ETSI TS 101 888-2 V4.1.1 (2002-05)

Technical Specification

Telecommunications and Internet Protocol Harmonization Over Networks (TIPHON) Release 4; Security Test Specifications; Part 2: H.323 Environment

Reference

DTS/TIPHON-06014-2R4

Keywords

H.323, IP, protocol, testing, security, VoIP

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

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://webapp.etsi.org/IPR/home.asp).

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 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 Project Telecommunications and Internet Protocol Harmonization Over Networks (TIPHON).

The present document is part 2 of a multi-part deliverable covering Security Test Specifications, as identified below:

Part 1: "Framework";

Part 2: "H.323 Environment".

1 Scope

The present document is one part of the security testing standards for which a framework is available in TR 101 888-1.

The scope of the present document is to define the security test specifications for TIPHON Release 4 for the H.323 environment.

The security methods considered in the present document are related only to IP based networks. The signalling path and the media path in the SCN is considered to be secure ("Trust by wire").

2 References

The following documents contain provisions which, through reference in this text, constitute provisions of the present document.

- References are either specific (identified by date of publication and/or edition number or version number) or non-specific.
- For a specific reference, subsequent revisions do not apply.
- For a non-specific reference, the latest version applies.

[1] ITU-T Recommendation H.225.0: "Call signalling protocols and media stream packetization for packet-based multimedia communication systems".

[2] ITU-T Recommendation H.235: "Security and encryption for H.Series (H.323 and other H.245-based) multimedia terminals".

[3] ITU-T Recommendation H.245: "Control protocol for multimedia communication".

[4] ITU-T Recommendation H.323: "Packet-based multimedia communications systems".

3 Definitions and abbreviations

3.1 Definitions

For the purpose of the present document, the terms and definitions given in the IUT-T Recommendations H.225.0 [1], H.235 [2], H.245 [3] and H.323 [4] apply.

3.2 Abbreviations

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

A	Audio
D	Data
IP	Internet Protocol
SCN	Switched Circuit Networks

4 Security Test Strategy

Security testing should be performed after a vendor has completed product and system testing with the ETSI testing standards.

The basic idea for security testing is to show the generation and insertion of the security bits into the specific parameters of the H.323 messages. Because this mechanism is exactly the same on the senders and the receiver's side, no distinction is necessary.

To test entities for their implementation of security two entities (that are already interworking) need to be connected. In the case of an incorrect security information it is necessary to go into the detail of the generation of the security bits. In order to be able to determine the reason for this failure the security tests strategy is just to look at the different steps of the generation and insertion of the security bits into the protocol elements. This is the only way to determine the failure.

The Security testing shall be performed for the following configurations:

- Signalling path:
  - Gatekeeper and Terminal;
  - Gatekeeper and Gateway;
  - Gatekeeper and Gatekeeper.
- Media path:
  - Terminal and Terminal;
  - Terminal and Gateway;
  - Gateway and Gateway.
- Global Service Providers:
  - BES and TRC;
  - BES and CH;
  - BES and CA.

The security testing shall be performed in three different parts where the first part deals with the security testing for the signalling path (Terminal, Gatekeeper, Gateway) using ITU-T Recommendation H.235 [2] annex D. The second part deals with the security aspects for the signalling path equivalent to the first but using ITU-T Recommendation H.235 [2] annex F and the media path using H.235. The third part handles the security testing from the BES to the global service providers.

5 H.235 annex D

5.1 Overview

Figure 1 shows the basic steps to be taken at the originating entity.

