COMMON ISIS-MTT SPECIFICATIONS
FOR INTEROPERABLE PKI APPLICATIONS

FROM T7 & TELETRUST

SPECIFICATION

PROFILE FOR AUTHENTICATION CERTIFICATES

VERSION 1.0.1 – 14 FEBRUARY 2005

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:

Hajo Bickenbach	2B Advice GmbH, Bonn
Tamás Horváth	D-Trust GmbH, Berlin

Contributions by

Benjamin Brumaire	Sun Microsystems GmbH
Detlef Hillen	SRC Security Research & Consulting GmbH
Gerold Hübner	Microsoft Deutschland GmbH
Heinz Strauß	Sun Microsystems GmbH
Christian Stüble	Ruhr-Universität Bochum
Klaus-Peter Schmidt	T-Systems International GmbH
Dieter Schwara	DATEV eG
Jung-Uh Yang	Microsoft Deutschland GmbH

Document History

VERSION DATE	CHANGES
0.1 1-Sept-04	First draft version.
0.2 9-Sept-04	Second draft version including comments out of TeleTrusT AG8. First public draft version.
0.9 02-Nov-04	Pre-final draft version including various comments.
1.0 02-Nov-04	Final version
1.0.1 14-Feb-05	Minor error corrections, especially typos, and wording in footnote on page 13

Table of Contents

1 Preface

ISIS-MTT V.1.1 already comprises some guidance on authentication certificates. But with the growing interest in this field the ISIS-MTT board became more and more aware of the fact that a number of problems and questions around authentication and autorisation have to be clarified to promote interoperability. For this reason the authentication project was launched in June 2004 and lead after many discussions to the present paper.

The intention was to provide an authentication certificate profile that can be used in many system environments and in many widely used applications. The requirements of those target systems and applications are described in the paper too, providing thus the reasons for choosing specific settings. That is why this paper has a larger text portion than the rest of ISIS-MTT.

The paper is now published as a stand-alone document. As such it can still be regarded as work in progress. Further discussion in the ISIS-MTT board and in the working groups will probably lead to a way of publishing the document content as part of the ISIS-MTT standard document but it has not yet been decided in which form this will happen.

Since authentication is closely related to the major system platforms and application software it was very important that we had much support from their vendors. We would like to thank them and all other contributors possibly not named in the paper who have been enormously valuable for the success of the project. Further comments to the paper are cordially invited.

Hajo Bickenbach, Editor

Tamás Horváth, Editor

Arno Fiedler, ISIS-MTT Project Manager

2 Overview

Authentication encompasses two major aspects: **identification** and **authenticity**. In short, authentication is about the authentic (i.e. provable and traceable) identification of an entity for the purpose of system, service or data access or for the sake of message origin verification. Entities can be people but also machines (workstations, routers, servers, …).

In general authentication is based on credentials a requestor entity has to provide in order to prove that it is in fact the entity that should be granted the requested access or to prove that it is the originator of the message. Among the various types (username/password, one-time-passwords, PIN-TAN, digital certificates, etc.) only PKI-based authentication will be treated in the context of ISIS-MTT. This is due to the fact that ISIS-MTT is primarily concerned with PKI services and PKI-based application where digital certificates play a central role. An authentication profile is thus an enrichment of the existing definitions within ISIS-MTT without widening the scope.

Identification in the context of ISIS-MTT is about credentials in the form of X.509 certificates specifically tailored for the purpose of authentication. These certificates shall be used to ensure that the receiving side of an authentication request can assess the identity claimed by the requestor by means of verifying a certificate chain up to a trust anchor. The trustworthiness of identity assertion will rely on the underlying certification policy employed by the certification service providers. This can be used to define the appropriate assurance level for an envisaged application.

Authenticity of a message can be assessed by the receiving party by verifying the signature over the message. Using public key procedures it is also possible to prove the authenticity of a request for accessing some system, service or data. This usually implies a proof of possession of the private key by means of some challenge-response procedure.

Authentication is usually performed for the following purposes:

1) **Session Authentication:** Recognition of the identity (possibly mutually) leads to the establishment of a secure communication channel usually encompassing encryption for both data security and persistence of the authenticated state of the communication channel. Any information conveyed through this channel can be regarded as authenticated by the receiving party and used accordingly.
2) **Access Control (System Logon) and Authorization**: Authentication can be the precondition for granting access to the requestor to access a system, service or some data. The most basic form of this kind of authentication is system logon. In most cases, however, the requestor – although authenticated as an accepted communication partner – is not granted unlimited access. In a second step after authentication an authorization system may grant only some selected access rights to the authenticated entity based on some pre-defined access rules fitting for the requestor. Authorization as such is not in the scope of this document. But information used in the authorization step can be included in form of attributes in the certificates of the requestor. This corresponds to some extent to the handling of certificate attributes in ISIS-MTT and will be discussed accordingly in chapter **Fehler! Verweisquelle konnte nicht gefunden werden.**.

   Note that access control is often coupled to authenticated communication in a session. For example SSL/TLS supports this kind of authenticated/authorized sessions
3) **Data Origin Authentication (Message Authentication):** A single well defined message, document, code or other data object can also be authenticated by means of public key procedures in order to prove the originator of the data object as well as its

integrity. This is technically identical to an electronic signature in the legal sense but must be regarded as a different type of statement with a different meaning. Besides the most widely used email authentication there are a number of use cases for this authentication type especially in the financial sector but more can be expected in other areas as well. Data origin authentication is especially useful in cases where the establishment of an authenticated channel seems inappropriate. It may well make sense to authenticate documents for workflow purposes when passing them to some other entity without establishing an authenticated communication channel for this purpose. Another aspect is the fact that in this case the authentication data is retained within the document while in case of an authenticated channel it is dropped as soon as the communication channel gets closed.

All three cases are considered as important by the user community of ISIS-MTT and are therefore included into the scope of this document.

The basic aim of this document is to provide an authentication certificate profile that can be used in the above scenarios. The various aspects are discussed in chapters 4 and 5 and the actual profile is defined in chapter 6.

Since an ever greater number of applications, system platforms and web based services are PKI-enabled thus bringing strong certificate based authentication along it was necessary to restrict the number of sources that could be taken into consideration for the present specification. As a result of the preliminary discussions during project setup the working assumption was that an authentication profile will at least have to fit into the TLS-type and VPN-type scenarios plus any constraints imposed by major system platform manufacturers.

3 Focus

Since authentication is broadly used for many different use cases and for various assurance and security levels it is useful to precisely define the focus of the present definition.

3.1 In Scope

The scope of the present document is limited to certificate based

- Session Authentication, i.e. authentication of a communication channel
- Authentication for System Logon
- Single Message Authentication
- Authorization information as part of certificate content

The implications of the definitions set in the present document will be discussed on protocol level (as opposed to API level, see next chapter).

Protocols and functional areas to be discussed emphasize

- SSL/TLS / Web Authentication
- IPSEC
- System Logon functionality
- VPN protocols

This selection is not only due to the fact that these protocols cover much of the basic authentication use cases but is also a reflection of contributions to the discussion that led to the present paper. If anyone of the readers should feel the need to add a specific protocol please send a note to the editors.

