COMMON ISIS-MTT SPECIFICATIONS
FOR INTEROPERABLE PKI APPLICATIONS

FROM T7 & TELETRUST

SPECIFICATION

OPTIONAL PROFILE

SIGG-PROFILE

VERSION 1.1 – 16 MARCH 2004

Contact Information

ISIS-MTT Working Group of the TeleTrusT Deutschland e.V.: www.teletrust.de

The up-to-date version of ISIS-MTT can be downloaded from the above web site, from www.isis-mtt.org or from www.isis-mtt.de

Please send comments and questions to isismtt@teletrust.de

Chief Editors:

Jürgen Brauckmann

Alfred Giessler

Tamás Horváth

Hans-Joachim Knobloch

Document History

VERSION DATE	CHANGES
1.0 30.09.2001	First public edition
1.0.1 15.11.2001	A couple of editorial and stylistic changes: - references to SigG-specific issues eliminated from core documents - core documents (Part 1-7) and optional profiles have been separated in different PDF documents.
1.0.2 19.07.2002	Several editorial changes and bug-fixes. The most relevant changes affecting technical aspects are: 1) The Dname attribute *nameDistinguisher*, used in legacy systems and older PKCs, MUST be supported by processing applications in *issuer* and *subject* names. (T1.#18) 2) Dname attributes in *Procuration* limited to RFC3039 attributes. (T4.#7,[2]) 3) *QcEuLimitValue* may be included in an AC as attribute in place of *MonetaryLimit*. (T10.#10,[2]) 4) Table 12 contains all OIDs defined for ISIS-MTT 5) Profiling information with respect to Part 5 added to adopt validity model to SigG. (Section 3)
1.0.2 11.08.2003	Incorporated all changes from Corrigenda version 1.2
1.1 16.03.2004	Several editorial changes and bug-fixes. The most relevant changes affecting technical aspects are: 1) The policy identifier *id-isismtt-cp-sigGconform* has been renamed to *id-isismtt-cp-accredited* in order to better reflect the correct semantics. 2) Added a new extension/attribute *AdditionalInformation*. 3) Added a new section about algorithms 4) The permitted size if an *ICCSN* was increased to 20 octets (corresponding to the decimal character representation of a 64 bit value). 5) Definitions of ISIS-MTT private attributes for attribute certificates have been moved from the optional SigG Profile to core Part 1. 6) Key usage has been aligned with ETSI TS 102 280. 7) The qualified certificate statement QcSSCD has been added to the list of QCs. 8) Added profession OID values for the *Admission* attribute.

Table of Contents

1 Preface

The *German Signature Act* (SigG) and the *Ordinance on Digital Signatures* (SigV) raise a couple of special requirements on technical components as well as on the certificate policy of certification service providers (CSPs). This profile addresses these technical requirements. These requirement affect certificate contents, CSP service protocols as well as the validity model, implied by the SigG. Besides providing means to fulfil technical requirements, induced by the SigG, this profile specifies new certificate contents, in form of private extensions and attributes, required in common business cases, that rely on the legal instruments of the SigG.

This profile is intended for system and application developers who intend to design components that:

- fulfil the requirements induced by the SigG and the SigV on technical means;
- should either be employed in the technical arsenal of CSPs that provide qualified services in the context of SigG and either aspire an *accreditation* in the sense of the SigG, or intend to operate without an accreditation;
- or in end-entity components in SigG-related applications that rely on the qualified services of either accredited or non-accredited CSPs.
- interoperate with PKIs and components designed to comply with the ISIS-MTT Core Documents.

The association T7 of accredited CSP commits itself to this profile, i.e. services and technical components provided by accredited CSP MUST comply with this profile. Non-accredited CSP and third-party software manufactures MAY choose to comply with this profile.

1.1 Interoperability Aspects

The German Signature Act (Signaturgesetz, [SigG]) defines the general framework for so-called qualified electronic signatures that can be used in legal actions. The SigG has been first passed in 1997 and has been modified in 2001 to comply with *the Directive on Electronic Signatures of the European Community* [ECDir]. The signature law and the ordinance on its technical realization (Signaturverordnung, [SigV01]) put very strong security requirements on the entire public key infrastructure providing means for “qualified electronic signatures”, i.e. on signature devices, signature software as well as CA services. The GISA – German IT Security Agency (Bundesamt für Sicherheit in der Informationstechnik, BSI) has issued a “Signature Interoperability Specification” (SigI), promoting uniform signature and certificate formats for SigG-related applications. Companies providing qualified CA services have founded the association “T7” and have issued the standard “Industrial Signature Interoperability Standard” (ISIS), which is an enhancement of a subset of SigI.

The EU-Directive and the German Signature Act classifies electronic signatures as follows:

1. “**electronic signature**” means data in electronic form which are attached to or logically associated with other electronic data and which serve as a method of authentication;
2. “**advanced electronic signature**” means an electronic signature which meets the following requirements:

   (a) it is uniquely linked to the signatory;

   (b) it is capable of identifying the signatory;

(c) it is created using means that the signatory can maintain under his sole control; and

(d) it is linked to the data to which it relates in such a manner that any subsequent change of the data is detectable;

3. "**qualified electronic signature**" means an advanced electronic signatures which:

(a) is based on a "qualified certificate" that was valid at the time of signature-creation;

(b) was generated by a "secure-signature-creation device";

Based on MailTrusT (a specification of TeleTrusT for PKI-based secure email), SigI and ISIS, this ISIS-MTT specification aims to provide a common specification for client applications that integrate secure email or other functions and qualified (i.e. SigG-conforming) signature functions. Interoperability among client components as well as CA-services should be provided regardless of the aspired level of security or trust. This characteristics is also referred to as *vertical interoperability*.

More in detail this means:

- components offering the same security level MUST be unconditionally interoperable;
- components offering different security level must be interoperable as far as possible: qualified components MUST be conform with any lower security levels. For example, qualified client software (containing a "secure signature-creation device" and a "secure signature-verification device" in the sense of SigG) MUST be able to verify signatures generated by any other ISIS-MTT-compliant components, where the user must be given a note about the actual assumable level of trust. Non-qualified components are STRONGLY RECOMMENDED to support data structures (e.g. qualified certificates) and CA services as described in this document. Accordingly, non-qualified client software should be able to verify qualified signatures, where of course, the verification can be trusted only to the same extent as the client environment can be trusted.
- Interoperability with common Internet components and data formats based on PKIX standards is enforced.
- *Qualified* components and related data formats (the subject of this SigG-Profile) are specified in a manner to meet the requirements of the SigG and of the SigV and to fully comply with the standards of ETSI (European Communications Standards Institute).

In order to achieve the above interoperability and conformity goals, a special "sub"-profile of ISIS-MTT for components and services related to qualified signatures will be defined in this document. Thus, the SigG-Profile is (up to the optional validity model introduced here) fully compliant with the more general ISIS-MTT profile.