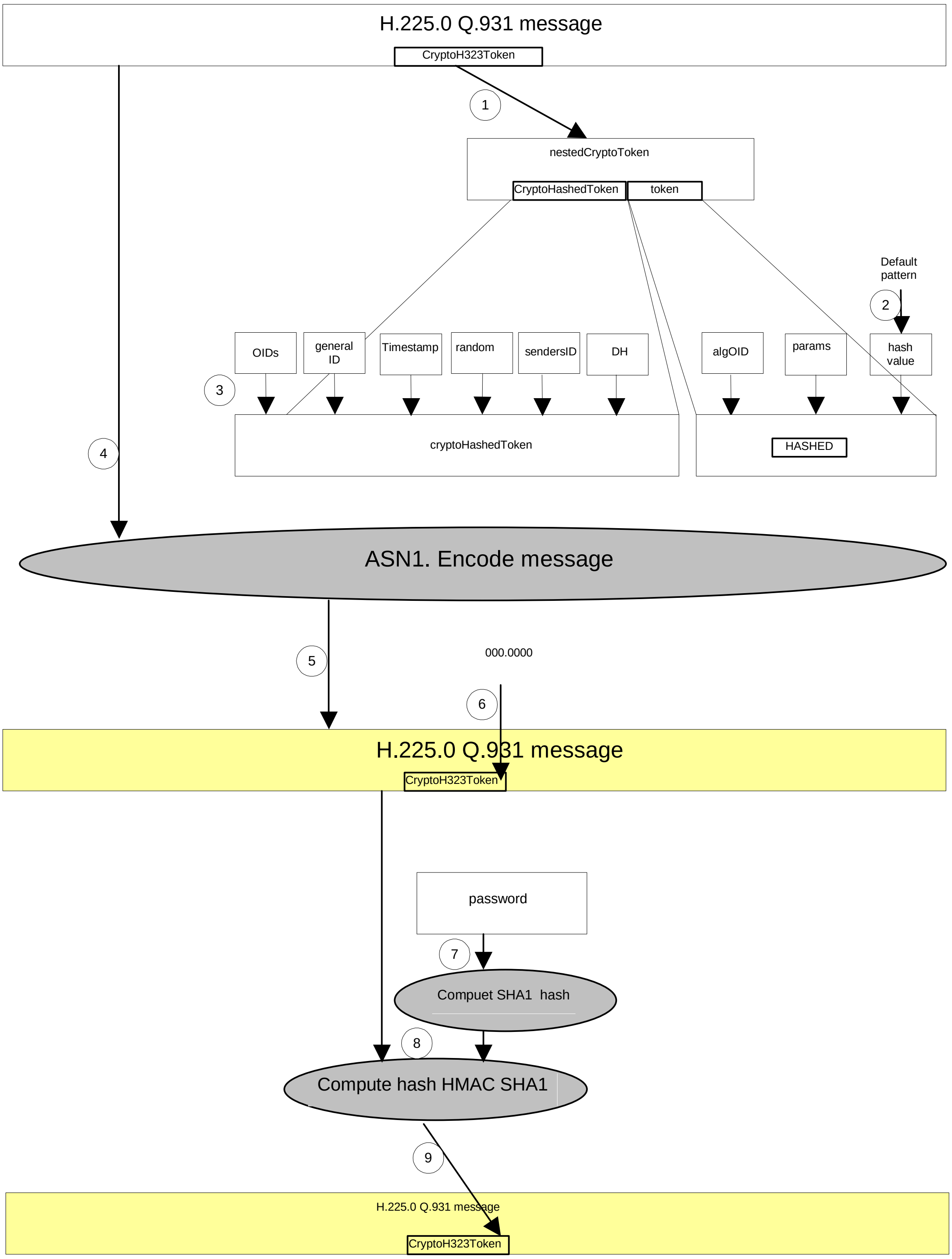

Figure 1: Stepwise approach for sender

Figure 2 shows the basic steps to be taken at the receiving side starting with the entire message, decoding, breaking it into pieces and extracting the necessary parts and the final computation/verification step.

NOTE 1: The figures just visualize the essential steps as an example and correlate with the print out in clause 5.3; in any case, the procedures and description of H.235 [2] annex D take precedence.

NOTE 2: The figures and print out reflect H.235v1, i.e. sendersID is not used.

NOTE 3: The figures and print out reflect a scenario endpoint to gatekeeper; other scenarios and examples are not shown.

