International Telecommunication Union

ITU-T

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

Y.3056

(02/2021)

SERIES Y: GLOBAL INFORMATION INFRASTRUCTURE, INTERNET PROTOCOL ASPECTS, NEXT-GENERATION NETWORKS, INTERNET OF THINGS AND SMART CITIES

Future networks

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

Recommendation ITU-T Y.3056

ITU-T Y-SERIES RECOMMENDATIONS

GLOBAL INFORMATION INFRASTRUCTURE, INTERNET PROTOCOL ASPECTS, NEXT-GENERATION NETWORKS, INTERNET OF THINGS AND SMART CITIES

GLOBAL INFORMATION INFRASTRUCTURE	
General	Y.100–Y.199
Services, applications and middleware	Y.200–Y.299
Network aspects	Y.300–Y.399
Interfaces and protocols	Y.400–Y.499
Numbering, addressing and naming	Y.500–Y.599
Operation, administration and maintenance	Y.600–Y.699
Security	Y.700–Y.799
Performances	Y.800–Y.899
INTERNET PROTOCOL ASPECTS	
General	Y.1000–Y.1099
Services and applications	Y.1100–Y.1199
Architecture, access, network capabilities and resource management	Y.1200–Y.1299
Transport	Y.1300–Y.1399
Interworking	Y.1400–Y.1499
Quality of service and network performance	Y.1500–Y.1599
Signalling	Y.1600–Y.1699
Operation, administration and maintenance	Y.1700–Y.1799
Charging	Y.1800–Y.1899
IPTV over NGN	Y.1900–Y.1999
NEXT GENERATION NETWORKS	
Frameworks and functional architecture models	Y.2000–Y.2099
Quality of Service and performance	Y.2100–Y.2199
Service aspects: Service capabilities and service architecture	Y.2200–Y.2249
Service aspects: Interoperability of services and networks in NGN	Y.2250–Y.2299
Enhancements to NGN	Y.2300–Y.2399
Network management	Y.2400–Y.2499
Network control architectures and protocols	Y.2500–Y.2599
Packet-based Networks	Y.2600–Y.2699
Security	Y.2700–Y.2799
Generalized mobility	Y.2800–Y.2899
Carrier grade open environment	Y.2900–Y.2999
FUTURE NETWORKS	**Y.3000–Y.3499**
CLOUD COMPUTING	Y.3500–Y.3599
BIG DATA	Y.3600–Y.3799
QUANTUM KEY DISTRIBUTION NETWORKS	Y.3800–Y.3999
INTERNET OF THINGS AND SMART CITIES AND COMMUNITIES	
General	Y.4000–Y.4049
Definitions and terminologies	Y.4050–Y.4099
Requirements and use cases	Y.4100–Y.4249
Infrastructure, connectivity and networks	Y.4250–Y.4399
Frameworks, architectures and protocols	Y.4400–Y.4549
Services, applications, computation and data processing	Y.4550–Y.4699
Management, control and performance	Y.4700–Y.4799
Identification and security	Y.4800–Y.4899
Evaluation and assessment	Y.4900–Y.4999

For further details, please refer to the list of ITU-T Recommendations.

Recommendation ITU-T Y.3056

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 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. Recommendation ITU-T Y.3056 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, this Recommendation provides the information flows required to enable the bootstrapping capabilities.

History

Edition	Recommendation	Approval	Study Group	Unique ID*
1.0	ITU-T Y.3056	2021-02-13	13	11.1002/1000/14594

Keywords

Authentication, authorization, bootstrapping, bootstrap_token, trusted application, trusted device

* To access the Recommendation, type the URL http://handle.itu.int/ in the address field of your web browser, followed by the Recommendation's unique ID. For example, http://handle.itu.int/11.1002/1000/11830-en.

FOREWORD

The International Telecommunication Union (ITU) is the United Nations specialized agency in the field of telecommunications, information and communication technologies (ICTs). The ITU Telecommunication Standardization Sector (ITU-T) is a permanent organ of ITU. ITU-T is responsible for studying technical, operating and tariff questions and issuing Recommendations on them with a view to standardizing telecommunications on a worldwide basis.

The World Telecommunication Standardization Assembly (WTSA), which meets every four years, establishes the topics for study by the ITU-T study groups which, in turn, produce Recommendations on these topics.

The approval of ITU-T Recommendations is covered by the procedure laid down in WTSA Resolution 1.

In some areas of information technology which fall within ITU-T's purview, the necessary standards are prepared on a collaborative basis with ISO and IEC.

NOTE

In this Recommendation, the expression "Administration" is used for conciseness to indicate both a telecommunication administration and a recognized operating agency.

Compliance with this Recommendation is voluntary. However, the Recommendation may contain certain mandatory provisions (to ensure, e.g., interoperability or applicability) and compliance with the Recommendation is achieved when all of these mandatory provisions are met. The words "shall" or some other obligatory language such as "must" and the negative equivalents are used to express requirements. The use of such words does not suggest that compliance with the Recommendation is required of any party.

INTELLECTUAL PROPERTY RIGHTS

ITU draws attention to the possibility that the practice or implementation of this Recommendation may involve the use of a claimed Intellectual Property Right. ITU takes no position concerning the evidence, validity or applicability of claimed Intellectual Property Rights, whether asserted by ITU members or others outside of the Recommendation development process.

As of the date of approval of this Recommendation, ITU had not received notice of intellectual property, protected by patents/software copyrights, which may be required to implement this Recommendation. However, implementers are cautioned that this may not represent the latest information and are therefore strongly urged to consult the appropriate ITU-T databases available via the ITU-T website at http://www.itu.int/ITU-T/ipr/.

Table of Contents

Recommendation ITU-T Y.3056

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]: 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]: 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]: 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]: 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. Trust 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]: 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 term:

3.2.1 bootstrapping: A cryptographic process of binding user identity(ies) to the keying material provisioned in the secure element of the user's device, enabling the device to communicate securely with trusted services.

NOTE – Bootstrapping for open access acknowledges the existence and need for one or more identities, which may be associated with any of the bootstrapping for open access actors e.g., user, user's network, user's device, application services provider (ASP) or trusted application.

