INTERNATIONAL TELECOMMUNICATION UNION

TELECOMMUNICATION STANDARDIZATION SECTOR

STUDY PERIOD 2017-2020

SG13-TD494/WP3

STUDY GROUP 13

English only

Question(s): 16/13 Virtual, 7 December 2020

TD

Source: Editors

Title: Draft new Recommendation ITU-T Y.OBF_Trust: "Framework for bootstrapping of devices and applications for open access to trusted services in distributed ecosystems" (output of e-meeting, 28-29 September 2020)

Purpose: Information

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

Contact: Ranjana Sivaram
Telecom Engineering Centre (TEC)
India
Tel: +91 9868136990
E-mail: ranjana.sivaram@gov.in

Contact: Sharad Arora
Sensorise Digital Services Pvt Ltd
Tel: +91 9212109999
E-mail: sharad.arora@sensorise.net

Keywords: Y.OBF_Trust; Q16/13; 28-29 September 2020

Abstract: This document is the output of draft Recommendation ITU-T Y.OBF_Trust: "Framework for bootstrapping of devices and applications for open access to trusted services in distributed ecosystems". It includes the discussion results during the interim meeting of SG13/WP3/Q16, which was held virtually from 28 to 29 September 2020.

Meeting discussions and outcome:

During introductory remarks, the Chairman stressed the need for a detailed review of the contribution on Y.OBF_Trust as this is a candidate document for consent in the December 2020 Plenary and also mentioned that more time is being allotted for this review of this document.

During this meeting, it was agreed to make editorial changes in the document as below.

- The redundant/ duplicate texts to be removed.
- Consistency of key words to be checked and aligned in the whole document.
- Diagram 6-1 seems to be redundant and may be removed.
- The reference to identification and numbering in clause 7 and thereafter may be re-worded to avoid an unintentional overlap with the work being carried out in other Study Groups.
- The texts relating to security parameters immediately after the functional diagram may be suitably moved to some other part in the document.
- The updated document with proposed changes may be presented again.

Based on the above meeting discussions, the draft Recommendation was updated and the revised contribution (C-152-R1) was presented in the afternoon session on 28-September 2020.

The meeting decided to accept the updated document (C-152-R1) and made following observations for further updation/ contribution to be submitted in the next SG-13 plenary planned in December 2020:

- The document organization is now stable and may be retained as it is.
- Check for English language corrections, if any.
- Information flows may be reviewed thoroughly for consistency check.
- Each requirement should be thoroughly reviewed for ensuring correct mapping with respective functions in the functional architecture.

The following table shows discussion results for the contribution.

Document Number	**Source**	**Title**	**Meeting results**
C-152	Telecommunication Engineering Centre, Ministry of Communications (India)	Draft recommendation of ITU-T Y.OBF_Trust "Framework for bootstrapping of devices and applications for open access to trusted services in distributed ecosystems".	The contributors/ editors were asked to provide an updated version based on discussions in the meeting dated 28 September 2020 (forenoon session)
C-152-R1	Telecommunication Engineering Centre, Ministry of Communications (India)	Draft recommendation of ITU-T Y.OBF_Trust "Framework for bootstrapping of devices and applications for open access to trusted services in distributed ecosystems".	Accepted with modifications.

Annexure-I

Draft new Recommendation ITU-T Y.OBF_Trust

Framework for bootstrapping of devices and applications for open access to trusted services in distributed ecosystems

Summary

Rapid advancements in communications and associated technologies has led to the emergence of distributed ecosystems with a large number of devices, applications and use cases requiring open access to trusted services. This nature of open access to trusted services in distributed ecosystems can be provisioned by using the inherent security capabilities and mechanisms already present in the devices and the underlying networks. This recommendation provides a concept of bootstrapping of devices and applications by network operators who can share the network security capabilities with users and providers of new devices and services. It describes the requirements to be fulfilled by the entities of the ecosystem such that they may benefit from the bootstrapping capabilities. Based on the requirements, a reference model as well as a functional architecture is provided, which together describe the elements, functions and reference points needed for provisioning of the bootstrapping capabilities. Finally, the recommendation provides the information flows required to enable the bootstrapping capabilities.

Keywords

Bootstrapping; bootstrap_token; trusted device; trusted application; authentication; authorization;

Contents

Page

Draft new Recommendation ITU-T Y.OBF_Trust

Framework for bootstrapping of devices and applications for open access to trusted services in distributed ecosystems

1 Scope

This Recommendation describes the concept, architecture and information flows for bootstrapping of devices and applications by network operators, by providing:

- a bootstrapping concept for entities requiring open access to trusted services in their interactions;
- the requirements imposed on the entities for enabling the bootstrapping capabilities;
- a reference model showing the elements required for bootstrapping;
- a functional architecture diagram showing functions, reference points and security parameters; and
- information flows for the operation of the bootstrapping processes.

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

3 Definitions

3.1 Terms defined elsewhere

This Recommendation uses the following terms defined elsewhere:

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

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

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

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

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

3.1.5. user [b-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: a cryptographic process of binding the user's identity to the keying material provisioned in the secure element of the user's device, enabling the device to communicate securely with trusted services.

4 Abbreviations and acronyms

This Recommendation uses the following abbreviations and acronyms:

3GPP	3rd Generation Partnership Project
AKA	Authentication and Key Agreement
API	Application Programming Interface
ASP	Application Services Provider
CIN	Company Identification Number
FQDN	Fully Qualified Domain Name
GBA	Generic Bootstrapping Architecture
HTTP	Hyper Text Transfer Protocol
IMEI	International Mobile Equipment Identity
IoT	Internet of Things
IPSec	Internet Protocol Security
KYC	Know Your Customer
MAC	Media Access Control
MSISDN	Mobile Subscriber International Services Digital Network
PSK-TLS	Pre-Shared Key Cipher suites for Transport Layer Security
SIM	Subscriber Identification Module
TLS	Transport Layer Security
UID	Universal Identifier or Public Entity Identifier
URL	Uniform Resource Locator

5 Conventions

In this Recommendation, requirements are classified as follows:

- The keywords "**is required to/ 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 ~~Concept of bootstrapping~~ Introduction

The concept of trusted devices, operator trust and bootstrapping of devices and role of network operator in enabling trusted services is described below.

~~The rapid developments in electronics, communications and applications domain is leading to the emergence of new ecosystems of users, devices, applications, service providers and network operators, which require open access to trusted services to all the entities in these distributed ecosystems.~~

~~The network operators have played a critical role in provisioning of trusted services by undertaking the subscriber and device verification prior to permitting access to its services. The network operator's trust relationship is possible to be extended to the new users, devices, applications and service providers that require trustful interactions for the orderly proliferation of the trusted services.~~

~~Devices use secure elements that authenticate themselves to the network's security nodes by using cryptographic processes. The network operator conducts a subscriber verification prior to allowing them the use of its network services and resources. With some enhancements in its network, network operators can add capabilities to on-board application services providers (ASPs) from distributed ecosystems to allow subscribers of its network to securely access the trusted services of the ASP. Network operators can extend the trust from the existing verification of users and devices, by bootstrapping the devices and ASP applications using the network operator trust infrastructure.~~

~~The bootstrapping capabilities built into the devices and the network elements that secure the transactions between the subscribers of the network and the services offered by the network operator can be easily extended to provide open access to trusted services to the entities within the distributed ecosystems. The entities can also be provided the facility to change the bootstrapping for trusted services when changing the network operator.~~

6.1 Concept of trusted services

Certain services require additional checks prior to making them available to a beneficiary. License to drive cars, access to restricted premises, permissions for online banking are few examples of trusted services each of which require some previous introduction between the intended beneficiary and the provider of the trusted service. Further, these services may have applications which require privacy and security of the information exchanged with the user/ device that is using the application. Such services, that require user verification and security of the information, are referred to as trusted services.

Rapid developments in embedded electronics and information and communication technology (ICT) are leading to new and evolving ecosystems of devices and applications that are enabling advanced services by interconnecting physical and virtual things.

These developments require suitable improvements in the ICT infrastructure for identification, authentication and authorization amongst the unrelated and diverse set of entities within the ecosystem including users, service providers, devices, networks and applications. Network operators play an important role in connecting the user's devices to the internet and with the applications. With some improvements in its ICT infrastructure, the network operator can extend its role to provide the required trust between hitherto unknown entities of the ecosystem. If the ICT layer interfaces and related processes are standardized over a wide range of network technologies, the new standardized infrastructure can be used for an open yet secured access and interactions between devices and applications in distributed ecosystems. This Recommendation provides for the required capabilities and functions to achieve this end.

6.2 Operator trust and bootstrapping of devices

The network operator establishes a trust relationship with its subscribers and devices by

- undertaking the verification of every new customer prior to permitting the person access to its services and infrastructure. (subscriber verification)
- authenticating the devices through the associated secure element of the devices and the network's security nodes. (device authentication)

Once the trust relationship is established between the person and the network operator, the person is referred to as a subscriber as it becomes eligible to use and pay for the network services like calling, messaging, internet access etc.

Whilst operator trust ensures identification of users, modern networks have placed a lot of focus on ensuring that the device –

- has a valid and unique identifier that cannot be easily created by an entity other than the manufacturer and can be used for authentication; and
- can be permitted or restricted from connecting to the network i.e. can be authorized or rejected.

The concept of bootstrapping of devices and that of operator trust provide an important basis for fulfilling the requirements of authentication and authorization when considering the interactions between the users, devices and applications from distributed and diverse ecosystems. However, different network technologies have different schemas and mechanisms for establishing the trust. Network operators may also use different processes for implementing the operator trust even within the same network technology. If a uniform mechanism was provided such that the network operator trust could be used for enabling secure interactions between diverse and distributed ecosystems of devices and applications, it could make it very easy for orderly proliferation of trusted services.

For example, in cellular mobile networks, the authentication between the device and the network is managed by a secure procedure that is executed between the device, the secure element of the SIM and an authorization node in the network. The concept of bootstrapping of devices is meant to extend the authentication and authorization process described above for use by third party applications. To accomplish this, the existing capabilities of the device and networks are extended (e.g. by using different authentication algorithms, or different keys, etc) to the device-based applications, often by the use of a security token that is generated at the time of enrolling the device for enabling such an extended network authentication.

The security token has a crucial role as it acts like a digital identifier for all of the following:

1. the application for which it is generated;

2. the device on which it is generated; and

3. the network and the network node for which it is generated.

Other than acting as the identifier, the security token also embodies the keying material (Key data which is used to protect the security communication of the device and the network) the key length, generation algorithm and lifetime are set according to parameters, such as service type and security degree) required for securing the interactions between the device and the third-party applications.

Transferring of security token(s) between the device and the network is best avoided as it represents a risk of compromise of the token during the process of transfer. As a result, mechanisms exist that allow security tokens to be independently generated by the device and the network. These mechanisms are standardized as authentication and key agreement (AKA) processes.

