Suite B Profile of Certificate Management over CMS  
draft-turner-suiteb-cmc-00.txt

Status of this Memo

This Internet-Draft is submitted to IETF in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at http://www.ietf.org/ietf/1id-abstracts.txt

The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html

This Internet-Draft will expire on June 2, 2010.

Copyright Notice

Copyright (c) 2009 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust’s Legal Provisions Relating to IETF Documents in effect on the date of publication of this document (http://trustee.ietf.org/license-info). Please review these documents carefully, as they describe your rights and restrictions with respect to this document.
Abstract

The United States Government has published guidelines for "NSA Suite B Cryptography", which defines cryptographic algorithm policy for national security applications. This document specifies a profile of the Certificate Management over CMS (CMC) protocol for managing Suite B X.509 public key certificates. This profile is a refinement of RFC5272, RFC5273, and RFC5274.

1. Introduction

This document specifies a profile for using the Certificate Management over CMS (CMC) protocol, defined in [RFC5272], [RFC5273], and [RFC5274], to manage X.509 public key certificates compliant with the United States National Security Agency’s Suite B Cryptography as defined in the Suite B Certificate and Certificate Revocation List (CRL) Profile [SBCERT]. This document specifically focuses on defining CMC interactions for both initial enrollment and rekey of Suite B public key certificates between a client and a Certification Authority (CA). One or more Registration Authorities (RAs) may act as intermediaries between the client and the CA. This profile may be further tailored by specific communities to meet their needs. Specific communities will also define Certificate Policies that implementations must comply with.

2. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

The terminology in [RFC5272] Section 2.1 applies to this profile.

3. Requirements and Assumptions

All key pairs are on either the curve P-256 or the curve P-384. FIPS 186-3 [FIPS1863] or NIST Special Publication 800-56A [SP80056A] provides useful guidance for elliptic curve key pair generation.

This document assumes that the required trust anchors have been securely provisioned to the client and, when applicable, any RAs.

All requirements in [RFC5272], [RFC5273], and [RFC5274] apply, except where overridden by this profile.

This profile was developed with the scenarios described in Appendix A in mind. However, use of this profile is not limited to just those scenarios.
The term "client" in this profile typically refers to an End-Entity (EE). However, it may instead refer to a third party acting on the End-Entity’s behalf. The client may or may not be the entity that actually generates the key pair, but it does perform the CMC protocol interactions with the RA and/or CA. For example, the client may be a token management system that communicates with a cryptographic token through an out-of-band secure protocol.

This profile may be used to manage RA or CA certificates. In that case, the RA or CA whose certificate is being managed is considered to be the End-Entity.

4. Client Requirements: Generating PKI Requests

This section specifies the conventions employed when a client requests a certificate from a Public Key Infrastructure (PKI).

The Full PKI Request MUST be used and it MUST be encapsulated in a SignedData, as per [RFC5008]. The PKIData content type complies with [RFC5272] with the following additional guidance:

- controlSequence MUST be present. It MUST include the following CMC controls: Transaction ID and Sender Nonce. Other CMC controls MAY be included.

- If the request is being authenticated using a shared secret, then the following guidance in this paragraph applies: Identity Proof Version 2 control MUST be included. hashAlgId MUST be id-sha256 for P-256 certificate requests, and MUST be id-sha384 for P-384 certificate requests, as per [SHA2]. macAlgId MUST be HMAC-SHA256 for P-256 certificate requests, and MUST be HMAC-SHA384 for P-384 certificate requests, as per [RFC4231]. If the subject included in the certificate request is NULL or otherwise does not uniquely identify the End-Entity, then the POP Link Random control MUST be included to prevent substitution attacks, and the POP Link Witness Version 2 control MUST be included in the inner PKCS #10 or CRMF request as described in Sections 4.1 and 4.2.

- reqSequence MUST be present. It MUST include at least one tcr (see Section 4.1) or crm (see Section 4.2) TaggedRequest. Support for the orm choice is OPTIONAL.

