International Telecommunication Union

ITU-T

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

X.1193

(10/2011)

SERIES X: DATA NETWORKS, OPEN SYSTEM COMMUNICATIONS AND SECURITY

Secure applications and services – IPTV security

Key management framework for secure Internet protocol television (IPTV) services

Recommendation ITU-T X.1193

International Telecommunication Union

ITU-T X-SERIES RECOMMENDATIONS

DATA NETWORKS, OPEN SYSTEM COMMUNICATIONS AND SECURITY

PUBLIC DATA NETWORKS	X.1–X.199
OPEN SYSTEMS INTERCONNECTION	X.200–X.299
INTERWORKING BETWEEN NETWORKS	X.300–X.399
MESSAGE HANDLING SYSTEMS	X.400–X.499
DIRECTORY	X.500–X.599
OSI NETWORKING AND SYSTEM ASPECTS	X.600–X.699
OSI MANAGEMENT	X.700–X.799
SECURITY	X.800–X.849
OSI APPLICATIONS	X.850–X.899
OPEN DISTRIBUTED PROCESSING	X.900–X.999
INFORMATION AND NETWORK SECURITY	
General security aspects	X.1000–X.1029
Network security	X.1030–X.1049
Security management	X.1050–X.1069
Telebiometrics	X.1080–X.1099
SECURE APPLICATIONS AND SERVICES	
Multicast security	X.1100–X.1109
Home network security	X.1110–X.1119
Mobile security	X.1120–X.1139
Web security	X.1140–X.1149
Security protocols	X.1150–X.1159
Peer-to-peer security	X.1160–X.1169
Networked ID security	X.1170–X.1179
IPTV security	**X.1180–X.1199**
CYBERSPACE SECURITY	
Cybersecurity	X.1200–X.1229
Countering spam	X.1230–X.1249
Identity management	X.1250–X.1279
SECURE APPLICATIONS AND SERVICES	
Emergency communications	X.1300–X.1309
Ubiquitous sensor network security	X.1310–X.1339
CYBERSECURITY INFORMATION EXCHANGE	
Overview of cybersecurity	X.1500–X.1519
Vulnerability/state exchange	X.1520–X.1539
Event/incident/heuristics exchange	X.1540–X.1549
Exchange of policies	X.1550–X.1559
Heuristics and information request	X.1560–X.1569
Identification and discovery	X.1570–X.1579
Assured exchange	X.1580–X.1589

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

Recommendation ITU-T X.1193

Key management framework for secure Internet protocol television (IPTV) services

Summary

Recommendation ITU-T X.1193 describes the requirements and architecture for key management, including key hierarchy, for unicast and multicast IPTV services in the IPTV context. It also specifies key management for downloadable service and content protection (SCP), if deployed. This Recommendation does not include any other key management architecture or mechanisms described in Recommendation ITU-T X.1191.

History

Edition	Recommendation	Approval	Study Group
1.0	ITU-T X.1193	2011-10-14	17

Keywords

Authentication, authorization, content protection, encryption, rights management, scrambling, security, service protection.

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, 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 TSB patent database at http://www.itu.int/ITU-T/ipr/.

Table of Contents

Recommendation ITU-T X.1193

Key management framework for secure Internet protocol television (IPTV) services

1 Scope

This Recommendation describes the requirements and architecture for key management, including key hierarchy, for unicast and multicast IPTV services in the IPTV context. It also specifies key management for downloadable SCP, if deployed. This Recommendation does not include any other key management architecture or mechanisms described in [ITU-T X.1191].

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.509] Recommendation ITU-T X.509 (2005) | ISO/IEC 9594-8:2005, *Information technology – Open Systems Interconnection – The Directory: Public-key and attribute certificate frameworks*.

[ITU-T X.800] Recommendation ITU-T X.800 (1991) | ISO/IEC 7498-2:1989, *Security architecture for Open Systems Interconnection for CCITT applications*.

[ITU-T X.805] Recommendation ITU-T X.805 (2003) | ISO/IEC 18028-2:2006, *Security architecture for systems providing end-to-end communications*.

[ITU-T X.1191] Recommendation ITU-T X.1191 (2009), *Functional requirements and architecture for IPTV security aspects*.

3 Definitions

3.1 Terms defined elsewhere

This Recommendation uses the following terms defined elsewhere:

3.1.1 access control [ITU-T X.800]: The prevention of unauthorized use of a resource, including the prevention of use of a resource in an unauthorized manner.

3.1.2 acquisition [ITU-T X.1191]: The process of obtaining content by the end user.

NOTE – For content with accessibility features, this means that the content will be available in a form that can be used by the end user.

3.1.3 authentication [ITU-T X.800]: See data origin authentication and peer-entity authentication.

3.1.4 authorization [ITU-T X.800]: The granting of rights, which includes the granting of access based on access rights.

3.1.5 confidentiality [ITU-T X.800]: The property that information is not made available or disclosed to unauthorized individuals, entities, or processes.

3.1.6 content protection [ITU-T X.1191]: Ensuring that an end user can only use the content that they have already acquired, in accordance with the rights that they have been granted by the rights holder. Content protection includes protecting contents from illegal copying and distribution, interception, tampering, unauthorized use, etc.

3.1.7 data origin authentication [ITU-T X.800]: The corroboration that the source of data received is as claimed.

3.1.8 denial of service [ITU-T X.800]: The prevention of authorized access to resources or the delaying of time-critical operations.

3.1.9 digital signature [ITU-T X.800]: Data appended to, or a cryptographic transformation (see cryptography) of a data unit, that allows a recipient of the data unit to prove the source and integrity of the data unit and protect against forgery, e.g., by the recipient.

3.1.10 forward secrecy [b-NIST SP 800-120]: A compromise of long-term private or pre-shared secret keys does not enable an adversary to compute the *MK* generated in previous EAP executions.

3.1.11 group controller and key server [b-IETF RFC 3740]: Entity and functions that are responsible for issuing and managing cryptographic keys used by a multicast group.

3.1.12 implicit key authentication [b-NIST SP 800-120]: A property of key establishment protocols that provides assurance to one protocol participant that the other protocol participant is the only other party that could possibly be in possession of the correct established key.

3.1.13 initiator [b-IETF RFC 3830]: The initiator of the key management protocol, not necessarily the initiator of the communication.

3.1.14 key [ITU-T X.800]: A sequence of symbols that controls the operations of encipherment and decipherment.

3.1.15 key confirmation [b-NIST SP 800-120]: A procedure to provide assurance to one party (the key confirmation recipient) that another party (the key confirmation provider) actually derived the correct secret keying material as a result of a key establishment.

3.1.16 key derivation [b-NIST SP 800-120]: The process that derives keys from another key or from a secret output value, called shared secret, obtained through a key establishment procedure.

3.1.17 key establishment [b-NIST SP 800-120]: A procedure, conducted by two or more participants, which culminates in the derivation of keying material by all participants (see keying material). Key establishment can be based on pre-shared keys or on public key-based schemes.

3.1.18 key hierarchy [b-NIST SP 800-120]: A tree structure that represents the relationship of different keys. In a key hierarchy, a node represents a key used to derive the keys represented by the descendent nodes. A key can only have one precedent, but may have multiple descendent nodes.

3.1.19 key management [ITU-T X.800]: The generation, storage, distribution, deletion, archiving and application of keys in accordance with a security policy.

3.1.20 keying material [b-NIST SP 800-120]: The output of a key derivation function, a segment of which, with the required length, can be used as a cryptographic key.

3.1.21 key transport [b-NIST SP 800-120]: A procedure to deliver a key from one entity to another entity in a protected manner. In this Recommendation, key transport refers to key delivery from the EAP server to another entity, for example, to an authenticator.

3.1.22 masquerade [ITU-T X.800]: The pretence by an entity to be a different entity.

3.1.23 peer-entity authentication [ITU-T X.800]: The corroboration that a peer entity in an association is the one claimed.

3.1.24 policy server [b-IETF RFC 3740]: Entity and functions that are responsible for creating and managing security policies specific to a multicast group.

3.1.25 privacy [ITU-T X.800]: The right of individuals to control or influence what information related to them may be collected and stored and by whom and to whom that information may be disclosed.

3.1.26 repudiation [ITU-T X.800]: Denial by one of the entities involved in a communication of having participated in all or part of the communication.

3.1.27 responder [b-IETF RFC3830]: The responder in the key management protocol.

3.1.28 rights [ITU-T X.1191]: One or more legal or business entitlements to use or employ content, e.g., to view, record, redistribute content.

3.1.29 rights expression [ITU-T X.1191]: The syntactic embodiment of rights in a concrete, formal form.

3.1.30 security label [ITU-T X.800]: The marking bound to a resource (which may be a data unit) that names or designates the security attributes of that resource.

NOTE – The marking and/or binding may be explicit or implicit.

3.1.31 security policy [ITU-T X.800]: The set of criteria for the provision of security services.

3.1.32 scrambling algorithm [ITU-T X.1191]: An algorithm used in a scrambling (encryption) or descrambling (decryption) process.

3.1.33 security association [b-IETF RFC 3740]: A set of policy and cryptographic keys that provide security service to network traffic that corresponds to that policy.

3.1.34 service and content protection [ITU-T X.1191]: A combination of service protection and content protection, or a system or implementation thereof.

3.1.35 service protection [ITU-T X.1191]: Ensuring that an end user can only acquire a service and, by extension, the content contained therein, that they are entitled to receive. Service protection includes protecting service from unauthorized access as IPTV contents traverse through the IPTV service connections.

3.1.36 terminal device protection [ITU-T X.1191]: Ensuring that a terminal device employed by an end user in the reception of a service can reliably and securely use content while enforcing the rights of use granted for that content, and while physically and electronically protecting the integrity of the terminal device, and the confidentiality of the content and critical security parameters (e.g., saved keys) that are not otherwise protected.

3.1.37 threat [ITU-T X.800]: A potential violation of security**.**

3.2 Terms defined in this Recommendation

This document defines the following terms:

3.2.1 accessibility: The property of being accessible and useable upon demand by an authorized entity.

3.2.2 authentication proxy: A server, residing on the IPTV service provider's side, which is responsible for managing a secure downloadable service and content protection (SCP message exchange with the security module (SM)) on the IPTV terminal's side. It authenticates a security module (SM) and acts as proxy for downloadable SCP services. The authentication proxy plays the role of sub-function of the right and key management function.

3.2.3 authentication server: A server that performs the secure downloadable service and content protection (SCP function with the security module (SM)) on the terminal side in the IPTV context.

3.2.4 conditional access: The function served by a conditional access system; often used as an abbreviation for conditional access system.

3.2.5 conditional access system: A component of a service and content protection system, the purpose of which is to prevent unauthorized (unentitled) access to a service or to content.

3.2.6 content key: A key used to protect the IPTV data stream(s).

3.2.7 digital rights management: A synonym for service and content protection or content protection, depending upon the context of use.

NOTE – In this Recommendation, the terms "service protection" and "content protection" are used instead of digital rights management.

3.2.8 entitlements: Rights granted to a subscriber by a contract with an IPTV service provider.

3.2.9 integrity: The property that data has not been altered or destroyed in an unauthorized manner.

3.2.10 master key (MK): A key derived from authentication and key management between SM and AP or authentication server for the secure downloadable service and content protection (SCP).

3.2.11 multicast IPTV service: An IPTV service provided to a sub-group of IPTV users so that content stream is delivered simultaneously to all terminal devices (TDs) belonging to the sub-group when transmitted by the service and content protection (SCP) function.

3.2.12 multicast-based downloadable SCP scheme: A downloadable service and content protection (SCP) scheme that can download the SCP operating code to multiple terminal devices (TDs) belonging simultaneously to a sub-local area group (SLAG).

3.2.13 pairwise master key (PMK): A key derived from the master key shared by the security module (SM) and the authentication proxy (AP) for the secure downloadable service and content protection (SCP).

3.2.14 pre-shared key (PSK): A key shared in advance by the security module (SM) and the authentication proxy (AP) or authentication server for the secure downloadable service and content protection (SCP).

3.2.15 rekey: A process to change the service and content protection (SCP) key so as to limit the amount of data encrypted with the same SCP key.

3.2.16 SCP client code: The software code that performs the service and content protection (SCP) client function. It can be downloaded to an IPTV terminal device (TD) if the downloadable SCP is deployed.

3.2.17 security module: A hardware or software module with tamper resistance to perform the downloadable service and content protection (SCP) protocol, generate and store keys, and store SCP client data and code. The security module acts as a sub-function of the SCP client on the IPTV terminal.

3.2.18 service: A set of functionality enabled by a provider for end users; for example, providing IP connectivity with managed quality of service, providing an IPTV service, providing a content-on-demand service, etc.

3.2.19 sub-local area group (SLAG): A multicast group consisting of many terminal devices (TDs) that reside in the same physical area or belong to the same logical area.

3.2.20 unicast IPTV service: A service provided to a single terminal device (TD) so that content stream is delivered to a single TD in the IPTV context when transmitted by the service and content protection (SCP) function.

3.2.21 unicast-based downloadable SCP: A downloadable service and content protection (SCP) scheme that can download SCP operating code to a single TD.

