TCG

TCG Infrastructure Workgroup Subject Key Attestation Evidence Extension

Specification Version 1.0
Revision 7
16 June 2005
Published

Contact: techquestions@trustedcomputinggroup.org

TCG Published

Disclaimer

THIS SPECIFICATION IS PROVIDED "AS IS" WITH NO WARRANTIES WHATSOEVER, INCLUDING ANY WARRANTY OF MERCHANTABILITY, NONINFRINGEMENT, FITNESS FOR ANY PARTICULAR PURPOSE, OR ANY WARRANTY OTHERWISE ARISING OUT OF ANY PROPOSAL, SPECIFICATION OR SAMPLE. Without limitation, TCG disclaims all liability, including liability for infringement of any proprietary rights, relating to use of information in this specification and to the implementation of this specification, and TCG disclaims all liability for cost of procurement of substitute goods or services, lost profits, loss of use, loss of data or any incidental, consequential, direct, indirect, or special damages, whether under contract, tort, warranty or otherwise, arising in any way out of use or reliance upon this specification or any information herein.

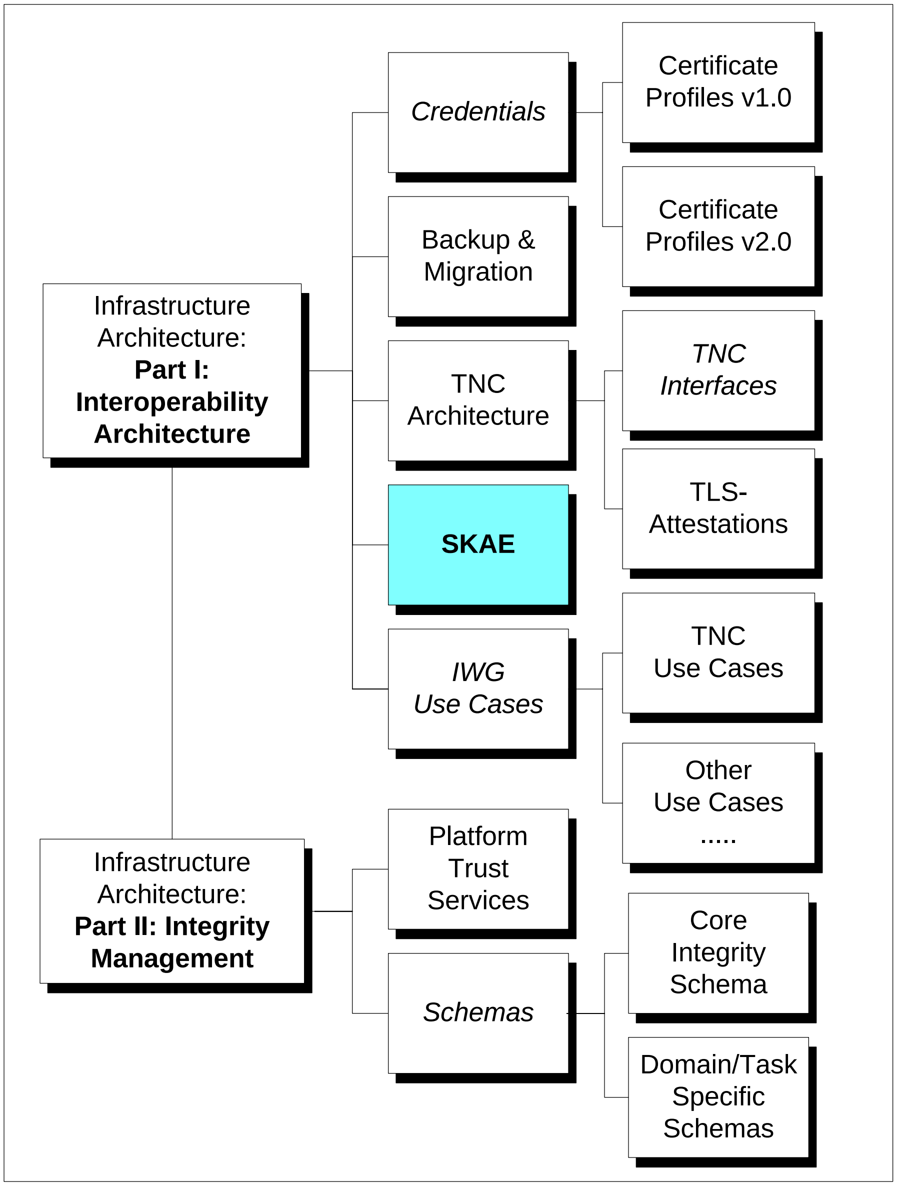
Contact the Trusted Computing Group at www.trustedcomputinggroup.org for information on specification licensing through membership agreements.

The only official, normative version of a TCG Specification or related document is the English-language text adopted by TCG under its Bylaws and published on the TCG website, www.trustedcomputing.org. TCG does not guarantee the accuracy or completeness of any other version. Other language versions of Specifications and related documents are provided for convenience and are intended to be technically identical to the official English version, but they may contain translation and other errors.

