ETSI TS 187 021 V3.1.1 (2011-09)

Telecommunications and Internet converged Services and Protocols for Advanced Networking (TISPAN); Security services and mechanisms for customer premises networks connected to TISPAN NGN

Reference
DTS/TISPAN-07047-NGN-R3

Keywords
gateway, IP, security

ETSI

650 Route des Lucioles
F-06921 Sophia Antipolis Cedex - FRANCE

Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Siret N° 348 623 562 00017 - NAF 742 C
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° 7803/88

Important notice

Individual copies of the present document can be downloaded from:
http://www.etsi.org

The present document may be made available in more than one electronic version or in print. In any case of existing or perceived difference in contents between such versions, the reference version is the Portable Document Format (PDF). In case of dispute, the reference shall be the printing on ETSI printers of the PDF version kept on a specific network drive within ETSI Secretariat.

Users of the present document should be aware that the document may be subject to revision or change of status. Information on the current status of this and other ETSI documents is available at
http://portal.etsi.org/tb/status/status.asp

If you find errors in the present document, please send your comment to one of the following services:
http://portal.etsi.org/chaircor/ETSI_support.asp

Copyright Notification

No part may be reproduced except as authorized by written permission.
The copyright and the foregoing restriction extend to reproduction in all media.

© European Telecommunications Standards Institute 2011.
All rights reserved.

DECT™, **PLUGTESTS**™, **UMTS**™ and the ETSI logo are Trade Marks of ETSI registered for the benefit of its Members.
3GPP™ and **LTE**™ are Trade Marks of ETSI registered for the benefit of its Members and of the 3GPP Organizational Partners.
GSM® and the GSM logo are Trade Marks registered and owned by the GSM Association.

Contents

Intellectual Property Rights

IPRs essential or potentially essential to the present document may have been declared to ETSI. The information pertaining to these essential IPRs, if any, is publicly available for **ETSI members and non-members**, and can be found in ETSI SR 000 314: *"Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards"*, which is available from the ETSI Secretariat. Latest updates are available on the ETSI Web server (http://ipr.etsi.org).

Pursuant to the ETSI IPR Policy, no investigation, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in ETSI SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become, essential to the present document.

Foreword

This Technical Specification (TS) has been produced by ETSI Technical Committee Telecommunications and Internet converged Services and Protocols for Advanced Networking (TISPAN).

1 Scope

The present document specifies the functional models and information flows (stage 2) and protocols (stage 3) which implement the security services and mechanisms required to provide security in a Customer Premises Network (CPN) to support the overall security architecture for NGN release 3. CPN security services and mechanisms are used either singly or in combination to realize the CPN security requirements specified in TS 187 001 [1] (NGN Security requirements). Reference will be made to TR 185 012 [i.1] for security mechanisms that have been shown to be appropriate for CPN environment.

2 References

References are either specific (identified by date of publication and/or edition number or version number) or non-specific. For specific references, only the cited version applies. For non-specific references, the latest version of the reference document (including any amendments) applies.

Referenced documents which are not found to be publicly available in the expected location might be found at http://docbox.etsi.org/Reference.

NOTE: While any hyperlinks included in this clause were valid at the time of publication, ETSI cannot guarantee their long term validity.

2.1 Normative references

The following referenced documents are necessary for the application of the present document.

[1] ETSI TS 187 001: "Telecommunications and Internet converged Services and Protocols for Advanced Networking (TISPAN); NGN SECurity (SEC); Requirements".

[2] ETSI TS 185 006: "Telecommunications and Internet converged Services and Protocols for Advanced Networking (TISPAN); Customer Devices architecture and Reference Points".

[3] ETSI TS 185 003: "Telecommunications and Internet converged Services and Protocols for Advanced Networking (TISPAN); Customer Network Gateway (CNG) Architecture and Reference Points".

[4] ETSI TS 187 003: "Telecommunications and Internet converged Services and Protocols for Advanced Networking (TISPAN); NGN Security; Security Architecture".

[5] Broadband Forum TR-069 Amendment 3: "CPE WAN Management Protocol", November 2010.

[6] Broadband Forum TR-157 Amendment 3: "Component Objects for CWMP", November 2010.

2.2 Informative references

The following referenced documents are not necessary for the application of the present document but they assist the user with regard to a particular subject area.

[i.1] ETSI TR 185 012: "Telecommunications and Internet converged Services and Protocols for Advanced Networking (TISPAN) Feasibility study of security mechanisms for customer premises networks connected to TISPAN NGN".

[i.2] IETF RFC 5209 (June 2008): "Network Endpoint Assessment (NEA): Overview and Requirements".

[i.3] ETSI ES 282 003: "Telecommunications and Internet converged Services and Protocols for advanced Networking (TISPAN); Resource and Admission Control Sub-System (RACS): Functional Architecture".

[i.4] ETSI TS 102 825 (all parts): "Digital Video Broadcasting (DVB); Content Protection and Copy Management (DVB-CPCM)".

[i.5] ETSI TS 183 065: "Telecommunications and Internet converged Services and Protocols for Advanced Networks(TISPAN); Customer Network Gateway Configuration Function; e3 Interface based upon CWMP".

[i.6] Broadband Forum TR-069: "CPE WAN Management Protocol".

[i.7] IEEE 802.16: "IEEE Standard for Local and metropolitan area networks Part 16: Air Interface for Broadband Wireless Access Systems".

[i.8] IEEE 802.1b: "IEEE Standard for Local and Metropolitan Area Networks - Local and Metropolitan Area Network: LAN/MAN Management".

[i.9] Home Gateway Initiative: "Home Gateway Technical Requirements V.1.0".

3 Definitions and abbreviations

3.1 Definitions

For the purposes of the present document, the terms and definitions given in TS 187 003 [4] and Broadband Forum TR-157 [6] apply.

3.2 Abbreviations

For the purposes of the present document, the following abbreviations apply:

ACS	Auto-Configuration Server
AKA	Authentication and Key Agreement
ALG	Application Level Gateway
API	Application programming Interface
B2BUA	Back to back User Agent
BGF	Border Gateway Function
BL1	Boot Loader image
CA	Conditional Access
C-BGF	Core- Border Gateway Function
CND	Customer Network Device
CND-CMF	CND-Configuration and Management Function
CND-CPF	CND-Content Protection Function
CND-CSMF	CND-Communication Service Media Function
CND-SPF	CND-Service protection Function
CNG	Customer Network Gateway
CP	Content Protection
CPE	Consumer Premise Equipment
CPN	Customer Premises Network
CW	Control Words
DLNA	Digital Living Network Alliance
DMZ	Demilitarized Zone
DOS	Denial of Service
DRM	Digital Right Management
DSL	Digital Subscriber Line
DVB	Digital Video Broadcasting
DVB-CPCM	DVB Content Protection & Copy Management
FW	Firmware
HGI	Home Gateway Initiative
HTTP	Hyper Text Transfer Protocol
ICMP	Internet Control Message Protocol
IMS	IP Multimedia Subsystem

IPSEC	Internet Protocol SECurity
IPTV SP	IPTV Service Provider
IPTV	Internet Protocol TeleVision
IPv6	Internet Protocol version 6
ISP	Internet Service Provider
MCF	Media Control Function
MDF	Media Delivery Function
MF	Media Function
MFC	Media Control Function
NAC	Network Access Control
NAT	Network Address Translation
NEA	Network Endpoint Assessment
OMA	Open Mobile Alliance
PAT	Port Address Translation
PC	Protection Client
PCL	Protection Client Loader
PCO	Protection Client Owner
P-CSCF	Proxy-Call Session Control Function
PDA	Personal digital assistant
PPP	Point to Point Protocol
QoS	Quality of Service
ROM	Read Only Memory
RTP	Real-time Transport Protocol
RTSP	Real Time Streaming Protocol
SMS	Short Message Service
SOC	Security-on-Chip
SP	Service Protection
SP/CP	Service Protection and/or Content Protection
SSL	Secure Socket Layer
STB	Set Top Box
TCP	Transmission Control Protocol
TLS	Transport Layer Security
UC	Unsolicited Communication
UDP	Datagram Protocol
UE	User Equipment
UICC	Universal Integrated Circuit Card
UPnP	Universal Plug and Play
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
UTM	Unified Threat Management
VOD	Video On Demand
VPN	Virtual Private Network
WFA	Wi-Fi Alliance
Wi-Fi	Wireless Fidelity
WPA2	Wi-Fi Protected Access 2

4 General overview

This clause introduces the subset of security mechanisms to be evaluated and specified in details within the present document. The security mechanisms has been selected mainly (but not only) from the contents of the TR 185 012 [i.1].

5 Firewalling

The main mechanism to perform Network Access Control is a firewall, i.e. a system designed to permit, deny or proxy data traffic to or from the customer's network. A firewall is positioned to control all incoming and outgoing traffic; hence the CNG is the perfect candidate to perform the firewall functions.

5.1 Firewalling: basic description

There are several approaches to implements firewall functionalities, such as:

- Packet Filtering: the simplest one inspects each incoming or outgoing IP packet permitting, dropping or rejecting it on the basis of simple policies (usually defined as access control list) such as the IP address and the protocol type.
- Stateful Firewall: in addition to a Packet Filter, keeps track on IP packets belonging to the same connection thereby detecting whether a packet is part of an existing connection or a start of a new connection.
- Application Level Gateway: In addition to a stateful firewall can understand the behaviour of some applications and can detect e.g. if an illegal protocol is used for a given application or dynamically open ports for additional sessions belonging to a flow.

Firewalls can divide the network into subnets each one with a different level of security and different security policy as for example a demilitarized zone.

The firewall could have several configuration alternatives.

- A basic/minimum configuration to ensure a minimum level of security.
- One or several default configurations provided and managed by the operator/service provider through a remote management system.
- Additional alternative configurations that can depend on the user (e.g. there can be different configurations for parents and children). These user specific configurations could be managed by the same entity managing the user identity (e.g. the UICC).

5.2 Firewalling: architecture

In the CPN context, the CNG sits between the NGN and the internal network and this aspect makes the CNG as the perfect candidate to host the firewall functions. Figure 1 shows a typical scenario where the CNG and the Firewall are co-located on the same device. The external interface is the one that is connected to the NGN via e.g. xDSL, IEEE 802.16 [i.7] wireless modem, FTTx, etc., and is often referred to as the unsecure (red) interface. The secure (black) internal interfaces are connected to the CNDs and can be based on ethernet, IEEE 802.1b [i.8] and other wired or wireless communication technologies. The firewall may also implement a DMZ.

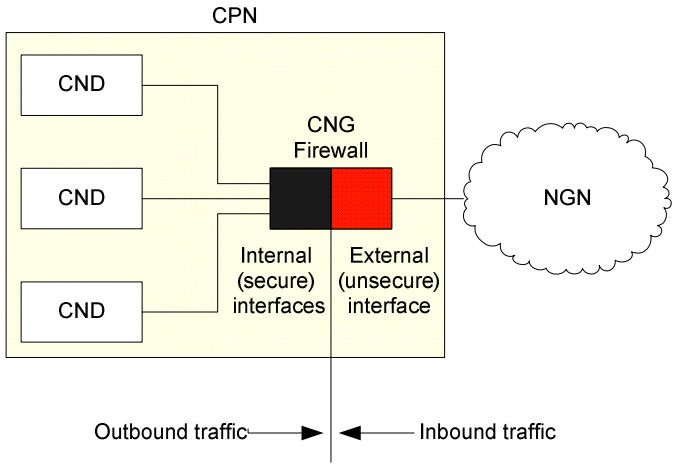


Figure 1: Firewall in the CPN

The advantages of using a Firewall as shown in the picture (i.e. co-located on the CNG) is that the CNG appears to the external network (i.e. NGN) as the only point of contact for the CPN, simplifying the protection of the CNDs against threats that originate on the NGN.

5.3 Firewalling: implementation details

For the protection of the CPN, a firewall should support some basic features, such as security policy definition and enforcing, firewall management, logging functions and so on. The following clause describes in details such features.

5.3.1 Stateful inspection

The stateful firewall function is mandatory for the protection of the CPN, such a firewall function may be implemented in the CNG. While a packet filter decides whether or not to drop a packet based on few information contained in the packet headers (e.g. addressing information), a stateful packet filter takes its decisions also on the state information that the firewall keeps in memory about all active connections travelling across it.

For connection-oriented protocols, such as TCP, the state of the connection is equivalent to the protocols definition of a connection (i.e. three-way handshake), whereas for a connection-less protocol, such as UDP, the state of the connection is the set of packets that are sent between common endpoints (i.e. source IP address/port and destination IP address/port) without interruption, i.e. the lack of any packets matching that flow for a given period of time. For the CPN context such a period of time shall be one minute.

The stateful firewall shall also perform additional structural checks on network packets. These checks include e.g. quickly dropping of malformed packet and enforcing the TCP three-way handshake to establish and teardown network connections.

5.3.2 Communication technologies

The Firewall shall be enabled on the local CPN network including all kind of wired and wireless connectivity used on the CPN, as well as remote access connections such as PPP over Ethernet and Virtual Private Network on the WAN side of the CNG. Note however that the firewall cannot be enabled when the CNG acts as a network bridge.

IPv6 firewalling shall be implemented in case the CNG supports IPv6 traffic.

5.3.3 Security policy

The firewall could have several configuration alternatives. In order to simplify the management of the security policy and still provide a basic level of security to the CPN it is proposed to define one or more security profiles.

As defined by HGI in [i.9] at least the following basic configurations shall be supported by the firewall: *HIGH* security configuration and *LOW* security configuration.

The *HIGH* security configuration foresees the following behaviour:

- For the traffic originated from the NGN toward the CPN (inbound): to refuse connections in TCP, UDP and ICMP; to authorize already established connections only (and known by the stateful firewall).
  Based on the Operator/Service Provider local policy, the firewall could accept incoming connections for specific services/ports, such as 5 060 for SIP (e.g. inbound SIP calls).

- For the traffic originated from the CPN toward the NGN (outbound): to authorize only well known ports, such as:

  - 25 - SMTP
  - 80 - HTTP
  - 443 - SSL
  - 554 - RTSP
  - 995 - POP3
  - 123 - NTP
  - 5 060 - SIP

A second alternative basic firewall configuration shall be supported by the firewall, the *LOW* security configuration: all traffic (inbound and outbound) is authorized by default. Anyway the stateful firewall still performs the security check on the TCP/UDP active sessions.

