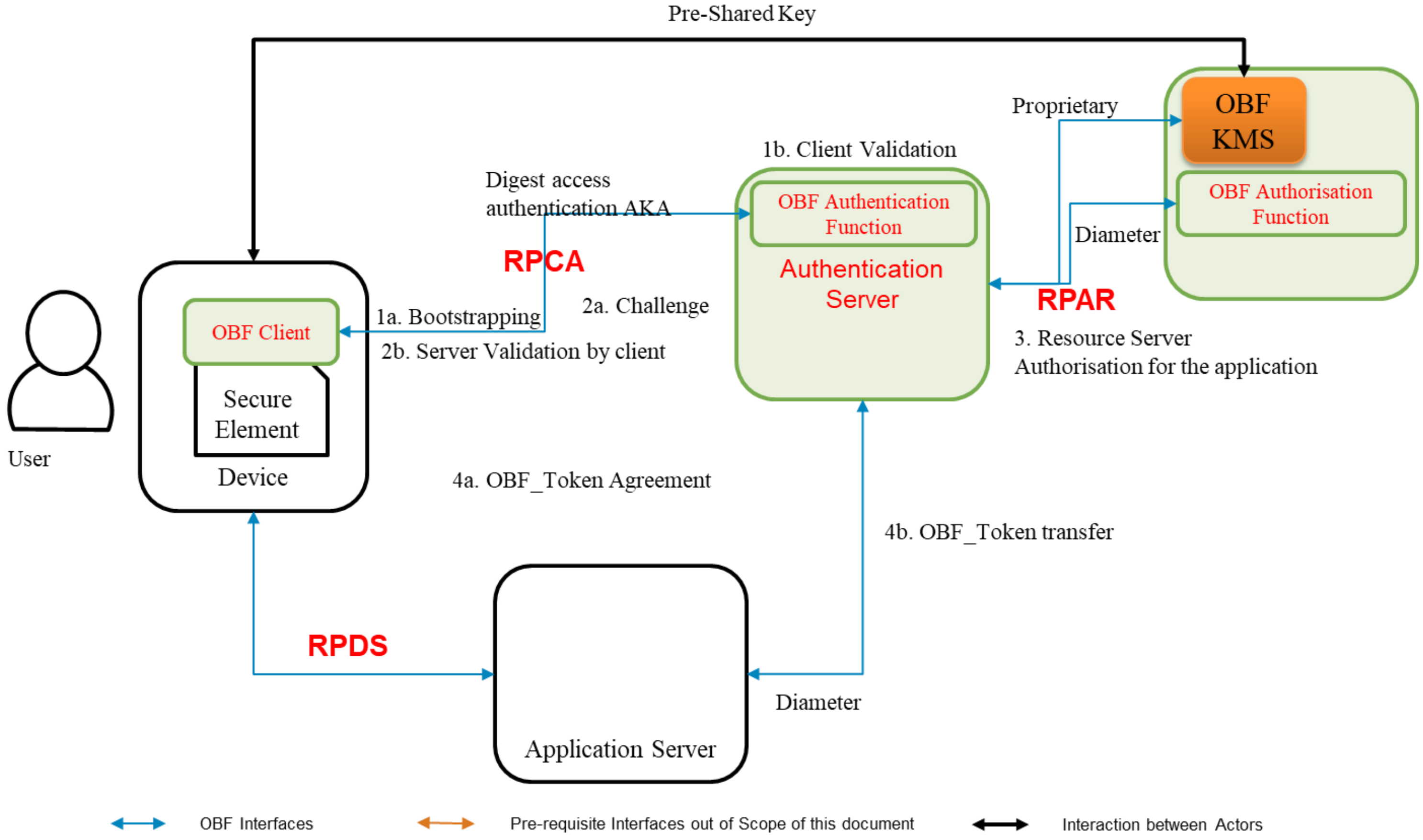
Pre-Shared Key
Proprietary
OBF KMS
OBF Authorisation Function
1b. Client Validation
Digest access authentication AKA
OBF Authentication Function
Authentication Server
Diameter
RPCA
RPAR
OBF Client
Secure Element
Device
User
1a. Bootstrapping
2a. Challenge
2b. Server Validation by client
3. Resource Server Authorisation for the application
4a. OBF_Token Agreement
4b. OBF_Token transfer
RPDS
Application Server
Diameter
OBF Interfaces
Pre-requisite Interfaces out of Scope of this document
Interaction between Actors