Translations may be provided by volunteers through the Trusted Computing Group's translation program (see www.trustedcomputinggroup.org/specifications.

Other legal notices and terms governing the publication of materials on the TCG website are found at www.trustedcomputinggroup.org/about/legal. TCG incorporates by reference the same notices and terms with respect to TCG-authorized translations of Specifications and related documents, whether published on the TCG website or at another online location.

IWG Document Roadmap


Infrastructure Architecture: Part I: Interoperability Architecture
Credentials
Backup & Migration
TNC Architecture
SKAE
IWG Use Cases
Certificate Profiles v1.0
Certificate Profiles v2.0
TNC Interfaces
TLS-Attestations
TNC Use Cases
Other Use Cases .....
Infrastructure Architecture: Part II: Integrity Management
Platform Trust Services
Schemas
Core Integrity Schema
Domain/Task Specific Schemas

Revision History

Rev 1.0_r1	Initial revision. Document started by Mihran Dars, mdars@wavesys.com	05/20/2004
Rev 1.0_r6	Multiple corrections from various reviewers.	10/21/2004
Rev 1.0_r7	Version 1.0_r7 submitted for 60-day internal TCG review.	11/24/2004
	TCG Board of Directors approval for Version 1.0_r7 publication.	6/16/2005

Acknowledgement

The TCG wishes to thank all those who contributed to this specification. This document builds on considerable work done in the various working groups in the TCG.

Special thanks to the members of the IWG group and others contributing to this document:

Geoffrey	Strongin	AMD
Malcolm	Duncan	CESG
Kazuaki	Nimura	Fujitsu Limited
John	Selep	Hewlett-Packard
Diana	Arroyo	IBM
Roger	Zimmermann	IBM
Johann	Schoetz	Infineon
David	Grawrock	Intel Corporation
Ned	Smith (IWG co-chair)	Intel Corporation
Daniel	Wong	Microsoft
Mark	Williams	Microsoft
Mark	Redman	Motorola
Nicholas	Szeto	Sony
Jeff	Nisewanger	Sun Microsystems
Paul	Sangster	Sun Microsystems
Thomas	Hardjono (IWG co-chair)	VeriSign, Inc.
Alex	Deacon	VeriSign, Inc.
Greg	Kazmierczak	Wave Systems Corp.
Jack	Cai	Wave Systems Corp.
Len	Veil	Wave Systems Corp.
Michael	Gile	Wave Systems Corp.
Mihran	Dars (Editor)	Wave Systems Corp.

Table of Contents

Table of Figures

1 Introduction

As the transition to trusted computing implementations occur, participating entities have the need to verify the security properties and associated integrity of keys which are used to authorize and authenticate participants in multi-party transactions. It is also desirable to create this assertion in a manner which seamlessly integrates into existing infrastructures.

With the advent of TCG technology, the capability exists to attest to the fact that a key used in a transaction was created within a well known trustworthy hardware environment, and that operations associated with that key have occurred in the corresponding trusted hardware environment. By creating this assertion in a standard fashion which can be readily integrated into X509 compliant PKIs, the implementation of this particular aspect of trusted computing can be performed with minimal impact to currently deployed environments, providing for this critical value-added assertion.

This document defines the Subject Key Attestation Evidence (SKAE) X.509 certificate extension which enables the cryptographic binding of TCG-oriented security assertions within a common certificate. Because this extension can be added by a CA to a standards-based certificate, this allows for secure communication of TCG properties over widely deployed certificate-based security protocols such as SSL, TLS or IKE (used with IPSec).

For example, SKAE includes an assertion that the asymmetric private key (corresponding to an SSL certificate’s public key) was generated and always stored in the protected storage of a TPM. This information is useful for the relying party to gain confidence in the authenticity of the holder of the key since its far less likely that the private key was stolen and be used fraudulently by the requestor.

This document specifies several possible scenarios of how the SKAE extension can be used to verify the protections afforded by TCG technologies to private keys to include the trustworthiness of transactions.

.

2 Subject Key Attestation Evidence

The SKAE extension specification defines a standard mechanism to represent a Certified Credential in X509 v.3 certificates. This mechanism allows a verifier to ensure that the use of the private key, represented by the corresponding public key certificate, was performed with a TCG compliant TPM environment. Most commonly, this TPM environment will imply the use of a hardware TPM device, ensuring the verifier of strict security properties of the environment in which the private key operations were performed.

The SKAE extension carries in the TPM_CERTIFY_INFO structure the certification evidence of the key referenced by the Attestation Identity Key (AIK) using TCG enabled platform. It could be used as an extension in X.509 v.3 certificates [X509, RFC3280], attribute certificates [RFC3281], certificate requests [RFC2314, RFC2511], various authentication and authorization protocols or elsewhere.

The trustworthiness of the TPM certified key can be attested to by an AIK Credential. The introduction of this new extension provides a standard [X509, RFC3126] mechanism by which the binding of TPM certified key with an AIK can be verified. This feature leverages the interoperability of TPM and legacy security systems since X.509 v3 public key certificates are extensible and can be used for authentication and/or authorization. Verification of this extension can be delegated to other TCG aware applications.