4 Abbreviations and acronyms

This Recommendation uses the following abbreviations and acronyms:

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
GSM	Global System for Mobile communication
HTTP	Hyper Text Transfer Protocol
ICT	Information and Communication Technology
IMEI	International Mobile Equipment Identity
IMSI	International Mobile Subscriber Identity
IoT	Internet of Things
IPSec	Internet Protocol Security
KYC	Know Your Customer
MAC	Media Access Control
MSISDN	Mobile Subscriber International Services Directory Number
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
USIM	Universal Subscriber Identity Module

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 Introduction

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

6.1 Concept of trusted services

Certain services require additional checks prior to making them available to a beneficiary. A license to drive cars, access to restricted premises, permissions for online banking are a 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.

NOTE – This Recommendation incorporates support for the establishment of trust by the operator for the user and user's device. The reciprocal aspect, in which the user or the user's device establishes the trustworthiness of the network/network operator is also important, especially in the context of the imminent proliferation of private networks, and may be a candidate for further standardization work.

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).

The network operator creates a measurable degree of trust by verifying its subscribers as per the regulatory or ecosystem norms. As an example, the requirement for verification of subscribers differs when considering prepaid versus post-paid users, or when considering the offer of domestic versus international roaming services. The trust generated by the network operator may create a basis for a differentiated offering of trusted services to a user. For example, a prepaid user that has a lower degree of trustworthiness may be permitted to use non-critical services by an ASP, whereas a post-paid user, verified to a higher degree of trust, may be offered a different class of services.

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 concepts 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 subscriber identification module (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:

– the application for which it is generated;

– the device on which it is generated; and

– 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:

– a device already registered in a network is given certain additional privileges;
– the device and the network have an agreed AKA for the generation of secure tokens; and
– 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 realms available for bootstrapping at one time.

6.3 Role of network operators in enabling trusted services

The approach to securing trusted services by the extension of operator trust through bootstrapping of devices is shown in Figure 6-1.

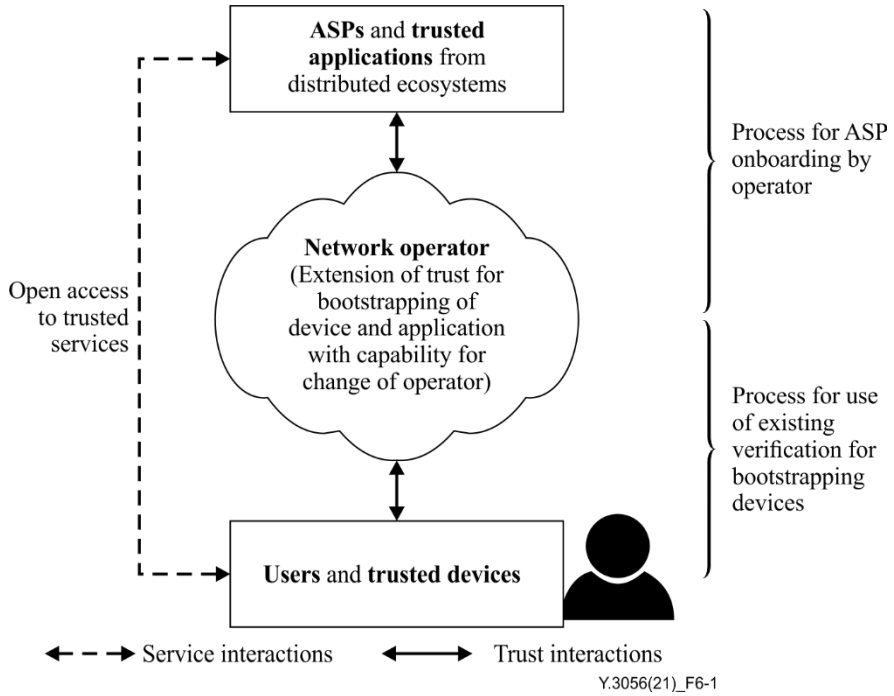

Figure 6-1 - Role of network operator in connecting diverse ecosystems of trusted services

The bootstrapping concept involves the following entities:

– **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.

– **User**: A person that is a verified subscriber of the network operator, desirous of using trusted services from ASPs.

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.

– **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.

– **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.

– **Trusted application**: ASP application on-boarded by the network operator, which is capable of controlling access to users of trusted devices using cryptographic capabilities.

The onboarding of an ASP by the network operator requires that the parties enter into certain mutual covenants that allow the extension of operator's trust (of the user and the user's device) for the benefit of the ASP and ASP's trusted applications.

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 must follow as per existing standards and norms.

The following pre-conditions are applicable for this Recommendation:

– 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., company identification number (CIN));

– reuse the identification and numbering of trusted applications (e.g., IP address/fully qualified domain name (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;

– use identities representing each of, the network, the trusted device and the trusted application for purposes of mutual authentication transactions; and

– support the existence of, and choose from, the multiple network operators that may be offering bootstrap capabilities.

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 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., international mobile equipment identity (IMEI) or MAC), the network such as mobile subscriber international services directory number (MSISDN) or international mobile subscriber identity (IMSI) for the global system for mobile communication (GSM) and applications (e.g., IP, URL, etc.) which together fulfil the requirements for addressing within communication protocols.

7.2 Requirements for the security token

The security 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 for the cryptographic processes that establish a secure session between the trusted device and the trusted application;

– bind the identities 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 in the device and the network, as per the security protocols and parameters published by the network operator.

NOTE – Hereinafter, the security token of an operator realm is referred to as 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.

7.4 Requirements for the trusted device entity

The trusted device is required to have:

– capability for 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 access to trusted applications using session controls and end to end data privacy and security.

7.5 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 enhancements in its network for 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;
– have capabilities for storage and retrieval of keys and other sensitive data for enabling trust interactions;
– protect the subscriber's identity and the device's network identity; and
– allow the user to change its device's registration for bootstrapping to a different network operator.

7.6 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 that provide secure access to only such network operator registered devices that have a valid subscription;
– provide the protocols and capabilities that enable a trusted device to interact with it securely; and
– establish secure sessions with the trusted device using the bootstrap_token.

7.7 Requirements for the ASP entity

The ASP is required to:

– register with the network operator(s) using its public identity (e.g., CIN);
– 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.);
– 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.

8 Reference model

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

The reference model is described in Figure 8-1.