Figure 5: Bootstrapping and Session Key Management

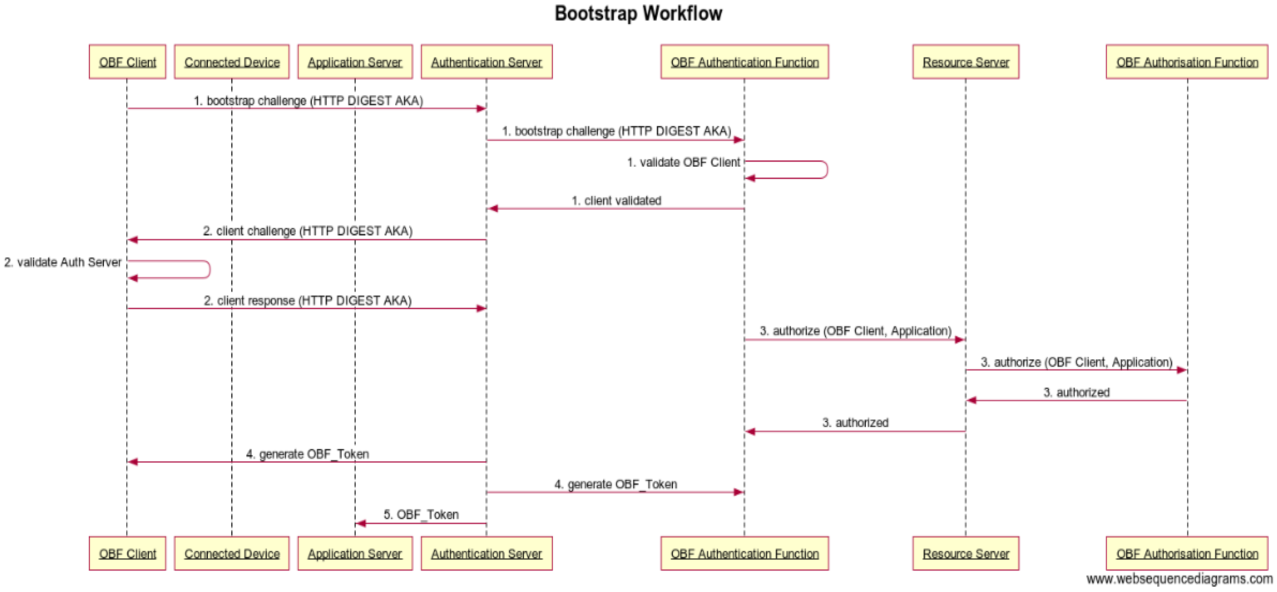
Bootstrap Workflow
OBF Client
Connected Device
Application Server
Authentication Server
OBF Authentication Function
Resource Server
OBF Authorisation Function
1. bootstrap challenge (HTTP DIGEST AKA)
1. bootstrap challenge (HTTP DIGEST AKA)
1. validate OBF Client
1. client validated
2. client challenge (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
www.websequencediagrams.com

Figure 6: Bootstrap and OBF Token generation Sequence

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 7 & 8):