2.1 Terminology

In order to ensure a consistent interpretation of this specification, the following terms are used in this document:

Subject Key: An asymmetric key pair where the public portion of it is bound to a user information via use of X.509 public key certificates or other verifiable cryptographic structures.

Attested Subject Key: TPM originated and resident non-migratable or CMK subject key certified by AIK.

Certified Credential: Is a public-key certificate issued by a CA to end entity, where the public-key included into the certificate has been cryptographically bound to an AIK and it includes enough information for the relying party to validate that binding.

Note: An AIK can certify (cryptographically bind) only non-migratable or CMK keys [TCGSPC].

2.2 SKAE Extension Definition

The SKAE extension is identified by the following object identifier:

```
id-tcg-ce-skae-subjectKeyAttestationEvidence OBJECT IDENTIFIER ::=
        { joint-iso-itu-t(2) international-organizations(23) tcg(133)
                ce(6) skae(1) subjectKeyAttestationEvidence(1)  }
```

The syntax of the SKAE extension is defined as follows:

```
SubjectKeyAttestationEvidence ::= SEQUENCE {
        tcgSpecVersion                  TCGSpecVersion,
        keyAttestationEvidence          KeyAttestationEvidence }

KeyAttestationEvidence ::= CHOICE {
        attestEvidence                  [0] AttestationEvidence,
        envelopedAttestEvidence         [1] EnvelopedAttestationEvidence }

AttestationEvidence ::= SEQUENCE {
        tpmCertifyInfo                  TPMCertifyInfo,
        tpmIdentityCredAccessInfo       TPMIdentityCredentialAccessInfo
                    --  Access information to a TPM identity credential used to sign TPMCertInfo --  }

TCGSpecVersion          ::= SEQUENCE {
        major                           INTEGER,
        minor                           INTEGER   }

TPMCertifyInfo ::= SEQUENCE {
        CertifyInfo                     BIT STRING,
                    --  TPM_CERTIFY_INFO structure as a bit string
        signature                       BIT STRING   }

TPMIdentityCredentialAccessInfo  ::= SEQUENCE {
        authorityInfoAccess             AuthorityInfoAccessSyntax,
        issuerSerial                    IssuerSerial        OPTIONAL }

IssuerSerial ::= SEQUENCE {
        issuer                          GeneralNames,
        serialNumber                    CertificateSerialNumber }

EnvelopedAttestationEvidence ::= SEQUENCE {
        recipientInfos                  RecipientInfos,
        encryptedAttestInfo             EncryptedAttestationInfo }

EncryptedAttestationInfo::= SEQUENCE {
        encryptionAlgorithm             AlgorithmIdentifier,
        encryptedAttestEvidence                 OCTET STRING
                    -- The ciphertext resulting from the encryption of  DER-encoded AttestationEvidence -- }
```

The fields of type **SubjectKeyAttestationEvidence** have the following meanings:

- **TCGSpecVersion** is the version number of the TCG TPM main specification.
- **TPMCertifyInfo** contains two fields - TPM_CERTIFY_INFO (or TPM_CERTIFY_INFO2) structure and the **signature.** The **signature** field contains the digital signature upon the TPM_CERTIFY_INFO structure computed using the AIK.
- **TPMIdentityCredentialAccessInfo** indicates how to access Privacy CA information and services for the AIK Credential. *Note:* **AuthorityInfoAccessSyntax** *is defined in [RFC3280]*
- **CertificateSerialNumber** is the serial number of the AIK Credential.
- .
- **EnvelopedAttestationEvidence** field is fulfilling demands to shield the direct reference to the AIK Credential in the user's public key certificate only for the entities referenced in the **RecipientInfo** [RFC3369]. The **EncryptedAttestationInfo** field contains **AttestationEvidence** encrypted with the recipient public key.

2.3 Assumptions

The following assumptions have been made:

1. The client (TPM user) application will need to support inclusion of the new SKAE extension (see Section 6 for SKAE ASN.1 definition) into the certificate request.
2. The CA/RA processing of the certificate request may need to validate SKAE extension and include it into the end entity certificate.
3. TCG aware applications shall be able to validate the SKAE extension or request validation from corresponding validation service providers.

3 Overview of Approach

The following is a simplified architectural model of SKAE extension creation and processing.

The components of this model are:

- TPM user (client, end entity)
- Certification authority (CA) an entity that provides users with public key certificates, optionally it could be accompanied by Registration Authority (RA).
- Privacy CA that issues AIK Credentials (aka Identity Credentials)
- Application Service Provider (Server) is an entity that provides users with services and uses PKI enabled client authentication and authorization methods (firewalls, web servers, email servers, VPN, domain controllers, etc).
- Validation Service (VS) is a trusted service provider entity to which Server may delegate the certificate and/or SKAE extension validation.