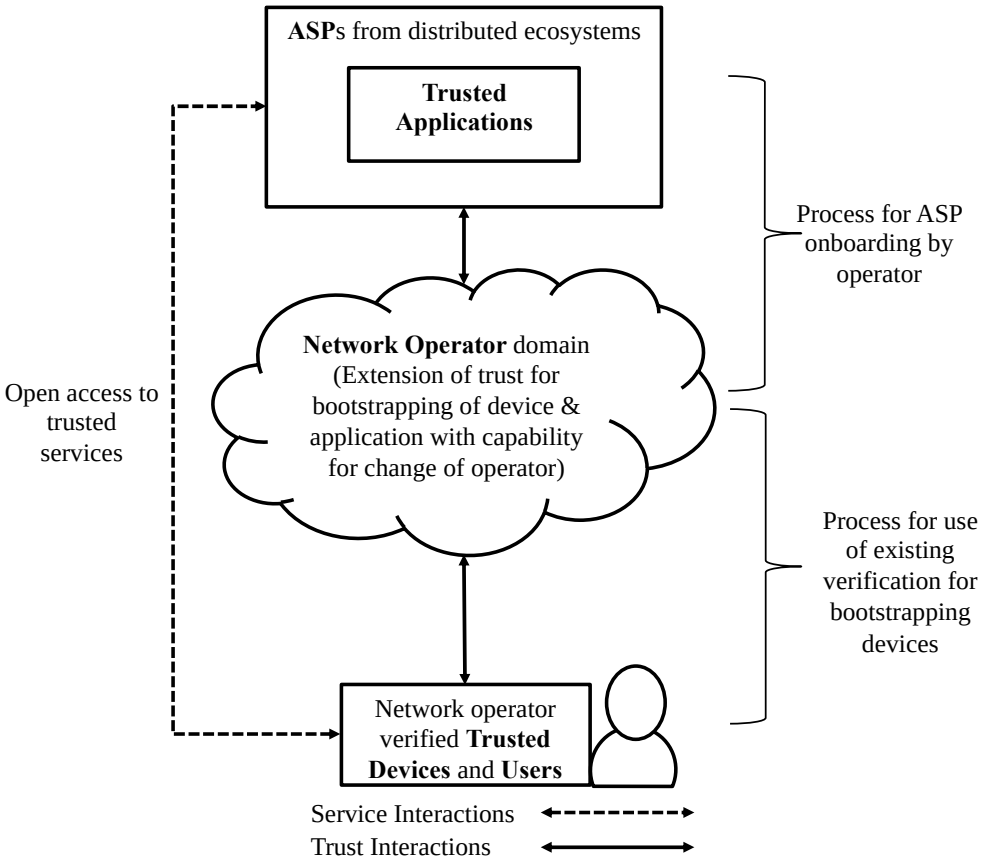
The bootstrapping of a device may thus be described as a process in which

1. a device already registered in a network is given certain additional privileges;
2. the device and the network have an agreed AKA for the generation of secure tokens; and
3. the device and the network have an agreed mechanism by which they are able to identify and allow access to certain third-party applications.

The provisioning of such a capability by a network is referred to as a network operator realm. There may be multiple network operators that offer bootstrapping capabilities, and hence, there may be multiple reams available for bootstrapping at one time.

6.3 Role of network operator in enabling trusted services

The ~~concept~~ approach to securing trusted services by the extension of operator trust through bootstrapping of devices is shown in the diagram below:



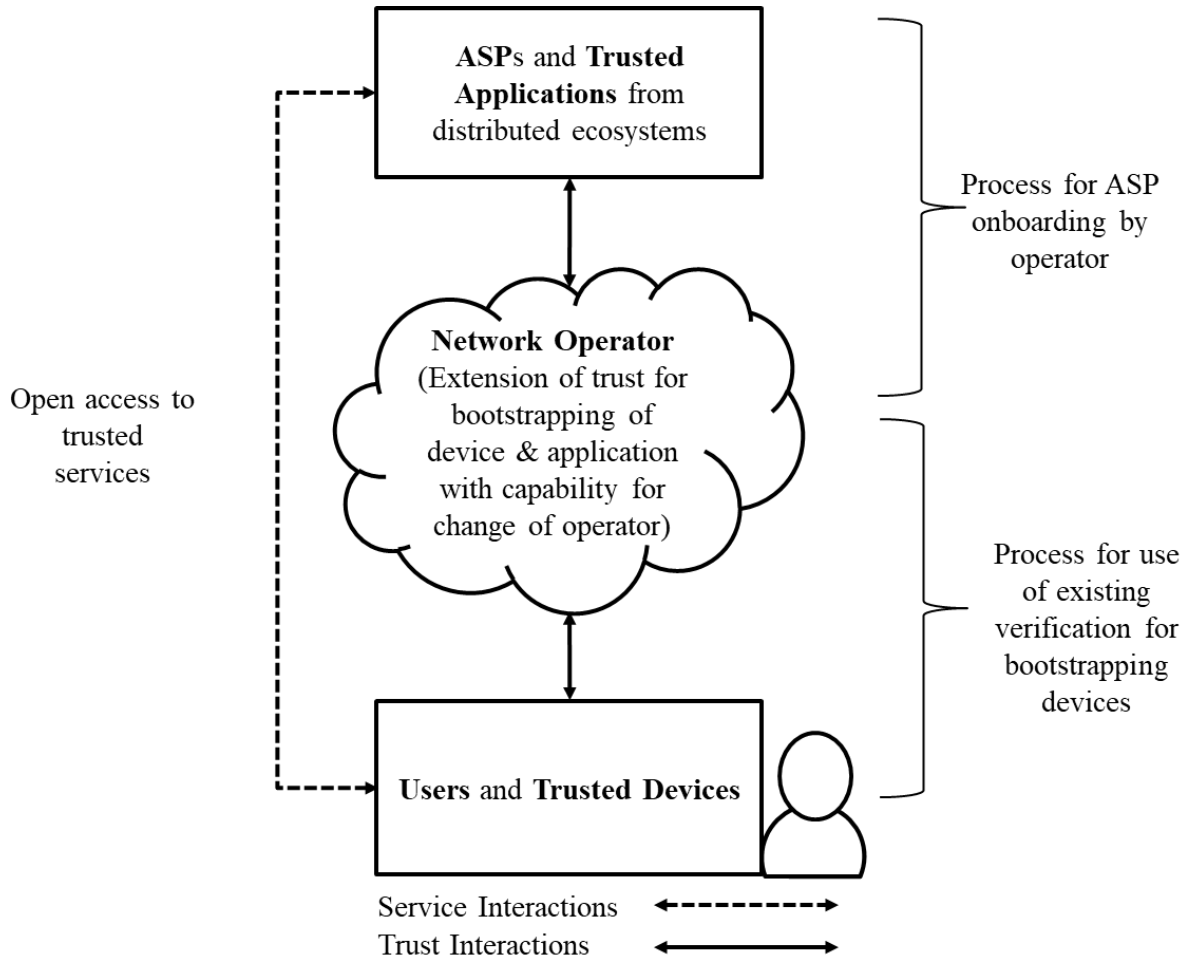


Figure 6-1: Role of Network Operator in connecting diverse ecosystems of trusted services

The bootstrapping concept involves the following entities:

1. **Trusted device**: A device with an associated secure element which is on-boarded by the network operator.

   NOTE – Secure element is defined in clause 3.1.1.

2. **User:** A person that is a verified subscriber of the network operator, desirous of using trusted services from ASPs. ~~The user provides its credentials to the ASP, whose services it intends to consume, via the network operator or service provider that holds the verified credentials of the user by virtue of an earlier verification process.~~

   NOTE 1– A subscriber is a person/ entity, who subscribes to the services of a network operator and whose credentials are verified by the network operator before providing services.

   NOTE 2 – The user provides its credentials to the ASP, whose services it intends to consume, via the network operator or service provider that holds the verified credentials of the user by virtue of an earlier verification process.

3. **Network operator:** An entity that provides network connectivity services and undertakes the physical verification of the subscriber and the device. It can share trust generated from this verification information to bridge new relationships between providers of trusted services and users of trusted devices by deploying the bootstrapping capabilities in its network.
4. **Application services provider (ASP):** An entity that develops and offers trusted services and applications, and has a requirement for a minimum level of authentication and authorization prior to the use of its application and services by the users. However, the ASP does not have a direct relationship with the users, unlike the relationship between the network operator and its subscriber. The ASP has an expectation of deriving its trust from the relationship between the network operator and its subscriber.
5. **Trusted application:** ASP application on-boarded by the network operator, which are capable of controlling access to users of trusted devices using cryptographic capabilities.

The interactions between the entities that are intended for the establishment of the trust between the entities are referred to as the trust interactions. When the entities interact such as to use the functionality of the trusted applications, these interactions are referred to as the service interactions.

The required solutions to enforce trustworthy interactions between the subscribers, devices and services within the network operator domain already exist. The objective of the next clauses is to provide the requirements, architecture and information flows to extend the underlying network and device security capabilities for use by ASP trusted applications that are outside the network operator domain. An important consideration for this Recommendation is that it ensures independence from a specific network technology and permits change of network operators for the user and the ASPs. Another important consideration is the ability to handle multiple network realms and the transfer of bootstrapping from one realm to another.

7 Requirements

7.1 Pre-conditions

A pre-condition is a logical predicate that must be true for the application of this Recommendation. The pre-conditions stated in this clause are not a new requirement of this Recommendation, but a given condition that the entities ~~ecosystem~~ must follow as per existing standards and norms.

The following pre-conditions are applicable for this Recommendations:

- ability to manage bootstrapping capabilities from multiple network operators;
- reuse the identification and numbering of trusted devices and network elements as per the network technology;
- reuse the identification and naming of the ASP (e.g. CIN)
- reuse the identification and numbering of trusted applications (e.g. IP Address/ FQDN, URL, oneM2M App-ID, GS1 Application ID etc.);
- reuse the subscriber's credentials as recorded during the subscriber verification, for purposes of user registration;
- reuse the device credentials as recorded during the device authentication, for purposes of device registration;
- ~~reuse the identification and numbering of trusted devices and network elements as per the network technology layer;~~
- ~~reuse the identification and numbering of trusted applications;~~

- use identities representing ~~from~~ each of, the network, the trusted device and the trusted application ~~domain security parameters~~ for purposes of mutual authentication transactions ~~by using an identifier from each of the network, the trusted device and the trusted application domain~~; and
- support the existence of, and choose from, the ~~–~~multiple network operators that may be offering bootstrap capabilities.

~~Note~~ NOTE – Numbering and/ or identification systems are out of the scope of this Recommendation, which mainly focuses on bootstrapping using existing numbering and identifiers.

Further, the pre-condition for bootstrapping requires that ~~the bootstrapping identifier~~a security token is specified by the network operator which:

- is based on the device on which it is generated, the application for which it is generated, and the network and the network node for which it is generated; and
- inheriting the existing identities of the device (e.g. IMEI or MAC), the network (e.g. MSISDN or IMSI for GSM) and application (e.g. IP, URL, etc.) which together fulfil the requirements for addressing within communication protocols.

7.2 Requirements for the security token

The ~~security~~ Security ~~Token~~ token is required to:

- be used for carrying the identities of the network operator, trusted device and the trusted application for mutual authentication;
- have the keying material necessary ~~include keys to be used in the~~ for the cryptographic processes ~~–~~that establish a secure session between the trusted device and the trusted application;
- bind the ~~bootstrapping identities~~ identities ~~fier~~ of the trusted device and trusted application to the keying material;
- be an identifier of the network operator realm and the trusted device to which it is issued and have lifecycle management capabilities; and

be generated independently ~~io~~in the device and the network, as per the security protocols and parameters published by the network operator; ~~node as per the agreed AKA process~~

~~(a) The security token is hereafter called the bootstrap_token.~~

7.3 Requirements for the user entity

The user is required to register itself and its trusted device(s) with the network operator for subscribing to the trusted service(s) of an ASP.

7.~~3~~4 Requirements for the trusted device entity

The trusted device is required to have:

- ~~capabilities to use its secure element for enabling trust interactions; and~~
- ~~an application~~capability for ~~initiation and management of bootstrapping of the device with the network operator~~ registration of a device client with the network operator that can subsequently be used for provisioning of secure access to trusted applications;
- securely transfer its network identifier to the device client for use in secure authentication of the user/ device towards trusted applications;
- capability of encryption and decryption of data interchanged with the trusted application(s);

- capabilities to use its secure element for storage and retrieval of keys and sensitive data for enabling trust interactions;
- capabilities to manage bootstrap tokens from multiple network operator realms, ensuring that only one realm is active at a given point in time; and
- capabilities for secure ~~using a network identifier for the purpose of identification of the device; andan application for~~ access to trusted applications using session controls and end to end data privacy and security.