Also Internet Control Message Protocol (ICMP) messages should be managed because these messages can be used in hacking and DOS attacks. The firewall should block or allow specific ICMP options (e.g. Echo Requests, destination unreachable).

Additional alternative configurations can depend on the user preferences and/or Operator/Service Provider local policy.

5.3.4 ALG for standard protocols support

Also a stateful firewall is not effective or could limit specific services with applications that include IP addresses and TCP/UDP port information in the payload (e.g. FTP, SIP protocols, peer to peer applications). To filter these protocols, and at the same time permit the access to such services, the firewall has to be augmented by specific Application Level Gateway.

The firewall should contain support for the NGN standard protocols, such as SIP and RTSP, for the pinholing of the media ports so that the inbound and outbound traffic could flow through the CNG.

Note that when B2BUA is implemented inside the CNG, TS 185 003 [3], it acts as a SIP ALG; in this case the B2BUA shall interact with the firewall.

5.3.5 Firewall management

The firewall should be manageable from the CPN and by the IPS/Operator and it should enable the ISP/Operator also to upgrade the firewall functionality via download of a new configuration file. To implement this operation, the management centre downloads to the CNG firewall the configuration file. This file integrates the basic firewall configuration that includes the HIGH and LOW configurations. As described by HGI in [i.9], DSL Forum TR-069 [i.6] provides mechanisms for configuration file downloads. The TISPAN CPN firewall should support TR-069 [i.6]. However, some additional mechanisms and specifications could be needed to fully support the CPN security requirements, for example OMA device management which supports also the security of the management.

The management features should permit the upgrade of the software firewall, the management of the security policy and the access to the logging information.

5.3.6 Logging

The firewall should have the ability to log network traffic and main security events. Basic logging options should be supported (by default all logging options should be disabled). The logging function should capture at least the following events:

- Log of changes to firewall policy.
- Network connection logs, which include dropped and rejected connections (for both inbound and outbound packets).
- Log of software firewall upgrade events.

The log files should be accessible from the remote management.

6 SP and/or CP secure upgrade

6.1 SP and/or CP secure upgrade: introduction and scope

6.1.1 Introduction

Interoperability of the CPE (IPTV CND or CNG) means that the end user can switch the CPE to another IPTV service provider without having to change the CPE (assuming that the transmission technology does not change).

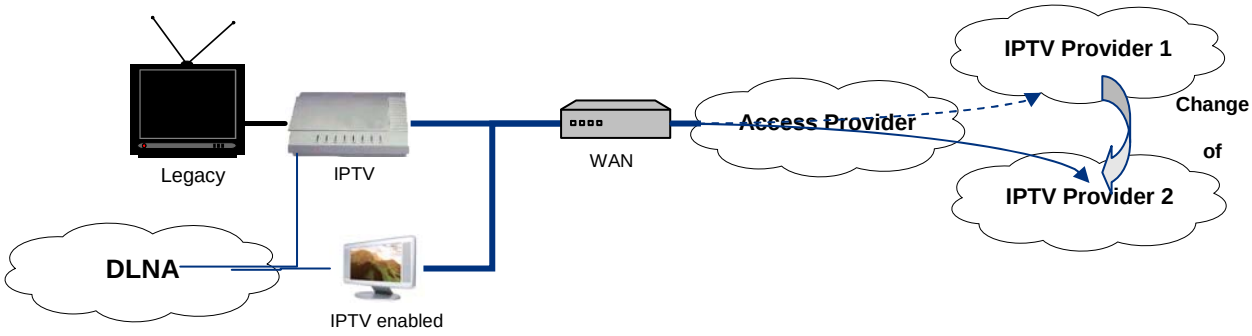


Figure 2: SP & CP architecture

An IPTV specific requirement which has a strong influence on interoperability is the use of service and/or content protection (SP/CP) systems aka CA/DRM systems.

Service Protection/Content Protection Interoperability (in short SP/CP Interoperability) of CPE with a IPTV Service Providers offering means that an end user can switch to another Service Provider (using a different SP/CP system) to obtain service from whilst retaining his CPE equipment.

In order for CPE not to have to implement every (possibly proprietary) SP/CP system that exists, for such kind of interoperability CPE is required to support upgrade/renewal of its SP/CP software and potentially the cryptographic algorithms used by the SP/CP system for scrambling content.

NOTE 1: This type of interoperability does not solve the reuse of previously bought protected content, when the SP/CP system changes. Frameworks like DVB-CPCM (see all parts of TS 102 825 [i.4]) solve that inter CP interoperability problem on the CPE side, this is also an important topic but it is complementary to the CPE - service provider interoperability addressed in this clause.

NOTE 2: Once a SP/CP software is upgraded and taken into operation there is no further dependency on the SP/CP upgrade mechanism.

The requirements that the secure SP/CP upgrade mechanism needs to comply with are specified in TS 187 001 [1], clause 4.19.5. The subsystem in CND that implements the CPN side of the mechanism is specified in TS 185 006 [2].

6.1.2 Scope

The scope of this clause is to describe a solution that enables "Secure upgrade of Service Protection and/or Content Protection" software. The solution has to be generalisable to other types of software.

The following will be covered to specify a solution to "Secure upgrade of Service Protection and/or Content Protection":

- stakeholder model;
- trust hierarchy model;
- upgrade use cases;
- functional architecture;
- threats;
- security architecture;
- upgrade protocol details.

The solution for "Secure upgrade of Service Protection and/or Content Protection software" will have to cover different aspects to achieve a trusted platform for Firmware, service protection and/or content protection, or other software. These aspects are:

- Secure loading of software by the CND. This enables secure boot time and runtime loading of software, to achieve a trusted environment for SP and/or CP or other software at all times.
- Secure upgrade of software on the CND. This enables secure upgrade of software, to achieve a trusted environment for SP and/or CP or other software.

6.2 SP and/or CP secure upgrade: architecture

6.2.1 SP and/or CP upgrade stakeholders

The following stakeholders in the CND upgrade process have been identified:

CND Custodian: A CND custodian would act as a trust authority for CND's and certify all public device keys of compliant devices. With them a service provider can verify that a connected device is authorized to download a new SP and/or CP implementation. A CND provider or chip(-set) provider can obtain public-private key pairs from the Custodian for use in production of certified implementations of the CND upgrade system, the security of this process shall be under contractual agreements between the custodian and the CND provider.

Internet Service Provider Custodian: An ISP custodian would act as a trust authority for ISP's certify all public device keys of trusted Internet Service Providers. With them a CND can verify that an Internet Service Provider is authentic and authorized to download new CND software. Once connected to the Internet Service Provider, a CND obtains Internet Service Provider's public key with a valid certificate from that Custodian and can proof it.

IPTV Service Provider Custodian: An IPTV Service Provider custodian would act as a trust authority for IPTV Service Providers and certify all public device keys of trusted IPTV Service Providers. With them a CND can verify that an IPTV Service Provider is authentic and authorized to download a new CND implementation. Once connected to the IPTV Service Provider, a CND obtains IPTV Service Provider's public key with a valid certificate from that Custodian and can proof it.

SP/CP Custodian: An SP/CP custodian would act as a trust authority for SP providers and CP providers and certify all public keys used during the SP or CP software upgrades for 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 CP provider will obtain certified public keys needed for the validation of the CND.

Internet Service Provider: The Internet Service Provider is the actual provider of the IP connectivity service that a user wants to use. The Internet Service Provider can use remote management interface with the connected CND that it is entitled to manage in order to upgrade the CND software or security credentials on the CND side in order to match the network side of the IP connectivity service offering.

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

SP Provider: 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 CND in order to interoperate with the IPTV Service Providers service offering.

CP Provider: 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 CND in order to interoperate with the IPTV Service Providers service offering.

CND Manufacturer: Fabricates the equipment that complies with the CND upgrade mechanism.

Chip Manufacturer: Fabricates the chip-set that complies with the CND upgrade mechanism.

User: The consumer of the service that the Service Provider offers. Normally the user is not aware of the upgrade mechanism; this should normally not need any user interaction. The only case where user consent might be needed is the Service Provider handover case, where the user needs to confirm that it is OK that a second Service Provider gets entitled to perform software upgrades in order to allow the CND to be used with this second Service Provider.

Figure 3 shows the an example case when the CND upgrade is for Service Protection and/or Content protection (SP/CP).

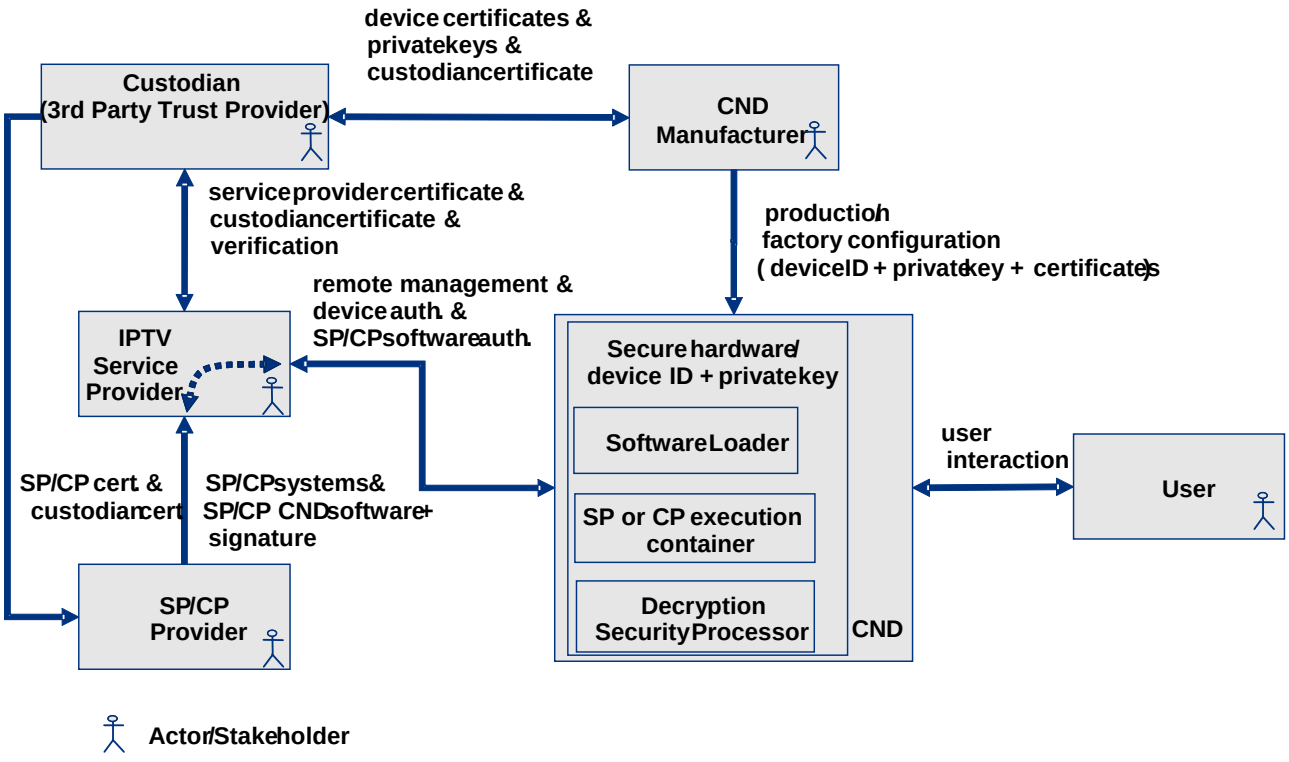


Figure 3: CND upgrade for SP CP example

Other concepts depicted in figure 3:

Secure hardware with device ID and private key: Secure hardware implementation of the software upgrade system in the CND which contains 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: A build-in bootstrap loader that loads the SP or CP system software and additional decryption algorithms from the service provider at first startup or during upgrade.

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

Security processor: An on-chip security processor that has some build-in hardwired or software decryption algorithms.

6.2.1a CND secure upgrade trust hierarchy

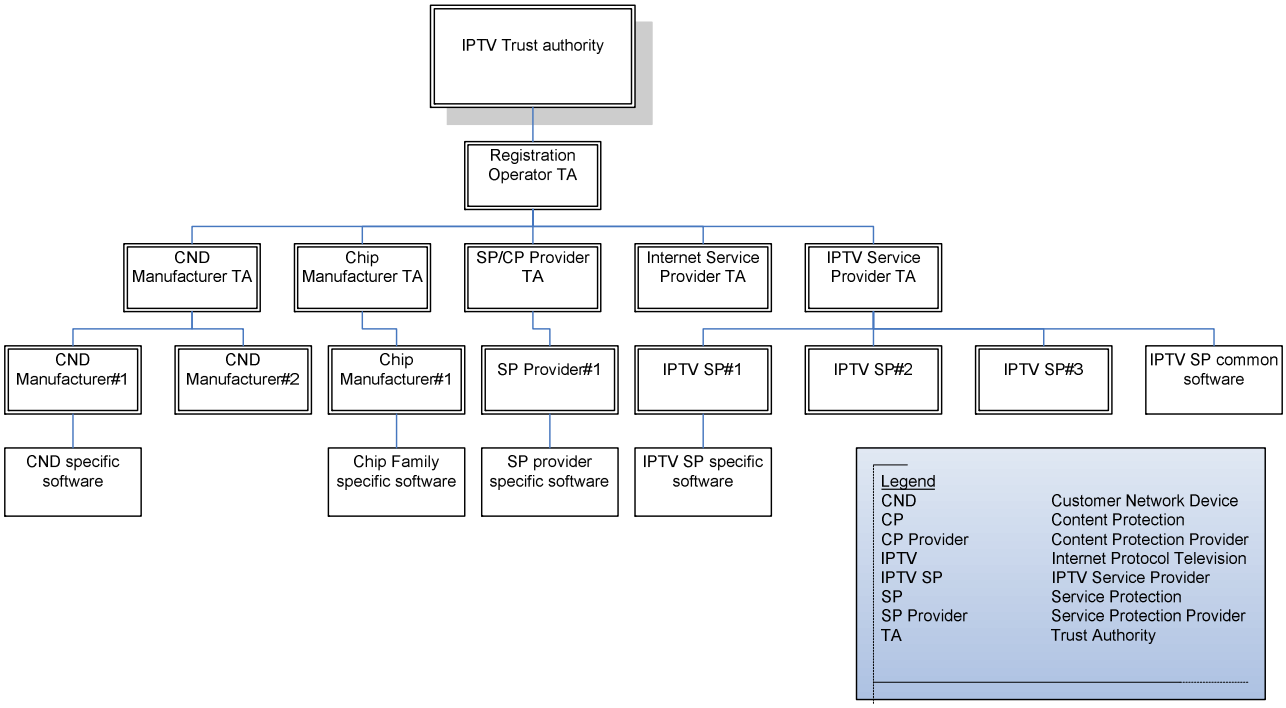


Figure 4: X CND secure upgrade trust hierarchy

6.2.1a.1 IPTV trust authority

This is root of trust that will ensure that only an authorised Registration Operator can add new trust authorities for:

- Internet Service Provider.
- IPTV Service Provider.
- SP provider or CP Provider.
- CND 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.