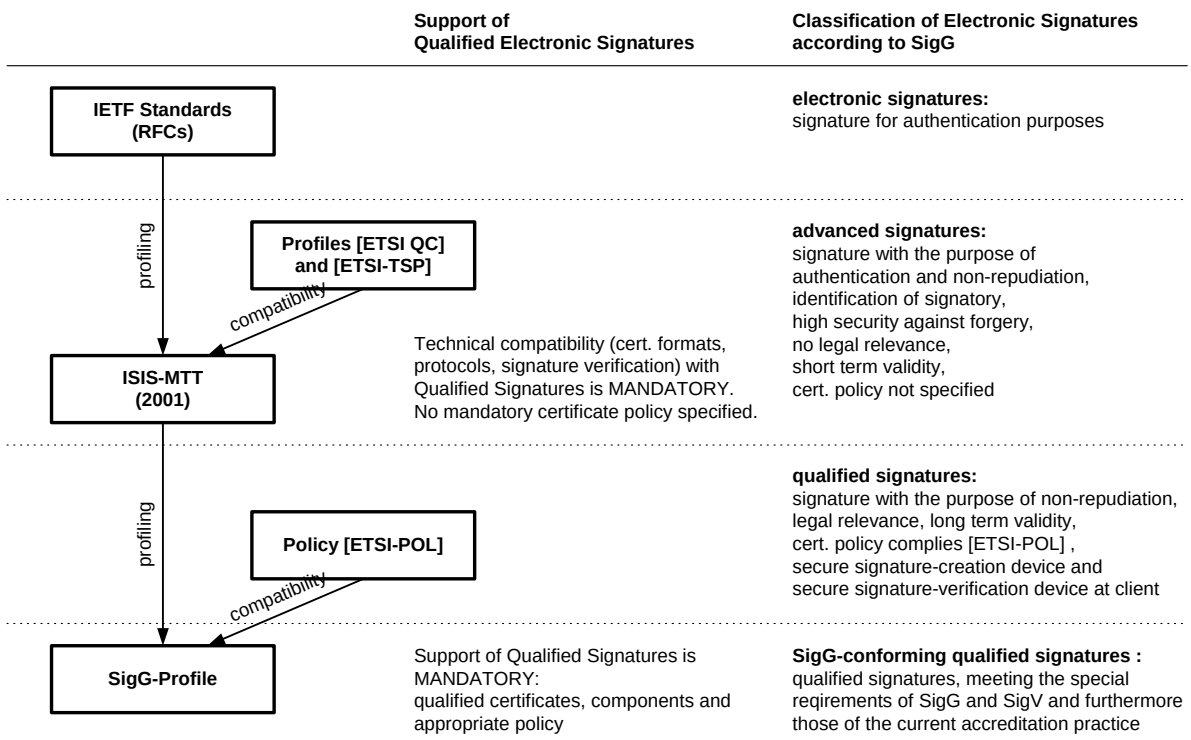

Figure 1: The relation of electronic signature standards and security levels

1.2 Requirements on technical components

The SigG and the SigV induces a couple of special requirement on technical components (especially certificates and directory services) used SigG-conforming services or SigG-related applications. Among many others, the following requirements apply:

(1) the validity of qualified certificates is limited to 5 years (SigV §14 (3))

(2) long term verifiability: it must be possible to verify a signature after expiry and even after revocation of relevant certificates. This period is set at a minimum of 5 years for non-accredited CAs and at a minimum of 30 years for accredited CAs (SigV §4 (1) and (2))

(3) a flat, 3-layer certification hierarchy: a governmental agency at the top level (responsible for policies, accreditation and subsequent supervision), certification service providers at the middle level (providing CA services for end entities, but not permitted to issue certificates for other CAs) and end entities at the bottom.

(4) SigG §19 (5): The user certificates issued by a conforming CA remain valid even if the accreditation of the issuing CA gets revoked. In this case all certificates of the CA must be revoked.

(5) SigG §8 (1): A retrospective revocation of certificates is forbidden.

(6) SigG §5 (1) distinguishes between user certificates that are *verifiable* from those being *downloadable*. While providing status information about all certificates, directory services must not publish only *verifiable* certificates.

2 Certificate and CRL Formats

The special requirements on certificate and CRL contents are collected in the following tables. Profiling information on specific data components are linked via references to corresponding definitions in Part 1. Note that certificate and CRL formats conforming this SigG-Profile are fully compliant with the more general ISIS-MTT Core profile, laid down in Part 1.

2.1 Public Key Certificate Format

Table 1 : Special requirements on SigG-conforming qualified PKCs

#	DATA FIELD	SEMANTICS AND SIGG PROFILING INFORMATION (CONSTRAINT OR ENHANCEMENT WITH RESPECT TO CORE)	CRITI-CAL	SUPPORT		REFE-RENCE	NO TES
				GEN	PROC		
0	validity	According to the ordinance on signatures [SigV01], §14, the validity of qualified certificates is limited to 5 years.		++	++	P1.T2.#6	
	STANDARD EXTENSIONS						
1	KeyUsage	The following restriction applies in end-entity qualified signature certificates: the *nonRepudiation* bit and only this bit MUST be set if these certificates are used to validate commitment to signed content, such as electronic signatures on agreement and or transactions (ETSI TS 102 280, 4.4.3). These certificates MUST NOT be used for other purposes, like authentication or encryption. The *nonRepuditaion* and *digitalSignature* bits MAY be combined if these certificates are to be used for other purposes.	++ (RFC 3039 +)	++ (RFC 3039 +-)	++ (RFC n.a.)	P1.T12	
2	CertificatePolicies	Legacy systems use the *CertificatePolicies* extension to mark qualified certificates and to recognize this fact in components.	- (RFC 2459+-)	+-	++	P1.T14	[1]
3	id-isismtt-cp-accredited	The *id-isismtt-cp-accredited* OID indicates that the certificate is a qualified certificate according to [EUDIR], which additionally conforms the special requirements of the SigG and has been issued by an accredited CA. This latter means that the security of all relevant components (CA, DIR, smartcards etc.) have been proven by an independent accredited laboratory and provide an appropriately high level of trust according to ITSEC. The voluntary accreditation process for CAs is described in §15 and §16 of the novel signature act [SigG] from 2001. Since many of the currently used QCs do not include a *QCStatement*, SigG-conforming		+-	++	P1.T14	

		components MUST be able to evaluate both the *id-isismtt-cp-accredited* policy OID and *QCStatements*. New qualified certificates MUST be issued with a proper *QCStatement* (see #6) and MAY include the *id-isismtt-cp-accredited* policy OID to indicate *voluntary accreditation* of the issuing CA. Non-accredited CAs issuing SigG-conforming certificates MUST NOT use this OID, but SHOULD mark the certificate by including a proper policy OID in *QCStatements*. ATTENTION! Currently used qualified certificates have been issued including merely the *id-isismtt-cp-accredited* policy OID (i.e. no *QCStatement* present). As *voluntary accreditation* of the CA implies that all issued certificates are qualified ones, components MUST be able to recognize this fact in the absence of a *QCStatement*.					
4	`SubjectDirectoryAttributes`	Qualified PKCs MAY include legal identification data of the subject in the *subjectDirectoryAttributes* extension. The same kind of information MAY be included in attribute certificates as separate attribute (i.e. in the 'attributes' field instead of an extension) but using the same *SubjectDirectoryAttributes* syntax. The following attributes MAY be inserted by compliant CAs: Standard attributes: *commonName, surname, givenName, title, postalAddress* (with the address of permanent residence) RFC3039 attributes: *dateOfBirth, placeOfBirth, gender, countryOfCitizenship, countryOfResidence,* ISIS-MTT attribute: *nameAtBirth* SigG-conforming components MUST be prepared to process these Dname attribute types. Clients SHOULD be able to process all attribute types that may occur in the subject field. According to the German law, the following items are required for a legally valid identification record: *surname, givenName, title, dateOfBirth, placeOfBirth, nameAtBirth, countryOfCitizenship, postalAddress*. No attributes have yet been introduced for further data items of a German ID card, like ID card number, height, colour of eyes, issuing institution, issuing date.	--	+-	CORE+ SigP++	P1.T17	
	RFC3039 (QC) PRIVATE EXTENSIONS						
5	`QCStatements`	*QCStatements* (Qualified Certificate Statements) extension MUST be recognized and evaluated by SigG-conforming components.	- (RFC 3039 +-)	CORE+- SigP++	CORE+ SigP++	P1.T25	[1]