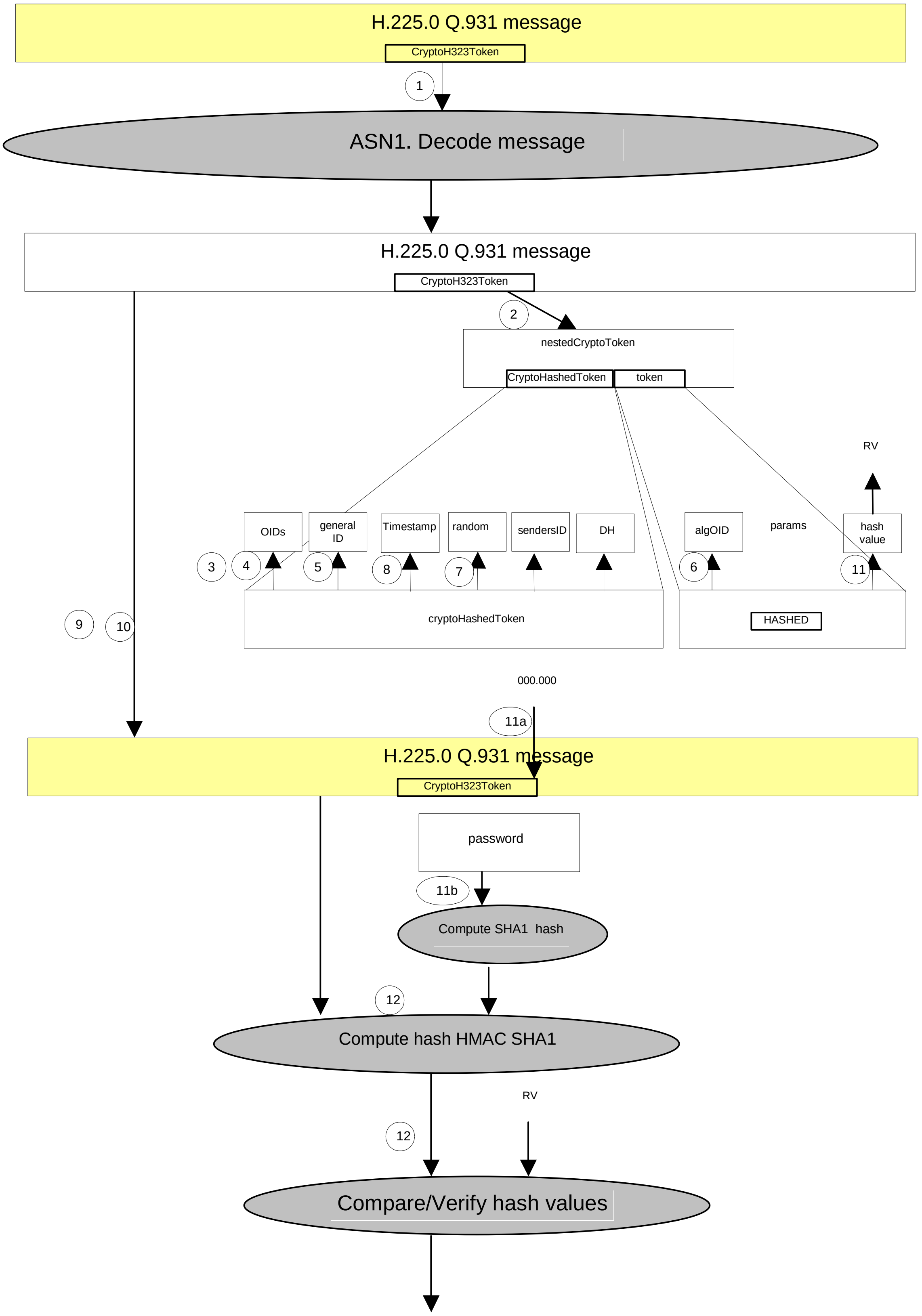

Figure 2: Stepwise approach for receiver

The example shown uses the RRQ that has been sent by a terminal and received at the gatekeeper.

- The received RRQ message in binary and with all fields shown.
- The received binary message part and the separate steps for the verification.

