ETSI TS 102 822-7 V1.1.1 (2003-10)

Technical Specification

Broadcast and On-line Services: Search, select and rightful use of content on personal storage systems ("*TV-Anytime* Phase 1"); Part 7: Bi-directional metadata delivery protection

Reference

DTS/JTC-TVA-PH1-07

Keywords

broadcasting, content, TV, video

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, send your comment to:
editor@etsi.org

Copyright Notification

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 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 Joint Technical Committee (JTC) Broadcast of the European Broadcasting Union (EBU), Comité Européen de Normalisation ELECtrotechnique (CENELEC) and the European Telecommunications Standards Institute (ETSI).

The present document is part 7 of a multi-part deliverable covering Broadcast and On-line Services: Search, select and rightful use of content on personal storage systems ("*TV-Anytime* Phase 1"), as identified below:

Part 1: "Phase 1 Benchmark Features" (Informative);

Part 2: "System description";

Part 3: "Metadata";

Part 4: "Content referencing";

Part 5: Not currently applicable in *TV-Anytime* Phase 1;

Part 6: "Delivery of metadata over a bi-directional network";

Part 7: "Bi-directional metadata delivery protection".

Introduction

The present document document is based on a submission by the *TV-Anytime* forum (http://www.tv-anytime.org).

'*TV-Anytime* Phase 1' (TVA-1) is the first full and synchronized set of specifications established by the *TV-Anytime* Forum. TVA-1 features enable the search, selection, acquisition and rightful use of content on local and/or remote personal storage systems from both broadcast and online services.

The features are supported and enabled by the specifications for Metadata, Content Referencing, and Bi-directional Metadata Delivery Protection, TS 102 822-3 sub-parts 1 [1] and 2 [2], TS 102 822-4 [3], TS 102 822-6 [4] and TS 102 822-7 (the present document) respectively. All Phase 1 Features listed in TV035r6 are enabled by the normative *TV-Anytime* tools specifications. This list of Phase 1 Features is to be used as guidance to manufacturers, service providers and content providers regarding the implementation of the Phase 1 *TV-Anytime* specifications.

There will be further *TV-Anytime* phases published and Business Models for Post-Phase 1 are currently being defined to include Private and public domains, portable recordable media, super distribution (legal sharing of content between consumers), peripheral device support and mobile devices, amongst others.

1 Scope

The present document is the seventh document of a series of "S-documents" produced by the *TV-Anytime* Forum. These documents establish the fundamental specifications for the services, systems and devices that will conform to the *TV-Anytime* standard, to a level of detail that is implementable for compliant products and services.

As is common practice in such standardization efforts, these specification documents were preceded by requirements documents ("R-series"), which define the requirements for the *TV-Anytime* services, systems, and devices.

Congruent with the structure defined in the initial *TV-Anytime* Call for Contributions (TV014r3), these specifications are parsed into three major areas: Metadata, Content Referencing and Rights Management and Protection. Within these general areas, four specifications have been developed to date: Metadata (S-3), Content Referencing (S-4), Bi-directional Metadata (S-6) and Metadata Protection (S-7). A specification for Rights Management and Protection (S-5) is still under development. See the several *TV-Anytime* Calls for Contributions for more detail on the derivation and background of these categories and their respective roles in the *TV-Anytime* standardization process.

Two documents in the *TV-Anytime* S-series are intended to define the context and system architecture in which the standards in S-3, S-4, S-6 and S-7 are to be implemented in "Phase 1" of the *TV-Anytime* environment. The first document in the series (S-1) provides benchmark business models against which the *TV-Anytime* system architecture is evaluated to ensure that the specification enable key business applications. The next document in the series (S-2) presents the *TV-Anytime* System Architecture. These two documents are placed ahead of the other three for their obvious introductory value. (Note that S-1 and S-2 are largely informative documents, while the remainder of the S-series is normative. Also note that a "Phase 2" of the *TV-Anytime* process is currently underway, in which additional requirements and specifications that will build on Phase 1 are being developed. Readers are encouraged to check the *TV-Anytime* Forum's website at www.tv-anytime.org for the most recent status of its specifications.)