-

- 7.~~4~~5 Requirements for the network operator entity

The network operator is required to:

- publish the protocols and parameters necessary for bootstrapping of trusted devices and trusted applications;
- publish the systems and processes for bootstrapping;
- make ~~network~~ enhancements in its network for ~~to support the~~ on-boarding of ASPs and the ASP's trusted applications~~;~~ and to allow users' to register their trusted devices for bootstrapping;
- securely store and use the credentials recorded during the subscriber verification and device authentication to support the trust interactions between the trusted device and the trusted application;
- capabilities for storage and retrieval of keys and other sensitive data for enabling trust interactions;

  protect the ~~user's and~~ subscriber's identity and the device's network identity ~~against discovery and misuse, both in transit and at rest, when in communication with the ASP ;~~

- ~~extend the existing trust relationship and security capabilities between subscriber and network operator to that between the user and the ASP;~~
- ~~publish the security parameters for bootstrapping of trusted devices and applications;~~
- ~~support systems and processes to extend the existing user/ device verification for bootstrapping devices to trusted applications; and~~
- allow the user to change its device's registration for bootstrapping ~~registration to~~ a different network operator.~~.~~

7.~~5~~ 6 Requirements for the trusted application entity

The trusted application is required to:

- have an identification/ numbering as per the agreed schema (e.g. IP Address/ FQDN, URL, oneM2M App-ID, GS1 Application ID etc.);
- support the security protocols and parameters published by the network operator;
- ~~have functions to benefit from network operator offered bootstrapping capabilities;~~
- ~~have unique identifiers~~ have functions that ~~and access control~~ provide secure access to only such network operator registered ~~capabilities~~devices that have a valid subscription;
- provide the protocols and capabilities that enable a trusted device to interact with it securely; and

- establish secure ~~connections~~ sessions with the trusted device using the ~~security parametersbootstrap_token~~bootstrap token ~~example session k.ey;~~

7.~~6~~7 Requirements for the ASP entity

The ASP is required to:

- register with the network operator(s) using its public identity (e.g. Company Identification Number);~~s that offer bootstrapping capabilities;~~
- register its trusted application(s) with the network operator using the application identity (e.g. IP number, URL, oneM2M App-ID, GS1 Application ID etc.);
- ~~and~~ publish its ~~trusted~~ applications;
- expose a registration process for subscribers of network operators to discover and register to its trusted applications; and
- provide means for access control (e.g., add, delete and modify) of trusted applications that can be administered by the ASP, the user or the network operator, as required, based on the mapping of the trusted application with trusted devices provisioned by the ASP.

~~and manage access control (e.g., add, delete and modify) configurations;~~

8 Reference model

A reference model has been provided which presents ~~defines~~ the elements within the entities (described in clause 6) and the required ~~and the requisite~~ trust and service interactions between the elements to meet the requirements stated in the clause 7 above.

The reference model is described in the diagram below.

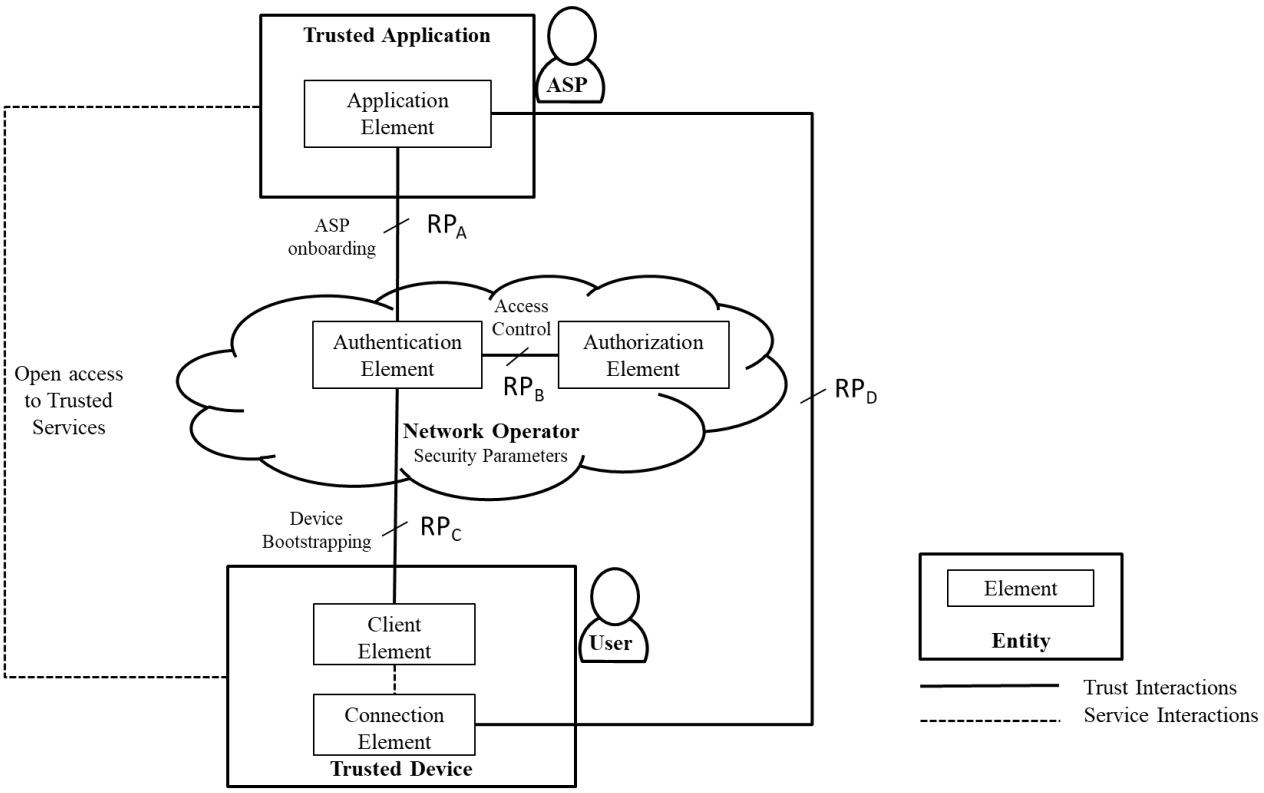
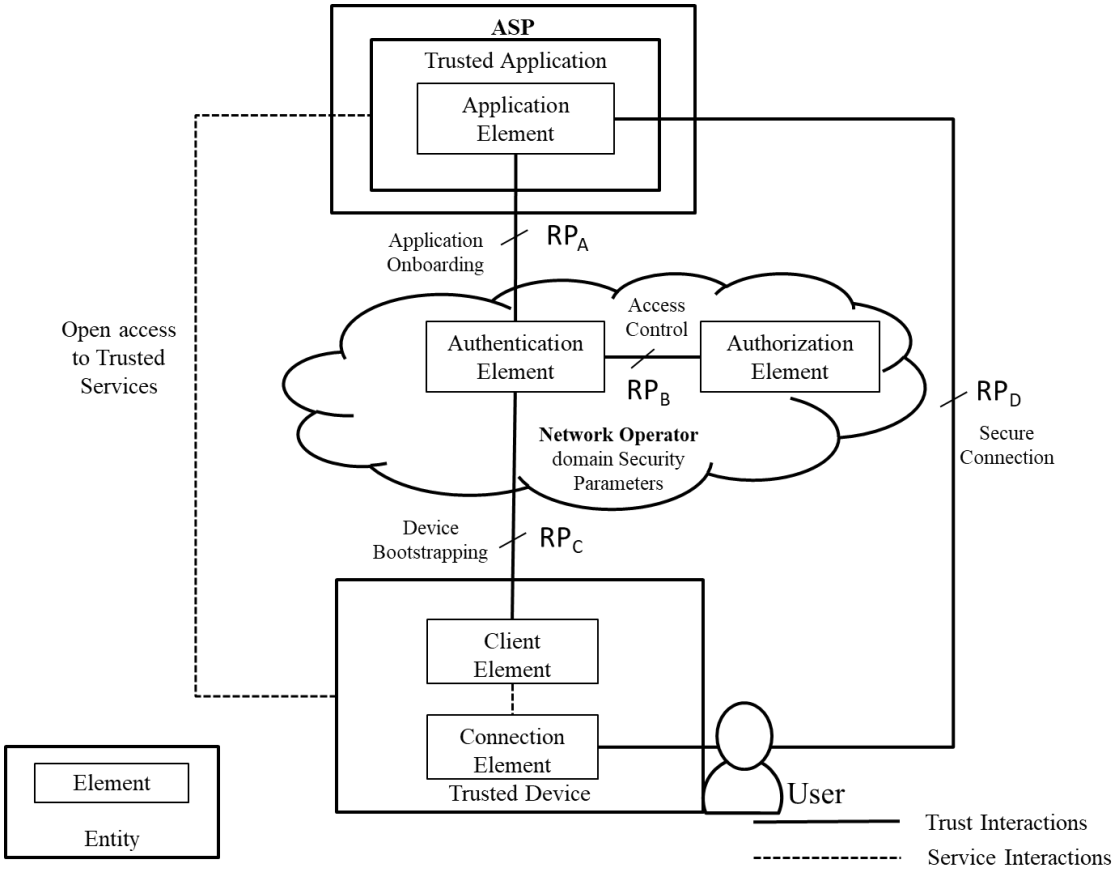
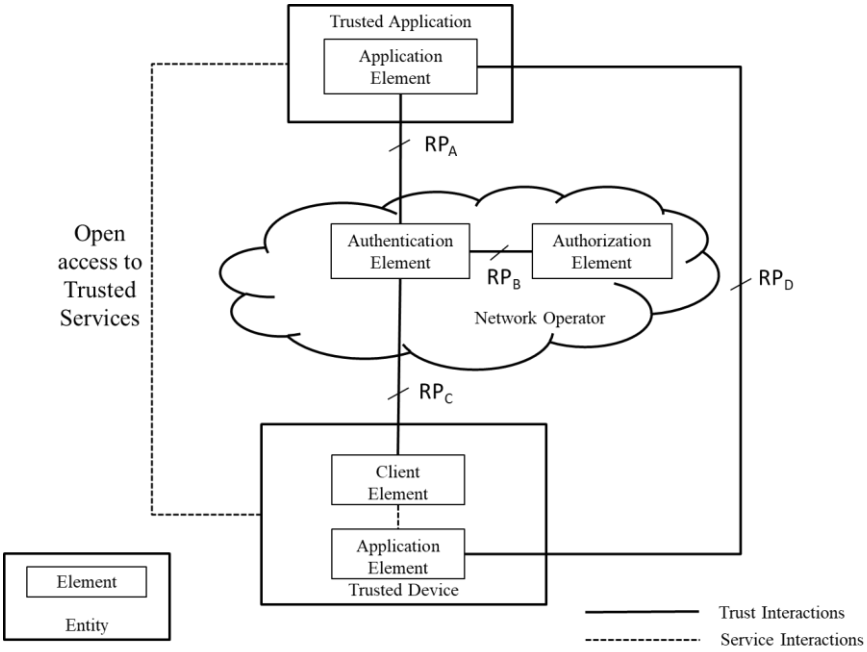


Figure 8-1: Reference model


ASP
Trusted Application
Application Element
Application Onboarding
RP_A
Open access to Trusted Services
Access Control
Authentication Element
Authorization Element
RP_B
RP_D
Secure Connection
Network Operator domain Security Parameters
Device Bootstrapping
RP_C
Client Element
Connection Element
Trusted Device
User
Element
Entity
Trust Interactions
Service Interactions



8.1 Elements of the trusted device entity

The trusted device hosts a client element and an application element for supporting the trust and service interactions, respectively. These elements are described below.