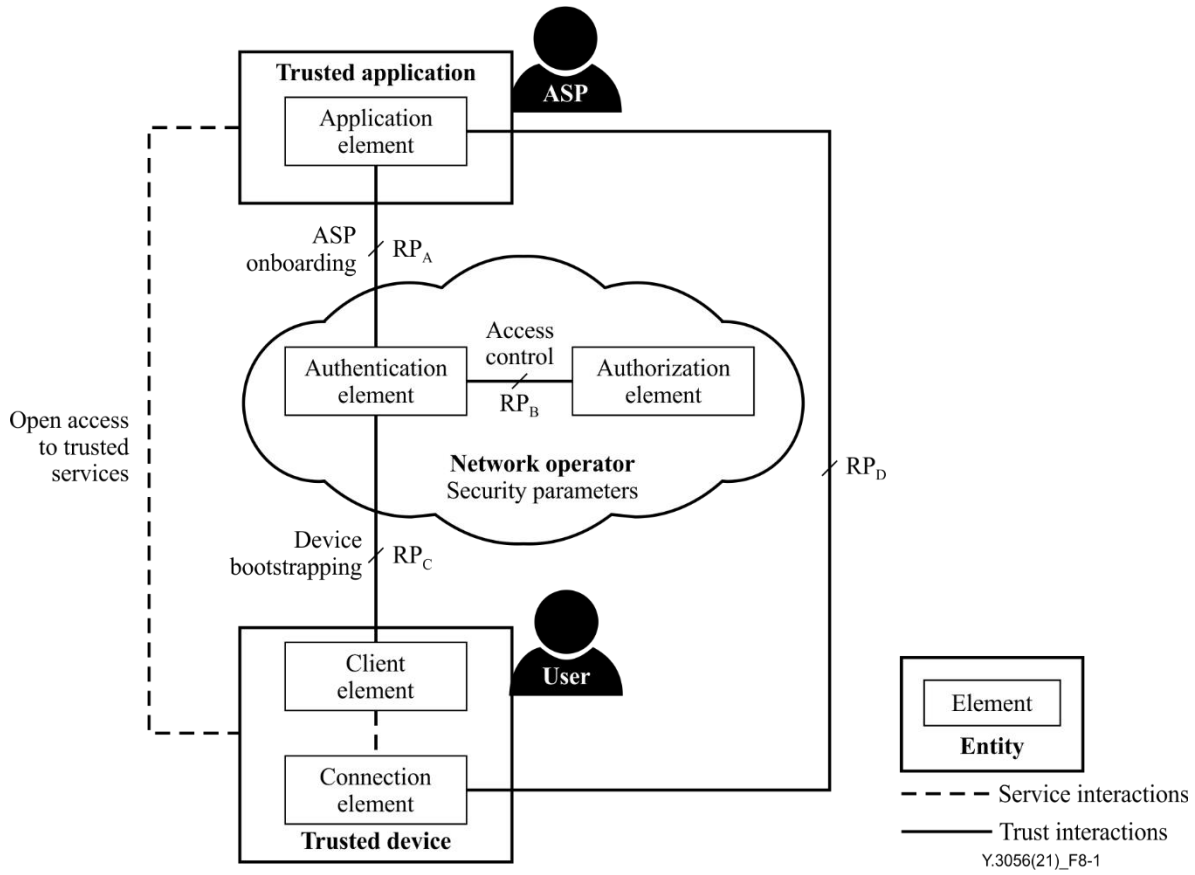

Figure 8-1 – Reference model

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 in clauses 8.1.1 and 8.1.2.

8.1.1 Client element

The client element is a 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 Connection element

This element is a part of the trusted application, responsible for setting up the secure session between the trusted device and application using the bootstrap_token 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 in clauses 8.2.1 and 8.2.2.

8.2.1 Authentication element

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

8.2.2 Authorization element

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 standardized protocols (e.g., OAUTH, DIAMETER, etc.) and the mapping of the access controls between the trusted devices and applications.

8.3 Application element

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

The application element sets up the secure sessions between the trusted devices and applications using the network operator specified protocols and security parameters. The application element is deployed in each trusted application.

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 1 – 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, 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 2 – Subscription information can be as per [ITU-T X.1124].

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 described in the clauses 9 and 10.

9 Functional architecture

The bootstrapping requirements can be realized as per the functional architecture shown in Figure 9-1, which presents the required functions and the interfaces between the entities.

NOTE 1 – An implementation of the bootstrapping functional architecture by a network operator is referred to as a realm. The instantiated 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.

NOTE 2 – The functional architecture limits itself to the entities, elements and functions which are necessary to be specified from the perspective of this Recommendation. However, there are other important interactions between the actors such as the user(s), the ASP(s) and the network operator(s) which are necessary to enable the bootstrapping facilities but are left to the choice of the actor that has to enable the facility. The details of some such interactions are more fully described in clause 10 on information flows.