3.2 Out of Scope

The following aspects have been left out of scope:

- Authorization as a function that assigns access rights based on authentication (i.e. identification) information

  Authorization is very much in the domain of the system the access is to be granted for. This means that authorization is very much driven by features and also flaws of the actual system and is usually a proprietary feature of said systems itself. Although there are common developments like role-based access models, directory models and identity management systems to handle huge numbers of users it does not seem necessary to define standards for this area for the time being.

- Delegation and Federation

  Delegation and federation of identities are very interesting concepts and have been developed because single-domain authorization has proven to be too restricted for the world of web services and related activities. Delegation means that a user entitles a web service (that he or she will usually have to be authenticated against beforehand) to request additional services from some other entities in his or her name. Federation on the other hand means that one can use his or her identity acquired in the context of one service for authentication and authorization against another service.

  Both are useful concepts and may well lead to substantial changes in the authentication arena. For the time being they are left out of consideration because no need for profiling them has been expressed by the users of ISIS-MTT.

- API-Level

  As opposed to communication protocols (e.g. SSL/TLS) there are several APIs that are included in the various system implementations and that, among other functions, provide authentication functionality. Since interoperability does usually not take place on API level but during system interaction on protocol level APIs are generally left out of consideration.

- Tokens

  For most cases there is no need to distinguish between digital certificates in the two forms of Software-PSEs and Hardware-PSEs, i.e. smartcards or other tokens. It is assumed that storage and protection of the private keys are transparent for the respective system and do not in themselves create interoperability problems. But note that if a software token is to be used on arbitrary machines special precautions have to be taken that cannot be discussed here.

4 Use Cases

In order to facilitate the definition of an authentication certificate profile the most relevant use cases are discussed in this section. This is not meant to be an exhaustive description in any way but is intended to identify typical scenarios to relate them to authentication mechanisms and to inform about the requirements on the certificate contents originating from the underlying protocols.

To enable a certificate to be used for authentication purposes most protocols and systems presume the presence of some key usage flag, some name forms of the certificate owner or some other specific contents in the certificate. These sensitive details will be handled with special care in the discussion below and will be summarized in Section 6.

In ISIS-MTT Part 1, Table 21 an open list of extended key usages is referenced which is identical to the list named in [RFC3280]. It comprises the basic entries for the ExtendedKeyUsage certificate extension that are also important for authentication purposes.

In this chapter important requirements from various sources are discussed. As a result the profile set up by this paper can be found in table 10.

4.1 Web Authentication

Two of these extended key usages are a direct reference to the Transport Layer Security Protocol (TLS). It is widely employed within SSL protected web sites and it is the de-facto standard for certificate based web authentication.

4.1.1 Server Authentication

Most SSL protected web sites use TLS for server authentication and session key generation in order to be able to enforce privacy protection for the transport of sensitive data from the client to the server like credit card numbers or the like. The relevant extended key usage is named *serverAuth*.[1] The key usage bits *digitalSignature* and *keyEncipherment* must be set.[2]

Server authentication is a very important functionality for all web-based services like internet banking. It allows the user to make sure that the web server is actually the machine identity he or she wants to do business with.

4.1.2 Client Authentication

Much less used in the open internet is the client authentication feature of TLS. But wherever client authentication is needed as in intranet or other more closed user groups it is again the de-facto standard for web client authentication. The advantages are obvious especially when web technology is employed on the basis of a heterogeneous technology platform and no homogeneous system logon or access control is available. The relevant extended key usage is named *clientAuth*.[3] The key usage bits *digitalSignature* and *keyEncipherment* must be set.

Support for both certificate based TLS server and client authentication is provided by most web server products and can be activated by simple administration means. It is therefore not

[1] 1.3.6.1.5.5.7.3.1; Note that although there has been some discussion on this topic, these OIDs are NOT defined in [TLS] or in some related document, but in the general certificate profile [RFC3280] at PKIX.

[2] according to [TLS], section 7.4.2

[3] 1.3.6.1.5.5.7.3.2 [RFC3280]

an implementation issue. Special care needs to be taken for the scalability of a solution when big or huge numbers of users need to be managed which is usually not supported by the web servers themselves.

4.2 Mutual Authentication

A number of protocols and scenarios although often used in a client-server environment do not follow the client-server rationale but work on the concept of mutual authentication. IPSEC is a prototype of this case where every IPSEC-enabled network node must be able to authenticate itself against any IPSEC-enabled communication partner. Authentication can be performed using various mechanisms but only public key based mechanisms are of interest here.

There are a number of requirements defined in [IKE_PKIX] that regard the usage of public key procedures for authentication in an IPSEC environment. The relevant requirement for end entity certificates concerns presence of the extended key usage flag *iKEIntermediate*[4]. Note that [IKE_PKIX] has never become an RFC and for this reason is no longer available at the IETF. Nevertheless if any certificate key usage information is evaluated in an IPSEC environment the flag *iKEIntermediate* is likely to be the one.

VPN authentication shares to some extent the characteristics of IPSEC authentication in that it usually comprises mutual authentication. Some of the relevant protocols like L2TP rely directly on IPSEC for authentication purposes. Generally speaking the protocols used to instantiate VPNs are on a more abstract level and do not define any specific requirements on certificates.

The EAP protocol family is another type of authentication protocol that is also used to perform authentication inside VPN or RAS initialization. Among other authentication types it can be used for mutual public key based authentication. Especially its newer children EAP-TLS and EAP-TTLS are much discussed as a solution to secure WLAN. But again no specific requirements on certificate content are defined in these protocols other than the ones defined in TLS.

Finally the same is true for RADIUS as the most widely used Authentication, Authorization and Accounting (AAA) Server.

As a result it can be stated that there are no requirements on authentication certificates defined in the relevant protocols employed for mutual authentication. The only extended key usage flag that may be enforced in IPSEC environments is the *iKEIntermediate* value.

But it should be noted that although the protocols themselves do not impose any distinct certificate requirements such requirements may still be enforced by the system or application the user requests access for.

4.3 User Authentication and/or Machine Authentication

It should be generally noted that client authentication may be ambiguous in that it may have the two meanings of “user authentication” as well as “machine authentication”. Especially when both types of entities can have their own certificates a number of additional options can be considered:

- User Authentication only

  This is typically used in a web scenario where the client machine does not become a

[4] iKEIntermediate: 1.3.6.1.5.5.8.2.2 [RFC2409]

part of the server domain but is simply the machine hosting a standard client software like an internet browser. In a scenario where access and user activity is completely controlled by the entry point (e.g. the web server enforcing user authentication) user authentication is likely to be a strong enough mechanism.

- Machine Authentication only

  Machine authentication alone is more or less related to unattended machines like routers, gateways or web or application servers that nevertheless need to authenticate themselves in order to be able to perform their duties. Web server authentication is a typical case for machine authentication.