6.2.1a.2 Registration operator trust authority

The role of the Registration Operator is day to day adding and removing Internet Service Provider, IPTV Service Provider, SP provider, CP Provider, CND Manufacturer and Chip-set Manufacturer trust authorities by registering and deregistering them and providing the necessary credentials to communicate securely with the CND and subscribers.

While the number of IPTV Service Providers and CND providers may be small, the number and amount of change in common applications may eventually be very large, so it is important to isolate the higher Trust Authorities from day to day operational role.

6.2.1a.3 ISP trust authority

The service may be dependent on agreement with certain ISP's to provide, say, a minimum level of QoS which requires a secure distribution of configuration parameters to the CND and possibly IPTV Service Provider. This branch provides ISP and CND specific security credentials for this configuration

6.2.1a.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 CND software and applications is ensured at:

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

This branch provides IPTV Service Provider specific security credentials to ensure this confidentiality and integrity.

6.2.1a.5 SP/CP 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 ensured at:

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

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

6.2.1a.6 CND trust authority

Many aspects of end to end security are critically dependent on ensuring that the integrity of the CND is ensured at:

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

This branch provides CND specific security credentials to ensure this integrity. For example, CND specific Boot, Kernel™ and Root File system access keys.

6.2.1a.7 Chip manufacturer trust authority

Many aspects of end to end security are critically dependent on ensuring that the integrity of the processor chip or security chips. This branch provides chip manufacturer specific security credentials to ensure this integrity. For example SOC specific identities and access keys.

6.2.1a.8 IPTV service provider specific trusted platform software and applications

There will be applications that are specific to a particular IPTV Service Provider.

6.2.1a.9 IPTV service provider 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 need to be restricted to certain IPTV Service Providers only. These would include standardised VOD applications, etc.

6.2.2 SP and/or CP upgrade architecture

6.2.2.1 Overview

Figure 5 is a depiction of a CND architecture that shows the typical functions needed in an IPTV CND that supports consumption of protected linear TV content and allows upgrade of the SP and/or CP or other software to enable change of IPTV service provider.

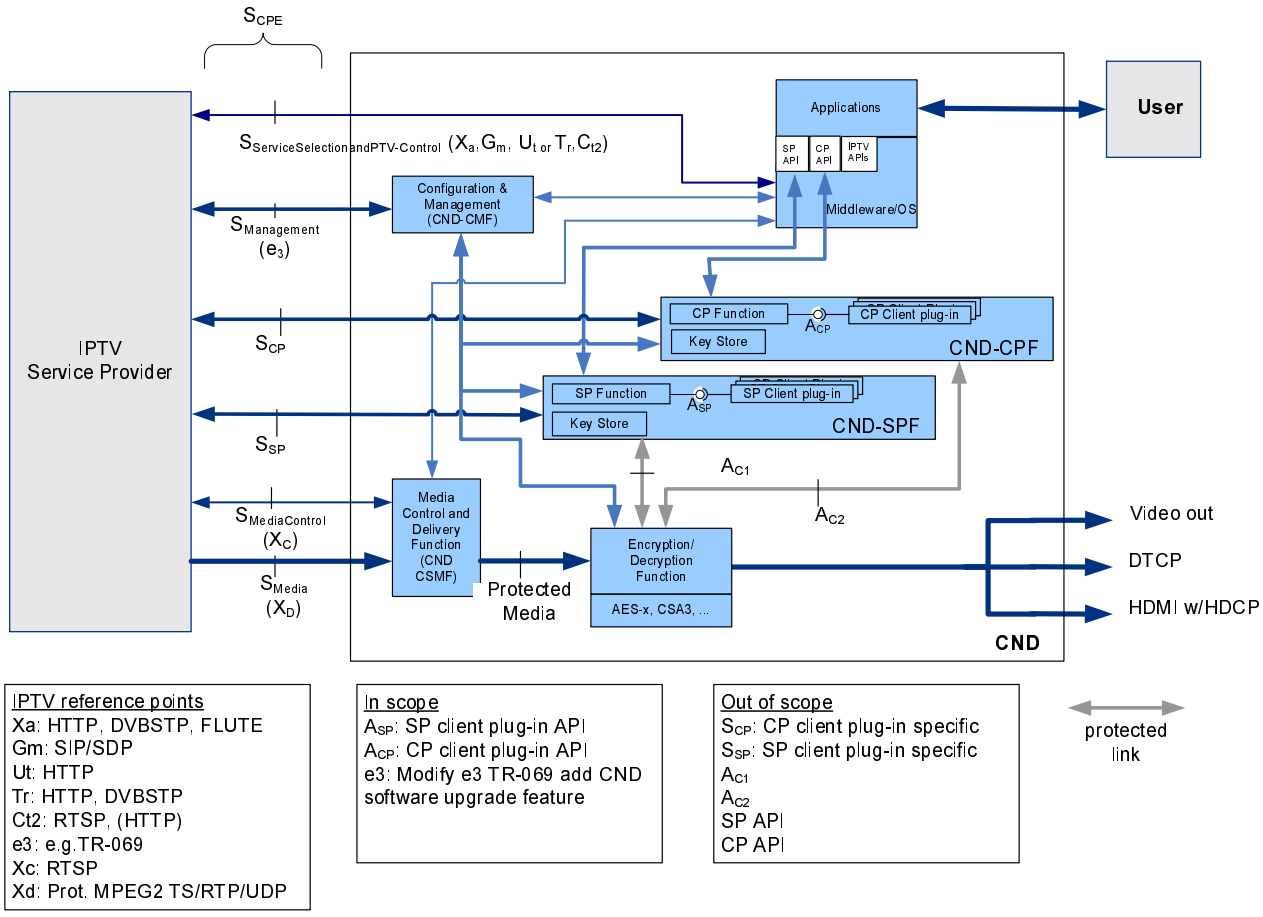


Figure 5: CND functional architecture for SP CP upgrade

6.2.2.2 Functional entities

Configuration & management (CND-CMF): This function is responsible for, and allows, the remote configuration and management of the CND. CND-CMF is described in TS 185 006 [2].

Middleware/OS: This is the Firmware that operates the CND and allows the CND to be used as an IPTV CND. The Firmware also interacts with the CND-SPF and the CND-CPF.

CND-SPF: This is the functional entity in the CND enclosing the SP Function, the SP Client Plug-in and the corresponding key store. A secure environment is part of it ensuring data integrity, authenticity and confidentiality of data stored and exchanged. The CND-SPF function interacts with the Encryption/Decryption Function to control the encryption or decryption resources and provides the keys needed to perform encryption and/or decryption.

CND-CPF: This is the functional entity in the CND enclosing the CP Function, the CP Client Plug-in and the corresponding key store. A secure environment is part of it ensuring data integrity, authenticity and confidentiality of data stored and exchanged. The CND-CPF function interacts with the Encryption/Decryption Function to control the encryption or decryption resources and provides the keys needed to perform encryption and/or decryption.

SP function: This function is responsible for service protection. For this it interacts with the Middleware that instructs the service protection mechanism to start enable processing of protected media in order to allow it to be rendered. The SP function has then to instantiate and start the correct SP Client plug-in, which contains the actual SP software.

CP function: This function is responsible for content protection. For this it interacts with the Middleware that instructs the content protection mechanism to start enable processing of protected media in order to allow it to be rendered. The CP function has then to instantiate and start the correct CP Client plug-in, which contains the actual CP software.

SP client plugin: This is the component that contains the actual SP software. This can be a standardized or proprietary service protection system. For this it interacts with the IPTV Service Providers systems using the SP specific protocol messages to exchange the keys needed to decrypt the protected media. The SP client plug-in communicates with the rest of the CND using the API offered by the SP Function.

CP client plugin: This is the component that contains the actual CP software. This can be a standardized or proprietary content protection system. For this it interacts with the IPTV Service Providers systems using the CP specific protocol messages to exchange the keys needed to decrypt the protected media. The CP client plug-in communicates with the rest of the CND using the API offered by the CP Function.

Key store: This function provides storage for keys so that they can only be accessed by subsystems that are entitled to. For example when SP Client Plug-in 1 stores specific keys for later reuse, then only SP Client 1 can access them and no other subsystem or software.

Media Control and Delivery (CND-CSMF): This function is responsible for the provision for the resources dealing with the reception of unicast or multicast media streams from the IPTV service provider. The resources are allocated/configured on request of the Middleware/OS. Received media will be sent to the Encryption/Decryption Function for further processing.

Applications: Applications are additional pieces of software that provide additional value but are not necessary to operate the device. Examples are widgets, web 2.0 applications , etc. these applications may have access to the CND-SPF and/or CND-CPF function through a JavaScript API that is provided by the Middleware/OS.

Encryption/Decryption function: This function is responsible for the decryption of protected media streams based on the cipher that is used to encrypt the media stream. Reservation of the correct cipher resources and the provision of control words or traffic keys that are needed to perform a successful decryption of cipher text is the responsibility of the CND-SPF or CND-CPF.

6.2.2.3 Affected interfaces and reference points

e3: The e3 interface's primitives for Firmware upgrade need to be extended with an indication that the provided software is a CND software upgrade. Also the proper authentication needs to be in place on this interface to ensure that only entitled parties may perform a remote CND software update.

A_{SP}: The plugin API for use by the SP Function and its callback API on the SP Function, that allows the dedicated SP client to interact with the rest of the CND.

A_{CP}: The plugin API for use by the CP Function and its callback API on the CP Function, that allows the dedicated CP client to interact with the rest of the CND.

S_{SP}: The reference point representing the interaction between the CND-SPF and an IPTV service provider's SP system. Note that the protocols over this reference point are SP Client specific (proprietary or open standards) and hence are not further specified here.

S_{CP}: The reference point representing the interaction between an CND-CPF and an IPTV service provider's CP system. Note that the protocols over this reference point are CP Client specific (proprietary or open standards) and hence are not further specified here.

SP API: API for giving Applications control over the CND-SPF.

CP API: API for giving Applications control over the CND-CPF.

A_{C1}: The A_{C1} reference point gives the CND-SPF control over the Encryption/Decryption Functions resources. As security sensitive information like traffic keys are passed over this reference point, the messages exchanged over this reference point must be confidentiality and integrity protected.

A_{C2}: The A_{C2} reference point gives the CND-CPF control over the Encryption/Decryption Functions resources. As security sensitive information like traffic keys are passed over this reference point, the messages exchanged over this reference point must be confidentiality and integrity protected.

6.2.3 SP and/or CP upgrade use cases

6.2.3.1 General

Firmware owner: In the security architecture described in clause 6.2.4 it is assumed that one party is entitled to upgrade the firmware, this party is called the firmware owner. This party can be modified in a controlled way if allowed by a custodian or trust provider.

SP owner: In the security architecture described in clause 6.2.4 it is assumed that one party is entitled to upgrade SP Software Module, this party is called the SP owner. This party can be modified in a controlled way if allowed by a custodian or trust provider.

CP owner: In the security architecture described in clause 6.2.4 it is assumed that one party is entitled to upgrade CP Software Module, this party is called the CP owner. This party can be modified in a controlled way if allowed by a custodian or trust provider.

The upgrade use cases will be described in terms of firmware owner, SP owner and CP owner as 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 where the service provider is the firmware owner and controls the complete IPTV CND, in other cases this responsibility is outsourced to a security provider (SP provider or CP provider) or CND provider. The same observation can be made for SP owner or CP owner, sometimes this is completely in hands of the IPTV service provider in other cases the responsibility is outsourced to a security provider.

The basic entity to be managed in the process of a remote software upgrade (software other than the firmware) is called a Software Module. This term is taken from [5], it refers to a general piece of software that can be installed on a CND or CNG. The conceptual framework for management of software modules in various CPN devices is described in Appendix II "Software Module Management" of [6], it provides the capability of remote installation, deinstallation and upgrade and starting and stopping of Software Modules in named Execution Environments within specific CPN devices. It does not make assumptions on the specific technology of the Execution Environments, typical examples include Linux™, OSGi, .NET and Java ME.

As the SP and/or CP upgrade is in fact a software upgrade albeit with stringent security requirements it is assumed here that SP software can be provisioned by packaging it as a SP Software Module and the CP software can be provisioned by packaging it as a CP Software Module.

NOTE: The ownership property that is pivotal to the security of the solution specified here can easily be generalised to other types of software, in general one can then speak about a Software Module Owner.

6.2.3.2 User changes service provider

When a user changes from IPTV service provider the following steps may be needed before the user can enjoy the services of the IPTV service provider:

1) Upgrade of Firmware, in case the new service provider requires ownership and use of a dedicated Firmware.

2) Upgrade of the SP Software Module, to adapt the IPTV CND to the SP system used by the IPTV Service Provider.

3) Upgrade of the CP Software Module, to adapt the IPTV CND 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 CND the service provider needs to request ownership of the Firmware, the security solution described in clause 6.2.4 implies that the service provider in this case first needs to provide a secondary bootloader signed by the trust provider and a public key certificate key also signed by a trust provider.

Therefore upgrade of the Firmware in case of change of service provider might include 2 steps:

1a) Service Provider requests to be Firmware owner.

1b) Service Provider requests upgrade of Firmware.

The individual sub steps 1a) and 1b) are described in further detail in clauses 6.2.3.3 and 6.2.3.4 respectively.

