Network Working Group
Internet-Draft
Intended Status: Informational
Expires: December 30, 2010

Lydia Zieglar, NSA
Sean Turner, IECA
Mike Peck
June 30, 2010

Suite B Profile of Certificate Management over CMS
draft-turner-suiteb-cmc-03.txt

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 RFC 5272, RFC 5273, and RFC 5274.

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 December 30, 2010.

1. Introduction

This document specifies a profile for using the Certificate Management over CMS (CMC) protocol, defined in [RFC5272], [RFC5273], and [RFC5274], and updated by [CMCbis], 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 [RFC5759]. 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 need to 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 [DSS] Appendix B.4 provides useful guidance for elliptic curve key pair generation that SHOULD be followed by systems that conform to this document.

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

All requirements in [RFC5272], [RFC5273], [RFC5274], and [CMCbis] 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. 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 uses the term "rekey" in the same manner as does CMC (defined in section 2 of [RFC5272]). The profile makes no specific statements about the ability to do "renewal" operations, however the statements applicable to rekey should be applied to renewal as well.

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

This profile does not support key establishment certificate requests from cryptographic modules that cannot generate a one-time signature with a key establishment key for proof-of-possession purposes. In that case, a separate profile would be needed to define the use of another proof-of-possession technique.

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; it MUST be encapsulated in a SignedData; and the SignedData MUST be constructed as defined in

[RFC5008]. The PKIData content type complies with [RFC5272] with the following additional requirements:

- controlSequence SHOULD be present; and it SHOULD 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 requirements in this paragraph apply: Identity Proof Version 2 control, as defined in [RFC5272], MUST be included; hashAlgId MUST be id-sha256 or id-sha384 for P-256 certificate requests, and MUST be id-sha384 for P-384 certificate requests, both algorithm OIDs are defined in [RFC5754]; macAlgId MUST be HMAC-SHA256 when the hashAlgId is id-sha256, and MUST be HMAC-SHA384 when the hashAlgId is id-sha384, both HMAC algorithms are defined in [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, 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 certificate request, 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. If the Full PKI Request contains a P-384 public key certificate request, then the SignedData MUST be generated using SHA-384 and ECDSA with P-384.

A Full PKI Request MUST be signed using the private key that corresponds to the public key of an existing signature certificate unless an appropriate signature certificate does not yet exist, such as during initial enrollment.

If an appropriate signature certificate does not yet exist, a Full PKI Request includes one or more certificate requests, and is authenticated using a shared secret (because no appropriate certificate exists yet to authenticate the request), the Full PKI Request MUST be signed using the private key corresponding to the

public key of one of the requested certificates. A Full PKI Request MAY be signed using a key pair intended for use in a key establishment certificate when necessary because there is no existing signature certificate and there is no signature certificate request included. However, servers are not required to allow this behavior.

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 requirements:

- o certificationRequestInfo:
  - o subjectPublicKeyInfo MUST be set as defined in 4.4 of [RFC5759];
  - o attributes:
    - o The ExtensionReq attribute MUST be included and contain:

      - o The Key Usage extension MUST be included and it MUST be set as defined in [RFC5759].
      - o For rekey requests, the SubjectAltName extension MUST be included and set equal to the SubjectAltName of the certificate which is being used to sign the SignedData encapsulating the request (i.e., not the certificate being re-keyed) if its Subject field of the certificate being used to generate the signature is NULL.
      - o Other extension requests MAY be included as desired.
    - o The ChangeSubjectName attribute, as defined in [CMCbis], MUST be included if the Full PKI Request encapsulating this Tagged Certificate Request is being signed by a key for which a certificate currently exists and the existing certificate's Subject or SubjectAltName does not match the desired Subject or SubjectAltName of this certificate request.

- The POP Link Witness Version 2 attribute MUST be included 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. In the POP Link Witness Version 2 attribute, keyGenAlgorithm MUST be id-sha256 or id-sha384 for P-256 certificate requests and MUST be id-sha384 for P-384 certificate requests, as defined in [RFC5754]; macAlgorithm MUST be HMAC-SHA256 when the keyGenAlgorithm is id-sha256, and MUST be HMAC-SHA384 when the keyGenAlgorithm is id-sha384, as defined in [RFC4231].