5.2 Received message

A received RRQ message with embedded Cryptotoken:

```
*********************************
*  RECEIVE RRQ FROM EP AT GK  *
*********************************


14:34:12 TPKTCHAN  : Address:
14:34:12 TPKTCHAN  :  0> <14> TransportAddress = (0) .  <1084> CHOICE ...
14:34:12 TPKTCHAN  :  1> . <289> ipAddress = (0) .  <1081> SEQUENCE
14:34:12 TPKTCHAN  :  2> . . <290> ip = (4) '...j' =0x8b17ca6a <139.23.202.106> .  <1066> OCTET
STRING (4..4)
14:34:12 TPKTCHAN  :  2> . . <292> port = (1720) .  <115> INTEGER (0..65535)
14:34:21 UDPCHAN   : New message (channel 0) recv <-- registrationRequest:
14:34:21 UDPCHAN   : Address:
14:34:21 UDPCHAN   :  0> <669> TransportAddress = (0) .  <1084> CHOICE ...
14:34:21 UDPCHAN   :  1> . <670> ipAddress = (0) .  <1081> SEQUENCE
14:34:21 UDPCHAN   :  2> . . <671> ip = (4) '...j' =0x8b17ca6a <139.23.202.106> .  <1066> OCTET
STRING (4..4)
14:34:21 UDPCHAN   :  2> . . <673> port = (1151) .  <115> INTEGER (0..65535)
14:34:21 UDPCHAN   : Binary:
14:34:21 UDPCHAN   : 00000   0f 80 3a 27 06 00 08 91 4a 00 02 00 08 2b 0c 02    |.€:'...‘J....+..|
14:34:21 UDPCHAN   : 00016   88 53 02 06 01 80 84 01 40 00 08 00 00 00 00 00    |ˆS...€„.@.......|
14:34:21 UDPCHAN   : 00032   00 21 72 00 5b 6f 20 00 52 00 07 00 00 fb 38 00    |.!r.[o .R....û8.|
14:34:21 UDPCHAN   : 00048   12 fa 68 00 12 c5 19 00 50 6f 20 00 52 00 07 00    |.úh..Å..Po .R...|
14:34:21 UDPCHAN   : 00064   00 fb 38 00 12 fa 68 00 12 00 00 00 00 00 00 00    |.û8..úh.........|
14:34:21 UDPCHAN   : 00080   00 6c c0 00 50 fb 38 00 12 fa 94 00 12 fa 9c 00    |.lÀ.Pû8..ú"..úœ.|
14:34:21 UDPCHAN   : 00096   12 01 ec 00 00 02 36 00 00 00 0e 00 00 02 36 00    |..ì...6.......6.|
14:34:21 UDPCHAN   : 00112   00 60 76 3d 18 20 ec f3 2e 00 00 00 00 9d b5 72    |.`v=. ìó.....•µr|
14:34:21 UDPCHAN   : 00128   5a 00 50 00 c2 01 ee 00 00 00 00 00 00 ff ff ff    |Z.P.Â.î......ÿÿÿ|
14:34:21 UDPCHAN   : 00144   ff 20 31 20 33 32 31 32 20 1e 00 00 01 00 8b 17    |ÿ 1 3212 .....‹.|
14:34:21 UDPCHAN   : 00160   ca 6a 04 80 01 00 8b 17 ca 6a 04 7f 22 c0 0b 0b    |Êj.€..‹.Êj.•"À..|
14:34:21 UDPCHAN   : 00176   00 0b 0f 54 65 73 74 20 61 70 70 6c 69 63 61 74    |...Test applicat|
14:34:21 UDPCHAN   : 00192   69 6f 6e 08 52 41 44 56 69 73 69 6f 6e 00 02 08    |ion.RADVision...|
14:34:21 UDPCHAN   : 00208   00 46 c3 56 53 54 39 34 48 54 04 00 35 00 33 00    |.FÃVST94HT..5.3.|
14:34:21 UDPCHAN   : 00224   34 00 30 00 33 60 0b 0b 00 0b 0f 54 65 73 74 20    |4.0.3`.....Test |
14:34:21 UDPCHAN   : 00240   61 70 70 6c 69 63 61 74 69 6f 6e 08 52 41 44 56    |application.RADV|
14:34:21 UDPCHAN   : 00256   69 73 69 6f 6e 12 2b 80 56 01 74 07 00 08 81 6b    |ision.+€V.t...•k|
14:34:21 UDPCHAN   : 00272   00 01 01 45 00 07 00 08 81 6b 00 01 05 c0 3a 22    |...E....•k...À:"|
14:34:21 UDPCHAN   : 00288   62 db 01 29 22 00 53 00 69 00 65 00 6d 00 65 00    |bÛ.)".S.i.e.m.e.|
14:34:21 UDPCHAN   : 00304   6e 00 73 00 20 00 47 00 61 00 74 00 65 00 6b 00    |n.s. .G.a.t.e.k.|
14:34:21 UDPCHAN   : 00320   65 00 65 00 70 00 65 00 72 07 00 08 81 6b 00 01    |e.e.p.e.r...•k..|