- User plus Machine Authentication

  Whenever users must be authenticated not only for a limited set of applications and access rights but the connection to be established becomes more like a system logon and the users access heterogeneously stored business data and the machines needing access to network resources like printers, faxes etc. a combination of the two types of authentication should be employed.

Decision for one or the other of these options depends on the overall security design.

For a certificate profile it is important that all types can be supported. Apart from (extended) key usage flagging this presupposes that the naming conventions allow for machine names like DNS names, IP addresses, URLs or other types of machine identifiers. In a practical setting these may be needed in combination with user identity information in a combined certificate or separately in a machine certificate.

The naming conventions within ISIS-MTT are noted in ISIS-MTT Part1, Table 8: The different forms of "General Name" allow for all necessary name types so that no new naming definitions need to be made for authentication purposes. In order to provide a certificate profile that can be used in as many environments as possible recommendations on usage of name attributes are given in chapter 6.

4.4 Data Origin Authentication (Message Authentication)

There are use cases where establishing an authenticated and secured channel is not appropriate while there is still a need for authentication regarding the origin of an object like in the case of code signing or email authentication. Technically this is identical to an electronic signature but the purpose is the assertion of the origin rather than a legal statement like the commitment of the originator to the content.

To date the purposes for electronic signatures are indicated in the corresponding X.509 certificate as permitted (extended) key usages. This is an important feature designed to protect the certificate holder.

Following this trail the well understood way to permit data origin authentication is setting the key-usage and/or extended-key-usage extensions of the authentication certificate.

We suggest that data origin authentication shall be regarded as a function of any electronic signature applied to a data object and not as a separate type of electronic signature nor as a separate type of authentication.

The requirement for an authentication certificate profile is then to support data origin authentication purposes by means of electronic signatures as well as the other authentication types identified previously in this chapter.

In the case of a signature expressing a legal statement in terms of content commitment using a certificate with the non-repudiation or the new *content-commitment*[5] flag set this object authentication function is only a subset of the full meaning of the signature but nevertheless it is present.

Besides definitions for key-usage and extended-key-usage there is a second way of expressing signature purposes: Both [CAdES] for CMS type and [XAdES] for XML type signatures define commitment types for signature data structures. This approach allows for much greater flexibility since the information can be placed in the signature and need not be defined beforehand at certificate creation time. Using this mechanism the distinction between data origin authentication (i.e. proving the sender or origin of the data object) and electronic signature as a legal statement becomes an explicit part of the document and need not be deduced from implicit certificate information. This explicit statement MUST conform with the key usage information in the certificate in order to protect the certificate holder.

This second approach is RECOMMENDED in all cases where the key-usage in the certificate is not restricted to exactly one purpose.

4.5 Other Protocols

There are a number of protocols and applications that are widely used and that do support strong authentication based on public key certificates but that do not define any specific requirements for the certificates to be used.

Such protocols are not named in particular in this profile because they do not add to the requirements of an authentication profile.

4.6 Trust Center Service Authentication

Certificate Service Providers (CSPs) may want to employ authentication in conjunction with various services like directory access, certificate status information requests, certification service, time stamp service etc. The reasons for enforcing authentication may be:

- limitation of the services to a narrower customer group,
- setting prices for the services, to link to an accounting system,
- setting quotas for the usage of the services,
- saving computational power by avoiding unwanted use,
- provision against denial-of-service attacks.

Although some of the application protocols (like OCSP) provide means for certificate based authentication within the respective protocol itself this paper (later part of ISIS-MTT) RECOMMENDS providing TLS authentication at the transport layer. This handling provides the advantage over application level authentication that the authentication function can be decoupled from the actual service. Decoupling has the following benefits:

- it is possible to employ the same technology in conjunction with the various services thus providing a uniform interface to backend systems relying on authentication information (access control, accounting, quote monitoring, statistics),
- it might further be possible to use the very same system (or computer) for

[5] Note that it has been indicated out of the ITUT working group that the new version of X.509 will name the Bit 1 of the key usage “content commitment” and deprecate the ‘old’ naming “non-repudiation”. Although not official at this point in time the new name is used in this document as an alternative for non-repudiation.

authentication, offering thus one single interface point to the backend systems; a benefit in terms of administration, hardware and licenses,

- it is possible to employ TLS products of major vendors or to rely on open source or on other standard components in the development of authentication software modules. This might be highly beneficial with regard to software quality, reliability and costs,
- a uniform handling across different PKIs and CSPs would allow delegation and federation of user access rights, while retaining interoperability.

This recommendation lies in the broadening trend of employing TLS/SSL for web service security and access control, we are witnessing nowadays.

4.7 System Logon

Most of the use cases discussed so far are based on open standards that have been successfully implemented by a large number of different vendors. Due to standardization of these protocols interoperability can be achieved even when using products of different manufacturers.

This is not the case for system logon. So for the sake of a single certificate profile that can be used for system logon on various system platforms the requirements specific for these platforms must be discussed to some extent.

Platforms are included in this paper on the basis of contributions by the vendors or in the case of the open source platforms by members of that community. Additional contributions for the future for more platforms would be most welcome and are cordially invited.

4.7.1 Microsoft

- XP clients can be configured for certificate based system logon and particularly for smartcard logon. In both cases the User Principal Name (UPN) must be present in the *SubjectAltNames* extension as well as a Microsoft specific OID[6] as EKU in the case of smartcard logon[7]. The same applies for Terminal Services Logon. A pre-requisite is that any domain controller participating in the smartcard logon environment must be supplied with domain controller certificates.
- Internet Information Server (IIS) supports TLS server authentication and can be configured to enforce TLS client authentication. Note that it expects the UPN and maps it to a particular user account either in the domain active directory or locally. The server certificate must contain the DNS name either in the commonName attribute of the *subject* field or in the *SubjectAltNames* extension.
- Both VPN and wireless connections can be authenticated using EAP-TLS (among other protocols) and by default perform machine and user authentication consecutively. Again the UPN must be present.
- If IPSEC machine authentication should be supported, either the *iKEIntermediate* EKU or the TLS Client/Server Authentication EKU plus the DNS name must be present in the certificate. Smartcards are not supported in the current products for this purpose.

[6] szOID_NT_PRINCIPAL_NAME, (1.3.6.1.4.1.311.20.2.3); the user principal name (UPN) consists of the account name, the at sign (@), and a user principal name suffix. The user principal name suffix is the DNS domain name of the forest root domain. Starting with Windows 2000 the UPN is the name-form employed.

[7] szOID_KP_SMARTCARD_LOGON (1.3.6.1.4.1.311.20.2.2)

- If secure email shall be supported the *emailProtection* EKU must be present.[8]
- If code signing shall be supported the *codeSigning* EKU must be set.[9]

4.7.2 SUN Microsystems

- Certificate Authentication Attributes

  Almost any certificate type can be used for authentication. Certificate attributes used for user identification and account mapping can be selected by the system administrator out of a number of standard certificate naming attributes. No specific certificate attributes are mandated.[10]
- Smartcard Framework

  On the basis of the smartcard framework it is possible to use smartcards for all types of authentication scenarios.
- Single Sign On

  Based on the initial authentication an SSO authentication token (based on the SAML specification) is created and stores the authentication level. Based on this token subsequent logins can be performed automatically if the authentication level is sufficient for the desired application. Re-authentication is only necessary if the initial authentication level is not sufficient. The same mechanism can be used for inter-domain authentication.