To allow the service provider to upgrade the SP Software Module or provide the SP Software Module to be downloaded by the IPTV CND the service provider needs to request ownership of the SP Software Module, the security solution described in clause 6.2.4 implies that the service provider in this case first needs to provide an SP loader signed by the trust provider and a public key certificate key also signed by a trust provider.

Therefore upgrade of the Firmware in case of change of service provider might include 2 steps:

2a) Service Provider requests to be SP owner.

2b) Service Provider requests upgrade of SP Software Module.

If other stakeholders need to take control of the Firmware or the SP Software Module, the steps taken are the same.

The individual sub steps 2a) and 2b) are described in further detail in clauses 6.2.3.5 and 6.2.3.6 respectively.

To allow the service provider to upgrade the CP Software Module or provide the CP Software Module to be downloaded by the IPTV CND the service provider needs to request ownership of the CP Software Module, the security solution described in clause 6.2.4 implies that the service provider in this case first needs to provide an CP loader signed by the trust provider and a public key certificate key also signed by a trust provider.

Therefore upgrade of the Firmware in case of change of service provider might include 2 steps:

3a) Service Provider requests to be CP owner.

3b) Service Provider requests upgrade of CP Software Module.

If other stakeholders need to take control of the Firmware or the CP Software Module, the steps taken are the same.

The individual sub steps 3a) and 3b) are described in further detail in clauses 6.2.3.7 and 6.2.3.8 respectively.

6.2.3.3 A stakeholder X requests to be firmware owner

The following steps will effectively change the Firmware owner:

1) The stakeholders X's ACS initiate a remote management connection with the IPTV CND.

2) During connection establishment mutual authentication is performed between the requesting stakeholder X's ACS and the IPTV CND.

NOTE 1: The methods for mutual authentication between the IPTV CND and the ACS could be, for example, the methods specified in TS 183 065 [i.5] (TR-069 [i.6]). Protocols other than TR-069 [i.6] could be used.

3) The stakeholders ACS instruct the IPTV CND to download the secondary bootloader package.

4) Before initiating such download the IPTV CND asks the user if the stakeholder X is allowed to take over the IPTV CND.

5) If yes, the IPTV CND will perform the requested download of the secondary bootloader package, which contains the secondary bootloader image, a public key of the stakeholder X, a signature from the trust provider and a public key certificate.

6) Only when the secondary bootloader can be verified to be authentic (see clause 6.2.4), the secondary bootloader is allowed to be installed into the IPTV CND.

NOTE 2: The methods for initiating the CND to download a Software Module may be done, for example, according to TS 183 065 [i.5] (TR-069 [i.6] endorsement) Download method. Protocols other than TR-069 [i.6] could be used to carry data to the IPTV CND. TR-069 [i.6] might need an extra value for the filetype field to indicate "bootloader2". This needs to be detailed in the protocol clause.

Only when 6) is successfully completed, the new secondary bootloader allows loading of Firmware images that are signed by stakeholder X.

6.2.3.4 Firmware owner requests upgrade of firmware

The following steps will upgrade the Firmware:

1) The Firmware owner's ACS initiates a remote management connection with the IPTV CND.

2) During connection establishment mutual authentication is performed between the requesting Firmware owner's ACS and the IPTV CND.

NOTE 1: The methods for mutual authentication between the IPTV CND and the ACS could be, for example, the methods specified in TS 183 065 [i.5] (TR-069 [i.6]). Protocols other than TR-069 [i.6] could be used.

3) The Firmware owner's ACS instructs the IPTV CND to download the Firmware.

4) The IPTV CND 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 to be authentic (see clause 6.2.4), the Firmware is allowed to be installed into the IPTV CND.

NOTE 2: The methods for initiating the CND to download a Software Module may be done, for example, according to TS 183 065 [i.5] (TR-069 [i.6] endorsement) download method. Protocols other than TR-069 [i.6] could be used to carry data to the IPTV CND. TR-069 [i.6] might need an extra value for the filetype field to indicate "bootloader2". This needs to be detailed in the protocol clause.

Only when 5) is successfully completed, the new Firmware will be started by the CND on the next reboot.

6.2.3.5 A stakeholder Y requests to be SP owner

The following steps will effectively change the SP owner:

1) The stakeholders Y's ACS initiates a remote management connection with the IPTV CND.

2) During connection establishment mutual authentication is performed between the requesting stakeholder Y's ACS and the IPTV CND.

NOTE 1: The methods for mutual authentication between the IPTV CND and the ACS could be, for example, the methods specified in TS 183 065 [i.5] (TR-069 [i.6]). Protocols other than TR-069 [i.6] could be used.

3) The stakeholders ACS instruct the IPTV CND to download the SP loader package.

4) Before initiating such download the IPTV CND asks the user if the stakeholder Y is allowed to install modules for consumption of IPTV.

5) If yes, the IPTV CND will perform the requested download of the secondary bootloader package which contains the SP loader image, a public key of the stakeholder Y, a signature from the trust provider and a public key certificate.

6) Only when the SP loader can be verified to be authentic (see clause 6.2.4), the secondary bootloader is allowed to be installed into the IPTV CND.

NOTE 2: The methods for initiating the CND to download a Software Module may be done, for example, according to TS 183 065 [i.5] (TR-069 [i.6] endorsement) download method. Protocols other than TR-069 [i.6] could be used to carry data to the IPTV CND. TR-069 [i.6] might need an extra value for the filetype field to indicate "SP loader". This needs to be detailed in the protocol clause.

Only when 6) is successfully completed, the new SP loader allows loading of SP images that are signed by stakeholder Y.

6.2.3.6 SP owner requests upgrade of SP software module

Following steps will upgrade the SP Software Module:

1) The Firmware owner's ACS initiates a remote management connection with the IPTV CND.

2) During connection establishment mutual authentication is performed between the requesting Firmware owner's ACS and the IPTV CND.

NOTE 1: The methods for mutual authentication between the IPTV CND and the ACS could be, for example, the methods specified in TS 183 065 [i.5] (TR-069 [i.6]). Protocols other than TR-069 [i.6] could be used.

3) The Firmware owner's ACS instructs the IPTV CND to download the SP Software Module.

4) The IPTV CND will perform the requested download of the SP Software Module, which contains the SP image and a signature from the SP owner.

5) Only when the SP Software Module can be verified to be authentic (see clause 6.2.4), the SP Software Module is allowed to be installed into the IPTV CND.

NOTE 2: The methods for initiating the CND to download Software Module could be done, by example, according to TS 183 065 [i.5] (TR-069 [i.6] endorsement) download method. Protocols other than TR-069 [i.6] could be used to carry data to the IPTV CND. TR-069 [i.6] might need an extra value for the filetype field to indicate "bootloader2". This needs to be detailed in the protocol clause.

Only when 5) is successfully completed, the new SP Software Module will be started by the IPTV CND.

6.2.3.7 A stakeholder Y requests to be CP owner

Following steps will effectively change the CP owner:

1) The stakeholders Y's ACS initiates a remote management connection with the IPTV CND.

2) During connection establishment mutual authentication is performed between the requesting stakeholder Y's ACS and the IPTV CND.

NOTE 1: The methods for mutual authentication between the IPTV CND and the ACS could be, for example, the methods specified in TS 183 065 [i.5] (TR-069 [i.6]). Protocols other than TR-069 [i.6] could be used.

3) The stakeholders ACS instruct the IPTV CND to download the CP loader package.

4) Before initiating such download the IPTV CND asks the user if the stakeholder Y is allowed to install modules for consumption of IPTV.

5) If yes, the IPTV CND will perform the requested download of the secondary bootloader package, which contains the CP loader image, a public key of the stakeholder Y, a signature from the trust provider and a public key certificate.

6) Only when the CP loader can be verified to be authentic (see clause 6.2.4), the secondary bootloader is allowed to be installed into the IPTV CND.

NOTE 2: The methods for initiating the CND to download a Software Module may be done, for example, according to TS 183 065 [i.5] (TR-069 [i.6] endorsement) download method. Protocols other than TR-069 [i.6] could be used to carry data to the IPTV CND. TR-069 [i.6] might need an extra value for the filetype field to indicate "CP loader". This needs to be detailed in the protocol clause.

Only when 6) is successfully completed, the new CP loader allows loading of SP images that are signed by stakeholder Y.

6.2.3.8 CP owner requests upgrade of CP software module

Following steps will upgrade the CP Software Module:

1) The Firmware owner's ACS initiates a remote management connection with the IPTV CND.

2) During connection establishment mutual authentication is performed between the requesting Firmware owner's ACS and the IPTV CND.

NOTE 1: The methods for mutual authentication between the IPTV CND and the ACS could be, for example, the methods specified in TS 183 065 [i.5] (TR-069 [i.6]). Protocols other than TR-069 [i.6] could be used.

3) The Firmware owner's ACS instructs the IPTV CND to download the CP Software Module.

4) The IPTV CND will perform the requested download of the CP Software Module, which contains the CP image and a signature from the CP owner.

5) Only when the CP Software Module can be verified to be authentic (see clause 6.2.4), the Firmware is allowed to be installed into the IPTV CND.

NOTE 2: The methods for initiating the CND to download a Software Module could be done, by example, according to TS 183 065 [i.5] (TR-069 [i.6] endorsement) download method. Protocols other than TR-069 [i.6] could be used to carry data to the IPTV CND. TR-069 [i.6] might need an extra value for the filetype field to indicate "bootloader2". This needs to be detailed in the protocol clause.

Only when 5) is successfully completed, the new CP Software Module will be started by the IPTV CND.

6.2.4 SP and/or CP upgrade security architecture

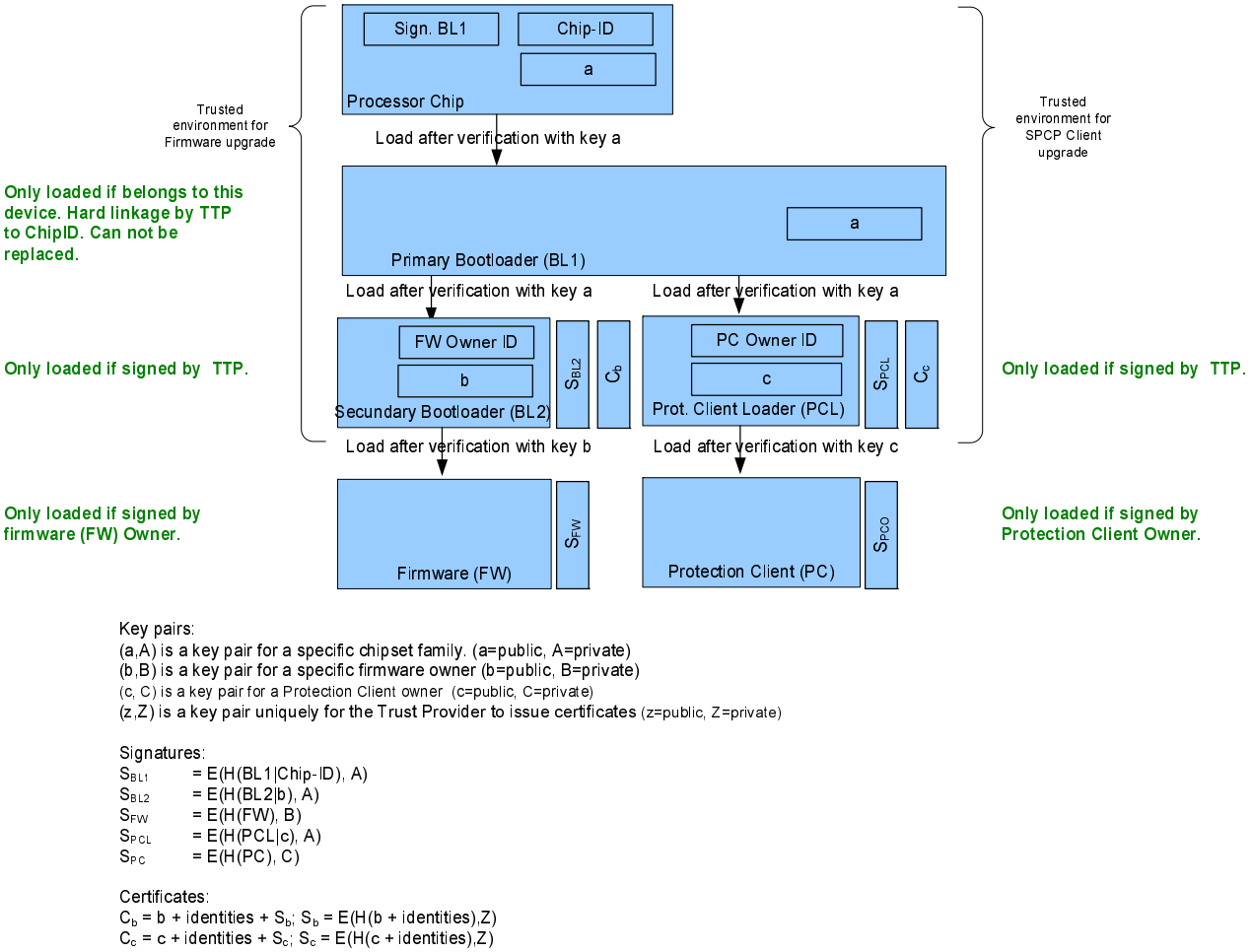
6.2.4.1 Trusted environment architecture for SP/CP

6.2.4.1.1 Hardware supported trusted environment preventing Hi-Jacking

To ensure that an SP or CP client is authentic and secure, a CND needs to provide a trusted environment. For this, all layers of loaded software need to be verified to be integer and authentic. In a hardware supported trusted environment the guarantee is provided by the lowest level (i.e. the hardware) 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 to be integer and authentic.

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

This multi layered boot process is shown in figure 6:



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 6: Multi layered boot process

CND Processor Chip: The CND processor chip supporting the trusted environment for SP and/or CP Client software provides facilities that enable loading of known integer and authentic primary bootloader software. At 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 integer and authentic. 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, only the trust provider 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 burned in signature S_{BL1} using public key *a* and comparing both results, if and only if both results are the same the image is considered integer and authentic and only then this image may be loaded. The creation of a signature over a concatenation of the primary Boot Loader image and the Chip-ID and burning this signature into the chip ensures that no other primary bootloader can ever be loaded by this device then the original one. This ensures that the first level of software loaded is always known and trusted.