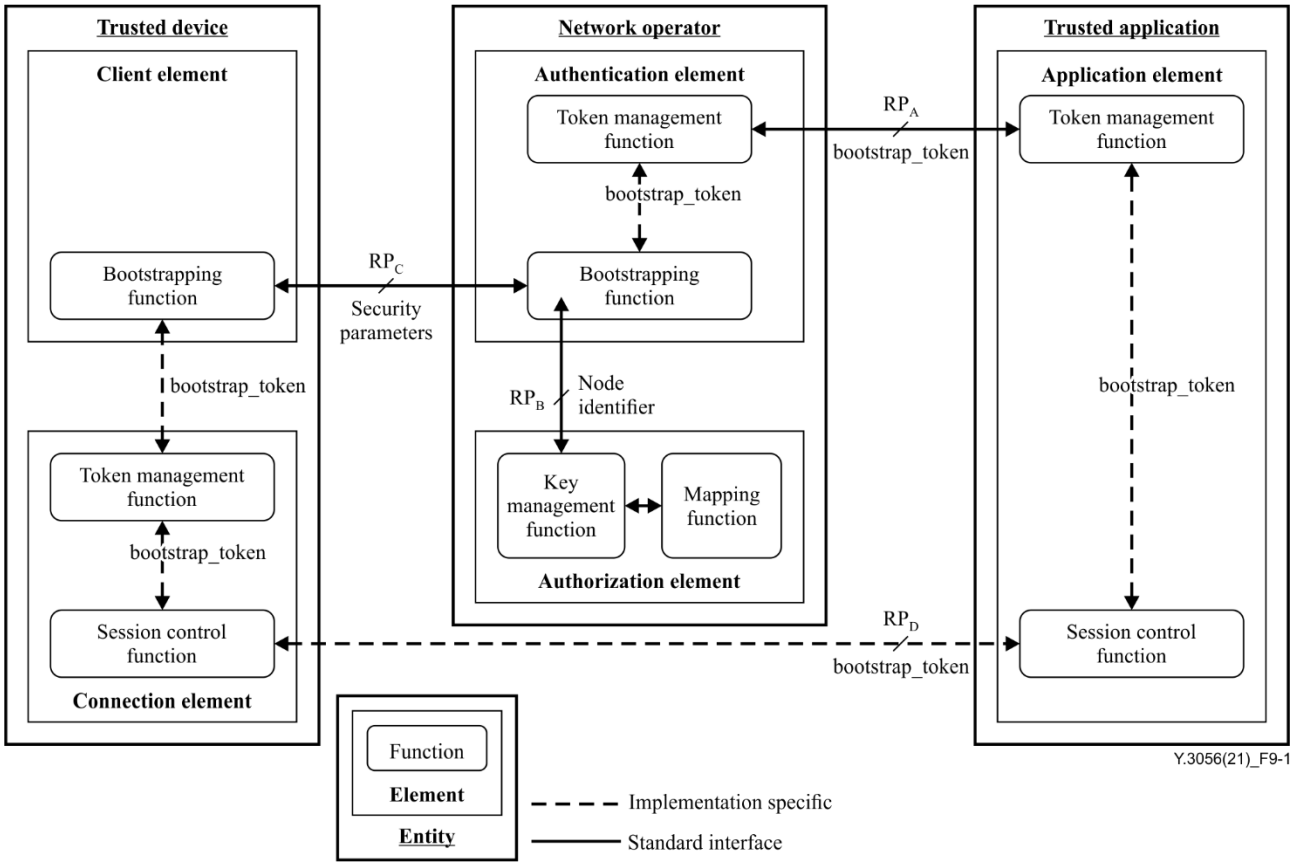

Figure 9-1 – Functional architecture

9.1 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 in clause 9.1.1, whereas the token management function is described in clause 9.4.

9.1.1 Bootstrapping function

The bootstrapping function is responsible for the mutual authentication between the client element of the trusted device and the authentication element of the network operator.

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 user's network identity against discovery and misuse during the trust and service interactions with the trusted application.

9.2 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. It maintains the secure identities 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 in clauses 9.2.1 and 9.2.2.

9.2.1 Key management function

This function provides the management, storage and retrieval of keys and other sensitive data corresponding to the trusted devices. 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 registration 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 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 mapping and registration function provides the following functionalities:

– registers 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.);

– registers 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.);

– registers the ASP that has the 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);

– supports the addition / deletion of authorized client element of trusted devices;

– supports the delegation / revocation of access control rights to authorized client element(s) through standardized API or user interfaces; and

– supports the protocols required over the reference point RP_B.

9.3 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;

– generates the bootstrap_token as per the format and security parameters and manages the bootstrap_token lifecycle as specified by the authentication element during the bootstrapping process;
– selects the bootstrap_token corresponding to the active network operator realm; and
– allows only one bootstrap_token to be active per application at a given point in time.

NOTE – A trusted device may have access to several trusted applications through various network realms at any given point in time. Also, it is possible that a trusted device has access to subscriptions from several network realms, which may lead to the existence and storage of multiple bootstrap_tokens from multiple network operator realms in the client element of a trusted device.

9.4 Token management function

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.

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.

9.4.3 Token management function of the trusted application

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

– receiving 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.

9.5 Session control function

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 function is implemented using session control protocols such as transport layer security (TLS), pre-shared key cipher suites for transport layer security (PSK-TLS), Kerberos and Internet protocol security (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). 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 in clauses 9.6.1 to 9.6.4.

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;
– allows the application to send its address (e.g., FQDN), public identity (e.g., UID), basic key material (e.g., a shared secret or a public-key certificate), 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 standardized protocols e.g., the Diameter protocol described in [b-IETF RFC 6733] and [b-IETF RFC 7155].

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 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 service permission flag for the client element to access certain trusted applications.

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

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.

NOTE – The characteristics of the reference point may be fully met by standardized protocols e.g., the HTTP Digest protocol [b-IETF RFC 7616].

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 standardized protocols e.g., certificate-based TLS authentication mechanism, PSK-TLS, IPSec, etc.

10 Information flows

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

10.1 Network operator bootstrapping capability exposure

NOTE – The implementation aspects by which the network operator informs and onboards its subscribers as part of the bootstrapping process is left to the choice of the network operator.

In order to allow its subscribers to access an ASP's trusted applications, the network operator provides the information for users and ASPs to opt for the bootstrapping capability in the network.

The flow is described in Figure 10-1.

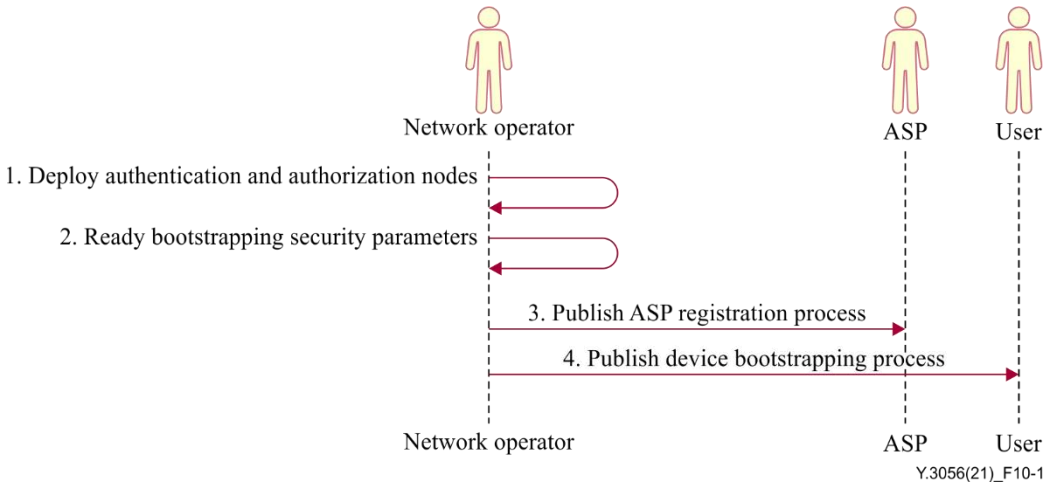

Figure 10-1 – Network operator bootstrapping capability exposure

Step 1: Deployment of an authentication as well as an authorization node within the network operator realm as per the bootstrapping functional architecture.