4.7.3 Open Source

Many open-source UNIX derivates including Linux and FreeBSD support Pluggable Authentication Methods (PAM) (see, e.g., [Linux-PAM]), a flexible authentication interface that allows to write applications that are authentication mechanism independent. Moreover, PAM allows different applications to use different authentication methods: the webserver, for instance, can authenticate users against a SQL database, while users trying to log on the system through SSH are authenticated against a Kerberos server. Different Linux PAM modules exist that implement a X.509 certificate based user authentication [PKCS11-PAM] [X.509 PAM] using the OpenSSL library [OpenSSL]. Based on these PAM modules, several solutions implement user authentication with smart cards [SC-Login][SC-Netlogin] and Single Sign On [PAM-SSO].

[8] In the authentication context this is relevant for the case of message authentication as a function of digital signatures that can be applied to emails.

[9] See last footnote

[10] Note that the UniqueID field in the subjetct portion of the certificates can be used in a SUN sysetm environment while it is forbidden in ISIS-MTT Part 1, Table 2 #10 and therefore also in this profile.

4.8 Authorization Based on Certificate Content

The X.509 standard defines authorization as the "conveyance of privilege from one entity that holds such privilege, to another entity". A typical application is an access control system that permits or denies access to some web service. In this case, the user *asserts* his privilege to access the service (typically implicitly by logging on to the service) and the system *verifies* his privilege by some means (typically looking up an access list or checking a directory). If the user proves to have the privilege, the system *authorises* the user to access the required service, i.e. conveys the privilege (inherently owned by the access control system) to the user to access the service.

As readily declared in the introduction, this ISIS-MTT specification does not intend to describe means for privilege management methods as such. There is however a point of contact of that area to PKI: a PKI, more closely attribute certificates, provide an interesting method and means for issuing, carrying, asserting and verifying privileges and thus to provide the underlying technology with a transport mechanism for privilege management. The advantages of PKI-based privilege management compared to the traditional, centrally-managed methods may be the following:

- the same infrastructure can be used for privilege management as for security services: maintaining (issuing, revoking, storing) privileges can be done within the same infrastructure, namely the CAs, directories etc. of the PKI,
- the same technology can be used for verifying privileges (e.g. access to a web service) as for security services (e.g. providing secrecy and integrity during the session): privilege verification reduces to certificate verification,
- the same technology can be used to distribute, store, carry, assert privilege information to/at/from the clients as given by a PKI: privileges can be stored and transported as certificates,
- privileges can be distributed to a wide user group and can be used for accessing several services (targets). Privileges can be verified “off-line”, i.e. without needing to access some central privilege management system. Thus, issuing privileges and verification (authorization) can be organizationally separated.

Public key certificates can provide an identity to access control decision functions. However, in many contexts the identity is not the criterion that is used for access control decisions, rather the role or group-membership of the accessor is the criterion used. Such access control schemes are called role-based access control. [X.509-2000] and RFC3281 describe attribute structures for carrying information for authorization (i.e. privilege management) purposes. The proposed attributes specify group membership, role assignment, security clearance or other authorization information associated with the attribute holder. Authorization information may be placed in a public key certificate extension or placed in a separate attribute certificate (AC). The placement of authorization information in public key certificates is usually undesirable for two reasons:

- First, authorization information often does not have the same lifetime as the binding of the identity and the public key. When authorization information is placed in a PKC extension, the general result is the shortening of the PKC useful lifetime.
- Second, the public key certificate issuer is not usually authoritative for the authorization information. This results in additional steps for the public key certificate issuer to obtain authorization information from the authoritative source.

For these reasons RFC3281 and ISIS-MTT promote separating authorization information from the PKC. Yet, authorization information also needs to be bound to an identity. An attribute certificate (AC) provides this binding; it is simply a digitally signed (or certified) identity and set of attributes. When making an authorization decision based on an AC, the decision function may need to ensure that the appropriate AC holder is the entity that has requested the service. The most straightforward way to achieve a linkage between the request or identity and the AC is the inclusion of a reference to a PKC within the AC and the use of the private key corresponding to the PKC for authentication within the access request.

4.8.1 Using Authorization Certificates

The X.509 standard defines authorization as the "conveyance of privilege from one entity that holds such privilege, to another entity". An AC is one authorization mechanism. An ordered sequence of ACs could be used to verify the authenticity of a privilege asserter's privilege. In this way, chains or paths of ACs could be employed to delegate authorization. Currently only a single level AC scheme is promoted by [RFC3281] and by ISIS-MTT: users are expected to assert one single AC and privilege verifiers are expected to be able to process just one single AC.

[RFC3281] describes two possible models how authorization ACs can be employed: the “push” and the “pull” models.

The “push” model:

In some environments, it is suitable for a client to "push" the AC to the requested server. This means that:

- the ACs are distributed to the user of the service
- no new connections between the client and server are required
- no search burden is imposed on the server, which improves performance and that the AC verifier is only presented with what it "needs to know."

The "push" model is especially suitable in inter-domain cases where the client's privileges should be assigned within the client's "home" domain, but should be asserted in external server domains.

The “pull” model:

In other cases, it is more suitable for a client to simply authenticate to the server and for the server to request or "pull" the client's AC from an AC issuer or a repository (directory). A major benefit of the "pull" model is that:

- it can be implemented without changes to the client or to the client-server protocol
- it provides a smooth way for migration in privilege management system readily using some directory to store privilege information.

The "pull" model is especially suitable for inter-domain cases where the client's privileges should be assigned within the server's domain, rather than within the client's domain.

There are a number of possible exchanges involving three entities: the client, the server, and the AC issuer. In addition, a directory service or other repository for AC retrieval MAY be supported. [RFC3281] does not specify a protocol for exchanging information among these parties. ISIS-MTT strongly recommends relying on standard service protocols (e.g. LDAP).

5 Additional Aspects

5.1 Policy Aspects of Authentication Certificates

In some environments it is desirable to take additional certificate quality information into account for authentication decisions. This could be quality information on the identification, on the certificate production system or e.g. on smartcard support. Such features and rules of the service comprise the certificate policy and contribute to the assurance level[11] a relying party can assume if using the authentication certificate. As an example, the aim might be to grant access to sensitive personal data only after the user could be identified properly using an authentication certificate (correct key usage) of sufficient quality (high enough assurance level). Note that key usage information of any kind yields only the purpose of certificate usage and is not directly related to quality information of any kind.

The existing technical mechanism to indicate the assurance level is to produce a certificate policy document, give it an OID and include the OID and a web link to the policy document in the CertificatePolicies extension. This solution is feasible and much used in many single domain solutions. But an ever growing number of solutions have a scope beyond a single security domain where individual policies raise an interoperability barrier. Also this mechanism makes the assessment of the assurance levels an out-of-band mechanism: The relying party must read the policy and decide whether or not to accept the given certificate.