If the Full PKI Request contains a P-256 public key, then the SignedData encapsulating the Full PKI Request MUST be generated using either SHA-256 and ECDSA with P-256, or using SHA-384 and ECDSA with P-384, as per [RFC5008]. If the Full PKI Request contains a P-384 public key, then the SignedData MUST be generated using SHA-384 and ECDSA with P-384, as per [RFC5008].
The Full PKI Request SHOULD be signed using the private key that corresponds to the subject public key of an existing signature certificate. However, during initial enrollment, an appropriate signature certificate may not yet exist. If the Full PKI Request includes one or more signature certificate requests, then for initial enrollment authenticated using a shared secret when no appropriate certificate yet exists, the Full PKI Request MAY instead be signed using the private key corresponding to the subject public key of one of the requested signature certificates. The Full PKI Request MUST NOT ever be signed using a key pair intended for use in a key establishment certificate.

4.1. Tagged Certificate Request

The reqSequence tcr choice conveys PKCS #10 [RFC2986] syntax. The CertificateRequest MUST comply with [RFC5272] Section 3.2.1.2.1 with the following additional guidance:

- subjectPublicKeyInfo MUST be set as defined in 4.4 of [SBCERT];

- attributes MUST include one ExtensionReq attribute. The Key Usage extension MUST be included and it MUST be set as per [SBCERT]. For rekey, if the subject is NULL, then the SubjectAltName extension MUST be included and set equal to the SubjectAltName of the certificate being rekeyed. Other extension requests MAY be included as desired.

- For non-rekey requests, if the Full PKI Request encapsulating this Tagged Certificate Request is being signed by a key for which a certificate currently exists, and the Subject in the certificate request is NULL or otherwise does not uniquely identify the End-Entity, then the SigningCertificateV2 attribute, as defined in [RFC5035], SHOULD also be included in the attributes field. Support for the policies field is optional. The ESSCertIDV2 hashAlgorithm field MUST be set to id-sha256 for P-256 certificate requests and id-sha384 for P-384 certificate requests. Presence of this attribute protects against substitution attacks.

- If the request is being authenticated using a shared secret, and the Subject in the certificate request is NULL or otherwise does not uniquely identify the End-Entity, then the POP Link Witness Version 2 control MUST be included in the attributes field. keyGenAlgorithm MUST be id-sha256 for P-256 certificate requests and MUST be id-sha384 for P-384 certificate requests. macAlgorithm MUST be HMAC-SHA256 for P-256 certificate requests, and MUST be HMAC-SHA384 for P-384 certificate requests. Presence of this attribute protects against substitution attacks.
signatureAlgorithm MUST be ecdsa-with-sha256 for P-256 certificate requests, and MUST be ecdsa-with-sha384 for P-384 certificate requests;

- signature MUST be generated using the private key corresponding to the public key in the CertificationRequestInfo, for both signature and key establishment certificate requests. The signature provides proof of possession of the private key to the Certification Authority.

### 4.2. Certificate Request Message

The reqSequence crm choice conveys Certificate Request Message Format (CRMF) [RFC4211] syntax. The CertReqMsg MUST comply with [RFC5272] Section 3.2.1.2.2 with the following additional guidance:

- popo MUST be included using the signature (POPOSigningKey) Proof of Possession choice and set as defined in [RFC4211] section 4.1 for both signature and key establishment certification requests. The POPOSigningKey poposkInput field MUST be omitted. The POPOSigningKey algorithmIdentifier MUST be ecdsa-with-sha256 for P-256 certificate requests, and MUST be ecdsa-with-sha384 for P-384 certificate requests. The signature MUST be generated using the private key corresponding to the public key in the CertTemplate.

The CertTemplate MUST comply with [RFC5272] Section 3.2.1.2.2 with the following additional guidance:

- version MAY be included and, if included, it MUST be set to 2 as per paragraph 4.3 of [SBCERT];

- publicKey MUST be set as defined in 4.4 of [SBCERT];

- extensions MUST include at least the Key Usage extension and it MUST be set as per [SBCERT]. For rekey, if the subject is NULL, then the SubjectAltName extension MUST be included and set equal to the SubjectAltName of the certificate being rekeyed. Other extension requests MAY be included as desired.