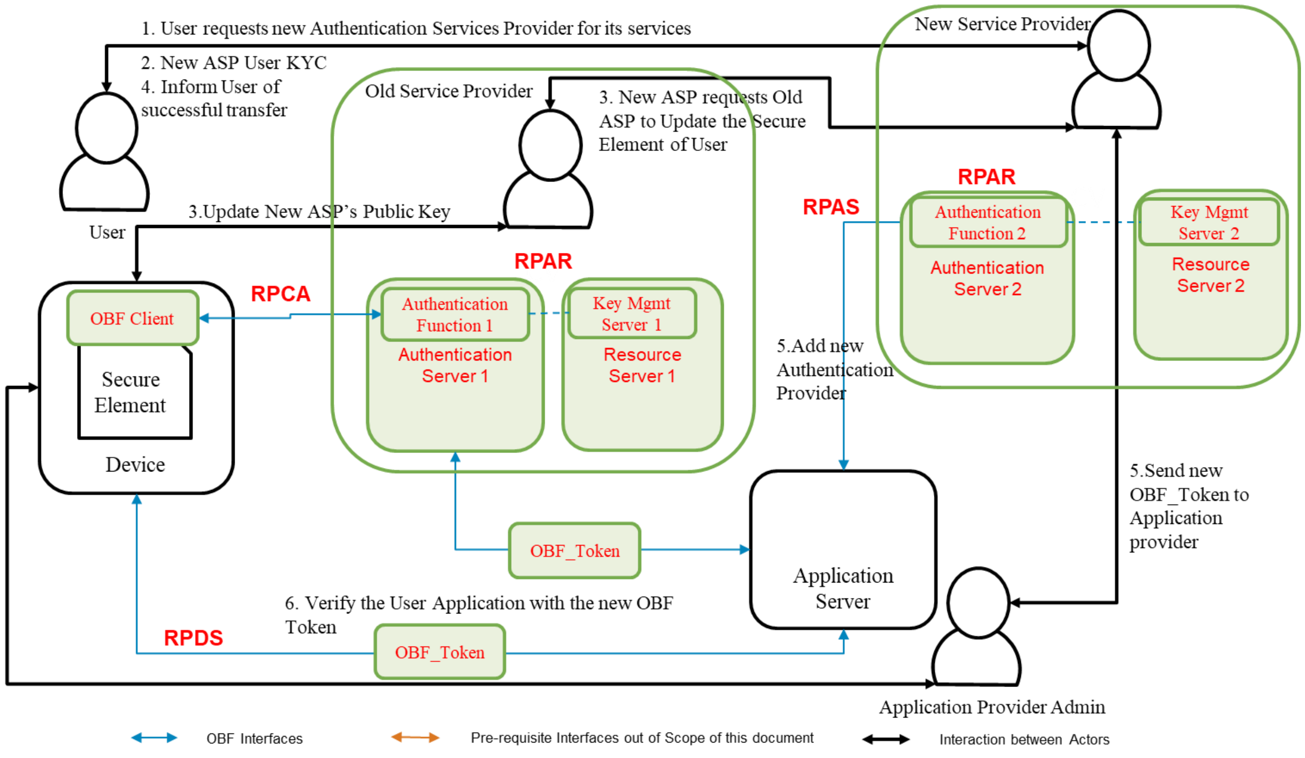
1. User requests new Authentication Services Provider for its services
New Service Provider
2. New ASP User KYC
4. Inform User of successful transfer
Old Service Provider
3. New ASP requests Old ASP to Update the Secure Element of User
User
3.Update New ASP's Public Key
RPAR
RPAS
Authentication Function 2
Authentication Server 2
Key Mgmt Server 2
Resource Server 2
RPCA
RPAR
OBF Client
Secure Element
Device
Authentication Function 1
Authentication Server 1
Key Mgmt Server 1
Resource Server 1
5.Add new Authentication Provider
5.Send new OBF_Token to Application provider
OBF_Token
Application Server
6. Verify the User Application with the new OBF Token
RPDS
OBF_Token
Application Provider Admin
OBF Interfaces
Pre-requisite Interfaces out of Scope of this document
Interaction between Actors

Figure 7: Authentication Provider Switch (Asymmetric keys)

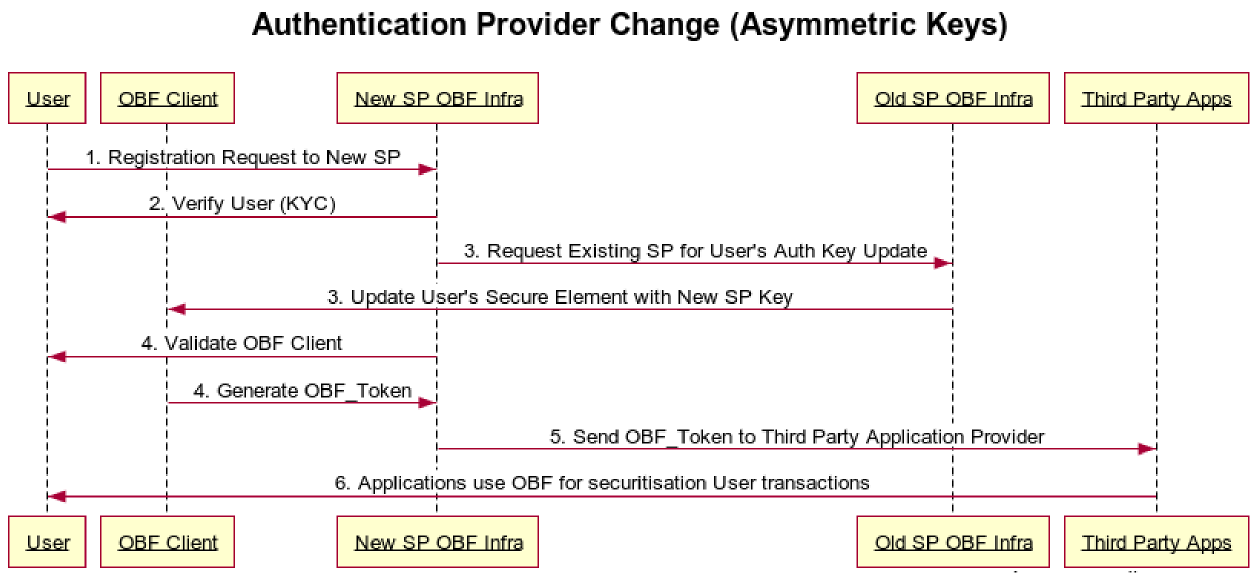
Authentication Provider Change (Asymmetric Keys)
User
OBF Client
New SP OBF Infra
Old SP OBF Infra
Third Party Apps
1. Registration Request to New SP
2. Verify User (KYC)
3. Request Existing SP for User's Auth Key Update
3. Update User's Secure Element with New SP Key
4. Validate OBF Client
4. Generate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