Since the aim of this paper is not to invent new mechanisms but to profile existing ones we have abstained from defining a mechanism here. But we would like to name a number of requirements for an interoperable solution:

- The assurance level for authentication certificates should be included in the certificate and should be easily identifiable. It should convey enough substantial information for an authentication decision to be taken on that basis.
- A very limited set of assurance levels should be defined. A solution with some self-evidence would be to define those levels according to the ones for signature certificates (simple, advanced, qualified, possibly also qualified with accreditation). Thus no new legal definitions would be necessary but the established “signature” policy rules (apart from key usage restrictions) could be adapted to authentication certificates.
- A mechanism should be defined to assert these classes of assurance levels in the certificates. Possible ways are generic policy OIDs or specially designed new extensions. Both have serious drawbacks: A policy (and its OID) usually covers much more than one single aspect. The more additional aspects become part of the policy the more likely the OID will not be acceptable for a given environment. On the other hand new private extensions are a threat to interoperability and thus intended to be avoided in ISIS-MTT.
- A maximum approach might look for machine-readable policies or a similar mechanism to convey the wanted type of information. Although much research is under way in this field none of those solution seems to get widely accepted for the time being.

[11] We do not use this term in the formal sense of an “Evaluation Assurance Level” as in the Common Criteria but as a more general term. In particular it does not refer to any formal evaluation of the relevant components.

In order to address the present needs of a number of large-scale projects (government and others) it might be a short-term solution that at least for the authentication certificates accompanying qualified certificates on smartcards a policy OID is negotiated among the certificate service providers.

Any solution will be happily integrated into ISIS-MTT when it has been agreed upon in the relevant working groups.

5.2 Algorithms

Since authentication certificates will often aim at a certain existing authentication scenario, it does not make much sense to forbid any algorithms that may be imposed by or just beneficial in that environment.

For this reason the rule for algorithm usage in the context of ISIS-MTT is the following:

- For coherence with the rest of ISIS-MTT one cipher suite consisting of RSA, 3DES and SHA1 is mandated. In terms of the TLS specification this means that it MUST support TLS_RSA_WITH_3DES_EDE_CBC_SHA[12].
- A compliant solution SHOULD support the algorithms that are listed in ISIS-MTT Part 6.
- A compliant solution MAY use other algorithms for authentication purposes.[13]

5.3 Differences to other Certificate Types

5.3.1 Content Commitment (Non-Repudiation) Certificates

Specialized authentication certificates are not intended to produce electronic signatures as commitment to the signed content. But there are reasons to minimize the number of certificates in certain environments e.g. because of limited capacity on a special device. Multi-usage certificates with the *non-repudiation* (or *content-commitment*[14]*)* plus the *digitalSignature* key-usage bits set (ISIS-MTT P1.T12.[1]) are not forbidden. But there are a number of security issues related to them as e.g. illustrated by the following scenario:

Imagine a server operator that manipulates the server in a way that it does not offer random challenge values but hash values (of some document) during a challenge-handshake in an authentication protocol. After the server receives the encrypted value back as proof of possession of the private key, the operator would be able to add this encrypted hash value to the document the hash value was calculated upon beforehand. All of a sudden there exists a message with a hash value and an encryption thereof that in the extreme case could not be distinguished from an electronic signature as commitment to the signed content.

No matter if this attack would be possible with any one of the protocols discussed in this paper extreme care should be taken with usage of certificates for authentication purposes that

[12] This cipher suite is identical to SSL_RSA_WITH_3DES_EDE_CBC_SHA of the SSL 3.0 specification [SSL3.0].

[13] Note that although hashing is not necessarily part of an authentication handshake protocol there is no “official” algorithm definition e.g. for RSA with PKCS#1 padding without definition of a hash algorithm. At the time of completion of this document there is an ongoing discussion whether or not such a definition should be included into ISIS-MTT Part 6 among the other algorithms named there.

[14] see above footnote no 5

have the non-repudiation (or content-commitment) bit set. There are certainly ways to avoid this trap but they will usually end up with a non-standard protocol which is probably not a good choice.

5.3.2 Other Combined User Certificates

Combination of TLS-client-authentication and email-protection already covers a lot of the actual user needs. This is particularly necessary for a number of mail-clients. There is no security reason not to combine authentication and email protection features other than a general reluctance to secure too many functions with a single certificate as one single security anchor.

6 Certificate Profile

The following table gives a selection of extensions and attributes that are relevant for authentication purposes. Note that all definitions from Part 1 remain valid for this profile.

Since the systematic ordering is given in the respective parts of ISIS-MTT (mostly in Part 1) we think that we can do without a full list here and rather provide a convenient synopsis of the relevant items that can be used more easily.

Table 1: Overview of the Authentication Certificate Profile

#	CERTIFICATE FIELD	ISISMTT REQUIREMENTS (CLIENT/SERVER)		REMARK	REQ'S. TLS (CLIENT/SERVER)	REQ'S MICROSOFT	REQ'S SUN [1]	REQ'S OPEN SOURCE [2]	ISISMTT REFERENCE	NOTES
		GEN	PROC							
	SUBJECT DNAME									
1)	commonName	++	++	mandated in Part 1		IIS/TLS server auth.: DNS-name MUST be present either in *CN* or in *SubjectAltNames*			P1.T7.1	[3]
2)	other subject DName attributes	+-	+-				PAM: supported		P1.T7	
3)	emailAddress	-	+	According to P1, this attribute is tolerated, but NOT RECOMMENDED.				Netscape-Mail: SHOULD be present	P1.T7.17	
	SUBJECTALTNAMES									
4)	rfc822Name (email address)	+-	+-	OPTIONAL in end user certs., but SHOULD be present when email application targeted		Outlook: SHOULD be present for autom. handling of addresses	PAM: supported		P1.T8.4	[4]
5)	User Principal Name (UPN)	+-	+-	OPTIONAL, but MUST be present for mentioned target applications		XP-Logon, IIS-TLS client auth., EAP-TLS auth.: MUST be present	PAM: supported		n.a.	