6	id-etsi-qcs-QcCompliance	In accordance with [ETSI-QC], qualified signature certificates to be used in the context of the signature act (SigG) MUST include a *QCStatement* (Qualified Certificate Statement) extension with this OID. This applies to end entity as well as to CA certificates. The meaning of this OID is that the certificate policy is compliant with the policy described in [ETSI-POL]. This QC statement is RECOMMENDED to be included in SigG-conforming certificates issued until June 30, 2005 and it MUST be present in certificates issued later.		CORE+ SigP++	CORE+ SigP++	P1.T25	
6a	id-etsi-qcs-QcSSCD	In accordance with [ETSI-QC], qualified signature certificates to be used in the context of the signature act (SigG) MAY include a *QCStatement* (Qualified Certificate Statement) extension with this OID. This applies to end entity as well as to CA certificates. The meaning of this OID to indicate that the CA vouches that the private key associated with the public key in the certificate is stored in an SSCD according to Annex III of [ECDIR].		+-	CORE+ SigP++	P1.T25	
7	id-etsi-qcs-QcLimitValue	The *QcLimitValue* statement SHOULD be used in new certificates in place of the extension/attribute *MonetaryLimit*. Nevertheless, *MonetaryLimit* MAY still be used until December 31, 2003. After this date, *MonetaryLimit* MUST NOT be used any longer. For the sake of backward compatibility with certificates already in use, components MUST support *MonetaryLimit* (as well as *QcEuLimitValue*). If both *QcEuLimitValue* and *MonetaryLimit* occur in the same certificate, they MUST assert the same value and currency. A certificate SHOULD use only one form.		+-	CORE+ SigP++	P1.T25	
8	id-etsi-qcs-QcRetentionPeriod	The *QcRetentionPeriod* statement indicates CAs or a relevant name registration authority retains external information (i.e. registration documents) about the owner of qualified certificates. This information allows identifying the physical person in case of dispute. SigG-compliant client MUST support this statement.		+-	CORE+ SigP++	P1.T25	
	RFC2560 (OCSP) **PRIVATE EXTENSIONS**						
9	OCSPNocheck	OCSP clients need to know how to check that an authorized OCSP responder's certificate has not been revoked. A CA MAY specify that an OCSP client can trust a responder for the lifetime of the responder's certificate, i.e. the client need no CRL information. The CA does so by including the extension *OCSPNocheck*. SigG-compliant CAs MUST provide status information on the responder's certificate. Hence, this extension MUST NOT be included in qualified certificates.	-	CORE+- SigP--	+	P1.T26	
	ISIS-MTT SIGG **PRIVATE EXTENSIONS**						**[2]**

10	`LiabilityLimitationFlag`	Indicates that an attribute certificate exists, which limits the usability of this public key certificate. Whenever verifying a signature with the help of this certificate, the content of the corresponding attribute certificate should be concerned. This extension MUST be included in a PKC, if a corresponding attribute certificate (having the PKC as base certificate) contains some attribute that restricts the usability of the PKC too. Attribute certificates with restricting content MUST always be included in the signed document.	-	SigP+-	SigP++	SigP.T2	[1]
11	`DateOfCertGen`	The CA MAY include the *DateOfCertGen* extension, if the certificate is issued right before its validity period, i.e. the signing time T_s lies before *validity.notBefore*. Otherwise the extension SHOULD NOT be included. This information plays a role, if a relying component decides to validate the certificate according to the SigG-specific validity model, described in Section 6.	--	SigP+-	SigP++	SigP.T3	
12	`Procuration`	This attribute may also be used as an extension. As an extension it is single-valued. At the current legal situation, only natural persons and no juridical persons (organizations) may be substituted.	--	SigP+-	SigP++	P1.T29a	
13	`Admission`	This attribute may also be used as an extension.	--	SigP+-	SigP++	P1.T29b	[3]
14	`MonetaryLimit`	The *QcEuMonetaryLimit* QC statement MUST be used in new certificates in place of the extension/attribute *MonetaryLimit* since January 1, 2004. For the sake of backward compatibility with certificates already in use, SigG conforming components MUST support *MonetaryLimit* (as well as *QcEuLimitValue*).	-	SigP--	SigP++	P1.T29c	[1]
15	`DeclarationOfMajority`	This attribute may also be used as an extension.	--	SigP+-	SigP++	P1.T29d	
16	`Restriction`	This attribute may also be used as an extension.	-	SigP+-	SigP++	P1.T29e	[1]
16a	`AdditionalInformation`	This attribute may also be used as an extension.	-	SigP+-	SigP++	P1.T29f	[1]
17	`ICCSN`	Smartcard serial number, to bind a public key to a smart card that stores the corresponding private key.	--	SigP+-	SigP+-	SigP.T9	
	DNAME ATTRIBUTES						
18	`nameDistinguisher`	Legacy systems, software and certificates use this Dname attribute in conjunction with the OID id-*isismtt-at-nameDistinguiser* to distinguish Dnames if different entities, if their Dnames are otherwise identical. [RFC3039] and ISIS-MTT recommends using the attribute *serialNumber* for this purpose. For backward compatibility, S	--	--	SigP++		
[1]	Notes on criticality: For the sake of *vertical interoperability*, these extensions SHOULD NOT be marked critical, in spite of the fact that their contents restrict the usability of the certificate in some way. As these information are extremely relevant in verifying the legal validity of the signature, SigG-conforming components MUST evaluate them.						
[2]	All SigG-specific extensions, except ICCSN, MUST be evaluated by SigG-conforming components.						

[3] Profession OIDs should always be defined under the OID branch of the responsible naming authority.
At the time of this writing, the work group “Recht, Wirtschaft, Steuern” (“Law, Economy, Taxes”) is registered as the first naming authority under the OID *id-isis-at-namingAuthorities* and defined the following profession OIDs:

`id-isismtt-at-namingAuthorities-RechtWirtschaftSteuern {id-isis-at-namingAuthorities 1}`

Rechtsanwältin	`{id-isismtt-at-namingAuthorities-RechtWirtschaftSteuern 1}`
Rechtsanwalt	`{id-isismtt-at-namingAuthorities-RechtWirtschaftSteuern 2}`
Rechtsbeistand	`{id-isismtt-at-namingAuthorities-RechtWirtschaftSteuern 3}`
Steuerberaterin	`{id-isismtt-at-namingAuthorities-RechtWirtschaftSteuern 4}`
Steuerberater	`{id-isismtt-at-namingAuthorities-RechtWirtschaftSteuern 5}`
Steuerbevollmächtigte	`{id-isismtt-at-namingAuthorities-RechtWirtschaftSteuern 6}`
Steuerbevollmächtigter	`{id-isismtt-at-namingAuthorities-RechtWirtschaftSteuern 7}`
Notarin	`{id-isismtt-at-namingAuthorities-RechtWirtschaftSteuern 8}`
Notar	`{id-isismtt-at-namingAuthorities-RechtWirtschaftSteuern 9}`
Notarvertreterin	`{id-isismtt-at-namingAuthorities-RechtWirtschaftSteuern 10}`
Notarvertreter	`{id-isismtt-at-namingAuthorities-RechtWirtschaftSteuern 11}`
Notariatsverwalterin	`{id-isismtt-at-namingAuthorities-RechtWirtschaftSteuern 12}`
Notariatsverwalter	`{id-isismtt-at-namingAuthorities-RechtWirtschaftSteuern 13}`
Wirtschaftsprüferin	`{id-isismtt-at-namingAuthorities-RechtWirtschaftSteuern 14}`
Wirtschaftsprüfer	`{id-isismtt-at-namingAuthorities-RechtWirtschaftSteuern 15}`
Vereidigte Buchprüferin	`{id-isismtt-at-namingAuthorities-RechtWirtschaftSteuern 16}`
Vereidigter Buchprüfer	`{id-isismtt-at-namingAuthorities-RechtWirtschaftSteuern 17}`
Patentanwältin	`{id-isismtt-at-namingAuthorities-RechtWirtschaftSteuern 18}`
Patentanwalt	`{id-isismtt-at-namingAuthorities-RechtWirtschaftSteuern 19}`

See http://www.teletrust.de/anwend.asp?Id=30200&Sprache=E_&HomePG=0 for an application form and http://www.teletrust.de/links.asp?id=30220,11 for an overview of registered naming authorities.