Given that:

1. The Server has agreed to provide some kind of special services for TPM enabled clients (exclusive videos, higher limit for money transfer, access to privileged sites, etc).
2. The Server and/or VS have knowledge about SKAE extension and are capable of validating it.
3. The CA is a conventional certification authority with no knowledge about the SKAE extension.
4. The Server trusts the CA and Privacy CA.
5. The TPM user plans to obtain a certificate for the non-migratable or CMK key with an SKAE extension in order to execute the following tasks:
    a. Get authenticated access to a Server and use its special services for the TPM enabled clients.
    b. The TPM user also plans to sign objects and prove to a relying party that the signing key was created within a TPM and is currently resident within a TPM.

Sections 3.1 and 3.2 provide a description of the necessary procedures to solve tasks **5a** and **5b**.

3.1 Obtaining a Certificate with the SKAE Extension

The following figure illustrates an example of the basic data flow of a client obtaining a certificate with an SKAE extension, for either a non-migratable or CMK signing key.

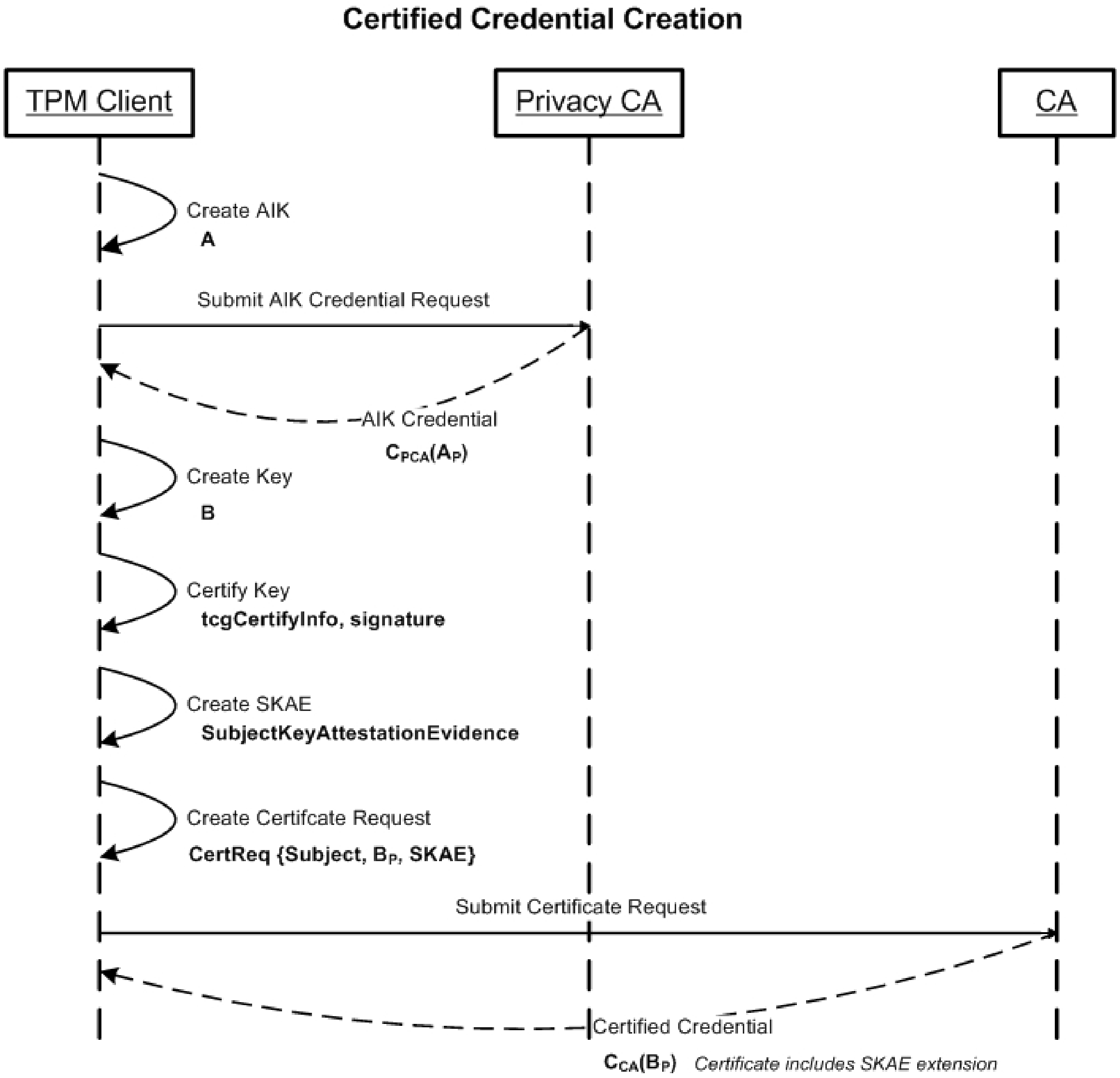

Certified Credential Creation
TPM Client
Privacy CA
CA
Create AIK
A
Submit AIK Credential Request
AIK Credential
C_PCA(A_P)
Create Key
B
Certify Key
tcgCertifyInfo, signature
Create SKAE
SubjectKeyAttestationEvidence
Create Certifcate Request
CertReq {Subject, B_P, SKAE}
Submit Certificate Request
Certified Credential
C_CA(B_P)
Certificate includes SKAE extension

Figure 1. Creation of a certificate with SKAE extension.