4 Abbreviations and acronyms

This Recommendation uses the following abbreviations and acronyms:

AAA Authentication, Authorization, and Accounting
ACS Auto-Configuration Server
AES Advanced Encryption Standard
AP Authentication Proxy
CAS Conditional Access System
CDN Content Delivery Network
CEF Content Encryption Function
CGK CK Generation Key
CK Content encrypting Key
CM Counter Mode
CP Content Protection; Content Provider
CPE Customer Premises Equipment
CRL Certificate Revocation List
CS Crypto Session
CSB Crypto Session Bundle
DH Diffie-Hellman
DoS Denial of Service
DRM Digital Rights Management
EAP Extensible Authentication Protocol
ECM Entitlement Control Message
EMM Entitlement Management Message
GCKS Group Controller and Key Server
HDD Hard Disk Drive
ID Identifier
IPTV Internet Protocol Television
ISP Internet Service Provider
KMF Key Management Function
MAC Message Authentication Code
MCF Media Control Function
MDF Media Distribution Function
MIC Message Integrity Code
MIKEY Multimedia Internet Keying
MK Master Key
PC Protection Client image
PCL Protection Client Loader

PK	Public Key
PKI	Public Key Infrastructure
PMK	Pairwise Master Key
PSK	Pre-Shared Key
QoS	Quality of Service
RKMF	Rights and Key Management Function
RTP	Real-Time Transport Protocol
RTSP	Real-Time Streaming Protocol
SA	Security Association
SCK	SCP Key
SDP	Session description protocol
SCF	SCP Client Function
SCP	Service and Content Protection
SIP	Session Initiation Protocol
SLAG	Sub Local Area Group
SM	Security Module
SP	Service Protection; Service Provider
SRTP	Secure RTP
TA	Trust Authority
TD	Terminal Device
TEK	Traffic Encrypting Key
TGK	TEK Generation Key
URK	User Root Key
VoD	Video on Demand
WM	Watermark(ing)

5 Conventions

In this Recommendation:

The keywords "**is required to**" indicate a requirement which must be strictly followed and from which no deviation is permitted if conformance to this document is to be claimed.

The keywords "**is recommended**" indicate a requirement which is recommended but which is not absolutely required. Thus this requirement need not be present to claim conformance.

The keywords "**is prohibited from**" indicate a requirement which must be strictly followed and from which no deviation is permitted if conformance to this document is to be claimed.

The keywords "**can optionally**" indicate an optional requirement which is permissible, without implying any sense of being recommended. This term is not intended to imply that the vendor's implementation must provide the option and the feature can be optionally enabled by the network operator/service provider. Rather, it means the vendor may optionally provide the feature and still claim conformance with the specification.

In the body of this document and its annexes, the words *shall*, *shall not*, *should*, and *may* sometimes appear, in which case they are to be interpreted, respectively, as *is required to*, *is prohibited from*, *is recommended*, and *can optionally*. The appearance of such phrases or keywords in an appendix or in material explicitly marked as *informative* are to be interpreted as having no normative intent.

6 Introduction

6.1 IPTV general architecture and content protection architecture

IPTV services are classified into two typical types: a unicast-based service and a multicast-based service. According to the number of senders, a multicast-based service is classified into two types: a 1-to-N service and an M-to-N service. In a 1-to-N multicast-based service, only one sender can transmit a message to multiple TDs, while in an M-to-N multicast-based service, many senders can transmit a message to many TDs. Key management is the process of distributing the keys needed to provide secure IPTV services. To provide various cryptographic keys, key management together with authentication play a critical role in providing secure IPTV services.

This Recommendation describes the set of requirements and architecture, including key hierarchy, for unicast and multicast IPTV services in the IPTV context. It also includes key management for the downloadable SCP.

6.2 IPTV general architecture and content protection architecture

The general security architecture for IPTV is illustrated in Figure 6-1 below. This general architecture is divided into two primary areas: one considered in-scope for the purpose of considering interoperability based on this document, and another considered out-of-scope. The first area encompasses the end-user, network provider, and service provider domains, whereas the second area covers the content provider domain.

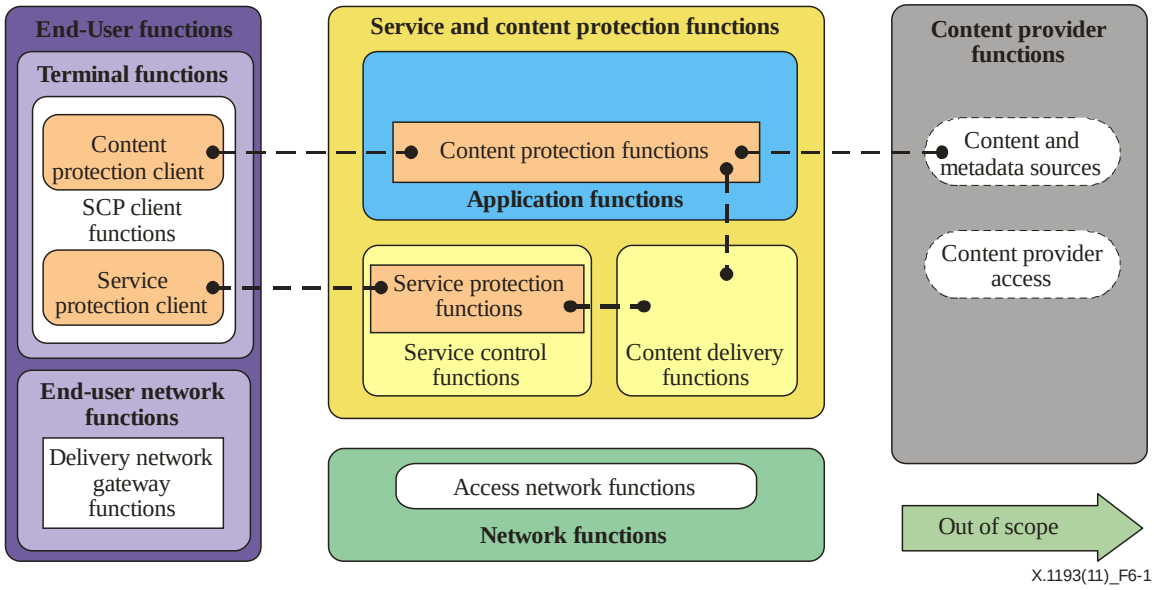
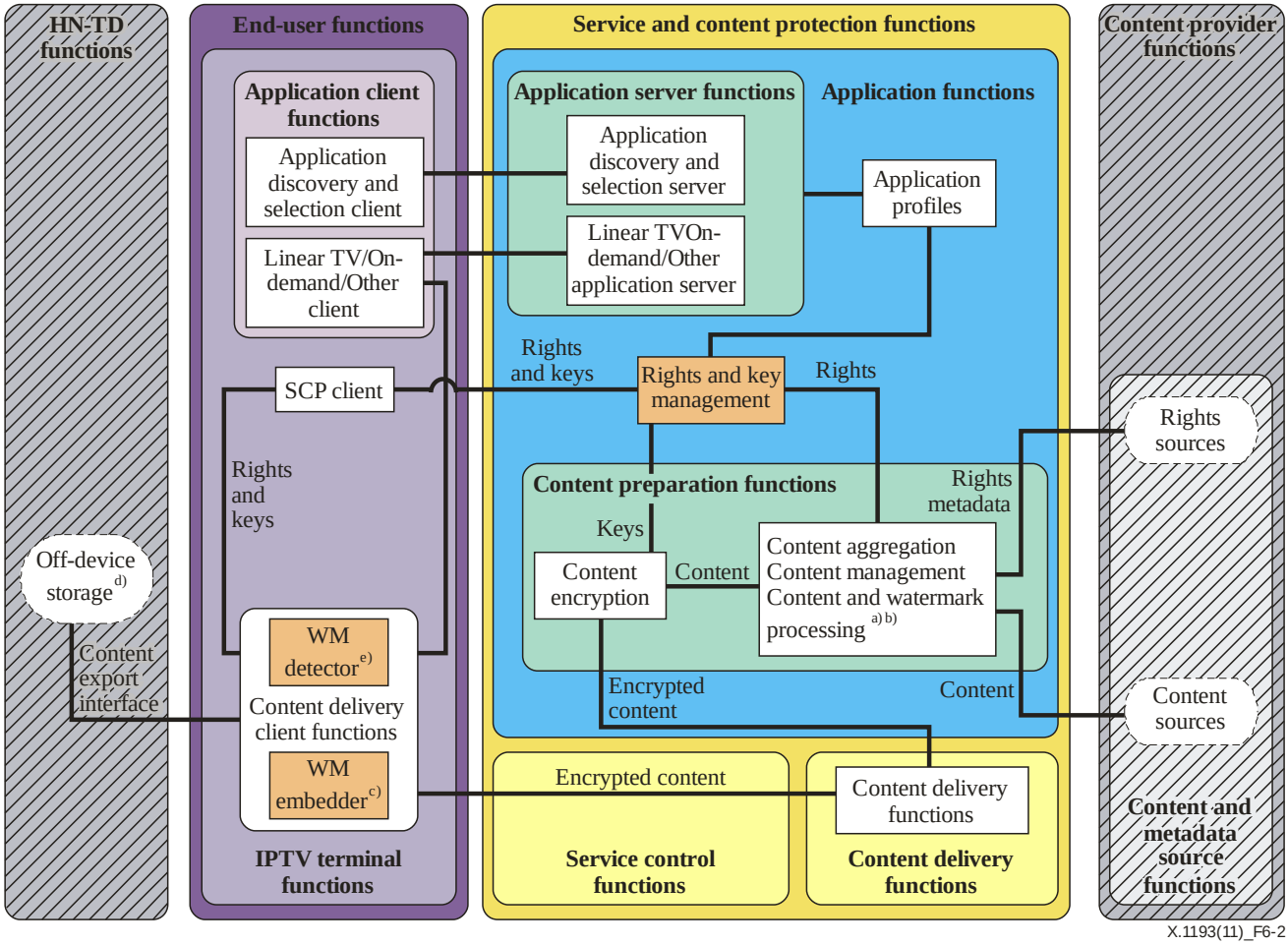


Figure 6-1 – IPTV general security architecture

The content protection architecture for IPTV is depicted in Figure 6-2 below. The primary function of the content protection architecture is to delineate the flow and processing of information pertaining to content usage rights, and information required to manage and facilitate such rights.

Ultimately, the rights of content-use originate with the content provider(s); but it should be noted that such rights may be modified (e.g., restricted, or perhaps even expanded) by service provider(s), according to their agreements with content providers and their operational and business policies.

From an operational and typical legal perspective, an end user's access and use of content involve the service provider, and not the content provider.



a) Optional watermark metadata generation to facilitate downstream watermark embedding
b) Optional watermark embedder to individuate content to networks, servers, and unicast deliveries
c) Optional watermark embedder to individuate multi-cast content instances
d) Optional off-device storage: a storage device inside HN-TD
e) Optional detector for copy protection watermarks

NOTE – Objects in the hashed grey color are out scope of IPTV security architecture

Figure 6-2 – IPTV content protection architecture

6.3 Reference model of group key management

The IPTV terminal can be classified into several groups: a group for the multicast IPTV service, a group for the secure downloadable SCP service, etc. This clause only describes the conceptual reference model for group key management. There are four entities: a policy server, a group controller key server (GCKS), a sender, and a receiver. GCKS is an entity responsible for issuing and managing the cryptographic keys used for a specific group and the policy server is an entity responsible for creating and managing security policies for managing groups.

The functions of the GCKS, the policy server, and the sender could be part of the rights and key management function in Figure 6-2. The function of the receiver could be part of the SCP client function in Figure 6-2.

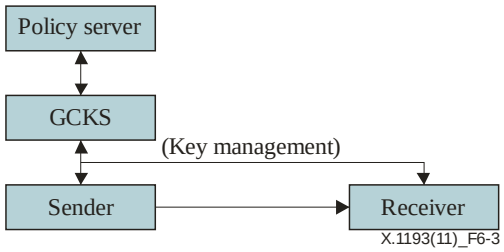


Figure 6-3 – Reference model for group key management

6.4 Grouping of TDs

Any TD may be classified into a sub-local area group. SLAG (Sub-Local Area Group) is a group consisting of many TDs that reside in the same physical area, or belong to a logical area. The communication between the IPTV service provider and all TDs in SLAG is assumed to be multicast-based communication.

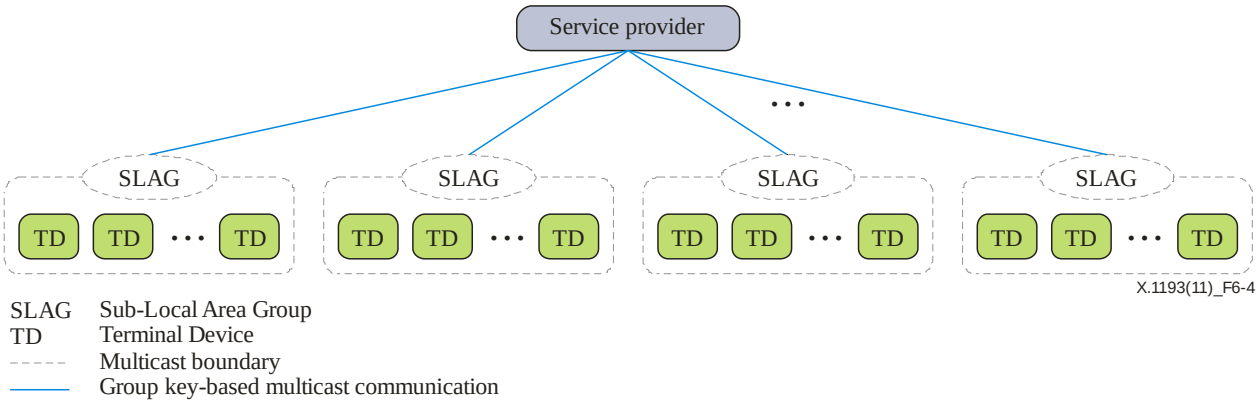


Figure 6-4 – Configuration of SLAG

7 Requirements for the key management scheme

7.1 Requirements for key management described in [ITU-T X.1191]

Clause 3.1.15 of [ITU-T X.1191] defines key management as the generation, storage, distribution, deletion, archiving, and application of keys, in accordance with a security policy. Clause 6.3 of [ITU-T X.1191], *Service security requirements*, describes the following key management requirements:

- The IPTV architecture is required to have the ability to use standard key management systems (e.g., MIKEY, EMM/ECM) as required for interoperability.
- If the IPTV architecture employs a key management system, then it is required to be designed for scalability, reliability, and interoperability.
- If the IPTV architecture employs a key management system, considering a hierarchical key management scheme is recommended to support scalability.
- If the IPTV architecture employs a key management system that uses a group key management protocol, considering a hierarchical key management and key management algorithm alternative is recommended to support scalability.
- If the IPTV architecture employs a key management system that uses short term keys, provisioning the media path such that NAT traversal and bandwidth constraints do not limit the key exchange is recommended.