14:34:21 UDPCHAN   : 00336   06 00 60 07 89 a6 ee 75 bb 59 c1 a6 ca a4 72 01    |..`.‰¦îu»YÁ¦Ê¤r.|
14:34:21 UDPCHAN   : 00352   00 01 00 01 00 01 00                               |.......|


14:34:21 UDPCHAN   : Message:
14:34:21 UDPCHAN   :  0> <584> RasMessage = (6502) .  <771> CHOICE ...
14:34:21 UDPCHAN   :  1> . <586> registrationRequest = (4294967185) .  <702> SEQUENCE ...
14:34:21 UDPCHAN   :  2> . . <587> requestSeqNum = (14888) .  <883> INTEGER (1..65535)
14:34:21 UDPCHAN   :  2> . . <588> protocolIdentifier = (6)  { itu-t recommendation h 2250 0 2 }.
<878> OBJECT IDENTIFIER
14:34:21 UDPCHAN   :  2> . . <590> nonStandardData = (4294967185) .  <972> SEQUENCE
14:34:21 UDPCHAN   :  3> . . . <591> nonStandardIdentifier = (10964) .  <969> CHOICE ...
14:34:21 UDPCHAN   :  4> . . . . <592> object = (8)  { iso identified-organization 12 2 1107 2 6 1
}.  <121> OBJECT IDENTIFIER
14:34:21 UDPCHAN   :  3> . . . <594> data = (132) '.@........!r.[o .R.....8...h.....Po
.R.....8...h..........l..P.8...............6.......6..`v=. .........rZ.P............. 1 321'
=0x014000080000000000002172005b6f2000.  <125> OCTET STRING
14:34:21 UDPCHAN   :  2> . . <601> discoveryComplete = (0) .  <83> BOOLEAN
14:34:21 UDPCHAN   :  2> . . <602> callSignalAddress = (1) .  <381> SEQUENCE OF
14:34:21 UDPCHAN   :  3> . . . <603> * = (6669) .  <1084> CHOICE ...
14:34:21 UDPCHAN   :  4> . . . . <604> ipAddress = (4294967185) .  <1081> SEQUENCE
14:34:21 UDPCHAN   :  5> . . . . . <605> ip = (4) '...j' =0x8b17ca6a <139.23.202.106> .  <1066>
OCTET STRING (4..4)
14:34:21 UDPCHAN   :  5> . . . . . <607> port = (1152) .  <115> INTEGER (0..65535)
14:34:21 UDPCHAN   :  2> . . <608> rasAddress = (1) .  <381> SEQUENCE OF
14:34:21 UDPCHAN   :  3> . . . <609> * = (6669) .  <1084> CHOICE ...
14:34:21 UDPCHAN   :  4> . . . . <610> ipAddress = (4294967185) .  <1081> SEQUENCE
14:34:21 UDPCHAN   :  5> . . . . . <611> ip = (4) '...j' =0x8b17ca6a <139.23.202.106> .  <1066>
OCTET STRING (4..4)
14:34:21 UDPCHAN   :  5> . . . . . <613> port = (1151) .  <115> INTEGER (0..65535)
14:34:21 UDPCHAN   :  2> . . <614> terminalType = (4294967185) .  <1050> SEQUENCE ...
14:34:21 UDPCHAN   :  3> . . . <615> vendor = (4294967185) .  <980> SEQUENCE ...
14:34:21 UDPCHAN   :  4> . . . . <616> vendor = (4294967185) .  <975> SEQUENCE ...
14:34:21 UDPCHAN   :  5> . . . . . <617> t35CountryCode = (11) .  <116> INTEGER (0..255)
14:34:21 UDPCHAN   :  5> . . . . . <618> t35Extension = (11) .  <116> INTEGER (0..255)
14:34:21 UDPCHAN   :  5> . . . . . <619> manufacturerCode = (11) .  <115> INTEGER (0..65535)
14:34:21 UDPCHAN   :  4> . . . . <620> productId = (16) 'Test application'
=0x546573742061706c69636174696f6e.  <979> OCTET STRING (1..256)
14:34:21 UDPCHAN   :  4> . . . . <622> versionId = (9) 'RADVision' =0x524144566973696f6e.  <979>
OCTET STRING (1..256)
14:34:21 UDPCHAN   :  3> . . . <624> terminal = (4294967185) .  <986> SEQUENCE ...
14:34:21 UDPCHAN   :  3> . . . <625> mc = (0) .  <83> BOOLEAN