A – Attestation identity key pair.
C_{PCA}(A_P) - AIK Credential issued by Privacy CA for the public portion of AIK **A_P**
B - Non-migratable or CMK key pair.
SKAE – SKAE extension as defined in chapter **Error! Reference source not found.**.
CertReq {Subject, B_P, SKAE} – Certificate request [RFC2511], where **Subject** is the certificate subject data.
C_{CA}(B_P) – Public-key certificate issued by CA for the TPM client with the SKAE extension.

The step-by-step procedure of obtaining a certificate with SKAE extension along with corresponding TSS functions calls is outlined below.

Step #	Action
1	The client creates an identity key **A** and prepares the certificate request package for Privacy CA. {Tspi_TPM_CollateIdentityRequest}
2	The client sends the certificate request package to Privacy CA.
3	Privacy CA validates the request and the included credentials, and then issues an AIK Credential and sends it back to the client.
4	The client receives the AIK Credential $\mathbf{C_{PCA}(A_P)}$ from Privacy CA and activates it. {Tspi_TPM_ActivateIdentity}
5	The client generates either a non-migratable or CMK key pair **B** that matches the Server requirements and will be used for authentication purposes with Server. {Tspi_Key_Create}
6	The client certifies the key **B** using the AIK key **A**. {Tspi_Key_CertifyKey (hKey_B, hKey_A, pTSS_Validation)} As a result of that function call the client will get TPM_CERTIFY_INFO structure and the signature data over the TPM_CERTIFY_INFO. **{tpmCertifyInfo, signature}.**
7	The client creates **SKAE** extension (**SubjectKeyAttestationEvidence**).
8	The client creates a certificate request (according either PKCS#10 or CRMF [RFC2511]) for the certified key $\mathbf{B_P}$ that includes also the **SKAE** as an attribute (extension) and submits it to CA.
9	CA issues X.509 v.3 certificate with the **SKAE** extension and sends its back to the client.
10	The client receives the certificate issued by the CA -> $\mathbf{C_{CA}(B_P)}$

3.2 Authenticating the Client

The figure below outlines the process of a TPM client authentication by the Server using PKI certificates.

