PKIX Working Group	J. Schaad
Internet-Draft	Soaring Hawk Consulting
Expires: June 6, 2008	M. Myers
	TraceRoute Security, Inc.
	December 04, 2007

Certificate Managmement Messages over CMS (CMC): Complience Requirements
draft-ietf-pkix-cmc-compl-05

Status of this Memo

Abstract

This document provides a set of compliance statements about the CMC (Certificate Management over CMS) enrollment protocol. The ASN.1 structures and the transport mechanisms for the CMC enrollment protocol are covered in other documents. This document provides the information needed to make a compliant version of CMC.

Table of Contents

1. Overview
2. Terminology

3. Requirements Terminology
4. Requirements for All Entities
    4.1. Cryptographic Algorithm Requirements
    4.2. Controls
    4.3. CRMF Feature Requirements
    4.4. Requirements for Clients
5. Requirements for Servers
6. Requirements for EEs
7. Requirements for RAs
8. Requirements for CAs
9. Security Considerations
10. IANA Considerations
11. Acknowledgements
12. References
    12.1. Normative References
    12.2. Informational References
§ Authors' Addresses
§ Intellectual Property and Copyright Statements

---

1. Overview

TOC

The CMC (Certificate Management over CMS) protocol is designed in terms of a client/server relationship. In the simplest case the client is the requestor of the certificate (i.e. the End Entity or EE) and the server is the issuer of the certificate (i.e. the Certificate Authority(CA)).
The introduction of an RA (registration authority) into the set of agents complicates the picture only slightly. The RA becomes the server with respect to the certificate requestor, and it becomes the client with respect to the certificate issuer. Any number of RAs can be inserted into the picture in this manner.
The RAs may serve specialized purposes that are not currently covered by this document. One such purpose would be a Key Escrow agent. As such all certificate requests for encryption keys would be directed through this RA and it would take appropriate action to do the key archival. Key recovery requests could be defined in the CMC methodology allowing for the Key Escrow agent to perform that operation acting as the final server in the chain of agents.
If there are multiple RAs in the system, it is considered normal that not all RAs will see all certificate requests. The routing between the RAs may be dependent on the content of the certificate requests involved.
This document is divided into six sections, each section specifying the requirements that are specific to a class of agents in the CMC model. These are 1) All agents, 2) all servers, 3) all clients, 4) all End Entities, 5) all Registration Entities, 6) all Certificate Authorities.

2. Terminology

TOC

There are several different terms, abbreviations and acronyms used in this document that we define here for convenience and consistency of usage:

End-Entity (EE) refers to the entity that owns a key pair and for whom a certificate is issued.

Registration Authority (RA) or Local RA (LRA) refers to an entity that acts as an intermediary between the EE and the CA. Multiple RAs can exist between the End-Entity and the Certification Authority. RAs may perform additional services such as key generation or key archival. This document uses the term RA for both RA and LRA.

Certification Authority (CA) refers to the entity that issues certificates.

Client refers to an entity that creates a PKI Request. In this document both RAs and EEs can be clients.

Server refers to the entities that process PKI Requests and create PKI Responses. In this document both CAs and RAs can be servers.

PKCS #10 refers to the Public Key Cryptography Standard #10 [PKCS10] (Nystrom, M. and B. Kaliski, “PKCS #10: Certification Request Syntax Specification v1.7,” November 2000.), which defines a certification request syntax.

CRMF refers to the Certificate Request Message Format RFC [CRMF] (Schaad, J., “Internet X.509 Certificate Request M2essage Format,” September 2005.). CMC uses this certification request syntax defined in this document as part of the protocol.

CMS refers to the Cryptographic Message Syntax RFC [CMS] (Housley, R., “Cryptographic Message Syntax (CMS),” July 2004.). This document provides for basic cryptographic services including encryption and signing with and without key management.

PKI Request/Response refers to the requests/responses described in this document. PKI Requests include certification requests, revocation requests, etc. PKI Responses include certs-only messages, failure messages, etc.

Proof-Of-Identity refers to the client proving they are who they say that are to the server.

Proof-Of-Possession (POP) refers to a value that can be used to prove that the private key corresponding to a public key is in the possession and can be used by an end-entity.

Transport wrapper refers to the outermost CMS wrapping layer.

3. Requirements Terminology

TOC

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 [MUST] (Bradner, S., “Key words for use in RFCs to Indicate Requirement Levels,” March 1997.).

4. Requirements for All Entities

TOC

All [CMC-STRUCT] (Schaad, J. and M. Myers, “Certificate Management Messages over CMS,” August 2006.) and [CMC-TRANS] (Schaad, J., Myers, M., Liu, X., and J. Weinstein, “CMC Transport,” December 2004.) compliance statements MUST be adhered to unless specifically stated otherwise in this document.
All entities MUST support Full PKI Requests, Simple PKI Responses and Full PKI Responses. Severs SHOULD support Simple PKI Requests.
All entities MUST support the use of the CRMF syntax for certification requests. Support for the PKCS#10 syntax for certification requests SHOULD be implemented by servers.
The extendedFailInfo field SHOULD NOT be populated in the CMCStatusInfoExt object; the failInfo field SHOULD be used to relay this information. If the extendedFailInfo field is used, it is suggested that an additional CMCStatusInfoExt item exist for the same body part with a failInfo field.
All entities MUST implement the HTTP transport mechanism as defined in [CMC-TRANS] (Schaad, J., Myers, M., Liu, X., and J. Weinstein, “CMC Transport,” December 2004.). Other transport mechanisms MAY be implemented.