Although each of the S-series documents is intended to stand alone, a complete and coherent sense of the *TV-Anytime* system standard can be gathered by reading all of the Phase 1 specification documents in numerical order.

This scope of the present document, comprises the protection of metadata delivered via bi-directional networks.

The requirements for this technology are outlined as follows:

- Provide message integrity
- Authenticate service provider (entity that delivers metadata)
- Support bi-directional transport models
- Optional encryption

With the present document, TV-Anytime Forum mandates TLS as its baseline method of securing bi-directional delivery of metadata over point to point network connections.

While the present document addresses metadata transport security during delivery, it does not address persistent protection of metadata within the consumer space. To ensure persistent protection of metadata additional means of protection have to be applied in conjunction with the present document. The basic aim of the present document is to provide means to enable the delivery of trusted metadata to end-users.

However, end to end content protection is to be addressed in a separate specification (TS 102 822-5) as per the fundamental TV-Anytime RMP requirements: "*TV-Anytime* RMP-compliant systems, which aim to securely manage content from creation to final consumption shall accommodate the various needs of the different players in the value chain, specially enabling content owners and distributors to persistently protect their intellectual property and enforce content usage rules within the full content lifecycle." (RMP CFC section 3)

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.

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

[1] ETSI TS 102 822-3-1: "Broadcast and On-line Services: Search, select, and rightful use of content on personal storage systems ("TV-Anytime Phase 1"); Part 3: Metadata; Sub-part 1: Metadata schemas".

[2] ETSI TS 102 822-3-2: "Broadcast and On-line Services: Search, select, and rightful use of content on personal storage systems ("TV-Anytime Phase 1"); Part 3: Metadata; Sub-part 2: System aspects in a uni-directional environment".

[3] ETSI TS 102 822-4: "Broadcast and On-line Services: Search, select, and rightful use of content on personal storage systems ("TV-Anytime Phase 1"); Part 4: Content Referencing".

[4] ETSI TS 102 822-6: "Broadcast and On-line Services: Search, select, and rightful use of content on personal storage systems ("TV-Anytime Phase 1"); Part 6: Delivery of metadata over a bi-directional network; Sub-part 1: Service and transport".

[5] IETF RFC 1750 - December 1994: "Randomness Recommendations for Security", D. Eastlake, S. Crocker and J. Schille.

[6] IETF RFC 2104 - February 1997: "HMAC: Keyed-Hashing for Message Authentication" H. Krawczyk, M. Bellare and R. Canetti.

[7] IETF RFC 2246 - January 1999: "The TLS Protocol Version 1.0", T. Dierks and C. Allen.

[8] IETF RFC 3268 - June 2002: "Advanced Encryption Standard (AES) Ciphersuites for Transport Layer Security (TLS)", P. Chown.

[9] IETF RFC 3280 - April 2002: "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", R. Housley, W. Polk, W. Ford, and D. Solo.

[10] S. Moriai, Addition of Camellia Ciphersuites to Transport Layer Security (TLS), Internet-Draft, August 2002.

[11] E. Rescorla, SSL and TLS, Addison Wesley, 2001.

3 Definitions and abbreviations

3.1 Definitions

For the purposes of the present document, the following terms and definitions apply:

baseline: minimum set of functions that should be implemented to be compliant with *TV-Anytime* Forum specifications

bi-directional network: network that supports two way, point-to-point, one-to-many, and many-to-many data delivery

NOTE: The Internet is an example of such a network. A PDR may access a bi-directional network using its return path.

certificate: as part of the X.509 (a.k.a. ISO Authentication framework), certificates are assigned by a trusted Certificate Authority and provide a strong binding between a party's identity or some other attributes and its public key

handshake: initial negotiation between client and server that establishes the parameters of their transactions

metadata: generally, data about content, such as the title, genre and summary of a television programme. In the context of *TV-Anytime*, metadata also includes consumer profile and history data

service provider: aggregator and supplier of content which may include gateway and management roles