For non-rekey requests, if the Full PKI Request encapsulating this Certificate Request Message is being signed by a key for which a certificate currently exists, and the Subject in the certificate request is NULL or otherwise does not uniquely identify the End-Entity, then the SigningCertificateV2 attribute, as defined in [RFC5035], SHOULD be included in the controls field. Support for the policies field is optional. The ESSCertIDV2 hashAlgorithm field MUST be set to id-sha256 for P-256 certificate requests and id-sha384 for
P-384 certificate requests. Presence of this attribute protects against substitution attacks.

If the request is being authenticated using a shared secret, and the Subject in the certificate request is NULL or otherwise does not uniquely identify the End-Entity, then the POP Link Witness Version 2 control MUST be included in the controls field. keyGenAlgorithm MUST be id-sha256 for P-256 certificate requests and MUST be id-sha384 for P-384 certificate requests. macAlgorithm MUST be HMAC-SHA256 for P-256 certificate requests and MUST be HMAC-SHA384 for P-384 certificate requests. Presence of this attribute protects against substitution attacks.

5. RA Requirements: Processing PKI Requests (Client to CA)

This section addresses the optional case where one or more RAs act as intermediaries between the client and CA as described in Section 7 of [RFC5272]. In this section, the term "client" refers to the entity from which the RA received the PKI Request. This section is only applicable to RAs.

5.1. RA Generated PKI Requests

If the RA encapsulates the client-generated PKI Request in a new RA-signed PKI Request, it will create a Full PKI Request encapsulated in a SignedData. If the request contains a certification request for a P-256 public key, then the SignedData MUST be generated using either SHA-256 and ECDSA with P-256 or SHA-384 and ECDSA with P-384, as per [RFC5008]. If the request contains a certification request for a P-384 public key, then the SignedData MUST be generated using SHA-384 and ECDSA with P-384, as per [RFC5008]. The PKIData content type complies with [RFC5272] with the following additional guidance:

- controlSequence MUST be present. It MUST include the following CMC controls: Transaction ID and Sender Nonce. Other appropriate CMC controls MAY be included.

- cmsSequence MUST be present. It contains the original, unmodified request received from the client.

6. CA Requirements

This section specifies the requirements for CAs that receive PKI Requests and that generate PKI Responses.
6.1. CA Processing of PKI Requests

CAs MUST ensure that only the permitted signature, hash, and MAC algorithms described throughout this profile are used, and otherwise reject the request.

For requests involving an RA, the CA MUST verify the RA’s authorization. Only RAs can include the Modify Certification Request control. The following certificate fields MUST NOT be modifiable using the Modify Certification Request control: version, publicKey, and the key usage extension. The request MUST be rejected if an attempt to modify those certificate request fields is present.

If the client generated PKI Request includes a SigningCertificateV2 attribute either in the CertRequest controls field for a CRMF request or in the tcr attributes field for a PKCS#10 request, then the CA SHOULD ensure that the certificate referenced in the attribute corresponds to the private key used to sign the PKI Request. If the CA performs this check, and the referenced certificate doesn’t correspond to the private key used to sign the PKI Request, then the CA MUST reject the PKI Request.

6.2. CA Generated PKI Responses

The Full PKI Response MUST be used and it MUST be encapsulated in a SignedData, as per [RFC5008]. The PKIResponse content type complies with [RFC5272] with the following additional guidance:

- controlSequence MUST be included. It MUST include the following CMC controls: Extended CMC Status Info, Transaction ID, Sender Nonce, and Recipient Nonce. Other appropriate CMC controls MAY be included.

- If the SigningCertificateV2 check is performed and does not pass, then the Extended CMC Status Info CMCStatus value MUST be set to failed.