Table 2: LiabilityLimitationFlag

#	ASN.1 DEFINITION	SEMANTICS	SUPPORT		REFERENCES		NOTES
			GEN	PROC	RFC	ISISMTT	
1	**id-isismtt-at-LiabilityLimitationFlag** OBJECT IDENTIFIER ::= {0 2 262 1 10 12 0}	OID for extension *LiabilityLimitationFlag*			n.a.	SigP.T12	
2	**LiabilityLimitationFlagSyntax ::=** BOOLEAN	The extension SHOULD be present, if it has value *true*.	SigP+-	SigP++	n.a.		

Table 3: DateOfCertGen

#	ASN.1 DEFINITION	SEMANTICS	SUPPORT		REFERENCES		NOTES
			GEN	PROC	RFC	ISISMTT	
1	**id-isismtt-at-dateOfCertGen** OBJECT IDENTIFIER ::= {id-isismtt-at 1}	OID for extension *DateOfCertGen*			n.a.	SigP.T12	
2	**DateOfCertGenSyntax ::=** GeneralizedTime	Date of the generation of the certificate. The format YYYYMMDDhhmmssZ MUST be used.	SigP+-	SigP++	n.a.		

Table 4: Obsoleted by Part 1 Table 29a

Table 5: Obsoleted by Part 1 Table 29b

Table 6: Obsoleted by Part 1 Table 29c

Table 7: Obsoleted by Part 1 Table 29d

Table 8: Obsoleted by Part 1 Table 29e

Table 9: ICCSN

#	ASN.1 DEFINITION	SEMANTICS	SUPPORT		REFERENCES		NOTES
			GEN	PROC	RFC	ISISMTT	
1	`id-isismtt-at-iCSSN OBJECT IDENTIFIER ::= {id-isismtt-at 6}`	OID for extension *ICCSN*			n.a.	SigP.T12	
2	`ICCSNSyntax ::= OCTET STRING (SIZE(8..20))`	Serial number of the smart card containing the corresponding private key	+-	+-	n.a.		[1]
[1]	**SIGG-PROFILE:** This information may be particularly useful in business applications, where the workflow of issuing a smartcard starts with producing the card, that will be bound to a person only a later stage. In such applications, the ICCSN can serve as the main reference to the client's data during the entire life cycle of the smartcard, e.g. for logging or billing particular transactions carried out by the card holder.						

2.2 Attribute Certificate Format

Table 10: Special requirements on SigG-conforming qualified attribute certificates

#	DATA FIELD	SIGG PROFILING INFORMATION (CONSTRAINT OR ENHANCEMENT WITH RESPECT TO CORE)	CRITICAL O.MULTI-VALUED	SUPPORT GEN	SUPPORT PROC	REFE-RENCE	NOTES
	BASIC AC FIELDS						
1	subject	SigG-conforming attribute certificates may exist only in conjunction with a key certificate (the base certificate) of the subject. Hence, such certificates MUST use the *baseCertificateID* option when filling the subject field.		++	++	P1.T28.#3	
2	attrCertValidityPeriod	According to the ordinance on signatures [SigV01], §7, the validity of an attribute certificate ends with the validity of the accompanying base certificate. Therefore the maximum validity period is 5 years.		++	++	P1.T28.#9	
	ISIS-MTT SIGG PRIVATE EXTENSIONS		**CRITICAL**				
3	DateOfCertGen	The same applies as to the corresponding PKC extension. See T1.#11	--	SigP+-	SigP++	T1.#11, SigP.T3	
	ISIS-MTT PRIVATE ATTRIBUTES		**MULTI-VALUED**				
4	Procuration	The same applies as to the corresponding PKC extension. See Table 1.#12	Y	+-	CORE +- SigP++	T1.#12, P1.T29a	
5	Admission	The same applies as to the corresponding PKC extension. See Table 1.#13	N	+-	CORE +- SigP++	T1.#13, P1.T29b	
6	MonetaryLimit	The same applies as to the corresponding PKC extension. See Table 1.#14	N	--	CORE +- SigP++	T1.#14, P1.T29c	[1]
7	DeclarationOfMajority	The same applies as to the corresponding PKC extension. See Table 1.#15	N	+-	CORE +- SigP++	T1.#15, P1.T29d	
8	Restriction	The same applies as to the corresponding PKC extension. See Table 1.#16	Y	+-	CORE +- SigP++	T1.#11, P1.T29e	[1]
8a	AdditionalInformation	The same applies as to the corresponding PKC extension. See Table 1.#16a	Y	+-	CORE +- SigP++	T1.#11, P1.T29f	[1]
9	SubjectDirectoryAttributes	The same applies as to the corresponding PKC extension. See Table 1.#4	N	+-	CORE +- SigP++	T1.#4 P1.T17	[3]
10	QcEuLimitValue id-etsi-qcs-QcLimitValue	This attribute MUST be processed by conforming applications.	N	+-	CORE +- SigP++	P1.T25.#13	

[1]	**SIGG-PROFILE:** In conjunction with setting the *LiabilityLimitationFlag* in the base certificate, this specification allows issuing attribute certificates that restrict the usability of the base certificate.

2.3 CRL Format

Table 11: Special requirements on CRLs of SigG-conforming qualified certificates

#	DATA FIELD	SIGG PROFILING INFORMATION (CONSTRAINT OR ENHANCEMENT WITH RESPECT TO CORE)	CRITI-CAL	SUPPORT		REFE-RENCE	NO TES
				CA	CLIENT		
	CRL ENTRY EXTENSIONS						
1	`CRLReason`	Only the reason codes *keyCompromise*, *cACompromise*, *affiliationChanged*, *cessationOfOperation* are allowed. As revoked SigG-conforming certificates cannot be released again, the reasons *certificateHold* and *removeFromCRL* never apply. As SigG-conforming certificates cannot be prolonged, the code *superseded* never applies.	--	+-	+-	P1.T38	
2	`HoldInstruction`	As SigG-conforming certificates MUST NOT be suspended (status *certificateHold*) in directories, this extension MUST NOT occur in CRL entries corresponding to such certificates.	--	CORE+- SigP--	+-	P1.T39	

2.4 ISIS-MTT Object Identifiers

The following table lists all ASN.1 object identifiers introduced in the ISIS-MTT Specification Core and in this SigG-Profile. Furthermore, obsolete OIDs, defined in [ISIS] or earlier ISIS-MTT versions, are listed too. These OID values are reserved and MUST NOT be used for any other purpose.