In short 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 an 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 that is stored in the device and *A* is the private key that is stored and only known 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 startup load BL1 from ROM.
- Verify that H(BL1|Chip-ID) = D(S_{BL1},*a*).
- If verification step succeeds then start BL1.

What is achieved:

- The primary bootloader is guaranteed to always be the one 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 CND that contains the processor chip.

The **primary bootloader:** When the processor chip starts the primary bootloader, the primary bootloader has the responsibility to:

- load or download the secondary bootloader image (BL2), a public key *b*, a signature S_{BL2} and a certificate for public key *b*; start BL2, if and only if the certificate for the public key *b* is valid and the concatenation of image BL2 and key *b* can be proven integer and authentic using public key *a*; and
- load or download one or more Protection Client Loader image(s) (PCL), a public key *c*, a signature S_{PCL} and a certificate for public key *c*; start PCL, if and only if the certificate for the public key *c* is valid and the concatenation of image PCL and key *c* can be proven integer and authentic using public key *a*.

The signatures $S_{BL2 \text{ and }} S_{PCL}$ have both been signed with the private key *A* by the trusted provider. This private key is in possession of the trust provider, only the trust provider can sign with this private key.

The verification of image BL2 is performed by creating a hash over BL2 concatenated with the public key *b*, decrypting the loaded/downloaded signature S_{BL2} using key *a* and comparing both results, if and only if both results are the same the image is considered integer and authentic and only then this image may be started by BL1. The signature over a concatenation of BL2 and the public key *b* means that the trust provider states that the linkage of BL2 with the owner of public key *b* is authentic. The certificate C_b authenticates public key *b* to belong to the identities as 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 CND manufacturer or in case 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 the 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 the image is considered integer and authentic and only then this image may be started by BL1. The signature over a concatenation of PCL and the public key *c* means that the trust provider states that the linkage of PCL with the owner of public key *c* is authentic. The certificate C_c authenticates public key *c* to belong to the identities as 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 case 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 that is delivered with the secondary bootloader and *B* is the private key that is stored by and only known to the Firmware owner.

- (*c, C*) is a key pair for a specific Protection Client owner. Where *c* is the public key that is delivered with the secondary bootloader and *C* is the private key that is stored by and only known to the Protection Client owner.
- (*z,Z*) is a key pair uniquely for the Trust Provider to issue public key certificates. Where *z* is the public key that can of the trust provider and *Z* is the private key that is stored by and only known 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 verification steps succeeded 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 verification steps succeeded then start PCL.

What is achieved:

- The secondary bootloader and protection client loader(s) are guaranteed to be authentic and to belong to a particular stakeholder it is linked with, this is guaranteed by the trust provider having signed this linkage. Further 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, doing so would make the CND dysfunctional.

The **secondary bootloader** is used to load or download, verify and start the Firmware belonging to a Firmware owner. The secondary boot loader verifies that a Firmware image (FW) has been signed by the Firmware owner (using private key *B*), if and only if this verification succeeds the FW will be started. The 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 verification steps succeeded then start FW.

What is achieved:

- The Firmware is guaranteed to be integer, authentic as signed by the Firmware owner.
- There is no way that a hacker can modify/replace the Firmware for this chip family with a Firmware that has not been signed by the Firmware owner, doing so would make the CND dysfunctional.
- This setup only allows the Firmware owner to replace or upgrade the Firmware. Typically this is the IPTV CND manufacturer or in case the device is owned by an IPTV service provider, the IPTV Service Provider.

NOTE: In practice the hash and the signatures are calculated over blocks of the Firmware and not over the entire Firmware image.

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 succeeds the protection client will be started. The 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 verification steps succeeded then start PC.

What is achieved:

- The protection client is guaranteed to be integer, authentic 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 CND dysfunctional.
- This setup 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 in case the IPTV Service Provider has outsourced this responsibility to a security provider, the SP or CP provider.

6.2.4.1.2 Hardware supported trusted environment, protecting the key flow

NOTE: The traffic encryption keys (also Control Words (CW)) that are transported between the CND-SPF and/or the CND-CPF and the decryption function needs to be protected against eavesdropping. It is needed to consider whether this requires the decryption function (in case implemented in software) to be part of the SP Client or CP Client or the protection client loader. Also the case where the decryption function is implemented in hardware needs to be looked at.

6.3 SPCP secure upgrade: implementation details

Clause 6.2.1a.6 sates that many aspects of end to end security are critically dependent on ensuring the integrity and authenticity of the CND at:

1) Each and every boot.
2) During run time of the CND operating system and run time of CND specific applications.
3) Before and after any remote upgrade to such software.

The stage 3 implementation details can be very much dependent on the specifics of the chipset used in the CND.

For some common types of chipset architectures, examples of the stage 3 protocols for the integrity and authenticity mechanisms to support for the 3 stages of operation listed above are outlined in an informative annex B to D.

7 Network Access Control (NAC)

The Network Access Control (NAC) is a gathering of methods linked to the control of a network's access. In terms of security, these methods aim to control access to a network with policies, including pre-admission endpoint security policy checks and post-admission controls over where users and devices can go on a network and what they can do. The combined tools used are usually enforced authentication, security policies for users, management of network resources, verification tools for security updates, and directory management.

7.1 NAC: basic description

The IETF NEA [i.2] architectures have been defined to assess the "posture" of endpoint devices for the purposes of monitoring compliance to an organization's posture policy and optionally restricting access until the endpoint has been updated to satisfy the posture requirements. Posture refers to the hardware and software configuration of an endpoint and may include knowledge that software installed to protect the device (e.g. patches, anti-virus, firewall, host-based intrusion detection system or any custom software) is enabled and up-to-date.

In the CPN context, the NEA architecture could be used to allow only compliant and trusted Customer Network Devices (CND), such as PCs, IP-phone, and PDAs, onto the network, restricting or blocking the access (at the network layer) of noncompliant devices, and thereby limiting the potential damage from security threats and risks. Then NEA allows operators and service providers to enforce specific security policies on all CND as they enter the customer premises network, regardless of their access methods, ownership, device types, application configurations, etc.

The general NEA architecture is a client-server architecture, where the server component evaluates the posture of an endpoint device and provides network authorization decisions. Moreover the NEA server interacts with the NEA client (i.e. CND) by means of a specific software agent installed on each managed element. Usually such agents have a small footprint and low impact on the CND activities. In the CPN context, the Customer Network Gateway (CNG) is the natural candidate to perform the NEA server role.

Figure 7 shows the general NAC architecture for a CPN environment. Depending on the business model and on the technologies adopted to implement the NAC services, different scenarios are possible. The main points to highlight are the following:

- CND controlled by means of a specific agent (called NAC client in the figure) able to check the local posture and enforce the policy locally. The agent could be both, installed permanently on the CND (e.g. at subscription time, the customer installs the agent on his CNDs such as PC, laptop and so on) or on demand and temporary installed by means e.g. of Java applet of other mobile code system.

- CND without any agent (permanent or temporary). Usually such devices cannot be managed by an agent because of their legacy/closed operative system (e.g. printer) or because explicitly required by the business model of the NAC service. Without an agent, it is necessary to assess the posture of the CND remotely by checking for vulnerability to attacks from the network. For example the CNG could be equipped with specific software able to look for open TCP/UDP ports, detect the Operative System type and detect applications running on a target system (i.e. the CND to be assessed). It is also necessary to define a separate enforcing point, e.g. on the CNG.

- NAC server implemented within the CPN e.g. in the CNG. In this case the NAC server is responsible for the security of the local CPN and checks the posture of the local CNDs depending on the policies locally defined by the customer or centrally defined by the Service Provider/Operator.

- The NAC server can reside inside the Operator's management network (e.g. in the NGN outside the CPN) and no specific software has to be installed on the CNG.

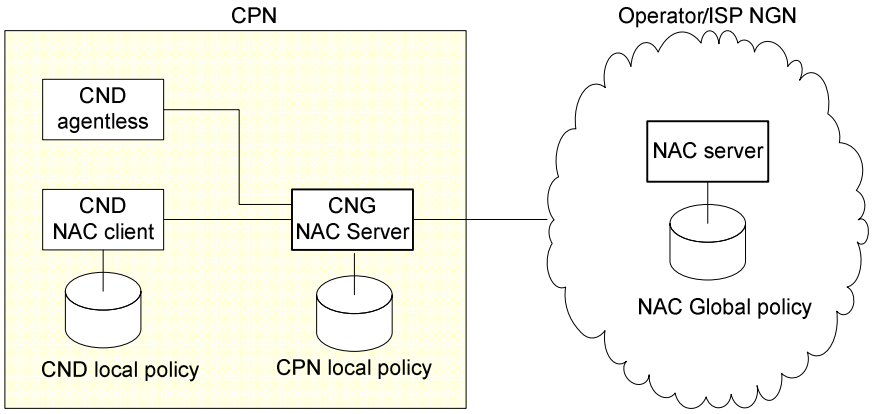


Figure 7: CPN NAC architecture

NAC (and NEA) frameworks could provide significant benefits to the security of the CPN environments, among them the following are the most relevant:

- assessment and identification of non-compliant CNDs;
- centralized security policies definition;
- monitoring of the CND's compliance over time.

In the CPN context the main issues are related to the implementation of the remediation process and the enforcement mechanisms. The implementation of a NAC service raises the expectation that some legitimate CNDs will be denied access to the CPN resources. Hence it is required a mechanism to remediate the CNDs vulnerabilities found during the assessment phase. There are at least two strategies for remediation: quarantine networks and captive portals. A quarantine network is a restricted IP network providing access only to certain hosts and applications (e.g. Antivirus server where the CND can download the latest signatures). Captive portals intercept access to web pages, redirecting users to a web application that provides instructions and tools for updating their devices. The enforcement point in the CPN can be implemented e.g. by defining specific security policy for the network firewall running in the CNG.

7.2 NAC: architecture

Figure 8 shows a sequence diagram related to a CND managed by means of a specific agent (i.e. NAC client). Such an agent could be permanent or temporary installed. Figure 8 shows the NAC server implemented outside the CPN (e.g. in the Operator's network operation centre) but the same behaviour (with minimal changes) also apply to the scenario where the NAC server is implemented in the CNG.

On step 1, the NAC process begins e.g. at CND boot time when the agent starts its assessment process. It sends a request to the NAC server in order to retrieve the latest version of the security policy defined the specific CND. This step permit the alignment of the local policy to the centrally defined ones and it is optional (e.g. because the geographic link could be down whereas the local network services are still working).

On step 2, the NAC server checks the latest version of the policy and, if an alignment is required, send to the NAC agent the new NAC policy.

After receiving the latest version of the policy (if available), the NAC agent begins to check the posture of the CND (e.g. by checking the presence of a properly configured personal firewall, up to date antivirus engine, up to date operative system patching and so on).

On step 3, depending on the posture evaluation result, the CND can finally access the CPN network resources or "quarantined" until a proper remediation action (not shown in figure 8). The security policy is enforced by the NAC client.

On step 4 the NAC client sends (optional) the results of the evaluation process to the NAC server, in order to update the centralized database with the status of the managed CND.

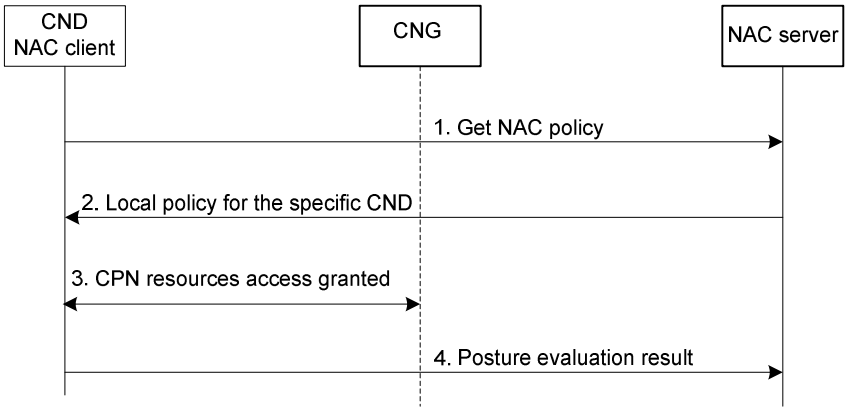


Figure 8: Agentful CND posture assessment process

Figure 9 shows a possible implementation when the CND has not an agent installed and it is not possible to use a mobile agent (e.g. Java applet). In this scenario the CNG will play a fundamental role as local policy enforcer and eventually as a local assessment device. Figure 9 describes the scenario where the CNG acts also as a local NAC server.

On step 1, the CNG/local NAC server collects the MAC/IP address of the locally connected CNDs and detects the activities performed by a CND not yet assessed.

On step 2, once the CND has been recognized as an agentless CND, the local NAC server starts a network scanning in order to assess the posture of the CND. As an alternative strategy, it is possible to use a whitelist; mechanism of authorized CNDs. Whenever the MAC address of the agentless CND has been detected, if that MAC address is contained in the whitelist, than the corresponding device is automatically allowed to access the CPN resources.

On step 3. The CPN enforces the policy decision, for example it could allow the CND to access all the local resources, otherwise it could block any connection attempt originated form that CND.

On step 4 the NAC client sends (optional) the results of the evaluation process to the global NAC server, in order to update the centralized database with the status of the managed CPN.

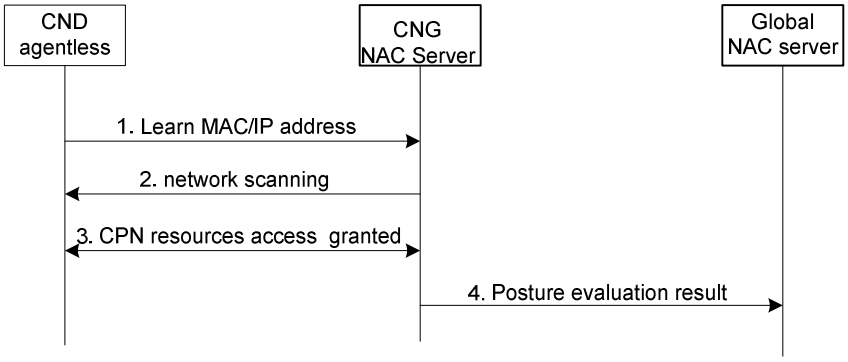


Figure 9: Agentless CND posture assessment process

8 Hosted-NAT solution for RTSP based services

Hosted Nat traversal for TISPAN IMS access is specified in ES 282 003 [i.3]. The mechanism has been defined to solve the NAT issue for the SIP based services (e.g. voice call) and it is based on an ALG collocated inside the P-CSCF and mandates the usage of the C-BGF in the access network. The C-BGF allocates and releases transport addresses according to the request coming from the ALG function of the P-CSCF. It ensures proper forwarding / binding of media packets coming from or going to the CND.