Step 2: Network operator readies the bootstrapping realm security parameters including the bootstrap_token format and the IP address/FQDN of the authentication and authorization nodes.

Step 3: Network operator publishes the ASP registration process. The ASP configures the trusted application with IP zddress/FQDN of the network operator nodes and complies with the bootstrap_token.

Step 4: Network operator publishes the process for device bootstrapping.

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 procedure for ASP on-boarding is shown in Figure 10-2.

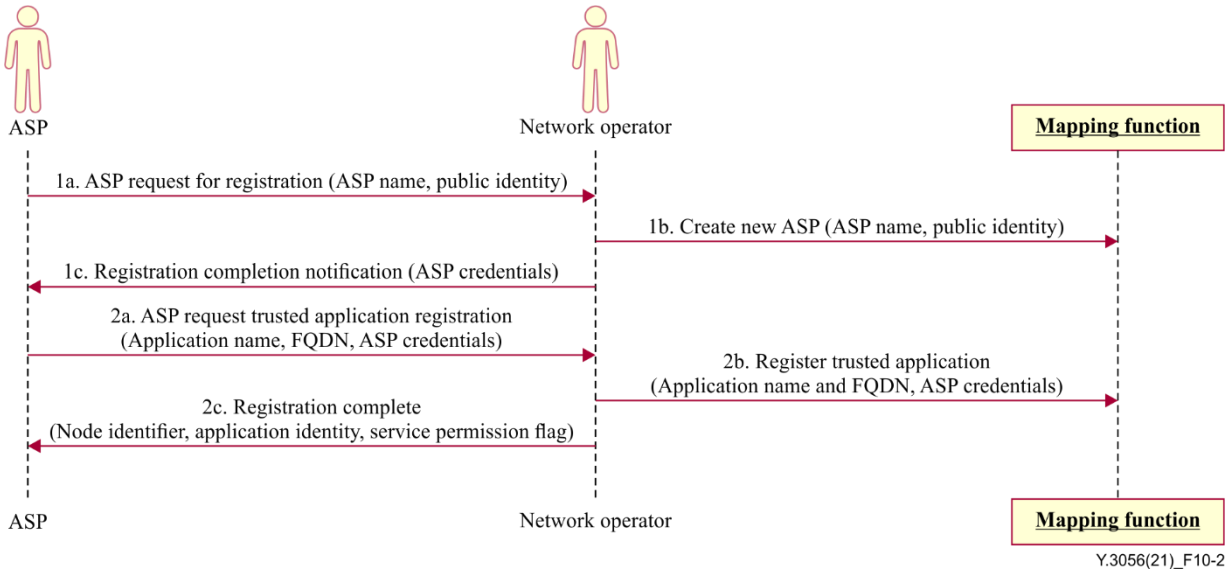

Figure 10-2 – ASP on-boarding flow

NOTE 1 – The implementation aspects of the process of ASP on-boarding is left to the choice of the network operator.

NOTE 2 – The mapping function is short for the mapping and registration function.

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

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

Step 1c: The network operator provides ASP the credentials for secure access to the nodes within the network realm.

Step 2a: The ASP initiates the registration of its trusted application with the network operator by providing the identity of trusted applications (e.g., IP address/FQDN, URL, oneM2M App-ID, GS1 application ID, etc.).

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

Step 2c: The mapping and registration function sends a notification of successful registration.

10.3 Trust extension flow for user and device

NOTE – The implementation aspects of the process by which the network operator informs its subscribers about ASP's services is left to the choice of the network operator.

For users that express an interest in ASP's trusted application(s), the network operator checks the user's existing verification information and securely provides the user's credentials as registered in the network operator's subscriber verification database to the ASP. The ASP assigns appropriate service permission flag for the user to access the trusted application(s).

The process is shown in Figure 10-3.