Figure 8: Authentication Provider Switch (Asymmetric keys) Sequence

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 9 & 10):

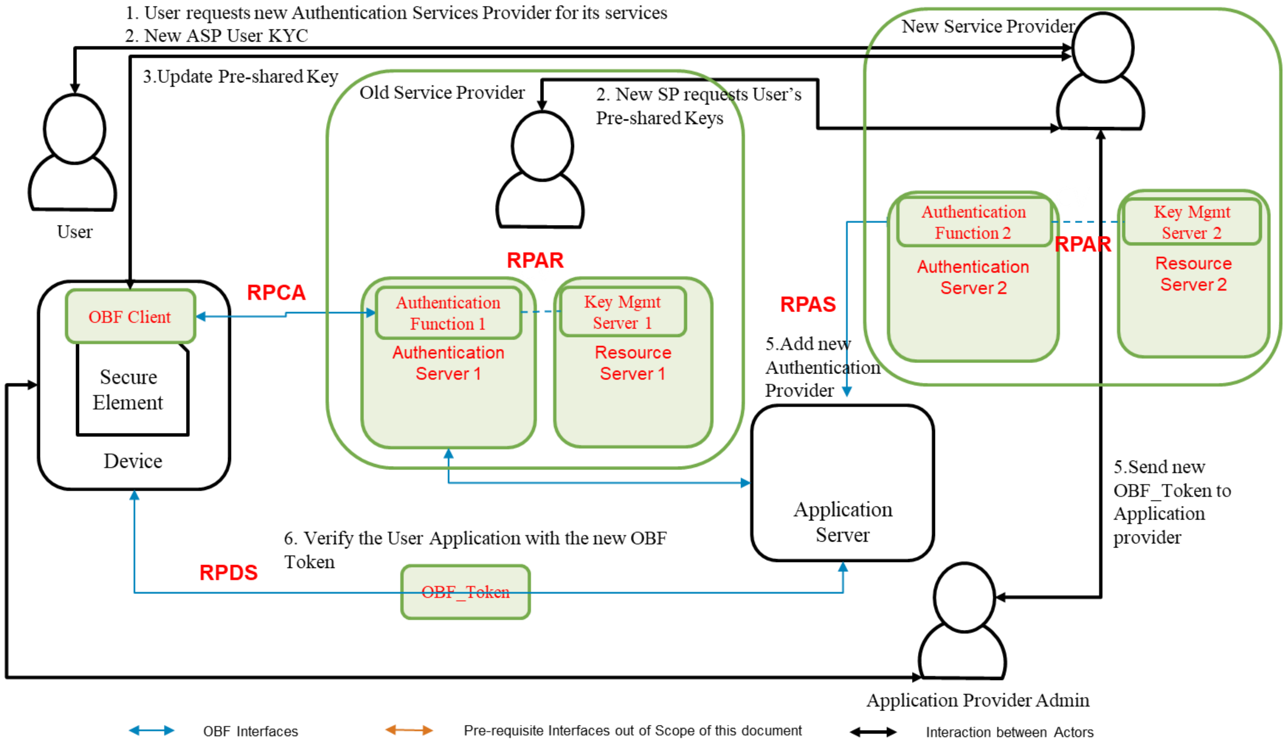
1. User requests new Authentication Services Provider for its services
2. New ASP User KYC
3.Update Pre-shared Key
User
Old Service Provider
2. New SP requests User's Pre-shared Keys
New Service Provider
RPCA
OBF Client
Secure Element
Device
RPAR
Authentication Function 1
Authentication Server 1
Key Mgmt Server 1
Resource Server 1
RPAS
5.Add new Authentication Provider
Authentication Function 2
Authentication Server 2
RPAR
Key Mgmt Server 2
Resource Server 2
Application Server
5.Send new OBF_Token to Application provider
6. Verify the User Application with the new OBF Token
RPDS
OBF_Token
Application Provider Admin
OBF Interfaces
Pre-requisite Interfaces out of Scope of this document
Interaction between Actors