8.1.1 Client element

The client element is a ~~an application~~ software resident in the trusted device, or optionally in its associated connectivity element (e.g. the SIM or the authentication element), that provides the keying material and the authentication mechanism for bootstrapping the trusted device to the network operator for purposes of secure access to trusted services.

8.1.2 ~~Application~~ Connection element ~~of the trusted device entity~~

Th~~is~~ ~~connection~~ element is a part of the trusted application, responsible for setting ~~sets~~ up the secure session~~connection~~ between the trusted device and application using the bootstrap_token ~~enablement~~ provided by the client element.

8.2 Elements of the network operator entity

The network operator adds two important elements, namely i) authentication element and ii) authorization element to address the capabilities of on-boarding ASPs and the trusted applications, and further to allow controlled access to the trusted services from the trusted devices of the subscribers of its network. These elements are described below.

8.2.1 Authentication element

The authentication element identifies and authenticates the client element of the trusted device using authentication protocols (e.g., ~~XXX~~AKA, EAP, RADIUS, DIAMETER) and security parameters (e.g., ~~XXX~~ random number, algorithm for key generation).

8.2.2 Authorization element

Figure 8-1: Reference model

The authorization element carries out the key and certificate management functions required to support the cryptographic processes for on-boarding trusted devices and applications. It also provides the keying material, support for industry standard protocols (e.g.~~, XXX~~ OAUTH,

DIAMETER etc.) and the mapping of the access controls between the trusted devices and applications.

8.3 Application element ~~of the trusted application entity~~

For ASPs to benefit from the bootstrapping capabilities exposed by the network operator, its trusted applications have an application element that complies~~y~~ to industry standard protocols (e.g. OAUTH, DIAMETER, etc.) for bootstrapping, access control and session management.

The application element ~~of the trusted application entity~~ sets up the secure session~~connection~~s between the trusted devices and applications using the network operator specified ~~industry standard~~ protocols and security parameters. The application element is deployed in each trusted application.

~~[Editor's Note – the Clause 9.1 has been moved here as clause 8.4, and the remaining clauses renumbered below]~~

8.4 Security parameters

The security parameters include network identifiers, trusted device identifiers, trusted application identifiers, subscription information and the keying material which together create the bootstrap_token. The purpose of the identifiers is to uniquely identify and address the trusted devices, trusted applications, nodes and the security protocols in a network operator realm. The purpose of the subscription information is to authenticate and authorize the secure interactions between trusted devices and applications. The bootstrap_token is a session key, independently generated in the trusted device as well as in the authentication element based on an agreed security schema between the client element and the authentication element. The bootstrap_token is generated by using the security parameters negotiated as part of the bootstrapping process. It is used for establishing a secure session between the trusted device and application.

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 deployment model, the underlying network technology, the AKA protocol, the numbering/ identification mechanism of the network and internet layer, the service type and the security degree required for the use case, etc.

As described in clause 7, the bootstrapping process inherits the device, network and application identifiers. The network operator specifies the security protocol that is used over reference point RP_D.

NOTE - As an example, in case of 3GPP, it is as per Annex-H of [b-3GPP TS 33.220].

The bootstrapping process uses subscription information which contains parameters such as the user's network identifier, 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.

NOTE - Subscription information can be as per ITU-T X.1124 (11/2007)

~~The bootstrapping process uses the bootstrap_token which binds the user's identity to the keying material for secure communication between the trusted device and the trusted application over the reference point RP_D.~~

8.5 Reference points

The reference points are a very important part of the reference model as they make the interactions between the five elements secure, standardised, interoperable and transferable. It is because of the reference points that the bootstrapping capabilities are openly accessible by trusted devices and applications without constraints of network technology or network operator domain.

The four reference points are described below:

(a) RP_A - the reference point between the authentication element of the network operator and the application element of the trusted application;

(b) RP_B - the reference point between authentication element and the authorization element belonging to the network operator;

(c) RP_C - the reference point between the client element hosted in the trusted device and the authentication element of the network operator; and

(d) RP_D - the reference point between the connection element of the trusted device and the application element of the trusted application.

The functionality required to support the features and the flow of information for the service and trust interactions are is described in the clauses below.

9 Functional architecture

The bootstrapping requirements can be realized as per the functional architecture ~~is provided to~~ shown in the diagram below, which presents ~~make it possible for the entities to implement~~the required functions and the interfaces ~~within~~ between the entities.

NOTE - An implementation of the bootstrapping functional architecture by a network operator is referred to as a realm. The instantiated ~~functions~~ elements within the realm are referred to as nodes. As an example, an authentication element, when instantiated in the network by the network operator entity, will be referred to as the authentication node in the realm of that network operator entity.

The functional architecture diagram shown in Figure 9-1 describes the following:

- the security parameters that are used to enable bootstrapping capabilities.
- the required functions within the elements; and
- the reference points required for the interfaces between the functions across elements.

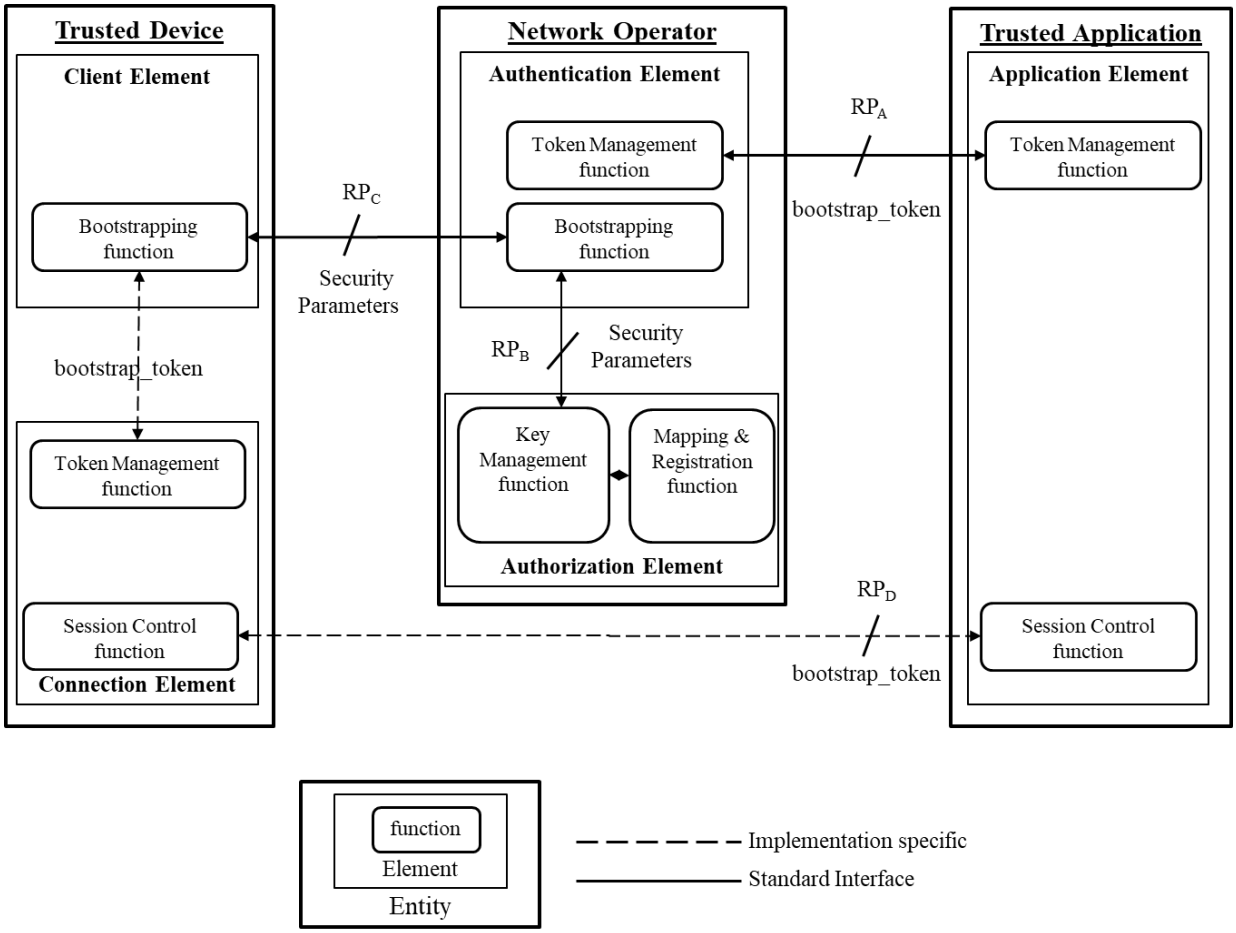


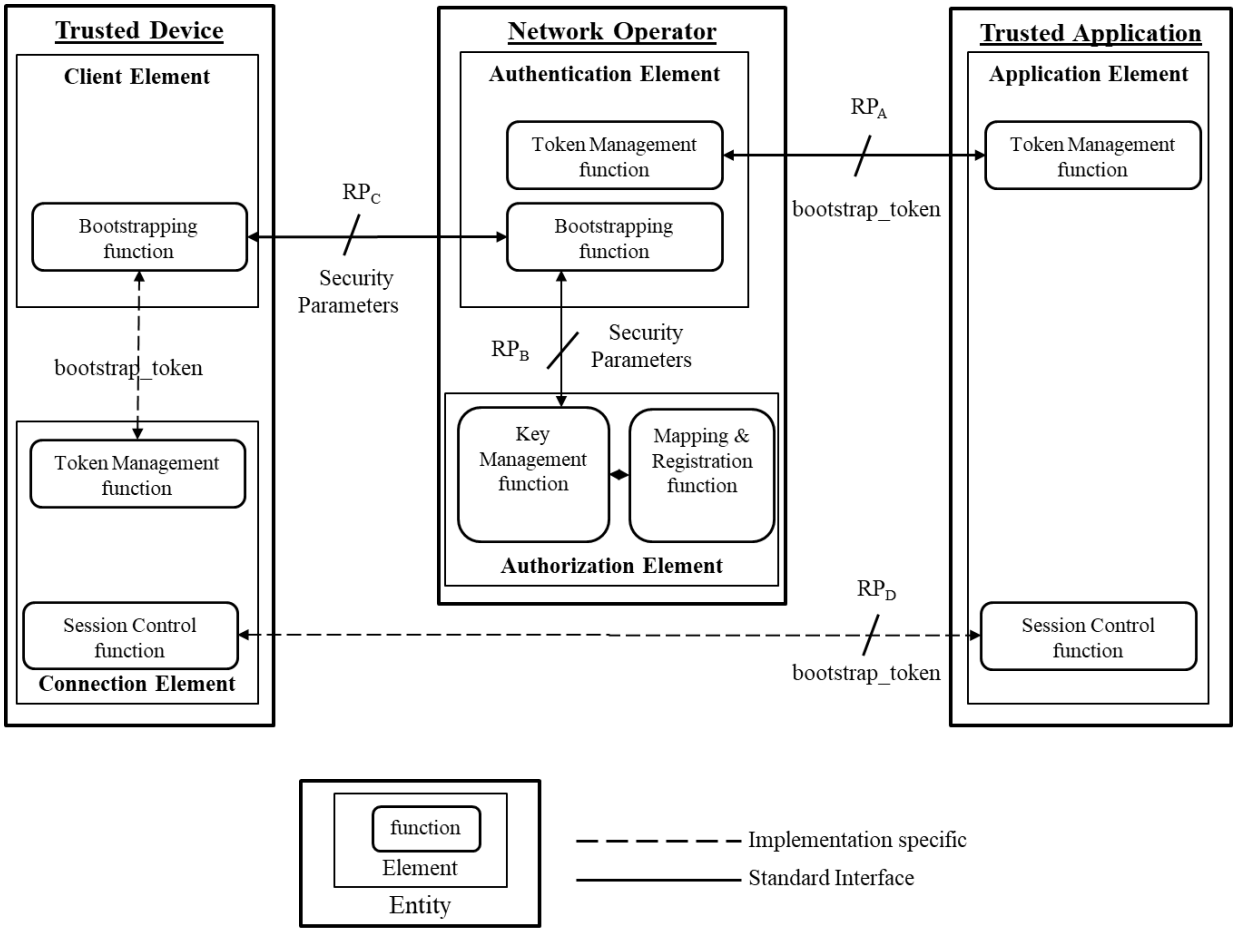
Figure 9-1: Functional architecture

- ~~the reference points required for the interfaces between the functions.~~
- ~~the security parameters that are used by the functions over the reference points to enable bootstrapping capabilities.~~