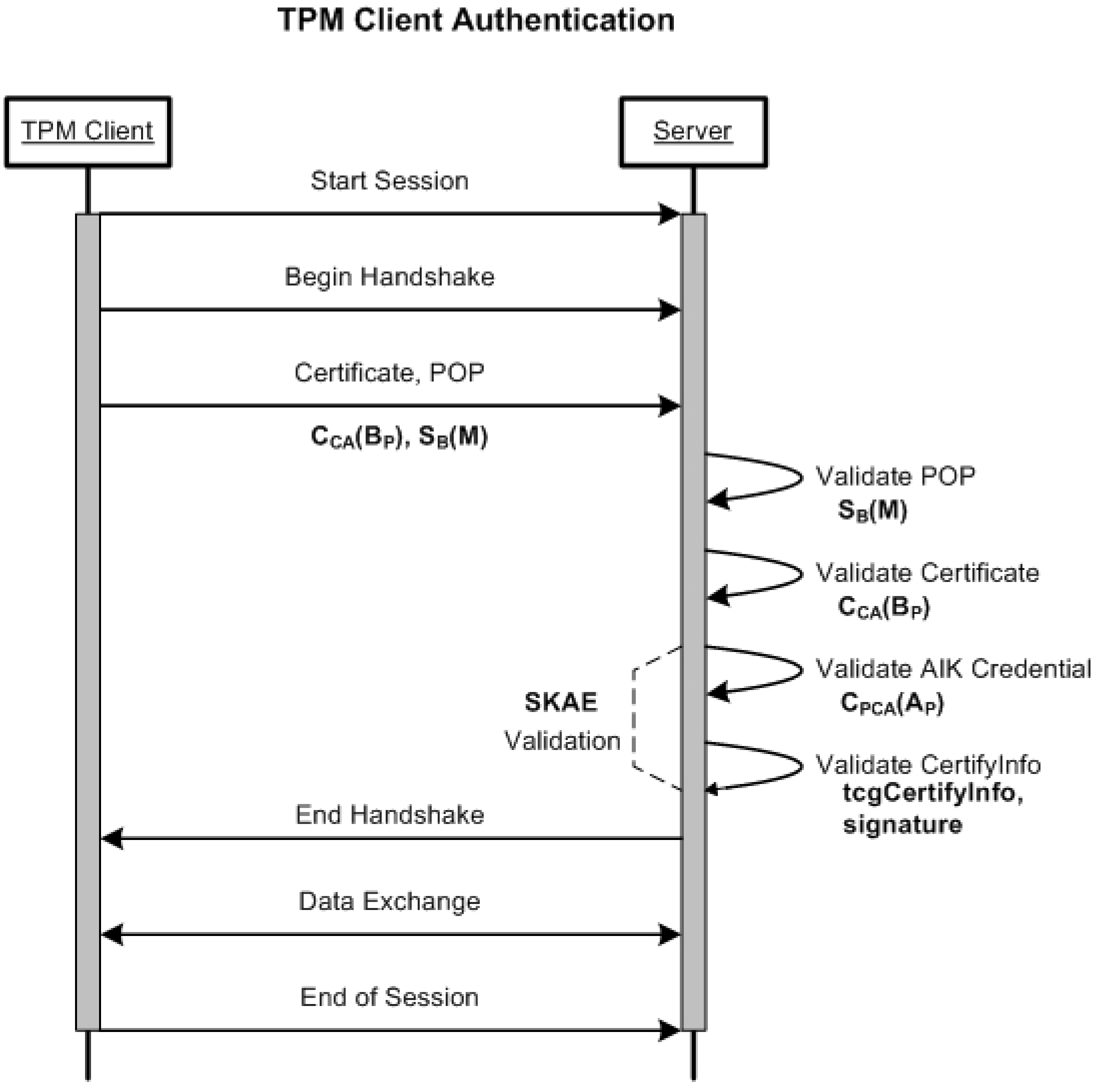


Figure2. TPM Client Authentication

The TPM client and Server perform PKI authentication based on validation of the certificates and signatures exchanged during handshake session similar to [RFC2246, RFC2716].
The step-by-step procedure of TPM client establishing an authenticated session with the Server is outlined below, step 6 especially outlines SKAE validation actions performed by the Server.

Step #	Action
1	TPM client connects to Server and starts a session for data exchange.
2	TPM client and Server start handshaking session to authenticate each other using certificates and generate shared secrets for data exchange.
3	TPM client sends its certificate **$C_{CA}(B_P)$** and proof of possession (POP) of the secret key

<table>
<tr><td></td><td>to the Server (POP = <b>S_B(M)</b> – a message signed using secret key <b>B_S</b>.This is distinctive for PKI aithentication protocols, for instance in [RFC2246] it is referenced as “<code>CertificateVerify</code>”, in IKE [RFC2409] as “signed data”.)</td></tr>
<tr><td>4</td><td>The Server validates the signature of <b>S_B(M)</b>. This will proof the possession of the corresponding secret key <b>B_s</b> by the TPM client.</td></tr>
<tr><td>5</td><td>The Server validates the certificate <b>C_{CA}(B_P)</b> according to certificate validation procedure defined in [RFC3280]</td></tr>
<tr><td>6</td><td>The Server validates SKAE extension as follows:
<table>
<tr><td>6.1</td><td>Obtain the AIK Credential (<b>C_{PCA}(A_P)</b>) referenced in SKAE extension, according to <b>SKAE TPMIdentityCredentialAccessInfo</b>.</td></tr>
<tr><td>6.2</td><td>Validate the AIK Credential according to certificate validation procedure defined in [RFC3280].</td></tr>
<tr><td>6.3</td><td>Verify the <b>signature</b> over TPM_CERTIFY_INFO structure using key <b>A_P</b>.</td></tr>
<tr><td>6.4</td><td>Compute a digest of the public key <b>(B_P)</b> and compare it with the public key digest field inside of TPM_CERTIFY_INFO structure.</td></tr>
</table>
<i>Note:</i> The result of steps 6.3 and 6.4 is validation of the cryptographic binding of key <b>B</b> by the key <b>A</b>.</td></tr>
<tr><td>7</td><td>If above steps are correct, then Server can be sure that <b>the client has a valid TPM and the key (B) is protected by TPM.</b></td></tr>
<tr><td>8</td><td>Server grants the TPM client access to its services according to the configured policy.</td></tr>
<tr><td></td><td></td></tr>
</table>

The order of validation steps 4-6 can be different so long as it derives the correct result.

3.3 Validating SKAE

The TPM user signs or encrypts an object using a private key which has a corresponding certificate with an SKAE extension. The signed or encrypted object and corresponding certified credential are presented to a relying party.

The relying party must validate the cryptographic operations. In the case of a signature operation, the signature validation should occur according to [RFC3126], and the relying party must also validate the certificate according to steps 5 and 6 of section 3.2. If all validation steps return TRUE, then the relying party can be sure that the end entity has signed the object by using a platform with the valid TPM, the signing key was protected by TPM, and the signature (private key operation) was performed “inside” of the TPM. Correspondingly, a public key encryption operation would need to validate according to steps 5 and 6 of section 3.2.

4 Security Considerations

This section highlights issues to be considered by entities implementing applications that utilize TCG\TPM enabled platforms and SKAE extension.

All security considerations from [RFC3369], [RFC3126] and [RFC3280] apply to applications that use procedures described in this document.

4.1 AIK Credential Related Considerations

4.1.1 AIK and Subject Key Secrets

Unless intended by the design, there is no need for an end entity to keep the AIK private key after they have been used to certify the subject key. The AIK Credential, SKAE extension and Certified Credential are compliant with [RFC3126] and are long term documents. The end entity can prove the possession of the subject private key, at the time of the signature, and provide evidence of its binding to an AIK to a relying party at any time, simply by presenting the SKAE.