3.2 Abbreviations

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

AES	Advanced Encryption Standard
ASN.1	Abstract Syntax Notation. One
CA	Certificate Authority
CRL	Certificate Revocation List
DER	Distinguished Encoding Rules
DES	Data Encryption Standard
DH	Diffie-Hellman
DHE	Ephemeral Diffie-Hellman
DSS	Digital Signature Standard
F4	Fermat's F4 prime
HMAC	Hash-based Message Authentication Code
HTTP	Hypertext Transfer Protocol
IETF	Internet Engineering Task Force
IP	Internet Protocol
MAC	Message Authentication Code
MD5	Message Digest version 5
OID	Object Identifier
PKI	Public Key Infrastructure
RC4	Ron's Code 4
RDN	Relatively Distinguished Name
RFC	Request For Comments
RMP	Rights Management and Protection
RSA	Rivest, Shamir, Adleman algorithm
SHA-1	Secure Hash Algorithm version 1
SOAP	Simple Object Access Protocol
SSL	Secure Socket Layer
TCP	Transmission Control Protocol
TLS	Transport Layer Security
URN	Uniform Resource Names
XML	Extensible Markup Language

4 Use of TLS to Protect Bi-directional Delivery of Metadata

Secure Sockets Layer (SSL) and its IETF successor, Transport Layer Security (TLS), are the leading Internet security protocols, providing security for e-commerce, web services, and many other network functions. The primary goal of the TLS Protocol is to provide privacy and data integrity between two communicating applications. TLS also provides choices of cipher suites where data encryption may be disabled. Therefore, TLS can be used to ensure the data integrity of metadata conveyed between Service provider (Server) and User (Client).

4.1 TLS Protocol (informative)

4.1.1 Overview

The TLS protocol version 1.0 [7] is composed of two layers: the *TLS Record Protocol* and the *TLS Handshake Protocol*. At the lowest level, layered on top of some reliable transport protocol (e.g., TCP), is the TLS Record Protocol. The TLS Record Protocol provides connection security that has two basic properties:

Secrecy. Symmetric cryptography is used for data encryption (e.g. TripleDES, RC4, AES [8], Camellia [10] etc.) The keys for this symmetric encryption are generated uniquely for each connection and are based on a secret negotiated by another protocol (such as the TLS Handshake Protocol). The Record Protocol can also be used without encryption.

Message Integrity. Message transport includes a message integrity check using a keyed MAC. Secure hash functions (e.g. SHA-1, MD5, etc.) are used for MAC computations. The Record Protocol can operate without a MAC, but is generally only used in this mode while another protocol is using the Record Protocol as a transport for negotiating security parameters.

The TLS Record Protocol is used for encapsulation of various higher level protocols. One such encapsulated protocol, the TLS Handshake Protocol, allows the server and client to authenticate each other and to negotiate an encryption algorithm and cryptographic keys before the application protocol transmits or receives its first byte of data. The TLS Handshake Protocol provides connection security that has three basic properties:

The peer's identity can be authenticated using public key cryptography (e.g., RSA, DSS, etc.). This authentication can be made optional, but is generally required for at least one of the peers.

The negotiation of a shared secret is secure: the negotiated secret is unavailable to eavesdroppers, and for any authenticated connection the secret cannot be obtained, even by an attacker who can place himself in the middle of the connection.

The negotiation is reliable: no attacker can modify the negotiation communication without being detected by the parties to the communication.

One advantage of TLS is that it is application protocol independent. Higher level protocols can layer on top of the TLS Protocol transparently. The TLS standard, however, does not specify how protocols add security with TLS; the decisions on how to initiate TLS handshaking and how to interpret the authentication certificates exchanged are left up to the judgment of the designers and implementers of protocols which run on top of TLS.

4.1.2 Handshake

A connection is divided into two phases, the *handshake* and *data transfer* phases. The handshake phase authenticates the server and establishes the cryptographic keys which are used to protect the data to be transmitted. The handshake must be completed before any application data can be transmitted. The purpose of the handshake is:

1) Client and Server agree on a set of algorithms which will be used to protect the data.

2) They establish a set of keys which will be used by those algorithms.

3) Handshake always authenticates Server. It may optionally authenticate Client.

Overall process works like this (see figure 1):

1) Client sends Server a list of the algorithms it's willing to support, along with a random number used as input to the key generation process.

2) Server chooses a cipher out of that list and sends it back along with a certificate containing Server's public key. The certificate also provides Server's identity for authentication purposes and Server supplies a random number which is used as part of the key generation process.