The Hosted Nat mechanism works also when the SIP packets are encapsulated inside an IPSEC tunnel.

The same (similar) mechanism can be used for the NAT traversal of the RTSP protocol (and related media flows), used in the IPTV IMS-based or in the dedicated subsystem.

8.1 Hosted-NAT for RTSP: basic description

When the RTSP protocol passes through the NAT device (e.g. CNG), the embedded address and port in the "transport" header must be translated in order for the connection to be successful (e.g. SETUP message). These issues could be managed by an ALG placed in the Operator Network (e.g. MFC in the IPTV context). The main steps involved in the NAT traversal procedure are the following:

- The ALG in the server entity (e.g. MCF) have to detect the presence of NAT comparing the client's external IP address with the source IP address in the IP header of the SETUP message (e.g. by checking the IP address in the "destination" field).
- The ALG requests the C-BGF to set up the media relay binding (IP and Port to be allocated for the media stream).
- The ALG send the (C-BGF) port information to the media server (e.g. by modifying the corresponding parameters inside the RTSP setup message) and then will forward to the CND the IP address and port that the C-BGF have been allocated for the media session.
- The CND, in order to receive the media flow (e.g. using the PLAY message) begins to send the keep-alive messages to the C-BGF port. Keep alive messages are used to punch the hole in the FW/NAT (e.g. CNG) and to aid the BGF for port binding and address mapping.

The main issues related to the Hosted Nat solution as NAT traversal mechanisms for the RTSP are:

- The mechanism could impose some constraints on the server side of the TISPAN NGN (e.g. performance).
- The CNDs need to support symmetric media in order for NAT and PAT mechanism to work on the CNG. Symmetric RTP means the device uses the same port for sending and receiving.
- The CND has to punch specific holes in the CNG in order to let the media (RTP packets) enter into the CPN by using keep-alive messages (to be defined).

The main advantages related to the Hosted Nat solution as a NAT-T mechanisms are:

- Even if the mechanism have been designed for the SIP services, it works also for other kind of services and protocols, including RTSP independently of the transport layer adopted (UDP vs. TCP).
- Minimal changes are required to the current TISPAN (r1 and r2) architecture, since it reuses the same interface used for the SIP Host Nat mechanism (e.g. Gq').
- It works also when the RTSP is protected with TLS (or IPSEC) and then it could support different level of security.
- It works in the environment with cascaded NAT routers (whereas UPnP cannot woks).

8.2 Hosted-NAT for RTSP: architecture

The sequence diagram in figure 10 describes an example of the main steps involved in the proposed Hosted Nat mechanism for RTSP. The scenario described foresee a CND (e.g. a STB) accessing the NGN IPTV dedicated subsystem. The CNG and the NAT device are collocated in the same device. In order to simplify the scenario (e.g. because the Xp reference point is not yet defined), also the MCF and MDF are collocated and seen as a single entity called Media Function (MF):

1) The CND sends the RTSP SETUP message with appropriate SDP description (or other session description mechanism) of the media request to the Media Function in order to receive the desired media channel (the RTSP server IP address could be obtained, for example, through HTTP communications - not shown in the figure). The CNG changes the sender IP address and TCP port of the RTSP packet and forwards it to the destination address (the IP address of the RTSP server, the MCF). It is worth to mention that the RTSP can be transported over TLS without any changes needed to the present solution. Anyway it is assumed that the RTSP is transported over TCP.

2) The MF detects the presence of a NAT device in the network between the CND and requests (on Gq') the allocation of specific addresses and ports on C-BGF for the RTP media flows; this information is used to update the IP and port address information in the SDP message that describes the RTP flow.

3) The MF generates a RTSP 200 OK message where the SDP contains the BGF address and port reserved for this media flow during the step 2; the RTSP message is sent back to the CND inside the TCP connection opened during the step 1 (the NAT device keeps the hole opened because the TCP is connection oriented).

4) When the IPTV customer wants to see the content, it sends the RTSP PLAY message to the MF.

5) The MF starts to send the media traffic to the address and port allocated by the C-BGF. That traffic is then discarded by the BGF because it does not yet know the actual address of the CND.

6) The CND starts sending keep alive messages that could consist of empty RTP packet with a payload type of 20 to the destination address and port contained in the 200 OK (i.e. the C-BGF); The frequency of the keep alive should also be defined; the keep-alive message opens the hole in the CNG and reaches the C-BGF.

7) The C-BGF learns the IP address and port where to send the RTP traffic from the keep-alive messages and starts to forward the media traffic to the CNG. After its usage, the C-BGF discards the keep-alive message received without forwarding it to other elements of the network (e.g. MF). Also the additional keep-alive messages received that are related to the same video session will be discarded. The NAT device finally delivers the flow to the CND by using its internal NAT table.

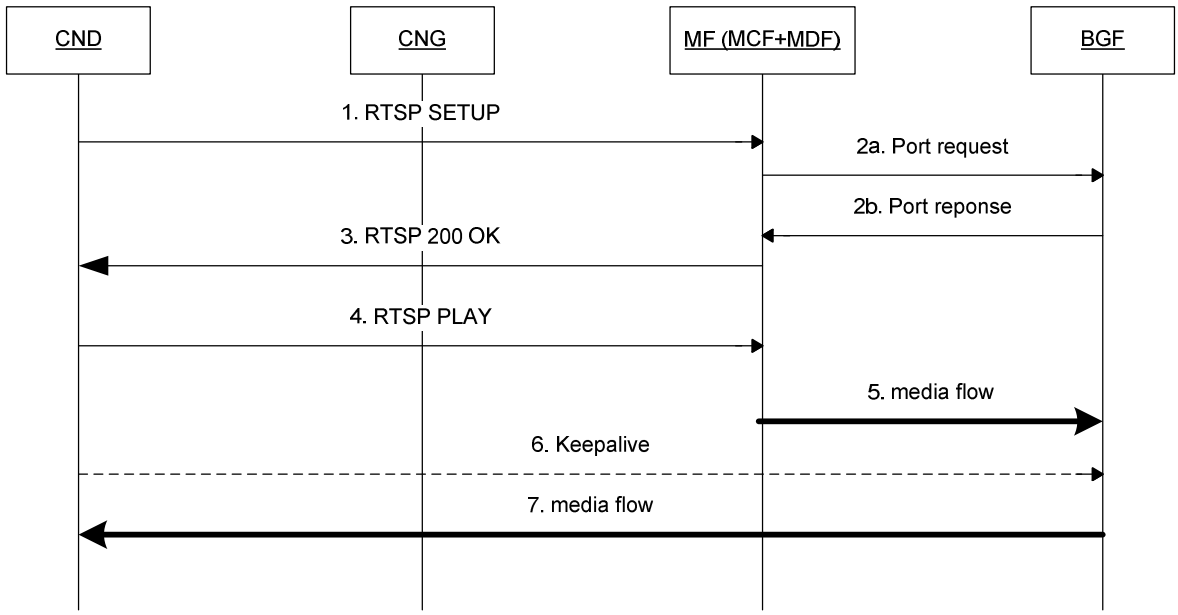


Figure 10: RTSP Hosted NAT flow diagram

It is also possible to adapt the described solution to a scenario where the C-BGF is missing. In such a scenario the MF shall support directly the symmetric NAT capabilities of the C-BGF.

Annex A (informative): Example of a secure boot protocol

A.1 Type 1 STB architecture

A.1.1 Primary boot loader

The primary master boot loader of the STB main job is to bring up the secondary boot loader from a power on scenario after verifying its signature is valid.

This boot loader must be stored in read only memory (so it cannot be overwritten during lifetime of STB – if it was in writable storage the STB security could be compromised).

If the checksums fails on the READ ONLY ROM (which stores secondary loader) we can be certain that either there is a major STB hardware failure (flash chip failed) or a hacker has attempted to compromise the system security, in either case the only option is a return to factory.

The secondary boot loader is itself signed and before the primary boot loader launches it, it must be checked ahead of execution by verifying its signature. Please see primary boot loader for more information.

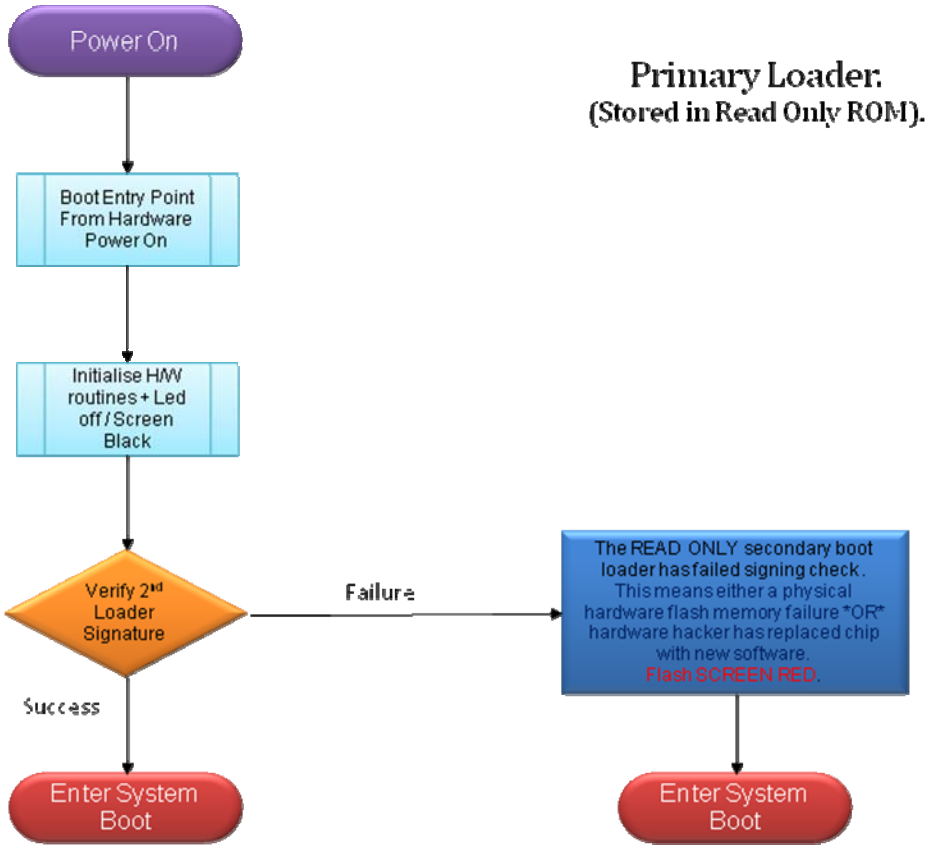


Figure A.1.1: Primary loader process flow

A.1.2 Secondary boot loader

Once the STB launches the secondary boot loader it needs to look at the flash memory and check for available, active bootable partitions. Furthermore the secondary loader must verifying the integrity and signing of the partitions before commencing boot execution on any of these partitions. If any of these checks fail the secondary loader must failover into a recovery mode.

If the secondary boot loader fails its signatures or fails to find an active boot partition in flash it cannot proceed any further in booting, if this happens there has been a critical failure and the entire software in the flash storage has either been corrupted, tampered with or there has been a physical hardware failure. In this scenario the boot loader must try to switch to a recovery screen and provide the customer with an option to recover the STBs flash storage to a normal bootable state.