Table 12: ISIS-MTT Object Identifiers

#	ASN.1 Definition		Semantics	Support		References		Notes
				Gen	Proc	RFC	ISISMTT	
1	**id-isismtt**	OBJECT IDENTIFIER ::= {1 3 36 8 }		++	++	n.a.		
2	**id-isismtt-cp**	OBJECT IDENTIFIER ::= {id-isismtt 1}	Branch for policies			n.a.	#1	
3	**id-isismtt-cp-accredited**	OBJECT IDENTIFIER ::= {id-isismtt-cp 1}		+-	++	n.a.	#2	
4	**id-isismtt-at**	OBJECT IDENTIFIER ::= {id-isismtt 3}	Branch for attrs. and extensions			n.a.	#1	
4	**id-isismtt-at-dateOfCertGen**	OBJECT IDENTIFIER ::= {id-isismtt-at 1}		+-	++	n.a.	SigP.T3	
5	**id-isismtt-at-procuration**	OBJECT IDENTIFIER ::= {id-isismtt-at 2}		+-	++	n.a.	P1.T29a	
6	**id-isismtt-at-admission**	OBJECT IDENTIFIER ::= {id-isismtt-at 3}		+-	++	n.a.	P1.T29b	
7	**id-isismtt-at-monetaryLimit**	OBJECT IDENTIFIER ::= {id-isismtt-at 4}		+-	++	n.a.	P1.T29c	
8	**id-isismtt-at-declarationOfMajority**	OBJECT IDENTIFIER ::= {id-isismtt-at 5}		+-	++	n.a.	P1.T29d	
9	**id-isismtt-at-iCSSN**	OBJECT IDENTIFIER ::= {id-isismtt-at 6}		+-	++	n.a.	SigP.T9	
10	**id-isismtt-at-pKReference**	OBJECT IDENTIFIER ::= {id-isismtt-at 7}	obsolete	--	-	n.a.	obsolete	
11	**id-isismtt-at-restriction**	OBJECT IDENTIFIER ::= {id-isismtt-at 8}		+-	++	n.a.	P1.T29e	
12	**id-isismtt-at-retrieveIfAllowed**	OBJECT IDENTIFIER ::= {id-isismtt-at 9}	obsolete	--	-	n.a.	obsolete	
13	**id-isismtt-at-requestedCertificate**	OBJECT IDENTIFIER ::= {id-isismtt-at 10}	obsolete	--	-	n.a.	obsolete	
14	**id-isismtt-at-namingAuthorities**	OBJECT IDENTIFIER ::= {id-isismtt-at 11}		+-	++	n.a.	P1.T29b	
15	**id-isismtt-at-certInDirSince**	OBJECT IDENTIFIER ::= {id-isismtt-at 12}	obsolete	--	-	n.a.	obsolete	
16	**id-isismtt-at-certHash**	OBJECT IDENTIFIER ::= {id-isismtt-at 13}		++	++	n.a.	P4.T15	
17	**id-isismtt-at-nameAtBirth**	OBJECT IDENTIFIER ::= {id-isismtt-at 14}		+-	++	n.a.	P1.T7	
17a	**id-isismtt-at-additionalInformation**	OBJECT IDENTIFIER ::= {id-isismtt-at 15}		+-	++	n.a.	P1.T29f	
18	**id-isismtt-at-liabilityLimitationFlag**	OBJECT IDENTIFIER ::= {0 2 262 1 10 12 0}		+-	++	n.a.	SigP.T2	
19	**id-isismtt-at-nameDistinguisher**	OBJECT IDENTIFIER ::= {0 2 262 1 10 7 20}	obsolete, backward compatibility!	--	++	n.a.	T1.#18	

3 LDAP

ISIS-MTT-compliant certification authorities MUST publish end entity and CA certificates. It is RECOMMENDED that certificates are downloadable from an LDAP server. No specific requirements apply for SigG-conforming systems and thus no profiling information is added here with respect to the Core Document Part 4.

4 OCSP

For SigG-conforming applications, the primary means of providing and obtaining revocation status information is declared by this profile to be OCSP. Accredited CSPs MUST provide an OCSP service, non-accredited CSP MAY choose to provide one.

For the sake of long term validation (Requirement (2) of Section 1.2), SigG-conforming directories MUST retain status information for a so called *retention period* of time after the end of the expiry year. The retention period is as long as 5 years for non-accredited CSPs and 30 years for accredited ones. Certificates MAY include the *RetentionPeriod* extension. Certificates MUST be kept in the directory for this period and OCSP responders MUST be able to deliver status information after the expiry of certificates. For the same reason, this profile RECOMMENDS against deleting revoked certificates from CRLs, which is common practice. The means for downloading certificates SHOULD be LDAP.

If requesting status information from a standard OCSP responder beyond the retention period, standard OCSP products may deliver the response '*good*' (meaning 'not known to be revoked' according to [RFC2560]) and may falsely lead to successful validation of a certificate. It is therefore crucial that the directory service of a CA is able to send a '*positive statement of availability***'** to the clients, indicating that the requested certificate is kept in the queried directory and the revocation information is thus reliable (i.e. help the client to be able to interpret '*good*' as 'certificate is known to the responder and has certainly not been revoked'). Each OCSP response given for SigG-conforming signature certificates MUST contain a positive statement in form of the *CertHash* extension.

Additionally, the retention period MAY be explicitly sent in the response, so that clients, querying the status of a certificate beyond the retention period, can detect that status information is no longer available. OCSP responders MAY send this information in a *ArchiveCutoff* extension of the response.

Relying components MUST be able to interpret the positive statement and the retention information and MUST involve them in the signature validation process.

Table 13: Special requirements on OCSP protocol elements

#	DATA FIELD	PROFILING INFORMATION (CONSTRAINT OR ENHANCEMENT WITH RESPECT TO ISIS-MTT)	CRITI-CAL	SUPPORT		REFE-RENCE	NO TES
				GEN	PROC		
	BASICOCSPRESPONSE FIELDS						
1	`signature`	[RFC2560]: All definitive response messages (*responseStatus=successful*) MUST be digitally signed. The key used to sign the response MUST belong to one of the following: (a) the CA who issued the certificate(s) in question (b) a Trusted Responder whose public key is trusted by the responder (and installed directly at the client), affected certificates include the *OCSPNocheck* extension (see Table 1.#5) (c) a CA Designated Responder (Authorized Responder) who holds a specially marked certificate issued directly by the CA, indicating in the *ExtendedKeyUsage* extension that the responder may issue OCSP responses for that CA. [RFC2560]: The above list is extended with the following option: (d) a key associated with the CA (i.e. a CA's 'OCSP Signing' key) **SIGG-PROFILE:** As described in (d) above, the responder's certificate MAY be issued for the CA by some other trusted authority. This set-up allows relying components to obtain reliable status information even if the key of the issuing CA has been compromised. SigG-conforming accredited CAs MUST obtain responder certificates from the German Federal Agency for Communication and Post (RegTP), which contains an 'OCSP signing' key. ATTENTION! Currently, the certificates issued by the RegTP for OCSP responders are marked with the *CRLSign*-bit in the *KeyUsage* extension, whereas the *ExtKeyUsage* extension is not included. Clients MUST temporarily accept this kind of flagging as authorization for OCSP signing.		++	++	P4.T8.#5	
2	`CertStatus 'good'`	[RFC2560]: ATTENTION! As status information delivered by OCSP may be obtained from CRLs, '*good*' does not necessarily mean that the certificate was ever issued or that the response time lies within the certificate's validity interval. Additional information regarding the status, such as positive statement about issuance, validity, may be included in response extensions. SigG-conforming CAs are obliged to provide positive statement about the issuance of a certificate. This ISIS-MTT Specification provides means for that by defining the private single response extension *CertHash*. See also #4.				P4.T8.#24	

	RFC 2560 EXTENSIONS						
3	`ArchiveCutoff`	extension in ResponseData: a responder MAY choose to retain revocation information beyond the certificate's expiry date. In this case, the responder SHOULD include the certificate's "cutoff" date, which is obtained by subtracting the retention period from the *producedAt* time. According to the SigG, compliant directory services are obliged to retain information for a period of 30 years in accredited directories and respectively for 7 years in non-accredited ones. The *ArchiveCutoff* extension with appropriate content SHOULD be present, independent of whether *CertHash* is present.	--	+	++ (RFC+-)	P4.T13	
	ISIS-MTT PRIVATE EXTENSIONS						
4	`CertHash (Positive Statement)`	*SingleResponse* extension: the responder may include this extension in a response to send the hash of the requested certificate to the requestor. This hash serves as evidence that the certificate is known to the responder (i.e. it has been issued) and will be used as means to provide a '*positive statement on issuance'*. According to the SigG (TODO §), compliant directory services MUST provide positive statement about the issuance of signature certificates. Hence, SigG-compliant responders MUST always include this extension in single responses.	--	CORE+- SigP++	++	P4.T15	

5 TSP

SigG-conforming certification authorities MAY offer time-stamping services. For the sake of interoperability, ISIS-MTT specifies a time stamp protocol (TSP) to acquire and obtain time stamp from a server. This protocol is fully compatible with the one defined in the PKIX standard [RFC3161]. No profiling information with respect to the ISIS-MTT Core Document Part 4 is added here for SigG-conforming applications.