If the PKI Response is in response to an RA encapsulated PKI Request, then the above PKI Response is encapsulated in another CA generated PKI Response. That PKI Response MUST be encapsulated in a SignedData, as per [RFC5008]. The above PKI Response is placed in the encapsulating PKI Response cmsSequence field. The other fields are as above. The following illustrates a successful CA response to an RA encapsulated PKI Request:

SignedData (applied by the CA)
PKIData
controlSequence (Extended CMC Status Info, Transaction ID, Sender Nonce, Recipient Nonce)
The same private key used to sign certificates MUST NOT be used to sign Full PKI Response messages. Instead, a separate certificate authorized to sign CMC responses MUST be used. Certificates authorized to sign Full PKI Responses SHOULD include the id-kp-cmcCA following Extended Key Usage (EKU):

```
id-kp-cmcCA OBJECT IDENTIFIER ::= { id-TBSL }
```

The certificate authorized to sign Full PKI Responses MAY also include the Cryptographic Message Syntax (CMS) Content Constraints (CCC) certificate extension [CCC]. CCC SHOULD be included if constraints are to be placed on the content types generated.

The signature on the SignedData MUST be generated using either ECDSA P-256 with SHA-256 or ECDSA P-384 with SHA-384. If the Full PKI Response is a successful response to a P-256 public key certificate request, then the SignedData MUST be generated using either SHA-256 and ECDSA with P-256 or SHA-384 and ECDSA with P-384, as per [RFC5008]. If the Full PKI Response is a successful response to a P-384 public key certificate request, then the SignedData MUST be generated using SHA-384 and ECDSA with P-384, as per [RFC5008]. If the Full PKI Response is an unsuccessful response to a PKI Request, then the SignedData MUST be signed by either SHA-256 and ECDSA with P-256 or SHA-384 and ECDSA with P-384. If the Full PKI Response is a successful response to a PKI Request that only contained a Get Certificate or Get CRL control, then the SignedData MUST be signed by either SHA-256 and ECDSA with P-256 or SHA-384 and ECDSA with P-384.

7. Client Requirements: Processing PKI Responses

Clients MUST authenticate all PKI Responses. This includes verifying that the PKI Response is signed by a CA whose certificate validates back to a trust anchor and that the CA’s certificate either includes the id-kp-cmcCA EKU, includes an appropriate CMS Content Constraints extension, or is determined to be authorized to sign responses through an implementation specific mechanism. The PKI Response MAY be signed by an RA if it is an error message, or if the PKI Response contains an inner PKI Response signed by a CA. In that case, each layer of PKI Response must still contain a valid signature signed by an entity with a valid certificate that verifies back to an acceptable trust anchor.
When a newly issued certificate is included in the PKI Response, the client MUST verify that the newly issued certificate's public key matches the public key that the client requested. The client MUST also ensure that the certificate's signature is valid and that the signature validates back to an acceptable trust anchor.

PKI Responses that do not pass these tests MUST be rejected.

8. CMC Controls

When the Identity Proof V2 control is used, the shared-secret MUST be randomly generated and securely distributed. The shared-secret MUST provide at least 128 bits of strength for P-256 certificate requests and at least 192 bits of strength for P-384 certificate requests.

9. Security Considerations

The security considerations in [RFC5272], [RFC5273], and [RFC5274] apply.

Compliant with NIST Special Publication 800-57 [SP80057], this profile defines proof-of-possession of a key establishment private key by performing a digital signature. Except for one-time proof-of-possession, a single key pair MUST NOT be used for both signature and key establishment.

This specification requires implementations to generate key pairs and other random values. The use of inadequate pseudo-random number generators (PRNGs) can result in little or no security. The generation of quality random numbers is difficult. NIST Special Publication 800-90 [SP80090] or FIPS 186 [FIPS1863] may offer guidance.

When RAs are used, the list of authorized RAs must be securely distributed out-of-band to CAs.

10. IANA Considerations

None: All identifiers are already registered. Please remove this section prior to publication as an RFC.