Figure 9: Change Authentication Service Provider (Symmetric Keys)

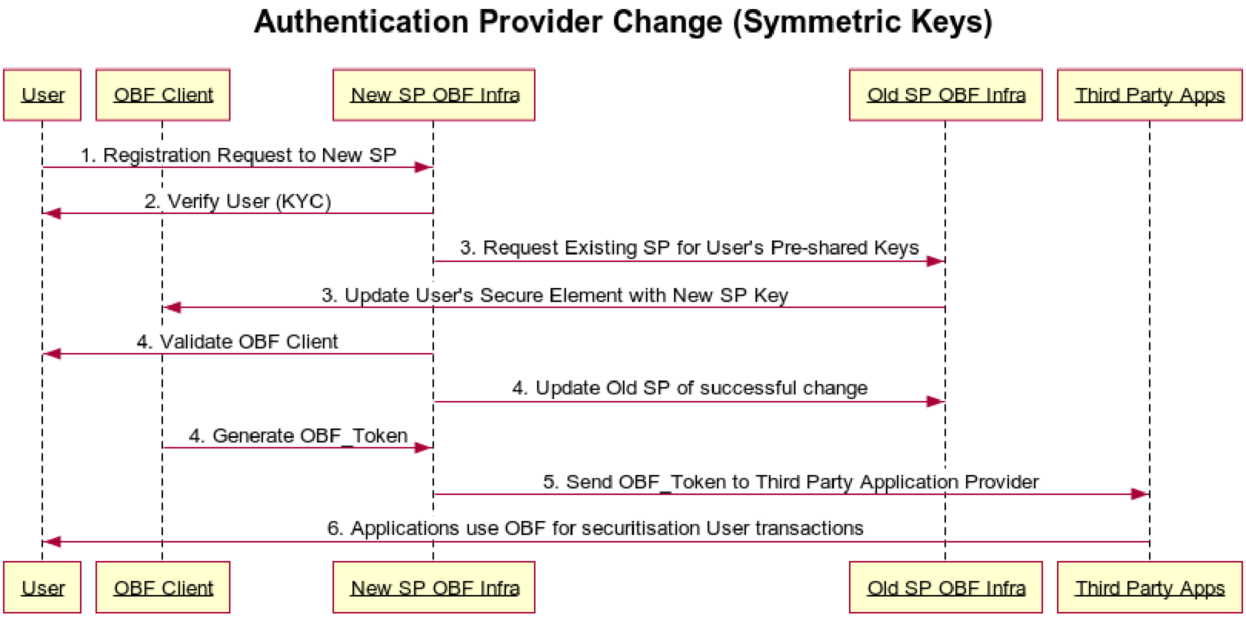
Authentication Provider Change (Symmetric Keys)
User
OBF Client
New SP OBF Infra
Old SP OBF Infra
Third Party Apps
1. Registration Request to New SP
2. Verify User (KYC)
3. Request Existing SP for User's Pre-shared Keys
3. Update User's Secure Element with New SP Key
4. Validate OBF Client
4. Update 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 10: Change Authentication Service Provider (Symmetric Keys) Sequence

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. MNO 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 (Mobile 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

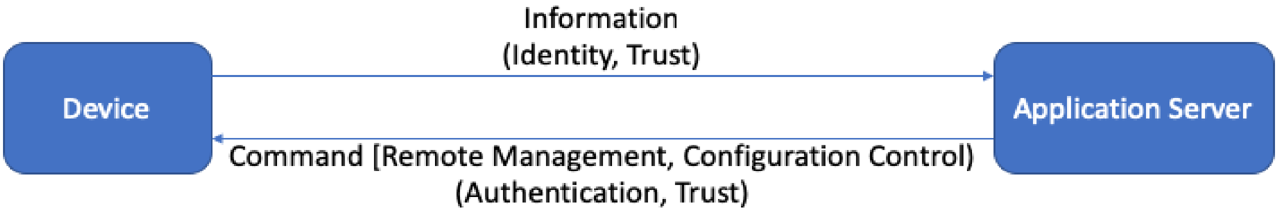

Figure I.1: Device-Application Server Communication

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 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 M2M device will send the data to Secure Element for decryption.