4.1.2 Privacy Concerns

As was described above, the relying party must validate the binding between the AIK Credential and the user certificate in order to make sure that the end entity is using a valid TPM platform and the keys are protected within that TPM. In case the users have used the same AIK Credential to certify the keys for a number of Certified Credentials with different distinguished names, then someone who has access to these certificates will be able to map them to the user and TPM platform since all SKAE extension will have a reference to the same AIK Credential. This could raise a privacy issue for some implementations.

The “correlation” issue can be resolved by several ways:

a) A particular implementation may allow the user to certify only one subject key by each AIK. In this case, no other entity (but Privacy CA) can map certificates with SKAE extensions to the same TPM platform. Moreover, the implementation may choose to delete (un-register) the AIK key after certifying the subject key, but keep and/or publish the AIK Credential. This approach assumes that there is not performance or economic issues with a Privacy CA issuing a significant number of AIK credentials.

b) The implementers may choose to use an encrypted reference to an AIK Credential using the **EnvelopedAttestationEvidence** form of SKAE. In this case only the chosen trusted verifier will be able to obtain the AIK Credential and validate the subject key binding. This requires the trusted verifiers to be known and specified during the creation of SKAE extension.

c) The implementers may omit **issuerSerial** field and submit it them to the trusted verifier (using a non-specified out-of-band mechanism) only after authenticating its verifier’s identity.

d) The implementers may issue X.509 attribute certificates [RFC3281, X509] with SKAE extension and apply similar to AIK Credentials “need to know” access policy.

Approaches b), c) and d) assume that the verifier is a trusted entity and will not try to correlate usage by the user.

It is assumed that there are no correlation concerns within an Enterprise.

4.2 Validation Algorithm

Conforming implementations of this specification are not required to follow the validation algorithm specified in step 6 of figure 2 in paragraph 3.2, but must provide equivalent functionality so long as it derives the correct result.

4.3 Attested Subject Key Cloning and Migration

The attested subject key cloning and migration procedures are out of the scope of the SKAE extension and this specification.

5 Deployment Analyses

If the subject key TPM origin and residence are going to be validated by the relying party, then the SKAE extension must be included into the end entity certificate. This will allow the relying party to process the SKAE extension to validate the subject key origin during the authentication of the end entity.

The CA/RA may provide the validation of the SKAE extension during the certificate issuance along with the validation of the binding of the subject's identity to their public key. The CA CPS (Certification Practice Statement) must adequately reflect the assurances provided by any validation of the SKAE extension.

The implementers may choose to dedicate a special CA which will validate the SKAE extension during the certificate issuance and will not include it into the end entity certificates. In this case, the relying party should examine the CA's certificate practice statement and implement similar "diligence" for it decision making procedures.

The implementers may chose not to include the SKAE extension into the certificate (or certificate request) and use any other trustworthy out-of-band distribution methods to deliver it to the relying party. These distribution methods are security critical processes that are beyond the scope of this specification.

The table below summarizes possible scenarios.

#	End Entity	CA\RA	Relying Party	Notes
1	Submits PKCS#10 & SKAE to CA	Does not validate SKAE Issues certificate with SKAE	Validates certificate & SKAE	
2	Submits PKCS#10 & SKAE to CA	Validates SKAE Issues certificate with SKAE	Validates certificate & SKAE	CA must state in CPS SKAE validation
3	Submits PKCS#10 & SKAE to CA	Validates SKAE Issues certificate without SKAE	Validates certificate	CA must state in CPS SKAE validation and scope. Serves only TPM users
4	Submits PKCS#10 to CA Sends SKAE out-of-band to RP	Issues certificate without SKAE	Validates certificate Receives SKAE out-of-band Validates SKAE	

The end entity must be capable of creating an SKAE extension for all cases; additionally for case 4 the end entity must support a new protocol for SKAE exchange with the relying party.

Case 1 is anticipated to be the most applicable case (for the early stage of TCG deployment); only the interested parties must implement the necessary changes to process the new objects and obtain/grant the corresponding privileges.

Case 2 should be considered for mission-critical implementations where all parties have incentive to exchange only validated objects.

In case 2 and 3, the CA/RA is taking on a new liability by processing and validating the SKAE extension. Case 3 narrows the scope of the CA only for the TPM enabled users. Alternatively, case 3 can be used for promotional causes in early stage of TCG deployment where not many parties would be able to process and validate SKAE and other TCG objects.

The relying party must be able to process and validate the SKAE extension for cases 1, 2 and 4. Cases 2 and 3 require the relying party to review the CPS and configure its “trust list”. Case 4 requires a new protocol for SKAE exchange with end entity.