~~[Editor's Notes – the clause 9.1 has been moved to 8.4, rest of the clauses renumbered accordingly]~~

9.~~1~~1 Functions of authentication element

The authentication element has two functions that enable the bootstrapping of the trusted device and the trusted application, namely the bootstrapping function and the token management function. -The



bootstrapping function is described below, whereas the token management function is described in clause 9.4.~~Each of the functions are described below.~~

9.1~~1~~.1 Bootstrapping function

The bootstrapping function is responsible for the mutual authentication between the client element of the trusted device ~~In addition, this function mutually authenticates the client element~~ and the authentication element of the network operator.~~, as an enabling step in the process towards generation of long-term keying material within the bootstrapping function. The function is executed over the reference point RP_C.~~

The bootstrapping function provides the following functionalities:

- configures and communicates the format of the bootstrap_token to the client element;
- fetches the identity of the client element from the trusted device;
- verifies the trusted device credentials with the subscriber verification database of the network operator and creates a record of the client element against the trusted device for mutual authentication;
- provides the client element identity to the token management function;
- protects the users' network identity against discovery and misuse during the trust and service interactions with the trusted application.
- ~~bootstrapping the device using an identity used by the network;~~
- ~~protects the subscriber's network identity against discovery and misuse by the ASP;~~

- protects the subscriber's network identity against discovery and misuse over the communication interfaces with the ASP;

- supports AKA protocols as per the underlying network technology;

- manages the lifecycle of the bootstrap_token;

- configures and communicates the format of the bootstrapping identifier to the client element;

- fetches the data from the authorization element; and

configures the bootstrapping security parameters in conjunction with the authorization element and communicates that to the client element., as an enabling step in the process towards generation of long-term keying material within the bootstrapping function. The function is executed over the reference point RP_C.

9.11.2 Token management function (authentication element)

- This function provides the functionality for generating the bootstrap_token by using the agreed security parameters as well as transferring the bootstrap_token to the trusted application, so it can be used by the session functions in the trusted application.

This function also securely transfers the bootstrap_token to the trusted application, so it can be used by the session functions in the trusted application.

NOTE – The bootstrap_token is specific to the client element and the trusted application for which it is generated. The lifetime of the bootstrap_token may vary significantly across various use cases. When the application element of the trusted device is invoked, or required to initiate the interaction, by a trusted application, the bootstrap_token is validated to ensure the lifetime of the token has not expired. If the lifetime has expired or if no current bootstrap_token is available or when indicated by the trusted application, the client element will use the token management function to obtain a new bootstrap_token.

9.22 Functions of authorization element

The authorization element has the capability to securely store the credentials of the trusted device and the subscriber which are recorded at the time of device authentication and subscriber verification.security parameters for the verified users and the trusted devices belonging to the subscribers of the network operator. It maintains the secure identitiesy of the ASPs and the ASP's trusted application(s) that are on-boarded by the network operator. It maintains the mapping of the trusted devices that have been authorized to access the trusted applications, and keeps the updated access control list.

The authorization element has two functions that are described below.

9.22.1 Key management function

This function provides the management, storage and retrieval of keys and other sensitive data corresponding to the trusted devices. and association of keys and algorithms between the mapping and registration function and the bootstrapping function of the client element. It stores the pre-shared keys or certificates corresponding to the trusted devices and manages the keys and lifecycle of the keying material as per the agreed AKA protocol.

9.2.2 Mapping and rregistration function

This function provides for the registration of the ASP(s) and their trusted application(s). The function hosts the repository of trusted applications that are allowed to be used by the client element of a trusted device(s).the functionality for registration of trusted applications by the ASP and

~~subscription to the trusted applications by users of bootstrapped devices. The function hosts the repository of authorized trusted applications that can be permitted for use by the trusted device, and also the mapping of the specific trusted applications that are allowed to be used by client element of a trusted device.~~

The session control function exists in the connection element of the trusted device and the application element of the trusted application. The functions enable the establishment and maintenance of the session and session security between the trusted device and trusted application. It uses the bootstrap token for mutual authentication.~~function validates the trusted devices and provides them access to the trusted applications based on the bootstrap_token sent in the authentication request.~~

The mapping and registration function provides the following functionalities:

- register the user as per the credentials already registered in the subscriber verification database of the network operator (e.g. Name, Address, National Identity, Passport number, etc.);
- ~~Register~~ register the user's trusted device(s) identity as per the device authentication information already registered in the device authentication database of the network operator (e.g. IP number, MSISDN, IMEI, etc.);
- register the ASP that has trusted applications user's trusted device(s) as per the device authentication information already registered in the device authentication database of the network operator (e.g. CIN, etc);
- supports the addition / deletion of authorized application providers / trusted applications through standardized API or user interfaces;
- provisions the users and trusted applications with the required security parameters;
- stores the mapping of the subscription to the trusted application(s) by trusted device(s);
- ~~that subscribe for a trusted application;~~
- ~~supports the protocols required over the reference point RP_A;~~
- ~~provisions the users and trusted applications with the required security parameters;~~
- ~~responds to the bootstrapping function over the reference points RP_A with the authentication vector and user's security parameters such as the key lifetime and user identities;~~
- supports the addition / deletion of ~~authorized~~ client element of ~~authorized~~ trusted devices ~~/ users through standardized API or user interfaces;~~
- supports the delegation / revocation of access control rights to authorized client ~~element~~ element(s) through standardized API or user interfaces; and
- supports the protocols required over the reference point RP_B;

9.~~3~~3 Bootstrapping function of the client element

The bootstrapping function of the client element corresponds to the bootstrapping function of the authentication element and has the same features as described in clause 9.1.1.

The bootstrap function of the client implements the following functionality:

- resides in the trusted device entity;
- interacts with the secure element of the trusted device;
- supports the required AKA protocol and~~;~~

- stores the keying material;
- fetches the subscription credentials of the user recorded in the authorization element from the authentication element during the bootstrapping process;
- ~~and select from one amongst several keys for security enablement;~~
- generates the bootstrap_token as per the format and security parameters and manages the bootstrap_token lifecycle as specified by the authentication element ~~security parameters negotiated~~ during the bootstrapping process; ~~and~~
- selects from one amongst the possible several ~~available~~ bootstrap_token(s) corresponding to multiple network operator realms; and
- allows ~~allowing~~ only one bootstrap_token to be active at a given point in time.

~~NOTE – The subscription information related to the user and its authentication is delivered to the client element from the authorization element via the authentication element during the bootstrapping process. The subscription information related to the trusted application (e.g. access to application allowed, type of certificates that may be issued) is sent to the client element.~~

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

~~The bootstrap_token is a session key, independently generated in the client element of the trusted device as well as in the authentication element based on an agreed security schema between the client element and the authentication element. The bootstrap_token is generated by using the security parameters negotiated as part of the bootstrapping process. It is used for establishing a secure session between the trusted device and application.~~

~~The characteristics of the bootstrap_token are as follows:~~

~~It binds the user identity to the keying material used in the reference points;~~

~~It is the globally unique identifier of the realm of the network operator in which it is issued;~~

~~It serves as a temporary identifier of the trusted device to which it is issued; and~~

~~(b) It identifies the key used in the cryptographic processes over reference point RP_C and RP_D;~~

~~(c)~~

9.44 Token Management Function ~~s of the application element~~

The token management function is present in the application element of the trusted application, the connection element of the trusted device and the authentication element. This function securely transfers the bootstrap_token from authentication element to the application element.

~~functions of the application element~~

9.4.1 Token Management Function of the authentication element

The token management function of the authentication element provides the following functionalities:

- uses an AKA algorithm as specified by the network operator;
- fetches the trusted application credentials from the authorization element;
- fetches the client element credentials from the bootstrapping function;

- verifies the mapping of the client element and the trusted application;
- generates the bootstrap_token by binding its own identity (e.g. an IP, URL, 3GPP Global Title, etc.), that of the client element (e.g. a combination of IP, IMEI, MAC, MSISDN, IMSI, etc.) and that of the trusted application (e.g. IP Address/ FQDN, URL, oneM2M App-ID, GS1 Application ID etc.) which together fulfil the requirements for addressing and mutual authentication between the entities;
- manages the lifecycle of the bootstrap_token;
- securely transfers the bootstrap_token to the trusted application; and
- protects the trusted devices' network identity against discovery and misuse during the trust and service interactions with the trusted application.

NOTE – The bootstrap_token is specific to the client element and the trusted application for which it is generated. The lifetime of the bootstrap_token may vary significantly across various use cases. When the application element of the trusted device is invoked, or required to initiate the interaction, by a trusted application, the bootstrap_token is validated to ensure the lifetime of the token has not expired. If the lifetime has expired or if no current bootstrap_token is available or when indicated by the trusted application, the client element will use the token management function to obtain a new bootstrap_token.

9.4.2 Token Management Function of the connection element of the trusted device

The token management function of the trusted device is responsible for the generation, storage and lifecycle management of the bootstrap_token on the trusted device using the secure element for storage.

~~are deployed in the trusted device and the trusted application. These functions enable establishment and maintenance of the session and session security between the trusted device and application.~~

~~The two functions of the application element are i) token management function and ii) session control function the functionality of which is described below.~~

~~9.4.41 Token management function of the application element~~

~~The token management function of the application element exists in both the trusted device and the trusted application. It corresponds to the token management function of the authentication element. It provides the storage and lifecycle management of the bootstrap_token. In the case of the trusted device, it is responsible for using the secure element for storage~~

9.4.3 Token Management Function of the trusted application

The token management function of the trusted application is responsible for:

- fetching the bootstrap_token from the authentication element;
- lifecycle management of the bootstrap_token as per the policies set by the authentication element;
- secure storage of the bootstrap_token as per the storage resource provided by the trusted application; and
- secure retrieval and exposure of the bootstrap_token to the session control function.
- ~~of the bootstrap_token. In case of the trusted application, it is responsible for using the storage as per the storage resource provided by the trusted application.~~

~~This function also securely transfers the bootstrap_token to the trusted application, so it can be used by the session functions in the trusted application.~~

9.5 Session control function ~~of application element~~

The session control function exists in the connection element of the trusted device and the application element of the trusted application. The functions enable the establishment and maintenance of the session and session security between the trusted device and trusted application. It uses the bootstrap_token ~~-~~ for mutual authentication.