4.1. Cryptographic Algorithm Requirements

TOC

All entities MUST verify DSA-SHA1 and RSA-SHA1 signatures in SignedData (see [CMS-ALG] (Housley, R., "Cryptographic Message Syntax (CMS) Algorithms," August 2002.)). Entities MAY be verify other signature algorithms. It is strongly suggested that RSA-PSS with SHA-1 be verified (see [CMS-RSA-PSS] (Schaad, J., "Use of the RSA PSS Signature Algorithm in CMS," December 2003.)). It is strongly suggested that SHA-256 using RSA and RSA-PSS be verified (see [RSA-256] (Schaad, J., Kaliski, B., and R. Housley, "Additional Algorithms and Identifiers for RSA Cryptography for use in the Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile," June 2005.)).

All entities MUST generate either DSA-SHA1 or RSA-SHA1 signatures for SignedData (see [CMS-ALG] (Housley, R., "Cryptographic Message Syntax (CMS) Algorithms," August 2002.)). Other signatures algorithms MAY be used for generation.

All entities MUST support AES as the content encryption algorithm for EnvelopedData (see [CMS-AES] (Schaad, J., "Use of the Advanced Encryption Standard (AES) Encryption Algorithm in Cryptographic Message Syntax (CMS)," July 2003.)). Other content encryption algorithms MAY be implemented.

All entities MUST support RSA as a key transport algorithm for EnvelopedData (see [CMS-ALG] (Housley, R., "Cryptographic Message Syntax (CMS) Algorithms," August 2002.)). All entities SHOULD support RSA-OAEP (see [CMS-RSA-OAEP] (Housley, R., "Use of the RSAES-OAEP Key Transport Algorithm in the Cryptographic Message Syntax (CMS)," July 2003.)) as a key transport algorithm. Other key transport algorithms MAY be implemented.

If an entity supports key agreement for EnvelopedData, they MUST support Diffie-Hellman (see [CMS-DH] (Rescorla, E., "Diffie-Hellman Key Agreement Method," June 1999.)).

If an entity supports PasswordRecipientInfo for EnvelopedData or AuthenticatedData, they MUST support PBKDF2 for key derivation algorithms. They MUST support AES key wrap (see [AES-WRAP] (Schaad, J. and R. Housley, "Advanced Encryption Standard (AES) Key Wrap Algorithm," September 2002.) as the key encryption algorithm.

If AuthenticatedData is supported, PasswordRecipientInfo MUST be supported.

Algorithm requirements for the Identity Proof Version 2 control (Section 6.2.1 of [CMC-STRUCT] (Schaad, J. and M. Myers, "Certificate Management Messages over CMS," August 2006.)) are: SHA-1 MUST be implemented for hashAlgId. SHA-256 SHOULD be implemented for hashAlgId. HMAC-SHA1 MUST be implemented for macAlgId. HMAC-SHA256 SHOULD be implemented for macAlgId.

Algorithm requirements for the Pop Link Witness Version 2 control (Section 5.3.1 of [CMC-STRUCT] (Schaad, J. and M. Myers, "Certificate Management Messages over CMS," August 2006.)) are: SHA-1 MUST be

implemented for keyGenAlgorithm. SHA-256 SHOUL be implemented for keyGenAlgorithm. PBKDF2 MAY be implemented for keyGenAlgorithm. HMAC-SHA1 MUST be implemented for macAlgorithm. HMAC-SHA256 SHOULD be implemented for macAlgorithm.
Algorithm requirements for the Encrypted POP and Decrypted POP controls (Section 6.7 of [CMC-STRUCT] (Schaad, J. and M. Myers, “Certificate Management Messages over CMS,” August 2006.)) are: SHA-1 MUST be implemented for witnessAlgID. SHA-256 SHULD be implemented for witnessAlgID. HMAC-SHA1 MUST be implemented for thePOPAlgID. HMAC-SHA256 SHOULD be implemented for thePOPAlgID.
Algorithm requirements for Publish Trust Anchors control (Section 6.15 of [CMC-STRUCT] (Schaad, J. and M. Myers, “Certificate Management Messages over CMS,” August 2006.)) are: SHA-1 MUST be implemented for hashAlgorithm. SHA-256 SHOULD be implemented for hashAlgorithm.
If an EE generates DH keys for certification, it MUST support section 4 of [DH-POP] (Prafullchandra, H. and J. Schaad, “Diffie-Hellman Proof-of-Possession Algorithms,” June 2000.). EEs MAY support section 3 of [DH-POP] (Prafullchandra, H. and J. Schaad, “Diffie-Hellman Proof-of-Possession Algorithms,” June 2000.). CAs and RAs that do POP verification MUST support section 4 of [DH-POP] (Prafullchandra, H. and J. Schaad, “Diffie-Hellman Proof-of-Possession Algorithms,” June 2000.) and SHOULD support section 3 of [DH-POP] (Prafullchandra, H. and J. Schaad, “Diffie-Hellman Proof-of-Possession Algorithms,” June 2000.).
EEs that need to use a signature algorithm for keys that cannot produce a signature MUST support Appendix C of [CMC-STRUCT] (Schaad, J. and M. Myers, “Certificate Management Messages over CMS,” August 2006.) and MUST support the Encrypted/Decrypted POP controls. CAs and RAs that do POP verification MUST support this signature algorithm and MUST support the Encrypted/Decrypted POP controls.