- If downloadable SCP is deployed, the IPTV architecture is required to perform integrity protection and data origin authentication for the downloaded SCP system.

Clause 6.5 of [ITU-T X.1191], *Terminal security requirements*, lists several requirements on key management, including the following:

- If downloadable SCP is deployed, the IPTV architecture is required to support the secure download and installation of the SCP operating code to TDs.
- The IPTV architecture is required to support a secure means of performing security-critical processes in the TD such as key management and media serialization to abort playback of content in the event of a security related malfunction, detection of tampering, or other indication of misuse.

7.2 General requirements for key management

Key management satisfies the following requirements:

- The key management architecture is required to support identification and mutual entity authentication between the SCP client function on the end-user side and the rights and key management function on the network side.
- The key management architecture is required to provide keying materials for both unicast service and multicast IPTV service in a secure manner, i.e., confidentially, and in an integrity-protected manner.
- If the key management architecture deploys a public key certificate, the public key certificate and the certificate revocation list (CRL) are required to comply with those specified in [ITU-T X.509].
- The key management architecture is required to identify and authenticate the SCP client function belonging to a specific sub-local area group.
- The key management architecture is required to provide keying materials for establishing a secure session between the rights and key management function and the SCP client function to protect the IPTV service stream if short-term keying material needs to be delivered.
- The key management architecture is recommended to rekey the SCP key periodically or in a specific manner.
- If a multicast IPTV service is provided, it is recommended that the key management architecture allow any member to leave the specific group or join it at any time, in a secure manner.
- The IPTV architecture can optionally have the capability to deploy the multicast-based downloadable SCP scheme described in clause 8.3.2.4.
- If a multicast-based downloadable SCP scheme is deployed, it is recommended that the IPTV architecture have the capability to classify each TD into SLAG.
- If a multicast-based downloadable SCP scheme is deployed, it is recommended that SLAG use the same unique group key(s).
- If a multicast-based downloadable SCP scheme is deployed, it is recommended that key management generate keying material to encrypt the SCP operating code with the group encryption key to SLAG.
- If a multicast-based downloadable SCP scheme is deployed, it is recommended that the key management architecture send the SCP operating code in multicast mode, encrypted with the group encrypting key, simultaneously to all TDs belonging to SLAG.
- If key establishment is used for key management, it is recommended that key establishment have mutual implicit key authentication, i.e., the capability to prove that the established keying material is known only to the relevant entities.

- If key establishment is used for key management, it is recommended that key establishment have key freshness, i.e., the established key is (pseudo-) random, and the probability of repeating a previously established key is low.
- If key establishment is used for key management, it is recommended that key establishment have key control, i.e., both parties should contribute data for key computation.
- If key establishment is used for key management, it is recommended that key establishment have key confirmation, i.e., both entities should obtain assurance that they have computed the derived key correctly. Key confirmation is commonly achieved by using one of the derived keys to generate a message authentication code. Mutual key confirmation, combined with mutual implicit key authentication, provides mutual explicit key authentication.
- If key establishment is used for key management, key establishment (for public key-based key establishment schemes) can optionally have forward secrecy (FS), i.e., a compromise of long-term private or pre-shared secret keys which does not allow an adversary to compute short-term keys generated in previous key establishment executions.

7.3 Functional requirements for key management

7.3.1 Functional key management requirements for unicast IPTV service

- The key management architecture for unicast service is recommended to have key hierarchy with at least two keys; content key (CK) and SCP key (SCK).
- The key management architecture for unicast IPTV service can optionally update the SCP key periodically, or in a specific manner, if a hierarchical key management scheme is deployed.
- The key management architecture for unicast IPTV service is required to update the content key periodically or in a specific manner, if hierarchical key management is deployed.
- The key management architecture for unicast IPTV is recommended to derive the SCP key for the unicast IPTV service using standardized key management schemes, i.e., MIKEY, which is described in Appendix I; EAPs, which are described in Appendix II, or the key management scheme for the downloadable SCP that is described in clause 8.3.

7.3.2 Functional key management requirements for multicast IPTV service

- The key management architecture for a multicast IPTV service is recommended to have a key hierarchy with at least three keys; content key (CK), SCP key (SCK), and intermediate key (IK).
- The key management architecture for a multicast IPTV service can optionally update the SCP key periodically or in a specific manner.
- The key management architecture for multicast IPTV (broadcast stream) service is required to update the content key periodically or in a specific manner.
- The key management architecture for multicast IPTV (broadcast stream) service is recommended to derive the SCP key for multicast service using the standardized key management schemes, i.e., MIKEY as described in Appendix I, EAPs as described in Appendix II, or the key management scheme for downloadable SCP as described in clause 8.3.

7.3.3 Functional key management requirements for the downloadable SCP scheme

Service protection and/or content protection software (or code) upgrade allows the adaptation of IPTV TD to the protection mechanisms used by a specific service provider. Since service protection and/or content protection are pivotal elements in the secure distribution of protected content, upgrading such part of the system needs to be protected against malicious parties trying to attack the upgrade (or download) procedure. Attackers' motivations may vary. Therefore, key management for a secure downloadable SCP scheme supporting software functions for the upgrade of the IPTV TD service and content protection functions should satisfy the following functional requirements:

- The key management architecture of a downloadable SCP scheme is required to enable IPTV TD to verify the authenticity (proof of origin and data integrity) and validity (e.g., credentials) of the downloaded SCP client operating code. Only once successful verification is completed, will IPTV TD be authorized to install and activate the downloaded SCP client operating code.
- The key management architecture of a downloadable SCP scheme is required to support mutual authentication between the SCP client and the rights and key management function.
- The key management architecture of a downloadable SCP scheme is required to provide a means of confirming the validity of the IPTV TD as a trusted environment for SCP software upgrade.
- The key management architecture of a downloadable SCP scheme is required to support, in a manner that guarantees the protection of confidentiality and integrity, the transport of all security sensitive data (especially keys, passwords, and credentials).
- The key management architecture of a downloadable SCP scheme is recommended to provide key control.
- The key management architecture of a downloadable SCP scheme is recommended to provide key confirmation.
- The key management architecture of a downloadable SCP scheme is recommended to support post-verification for all integrity-vulnerable information that has been exchanged before a transient integrity key is available.
- The key management architecture of a downloadable SCP scheme should generate the SCP key for downloading the SCP operating code.
- The key management architecture of a downloadable SCP scheme is recommended to generate an encryption key (EK) for decrypting a downloaded SCP operating code.
- The key management architecture of a downloadable SCP scheme is requested to generate a message integrity checking key for checking the integrity of a downloaded SCP operating code.
- The key management architecture of a secure SCP operating code download is recommended to generate the SCP key for the SCP function.
- The key management architecture of a secure SCP operating code download can optionally use the existing well-known standardized EAP methods for mutual authentication and key agreement.
- If a multicast-based downloadable SCP scheme is deployed, the IPTV TD belonging to a multicast group is recommended to share a group key for multicast-based downloadable SCP with a service provider.
- If a multicast-based downloadable SCP scheme is deployed, IPTV TD is recommended to have a group key consisting of an encryption key and a message authenticity/integrity key for decrypting the encrypted SCP operating code, and checking the validity of the authenticity and integrity of the SCP operating code, respectively.

- If a downloadable SCP scheme is deployed, IPTV TD is required to have the capability to decrypt the encrypted SCP operating code.
- If a downloadable SCP scheme is deployed, IPTV TD is required to authenticate the service provider to verify if it is entitled to perform an SCP client software upgrade.
- If a downloadable SCP scheme is deployed, and a handover of the IPTV TD to another service provider occurs, the IPTV TD is required to verify that only an entitled service provider may initiate an upgrade of the IPTV TD (the service provider must be authenticated by the IPTV TD and authorized to do this).
- If a downloadable SCP scheme is deployed, the service protection or content protection functions shall only be implemented in a trusted execution environment (hardware or virtual machine i.e., in software). Such an environment shall be tamper-proof, and contain a secure storage environment for security-sensitive information that shall not be revealed outside IPTV TD.
- If a downloadable SCP scheme is deployed, the service protection and/or content protection client subsystem in IPTV TD is required to offer the capability to authenticate the service protection and/or content protection client software that is in operation in IPTV TD as the authentic software provided by the service protection provider or content protection provider.
- It is required that the key management architecture of a downloadable SCP scheme for keys/certificates hard-linked with IPTV TD rely only on keys/certificates provided by a certification authority (trusted third party).
- If a downloadable SCP scheme is deployed, upgrading the SCP scheme software on request or on behalf of the owner of the IPTV TD shall be possible, independently of the currently active SCP scheme or services obtained.

7.3.4 Functional key management requirements for users possessing multiple IPTV terminals (N-screens)

- The key management architecture for users possessing multiple IPTV terminals, each of which with a different display resolution, is recommended to meet the requirements described in clause 7.3.2.
- Key management for users possessing multiple IPTV terminals is recommended to use the key management scheme described in clause 8.2 or group key management schemes [b-IETF RFC 4046].

8 Key management scheme for secure IPTV services

8.1 Key management scheme for unicast-based IPTV service

This clause describes the key management scheme for the unicast service when a hierarchical key management scheme is deployed.

8.1.1 Overview

A typical example of unicast service is the COD (content on demand) service. This clause describes the key management scheme for a unicast IPTV service.

The 3-layer key hierarchy should be used for the unicast IPTV service. The security of unicast IPTV service (i.e., CoD) faces different challenges compared to multicast services. There is no threat of conspiring users circumventing the security solution. Therefore, less complex security solutions can be employed for protecting the unicast service compared to the multicast service.

Key management for unicast service is based on three levels of key hierarchy. Figure 8-1 shows the 3-layer key hierarchy for unicast service protection; it consists of the bootstrapping layer, key stream layer, and traffic protection layer. The 3-layer key hierarchy is less complex because it does not require the mechanism of periodical update of decryption keys.

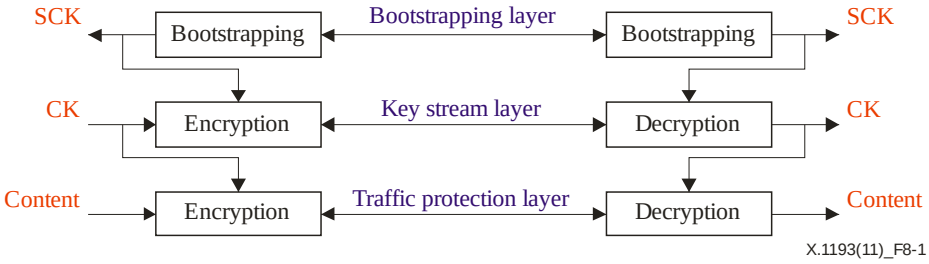


Figure 8-1 – 3-layer key hierarchy

8.1.2 Bootstrapping layer

This layer is used to establish the shared SCP key (SCK) between the SCF (SCP client function) in the end-user functions on the terminal side and the RKMF (rights and key management function) in the SCP functions on the server side. SK is specific to each user, and is used by the key stream layer to protect the delivery of the content key (CK).

8.1.3 Key stream layer

This layer is used to deliver the CK securely by encrypting the CK with the SCK. The CK is used by the traffic protection layer to encrypt and decrypt content.

8.1.4 Traffic protection layer

This layer is used to handle the encryption/decryption of content. The CK is used by the CEF (content encryption function) on the IPTV network side and by the CDCF (content delivery client functions) on the terminal side.

8.1.5 Procedure for key management for the multicast IPTV service

This clause describes the procedure of the key management protocol based on the 3-layer key hierarchy described in the previous clauses:

1) CEF transfers CK to RKMF.
2) CEF transfers the content encrypted with CK to CDF (content delivery function).
3) SCF on the terminal side, and RKMF on the IPTV network side, execute bootstrapping procedures to establish a shared SCK.
4) CK encrypted with SCK is transferred from KMF to SCF.
5) Content encrypted with CK is delivered from CDF on the network side to SCF on the terminal side.
6) SCF decrypts CK using SCK and decrypts the content using CK.

8.2 Key management scheme for multicast IPTV service

This clause describes the key management scheme for a multicast IPTV service to be used, if deployed.

8.2.1 Overview

A typical example of a multicast service is a real-time broadcast service. This clause describes the key management scheme for a real-time broadcast service.