~~The session control function of the application element exists in both the trusted device and the trusted application. It is application specific. It utilizes the bootstrap_token to initiate and maintain a secure session between the application element of the trusted device and that of the trusted application.~~ The function is implemented using session control protocols ~~within an industry standard session control~~ such as TLS, PSK-TLS, Kerberos, IPSec. It protects the security and privacy of the identities and data that is exchanged in the trust and service interactions between trusted device(s) and trusted applications(s). ~~use of the network subscriber identity against discovery and misuse.~~ It supports the application protocol in the reference point RP_D and initiates the request for bootstrap_token when indicated by the trusted application.

9.6 Specifications of reference points

The functionality of the four reference points is described below:

9. 6.1 Reference point RP_A

The reference point RP_A provides the following functionalities:

- enables secure communication between the authentication element and the application element;
- allows the transfer of the subscription information related to the trusted device to enforce access control policies between trusted devices and applications;
- ~~supports the DIAMETER [b-RFC 6733] and [b-RFC 7155] protocol;~~
- allows the application to send its address (e.g. FQDN), public entity identity (e.g., UID), basic key material (e.g., a shared secret or a public-key certificate), entity service permission flag, supported authentication mechanisms and the authentication inquiring and key generation mechanism to the bootstrapping function;
- allows the token management function of the authentication element to transfer the bootstrap_token to the token management function of the application element of the trusted application;
- allows the token management function of the application element to indicate to the token management function the authentication element the eligibility of the bootstrap_token for a single or multiple application.

NOTE – The characteristics of the reference point may be fully met by industry standard protocols e.g. the Diameter protocol described in [b-RFC 6733] and [b-RFC 7155] protocol.

9.6.2 Reference point RP_B

The reference point RP_B enables the mutual authentication between the bootstrapping function of the authentication element and the functions of the authorization element~~. It supports the DIAMETER [b-RFC 6733] and [b-RFC 7155] protocol.~~

It provides the subscription information regarding the client elements when trusted devices request access to trusted applications. The reference point also provides the keying material for the client

element for the bootstrapping information flow. It maintains the permissions for the client element to access certain trusted applications.

NOTE – The characteristics of the reference point may be fully met by industry standard protocols e.g. the Diameter protocol described in [b-RFC 6733] and [b-RFC 7155] protocol.

9. 6.3 Reference point RP_C

The reference point RP_C provides the interfaces for the bootstrapping of the client element to the authentication element. The reference point RP_C uses the agreed AKA for authentication between authentication element and the client element and establishes the security parameters and AKA for generation of the bootstrap_token.~~provides the following functionalities:~~

- ~~supports the HTTP Digest protocol [b-RFC7616] and may optionally support other industry standard protocols;~~

~~uses the agreed AKA for authentication between authentication element and the client element; and establishes the security parameters and AKA for generation of the bootstrap_token.~~ NOTE – The characteristics of the reference point may be fully met by industry standard protocols e.g. the HTTP Digest protocol [b-RFC7616].

9.6.4 Reference point RP_D

The reference point RP_D supports the interfaces for the secure interaction between the trusted device and application.

The reference point RP_D provides the following functionalities:

- supports the application-specific protocol between the trusted device and application;
- sends the indication from the trusted application to the trusted device that a valid or new bootstrap_token is required prior to connecting to the trusted application;
- supports the use of the bootstrap_token for creating the secure association between the trusted device and application; and
- allows the application element to signal to the client element regarding lifecycle management of keys;

NOTE – The characteristics of the reference point may be fully met by industry standard protocols e.g. certificate-based TLS authentication mechanism, PSK-TLS, IPSec, etc.

~~9.6 Security parameters~~

~~The security parameters include identifiers, subscription information and the keying material which together create the bootstrap_token. The purpose of the identifiers is to uniquely identify and address the trusted devices and the nodes in a network operator realm. The purpose of the subscription information is to authenticate and authorize the secure interactions between trusted devices and applications.~~

~~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 deployment model, the underlying network technology, the AKA protocol, the numbering/ identification mechanism of the network and internet layer, the service type and the security degree required for the use case, etc.~~

9.6.1 Identifiers

The identifiers uniquely identify a client element in a trusted device to an authentication element and the application element. The following identifiers are relevant:

a. Node identifier;
b. Trusted device identifier;
c. Trusted application identifier; and
d. security protocol identifier.

The description of the various identifiers are as below.

(a) **Node identifier**:

The node identifier comprises such minimum connection and security attributes that can uniquely address and fully support the authentication element from one of many in multiple technology domains. As an example, an authentication element will require the node's FQDN, the Global Title Address and the associated AKA to fully qualify the requirement of the node identifier, when such a node is deployed in a GSM network. The node identifier provides an implementation dependent address, connection and security details of the elements deployed in a network operator realm.

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

(c) **Trusted application identifier**:
It is an identifier of the trusted application that includes an FQDN and a unique identifier provided by the network operator or an application registry.

(d) **Security protocol identifier**:
It is an identifier, which is associated with a security protocol over reference point RP_D. The security protocol identifier is defined by the network operator and it is network technology specific.

NOTE - As an example, in case of 3GPP, it is as per Annex H of [b-3GPP TS 33.220].

9.6.2 Subscription information

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

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

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

~~9.6.3 Bootstrap_token~~

~~The bootstrap_token binds the user's identity to the keying material in the reference points. The bootstrap_token is a session key, independently generated in the client element of the trusted device as well as in the authentication element based on an agreed security schema between the client element and the authentication element. The bootstrap_token is generated by using the security parameters negotiated as part of the bootstrapping process. It is used for establishing a secure session between the trusted device and application.~~

~~The characteristics of the bootstrap_token are as follows:~~

~~(d) It binds the user identity to the keying material used in the reference points;~~

~~(e) It is the globally unique identifier of the realm of the network operator in which it is issued;~~

~~(f) It serves as a temporary identifier of the trusted device to which it is issued; and~~

~~(g) It identifies the key used in the cryptographic processes over reference point RP_C and RP_D;~~

10 Information flows

This clause specifies procedures for ASPs to access bootstrapping capabilities exposed by network operators in accordance with the functional architecture identified in clause 9. It describes seven (7) major flows that enable trust and service interactions within the ecosystem entities, namely, i) Network operator bootstrapping capability exposure ii) ASP on-boarding flow iii) bootstrap_token generation flow iv) trusted device and application session flow iv) Mapping of trusted device and application v) Authentication and authorisation flow vi) Operator change flow – symmetric Keys vii) Operator change flow – asymmetric keys

10.1 Network operator bootstrapping capability exposure

In order to allow its subscribers to access an ASP's trusted applications, the network operator must enhance its network with certain nodes that implement the bootstrapping functions described in the clause 9 above. The network operator provides the information for users and ASPs to opt for the bootstrapping capability in the network.

The flow is described in the diagram below:

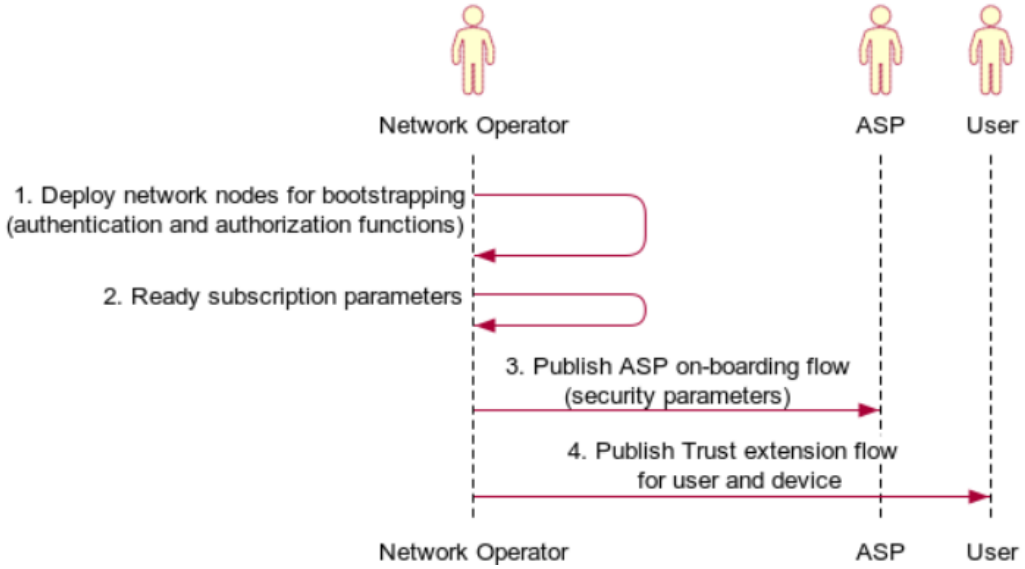


Figure 10-1: Network operator bootstrapping capability exposure

Step 1: Network operator deploys the authentication and authorisation functions in its network.

Step 2: Network operator defines and readies the security parameters as per industry standards.

Step 3: Network operator publishes the ASP registration process with its security parameters. The ASP configures the trusted application with network operator node identifiers to uniquely identify and address the elements in the network operator realm, and complies to the bootstrap_token containing the subscription information to authenticate and authorize the secure interactions between trusted devices and applications via the network operator nodes.

Step 4: Network operator publishes the process for device bootstrapping for subscribers who wish to access ASP trusted applications.

10.2 ASP on-boarding flow

The ASP on-boarding procedure enables ASPs to register themselves and their trusted applications on the network operator authentication and authorisation nodes. The flow readies the trusted applications for registration and controlled access by trusted devices and subscribers of the network operator.

The procedure for ASP on-boarding is shown in the diagram below:

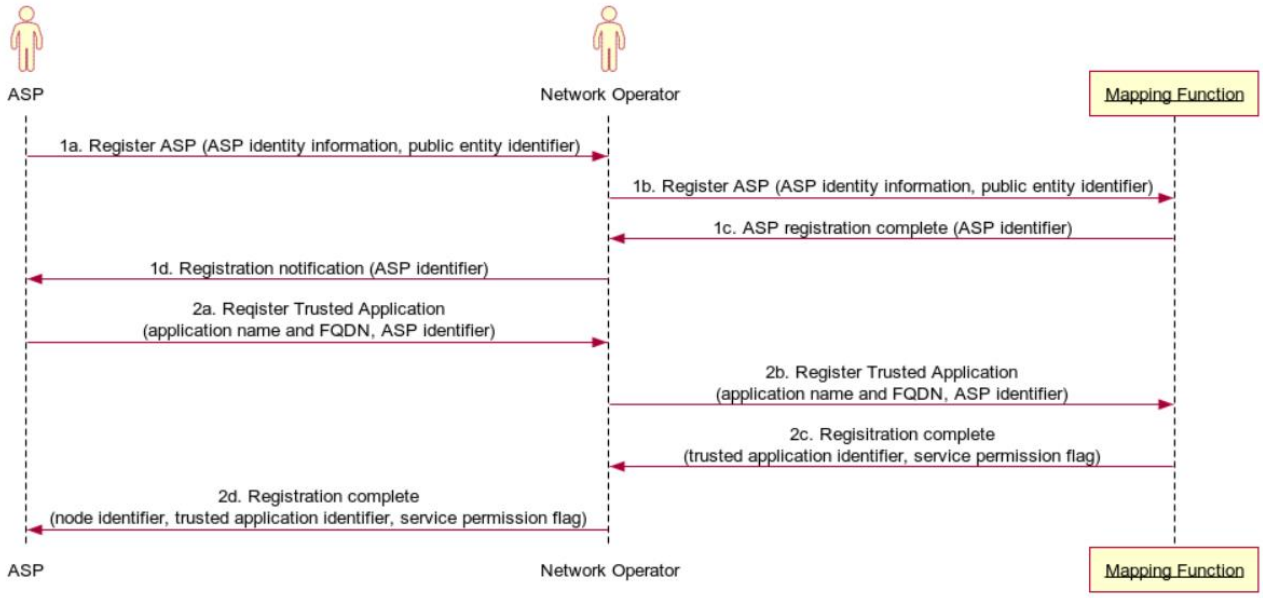


Figure 10-2: ASP on-boarding flow

Step 1a: The ASP initiates the registration with the network operator by providing its identity information, public entity identifier (e.g. UID).