3) Client verifies Server's certificate and extracts Server's public key. Client then generates a random secret string called the pre_master_secret and encrypts it using Server's public key. It sends the encrypted pre_master_secret to Server.

4) Client and Server independently compute the encryption and MAC keys from the pre_master_secret and Client and Server's random values.

5) Client sends a MAC of all the handshake messages to Server.

6) Server sends a MAC of all the handshake messages to Client.

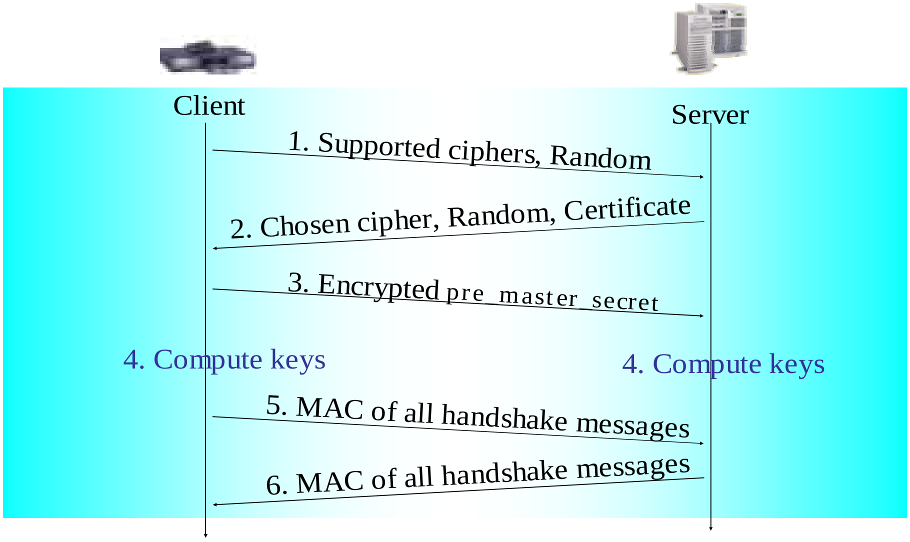
Client
Server
1. Supported ciphers, Random
2. Chosen cipher, Random, Certificate
3. Encrypted pre_master_secret
4. Compute keys
4. Compute keys
5. MAC of all handshake messages
6. MAC of all handshake messages

Figure 1: Overview of Handshake

4.2 Advantages of TLS

Advantages of TLS are:

1) Cryptographic security: TLS is a matured and well-deployed security protocol.

2) Extensibility: TLS provides a framework into which new public key and bulk encryption methods can be incorporated as necessary.

3) Application protocol independency: Higher level protocols can layer on top of the TLS Protocol transparently. HTTP, SOAP, and *TV-Anytime* XML, which are described as the Transport Protocols for delivery of Metadata over a bi-directional network [4], work with TLS (see figure 2). The only limitation for use of TLS is a reliable and bi-directional connection where the order of received packets or messages can be guaranteed.

4) Implementations availability: Free reference TLS implementations are available because it is currently an IETF proposed standard RFC.

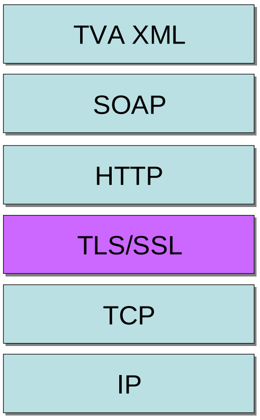

Figure 2: The bi-directional network transport stack

5 *TV-Anytime* TLS Profile for Bi-directional Metadata Delivery Protection

IMPORTANT: This implementation profile of TLS does not include client authentication.

5.1 TLS Cipher suites

5.1.1 Pre-Master Key Delivery

A *TV-Anytime* TLS implementation shall use Diffie-Hellman key agreement and should generate new Diffie-Hellman key pairs at both sides for each instantiation of the TLS handshake protocol. This is commonly called an Ephemeral-Ephemeral Diffie-Hellman exchange.

5.1.2 Digital Signature Algorithm

The algorithm shall be RSA. The RSA key size shall be at least 1 024 bits.