14:34:21 UDPCHAN   :  3> . . . <626> undefinedNode = (0) .  <83> BOOLEAN
```

```
14:34:21 UDPCHAN   :  2> . . <627> terminalAlias = (2) .  <380> SEQUENCE OF
14:34:21 UDPCHAN   :  3> . . . <628> * = (3942) .  <962> CHOICE ...
14:34:21 UDPCHAN   :  4> . . . . <629> e164 = (17) '13902320210601152'
=0x3133393032333230323130363031313532.  <961> IA5String (1..128) FROM '#*,0123456789'
14:34:21 UDPCHAN   :  3> . . . <631> * = (4187) .  <962> CHOICE ...
14:34:21 UDPCHAN   :  4> . . . . <632> h323-ID = (10) '.5.3.4.0.3' =0x00350033003400300033.  <960>
BMPString (1..256)
14:34:21 UDPCHAN   :  2> . . <634> endpointVendor = (4294967185) .  <980> SEQUENCE ...
14:34:21 UDPCHAN   :  3> . . . <635> vendor = (4294967185) .  <975> SEQUENCE ...
14:34:21 UDPCHAN   :  4> . . . . <636> t35CountryCode = (11) .  <116> INTEGER (0..255)
14:34:21 UDPCHAN   :  4> . . . . <637> t35Extension = (11) .  <116> INTEGER (0..255)
14:34:21 UDPCHAN   :  4> . . . . <638> manufacturerCode = (11) .  <115> INTEGER (0..65535)
14:34:21 UDPCHAN   :  3> . . . <639> productId = (16) 'Test application'
=0x546573742061707706c69636174696f6e.  <979> OCTET STRING (1..256)
14:34:21 UDPCHAN   :  3> . . . <641> versionId = (9) 'RADVision' =0x524144566973696f6e.  <979> OCTET
STRING (1..256)
14:34:21 UDPCHAN   :  2> . . <643> cryptoTokens = (1) .  <283> SEQUENCE OF
14:34:21 UDPCHAN   :  3> . . . <644> * = (4466) .  <832> CHOICE ...
14:34:21 UDPCHAN   :  4> . . . . <645> nestedcryptoToken = (9106) .  <192> CHOICE ...
14:34:21 UDPCHAN   :  5> . . . . . <646> cryptoHashedToken = (4294967185) .  <177> SEQUENCE
14:34:21 UDPCHAN   :  6> . . . . . . <647> tokenOID = (7)  { itu-t recommendation h 235 0 1 1 }.
<121> OBJECT IDENTIFIER
14:34:21 UDPCHAN   :  6> . . . . . . <649> hashedVals = (4294967185) .  <239> SEQUENCE ...
14:34:21 UDPCHAN   :  7> . . . . . . . <650> tokenOID = (7)  { itu-t recommendation h 235 0 1 5 }.
<121> OBJECT IDENTIFIER
14:34:21 UDPCHAN   :  7> . . . . . . . <652> timeStamp = (975332060) .  <281> INTEGER (1..-1)
14:34:21 UDPCHAN   :  7> . . . . . . . <653> random = (41) .  <280> INTEGER
14:34:21 UDPCHAN   :  7> . . . . . . . <654> generalID = (36) '.S.i.e.m.e.n.s. .G.a.t.e.k.e.e.p.e.r'
=0x00530069006500d0065006e0073002000.  <278> BMPString (1..128)
14:34:21 UDPCHAN   :  6> . . . . . . <657> token = (4294967185) .  <231> SEQUENCE
14:34:21 UDPCHAN   :  7> . . . . . . . <658> algorithmOID = (7)  { itu-t recommendation h 235 0 1 6
}.  <121> OBJECT IDENTIFIER
14:34:21 UDPCHAN   :  7> . . . . . . . <660> paramS = (4294967185) .  <226> SEQUENCE ...
14:34:21 UDPCHAN   :  8> . . . . . . . . <661> null = (4294967173) .  <95> NULL
14:34:21 UDPCHAN   :  7> . . . . . . . <662> hash = (96) '....u.Y....r'
=0x0789a6ee75bb59c1a6caa47200.  <139> BIT STRING
14:34:21 UDPCHAN   :  2> . . <664> keepAlive = (0) .  <83> BOOLEAN
14:34:21 UDPCHAN   :  2> . . <665> willSupplyUUIEs = (0) .  <83> BOOLEAN
14:34:21 UDPCHAN   :  2> . . <666> maintainConnection = (0) .  <83> BOOLEAN
14:34:21 UDPCHAN   :  2> . . <667> supportsAnnexECallSignalling = (0) .  <83> BOOLEAN
```