Step 1b: The ASP identity information and public entity identifier are added to the mapping and registration function of the network operator securely.

Step 1c: The mapping and registration function generates a unique identifier for the ASP and sends a notification of successful registration.

Step 1d: The network operator sends ASP its unique identifier upon successful registration.

Step 2a: The ASP initiates the registration of its trusted application with the network operator by providing the application name and FQDN.

Step 2b: The ASP trusted application name and FQDN is added to the mapping and registration function of the network operator securely.

Step 2c: The mapping and registration function generates a unique trusted application identifier and sends a notification of successful registration

Step 2d: The network operator sends the node identifier, trusted application identifier, service permission flag corresponding to the trusted application to the ASP

10.3 Trust extension flow for user and device

The network operator and the ASP inform the network operator's subscribers about the ASP trusted applications. For users that express an interest in ASP's trusted application(s), the network operator checks the user's existing verification information and shares the network identifiers with the ASP if the user credentials merit access to the trusted application(s). The ASP can then assign appropriate permissions for the user access to the trusted application(s).

The process is shown in the diagram below:

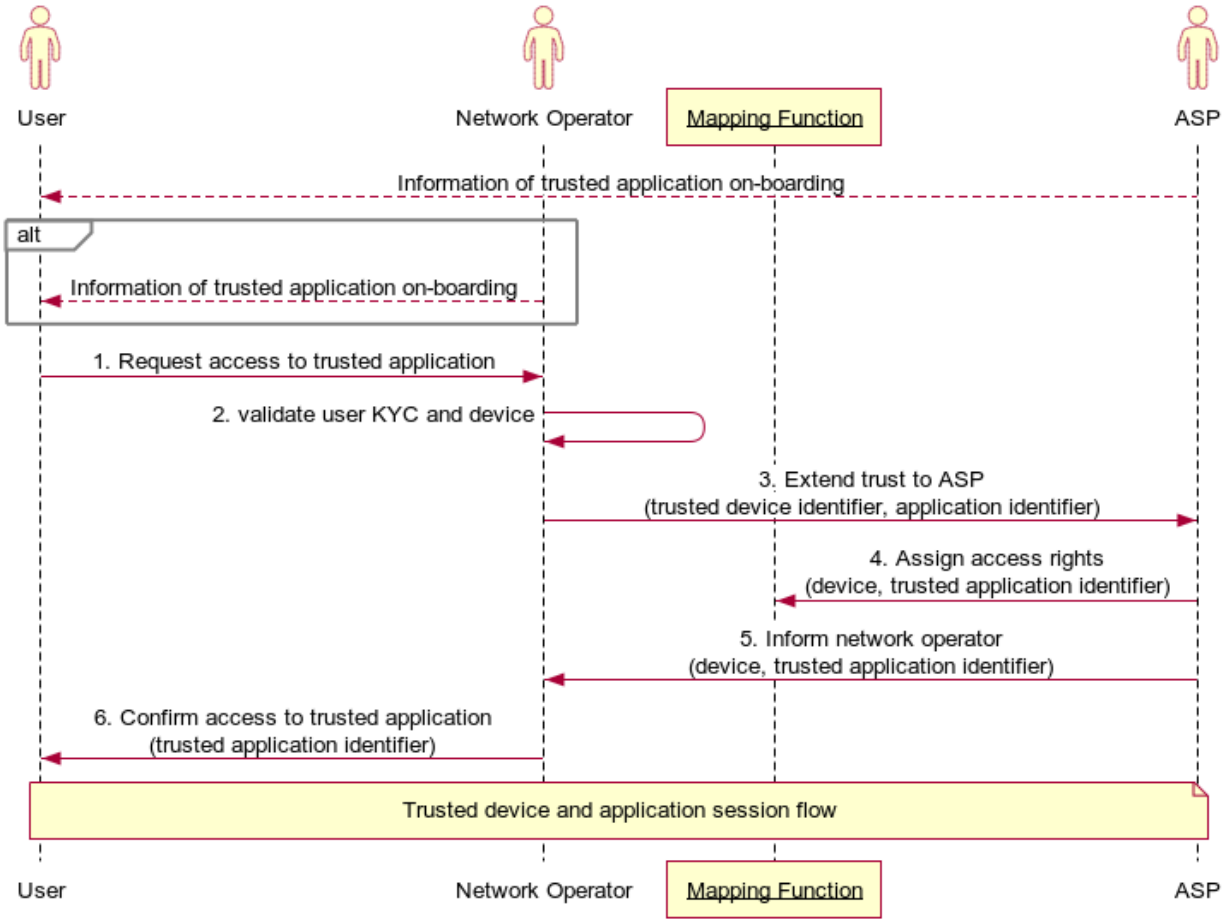


Figure 10-3: Trust extension flow for user and device

Step 1: User requests access to ASP trusted application.

Step 2: Network operator checks user's existing verification and device information.

Step 3: Network operator extends the interested user's trusted device information to the ASP.

Step 4: ASP provisions access rights to the trusted device identifier for the trusted application identifiers on the network operator's mapping and registration function.

Step 5: ASP informs the network operator about the provisioning of access rights to the trusted device for the trusted application.

Step 6: Network operator confirms to the user regarding the access and the trusted application identifiers

After this stage, the trusted device can follow the trusted device and application session flow to initiate the service and trust interactions.

10.4 Bootstrap_token generation flow

The bootstrap_token generation flow enables the generation of the bootstrap_token. It is invoked when a trusted device requests a session with a trusted application but the token management function does not find a valid bootstrap_token to use for the creation of a secure session.

The process is shown in the diagram below:

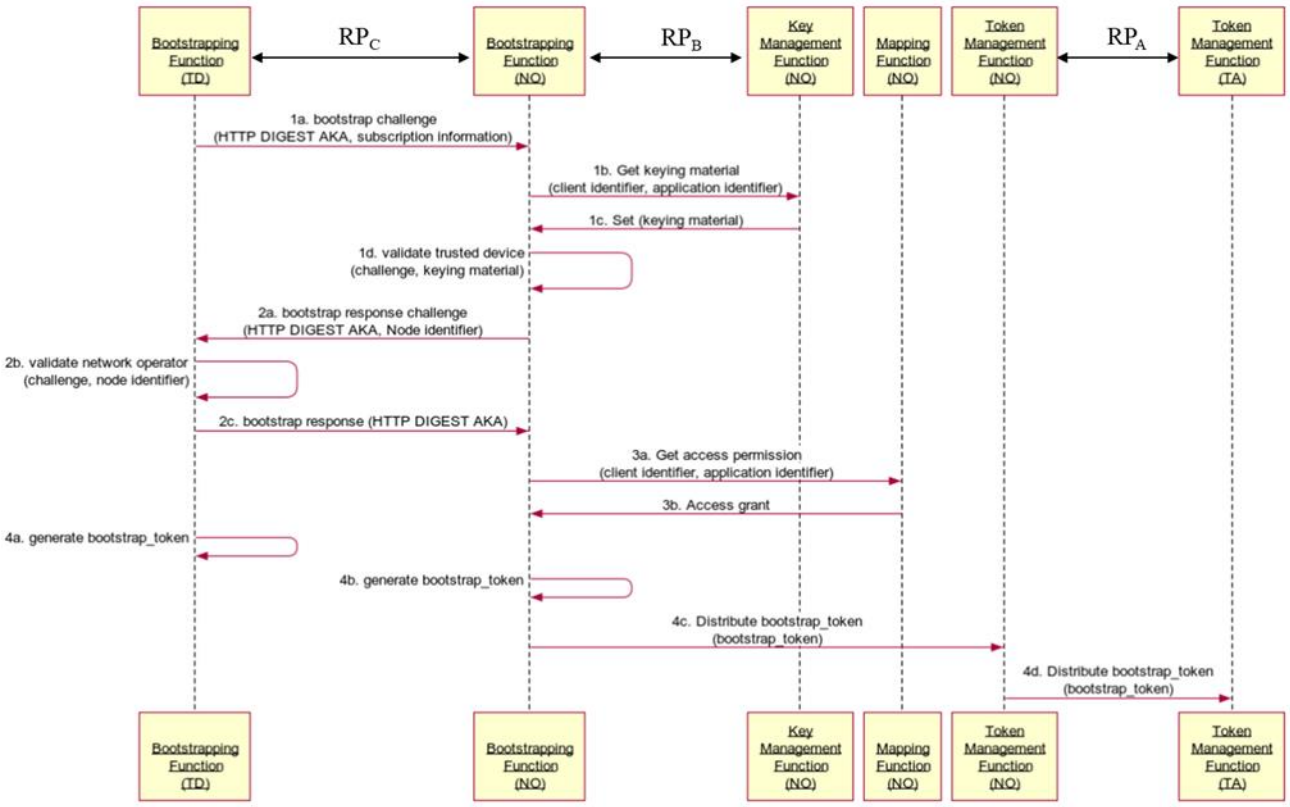


Figure 10-4: Bootstrap_token generation flow

Step 1a: At the start of the bootstrap_token generation process, the bootstrapping function of the trusted device uses the capabilities of the reference point RP_C to send a challenge to the

authentication element using the identifiers of the trusted device and the subscription information of the trusted application.

Step 1b: The bootstrapping function of the network operator uses the capabilities of the reference point RP_B for requesting the key management function for the keying material corresponding to the client element and the application identifier

Step 1c: The key management function sets the keying material for the bootstrapping function of the network operator

Step 1d: The bootstrapping function of the network operator validates the credentials of the client element for based on the keying material set in step 1c above using the HTTP Digest/AKA;

Step 2a: The bootstrapping function of the network operator sends back a challenge to the client element using its node identifier as a part of the security challenge.

Step 2b: The bootstrapping function of the trusted device validates the challenge from the network operator.

Step 2c: The bootstrapping function of the trusted device generates a response based on the challenge and the HTTP Digest/AKA.

Upon the successful mutual authentication, the bootstrapping functions check if the given trusted device is authorized to use the bootstrapping services for a given trusted application.

Step 3a: The bootstrapping function of the network operator requests the mapping and registration function for access permissions by supplying the client identifier and the trusted application identifier information.

Step 3b: The mapping and registration function approves the requested access if the permissions for the trusted device to access the trusted application are set by the ASP as part of the ASP registration process.

Step 4a: Upon successful confirmation in Step 3b, the bootstrapping function of the client element generates the bootstrap_token.

Step 4b: Upon successful confirmation in Step 3b, the bootstrapping function of the network operator generates the bootstrap_token.

Step 4c: The bootstrapping function of the network operator transfers the bootstrap_token to the token management function of the network operator using a proprietary interface.

Step 4d: The token management function of the network operator uses the capabilities of the reference point RP_A to transfer the bootstrap_token securely to the token management function of the trusted application.

At this stage, the token management functions in each of the client element, authentication element and the application element are updated with the newly generated bootstrap_token.

NOTE – The bootstrap_token generation flow shown above shows the use of symmetric keys for the establishment of secure session~~connection~~s; the flow with asymmetric keys is similar, with the exception that, in place of pre-shared keys the public keys are used for bootstrapping. That flow is not shown explicitly.