5.1.3 MAC (Message Authentication Code) Algorithm

The algorithm shall be HMAC-SHA1 [6]. The key for HMAC-SHA1 shall be 160 bits.

5.1.4 Cipher Algorithm

The required algorithms shall be both NULL encryption and AES. IETF defines a separate RFC for use of AES with TLS [8].

5.1.5 TLS Cipher suites

A *TV-Anytime* compliant application shall implement the cipher suites TLS_DHE_RSA_WITH_NULL_SHA and TLS_DHE_RSA_WITH_AES_128_CBC_SHA.

Cipher Suite	Key Exchange	Signature	Cipher	Hash	TVA Status
TLS_DHE_RSA_WITH_NULL_SHA	DH	RSA	NULL	SHA1	Required
TLS_DHE_RSA_WITH_AES_128_CBC_SHA	DH	RSA	AES	SHA1	Required
TLS_DHE_RSA_WITH_CAMELLIA_128_CBC_SHA	DH	RSA	Camellia	SHA1	Optional

5.2 Certificate Profile

The complexity of X.509 certificates is understood, but X.509 is required for standard TLS implementation. The certificate profile shall comply with RFC 3280 [9].

5.2.1 Certificate Trust Hierarchy

The required Certificate Trust Hierarchy is illustrated in figure 3:

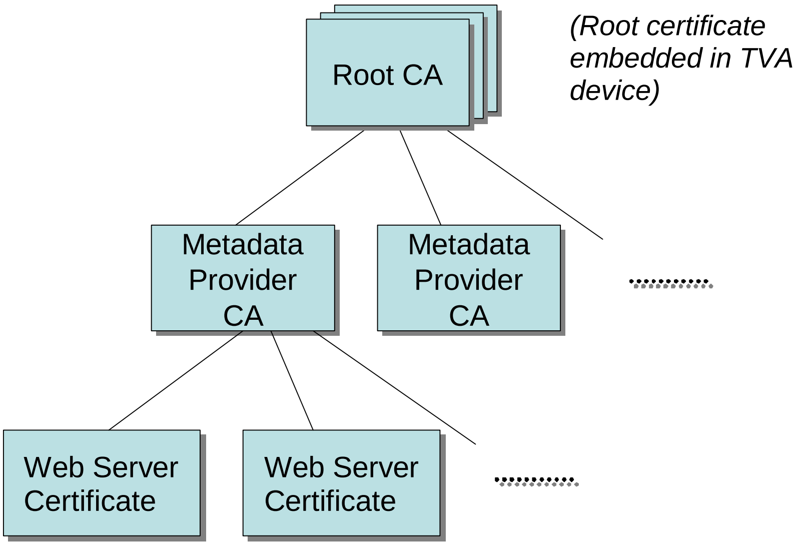

Figure 3: Certificate trust hierarchy

It is expected that the same Root CA(s) will be used to issue other certificates as well, not just for the purpose of secure TLS for metadata.

Each device receiving metadata shall include TVA-defined root CA certificates in order to validate the server's certificate chain. It is to be determined whether there will be more than one Root Certification Authority and if root CAs will be defined and/or operated by TVA or by other standards organizations.

The scope of a Metadata Provider CA is to issue web server certificates for a particular organization. It may be operated directly by that organization or may be operated on behalf of that organization by a commercial CA company.

5.2.1.1 *TV-Anytime* X.509 Certificate Profile Version

The version of the certificates shall be V3.

5.2.1.2 Public Key Type

Within X.509 certificates, a public key is defined as:

```
SubjectPublicKeyInfo ::= SEQUENCE {
        algorithm AlgorithmIdentifier,
        subjectPublicKey BIT STRING }
```

```
AlgorithmIdentifier ::= SEQUENCE {
        algorithm ALGORITHM.&id ({SupportedAlgorithms}),
        parameters ALGORITHM.&Type ({SupportedAlgorithms}{ @algorithm}) OPTIONAL }
```

The **AlgorithmIdentifier** consists of an OID (algorithm) that identifies the type of the public key followed by optional parameters field, where the syntax of the parameters is algorithm-dependent.