6 Appendix A. ASN.1 Module

This normative appendix describes the SKAE extension used by conforming PKI components in ASN.1 syntax.

```
SubjectKeyAttestationEvidence     {joint-iso-itu-t(2)     international-
organizations(23) tcg(133) ce(6) skae(1) module(0) }

DEFINITIONS IMPLICIT TAGS ::=
BEGIN
-- EXPORTS ALL –

IMPORTS
--PKIX Part 1 – Implicit
      GeneralNames, AuthorityInfoAccessSyntax
      FROM PKIX1Implicit88 {iso(1) identified-organization(3) dod(6)
      internet(1) security(5) mechanisms(5) pkix(7) id-mod(0)
      id-pkix1-implicit-88(2)}

      -- PKIX Part 1 – Explicit
      CertificateSerialNumber, AlgorithmIdentifier
      FROM PKIX1Explicit88 {iso(1) identified-organization(3) dod(6)
      internet(1) security(5) mechanisms(5) pkix(7) id-mod(0)
      id-pkix1-explicit-88(1)}

      -- Cryptographic Message Syntax
      RecipientInfos
      FROM CryptographicMessageSyntax {iso(1) member-body(2) us(840)
      rsadsi(113549)  pkcs(1)  pkcs-9(9)  smime(16)  modules(0)  cms-
      2001(14) }

-- the OID used to identify the certificate extension
id-tcg-ce-skae-subjectKeyAttestationEvidence     OBJECT IDENTIFIER ::=
      { joint-iso-itu-t(2) international-organizations(23) tcg(133)
            ce(6) skae(1)  subjectKeyAttestationEvidence(1)  }

-- Subject Key Attestation Evidence extension

SubjectKeyAttestationEvidence ::= SEQUENCE {
      tcgSpecVersion               TCGSpecVersion,
      keyAttestationEvidence       KeyAttestationEvidence }

KeyAttestationEvidence ::= CHOICE {
      attestEvidence               [0] AttestationEvidence,
      envelopedAttestEvidence [1] EnvelopedAttestationEvidence }

AttestationEvidence ::= SEQUENCE {
      tpmCertifyInfo                TPMCertifyInfo,
      tpmIdentityCredAccessInfo     TPMIdentityCredentialAccessInfo
          -- Access information to a TPM identity credential used to
          -- sign TPMCertifyInfo -- }

TCGSpecVersion    ::= SEQUENCE {
      major                         INTEGER,
      minor                         INTEGER   }

TPMCertifyInfo ::= SEQUENCE {
      tpmCertifyInfo                BIT STRING,
              -- TCG_CERTIFY_INFO structure as a bit string
      signature                     BIT STRING   }

TPMIdentityCredentialAccessInfo  ::= SEQUENCE {
      authorityInfoAccess           AuthorityInfoAccessSyntax,
      issuerSerial                  IssuerSerial OPTIONAL   }

IssuerSerial ::= SEQUENCE {
      issuer                        GeneralNames,
      serialNumber                  CertificateSerialNumber }


EnvelopedAttestationEvidence ::= SEQUENCE {
      recipientInfos                RecipientInfos,
      encryptedAttestInfo           EncryptedAttestationInfo }

EncryptedAttestationInfo::= SEQUENCE {
      encryptionAlgorithm           AlgorithmIdentifier,
      encryptedAttestEvidence       OCTET STRING
               -- The ciphertext resulting from the encryption of
               -- DER-encoded AttestationEvidence -- }
END
```

7 Appendix B. Applying SKAE Extension to IWG Use Cases

The table below includes references to IWG and TNC use cases [UC] where SKAE extension can be used for validating the binding of the user (or a device) signing and/or encryption keys to the TPM enabled platform through the AIK Credential.

UC ID	Name	Binding step #	Validation step #
UC1	Banking application that requires proof of signing key to platform binding	2,3	4
UC10	Certificate services for platform identities	1	
UC20	Client platform authentication & attestation by web server via TLS channel	7	8
UC22	Data migration enrollment and lifecycle – Proving to Migration Authority that client has a valid TPM	2	6
UC19	Platform identity key and credential provenance	2	4
UC27	Automated credential management for network devices / equipment	2	3, 4
UC28	Platform authentication on network connect	2, 3	6, 7
UC33	Centralized firewall policy creation and safe distribution	3	4
UC37	Motherboard failure and repair	2	3
UC38A	Digital Signature Application – Acquisition of Signing Credential	12	N+1
UC-AC1	An access requester may at different times be allowed access to networks with different compliance requirements	8	9

8 References

[RAI] TCG Infrastructure Committee, Reference Architecture for Interoperability, Specification Version 1.0, Rev. 0.01, 2 May 2004

[RFC2246] RFC 2246 The TLS Protocol Version 1.0

[RFC2314] RFC 2314 PKCS #10: Certification Request Syntax

[RFC2409] RFC 2409 The Internet Key Exchange

[RFC2511] RFC 2511 Internet X.509 Certificate Request Message Format

[RFC2716] RFC 2716 PPP EAP TLS Authentication Protocol

[RFC3126] RFC 3126 Electronic Signature Formats for long term electronic signatures

[RFC3280] RFC 3280 Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile

[RFC3281] RFC 3281 An Internet Attribute Certificate

[RFC3369] RFC 3369 Cryptographic Message Syntax

[TCGSPC] TCG Main Spec

[UC] TCG Infrastructure Committee, Use Cases, Specification Version 1.0 Rev. 0.25, 27 February 2004

[X509] ITU-T Recommendation X.509: The Directory - Public-Key and Attribute Certificate Frameworks. 2000