10.5 Trusted device and application session flow

The trusted device and application session flow establishes a secure session over which the service interactions can be carried out. The flow is described in the diagram below:

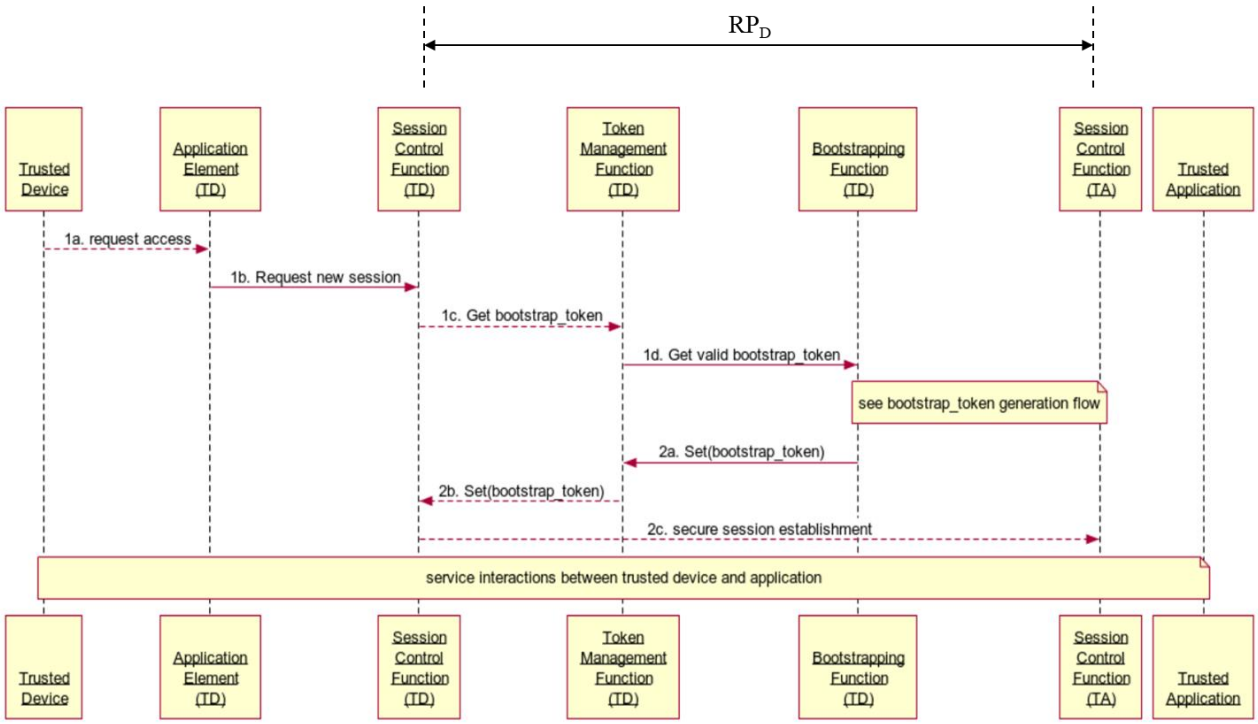


Figure 10-5: Trusted device and application session flow

Step 1a: Trusted device requests access to a trusted application.

Step 1b: The Application element of the trusted device requests a session.

Step 1c: The session control function of the trusted device requests the token management function for a valid bootstrap token.

Step 1d: The token management function either has a valid token, or requests the bootstrapping for a new bootstrap_token.

At this stage, the bootstrap_token generation flow is called if a new bootstrap_token is required.

Step 2a: The token management function gets the bootstrap_token from the bootstrapping function.

Step 2b: The token management function sets the bootstrap_token for session control function.

Step 2c: The session control function establishes a secure session over the reference point RP_D.

At this stage, the trusted device and application can initiate service interactions over the secure session.

10.6 Flow for change of network operator

A user that is a beneficiary of the bootstrapping capabilities provided by a network operator may require to change the network operator, but may want to continue the use of trusted services which were supported by the network operator. A process will be required for the transfer of bootstrapping capabilities from the old network operator to the new network operator. The changing of the network operator bootstrapping realm is enabled by the process defined below.

10.6.1 Change of network operator flow (symmetric keys)

The process for change of network operator offering bootstrapping services is shown in the diagram below:

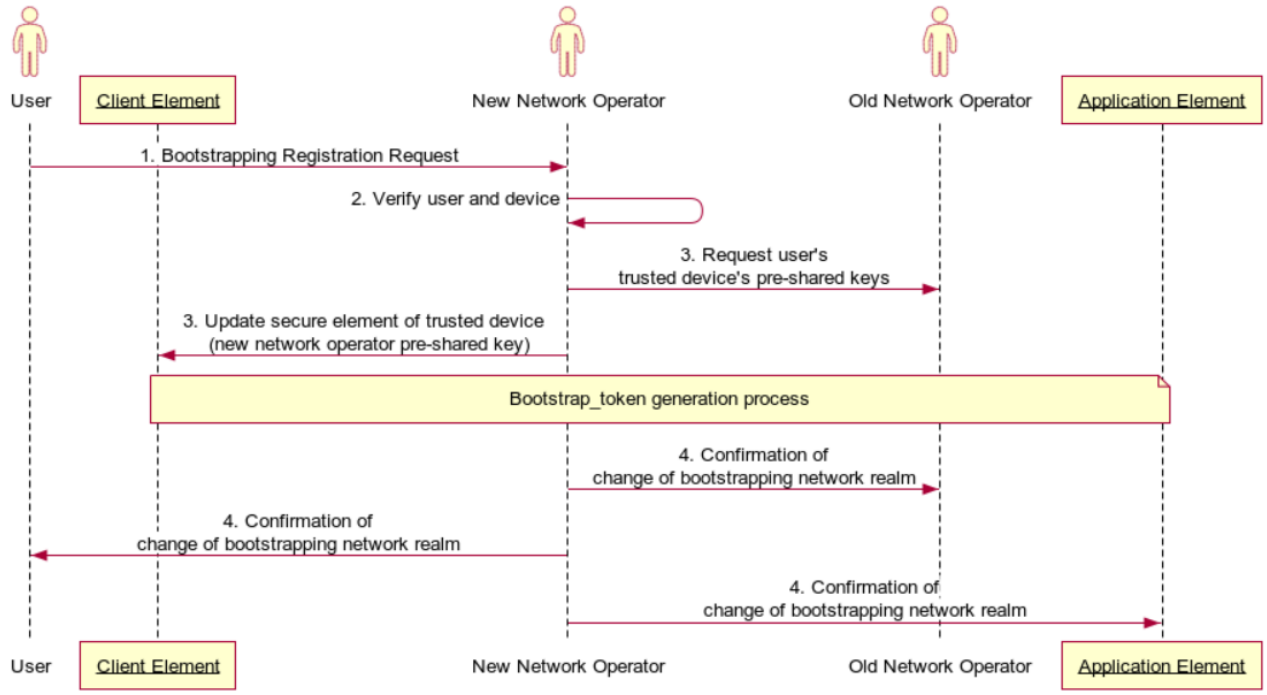


Figure 10-6: Change of network operator (symmetric keys)

Step 1: The user of the trusted application approaches the new network operator registration to the new network operator bootstrapping capabilities for access to trusted applications.

Step 2: The new network operator undertakes the verification of the user and the trusted device (machine KYC) and upon successful verification, requests the old network operator for the user's pre-shared keys.

Step 3: The new network operator updates the secure element of the user's trusted device with its own pre-shared key(s).

After this stage, the trusted device of the user is on-boarded to the new network operator as per the bootstrap_token generation flow.

Step 4: Upon success, the new network operator informs the user and the old network operator of the successful on-boarding of the user's trusted device to the new network operator.

NOTE – Machine KYC is the process of establishing a relationship between a machine and its user, usually accomplished by the network operator or IoT service provider by the use of physical or digital verification processes that establish the linkage between the identity of the user and the identity of the trusted device owned by the user.

10.6.2 Change of network operator flow (asymmetric keys)

In case asymmetric keys are used for authentication, the steps for change of the network operator are described in the diagram below:

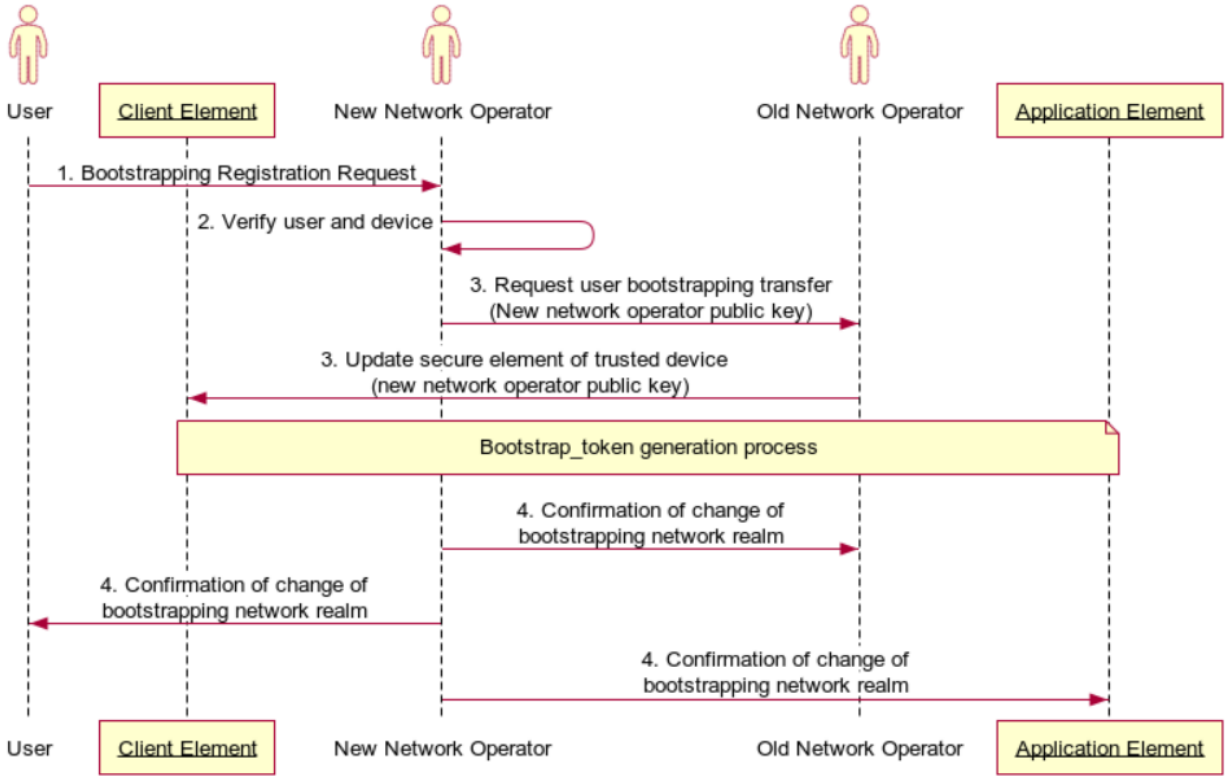


Figure 10-7: Change of network operator (asymmetric keys)

Step 1: The user of the trusted application approaches the new network operator registration to the new network operator bootstrapping capabilities for access to trusted applications.

Step 2: The new network operator undertakes the verification of the user and the trusted device (machine KYC) and upon successful verification, requests the old network operator to update the secure element of the user's trusted device by replacing the old network operator's public key(s) with those of the new network operator.

Step 3: The old network operator updates the secure element of the user's trusted device with the public key(s) of the new network operator.

After this stage, the trusted device of the user is on-boarded to the new network operator as per the Bootstrap_token generation flow.

Step 4: Upon success, the new network operator informs the user and the old network operator of the successful on-boarding of the user's trusted device to the new network operator.

Bibliography

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

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

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

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

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

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

[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).*

_____