5.2.1.3 RSA Public Keys

The **SubjectPublicKeyInfo** algorithm Object Identifier (OID) is:

1.2.840.113549.1.1.1 (**rsaEncryption**).

An **AlgorithmIdentifier** for an RSA public key does not have any parameters.

The RSA public key is represented in ASN.1 as:

```
RSAPublicKey ::= SEQUENCE {
        modulus INTEGER,          -- n
        publicExponent INTEGER    -- e  }
```

The **RSAPublicKey** is first DER-encoded and then used as a value of a BIT STRING to form the **subjectPublicKey** member of **SubjectPublicKeyInfo**.

The public exponent for all RSA keys is F4 - 65537.

5.2.1.4 Extensions

The following five extensions shall be used as specified in the sections below.

5.2.1.4.1 subjectKeyIdentifier

The **subjectKeyIdentifier** extension shall be included in all CA certificates. This extension shall include the **keyIdentifier** value composed of the 160-bit SHA1 hash of the value of the BIT STRING **subjectPublicKey** (excluding the tag, length and number of unused bits from the ASN1 encoding). This extension shall be marked as non-critical.

5.2.1.4.2 authorityKeyIdentifier

The **authorityKeyIdentifier** extension shall be included in all certificates, with the exception of the root certificate and shall include a **KeyIdentifier** value that is identical to the **subjectKeyIdentifier** in the issuing CA certificate. This extension shall be marked as non-critical.

5.2.1.4.3 keyUsage

The **keyUsage** extension shall be used in all CA certificates and shall be marked as critical with a value of **keyCertSign** and **cRLSign.**

5.2.1.4.4 basicConstraint

The **basicConstraint** extension shall be used in all CA certificates and shall be marked as critical**.**

5.2.1.5 Signature Algorithm

The signature mechanism used shall be SHA-1 with RSA Encryption. The specific OID is 1.2.840.113549.1.1.5 and the signature parameters listed in the signed certificate are NULL.

5.2.1.6 SubjectName and IssuerName

When an X.500 attribute value contains characters that fall entirely within the character set utilized by **PrintableString**, the ASN.1 type **PrintableString** shall be used. In all other instances, **UTF8String** shall be used.

When encoding an X.500 Name:

1) Each RelativeDistinguishedName (RDN) shall contain only a single element in the set of X.500 attributes.

2) The order of the RDNs in an X.500 name shall be the same as the order in which they are presented in the present document.

5.2.2 Certificate Chain Validation Requirements

The certificate chain validation shall comply with RFC 3280 [9]. Additional TVA requirements are:

- Verify the extended key usage of the web server certificate with the value as specified
- Verify the OU= attribute inside the X.500 names to be "TVA" (without quotes).

5.2.3 (Metadata) Web Server Certificate Profile

Web Server Certificate	
Intended Usage	For TLS server certificates, where the server is capable of delivering TVA metadata.
Version	V3
Serial Number	Unique value for the given certificate issuer
Signature	**Sha1WithRSAEncryption** (OID: 1.2.840.113549.1.1.5)
Signed By	Metadata provider CA
Validity Period	Based on policy, not more than 10 years.

Subject Name Form (open)	C=<country code>
	O=<company name>
	OU=TVA
	CN=<DNS Name>

Public Key Type	**RsaEncryption**
Extensions	ExtendedKeyUsage[n](id-kp-serverAuth) AuthorityKeyIdentifier[n]

This table shows values of certificate fields that are specific to this type of a certificate (Web server) - there are other fields also present in an X.509 certificate with values that are not specific to this certificate profile.

The Metadata provider CA key size should be at least 2 048 bits. It should be larger than the key size used for signing messages.

In this table, the notation "[n]" after each extension means that it is not critical, while "[c]" would mean that the extension is critical. The **extendedKeyUsage** extension in this case (set to the OID value of id-kp-serverAuth) indicates that the certificate is used to authenticate a Web server.

5.2.4 Metadata Provider CA Certificate

Metadata Provider CA Certificate	
Intended Usage	To issue web server certificates and CRLs.
Version	v3
Serial Number	Unique value for the given certificate issuer
Signature	**sha1WithRSAEncryption** (OID: 1.2.840.113549.1.1.5)
Signed By	TVA Root CA
Validity Period	Based on policy.