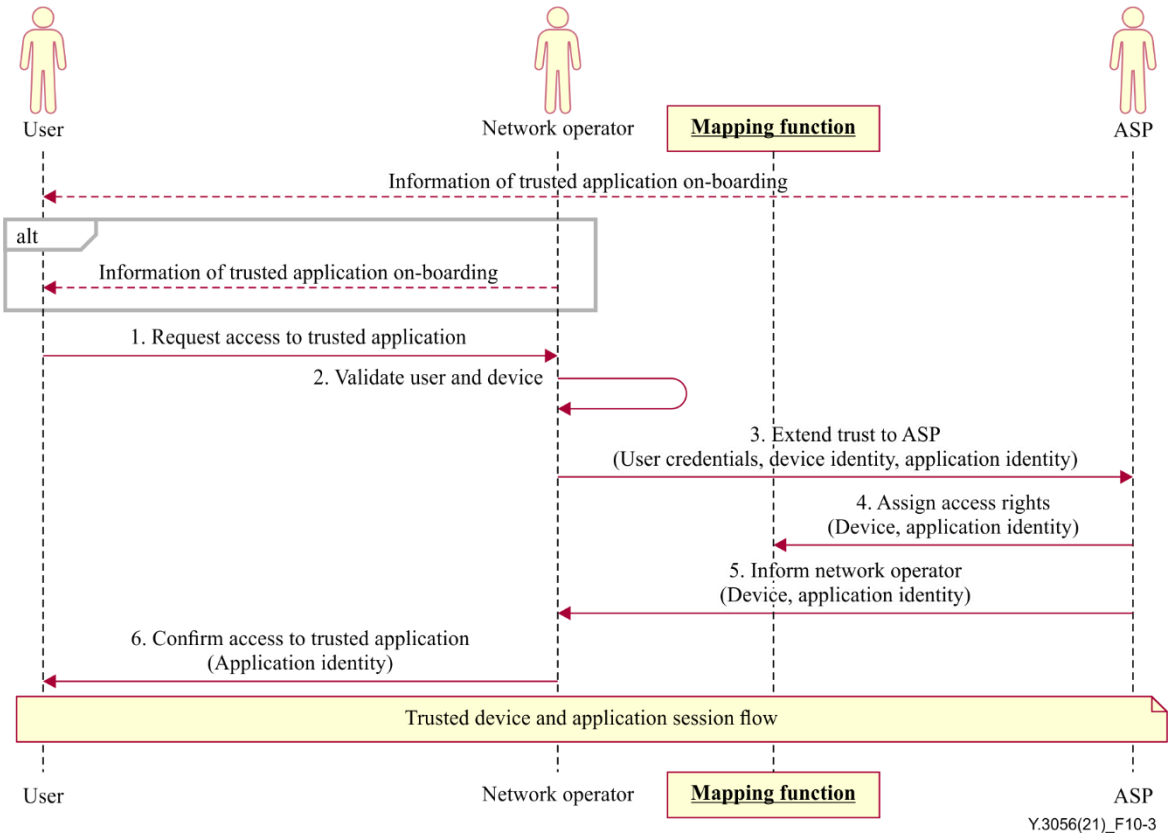

Figure 10-3 – Trust extension flow for user and device

Step 1: User requests network operator for access to an ASP's trusted application(s).

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

Step 3: Network operator extends the interested user's credentials and the identity of the client element of the trusted device entity to the ASP.

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

Step 5: ASP informs the network operator about the provisioning of access rights.

Step 6: Network operator informs the user regarding the provisioning of access to the trusted application.

At this stage, the trusted device and application are provisioned for service and trust interactions and can proceed to bootstrap token generation described in clause 10.4 for undertaking secure interactions which are further detailed in clause 10.5.

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 Figure 10-4.

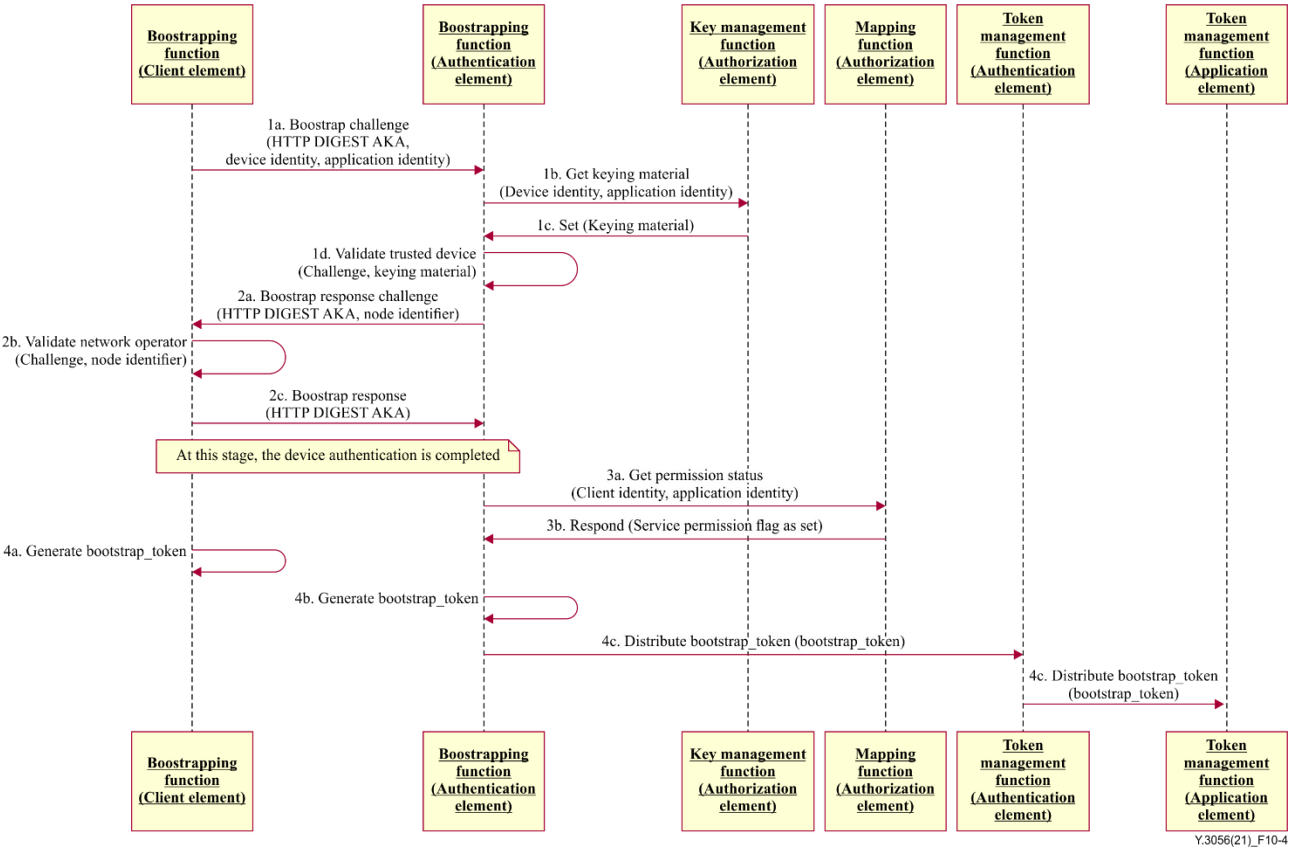

Figure 10-4 – Bootstrap_token generation flow

Step 1a: At the start of the bootstrap_token generation process, the bootstrapping function of the client element of the trusted device uses the capabilities of the reference point RP_C to send a challenge to the authentication element using its identity and that 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 that client element and the trusted application.

Step 1c: The key management function sets the keying material in the bootstrapping function of the network operator's authorization element.

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

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

Step 2b: The bootstrapping function in the client element of the trusted device validates the challenge from the network operator's authorization element.

Step 2c: The bootstrapping function in the client element 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 in the network operator's authorization element requests the mapping and registration function for the service permission flag by supplying the client identity and the trusted application identity.

Step 3b: The mapping and registration function responds with the service permission flag as set by the ASP as part of the trust extension flow described in clause 10.3.