6 Certificate Path Validation

Part 5 of the ISIS-MTT Specification describes a certificate path validation algorithm that fully complies with [RFC2459] and the validity implied by that PKIX profile. This model allows verifying long term signatures, even after the validity period respectively after the revocation of a signature certificate. This situation is illustrated in Figure 2. If a relying user wants to validate a signature at T_{val}, he/she/it must mathematically verify the signature over the document using the public key in the certificate of the signer and check whether this certificate and all certificates of its path were valid at the time T_{sig} of signing the document.

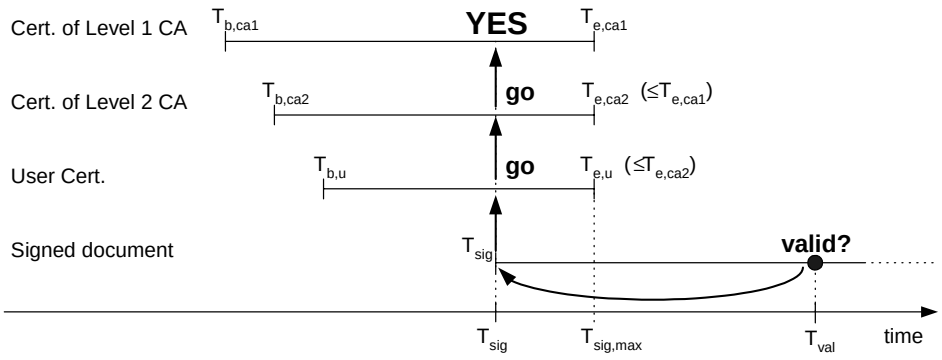

Figure 2: Successful validation of a signature according to the PKIX model

If a CA certificate in the path of the signing certificate has been revoked before the signing time T_{sig}, the signature is considered to be invalid in the PKIX model, as depicted in Figure 3. This also means that the latest time $T_{sig,max}$ a user can provide a valid signature is the of the revocation time $T_{rev,ca2}$ of the CA certificate in the path. After this time the user cannot generate valid signatures with its private key in conjunction with this user certificate, even if the certificate was not explicitly revoked.

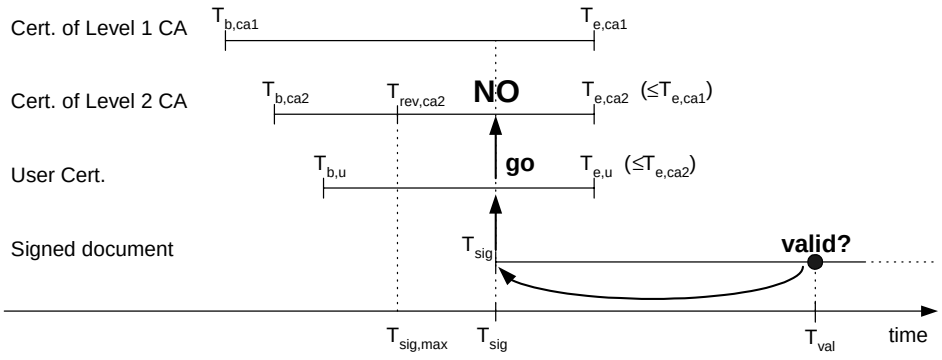

Figure 3: Signatures created after a revocation are invalid in the PKIX model

At this point, the SigG raises a different requirement in §19 (5), saying:

"§ 19: Supervision Measures …

(5) The validity of qualified certificates issued by a certification service provider shall not be affected by a ban on his operations and cessation of operations or by withdrawal and revocation of an accreditation." (unofficial translation, by courtesy of RegTP)

Well, there have been disputes for a long time, what the purpose of this clause could be and whether the legislator actually meant the validity of the signature (not the certificate) to remain unaffected after cessation of the CSP, in which case the PKIX model would exactly fit the legal requirements. Compared with the current formulation of §19 (5), the PKIX model is "too strict": in case of cessation of a CSP it delivers a negative *technical* judgement for a signature that is valid in the *juridical* sense. CSPs MAY take this into account and promote the PKIX model to be used in conjunction with their certificates. The reverse situation, i.e. interpreting a legally valid signature as technically invalid, can never occur.

Note furthermore, that if the CSP commits itself to a policy of revoking all user certificates before its own certificate gets revoked, the situation can never occur and the PKIX model always delivers a technical judgement of validity which is identical with the juridical one. It is being discussed whether such a revocation policy should be seen as an infringement of the law.

In the current vague situation, CSPs wanting to provide technical products that exactly fulfil the validity requirements of the SigG, MAY implement a slightly different variant of the PKIX model, called here the *SigG-model*. According to this model, validation follows exactly the "normal way" induced by the PKIX model and delivers the same results in the normal case. If, however, the relying component detects that the certificate of the CA that issued the user's signing certificate was revoked before the signing time T_{sig}, it shall not to cease with negative result, but try to validate the CA certificate with respect to the issuing time $T_{b,u}$ of the user's certificate. If it succeeds with this, the user's certificate shall be considered valid. This procedure is illustrated in Figure 4. If the time of issuance is different from the begin of the validity period (e.g. a certificate is issued with validity period in the future), the issuance time SHOULD be indicated in a *DateOfCertGen* extension of the user certificate.

Note that the "escape route" can only be taken, if the secret key of the CSP has not be compromised, but revoked for some other reason, which does not affect the reliability of the issued certificates. If the reason of revocation cannot be reliably determined, the component SHOULD consider the signature to be invalid.

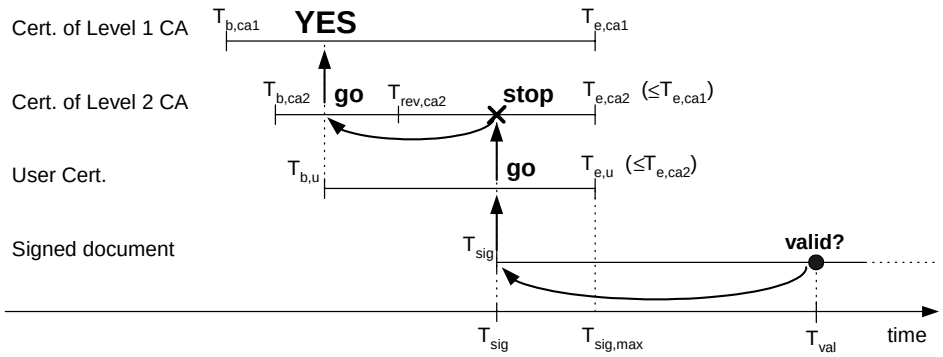

Figure 4: Signatures created after cessation of a CA are valid in the SigG model

In the following, we give a formal description of a path validation algorithm that implements the SigG-model. The algorithm is almost identical with the one specified in Section 2.2 of Part 5. Actually, one single step of the *ValidateCertPath*() function, namely Step #12 of P5.T4, needs to be altered to adopt the algorithm to the SigG-model. The description of this step is given in Table 14, using the same tabular form and notation as in Part 5.