6)	DNS-Name or IP Address	+-	+-	OPTIONAL, but MUST be present for mentioned target applications		IIS/TLS server auth.: DNS-name MUST be present either in *CN* or in *SubjectAltNames*	PAM: not supported		P1.T8.5 P1.T8.10	
7)	NetBIOS Name	--	--	proprietary attribute from Microsoft, forbidden for compliance with Part 1 and for general security considerations		supported		not supported	n.a.	
8)	Mapping to a user account (informational)			The mapping is out of the scope of this certificate profile and is mentioned here only for information.		Logon, IIS/TLS client-auth: Mapping based on UPN to ActiveDirectory or to local access list	PAM: configured in the admin. environment	PAM: configured in the admin. environment, mapping to SQL-DB, Kerberos, LDAP etc.	n.a.	
	OTHER ID FIELDS									
9)	issuer + certificate serial Nr.	++	+-	Used in the PKI context for unambiguous identification of a certificate.		not supported	PAM: not supported		P1.T28.4 P8.T8.2	[5]
10)	SubjectUniqueID	--	--	forbidden in Part 1			PAM: supported		P1.T2.10	
	KEYUSAGE									
11)	digitalSignature	++	++	mandated in TLS	++/++				P1.T12.2	
12)	keyEncipherment	++	++	mandated in TLS	++/++				P1.T12.3	
	EXTENDEDKEY-USAGE (EKU)									
13)	serverAuth	-/++	+	This ext. key usage MUST be indicated in certificates of server-type end entities.	n.a./++	EKU can also be used for IPSEC/IKE			P1.T21.4	
14)	clientAuth	++/-	+	This ext. key usage MUST be indicated in certificates of client -type end entities.	++/n.a.	EKU can also be used for IPSEC/IKE			P1.T21.5	
15)	codeSigning	+-	+-	defined for data origin authentication of program code					P1.T21.6	
16)	emailProtection	+-/-	+	defined for message authentication purposes.		Outlook: SHOULD be			P1.T21.7	

				OPTIONAL, but MUST be present for mentioned target applications		present in user cert.				
17)	iKEIntermediate	+	+	recommended for EE certs. to be employed for IPSEC/IKE		IPSEC-auth.: either *iKEInterm.* or *clientAuth* + *serverAuth* + DNS-name MUST be present			n.a.	
18)	szOID_KP_SMART CARD_LOGON	+-	+-	proprietary OID defined for Logon in XP environment, OPTIONAL, but MUST be present for mentioned target applications		Logon: MUST be present when the private key is stored in a smartcard.		PAM: not supported	n.a.	
19)	anyExtendedKey-Usage	-	-	SHOULD NOT be used in order to restrict usage of the certificate to authentication purposes.				PAM: not supported	P1.T21.2a	

[1]	Sun Microsystems employs a highly flexible handling of identification data within it's pluggable authentication module (PAM): the administrator can specify the identification profile to be employed by the individual system by selecting an arbitrary subset of the *supported* name attributes (marked in the column). Only those selected attributes will then be used for the identification of the user.
[2]	Most Linux variants provide pluggable authentication modules (PAM), that can be used for user logon or client authentication. Please refer to the references in the text of Section 4.7.3 for detailed information on requirements on certificate contents. *Editors' note: Readers should feel free to deliver information to empty table cells.*
[3]	Note that according to ISIS-MTT Part1 the *commonName* MUST be present, but MAY represent a pseudonym, which is marked with a trailing ":PN". Pseudonyms might cause problems during mapping to an identity database, if they are not unique.
[4]	The email address SHOULD be placed in the *SubjectAltNames* extension (P1.T2.[6]). Note that when the email address is used as the identification criterion, changing an email address in the identity database (which may happen more frequently than certificate re-issuing or at least independently thereof) will enforce certificate re-issuance or authentication based on this attribute will fail.
[5]	This attribute tuple represents a common way to refer to a single certificate at least in the PKI context. Thus it would be the perfect match for authentication as well if one wishes to enforce the usage of a specific certificate. As a drawback, it would require an administrational process that assigns the particular certificate to the user account AFTER certificate creation. Using a name attribute included in the certificate makes this process obsolete and relies with regard to name correctness and uniqueness on the certificate issuing process.

7 Attribute Certificate Profile

The basis for the following attribute certificate attribute specifications is [RFC3281]. Unfortunately, there is a slight, but fatal difference between the definitions of the attribute certificate format in ISIS-MTT Part 1 (version v1, based on [X.509-1997]) and respectively of that in RFC3281 (version v2, based on [X.509-2000]). The problem is described in ISIS-MTT P1.T28.[1] and [3] and has to do with the incompatible types used in the fields *subject* (type CHOICE) and respectively *holder* (type SEQUENCE) of the respective versions of *AttributeCertificateInfo*. According to this incompatibility, the v2 format is neither backward nor forward compatible with v1. This fact is considered by the authors of this document to be an engineering failure, which is not likely to be corrected in forthcoming specification versions. To further comply with RFCs and to avoid backwards compatibility problems, this paper (later ISIS-MTT) RECOMMENDS providers and software vendors to act as follows to overcome this unlucky situation:

1. For the generation (GEN) of ACs with the private ISIS-MTT attributes described in Part 1, CSPs MUST employ v1 for the time being. The possibility and time frame of a migration to v2 needs to be discussed.
2. For the processing (PROC) of ACs with the private ISIS-MTT attributes described in Part 1, systems MUST be prepared to work with v1. The requirement on supporting v2 too needs to be revisited, if a migration to v2 is planned on the generation (GEN) side.
3. For the generation (GEN) of ACs with the RFC3281 attributes, as also described in this paper, CSPs MUST employ v2.
4. For the processing (PROC) of ACs with the RFC3281 attributes, as also described in this paper, systems MUST be prepared to work with v2.

In general, systems processing ACs SHOULD be prepared to be able to work with both versions. This is also important for components, like directories, client applications, libraries etc. which have to be able to parse certificates during ordinary operation, even if no attribute information needs to be extracted."

7.1 Attribute Certificate Attributes

Table 2: An overview of attribute certificate attributes

#	ATTRIBUTES	OID	SEMANTICS	MULTI-VALUED	SUPPORT		REFERENCES		NOTES
					GEN	PROC	RFC	ISISMTT	
	RFC3281 ATTRIBUTES						**RFC3281**		
1	SvceAuthInfo	{id-aca 1}	This Service Authentication Information identifies the AC holder by a name to a server or service.	Y	+-	+-	4.4.1	AuthP.T2	
2	AccessIdentity	{id-aca 2}	Identifies the AC holder to a server or service.	Y	+-	+-	4.4.2	AuthP.T3	
3	ChargingIdentity	{id-aca 3}	Identifies the AC holder for charging purposes.	N	+-	+-	4.4.3	AuthP.T4	[1]
4	Group	{id-aca 4}	Group membership of the AC holder	N	+-	+-	4.4.4	AuthP.T5	[1]
5	Role	{id-at 72}	Role allocation of the AC holder	Y	+-	+-	4.4.5	AuthP.T6	
6	Clearance	{2 5 1 5 55}	Clearance information about the AC holder	Y	+-	+-	4.4.6	AuthP.T7	
[1]	RFC3281: Though these attributes are single-valued at the level of the *values* field (SET OF AttributeValue), they may contain multiple values in the *values* field of the underlying *IetfAttrSyntax* structure.								