The security of multicast IPTV faces challenges that differ from those faced by the security of services delivered over point-to-point services. In addition to the normal risk of eavesdropping, there is also a risk that valid subscribers might fail to respect the privacy and confidentiality of communications, and may even conspire to circumvent the security solution (for example, one of the subscribers may publish the decryption keys that enable non-subscribers to view the multicast content). Countering this threat requires that the encryption keys be frequently updated in a manner that may not be predicted by subscribers or attackers, while making efficient use of the transport network. This may require the use of at least a 4-level key hierarchy, as depicted in Figure 8-2. The frequency of update of the lowest level key (e.g., content keys in Figure 8-2) shall be chosen so that it will not be worthwhile extracting and publishing the key (i.e., on Internet) before the next update; e.g., once per second. Figure 8-2 depicts the 4-layer key hierarchy for the multicast service protection; it consists of the bootstrapping layer, key management layer, key stream layer, and traffic protection layer. The 4-layer key hierarchy can also be used for unicast service protection.

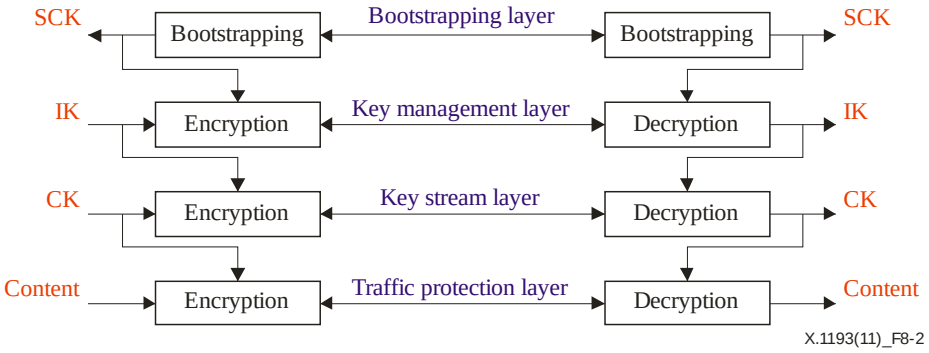


Figure 8-2 – 4-layer key hierarchy

8.2.2 Bootstrapping layer

This layer is used to establish the shared SCP key (SCK) between the SCP client function in the end-user functions on the client side and the rights and key management function in the SCP function on the server side. The SK is specific to each user, and it is used by the key management layer to protect the delivery of the intermediate key (IK).

8.2.3 Key management layer

This layer is used to securely deliver the IK between the rights and key management function and the SCP client function by encrypting the IK with SCK. The IK is used by the key stream layer to protect the delivery of CKs (TEKs).

8.2.4 Key stream layer

This layer is used to securely deliver CKs by encrypting CKs with IK. CKs are used by the traffic protection layer to encrypt and decrypt content. CKs are distributed along with the corresponding content.

8.2.5 Traffic protection layer

This layer is used to handle the encryption/decryption of content using CKs based on the content encryption function on the network side and content delivery client functions on the terminal side.

8.2.6 Procedure of key management for the multicast IPTV service

This clause describes the procedure of the key management protocol based on the 4-layer key hierarchy described in the previous clauses.

1) CEF encrypts the content and interacts with RKMF to acquire CKs encrypted with IK.

2) CEF transfers to CDF CKs encrypted with IK and content encrypted with CKs.

3) SCP on the terminal side and RKMF on the network side execute bootstrapping procedures to establish a shared SCK.

4) IK encrypted with URK is transferred from CDF on the network side to SCF on the terminal side; and SCF uses SCK to decrypt IK.

5) CKs encrypted with IK and content encrypted with CKs are delivered from CDF to SCF. For multicast service protection, CKs may be frequently updated, e.g., once per second. This will make it expensive or unfeasible for the user to disclose CKs that would enable non-subscriber users to view the content.

6) SCF decrypts CKs using IK and decrypts the content using CKs.

8.3 Key management scheme for a downloadable SCP scheme

8.3.1 Introduction

This clause describes two types of key management schemes: The interactive key management scheme, in clause 8.3.2; and the TA-based key management scheme, in clause 8.3.3.

8.3.2 Interactive key management for a downloadable SCP scheme

8.3.2.1 Overview of a downloadable SCP scheme

This clause presents an overview of a secure downloadable SCP scheme. The downloadable SCP scheme should satisfy the following:

- Identification of SM of IPTV TD to the IPTV service provider or other relevant TA, and vice versa.
- Entity authentication of IPTV TD to the IPTV service provider or other relevant TA, and vice versa.
- Procedure for key management (online or offline).
- Updating the SCP operating code to SM in IPTV TD.

To satisfy the requirements above, the authentication proxy on the IPTV service provider may send a broadcast message to the SM on the IPTV TD side to allow registration and attachment to the IPTV service provider. The procedure starts with some messages being exchanged for key management and authentication. This message exchange is done to prove that each SM is a legitimate entity that can act as proxy for an authorized user, and that the AP could act as legitimate proxy for updating key information. In addition, as a by-product, they could share a common root key. Therefore, messages should be exchanged for downloading the SM client operating code to the SM. At the end of the SCP code downloading procedure, the SM will report back the status of the downloading result to the AP.

8.3.2.2 Hierarchy for the real-time key management of the downloadable SCP scheme

This clause describes the key hierarchy for a downloadable SCP scheme. Any of the key management schemes described in Appendix I and Appendix II may be used for mutual authentication and key agreement.

Normally, the downloadable SCP scheme should be performed between the rights and key management function and an SCP client function.

Figure 8-3 illustrates the key hierarchy with three layers for a downloadable SCP scheme: a pre-shared key (PSK) layer, a master key (MK) layer, and a pairwise master key (PMK) layer. The pre-shared key is a root trust key for a secure download SCP service that is shared in advance between the authentication server and the SM. If the authentication server is part of the AP, the pre-shared key should be installed in both AP and SM. A master key is an interim key for deriving a pairwise master key; it is bound to the session between AP and SM. The pairwise master key consists of an

encrypting key, an MIC key, and a user root key. It is a collection of operational keys for downloading an SCP client securely. The third part of PMK is used as a user root key (URK) for the SCP function.

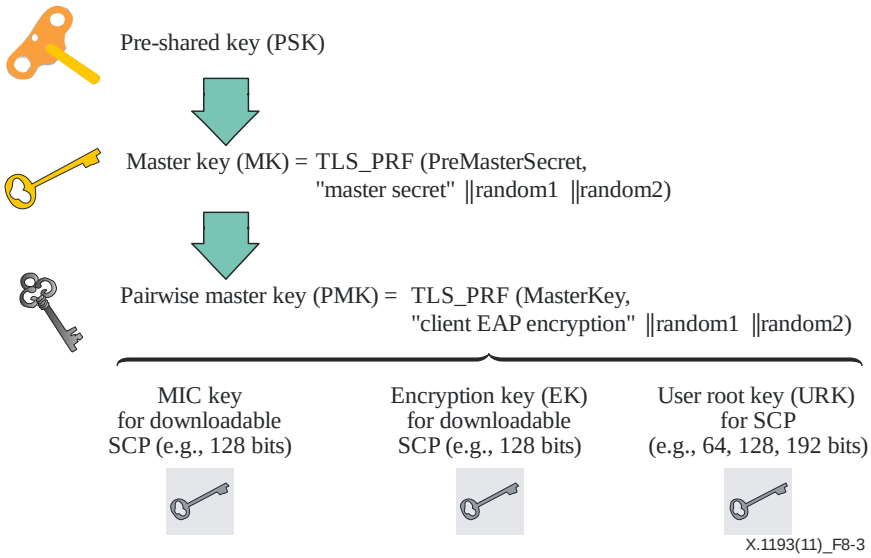


Figure 8-3 –Key hierarchy for a downloadable SCP scheme

8.3.2.3 Procedure for a unicast-based downloadable SCP scheme

This clause describes the procedure for a unicast-based downloadable SCP scheme. Figure 8-4 shows the typical procedures for the unicast-based downloadable SCP scheme. It consists of message exchanges for authentication and key exchange: a message for security module registration to the authentication proxy, and a message exchange for securely downloading an SCP operating code (or software) to the security module in the IPTV TD. The authentication server can be operated by a service provider, or it could serve as the trusted third-party trustee described in clause 8.3.3 (trust hierarchy model). When the authentication server is operated by the trusted third party described in clause 8.3.3, the secure channel between the authentication server and the authentication proxy should be established to distribute the PMK.

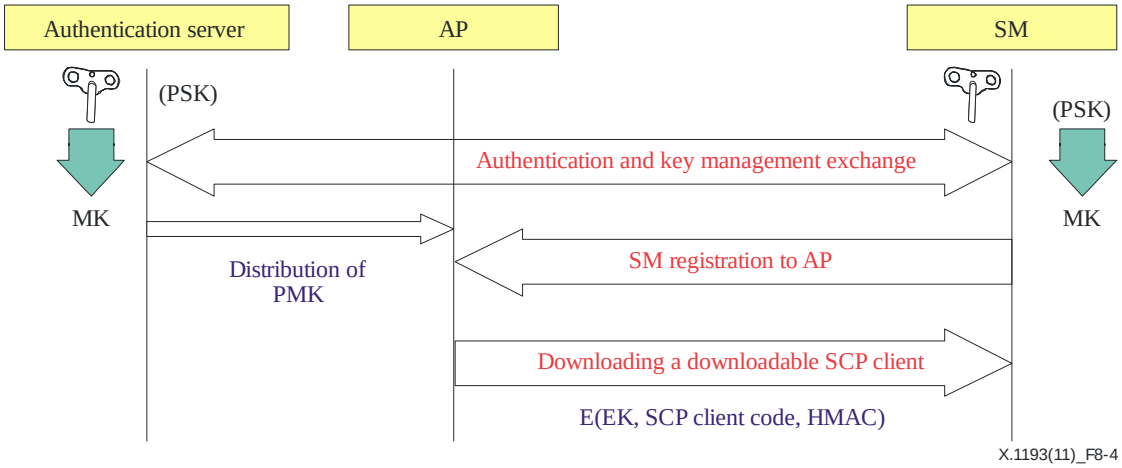


Figure 8-4 – Sequence for downloadable SCP

The first step is to authenticate the SM by the authentication server and the authentication server by the SM and derive a master key for a downloadable SCP. The PMK can be derived from a master key. The authentication and key exchange protocol can be distributed from the authentication server to the AP where there is a centralized AAA server. If there is no authentication server, the AP can act as authentication server; thus foregoing the need for distributing PMK. PMK consists of encrypting key (EK), MIC key and, optionally, an SCP key. EK is used for encrypting an SCP client, whereas the MIC key is used to verify the integrity of the SCP client.

8.3.2.4 Procedure of a multicast-based downloadable SCP scheme

This clause describes the procedure for a multicast-based downloadable SCP scheme as follows:

1) When a service provider needs to update the SCP operating code in a TD, it first generates the group key and broadcasts the SCP operating code encrypted with a group key that may be derived by using the procedure described in Appendix I. In addition, the service provider generates the licence that may contain the group key, together with the version number of the SCP operating code.
2) Following a request by a service provider to update the SCP operating code, the TD first stores a copy of the SCP operating code encrypted with the group key and initiates a licence distribution procedure with the service provider, which is used by the TD to obtain the group key as follows:
   i) The TD requests the service provider to initiate the authentication and key exchange procedure described in Appendix I or Appendix II to establish a secure session, which is then used to transmit the licence to the IPTV TD.
   ii) After mutual authentication of the TD and the service provider, the service provider sends the licence through the secure session established in step 1.
3) IPTV TD uses the encryption key in the licence to recover the SCP operating code from the encrypted SCP operating code. At the same time, the TD validates the message authenticity/integrity of the SCP operating code by using the authenticity/integrity key in the group key.
4) After checking the validity of the SCP operating code, the new SCP operating code is installed to replace the previous SCP operating code in the IPTV TD.

8.3.3 Trust authority-based key management for the downloadable SCP scheme

8.3.3.1 Introduction

SCP client software upgrade allows the adaptation of IPTV TDs to the protection mechanisms used by a specific service provider. Since service protection and content protection are pivotal elements in the secure distribution of protected content, upgrading such part of the system needs to be protected against malicious parties trying to attack the upgrade procedure.

This clause describes a key management scheme that enables "Secure upgrade (or download) of service protection and content protection" software. The solution may be generalized to other types of software.

The following points will be covered to specify a key management scheme for the "Secure upgrade of service protection and content protection":

- Stakeholder model.
- Trust hierarchy model.
- Use cases for the secure upgrade of SCP.
- Security architecture.

The solution for the "secure upgrade of service protection and content protection software" will have to cover different aspects to secure a trusted platform for firmware, service protection, content protection or other software. These aspects are:

- Secure loading of software by IPTV TD: this enables secure boot time and runtime loading of software to ensure a trusted environment for SP and/or CP or other software at all times.
- Secure upgrade of software on IPTV TD: this enables the secure upgrade of software to ensure a trusted environment for SP and/or CP or other software

The key management scheme that enables the secure upgrade of service protection and content protection software does not make any assumption on the type of stakeholder that is allowed to initiate such upgrades.