Table 14: ValidateCertPath()

#	PSEUDO-CODE	COMMENTS	REF. TO PART 4	NOTES

1	`if( CheckRevocationStatus( tbvCert,`
	`tbvCerts,`
	`refTime,`
	`initialPolicySet,`
	`trustedCerts,`
	`trustedCrls )==false )`
	`{`
	`**if( (tbvCert.certType==CACert ||**`
	`**tbvCert.certType==CrossCert) &&**`
	`**(tbvCert.revoked==true) &&**`
	`**(tbvCert.revocationReason!='unspecified') &&**`
	`**(tbvCert.revocationReason!='keyCompromise') )**`
	`**{**`
	`Certificate &eeCert = tbvCertPath.GetItem(n);`
	
	`Time eeCertSigningTime;`
	`if( eeCert.ContainsDateOfCertGen() )`
	`eeCertSigningTime = eeCert.GetDateOfCertGen();`
	`else`
	`eeCertSigningTime = eeCert.GetValidityNotBefore();`
	
	`if( CheckRevocationStatus( tbvCert,`
	`tbvCerts,`
	`**eeCertSigningTime,**`
	`initialPolicySet,`
	`trustedCerts,`
	`trustedCrls )==false )`
	`return false;`
	`**}**`
	`else`
	`return false;`
	`}`
		Step #12 of P5.T4 MAY be replaced by the one here, if the certificate path *tbvCertPath* should be validated according to the SigG-model.
		If *CheckRevocationStatus*() returns *false*, this indicates that either the certificate was revoked before *refTime* or no status information could be obtained. Instead of ceasing path validation immediately, as the BPVA of RFC2459 does, this algorithm variant checks, whether:
		- the certificate is a CA certificate or a cross certificate and
		- it was revoked and
		- the revocation reason could be determined (not '*unspecified*') and
		- it was not *keyCompromise*.
		If these conditions are met, the algorithm takes the "escape route" by calling *CheckRevocationStatus*() again with the time instance parameter changed from *refTime* to the signing time of the EE certificate, which is the last element of *tbvCertPath*.
		If any of the above conditions is not met, the function returns *false*, as the original algorithm.
			P4.T5.#12	

7 Algorithms

This RIPEMD-160 hash algorithm is published in [RegTP 2003] as an algorithm appropriate and allowed for signing according to the German law on digital signatures [SigG01]. It is also used in existing certificates of the Regulatory authority for Telecommunications and Post (RegTP). Therefore it is urgently RECOMMENDED that components compliant with this profile accept data elements signed using RIPEMD-160 as a hash function.

References

[DraftOCSPv2] Online Certificate Status Protocol, version 2, draft-ietf-pkix-ocspv2-02.txt, March 2001

[ECDIR] Directive 1999/93/EC of the European Parliament and of the Council of 13 December 1999 on a Community Framework for Electronic Signatures

[ETSI-POL] ETSI TS 101 456 v1.1.1 (2000-12): Policy Requirements for Certification Authorities Issuing Qualified Certificates, Technical Specification

[ETSI-QC] ETSI TS 101 862 v1.3.1 (2004-03): Qualified Certificate Profile, Technical Specification

[ETSI-SIG] ETSI ES 201 733 v1.1.3 (2000-05): Electronic Signature Format, ETSI Standard

[ISIS] Industrial Signature Interoperability Specification ISIS, Version 1.2, December 1999, T7 i.Gr., www.t7-isis.de

[RegTP03] Regulatory authority for telecommunications and post: Declaration on appropriate algorithms for electronic signatures (Bekanntmachung zur elektronischen Signatur nach dem Signaturgesetz und der Signaturverordnung), published in German federal law gazette (Bundesanzeiger) from 11. March 2003

[RFC2459] Internet X.509 Public Key Infrastructure - Certificate and CRL Profile, January 1999

[RFC2560] X.509 Internet Public Key Infrastructure Online Certificate Status Protocol -OCSP, June 1999

[RFC3039] Internet X.509 Public Key Infrastructure Qualified Certificates Profile, January 2001

[RFC3161] Internet X.509 Public Key Infrastructure - Time Stamp Protocol (TSP), RFC 3161, August 2001

[RFC3280] Internet X.509 Public Key Infrastructure – Certificate and Certificate Revocation List (CRL) Profile, April 2002

[RFC3281] An Internet Attribute Certificate Profile for Authorization, April 2002

[SigG01] Law on the Conditions for Electronic Signatures (Gesetz über Rahmenbedingungen für elektornische Signaturen und zur Änderung weiterer Vorschriften), Bundesgesetzblatt Nr. 22, 2001, S.876.

[SigV01] Ordinance on Digital Signatures (Verordnung zur digitalen Signatur – SigV), 2001

COMMON ISIS-MTT SPECIFICATIONS
FOR INTEROPERABLE PKI APPLICATIONS

FROM T7 & TELETRUST

SPECIFICATION

OPTIONAL PROFILE

OPTIONAL ENHANCEMENTS TO THE SIGG-PROFILE

VERSION 1.1 – 16 MARCH 2004

Contact Information

ISIS-MTT Working Group of the TeleTrusT Deutschland e.V.: www.teletrust.de

The up-to-date version of ISIS-MTT can be downloaded from the above web site, from www.isis-mtt.org or from www.isis-mtt.de

Please send comments and questions to isismtt@teletrust.de

Editors:

Jürgen Brauckmann

Alfred Giessler

Tamás Horváth

Hans-Joachim Knobloch

Document History

VERSION DATE	CHANGES
1.0 30.09.2001	First public edition
1.0.1 15.11.2001	A couple of editorial and stylistic changes: - references to SigG-specific issues eliminated from core documents - core documents (Part 1-7) and optional profiles have been separated in different PDF documents.
1.0.2 19.07.2002	Several editorial changes. The definition of *RequestedCertificate* extended in order to accept attribute certificates. (T6.#2)
1.0.2 11.08.2003	Incorporated all changes from Corrigenda version 1.2
1.1 16.03.2004	Several editorial changes.

Table of Contents

1 Preface

The ISIS-MTT Specification describes data structures and communication protocols for technical components of widely interoperable, secure, PKI-based Internet applications (e.g. email, file transfer or web applications). It is a major goal to provide for compatibility with international PKI-standards of the IETF and thus to allow client software and CA services to work in an international context. The optional SigG-Profile to ISIS-MTT is intended for use in relation with qualified signatures and services within the context of the German Signature Act (SigG).

This document is intended as an optional enhancement to that profile and describes data elements that MAY optionally be included in the protocols employed by SigG-conforming components. We stress that this document is either a part of the ISIS-MTT Core Specification nor is it essential for the SigG-Profile. Therefore, compliance with ISIS-MTT or with the SigG-Profile does NOT require supporting any of the features described in this document. This Optional Profile is only informational, presenting implementation details of legacy systems and applications.

2 Special Certificate Extensions

At the moment, only one additional extension is defined here for binding a certificate to a public key file on a smartcard.