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

(c) Encrypted Device Data Function

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

I.2.2.2 Secure Element Functions

(a) Decrypt Data Function

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

(b) Encrypt Device Data Function

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

I.2.2.3 Application Server Functions

(a) Key Import Function

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

(b) Decrypt Device Data Function

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

(c) Encrypt Server Data Function

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

I.2.3 Application Server to Device flow (Sample)

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

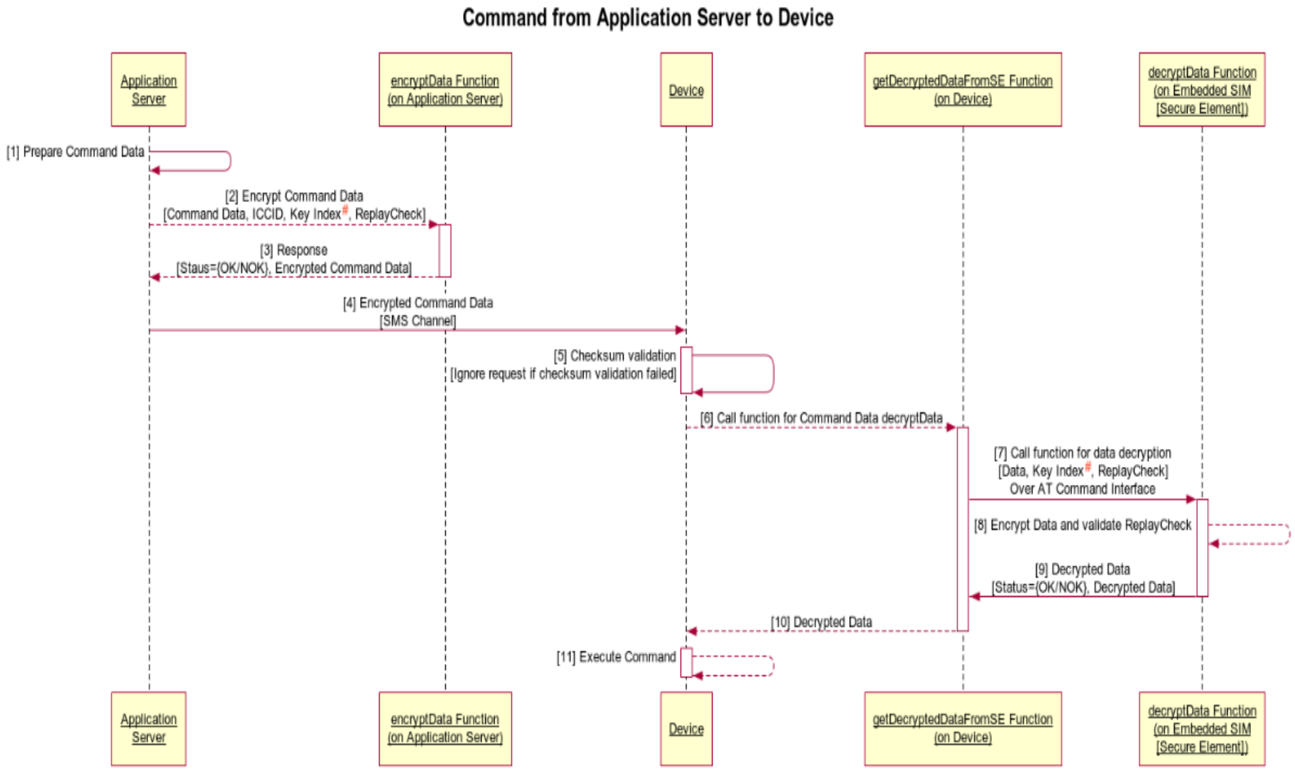
Command from Application Server to Device
Application Server
encryptData Function (on Application Server)
Device
getDecryptedDataFromSE Function (on Device)
decryptData Function (on Embedded SIM [Secure Element])
[1] Prepare Command Data
[2] Encrypt Command Data [Command Data, ICCID, Key Index#, ReplayCheck]
[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 2865] IETF, Request for Comments: 2865 (June 2000), *Remote Authentication Dial In User Service (RADIUS)*

[b-RFC6614] IETF, Request for Comments: 6614 (May 2012), *Transport Layer Security (TLS) Encryption for RADIUS*

[b-RFC7616] IETF, Request for Comments: 7616 (September 2015), *HTTP Digest Access Authentication*.