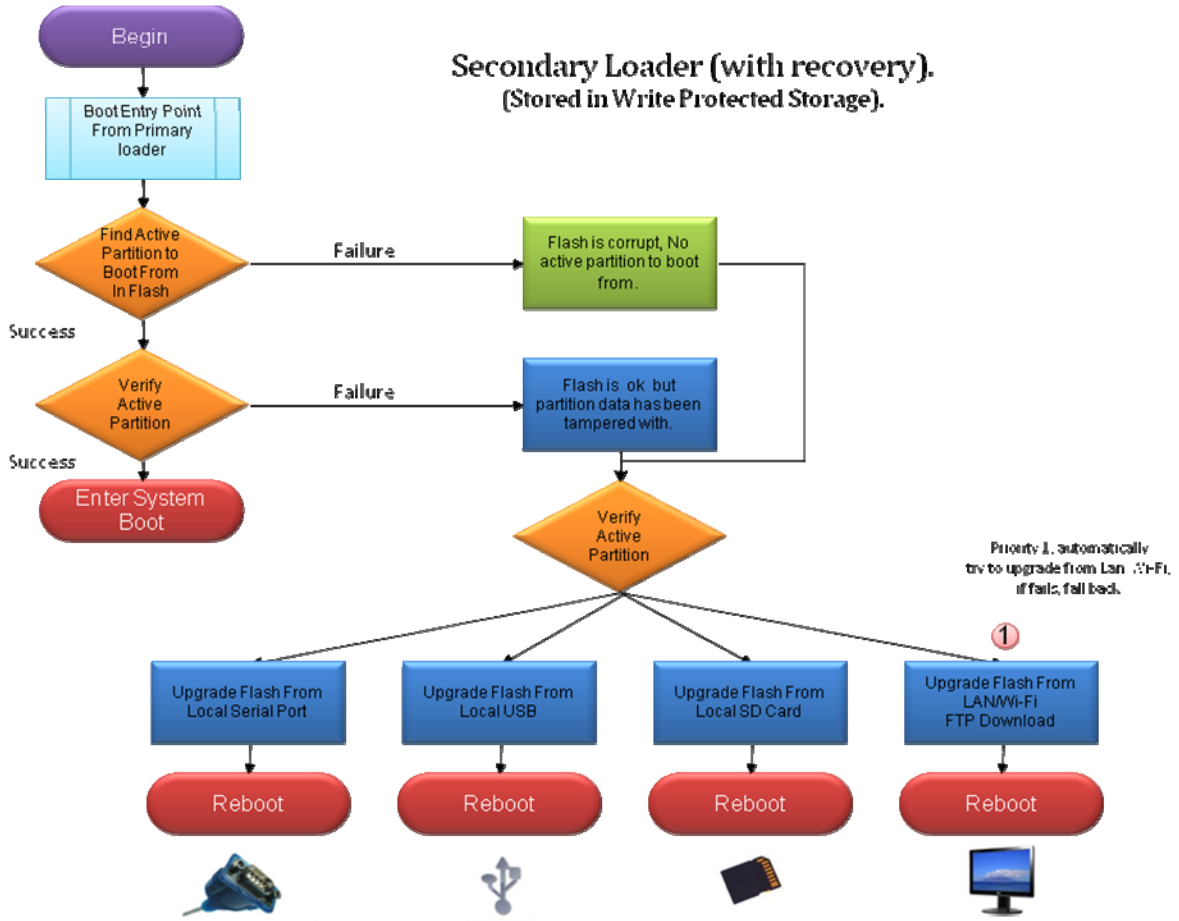


Figure A.1.2: Secondary loader process flow

A.1.3 Secure boot process flow

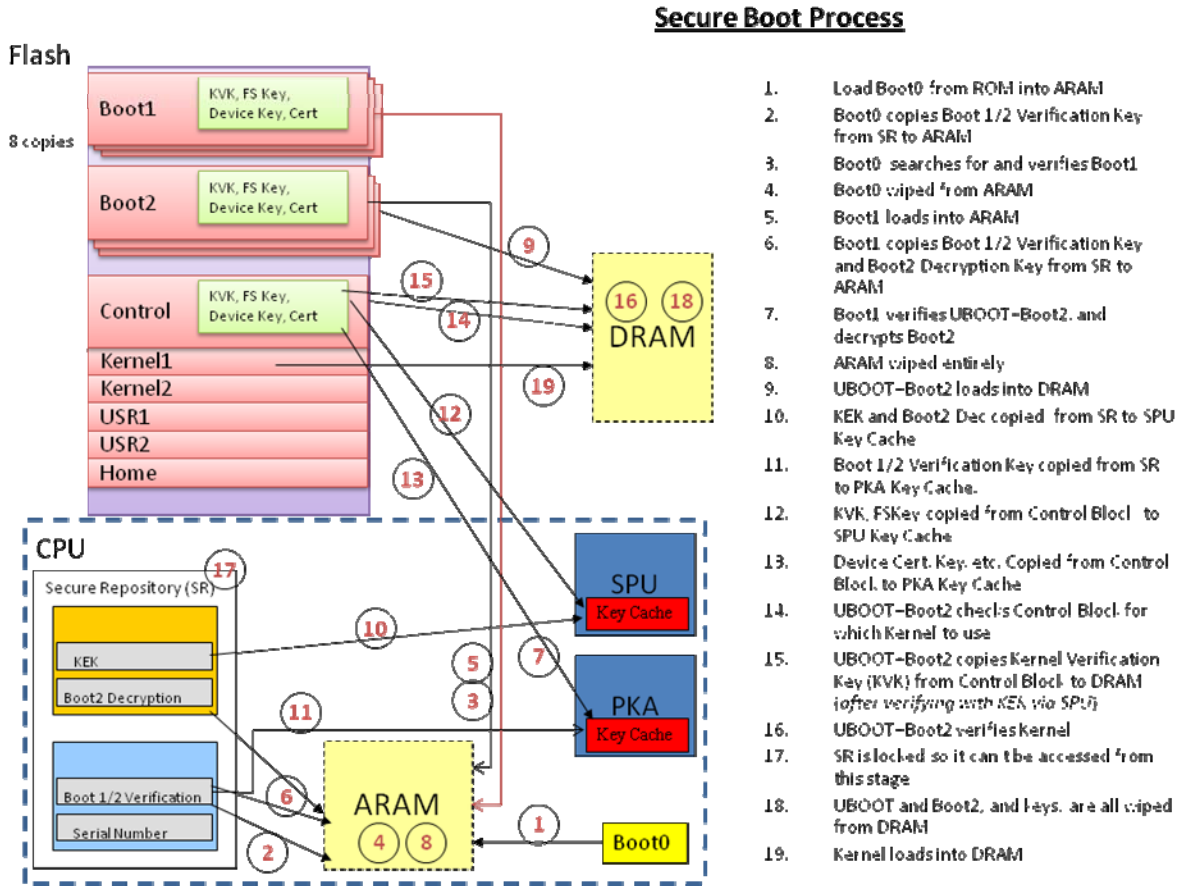


Figure A.1.3 Secure boot process detail

A key set to support the Secure Boot process, including creating the signatures, will need to be created for each test or live environment by the OEM branch trust and sent to the OEM securely. The OEM will be responsible for installing these keys to STB's.

Keys for the LIVE environment will be created (and sent to the OEM build process) after the TEST key set has been implemented and tested.

A.1.4 Error handing and recovery procedures

A.1.4.1 General

It must be possible to remotely recover the STB from the secondary boot loader in the event that all flash storage is either corrupted or in an erroneous (signature failed) state. This solution will not only reduce 'factory returns' from software corruption issues but will allow a customer's STB to remotely recover itself via LAN in the event of catastrophic software failure. If this feature is not implemented a 'truck roll' would be required for every flash corrupted STB.

A.1.4.2 Recovery sources

It should be possible to recover from several sources:

- SD Card slot.
- USB attached storage.

- LAN (via automatic DHCP configured Ethernet or Wi-Fi).
- Serial port download (if developer board).

The recovery images must recover the 'recovery image' partition and create a valid partition table setting the primary boot to be 'recovery'.

A.1.4.3 Recovery success verification & re-try

Once the recovery has completed its update it must verify the writing of the partition sectors to ensure there is not a flash failure fault. If verification fails it must retry (up to 3 times) before logging a failure and letting the customer know there is a hardware fault and the STB is physically faulty with failed flash.

There shall be provision to store information about the STB itself in a reserved area of read/write storage. This should include the following information and have reserve capacity for additional fields (as required by IPTV Service Provider. During a full 'Virgin' re-installation this information must be retained:

- Manufacturing date.
- First install date.
- Repair date.
- Installed boot OS version.
- Installed rescue OS version.
- Last IPTV Service Provider MMS remote connection date/time.
- Unique serial identification.
- Hardware version.
- Hardware supplier.
- Number of reinstallations.
- Number of reinstallation attempts.
- Result of last flash memory test.
- Result of last graphics memory test.
- Result of last main memory test.
- MAC address of LAN.
- IPTV Service Provider Remote Recovery Server DNS Name.
- IPTV Service Provider Remote Recovery Server Credentials.

A.1.4.4 Recovery firmware

The STB must implement a stand alone recovery firmware image that can automatically re-image the 'normal system boot' partition back to a factory virgin state in the event of flash corruption or software update issues on the main partition render a non functioning STB. The backup image should include a minimal completely standalone Linux™ boot system (not dependant on the main system partition at all) to recover the STB back into a working state.

A.1.4.5 Automated re-imaging of 'recovery partition'

If the entire recovery firmware has been corrupted (mitigation for which is described in clause A.1.4) the STB must be recovered by one of the recovery sources described in clause A.1.4.2. The intention of this recovery is to restore the 'recovery image' before rebooting which will initiate a virgin installation.

NOTE: The ability to recover automatically from a remote IPTV Service Provider package manager server requires the recovery from Wi-Fi or LAN to be supported by the secondary boot loader and is for further study.

This feature is critical to allow the STB to automatically recover from a catastrophic failure in the event that the entire flash content gets wiped. If automatic recovery cannot be completed successfully the STB should be able to recover from SD Card or USB. On development boards additionally recovery can be done via serial.

A.1.4.6 Recovery user interface

On booting of the recovery partition the customer must be shown that they have entered a 'recovery mode' after a catastrophic error via a simple on screen user interface, or simply if they manually forced a recovery using the reset mode.

A.1.4.7 Recovery functionality

From the recovery screen the customer should be presented with three options, they can choose from, either.

Abort recovery: In the event they entered into Recovery mode by mistake. This shall just reboot back into normal reboot mode.

Virgin Recover: All personal data is lost, all user setting are deleted, all private data removed.

Full factory reset.

Partial Recover: All system software is re-imaged but all personal data partitions are left intact. No personal data is lost.

A.1.4.8 UI fecovery screen

A progress indicator should show progress bar during recovery. The user should not be allowed to enter recovery if STB is NOT plugged into mains. (For a good user experience of the recovery process.).

The switch of the default boot partition back to 'Normal System Boot' shall not be set until either virgin recovery or partial recovery has been completed and verified as successful. This means that if the recovery failed (typically due to lack power) the STB will reboot back into recovery mode the next time it is powered on.

A.1.4.9 Start-up animation sequence

The video graphics subsystem must be initialised early in the boot strap (within 5 seconds of power on) so that it can perform a boot animation sequence.

The intention of the native application is to produce a high quality rendition of the IPTV Service Providers corporate logo.

The STB early init code must be capable of loading this animation and displaying it during the initial boot strap. This will form the first boot animation. Ideally this animation should execute in parallel to the continuing boot strap.

A.1.4.10 Start-up scripts & driver initialization

Boot scripts and driver loading priorities will be optimized.

Annex B (informative): Examples of a secure run time protocols

B.1 Type 1 STB architecture

B.1.1 Secure CND run time protocol

Since no acceptable anti-virus product for the CND (STB) may be suitable, protection against viruses will be provided by checking signatures on all code at runtime. Any code that has been altered will not be allowed to run.

NOTE: This applies at runtime, and is separate to controls over installation.

Executables and libraries are signed using signing keys kept on IPTV Service Provider, OEM or ISP's Offline Code Signing Server (separate keys for TEST and LIVE environment). The verification of these signatures is enforced through a 'patched' Kernel™ function, which checks signatures against an internal database of verification keys, as detailed in clause 6.3.3. This supports libraries as well as executables.

As an alternative, a protocol based on RSA key pairs, where the public (verification) key is stored in the Kernel™ may be implemented. But note that this allows executables and modules to be verified, but not libraries. The verification keys are protected from alteration as they are stored in the Kernel™, which is signed and this signature is verified in the Secure Boot process, by Boot2, before the Kernel™ loads.

B.1.2 Kernel™ signing patch

It is possible to patch to Linux Kernel™ (and associated std clib) to support a security model. That will ensure that only correctly signed applications installed with the IPTV Service Provider package manager are authorized to execute by the Linux Kernel™.

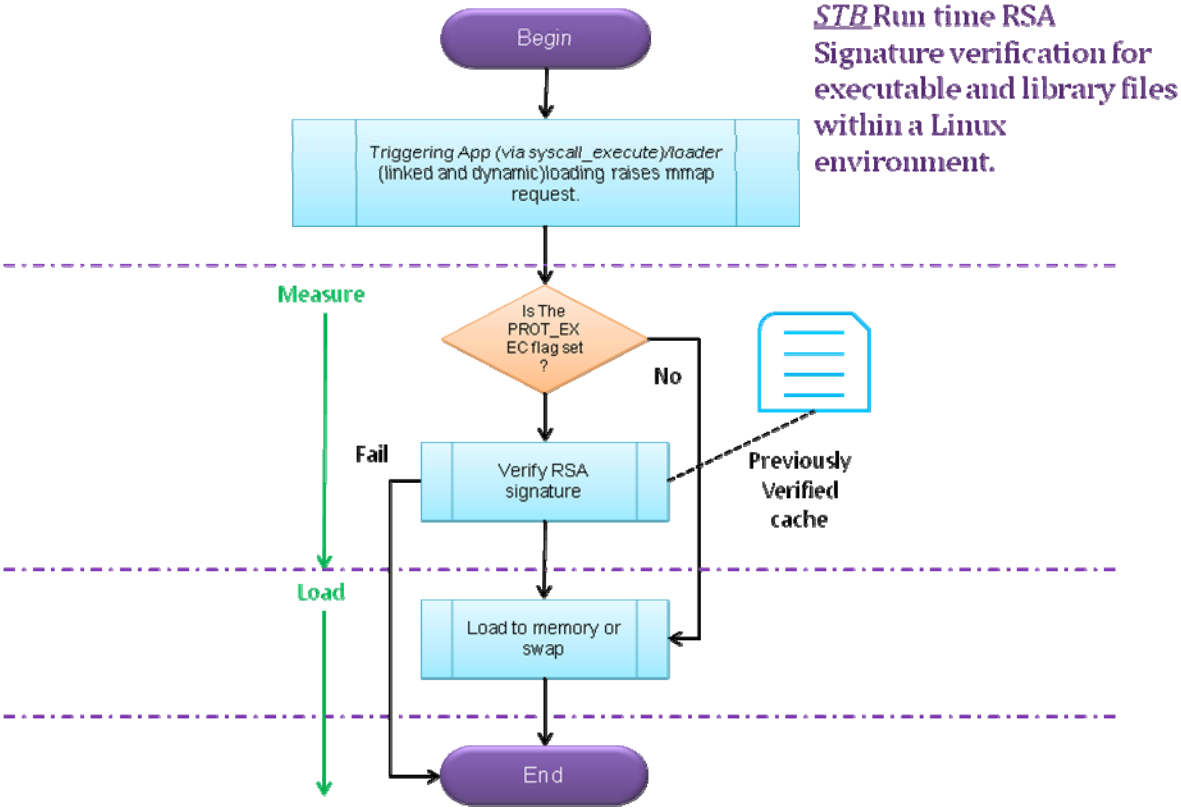


Figure B.1.1: Secure run time detail

Annex C (informative): Example of a secure package download protocol

C.1 Type 1 STB architecture

C.1.1 Secure package download overview

STB's receive new and updated software, including firmware and kernels™, through IPTV Service Provider, OEM or ISP's Update Server. These will be digitally signed, so the STB will only install software that has been checked, approved and signed by IPTV Service Provider, OEM or ISP.

Software updates are performed by the STB. Updates for locally connected equipment are also performed through the STB, since this is the only item that has Internet connectivity.

Code is developed internally, or received from a supplier, according to designs and processes. In either case, it will be signed and deployed for testing in the TEST environment, then after approval, signed and deployed in the LIVE environment. There is a different hierarchy used in TEST and LIVE, so code signed for one will not be accepted in the other.

