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