5.3 Separate Steps

Verification steps for the obtained CryptoToken:

```
*********************************
*  RECEIVE RRQ FROM EP AT GK  *
*********************************

00:08:31 | _UDP_IN_registrationRequest_for_nodeId_492_(packet length 215 Bytes)______________
0000:08:31 | 0000: 0e 80 3a 27  06 00 08 91  4a 00 02 00  01 00 8b 17  '..:'....J.......'
0000:08:31 | 0010: ca 6a 04 80  01 00 8b 17  ca 6a 04 7f  22 c0 0b 0b  '.j.......j.."...'
0000:08:31 | 0020: 00 0b 0f 54  65 73 74 20  61 70 70 6c  69 63 61 74  '...Test applicat'
0000:08:31 | 0030: 69 6f 6e 08  52 41 44 56  69 73 69 6f  6e 00 02 08  'ion.RADVision...'
0000:08:31 | 0040: 00 46 c3 56  53 54 39 34  48 54 04 00  35 00 33 00  '.F.VST94HT..5.3.'
0000:08:31 | 0050: 34 00 30 00  33 60 0b 0b  00 0b 0f 54  65 73 74 20  '4.0.3`.....Test '
0000:08:32 | 0060: 61 70 70 6c  69 63 61 74  69 6f 6e 08  52 41 44 56  'application.RADV'
0000:08:32 | 0070: 69 73 69 6f  6e 12 2b 80  56 01 74 07  00 08 81 6b  'ision.+.V.t....k'
0000:08:32 | 0080: 00 01 01 45  00 07 00 08  81 6b 00 01  05 c0 3a 22  '...E.....k....:"'
0000:08:32 | 0090: 62 db 01 29  22 00 53 00  69 00 65 00  6d 00 65 00  'b..)".S.i.e.m.e.'
0000:08:32 | 00a0: 6e 00 73 00  20 00 47 00  61 00 74 00  65 00 6b 00  'n.s. .G.a.t.e.k.'
0000:08:32 | 00b0: 65 00 65 00  70 00 65 00  72 07 00 08  81 6b 00 01  'e.e.p.e.r....k..'
0000:08:33 | 00c0: 06 00 60 07  89 a6 ee 75  bb 59 c1 a6  ca a4 72 01  '..`....u.Y....r.'
0000:08:33 | 00d0: 00 01 00 01  00 01 00                                '.......'
0000:08:33 | _______________________________________________________
0000:08:33 | -------------------------------------------------------
```

1) Determine IP-Address:

```
0000:08:33 | New message recv <- registrationRequest on RAS from 492
0000:08:33 | Read IP Address for EP 139.23.202.106:1151
```

2) Read alias:

```
0000:08:66 | EP Alias 53403-> Get User Info (from external database):
0000:08:66 | -> User=Fries, UID=53403, PWLen=20, LC=Wed Aug 25 13:52:19 1999
0000:08:66 | -> Hashed Passphrase (fries sha1-hashed):
0000:08:67 | 0000: 91 27 1c 95  f0 a3 a0 6f  0d 79 75 b1  19 5f a1 28  '.'.....o.yu.._.('
0000:08:67 | 0010: 8a 86 b6 d4                                          '....'
```