Subject Name Form (open)	C=<country code>
	O=<company name>
	OU=TVA
	CN=Metadata Provider CA

Public Key Type	**RsaEncryption**
Extensions	AuthorityKeyIdentifier[n] SubjectKeyIdentifier[n] keyUsage[c](keyCertSign, cRLSign) basicConstraints[c](cA=true, pathLenConstraint=0)

5.2.5 *TV-Anytime* Root CA

The *TV-Anytime* Root CA is currently undefined and in the future may be defined within a different standards body. *TV-Anytime* places the following requirements on a Root CA:

1) The RSA key size shall be at least 2 048 bits.

2) The private key should be protected using FIPS 140-2 Level 3 (or above) physical security.

3) This CA shall be capable of issuing TVA-compliant Certificate Revocation Lists.

5.2.6 Certificate Revocation

Certificate revocation is a critical and at the same time operationally complex part of PKI (Public Key Infrastructure). In the case that a public/private key pair is compromised (e.g., stolen) or in some cases if a business contract that was signed to obtain a certificate has been violated, a certificate needs to be revoked in order to make sure that it will not be accepted by anyone again.

All CRLs (Certificate Revocation Lists) shall be available to *TV-Anytime* devices to be downloadable over an HTTP connection from a URN.

All *TV-Anytime* compliant Certification Authorities shall be capable of issuing CRLs.

When a certificate is revoked, its serial number shall be added to a CRL that is generated by the same Certification Authority that previously issued the corresponding certificate. The format of a CRL is defined by the X.509 standard and its ASN.1 encoding is as follows:

```
CertificateList  ::=  SEQUENCE  {
        tbsCertList          TBSCertList,
        signatureAlgorithm   AlgorithmIdentifier,
        signatureValue       BIT STRING
}

TBSCertList  ::=  SEQUENCE  {
        version                 Version OPTIONAL,
                                     -- if present, shall be v2
        signature               AlgorithmIdentifier,
        issuer                  Name,
        thisUpdate              Time,
        nextUpdate              Time OPTIONAL,
        revokedCertificates     SEQUENCE OF SEQUENCE  {
             userCertificate         CertificateSerialNumber,
             revocationDate          Time,
             crlEntryExtensions      Extensions OPTIONAL
                                           -- if present, will be v2
        }  OPTIONAL,
        crlExtensions           [0]  EXPLICIT Extensions OPTIONAL
```

```
                                                -- will be v2
        }
```

The supported signature algorithms for a CRL shall be the same as what is defined for certificates.

The optional **nextUpdate** field shall be used within *TV-Anytime*-defined CRLs and indicates that time when a more up-to-date version of this CRL will become available. At a time **nextUpdate** + <CRL Grace Period>, the old CRL shall no longer be accepted, where CRL Grace Period is a configurable parameter.

The optional **crlExtensions** field shall be included and shall contain the following extensions:

- **authorityKeyIdentifier** (not critical). It shall be set to the **subjectKeyIdentifier** from the CA certificate)
- CRL Number (not critical). This is a CRL sequence number that starts with 0 and is incremented by 1 for each subsequent CRL issued by the same CA.

5.3 Device requirements

Each device receiving metadata shall include TVA-defined root CA certificates in order to validate the server's certificate chain. It is to be determined whether there will be more than one Root Certification Authority and if root CAs will be defined and/or operated by TVA or by other standards organizations.

The TLS protocol depends on an availability of a strong random number generator in a device for the Diffie-Hellman key exchange. RFC 1750 [5] provides some recommendations for good random number generation in both hardware and software.

It is recommended that a means to renew or update a list of root certificates in TVA devices is implemented by manufacturers. This could take the form of secure code download or update by an approved service centre.

A device receiving metadata shall attempt to obtain up-to-date, non-expired CRLs from a specified URN and shall utilize the CRLs to validate TLS server certificates.

Annex A (informative): Bibliography

Documents are available from the *TV-Anytime* web site http://www.tv-anytime.org.

"R-1: The *TV-Anytime* Environment" (TV035r6)

List of figures

History

Document history		
V1.1.1	October 2003	Publication