- 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 requirements:

- 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 requirements:

- o version MAY be included and, if included, it MUST be set to 2 as defined in paragraph 4.3 of [RFC5759];

- o publicKey MUST be set as defined in 4.4 of [RFC5759];

- o extensions:

  - o The Key Usage extension MUST be included and it MUST be set as defined in [RFC5759].

  - o For rekey requests, the SubjectAltName extension MUST be included and set equal to the SubjectAltName of the certificate which is being used to sign the SignedData encapsulating the request (i.e., not the certificate being re-keyed) if the Subject field of the certificate being used to generate the signature is NULL.

  - o Other extension requests MAY be included as desired.

- o controls:

  - o The ChangeSubjectName attribute, as defined in [CMCbis], MUST be included if the Full PKI Request encapsulating this Tagged Certificate Request is being signed by a key for which a certificate currently exists and the existing certificate's Subject or SubjectAltName does not match the desired Subject or SubjectAltName of this certificate request.

  - o The POP Link Witness Version 2 attribute MUST be included 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. In POP Link Witness Version 2 attribute, keyGenAlgorithm MUST be id-sha256 or id-sha384 for P-256 certificate requests and MUST be id-sha384 for P-384 certificate requests; macAlgorithm MUST be HMAC-

SHA256 when keyGenAlgorithm is id-sha256 and MUST be HMAC-SHA384 when keyGenAlgorithm is id-sha384.

5. RA Requirements

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 Processing of Requests

RAs conforming to this document MUST ensure that only the permitted signature, hash, and MAC algorithms described throughout this profile are used in requests; if they are not, the CA MUST reject those requests. The RA SHOULD return a CMCFailInfo with the value of badAlg [RFC5272].

When processing end-entity generated SignedData objects, RAs MUST NOT perform Cryptographic Message Syntax (CMS) Content Constraints (CCC) certificate extension [CCC] processing.

Other RA processing is as in [RFC5272].

5.2. RA-Generated PKI Requests

If the RA encapsulates the client-generated PKI Request in a new RA-signed PKI Request, it MUST create a Full PKI Request encapsulated in a SignedData and the SignedData MUST be constructed as defined in [RFC5008]. The PKIData content type complies with [RFC5272] with the following additional requirements:

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

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

RA certificates are authorized to sign Full PKI Requests either with an Extended Key Usage (EKU) and/or with the CCC certificate extension [CCC]. Certificates may also be authorized through local configuration. Authorized Certificates SHOULD include the id-kp-cmcRA Extended Key Usage (EKU) from [CMCbis]. Authorized certificates MAY also include the CCC certificate extension [CCC] or authorized certificate MAY just include the CCC certificate extension. If the RA is authorized via the CCC extension, then the CCC extension MUST include the object identifier for the PKIData

content type. CCC SHOULD be included if constraints are to be placed on the content types generated.

If the RA-signed PKI 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. 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. If the RA-signed PKI Request contains requests for certificates on the P-256 and P-384 curve, then the SignedData MUST be generated using 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.

5.3. RA-Generated Errors

RA certificates authorized with the CCC certificate extension [CCC] MUST include the object identifier for the PKIResponse content type to authorize them to generate responses.

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 conforming to this document MUST ensure that only the permitted signature, hash, and MAC algorithms described throughout this profile are used in requests; if they are not, the CA MUST reject those requests. The CA SHOULD return a CMCStatusInfoV2 control with CMCStatus of failed and a CMCFailInfo with the value of badAlg [RFC5272].

For requests involving an RA, the CA MUST verify the RA's authorization. The following certificate fields MUST NOT be modifiable using the Modify Certification Request control: publicKey and the key usage extension. The request MUST be rejected if an attempt to modify those certificate request fields is present. The CA SHOULD return a CMCStatusInfoV2 control with CMCStatus of failed and a CMCFailInfo with a value of badRequest.

When processing end-entity generated SignedData objects, RAs MUST NOT perform Cryptographic Message Syntax (CMS) Content Constraints (CCC) certificate extension [CCC] processing.