Table 3: Service Authentication Information

#	ASN.1 DEFINITION	SEMANTICS	SUPPORT		REFERENCES		NOTES
			GEN	PROC	RFC3281	ISISMTT	
1	**SvceAuthInfo** ::= SEQUENCE {						[1]
2	service GeneralName,	Name of the targeted server/service.					[2]
3	ident GeneralName,	Identifier of the AC holder at the targeted server/service.					[3]
4	authInfo OCTET STRING OPTIONAL }	Additional service specific authentication information.	+-				[4]
[1]	RFC3281: The Service Authentication Information attribute identifies the AC holder to the server/service by a name, and the attribute MAY include optional service specific authentication information. This attribute provides information that can be presented by the AC verifier to be interpreted and authenticated by a separate application within the target system. Note that this is a different use to that intended for the Access Identity attribute. This attribute type will typically be encrypted when the *authInfo* field contains sensitive information, such as a password. The encryption of attributes is described in Section 7.1 of RFC3281.						
[2]	**ISIS-MTT Profile**: Only the name forms directoryName, IPAddress or URI are allowed.						
[3]	**ISIS-MTT Profile:** Only the name forms directoryName, IPAddress, URI or Email address are allowed.						
[4]	RFC3281: Typically this will contain a username/password pair for a "legacy" application.						

Table 4: Access Identity

#	ASN.1 DEFINITION	SEMANTICS	SUPPORT		REFERENCES		NOTES
			GEN	PROC	RFC3281	ISISMTT	
1	`SvceAuthInfo ::= SEQUENCE {`						[1]
2	`  service  GeneralName,`						
3	`  ident  GeneralName,`						
4	`  authInfo  OCTET STRING OPTIONAL` `}`		--				[1]
[1]	RFC3281: The same syntax is used for the Access Identity attribute as for Service Authentication Information, except that the *authInfo* field MUST NOT be filled in Access Identity. This attribute is intended to be used to provide information about the AC holder, that can be used by the AC verifier (or a larger system of which the AC verifier is a component) to authorize the actions of the AC holder within the AC verifier's system. Note that this is a different use to that intended for the Service Authentication Information attribute.						

Table 5: Charging Identity

#	ASN.1 DEFINITION	SEMANTICS	SUPPORT		REFERENCES		NOTES
			GEN	PROC	RFC3281	ISISMTT	
1	`IetfAttrSyntax ::= SEQUENCE {`				4.4		[1]
2	`  policyAuthority IMPLICIT [0] GeneralNames OPTIONAL,`	Issuer of the applying attribute policy.					[2]
3	`  values  SEQUENCE OF CHOICE {`	Attribute values.					[3]
4	`    octets  OCTET STRING,`	May contains some more complex attribute value in some encoded form.					
5	`    oid  OBJECT IDENTIFIER,`	Attibute value expressed by an OID.					

6	`    string  UTF8String`
	`  }`
	`}`
		String attribute value.					

[1]	RFC3281: The Charging Identity attribute identifies the AC holder for charging purposes. In general, the charging identity will be different from other identities of the holder. For example, the holder's company may be charged for service.						
[2]	RFC3281: This attribute field allows a separation between the AC issuer and the attribute policy authority. This is useful for situations where a single policy authority (e.g. an organization) allocates attribute values, but where multiple AC issuers are deployed for performance or other reasons.						
[3]	RFC3281: The syntaxes allowed for values are restricted to OCTET STRING, OBJECT IDENTIFIER, and UTF8String, which significantly reduces the complexity associated with matching more general syntaxes. All multi-valued attributes using this syntax are restricted so that each value MUST use the same choice of value syntax. For example, AC issuers must not use one value with an OID and a second value with a string.						

Table 6: Group (Group Membership)

#	ASN.1 DEFINITION	SEMANTICS	SUPPORT		REFERENCES		NOTES
			GEN	PROC	RFC3281	ISISMTT	
1	`IetfAttrSyntax ::= SEQUENCE {`				4.4		
2	`  policyAuthority IMPLICIT [0] GeneralNames OPTIONAL,`	Issuer of the applying attribute policy.					T4.[2]
3	`  values SEQUENCE OF CHOICE {`	Attribute values.					T4.[3]
4	`    octets OCTET STRING,`						
5	`    oid OBJECT IDENTIFIER,`						

6	`    string UTF8String`
	`  }`
	`}`						

[1]	The Group attribute carries information about group memberships of the AC holder.						

Table 7: Role (Role Assigment)

#	ASN.1 DEFINITION	SEMANTICS	SUPPORT		REFERENCES		NOTES
			GEN	PROC	RFC3281	ISISMTT	
1	`RoleSyntax ::= SEQUENCE {`						[1]
2	`  roleAuthority IMPLICIT [0] GeneralNames OPTIONAL,`	Issuer of the role definition.	+-				[2]
3	`  roleName IMPLICIT [1] GeneralName` `}`	Role identifier					[3]
[1]	RFC3281: The role attribute carries information about role allocations of the AC holder.						
[2]	RFC3281: The *roleAuthority* field MAY be used to specify the issuing authority for the role specification certificate. There is no requirement that a role specification certificate necessarily exists for the *roleAuthority* (Editor: i.e. it can be simply the name of the manufacturer or maintainer of the authorization management system). **ISIS-MTT Profile:** Only the name forms directoryName, IPAddress and URI are allowed.						
[3]	RFC3281: The roleName field MUST be present, and roleName MUST use the uniformResourceIdentifier CHOICE of the GeneralName. **ISIS-MTT Profile:** Name forms directoryName, IPAddress and URI are allowed.						

Table 8: Clearance (Security Label)

#	ASN.1 DEFINITION	SEMANTICS	SUPPORT		REFERENCES		NOTES
			GEN	PROC	RFC3281	ISISMTT	
1	**Clearance** ::= SEQUENCE {						[1]
2	policyId IMPLICIT [0] OBJECT IDENTIFIER,	The policyId field is used to identify the security policy to which the clearance relates. The policyId indicates the semantics of the *classList* and *securityCategories* fields.					
3	classList IMPLICIT [1] ClassList DEFAULT {unclassified},	List (i.e. BIT STRING) of the classification values assigned to the AC holder.					[2]
4	securityCategories IMPLICIT [2] SET OF SecurityCategory OPTIONAL }						

5	**ClassList** ::= BIT STRING {
	unmarked (0),
	unclassified (1),
	restricted (2)
	confidential (3),
	secret (4),
	topSecret (5)
	}						

6	**SecurityCategory** ::= SEQUENCE {						[3]
	type IMPLICIT [0] IMPLICIT OBJECT IDENTIFIER,	OID of the syntax used in the value field below.					
	value IMPLICIT [1] ANY DEFINED BY type }						

[1]	RFC3281: The clearance attribute, specified in [X.501-1993], carries clearance (associated with security labeling) information about the AC holder.
[2]	RFC3281: Additional security classification values, and their position in the classification hierarchy, may be defined by a security policy as a local matter or by bilateral agreement. The basic security classification hierarchy is, in ascending order: unmarked, unclassified, restricted, confidential, secret, and top-secret. An organization can develop its own security policy that defines security classification values and their meanings. However, the BIT STRING positions 0 through 5 are reserved for the basic security classification hierarchy.
[3]	RFC3281: If present, the SecurityCategory field provides further authorization information. The security policy identified by the *policyId* field indicates the syntaxes that are allowed to be present in the *securityCategories* SET. An OBJECT IDENTIFIER identifies each of the allowed syntaxes. When one of these syntaxes is present in the *securityCategories* SET, the OBJECT IDENTIFIER associated with that syntax is carried in the SecurityCategory.type field.