8.3.3.2 SCP client upgrade stakeholders

The following stakeholders in an IPTV TD upgrade process have been identified:

IPTV TD custodian: An IPTV TD custodian would act as a trust authority for IPTV TDs. It certifies all public device keys of compliant devices. With them, a service provider can verify that a connected device is authorized to download a new SCP implementation. An IPTV provider, or a chip (-set) provider, can obtain public-private key pairs from the custodian for use in the production of certified implementations of the IPTV upgrade system. The security of this process shall be under contractual agreements between the custodian and the IPTV TD provider.

Internet service provider custodian: An ISP custodian would act as a trust authority for ISP certification of all public device keys of trusted Internet service providers. With them, the IPTV TD can verify that an Internet service provider is authentic and authorized to download new IPTV TD software. Once connected to the Internet service provider, the IPTV TD obtains the Internet service provider's public key with a valid certificate from the custodian, which it can prove.

IPTV service provider custodian: An IPTV service provider custodian would act as trust authority for IPTV service providers and certify all public device keys of trusted IPTV service providers. With them, the IPTV TD can verify that an IPTV service provider is authentic and authorized to download a new IPTV TD implementation. Once connected to the IPTV service provider, the IPTV TD obtains the IPTV service provider's public key with a valid certificate from that custodian and has proof of such.

SCP custodian: An SCP custodian would act as trust authority for SCP providers and certify all public keys used during SCP software upgrades for the validation of the software upgrade signatures. The validation of the signed software download can be performed through the SP provider or CP provider to be sure that the software download has been performed properly. An SP provider or a CP provider will obtain the certified public keys needed for the validation of IPTV TD.

Internet service provider: As the actual provider of the IP connectivity service that a user wants to use, the Internet service provider can use a remote management interface with the connected IPTV TD that it is entitled to manage, in order to upgrade the IPTV TD software or security credentials on the IPTV TD side, so as to match the network side of the IP connectivity service offering.

IPTV service provider: As the actual provider of the IPTV service that a user wants to use, the IPTV service provider can use a remote management interface with the connected IPTV TD that it is entitled to manage, in order to upgrade the IPTV TD software or security credentials on the IPTV TD side, so as to match the network side of the IPTV service offering.

SP provider: As the provider of the end-to-end service protection system that the service provider uses, the SP provider could, for example, provide the actual SP software and upgrades to be loaded by the IPTV TD to interoperate with the IPTV service provider's service offering.

CP provider: As the provider of the end-to-end content protection system that the service provider uses, the CP provider could, for example, provide the actual CP software and upgrades to be loaded by the IPTV TD so as to interoperate with the IPTV service provider's service offering.

IPTV TD manufacturer: Fabricates the equipment that complies with the IPTV TD upgrade mechanism.

Chip manufacturer: Fabricates the chip-set that complies with the IPTV TD upgrade mechanism.

User: This is the consumer of the service offered by the service provider. Normally, the user is not aware of the upgrade mechanism; this should normally forego the need for any user interaction. The only situation where user consent might be needed is the service provider handover case, where the user needs to confirm that it is alright for a second service provider to be entitled to perform software upgrades to allow IPTV TD to be used with this second service provider.

Figure 8-5 below shows the case where the IPTV TD upgrade is required for service protection and content protection (SCP).

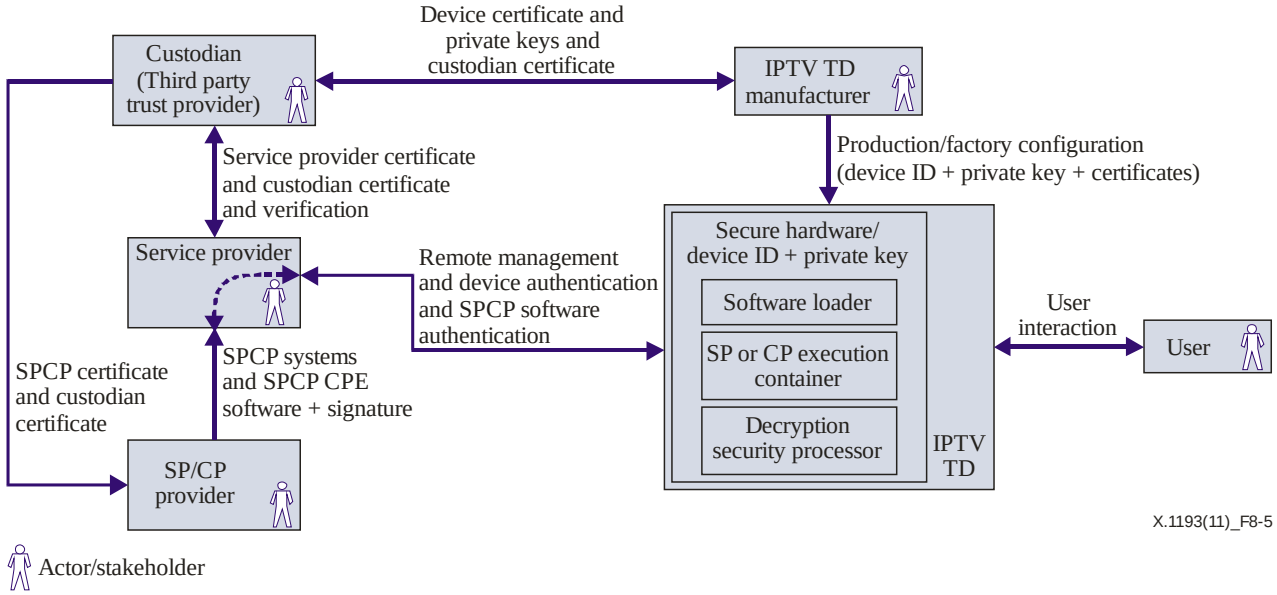


Figure 8-5 – SCP client upgrade stakeholders

Other concepts depicted in Figure 8-5:

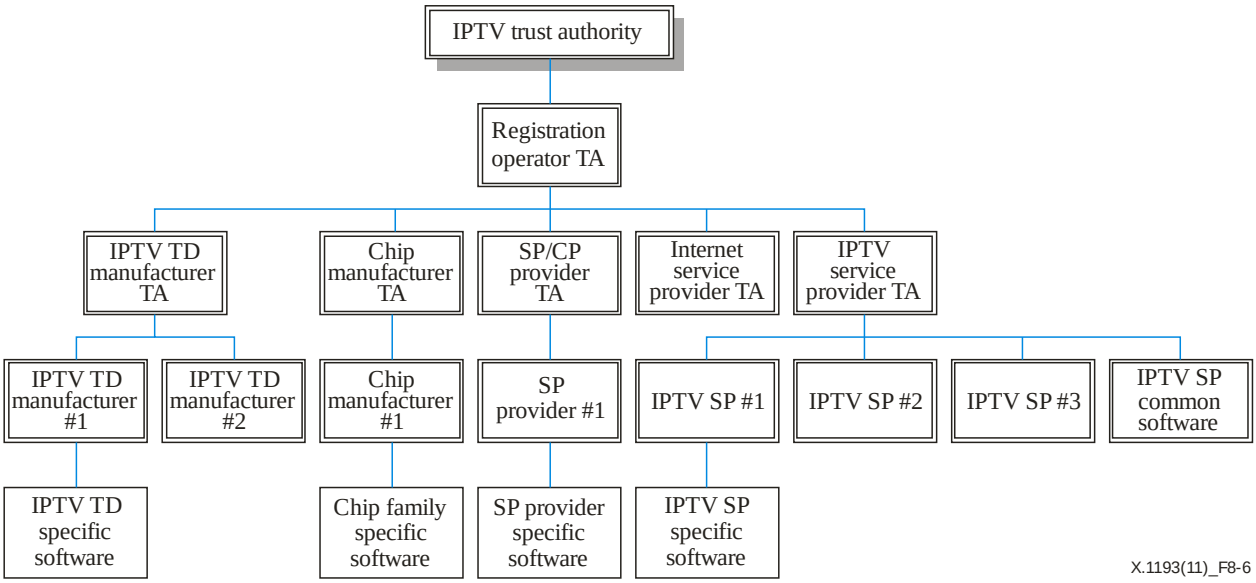
Secure hardware with device ID and private key: Secure hardware implementation of the software upgrade system in IPTV TD containing a tamper-proof, burned-in private key that is used in the process, whereby a service provider verifies that this is a trusted implementation.

Software loader: Built-in bootstrap loader that loads the SP or CP system software and additional decryption algorithms from the service provider at first start-up or during upgrade.

SP/CP execution container: SP/CP system software will be loaded into a secure hardware execution environment.

Security processor: On-chip security processor that is implemented in combination with hardware or software decryption algorithms.

8.3.3.3 Trust hierarchy model



TA	Trust authority
IPTV	Internet protocol television
SP provider	Service protection provider
CP provider	Content protection provider
TD	Terminal device
SP	Service provider

Figure 8-6 – IPTV TD secure upgrade trust hierarchy

8.3.3.3.1 IPTV trust authority

The IPTV trust authority is the root of trust that will ensure that only an authorized registration operator can add new trust authorities for:

- Internet service provider.
- IPTV service provider.
- SP provider or CP provider.
- IPTV TD manufacturer.
- Chip manufacturer.

It will also allow the registration operator to be changed at any time by revoking and replacing the certificate issued to the registration operator.

8.3.3.3.2 Registration operator trust authority

The role of the registration operator is the day-to-day addition and removal of Internet service providers, IPTV service providers, SP providers, CP providers, IPTV TD manufacturers, and chip-set manufacturer trust authorities by registering and deregistering them, and providing the necessary credentials to communicate securely with IPTV TD and subscribers.

While the number of IPTV service providers and IPTV TD providers may be small, the number and amount of change in common applications may eventually be very large; hence the importance of isolating the higher trust authorities from the day-to-day operational role.

8.3.3.3.3 ISP trust authority

The service may be dependent on agreement with certain ISPs to provide a minimum level of QoS that requires a secure distribution of configuration parameters to IPTV TD and possibly IPTV

service provider. This branch provides ISP and IPTV TD specific security credentials for this configuration.

8.3.3.3.4 IPTV service provider trust authority

Many aspects of end-to-end security are critically dependent on ensuring that the confidentiality and integrity of the IPTV service providers, IPTV TD software, and applications is ensured:

- at each and every service invocation;
- during run time of IPTV TD software;
- before and after any remote upgrade to such software.

This branch provides IPTV service provider-specific security credentials to ensure confidentiality and integrity.

8.3.3.3.5 SP/CP provider trust authority

Many aspects of end-to-end security are critically dependent on ensuring that the confidentiality and integrity of the SP or CP software are ensured:

- at each and every service invocation;
- during run time of the SPCP software,
- before and after any remote upgrade to such software.

This branch provides SP- and CP-specific security credentials to ensure confidentiality and integrity.

8.3.3.3.6 IPTV TD manufacturer trust authority

Many aspects of end-to-end security are critically dependent on ensuring that the integrity of the IPTV TD is ensured:

- at each and every boot;
- during run time of the IPTV TD operating system;
- during run time of IPTV TD-specific applications;
- before and after any remote upgrade to such software.

This branch provides IPTV TD-specific security credentials to ensure integrity, e.g., IPTV TD-specific boot, kernel, and root file system access keys.

8.3.3.3.7 Chip manufacturer trust authority

Many aspects of end-to-end security are critically dependent on ensuring the integrity of the processor chip or security chips. This branch provides chip manufacturer-specific security credentials to ensure integrity, e.g., SOC-specific identities and access keys.

8.3.3.3.8 IPTV service provider-specific trusted platform software and apps

There will be applications that are specific to a particular IPTV service provider.

8.3.3.3.9 IPTV service provider's common applications

As well as IPTV service provider-specific trusted platform software and apps, there will be applications that are common to all IPTV service providers but that need to be restricted to certain IPTV service providers only. These would include standardized VoD applications, etc.

8.3.3.4 Use cases for the secure upgrade of SCP

8.3.3.4.1 General

In the security architecture, one party is assumed to be entitled to upgrade the firmware; this party is called the firmware owner. This party can be modified in a controlled manner if allowed by a custodian or a trust provider.

In the security architecture, one party is assumed to be entitled to upgrade the SP software; this party is called the SP owner. This party can be modified in a controlled manner if allowed by a custodian or a trust provider.

In the security architecture, one party is assumed to be entitled to upgrade the CP software; this party is called the CP owner. This party can be modified in a controlled manner if allowed by a custodian or a trust provider.

The upgrade use cases will be described in terms of firmware owner, SP software owner, and CP software owner since this abstraction allows a stakeholder-agnostic description, enabling the upgrade use cases to be valid for different business scenarios. For example, there may be cases wherein the service provider is the firmware owner, controlling the complete IPTV TD; in other cases this responsibility is outsourced to a security provider (SP provider or CP provider) or IPTV TD manufacturer. The same observation can be made for the SP software owner or CP software owner. Sometimes, this is completely in the hands of the IPTV service provider; in other cases, the responsibility is outsourced to a security provider.

8.3.3.4.2 User changes service provider

When a user changes to another IPTV service provider, the following steps may be needed before the user can enjoy the services of the new IPTV service provider:

1) Upgrade of firmware in case the new service provider requires ownership and use of dedicated firmware.
2) Upgrade of the SP software to adapt the IPTV TD to the SP system used by the IPTV service provider.
3) Upgrade of the CP software to adapt the IPTV TD to the CP system used by the IPTV service provider.

To allow the service provider to upgrade the firmware or provide the firmware to be downloaded by the IPTV TD, the service provider needs to request ownership of the firmware. The security solution clause implies that the service provider in this case first needs to provide a secondary bootloader and a public key certificate key, both signed by a trust provider.