Table 1: An Overview of Special Certificate Extensions

#	EXTENSION	OID	SEMANTICS	CRITICAL	SUPPORT		REFERENCES		NOTES
					GEN	PROC	RFCS	ISISMTT	
	SPECIAL PRIVATE EXTENSIONS								
1	PKReference	{1 3 36 8 3 7}	Reference for a file of a smartcard that stores the public key of this certificate and that is used as "security anchor".	--	+-	+-		T2	

Table 2: PKReference

#	ASN.1 DEFINITION	SEMANTICS	SUPPORT		REFERENCES		NOTES
			GEN	PROC	RFCS	ISISMTT	
1	**id-isismtt-at-pKReference** OBJECT IDENTIFIER ::= {id-isimtt-at 7}	OID for extension *PKReference*			n.a.		
2	**PKReferenceSyntax** ::= OCTET STRING (SIZE(20))	Reference for a file of a smartcard that stores the public key of this certificate and that is used as "security anchor".	+-	+-	n.a.		[1]

[1]	This extension may be useful in smartcard applications. Because of the limited memory capacity of a smartcard, it may be necessary to store only the public keys of certificates that are used as “security anchor” for the application. The *PKReference* includes an acronym of the issuer and the serial number of the certificate and refers thus unambiguously to the corresponding certificate. Public keys may be stored on the smartcard as tuples (*PKReference* + public key) allowing applications to easily associate them with corresponding certificates. Clearly, this extension is only useful for the client component used by the card-holder him/herself. This extension MAY be used in public key certificates and MUST be flagged non-critical. Either ISIS-MTT-compliant nor SigG-compliant clients are required to support this extension.

3 Special OCSP Extensions

This section introduces special OCSP extensions to provide for certificate distribution over OCSP and respectively for enhanced security during the delivery of signature devices to the users.

Table 3: An Overview of Special OCSP Extensions

#	DATA FIELD	PROFILING INFORMATION (CONSTRAINT OR ENHANCEMENT WITH RESPECT TO THE ISIS-MTT CORE)	CRITICAL	SUPPORT		REFERENCE	NOTES
				GEN	PROC		
6	CertInDirSince	*SingleOCSPResponse* extension: Date, when certificate has been published in the directory and status information has become available. Currently, accrediting authorities enforce that SigG-conforming OCSP servers include this extension in the responses.	--	-	+-	T4	
7	RetrieveIfAllowed	(Single)*Request* extension: Clients may include this extension in a (single) *Request* to request the responder to send the certificate in the response message along with the status information. Besides the LDAP service, this extension provides another mechanism for the distribution of certificates, which MAY optionally by provided by certificate repositories.	--	+-	+-	T5	

8	`RequestedCertificate`	*SingleOCSPResponse* extension: The certificate requested by the client by inserting the *RetrieveIfAllowed* extension in the request, will be returned in this extension. The SigG allows publishing certificates only then, when the certificate owner gives his explicit permission. Accordingly, there may be '*non-downloadable*' certificates, about which the responder must provide status information, but MUST NOT include in the response. Clients may get therefore the following three kind of answers on a single request including the *RetrieveIfAllowed* extension: (a) the responder supports the extension and is allowed to publish the certificate: *RequestedCertificate* returned including the requested certificate (b) the responder supports the extension but is NOT allowed to publish the certificate: *RequestedCertificate* returned including an empty OCTET STRING (c) the responder does not support the extension: *RequestedCertificate* is not included in the response Clients requesting *RetrieveIfAllowed* MUST be able to handle these cases.	--	+-	+-	T6	

Table 4: CertInDirSince

#	ASN.1 DEFINITION	SEMANTICS	SUPPORT		REFERENCES		NOTES
			GEN	PROC	RFC2560	ISISMTT	
1	**id-isismtt-at-certInDirSince** OBJECT IDENTIFIER ::= {1 3 36 8 3 12}						
2	**CertInDirSince** ::= GeneralizedTime		-	+-			[1]

[1] This extension contains the date, when certificate has been published in the directory and status information has become available. Including this extension prevents impersonation attacks at the beginning of the validity period.
Consider the following scenario: a CA generates keys for Alice and issues a certificate for her. The CA publishes the certificate immediately in a directory and sends a message with the keys to Alice. Mallory intercepts the message and uses the key to sign some document which he sends to Bob. Bob queries the OCSP server and obtains a 'positive statement on issuance'. Therefore he accepts the signature and thinks it would be from Alice. Even worse, Alice may not even notice that the key, she then receives, has got intercepted.
Such an attack can be prevented by employing the extension *CertInDirSince* as follows: a certificate will first be published in the directory, when the certificate owner has acknowledged that he has received the key. The start of the validity is bound on this point in time, i.e. on the time given in *CertInDirSince*. Signatures created before that time are not accepted. Additionally, there should be some mechanism that allows Alice to detect an interception of her key and revoke the certificate at or before the time indicated in *CertInDirSince*. (Revocation to a time instance that lies in the past is not allowed in SigG-conforming systems!)
There is no need for such measures and for including *CertInDirSince* in the response, if:

a) the key is generated locally by the user,
b) there is a secure way of transporting the key, or
c) the publishing of the certificate is bound to an acknowledgement of reception of the key. In addition to that, there is a mechanism to detect an interception of the key, in which case the certificate will not be published at all and can those never lead to the verification of a faked signature.

Compliant OCSP responders SHOULD NOT use this extension, but SHOULD employ organizational measures listed above. Compliant clients MAY but need not (or: involve *CertInDirSince* in the verification, but may assume that CAs employ some of those measures.

A note on SigG-conformance: Accrediting authorities no longer enforce that SigG-conforming OCSP servers include this extension in the responses.

Table 5: RetrieveIfAllowed

#	ASN.1 DEFINITION	SEMANTICS	SUPPORT		REFERENCES		NOTES
			GEN	PROC	RFC2560	ISISMTT	
1	**id-isismtt-at-retrieveIfAllowed** OBJECT IDENTIFIER ::= {1 3 36 8 3 9}						
2	**RetrieveIfAllowed** ::= BOOLEAN		+-	+-			[1]
[1]	Clients may include this extension in a (single) *Request* to request the responder to send the certificate in the response message along with the status information. Besides the mandatory LDAP service, this extension provides another mechanism for the distribution of certificates, which MAY optionally by provided by certificate repositories.						

Table 6: RequestedCertificate

#	ASN.1 DEFINITION	SEMANTICS	SUPPORT		REFERENCES		NOTES
			GEN	PROC	RFC2560	ISISMTT	
1	**id-isismtt-at-requestedCertificate** OBJECT IDENTIFIER ::= {1 3 36 8 3 10}						
2	**RequestedCertificate** ::= CHOICE { Certificate Certificate, publicKeyCertificate [0] EXPLICIT OCTET STRING, attributeCertificate [1] EXPLICIT OCTET STRING }		+-	+-			[1]
[1]	**ISIS-MTT-Optional:** The certificate requested by the client by inserting the *RetrieveIfAllowed* extension in the request, will be returned in this extension. **ISIS-MTT-SigG:** The signature act allows publishing certificates only then, when the certificate owner gives his explicit permission. Accordingly, there may be '*non-downloadable*' certificates, about which the responder must provide status information, but MUST NOT include them in the response. Clients may get therefore the following three kind of answers on a single request including the RetrieveIfAllowed extension: a) the responder supports the extension and is allowed to publish the certificate: *RequestedCertificate* returned including the requested certificate b) the responder supports the extension but is NOT allowed to publish the certificate: *RequestedCertificate* returned including an empty OCTET STRING c) the responder does not support the extension: *RequestedCertificate* is not included in the response Clients requesting *RetrieveIfAllowed* MUST be able to handle these cases. If any of the *OCTET STRING* options is used, it MUST contain the DER encoding of the requested certificate.						

References

[RFC2459] Internet X.509 Public Key Infrastructure - Certificate and CRL Profiles, January 1999

[RFC2560] X.509 Internet Public Key Infrastructure Online Certificate Status Protocol - OCSP, June 1999