STB's will run a periodic process to check Update Sever for software updates. These will be downloaded over a mutually authenticated SSL connection.

The software packages are digitally signed under a code signing certificate issued by the IPTV Service Provider, OEM or ISP PKI.

The STB will check the signatures, which are checked using OpenSSL-based code on the STB, and verify the certificates including a check on revocation status. The process is transparent to the user and requires no involvement.

The IPTV Service Provider, OEM or ISP has the means to prevent code being installed in future by revoking the code signer certificate used to sign. The code signing process addresses the six threats shown in the figure C.1.1.

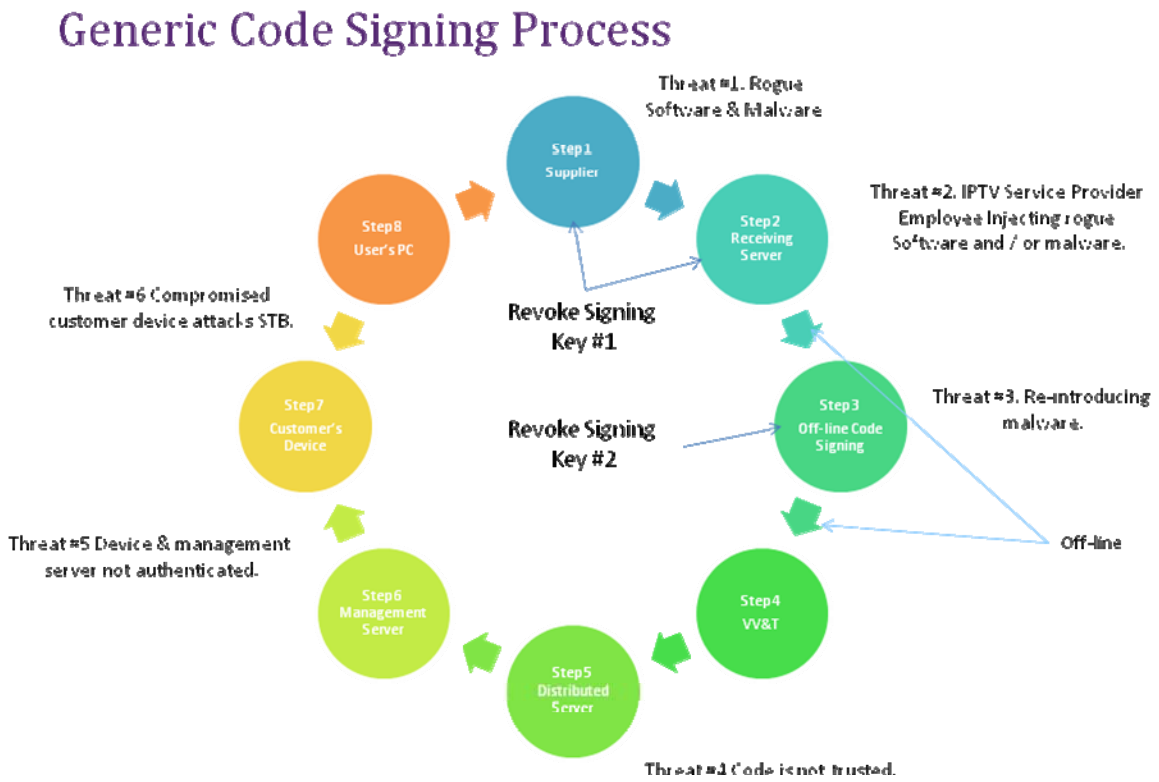


Figure C.1.1: Package download threat model

C.1.2 Secure package download protocol

The detailed protocol steps are described below, but before a package can be securely downloaded, the following prerequisites must be satisfied:

1) STB has fully loaded OS.

2) Network connection established.

3) STB clock set by NTP.

4) Device Private Key prepared:

a) Read from Control Block.

b) Decrypt within Secure Processor using KEK from SR.

c) Copy to encrypted File System.

5) Device certificate prepared:

a) Read from Control Block.

b) Verify HMAC within Secure Processor by KEK from SR.

c) Copy to encrypted File System.

6) Device ICA Certificate prepared:

a) Read from Control Block.

b) Verify HMAC within Secure Processor using KEK from SR.

c) Copy to encrypted File System.

7) Root CA Certificate prepared:

a) Read from Control Block.

b) Verify HMAC within Secure Processor using KEK from SR.

c) Copy to encrypted File System.

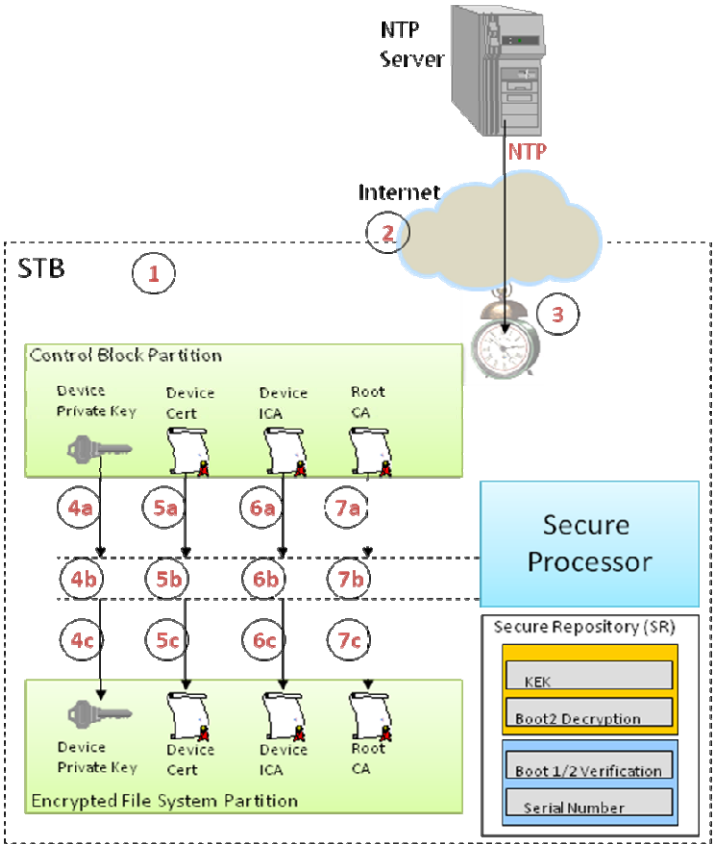


Figure C.1.2. Package management prerequisites

Only then can the package be downloaded according to the following process steps.

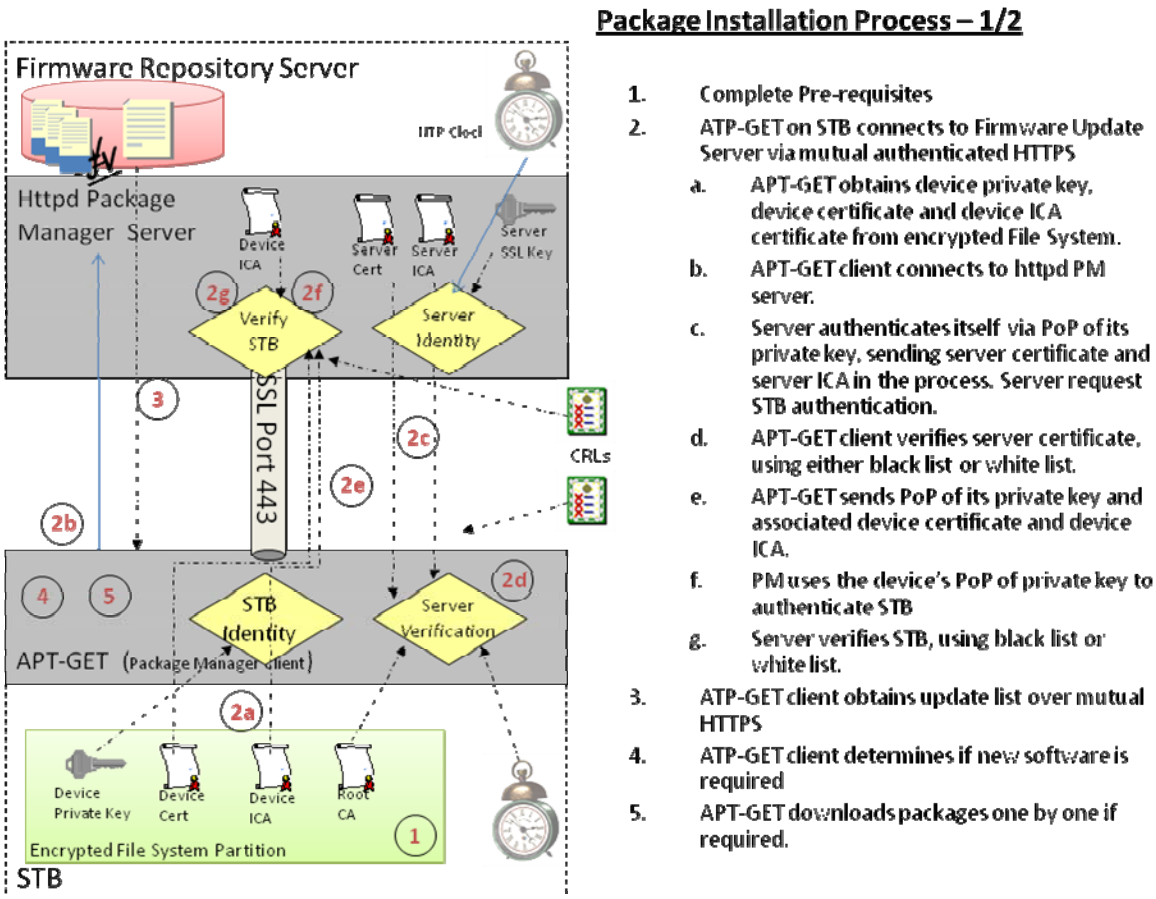


Figure C.1.3: Package installation process - 1/2

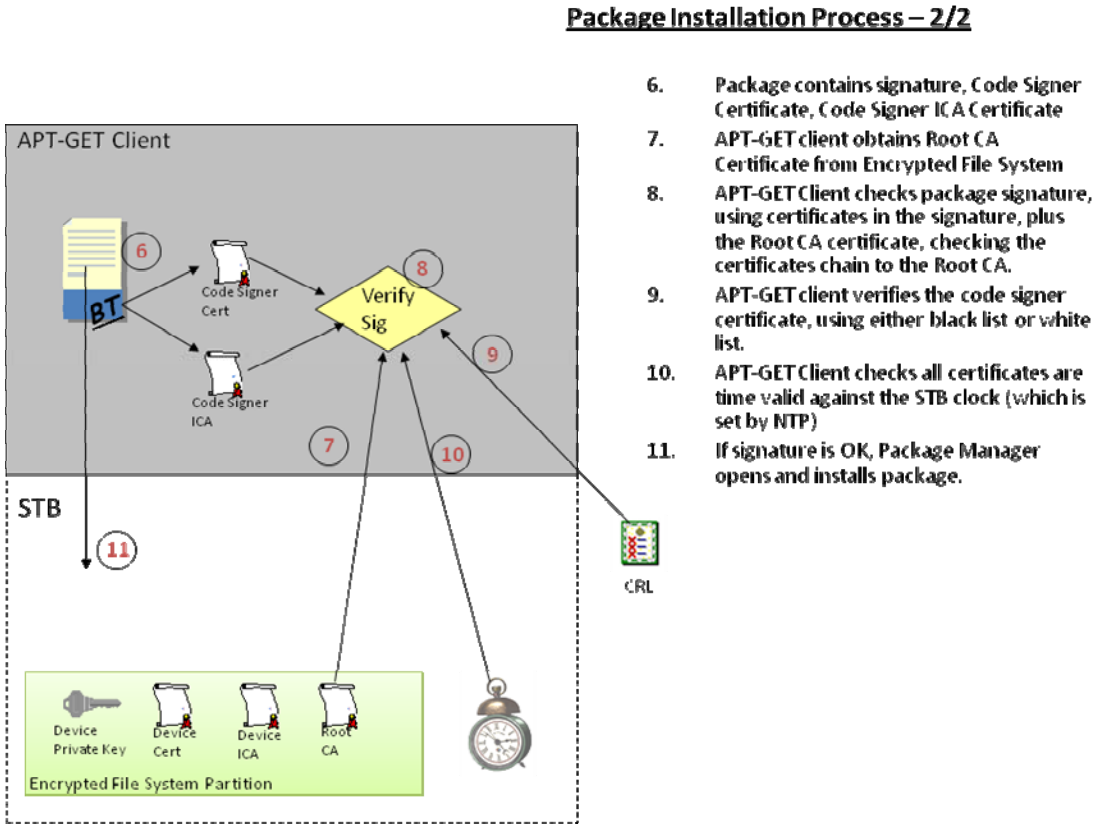


Figure C.1.4 Package installation process - 2/2

NOTE: The solution should ensure that the Root Certificate is in a separate protected storage to that used by a browser for normal operation, and contain ONLY the IPTV Service Provider, OEM or ISP Root CA certificate. Otherwise, code not signed by IPTV Service Provider, OEM or ISP might be allowed to install.

A safe and stable version of an SSL implementation, for example 0.9.8n OpenSSL, should be installed on STB's - confirm this on STB images.

Annex D (informative): Bibliography

- ETSI TS 133 203: "Digital cellular telecommunications system (Phase 2+); Universal Mobile Telecommunications System (UMTS); LTE; 3G security; Access security for IP-based services".
- ETSI TS 185 005: "Telecommunications and Internet converged Services and Protocols for Advanced Networking (TISPAN); Services requirements and capabilities for customer networks connected to TISPAN NGN".
- ETSI TR 187 008: "Telecommunications and Internet converged Services and Protocols for Advanced Networking (TISPAN); NAT traversal feasibility study report".
- ETSI TR 185 008: "Telecommunications and Internet converged Services and Protocols for Advanced Networking (TISPAN); Analysis of security mechanisms for customer networks connected to TISPAN NGN R2".
- ETSI TR 187 002: "Telecommunications and Internet converged Services and Protocols for Advanced Networking (TISPAN);TISPAN NGN Security (NGN-SEC);Threat, Vulnerability and Risk Analysis".

History

Document history		
V3.1.1	September 2011	Publication