If service permission flag is set for the combination of trusted device and application, the next steps for bootstrap_token generation and distribution are carried out:

Step 4a: The bootstrapping function of the client element generates the bootstrap_token for that application.

Step 4b: The bootstrapping function of the network operator's authorization element generates the bootstrap_token.

Step 4c: The bootstrapping function transfers the newly generated bootstrap_token securely to the token management function within the network operator's authentication element.

Step 4d: The token management function of the network operator's authentication element uses the capabilities of the reference point RP_A to transfer the bootstrap_token securely to the token management function of the application element 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 sessions; 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

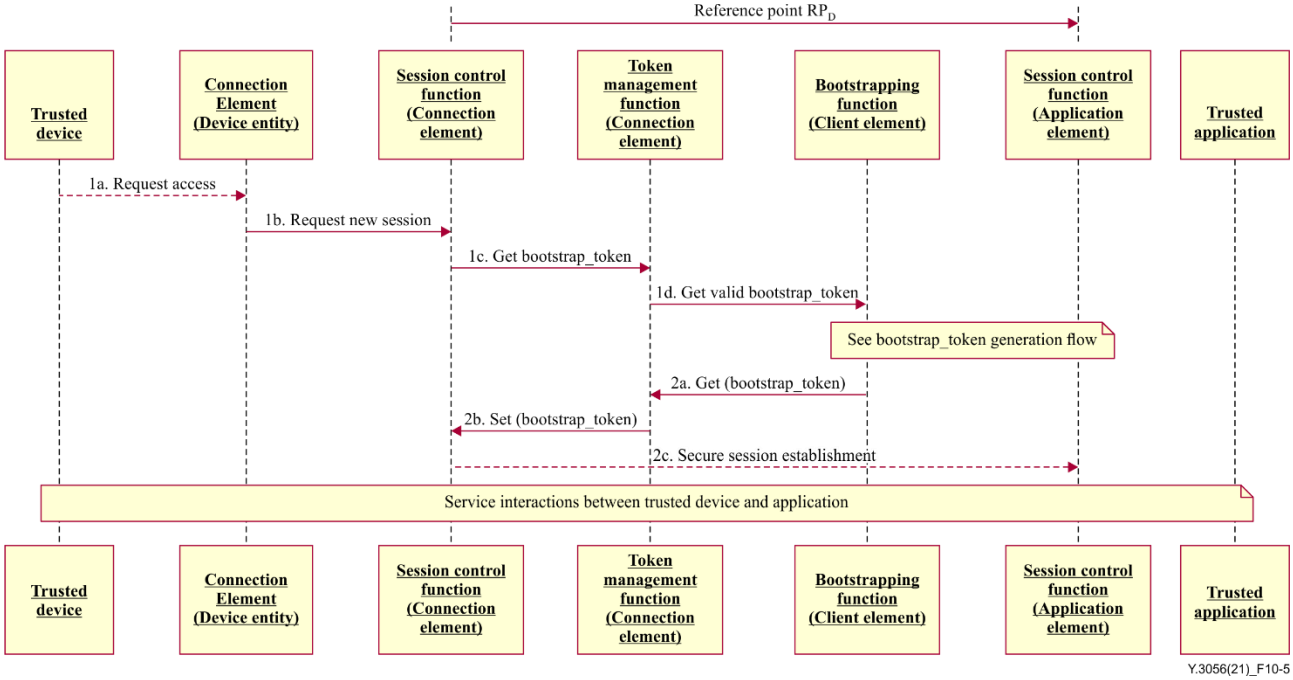

Figure 10-5 – Trusted device and application session flow

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 Figure 10-5.

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

Step 1b: The session control function of the connection element of the trusted device requests a secure session.

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

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

NOTE –The secure session is built on top of a connection established between the trusted device, the network and the trusted application based on the underlying device, network and ASP technology in use.

10.6 Flow for change of network operator

A user of the bootstrapping services provided by a network operator realm may require to change the network operator. The changing of the network operator bootstrapping realm is enabled by the process defined in clause 10.6.1.

10.6.1 Change of network operator flow (symmetric keys)

Figure 10-6 shows the change of network operator flow (symmetric keys).

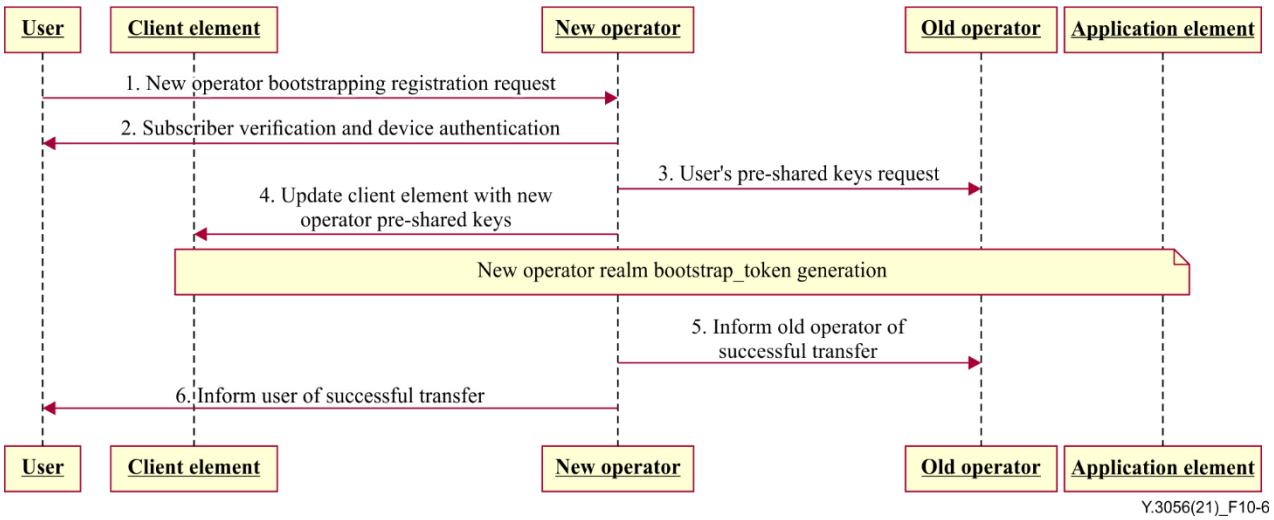

Figure 10-6 – Change of network operator (symmetric keys)

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

Step 2: The new network operator undertakes the subscriber verification and the trusted device authentication.