If the client-generated PKI Request includes a ChangeSubjectName 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 MUST ensure that name change is authorized. The mechanism for ensuring that the name change is authorized is out-of-scope. If the CA performs this check, and the name change is not authorized, then the CA MUST reject the PKI Request. The CA SHOULD return a CMCStatusInfoV2 control with CMCStatus of failed.

Other processing of PKIRequests is as in [RFC5272].

6.2. CA-Generated PKI Responses

If a Full PKI Response is returned; it MUST be encapsulated in a SignedData; and the SignedData MUST be constructed as defined in [RFC5008].

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 and the SignedData MUST be constructed as defined in [RFC5008]. The above PKI Response is placed in the encapsulating PKI Response cmsSequence field. The other fields are as above with the addition of the batch response control in controlSequence. The following illustrates a successful CA response to an RA-encapsulated PKI Request both of which include Transaction IDs and Nonces:

```
SignedData (applied by the CA)
  PKIData
    controlSequence (Transaction ID, Sender Nonce, Recipient
                     Nonce, Batch Response)
    cmsSequence
      SignedData (applied by CA and includes returned
                  certificates)
        PKIData
          controlSequence (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 are authorized to sign Full PKI Responses with an Extended Key Usage (EKU) and/or with the Cryptographic Message Syntax (CMS) Content Constraints (CCC) certificate extension [CCC]. Certificates may also be authorized through local configuration. Authorized Certificates

SHOULD include the id-kp-cmcCA EKU from [CMCbis]. Authorized certificates MAY also include the CCC certificate extension [CCC] or authorized certificate MAY just include the CCC certificate extension. If the CA is authorized via the CCC extension, then the CCC extension MUST include the object identifier for the PKIResponse

content type. 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. 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. If the Full PKI Response is a successful response to certificate requests on both the P-256 and P-356 curves, then the SignedData MUST be generated using SHA-384 and ECDSA with P-384. 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 an unsuccessful response to certificate requests on both the P-256 and P-356 curves, then the SignedData MUST be generated using 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.

If the PKI Response is in response to an RA-encapsulated PKI Request, the signature algorithm for each SignedData is selected independently.

7. Client Requirements: Processing PKI Responses

Clients conforming to this document MUST ensure that only the permitted signature, hash, and MAC algorithms described throughout this profile are used in responses; if they are not, the client MUST reject those responses.

Clients MUST authenticate all Full PKI Responses. This includes verifying that the PKI Response is signed by an authorized CA or RA

whose certificate validates back to a trust anchor. The authorized CA certificate MUST include the id-kp-cmcCA EKU and/or include a CCC extension that includes the object identifier for the PKIResponse content type. Or, the CA is determined to be authorized to sign responses through an implementation-specific mechanism. The PKI Response can be signed by an RA if it is an error message, if it a response to a Get Certificate or Get CRL request, or if the PKI Response contains an inner PKI Response signed by a CA. In the later case, each layer of PKI Response MUST still contain an authorized, valid signature signed by an entity with a valid certificate that verifies back to an acceptable trust anchor. The authorized RA certificate MUST include the id-kp-cmcRA EKU and/or include a CCC extension that includes the object identifier for the PKIResponse

content type. Or, the RA is determined to be authorized to sign responses through an implementation-specific mechanism.

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.

Clients MUST reject PKI Responses that do not pass these tests. Local policy will determine whether the client returns a Full PKI Response with an Extended CMC Status Info control with CMCStatus set to failed to a user console, error log, or the server.

If the Full PKI Response contains an Extended Status Info with a CMCStatus set to failed, then local policy will determine whether the client resends a duplicate certificate request back to the server or whether an error state is returned to a console or error log.

8. Shared Secrets

When the Identity Proof V2 and POP Link Witness V2 controls are 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

Protocol security considerations are found in [RFC2986], [RFC4211], [RFC5008], [RFC5272], [RFC5273], [RFC5274], [RFC5759], and [CMCbis]. When CCC is used to authorize RA and CA certificates, then the security considerations in [CCC] also apply. Algorithm security considerations are found in [RFC5008].