7.2 Authorization Certificate Extensions

Table 9: An overview of attribute certificate extensions

#	EXTENSION	OID	SEMANTICS	CRITICAL	SUPPORT		REFERENCES		NOTES
					GEN	PROC	RFC3281	ISISMTT	
	RFC3281 AC PRIVATE EXTENSIONS								
6	AuditIdentity	{id-pe 4}	This extension instructs the server or service not to log the real ID (subject name) of the AC holder, but to use the ID given in the *AuditIdentity* extension for log and audit purposes.	++	- (liebe r --?)	-	4.3.1	n.a.	[1]
7	Targets	{2 5 29 55}	Name of server(s) or service(s), the AC is intended for.	++	+-	+-	4.3.2	AuthP.T9	
8	NoRevAvail	{2 5 29 56}	Indicates that no revocation information will be available for the AC.	--	- (liebe r --?)	-	4.3.6	n.a.	[2]
[1]	**ISIS-MTT PROFILE:** At current time, ISIS-MTT does not recommend using this critical extension. If required for privacy protection, the subject name of the AC or that of the PKC referenced in the AC's subject name should contain a pseudonym, which just fulfils the intended purpose.								
[2]	**ISIS-MTT PROFILE:** ISIS-MTT requires certificate issuers to include information about CRL or OCSP access, if status information is maintained in CRLs or in an OCSP service. On the other hand, if status information is not maintained for a certificate, no corresponding *CrlDistributionPoints* or *AuthorityInfoAccess* extensions can be found in the certificate. The absence of these extensions indicates unambiguously that no status information is available. Hence, ISIS-MTT does not promote employing the *NoRevAvail* private extension.								

Table 10: Targets

#	ASN.1 DEFINITION	SEMANTICS	SUPPORT		REFERENCES		NOTES
			GEN	PROC	RFC3281	ISISMTT	
1	**Targets** ::= SEQUENCE of Target	A list of target servers, for which the AC describes privileges of the AC holder.			4.3.2		[1]
2	**Target** ::= CHOICE {	Target item					[2]
3	targetName IMPLICIT [0] GeneralName,	Name (directoryName), IP address or URL of the target server	+-	++			
4	targetGroup IMPLICIT [1] GeneralName,	Target group of servers, typically defined by a name (directoryName) or an domain name.	+-	++			[3]
5	targetCert IMPLICIT [2] TargetCert }	Certificate of the target server	--	+-			[4]
6	**TargetCert** ::= SEQUENCE {						
7	targetCertificate IssuerSerial,						
8	targetName GeneralName OPTIONAL,						
9	certDigestInfo ObjectDigestInfo OPTIONAL }						

[1]	RFC3281: To target an AC, the target information extension MAY be used to specify a number of servers/services. The intent is that the AC SHOULD only be usable at the specified servers/services. An (honest) AC verifier who is not amongst the named servers/services MUST reject the AC. If this extension is not present, the AC is not targeted and may be accepted by any server.
[2]	RFC3281: The targets check passes if the current server (recipient) is one of the *targetName* fields in the *Targets* SEQUENCE, or if the current server is a member of one of the *targetGroup* fields in the Targets SEQUENCE. In this case, the current server is said to "match" the targeting extension.
[3]	RFC3281: How the membership of a target within a targetGroup is determined is not defined here. It is assumed that any given target "knows" the names of the targetGroups to which it belongs or can otherwise determine its membership. For example, the targetGroup specifies a DNS domain, and the AC verifier knows the DNS domain to which it belongs.
[4]	RFC3281: The *targetCert* CHOICE within the *Target* structure is only present to allow future compatibility with [X.509-2000] and MUST NOT be used.

List of Tables

Abbreviations

3DES	triple DES symmetric encryption algorithm, see ISIS-MTT, Part 6
AAA	authentication, authorization and accounting server
AC	attribute certificate
API	application programming interface
CMS	cryptographic message syntax
CSP	certification service provider
DNS	domain name service
EAP	extensible authentication protocol
EAP-TLS	EAP variant using TLS
EAP-TTLS	EAP variant using “tunneled” TLS
EKU	extended key usage, in Microsoft documentation sometimes referenced as “enhanced” key usage
GEN	column concerning the generation entity in an ISIS-MTT table
IETF	internet engineering task force
IIS	Internet Information Server, Microsoft’s web server
IP	internet protocol
IPSEC	IP security, security protocol on IP level
ISIS-MTT	interoperability specification (www.isis-mtt.org); this paper is in the context of ISIS-MTT
L2TP	layer 2 tunneling protocol, VPN protocol
OCSP	online certificate status protocol
OID	unique object identifier
PAM	pluggable authentication module
PIN-TAN	authentication mechanism using a PIN (personal information number) as account information and a TAN (transaction number) per transaction, that becomes invalidated after the transaction is completed
PKC	public key certificate
PKI	public key infrastructure, general term for usage of public key technology
PKIX	working group at the IETF for PKI related definitions
PROC	column concerning the processing entity in an ISIS-MTT table
RAS	remote access service
RFC	“request for comment”, internet standard at the IETF
SHA1	hash algorithm, see ISIS-MTT, Part 6
SQL	database query language
SSH	secure shell remote login protocol
SSL	secure sockets layer

TLS	transport layer security
TS	technical specification, used for ETSI (European Telecommunications Standards Institute) standards
UPN	User principal name, see annotation no 6
VPN	virtual private network
X.509	in the X-series of the ITU (International Telecommunications Union) no 509 concerning digital certificates
XAdES	XML Advanced Electronic Signatures
XML	eXtensible Markup Language

References

[RFC2246] The TLS Protocol Version 1.0, January 1999

[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

[IKE_PKIX] Internet Draft by Rodney Thayer/ Charles Kunzinger, IBM/ Paul Hoffman, VPNC: draft-ietf-ipsec-pki-req-05.txt (July 10, 2000)

[CAdES] ETSI TS 101 733 v1.4.0: Electronic Signature Formats, September 2002

[XAdES] ETSI TS 101 903 V1.1.1 (2002-02): XML Advanced Electronic Signatures (XAdES), Technical Specification

[X.509-1997] ITU-T Recommendation X.509 : Information Technology – Open Systems Interconnection – The Directory: Authentication Framework, 1997

[X.509-2000] ITU-T Recommendation X.509: Information Technology – Open Systems Interconnection – The Directory: Public-Key and Attribute Certificate Frameworks, 2000

[Linux-PAM] http://www.kernel.org/pub/linux/libs/pam

[PKCS11-PAM] http://n.ethz.ch/student/mariost/pkcs11_login/

[X.509 PAM] http://pam-x509.sourceforge.net/

[OpenSSL] http://www.openssl.org

[SC-Login] http://www.strongsec.com

[SC-Netlogin] http://www.twi.ch/~strasmar/smartcard_netlogin

[PAM-SSO] http://www.oo-services.com/en/articles/sso.html

[SSL3.0] http://wp.netscape.com/eng/ssl3/draft302.txt