Therefore, upgrade of the firmware in case of change of service provider may consist of two steps:

1a) The service provider requests to be firmware owner.

1b) The service provider requests an upgrade of firmware.

The individual sub-steps 1a) and 1b) are described in further detail in clause 8.3.3.5.

To allow the service provider to upgrade the SP software or provide the SP software to be downloaded by the IPTV TD, the service provider needs to request ownership of the SP software. The security solution implies that the service provider in this case first needs to provide an SP loader and a public key certificate key, both signed by the trust provider.

Therefore, upgrade of the firmware in case of change of service provider may consist of two steps:

2a) The service provider requests to be SP owner.

2b) The service provider requests for upgrade of SP software.

If other stakeholders need to take control of the firmware or SP software, the steps taken are the same.

The individual sub-steps 2a) and 2b) are described in further detail in clause 8.3.3.5.

To allow the service provider to upgrade the CP software or provide the CP software to be downloaded by IPTV TD, the service provider needs to request ownership of the CP software. The security solution implies that the service provider in this case first needs to provide a CP loader and a public key certificate key, both signed by the trust provider.

Therefore, upgrade of the firmware in case of change of service provider may consist of two steps:

3a) The service provider requests to be CP owner.

3b) The service provider requests upgrade of CP software.

If other stakeholders need to take control of the firmware or CP software, the steps taken are the same.

The individual sub-steps 2a) and 2b) are described in further detail in clause 8.3.3.5.

8.3.3.4.3 Stakeholder X requests to be firmware owner

The following steps will effectively change the firmware owner:

1) Stakeholder X's ACS initiates remote management connection with IPTV TD.
2) During connection establishment, mutual authentication is performed between the requesting stakeholder X's ACS and IPTV TD.
3) Stakeholder X's ACS instructs the IPTV TD to download the secondary bootloader package.
4) Before initiating such download, the IPTV TD asks the user if stakeholder X is allowed to take over the IPTV TD.
5) If affirmative, the IPTV TD will perform the requested download of the secondary bootloader package, which contains the secondary bootloader image, a public key of stakeholder X, a signature from the trust provider, and a public key certificate.
6) Only when the secondary bootloader can be verified as authentic (see clause 8.3.3.5) is the secondary bootloader allowed to be installed into IPTV TD.

Only when step 6 is successfully completed does the new secondary bootloader allow the loading of firmware images signed by stakeholder X.

8.3.3.4.4 Firmware owner requests upgrade of firmware

The following steps will upgrade the firmware:

1) The firmware owner's ACS initiates remote management connection with IPTV TD.
2) During connection establishment, mutual authentication is performed between the requesting firmware owner's ACS and IPTV TD.
3) The firmware owner's ACS instructs the IPTV TD to download the firmware.
4) The IPTV TD will perform the requested download of the secondary firmware package, which contains the firmware image and a signature from the firmware owner.
5) Only when the firmware can be verified as authentic (see clause 8.3.3.5) is the firmware allowed to be installed into the IPTV TD.

Only when step 5 is successfully completed will the new firmware be started by the TD on the next reboot.

8.3.3.4.5 Stakeholder Y requests to be SP owner

The following steps will effectively change the SP owner:

1) Stakeholder Y's ACS initiates remote management connection with IPTV TD.
2) During connection establishment, mutual authentication is performed between the requesting stakeholder Y's ACS and IPTV TD.
3) Stakeholder Y's ACS instructs the IPTV TD to download the SP loader package.
4) Before initiating such download, the IPTV TD asks the user if stakeholder Y is allowed to install modules for the consumption of IPTV.
5) If affirmative, the IPTV TD will perform the requested download of the secondary bootloader package, which contains the SP loader image, a public key of Stakeholder Y, a signature from the trust provider, and a public key certificate.
6) Only when the SP loader can be verified as authentic (see clause 8.3.3.5) is the secondary bootloader allowed to be installed into IPTV TD.

Only when step 6 is successfully completed does the new SP loader allow the loading of SP images signed by stakeholder Y.

8.3.3.4.6 SP software owner requests upgrade of SP software

The following steps will upgrade the SP software:

1) The firmware owner's ACS initiates a remote management connection with the IPTV TD.
2) During connection establishment, mutual authentication is performed between the requesting firmware owner's ACS and IPTV TD.
3) The firmware owner's ACS instructs the IPTV TD to download the SP software.
4) The IPTV TD will perform the requested download of the SP software package, which contains the SP image and a signature from the SP owner.
5) Only when the SP software can be verified as authentic (see clause 8.3.3.5) is the firmware allowed to be installed into IPTV TD.

Only when step 5 is successfully completed will the new SP software be started by the IPTV TD.

8.3.3.4.7 Stakeholder Y requests to be CP owner

The following steps will effectively change the CP owner:

1) Stakeholder Y's ACS initiates remote management connection with IPTV TD.
2) During connection establishment, mutual authentication is performed between the requesting stakeholder Y's ACS and IPTV TD.
3) Stakeholder Y's ACS instructs the IPTV TD to download the CP loader package.
4) Before initiating such download, the IPTV TD asks the user if stakeholder Y is allowed to install modules for the consumption of IPTV.
5) If affirmative, the IPTV TD will perform the requested download of the secondary bootloader package, which contains the CP loader image, a public key of stakeholder Y, a signature from the trust provider, and a public key certificate.
6) Only when the CP loader can be verified as authentic (see clause 8.3.3.5) is the secondary bootloader allowed to be installed into the IPTV TD.

Only when step 6 is successfully completed does the new CP loader allow the loading of SP images signed by Stakeholder Y.

8.3.3.4.8 CP software owner requests upgrade of CP software

The following steps will upgrade the CP software:

1) The firmware owner's ACS initiates remote management connection with the IPTV TD.
2) During connection establishment, mutual authentication is performed between the requesting firmware owner's ACS and IPTV TD.
3) The firmware owner's ACS instructs the IPTV TD to download the CP software.
4) The IPTV TD will perform the requested download of the CP software package, which contains the CP image and a signature from the CP owner.
5) Only when the CP software can be verified as authentic (see clause 8.3.3.5) is the firmware allowed to be installed into the IPTV TD.

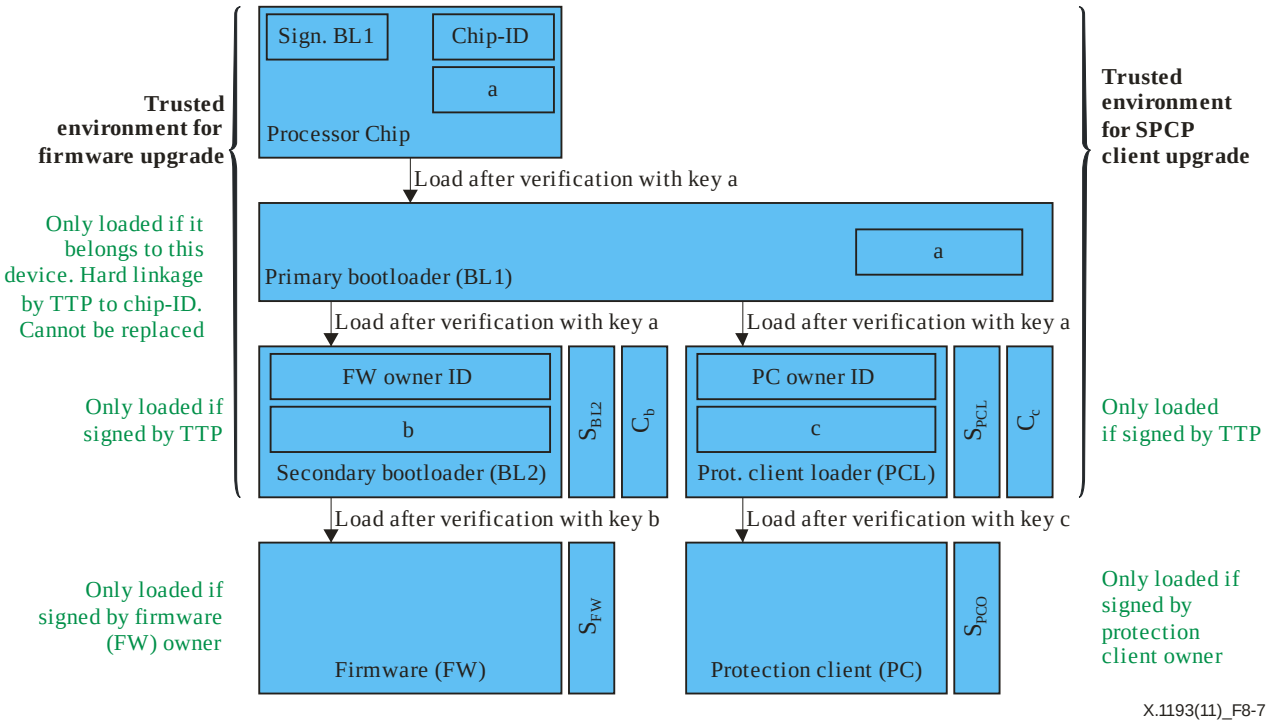
Only when step 5 is successfully completed will the new CP software be started by IPTV TD.

8.3.3.5 SP and/or CP upgrade security architecture

To ensure that an SP or a CP client is authentic and secure, the TD needs to provide a trusted environment. For this, all layers of the loaded software need to be verified for integrity and authenticity. In a hardware-supported trusted environment, the guarantee is provided by the lowest level (i.e., the hardware), and it will only load and start a trusted primary loader that in turn will load the next level, and so on. If this chain cannot be broken by attackers, then the whole environment provided by these layers is known and trusted for integrity and authenticity.

The following layers are foreseen for the SP and/or CP upgrade architecture: IPTV TD processor chip (i.e., hardware), primary and secondary boot loader, and firmware. The primary boot loader is used to load the secondary boot loader if it proves that it has integrity and authenticity. The secondary loader is used to load the TD IPTV firmware if it proves that it has integrity and authenticity. Since the only part of TD that is really trusted is the processor chip and the hard-linked primary bootloader (which cannot be changed after the production of the chip), this is also the basis for the load of the security systems, i.e., SP or CP loader. The SP or CP loader in turn loads the SP or CP client(s) (if required and available) after proving the authenticity and integrity.

This multi-layered boot process is shown in Figure 8-7.



Key pairs :
(a, A) is a key pair for a specific chipset family (a = public, A = private)
(b, B) is a key pair for a specific firmware owner (b = public, B = private)
(c, C) is a key pair for a protection client owner (c = public, C = private)
(z, Z) is a key pair uniquely for the trust provider to issue certificates (z = public, Z = private)

Signatures :

S_{BL1}	E(H(BL1|Chip-ID), A)
S_{BL2}	E(H(BL2|b), A)
S_{FW}	E(H(FW), B)
S_{PCL}	E(H(PCL|c), A)
S_{PC}	E(H(PC), C)

Certificates :
C_b = b + identities + S_b; S_b = E(H(b + identities), Z)
C_c = c + identities + S_c; S_c = E(H(c + identities), Z)

Figure 8-7 – Multi-layered boot process

TD processor chip: The TD processor chip supporting the trusted environment for SP and/or CP client software provides facilities that enable the loading of primary bootloader software with known integrity and authenticity. Upon power-on or restart of the processor chip, the chip has the responsibility to load and start the primary bootloader if, and only if, the image of the primary bootloader in ROM can be proven to have integrity and authenticity. For this proof, the chip has a burned-in public key *a* belonging to its chip series, a burned-in unique Chip-ID, and a burned-in signature of the primary bootloader. The primary bootloader concatenated with a Chip-ID has been signed with the private key *A* associated with the chip series. This private key is in possession of the trust provider, which is the only one that can sign with this private key. The verification of a primary boot loader image (BL1) is performed by creating a hash over BL1 concatenated with the Chip-ID, decrypting the burning in signature S_{BL1} using public key *a* and comparing both results. If, and only if, both results are identical, will the image be considered to have integrity and authenticity, and only then can this image be loaded. The creation of a signature over a concatenation of the primary bootloader image and the Chip-ID and burning of this signature into the chip ensures that no other primary bootloader can ever be loaded by this device aside from the original one. This ensures that the first level of software loaded is always known and trusted.

In short, the following are the responsibilities of the processor chip:

– Background:
  - E() is an encryption function that encrypts a bitstream provided in the first parameter with the key provided in the second parameter.
  - D() is a decryption function that decrypts a bitstream provided in the first parameter with the key provided in the second parameter.
  - H() is a hash function that calculates a hash value over a string of bits.

– Assume:
  - (*a*,*A*) is a key pair for a specific chipset family, where *a* is the public key stored in the device and *A* is the private key stored and known only to the trust provider.
  - S_{BL1} = E(H(BL1|ChipID), A).

– Precondition:
  - Chip-ID, S_{BL1}, and key *a* are embedded securely in the processor chip; the chip also contains the logic to perform D(), H(), and comparison of results.

– At start-up, load BL1 from ROM.

– Verify that H(BL1|Chip-ID) == D(S_{BL1},*a*).

– If the verification step is successful, then start BL1.

Advantages:

– The primary bootloader is guaranteed to always be the bootloader that the chip manufacturer has provided for this particular chip and that has been signed/certified by the trust provider.

– There is no way that a hacker can modify/replace the primary bootloader for this processor chip without breaking the TD that contains the processor chip.