3) Read CryptoTokenOID:

```
0000:08:67 | Recv/RecvFrom: Found Crypto Token: token len = 15 Bytes, tokenOID =
0000:08:67 | 0000: 30 20 30 20  38 20 32 33  35 20 30 20  31 20 31     '0 0 8 235 0 1 1'
```

4) Read ClearTokenOID:

```
0000:08:67 | Recv/RecvFrom: Found Crypto Token: token len = 15 Bytes, tokenOID (2) =
0000:08:67 | 0000: 30 20 30 20  38 20 32 33  35 20 30 20  31 20 35     '0 0 8 235 0 1 5'
```

5) Read generalID:

```
0000:08:68 | RecvFrom: Found Crypto Token: token len = 36 Bytes, generalID =
0000:08:68 | 0000: 00 53 00 69  00 65 00 6d  00 65 00 6e  00 73 00 20  '.S.i.e.m.e.n.s. '
0000:08:68 | 0010: 00 47 00 61  00 74 00 65  00 6b 00 65  00 65 00 70  '.G.a.t.e.k.e.e.p'
0000:08:68 | 0020: 00 65 00 72                                         '.e.r'
```

6) Read algorithmOID:

```
0000:08:68 | Recv/RecvFrom: Found Crypto Token: token len = 15 Bytes, algorithmOID =
0000:08:68 | 0000: 30 20 30 20  38 20 32 33  35 20 30 20  31 20 36     '0 0 8 235 0 1 6'
```

7) Read Sequence Number:

```
0000:08:68 | Recv/RecvFrom: Found Crypto Token: sequence_number = 41
```

8) Read timestamp:

```
0000:08:68 | Recv/RecvFrom: Found Crypto Token: timestamp = 975332060
```

9) Read token value:

```
0000:08:68 | Recv/RecvFrom: Found Crypto Token: token len = 96 Bits, token value =
0000:08:68 | 0000: 07 89 a6 ee  75 bb 59 c1  a6 ca a4 72               '....u.Y....r'
```

10) Perform verification checks:

```
0000:08:68 | Recv/RecvFrom: (h235_checkToken) clear token OID check passed
0000:08:68 | Recv/RecvFrom: (h235_checkToken) crypto token OID check passed
0000:08:68 | Recv/RecvFrom: (h235_checkToken) crypto algorithm OID check passed
0000:08:68 | Recv/RecvFrom: (h235_checkToken) time value in range
0000:08:68 | Recv/RecvFrom: (h235_checkToken) generalID check passed
```

11) Locate and read hash value:

```
0000:08:69 | Recv/RecvFrom: (h235_checkToken) found ICV in raw message on position 195
0000:08:69 | 0000: 07 89 a6 ee  75 bb 59 c1  a6 ca a4 72               '....u.Y....r'
```

12) Re-compute hash value:

```
0000:08:69 | Crypto-Module: Start Message Hash Session
0000:08:69 | Crypto-Module: End Message Hash Session
```

13) Verify hash value:

```
0000:08:69 | ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
0000:08:69 | +                                                                        +
0000:08:69 | +                      SUCCESSFUL INTEGRITY CHECK                        +
0000:08:69 | + Recv/RecvFrom: registrationRequest on RAS:
0000:08:69 | + VALID TOKEN received from User Fries (ID: 53403)
0000:08:69 | +                                                                        +
0000:08:69 | ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
```

5.4 Test configurations

5.4.1 Gatekeeper and Terminal

For further study.

5.4.2 Gatekeeper and Gateway

For further study.

5.4.3 Gatekeeper and Gatekeeper

The Gatekeeper Gatekeeper communications according to H.235 [2] annex D is very similar to the terminal Gatekeeper communication. The generalID and the sendersID are the only fields that have different values.

6 H.235 annex F

For further study.

7 Global Service Providers

For further study.

Annex A (informative): Bibliography

ETSI TR 101 888-1: "Telecommunications and Internet protocol Harmonization Over Networks (TIPHON) Release 4; Security Test specifications Framework".

ETSI TS 101 883: "Telecommunications and Internet Protocol Harmonization Over Networks (TIPHON) Release 3; Technology Mapping; Implementation of TIPHON architecture using H.323".

History

Document history		
V4.1.1	May 2002	Publication