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], FIPS 186-3 [DSS], and [RFC4086] offer random number generation guidance.

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

Presence of the POP Link Witness Version 2 and POP Link Random attributes protect against substitution attacks.

The Certificate Policy for a particular environment will specify whether expired certificates can be used to sign certificate requests.

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

[CCC] Housley, R., Wallace, C., and S. Ashmore, "Cryptographic Message Syntax (CMS) Content Constraints X.509 Certificate Extension", draft-housley-cms-content-

constraints-extn-06, work-in-progress.

[CMCbis] Schaad, J., "Certificate Management over CMS (CMC) Updates", draft-ietf-pkix-rfc5272-bis-00.txt, work-in-progress.

[DSS] National Institute of Standards and Technology (NIST), FIPS 186-3: Digital Signature Standard (DSS), June 2009.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", RFC 2119, BCP 14, March 1997.

[RFC2986] Kaliski, B., "PKCS #10: Certification Request Syntax v1.5", RFC 2986, November 2000.

[RFC4086] Eastlake, D., 3rd, Schiller, J., and S. Crocker, "Randomness Requirements for Security", BCP 106, RFC 4086, June 2005.

[RFC4211] J. Schaad, "Internet X.509 Public Key Infrastructure Certificate Request Message Format (CRMF)", RFC 4211, September 2005.

[RFC4231] M. Nystrom, "Identifiers and Test Vectors for HMAC-SHA-224, HMAC-SHA-256, HMAC-SHA-384, and HMAC-SHA-512", RFC 4231, December 2005.

[RFC5008] Solinas, J. and R. Housley, "Suite B in Secure/Multipurpose Internet Mail Extensions (S/MIME)", RFC 5008, September 2007.

[RFC5272] Schaad, J. and M. Myers, "Certificate Management over CMS (CMC)", RFC 5272, June 2008.

[RFC5273] Schaad, J. and M. Myers, "Certificate Management over CMS (CMC): Transport Protocols", RFC 5273, June 2008.

[RFC5274] Schaad, J. and M. Myers, "Certificate Management Messages over CMS (CMC): Compliance Requirements", RFC 5274, June 2008.

[RFC5754] S. Turner, "Using SHA2 Algorithms with CMS", RFC 5754,

January 2010.

[RFC5759] Solinas, J., and L. Zieglar, "Suite B Certificate and Certificate Revocation List (CRL) Profile", RFC5759, January 2010.

11.2. Informative References

[SP80057] National Institute of Standards and Technology (NIST), Special Publication 800-57 Part 1: Recommendation for Key Management, March 2007.

[SP80090] National Institute of Standards and Technology (NIST), Special Publication 800-90: Recommendation for Random Number Generation Using Deterministic Random Number Bits Generators (Revised), March 2007.

Appendix A. Scenarios

This section illustrates several potential certificate 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

This section describes three scenarios for authenticating initial 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 end-entity has had a signature certificate installed by the cryptographic module manufacturer. As the end-entity already has a signature certificate, it can be used to authenticate a request for a new certificate. The end-entity 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 end-entity.

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

In this scenario, the CA distributes a shared secret out-of-band to the end-entity that the end-entity uses to authenticate its certificate request. The end-entity signs the Full PKI Request with the private key for which the certification is being requested. The end-entity 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 end-entity's enclosed PKCS #10 or CRMF certification request message to identify the request. The end-entity 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 end-entity does not automatically authenticate its enrollment request to the CA, either because the end-entity has nothing to authenticate the request with, or because organizational policy requires RA involvement. The end-entity 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. 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. However, some organizational

policies might allow a grace period during which an expired certificate could be used to rekey.

A.2.1. 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.2.2. Rekey of Key Establishment Certificates

When a key establishment certificate is rekeyed, the Full PKI Request will generally be signed by the current private key corresponding to the current signature certificate. If there is no current signature certificate, one of the initial enrollment options in section A.1 may be used.

Authors' Addresses

Michael Peck
National Security Agency

Email: mpeck@alumni.virginia.edu

Lydia Zieglar
National Information Assurance Research Laboratory
National Security Agency

Email: llziegl@tycho.ncsc.mil

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

Email: turners@ieca.com