Step 3: New operator requests the old network operator for the user's pre-shared keys.

NOTE – Key sharing procedures may be mutually agreed between parties and implemented accordingly.

Step 4: The new network operator uses the pre-shared keys of the old network operator to authenticate and update the client element of the user's trusted device with its own pre-shared key(s).

At this stage, the trusted device may invoke the bootstrap_token generation flow for generating the bootstrap_token of the new network operator realm.

Step 5: Upon successful generation of bootstrap_token in the new realm, the new network operator informs the old network operator of the change in bootstrapping realm.

Step 6: Upon successful generation of bootstrap_token in the new realm, the new network operator informs the user of the change in bootstrapping realm.

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 Figure 10-7.

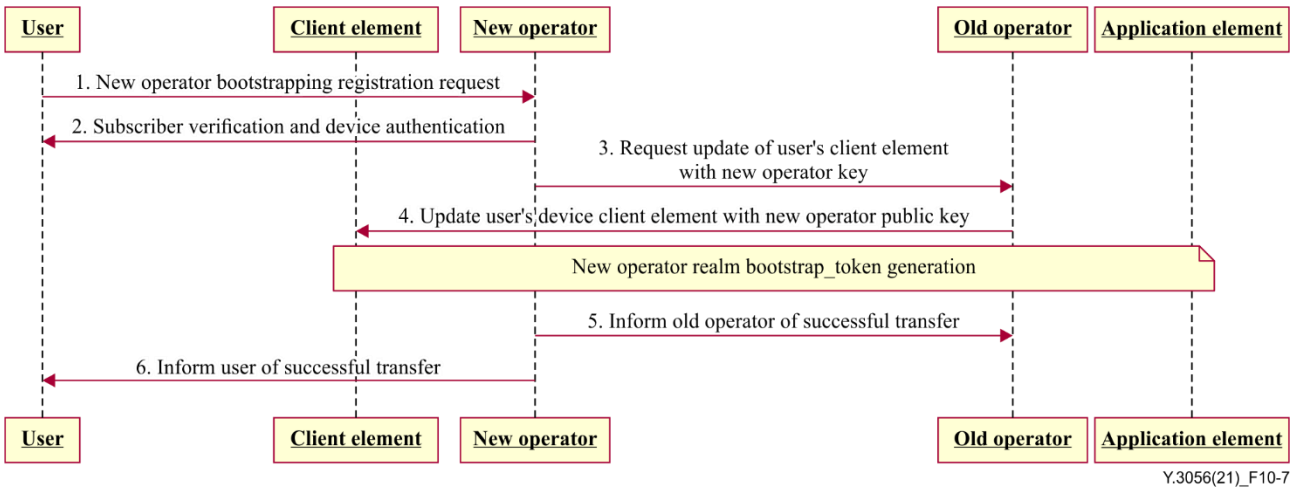

Figure 10-7 – Change of network operator (asymmetric keys)

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

Step 2: The new network operator undertakes the subscriber verification and the trusted device authentication.

Step 3: The new network operator requests the old network operator to update the client element of the user's trusted device by replacing the old network operator's public key(s) with that of the new network operator using its own private keys.

Step 4: Upon success, the old network operator provides the public key of the client element of the trusted device to the new operator.

At this stage, the trusted device may invoke the bootstrap_token generation flow for generating the bootstrap_token from the new network operator realm.

Step 5: Upon successful generation of bootstrap_token in the new realm, the new network operator informs the old network operator of the successful change of bootstrapping realm.

Step 6: Upon successful generation of bootstrap_token in the new realm, the new network operator informs the user of the successful change of bootstrapping ream.

11 Security considerations

This Recommendation proposes the existence of multiple simultaneous bootstrapping frameworks, which may be made up of IP and non-IP network realms. Thus, the security considerations are based on clauses 7 and 8 of [b-ITU-T Y.2701]. Additional information can be found in [b-ITU-T Y-Sup.19].

In order to mitigate security threats and potential attacks, issues of confidentiality, integrity, authenticity, non-repudiation, availability and traceability need to be addressed, and appropriate security and privacy protection schemes should be considered for the trusted devices, ASP applications and the interfaces between these and the network realms. Details are outside the scope of this Recommendation.

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.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-T Y.2701] Recommendation ITU-T Y.2701 (2007), *Security requirements for NGN release 1.*

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

[b-ITU-T Y-Sup.19] ITU-T Y.2200-series Supplement 19 (2012), *Supplement on the risk analysis service in next generation networks.*

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

[b-IETF RFC 6733] IETF RFC 6733 (2012), *Diameter Base Protocol.*

[b-IETF RFC 7155] IETF RFC 7155 (2014), *Diameter Network Access Server Application.*

[b-IETF RFC 7616] IETF RFC 7616 (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).*

SERIES OF ITU-T RECOMMENDATIONS

Series A	Organization of the work of ITU-T
Series D	Tariff and accounting principles and international telecommunication/ICT economic and policy issues
Series E	Overall network operation, telephone service, service operation and human factors
Series F	Non-telephone telecommunication services
Series G	Transmission systems and media, digital systems and networks
Series H	Audiovisual and multimedia systems
Series I	Integrated services digital network
Series J	Cable networks and transmission of television, sound programme and other multimedia signals
Series K	Protection against interference
Series L	Environment and ICTs, climate change, e-waste, energy efficiency; construction, installation and protection of cables and other elements of outside plant
Series M	Telecommunication management, including TMN and network maintenance
Series N	Maintenance: international sound programme and television transmission circuits
Series O	Specifications of measuring equipment
Series P	Telephone transmission quality, telephone installations, local line networks
Series Q	Switching and signalling, and associated measurements and tests
Series R	Telegraph transmission
Series S	Telegraph services terminal equipment
Series T	Terminals for telematic services
Series U	Telegraph switching
Series V	Data communication over the telephone network
Series X	Data networks, open system communications and security
Series Y	**Global information infrastructure, Internet protocol aspects, next-generation networks, Internet of Things and smart cities**
Series Z	Languages and general software aspects for telecommunication systems

Printed in Switzerland
Geneva, 2021