Primary bootloader: When started by the processor chip, the primary bootloader has the responsibility to:

– load or download the secondary bootloader image (BL2), public key *b*, and signature S_{BL2} as well as a certificate for public key *b*; start BL2 if and only if the certificate for public key *b* is valid and the concatenation of image BL2 and key *b* can be proven to have integrity and authenticity using public key *a*;

– load or download one or more protection client loader image(s) (PCL), public key *c* and signature S_{PCL} as well as a certificate for public key *c*; start PCL if and only if the certificate for public key *c* is valid and the concatenation of image PCL and key *c* can be proven to have integrity and authenticity using public key *a*.

Signatures S_{BL2} and S_{PCL} have both been signed with private key *A* by the trusted provider. This private key is in the possession of the trust provider, which is the only one that can sign with this private key.

The verification of image BL2 is performed by creating a hash over BL2 concatenated with public key *b*, decrypting the loaded/downloaded signature S_{BL2} using key *a*, and comparing both results. If, and only if, both results are identical, will the image be considered to have integrity and authenticity, and only then can this image be started by BL1. The signature over a concatenation of BL2 and public key *b* means that the trust provider states that the linkage of BL2 with the owner of public key *b* is authentic. Certificate C_b authenticates public key *b* as belonging to the identities presented in the certificate. This ensures that the second level of software loaded can be attributed to a specific stakeholder, i.e., the firmware owner that is known and authenticated by the trust provider. Typically, this is the IPTV TD manufacturer or, in the case where the device is owned by an IPTV service provider, the IPTV service provider.

The verification of image PCL is performed by creating a hash over PCL concatenated with public key *c*, decrypting the loaded/downloaded signature S_{PCL} using key *a*, and comparing both results. If, and only if, both results are the same, will the image be considered to have integrity and authenticity, and only then can this image be started by BL1. The signature over a concatenation of PCL and public key *c* means that the trust provider states that the linkage of PCL with the owner of public key *c* is authentic. Certificate C_c authenticates that public key *c* belongs to the identities presented in the certificate. This ensures that the second level of software loaded can be attributed to a specific stakeholder i.e., the protection client owner (PCO) that is known and authenticated by the trust provider. Typically, this is the IPTV service provider that is using the specific protection client or, in the case where the IPTV service provider has outsourced this responsibility to a security provider, the SP or CP provider.

Listing the responsibilities of the primary bootloader:

– Assume:
  - (*b,B*) is a key pair for a specific firmware owner, where *b* is the public key delivered with the secondary bootloader and *B* is the private key stored by and known only to the firmware owner.
  - (*c, C*) is a key pair for a specific protection client owner, where *c* is the public key delivered with the secondary bootloader and *C* is the private key stored by and known only to the protection client owner.
  - (*z,Z*) is a key pair unique to the trust provider for the issuance of public key certificates, where *z* is the public key of the trust provider and *Z* is the private key stored by and known only to the trust provider.
  - Signature S_{BL2} = E(H(BL2|b), A).
  - Signature S_{PCL} = E(H(PCL|c), A).
  - Certificate C_b = b + identities + S_b; Signature S_b = E(Hash(b + identities),Z).
  - Certificate C_c = c + identities + S_c; Signature S_c = E(Hash(c + identities),Z).
– Loading and verifying the secondary bootloader:
  - At start, load or download BL2|*b* + S_{BL2} + C_b.
  - Verify that H(BL2| *b*) == D(S_{BL2}, *a*).
  - Verify that *b* is a valid public key for a stakeholder by verifying that H(*b* + identities) == D($S_{b,}$ z).
  - If the verification steps are successful, then start BL2.
– Loading and verifying a protection client loader:
  - At start, load or download PCL|*c* + S_{PCL} + C_c.
  - Verify that H(PCL| *c*) == D(S_{PCL}, *a*).
  - Verify that *c* is a valid public key for a stakeholder by verifying that H(*c* + identities) == D($S_{c,}$ z).
  - If the verification steps are successful, then start PCL.

Advantages:

– The secondary bootloader and the protection client loader(s) are guaranteed by the trust provider having signed this linkage to be authentic and to belong to a particular stakeholder with whom it is linked. Furthermore, the identity of this owner stakeholder is certified by the trust provider for such identities.
– There is no way that a hacker can modify/replace the secondary bootloader for this chip family with a secondary bootloader that has not been signed by the trust provider, as so doing would make the TD dysfunctional.

The secondary bootloader is used to load or download, verify, and start the firmware belonging to a firmware owner. The secondary bootloader verifies that a firmware image (FW) has been signed by the firmware owner (using private key *B*); FW will be started if and only if this verification is successful. Keys *B* and *b* are owned by the firmware owner.

Listing the responsibilities of the secondary bootloader:

– Assume:
  - Signature S_{FW} = E(H(FW), *B*).
– Loading and verifying the firmware:
  - At start, load or download FW + S_{FW}.
  - Verify that H(FW) == D(S_{FW}, *b*).
  - If the verification steps are successful, then start FW.

Advantages:

– The firmware is guaranteed to have integrity and authenticity as signed by the firmware owner.
– There is no way that a hacker can modify/replace the firmware for this chip family with firmware that has not been signed by the firmware owner as so doing would make the TD dysfunctional.
– This set-up only allows the firmware owner to replace or upgrade the firmware. Typically, this is the IPTV TD manufacturer or, if the device is owned by an IPTV service provider, the IPTV service provider.

The protection client loader is used to load or download, verify, and start the protection client belonging to a protection client owner. The protection client loader verifies that a protection client image (PC) has been signed by the protection client owner (using private key *C).* If, and only if, this verification is successful, will the protection client be started. Keys *C* and *c* are owned by the protection client owner.

Listing the responsibilities of the secondary bootloader:

– Assume:
  - Signature S_{PC} = E(H(PC), *C*).
– Loading and verifying the firmware:
  - At start, load or download PC + S_{PC}.
  - Verify that H(PC) == D(S_{PC}, *c*).
  - If the verification steps are successful, then start PC.

Advantages:

– The protection client is guaranteed to have integrity and authenticity as signed by the protection client owner.
– There is no way that a hacker can modify/replace the protection client for this chip family with a protection client that has not been signed by the protection client owner; doing so would make the TD dysfunctional.
– This set-up only allows the protection client owner to replace or upgrade the protection client. Typically, this is the IPTV service provider that is using the specific protection client or, if the IPTV service provider has outsourced this responsibility to a security provider, the SP or CP provider.

Appendix I

Multimedia Internet keying

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

I.1 Overview of MIKEY

Multimedia Internet keying (MIKEY) is a key management scheme developed by IETF. It can be applied to real-time applications such as IPTV. In particular, MIKEY is designed for Secure Real-time Transport Protocol (SRTP) support. In addition, it is mainly intended for use by peer-to-peer, simple one-to-many, and small-sized (interactive) groups.

MIKEY is a two-way key exchange protocol that aims at sharing a master key and the security policy for the encryption process.

Figure I.1 presents an overview of the MIKEY key management procedure.

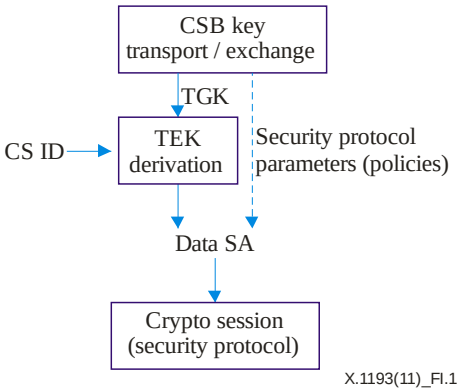


Figure I.1 – Overview of MIKEY key management procedure

The procedure for setting up CSB (crypto session bundle) and creating TEK (traffic-encrypting key) (and data SA) is the following:

- A set of security parameters and TGK (TEK generation key)(s) are agreed upon for the crypto session bundle.
- TGKs are used to derive TEK for each crypto session.

TEK, together with the security protocol parameters, represents the data SA used as input to the security protocol.

I.2 Protocol operation

There are three methods for key exchange: Pre-shared key, public key encryption, and Diffie-Hellman key exchange.

The following notations are used to describe the operation of the three types of key exchange:

- HDR: MIKEY header
- T: Timestamp
- IDx: Identity of entity x
- RAND: Random/pseudo-random byte-string
- SP: Security policy

I.2.1 Pre-shared key

This method uses the pre-shared key to derive key material for encryption (encr_key) and integrity (auth_key).

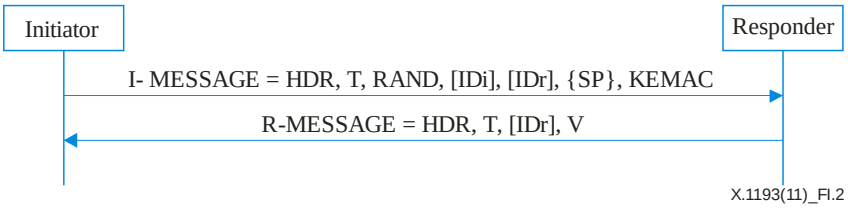


Figure I.2 – Pre-shared key

where KEMAC = E(encr_key, {TGK}) || MAC, MAC is a message authentication code covering the entire MIKEY message using auth_key, and V(verification message) is MAC computed over the Responder's entire message, the timestamp, and the two parties' identities using auth_key.

I.2.2 Public key encryption

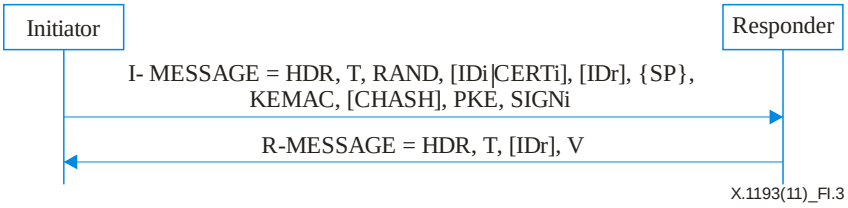


Figure I.3 – Public key encryption

where PKE = E(PKr, env_key), PKr is the responder's public key, KEMAC = E(encr_key, IDi || {TGK}) || MAC, and SIGNi is a signature covering the entire MIKEY message.

I.2.3 Diffie-Hellman key exchange

This method creates a DH-key, which is used as TGK.

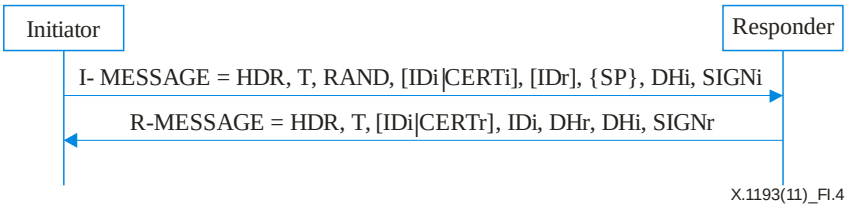


Figure I.4 – Diffie-Hellman key exchange

Appendix II

Extensible authentication protocol

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

This appendix is intended merely to give potential EAPs for use in the key management scheme in this Recommendation. EAPs do not refer to mandatory-to-use EAP. Mandatory-to-use key management methods can be selected according to the authentication policy.

II.1 EAP-AKA

The EAP-AKA [b-IETF RFC 4187] is an EAP method that uses the existing AKA (authentication and key agreement) mechanism that was developed for authentication and key exchange in the 3G cellular network. The AKA is used for mutual authentication and the session key derivation based on the shared symmetric key, which can be used to protect the data session in the air interface in the 3G cellular networks. On the peer side, it runs in a subscriber identity module, which is either a UMTS subscriber identity module (USIM) or a (removable) user identity module((R)UIM), similar to a smart card. In the 3G context, an entity called HLR (home location register) acts as the authentication server, an entity called VLR (visitor location register) acts as the authenticator, and a mobile station (MS) acts as the peer, respectively.

Basically, EAP-AKA incorporates AKA into the EAP method to perform authentication and session key derivation as well as optional identity privacy support and optional result indications including an optional fast re-authentication procedure. In addition, the peer is assumed to have access to the subscriber's USIM where the shared secret (K) is kept, and the actual AKA protocol is implemented. The master key (MK) is computed from the IK (Integrity key) and CK (cipher key) computed during the EAP-AKA method run. MK is used to compute the transient EAP session key (TEKs), MSK, and EMSK.

II.2 EAP-PSK

EAP-PSK [b-IETF RFC 4764] is EAP based on the "pre-shared key." It provides mutual authentication based on a 16-byte pre-shared secret between the peer and the EAP server. Mainly designed to apply to a context with restricted computational resources especially for mobile terminal in wireless networks, EAP-PSK uses only one primitive cryptographic algorithm, namely the AES algorithm. There are two types of EAP-PSK methods: standard EAP-PSK method and extended EAP-PSK method. The standard EAP-PSK method uses the protected channel to transmit a protected result indication, whereas the extended EAP-PSK uses the protected tunnel to transmit the arbitrary information in variable length. It is regarded as a typical challenge/response protocol in that two parties exchange their nonces, their identities, and a proof of knowledge of the secret. Authentication can be achieved by sending a MAC computed with the pre-shared key over the nonces and identities exchanged in the previous conversation.

EAP-PSK is based on AKEP2 (authenticated key exchange protocol 2). The assumption is that two parties should have shared two keys as a prerequisite: a1 and a2, where a1 is used for authentication purposes, and a2 for session key derivation.

EAP-PSK supports mutual authentication, key derivation, and dictionary attack resistance, but not identity protection, fast reconnect, and protected ciphersuite negotiation.