11. References

11.1. Normative References


/** RFC EDITOR: Please replace "SBCERT" with RFC#### when draft-solinas-suiteb-cert-profile is published.**/

11.2. Informative References


Authors’ Addresses

Michael Peck
National Security Agency
9800 Savage Road Ste 6704
Ft. Meade, MD 20755-6704
USA

Email: mpeck@restarea.ncsc.mil

Sean Turner
IECA, Inc.
3057 Nutley Street, Suite 106
Fairfax, VA 22031
USA

Email: turners@ieca.com

Appendix A. Scenarios

This section illustrates several potential certificate initial enrollment and rekey scenarios supported by this profile. This section does not intend to place any limits or restrictions on the use of CMC.
A.1. Initial Enrollment of Signature or CA Certificates

This section describes three scenarios for authenticating initial signature or CA certificate enrollment requests:

1. Previously installed signature certificate (e.g., Manufacturer Installed Certificate);

2. Shared secret distributed securely out-of-band;

3. RA authentication.

A.1.1. Previously Installed Signature Certificate

In this scenario, the EE has had a signature certificate installed by the cryptographic module manufacturer. As the EE already has a signature certificate, it can be used to authenticate a request for a new certificate. The EE signs the Full PKI Request with the private key that corresponds to the subject public key of a previously installed signature certificate. The CA will recognize the authorization of the previously installed certificate and issue an appropriate certificate to the EE. The Subject in the previously installed certificate and in the newly issued certificate are not required to match.

A.1.2. Shared Secret Distributed Securely Out-of-Band

In this scenario, the CA distributes a shared secret out-of-band to the EE that the EE uses to authenticate its certificate request. The EE signs the Full PKI Request with the private key for which the certification is being requested. The EE includes the Identity Proof Version 2 control to authenticate the request using the shared secret. The CA uses either the Identification control or the Subject in the EE’s enclosed PKCS #10 or CRMF certification request message to identify the request. The EE performs either the POP Link Witness Version 2 mechanism as described in [RFC5272] section 6.3.1.1 or the Shared-Subject/Subject DN Linking mechanism as described in [RFC5272] section 6.3.2. The Subject in the enclosed PKCS #10 or CRMF certificate request does not necessarily match the issued certificate, as it may just be used to help identify the request (and corresponding shared secret) to the CA.

A.1.3. RA Authentication

In this scenario, the EE does not automatically authenticate its enrollment request to the CA, either because the EE has nothing to authenticate the request with, or because organizational policy requires RA involvement. The EE creates a Full PKI Request and sends it to an RA. The RA verifies the authenticity of the request, then,
if approved, encapsulates and signs the request as described in
Section 5.2, forwarding the new request on to the CA. The Subject in
the PKCS #10 or CRMF certification request is not required to match
the issued certificate, it may just be used to help identify the
request to the RA and/or CA.

A.2. Initial Enrollment of Key Establishment Certificates

This scenario addresses the initial enrollment of End-Entity Key
Establishment Certificates. This scenario requires that the End-
Entity holds a private key corresponding to either a previously
installed signature certificate (see Section A.1.1) or End-Entity
signature certificate. The private key corresponding to the existing
certificate is used to sign the Full PKI Request for the Key
Establishment Certificate.

A.3. Rekey

There are two scenarios to support the rekey of certificates that are
already enrolled. One addresses the rekey of signature certificates
and the other addresses the rekey of key establishment certificates.
Typically, organizational policy will require certificates to be
currently valid to be rekeyed, and may require initial enrollment to
be repeated when rekey is not possible.

A.3.1. Rekey of Signature Certificates

When a signature certificate is rekeyed, the PKCS #10 or CRMF
certification request message enclosed in the Full PKI Request will
include the same Subject as the current signature certificate. The
Full PKI Request will be signed by the current private key
corresponding to the current signature certificate.

A.3.2. Rekey of Key Establishment Certificates

Rekey of a key establishment certificate is handled equivalently to
its initial enrollment as described in Section A.2.