II.3 EAP-TLS

First published in October 1999 as [b-IETF RFC 2716], which was replaced by [b-IETF RFC 5216] in March 2008, EAP-TLS [b-IETF RFC 2716] is considered a mature, stable, and widely deployed EAP method. It relies on transport layer security.

EAP-TLS uses a TLS handshake phase to authenticate the peer and the authentication server. Although the TLS handshake protocol actually sets up a secure tunnel between the peer and the authentication server, this tunnel is not used in the subsequent data session. Instead, as some keying materials are sent to the authenticator, the peer and the authenticator use them to protect the subsequent data session. In EAP-TLS, certificates are used to authenticate the EAP authentication server to the peer and – optionally – to authenticate the peer to the authentication server. Therefore, mutual authentication is enabled based on the ITU-T X.509 certificates, thereby protecting against MITM attacks and use of a rogue network access server. It also generates the symmetric keying material that can be used to protect the subsequent data session. After EAP-TLS is completed, the authentication server and the peer are able to share the pre-master secret used to generate the master secret (MS), which in turn is used to generate MSK, EMSK using the pseudo-random function.

EAP-TLS can be considered a secure EAP method; as such, it is now being deployed widely in many applications. It supports fast reconnect since new security associations can be generated by using the existing security association efficiently and fast. In sum, it supports most requirements, except channel binding and identity protection. Since EAP-TLS uses certificates, it inherits all certificate-related problems, e.g., a problem from unencrypted certificates and a problem of postponed verification of the certificate. The first problem arises when certificates are sent unencrypted. As a result, the identity contained in the certificate is revealed to the attacker, who is able to eavesdrop on the conversation. The second problem occurs when the peer is unable to verify the signature or the certificate chain. Furthermore, the peer is unable to verify whether the certificate of the authentication server has been revoked in the meantime. Therefore, there is no other means of avoiding the problem aside from postponing verification.

II.4 EAP-FAST

EAP-FAST [b-IETF RFC 4851] is an alternative EAP. It was originally proposed to reduce the workload of small wireless devices. FAST stands for "flexible authentication via secure tunnelling." The primary design goals of EAP-FAST include mutual authentication, resistance to brute-force dictionary attacks, immunity to the MITM attack, and large support for existing user database containing credentials. In general, EAP-FAST uses the TLS handshake protocol to establish a mutually authenticated tunnel between the peer and the authentication server. Unlike EAP-TTLS, however, the secure tunnel can be established using either the public key similar to EAP-TLS or a pre-shared symmetric key known as PAC (protected access credential). PAC can be considered a security token provided to the peer by the server to establish a secure tunnel for future optimized network authentication. EAP-FAST consists of two phases. In the first phase, the peer uses PAC to establish the secure TLS tunnel. If the peer does not have the corresponding PAC, the server requests the peer to initiate the full TLS handshake. Following this full TLS handshake, the peer requests the server to issue the PAC that can be used to establish the TLS tunnel later. In the second phase, EAP-TLS, such as authentication or legacy authentications, may be used to authenticate the peer within the secure tunnel. PAC consists of three components: a shared secret, an opaque element, and other optional information. The shared secret is used to establish the secure tunnel. The opaque element is provided to the peer and presented to the server when the peer wishes to obtain access to the network resource. The opaque element may include PAC as well as the peer's identity. The server uses a strong cryptographic algorithm to protect the opaque element and to recover the necessary information for the server to identify and authenticate the peer. "Other information" may be contained to confirm the integrity of the PAC issuer.

There are three kinds of authentication methods: certificate-based authentication that is used in EAP-TLS, combined authentication that is used in EAP-TTLS, or PAC (protected access credential)-based authentication. In certificate-based authentication, the peer and the authentication server use the certificates to authenticate each other. In PAC-based authentication, the peer uses PAC to establish a TLS tunnel. Therefore, EAP-FAST is considered an efficient EAP method that combines the features of EAP-TLS and EAP-TTLS and adopts the idea of using EAP-TLS with the pre-shared key. In short, EAP-FAST is a very flexible EAP method that is intended for the constricted mobile device since it supports mutual authentication by using a pre-shared key.

II.5 EAP-IKEv2

EAP-IKEv2 [b-IETF RFC 5106] was adopted in February 2008. Based on the mechanisms and payloads of IKEv2, this EAP method provides mutual authentication and session key establishment between an EAP server and an EAP peer. In order to provide mutual authentication, it supports various authentication techniques according to the types of credentials: asymmetric key pairs, symmetric keys, and a combination of both. A different authentication credential may be used in each direction. For instance, the EAP server authenticates itself using public key pairs, whereas the peer does so using the symmetric key.

II.6 EAP-TTLS

EAP-TTLS is in RFC 5281 [b-IETF RFC 5281]. An EAP (Extensible Authentication Protocol) method, EAP-TTLS is based on the TLS (Transport Layer Security) protocol. TTLS stands for "Tunnel Transport Layer Security". EAP-TTLS is considered an extension to EAP-TLS. Authentication in EAP-TLS is typically mutual, i.e., the authentication server and the peer authenticate each other. It uses the certificate to authenticate the authentication server and a simpler authentication method to authenticate the peer. It consists of two phases: the TLS handshake phase and the TLS tunnel phase. In the first phase, the authentication server is authenticated to the peer using the ITU-T X.509 certificate of the server. After the first phase is completed, a secure tunnel is established. In the second phase, all communications are protected by this secure channel. The client is authenticated to the authentication server by using the legacy authentication methods, such as clear-text password or challenge-response password, or a more advanced authentication mechanism, such as token-based authentication. EAP-TTLS supports identity protection since an attacker cannot see the user identity because the identity can be sent in the second phase. Nonetheless, EAP-TTLS is known to be vulnerable to the man-in-the-middle (MITM) attack. The tunnelled protocols require the session key derived from the first phase, which is used to provide a secure tunnel. In a certain environment, a peer is allowed to skip the first phase and to proceed directly to the second phase. In this case, an active MITM attack may occur if the attacker can hijack a valid authentication session. Note, however, that a cryptographic binding scheme was proposed to protect against the MITM attack in the tunnel-based EAP method. Therefore, EAP-TTLS can be considered to be secure if cryptographic binding is applied. In addition, as of December 2008, the IETF EMU (EAP methods update) working group has been developing the Internet draft on the "requirements of tunnel-based EAP method".

II.7 PEAP

As a proprietary protocol, PEAP (Protected Extensible Authentication Protocol) provides an encrypted, authenticated tunnel using the TLS handshake protocol that encapsulates further authentication mechanisms for the peer. It uses TLS to protect against rogue authenticators, protect against various attacks on the confidentiality and integrity of the inner EAP method exchange, and ensure EAP peer identity privacy. PEAP also provides support for chaining of multiple EAP mechanisms, cryptographic binding between authentications performed by the inner EAP mechanisms and the tunnel, exchange of arbitrary parameters, and fragmentation and reassembly. It uses public key cryptography for the authentication and negotiation of the key that can be used to

encrypt data. PEAP also uses TLS for server authentication and encryption yet foregoes the need for user certificates by using a second authentication protocol between the peer and the server, which is protected by the TLS encryption. The basic principles of EAP-TTLS and PEAP is nearly identical. The main difference between them is that PEAP can only use legacy authentication methods such as ID/password-based authentication in the second phase, whereas EAP-TTLS can use either other EAP methods or any legacy authentication method. However, as of December 2008, it is still in the Internet draft phase of IETF.

Bibliography

[b-ITU-T X.1034] Recommendation ITU-T X.1034 (2008), *Guideline on extensible authentication protocol based authentication and key management in a data communication network*.

[b-ITU-T Y.1901] Recommendation ITU-T Y.1901 (2009), *Requirements for the support of IPTV services*.

[b-ITU-T Y.1910] Recommendation ITU-T Y.1910 (2008*), IPTV functional architecture*.

[b-ITU-T Y-Sup.5] ITU-T Y.1900 series – Supplement on IPTV service use cases.

[b-ATIS 0800001] ATIS 08000001 (2006), *IPTV DRM Interoperability Requirements,* ATIS-IIF.

[b-ATIS 0800006] ATIS 08000006 (2007), *IIF Default Scrambling Algorithm (IDSA) IPTV Interoperability Specification*.

[b-ETSI TR 102 825-6] ETSI TR 102 825-6 (2011), *Digital Video Broadcasting (DVB); Content Protection and Copy Management (DVB-CPCM); Part 6: CPCM Security Test Vectors*.

[b-ETSI TR 102 825-8] ETSI TR 102 825-8 (2011), *Digital Video Broadcasting (DVB); Content Protection and Copy Management (DVB-CPCM); Part 8: CPCM Authorized Domain Management scenarios*.

[b-ETSI TR 102 825-11] ETSI TR 102 825-11 (2011), *Digital Video Broadcasting (DVB); Content Protection and Copy Management (DVB-CPCM); Part 11: CPCM Content Management Scenarios*.

[b-ETSI TR 102 825-12] ETSI TR 102 825-12 (2011), *Digital Video Broadcasting (DVB); Content Protection and Copy Management (DVB-CPCM); Part 12: CPCM Implementation Guidelines*.

[b-ETSI TR 102 825-13] ETSI TR 102 825-13 (2011), *Digital Video Broadcasting (DVB); Content Protection and Copy Management (DVB-CPCM); Part 13: CPCM Compliance Framework.*

[b-ETSI TR 187 013] ETSI TR 187 013 (2011), *Telecommunications and Internet Converged Services and Protocols for Advanced Networking (TISPAN); Feasibility study on IPTV Security Architecture*.

[b-ETSI TS 102 825-1] ETSI TS 102 825-1 (2011), *Digital Video Broadcasting (DVB); Content Protection and Copy Management (DVB-CPCM); Part 1: CPCM Abbreviations, Definitions and Terms*.

[b-ETSI TS 102 825-2] ETSI TS 102 825-2 (2011), *Digital Video Broadcasting (DVB); Content Protection and Copy Management (DVB-CPCM); Part 2: CPCM Reference Model*.

[b-ETSI TS 102 825-3] ETSI TS 102 825-3 (2011), *Digital Video Broadcasting (DVB); Content Protection and Copy Management (DVB-CPCM); Part 3: CPCM Usage State Information*.

[b-ETSI TS 102 825-4] ETSI TS 102 825-4 (2011), *Digital Video Broadcasting (DVB); Content Protection and Copy Management (DVB-CPCM); Part 4: CPCM System Specification*.

[b-ETSI TS 102 825-5] ETSI TS 102 825-5 (2011), *Digital Video Broadcasting (DVB); Content Protection and Copy Management (DVB-CPCM); Part 5: CPCM Security Toolbox*.

[b-ETSI TS 102 825-7] ETSI TS 102 825-7 (2011), *Digital Video Broadcasting (DVB); Content Protection and Copy Management (DVB-CPCM); Part 7: CPCM Authorized Domain Management*.

[b-ETSI TS 102 825-9] ETSI TS 102 825-9 (2011), *Digital Video Broadcasting (DVB); Content Protection and Copy Management (DVB-CPCM); Part 9: CPCM System Adaptation Layers*.

[b-ETSI TS 102 825-10] ETSI TS 102 825-10 (2011), *Digital Video Broadcasting (DVB); Content Protection and Copy Management (DVB-CPCM); Part 10: CPCM Acquisition, Consumption and Export Mappings*.

[b-ETSI TS 102 825-14] ETSI TS 102 825-14 (2011), *Digital Video Broadcasting (DVB); Content Protection and Copy Management (DVB-CPCM); Part 14: CPCM Extensions*.

[b-IETF RFC 2716] IETF RFC 2716 (1999), *PPP EAP TLS Authentication Protocol*.

[b-IETF RFC 3740] IETF RFC 3740 (2004), *The Multicast Group Security Architecture*.

[b-IETF RFC 3830] IETF RFC 3830 (2004), *Multimedia Internet Keying*.

[b-IETF RFC 4046] IETF RFC 4046 (2005), *Multicast Security (MSEC) Group Key Management Architecture*.

[b-IETF RFC 4187] IETF RFC 4187 (2006), *Extensible Authentication Protocol Method for 3rd- Generation Authentication and Key Agreement (EAP-AKA)*.

[b-IETF RFC 4764] IETF RFC 4764 (2007), *The EAP-PSK Protocol: A Pre-Shared Key Extensible Authentication Protocol (EAP) Method*.

[b-IETF RFC 4851] IETF RFC 4851 (2007), *The Flexible Authentication via Secure Tunneling Extensible Authentication Protocol Method (EAP-FAST*.

[b-IETF RFC 5106] IETF RFC 5106 (2008), *The Extensible Authentication Protocol-Internet Key Exchange Protocol Version 2(EAP-IKEv2) Method*.

[b-IETF RFC 5216] IETF RFC 5216 (2008), *The EAP-TLS Authentication Protocol*.

[b-IETF RFC 5281] IETF RFC 5281 (2008), *Extensible Authentication Protocol Tunneled Transport Layer Security Authenticated Protocol Version 0 (EAP-TTLSv0)*.

[b-NIST SP 800-120] NIST SP 800-120 (2009), *Recommendation for EAP Methods used in Wireless Network Access Authentication*.

[b-DCAS] OpenCable (2006), *DCAS System Overview Technical Report*.

SERIES OF ITU-T RECOMMENDATIONS

Series A	Organization of the work of ITU-T
Series D	General tariff principles
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	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	Terminals and subjective and objective assessment methods
Series Q	Switching and signalling
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 and next-generation networks
Series Z	Languages and general software aspects for telecommunication systems

Printed in Switzerland
Geneva, 2012