4.2. Controls

TOC

The following table lists the name and level of support required for each control.

Control	EE	RA	CA
Extended CMC Status Info	MUST	MUST	MUST
CMC Status Info	SHOULD	SHOULD	SHOULD
Identity Proof Version 2	MUST	MUST	MUST
Identity Proof	SHOULD	SHOULD	SHOULD
Identification	MUST	MUST	MUST
POP Link Random	MUST	MUST	MUST

POP Link Witness Version 2	MUST	MUST	MUST
POP Link Witness	SHOULD	MUST	MUST
Data Return	MUST	MUST	MUST
Modify Cert Request	N/A	MUST	(2)
Add Extensions	N/A	MAY	(1)
Transaction ID	MUST	MUST	MUST
Sender Nonce	MUST	MUST	MUST
Recipient Nonce	MUST	MUST	MUST
Encrypted POP	(4)	(5)	SHOULD
Decrypted POP	(4)	(5)	SHOULD
RA POP Witness	N/A	SHOULD	(1)
Get Certificate	optional	optional	optional
Get CRL	optional	optional	optional
Revocation Request	SHOULD	SHOULD	MUST
Registration Info	SHOULD	SHOULD	SHOULD
Response Information	SHOULD	SHOULD	SHOULD
Query Pending	MUST	MUST	MUST
Confirm Cert. Acceptance	MUST	MUST	MUST
Publish Trust Anchors	(3)	(3)	(3)
Authenticate Data	(3)	(3)	(3)
Batch Request	N/A	MUST	(2)
Batch Responses	N/A	MUST	(2)
Publication Information	optional	optional	optional
Control Processed	N/A	MUST	(2)

Table 1: CMC Control Attributes

Notes:

1. CAs SHOULD implement this control if designed to work with RAs.

2. CAs MUST implement this control if designed to work with RAs.

3. Implementation is optional for these controls. We strongly suggest that they be implemented in order to populate client trust anchors.

4. EEs only need to implement this if (a) they support key agreement algorithms or (b) they need to operate in environments where the hardware keys cannot provide POP.

5. RAs SHOULD implement this if they implement RA POP Witness.

Strong consideration should be given to implementing the Authenticate Data and Publish Trust Anchors controls as this gives a simple method for distributing trust anchors into clients without user intervention.

4.3. CRMF Feature Requirements

TOC

The following additional restrictions are placed on CRMF features:
The registration control tokens id-regCtrl-regToken and id-regCtrl-authToken MUST NOT be used. No specific CMC feature is used to replace these items, but generally the CMC controls identification and identityProof will perform the same service and are more specifically defined.
The control token id-regCtrl-pkiArchiveOptions SHOULD NOT be supported. An alternative method is under development to provide this functionality.
The behavior of id-regCtrl-oldCertID is not presently used. It is replaced by issuing the new certificate and using the id-cmc-publishCert to remove the old certificate from publication. This operation would not normally be accompanied by an immediate revocation of the old certificate, however that can be accomplished by the id-cmc-revokeRequest control.
The id-regCtrl-protocolEncrKey is not used.

4.4. Requirements for Clients

TOC

No additional requirements.

---

5. Requirements for Servers

TOC

No additional requirements.

---

6. Requirements for EEs

TOC

If an entity implements Diffie-Hellman, it MUST implement either the DH-POP Proof-of-Possession as defined in [DH-POP] (Prafullchandra, H. and J. Schaad, “Diffie-Hellman Proof-of-Possession Algorithms,” June 2000.) Section 4 or the challenge-response POP controls id-cmc-encryptedPOP and id-cmc-decryptedPOP.

---

7. Requirements for RAs

TOC

RAs SHOULD be able to do delegated POP. RAs implementing this feature MUST implement the id-cmc-lraPOPWitness control.
All RAs MUST implement the promotion of the id-aa-cmc-unsignedData as covered in section 3.8 of [CMC-STRUCT] (Schaad, J. and M. Myers, “Certificate Management Messages over CMS,” August 2006.)

---

8. Requirements for CAs

TOC

Providing for CAs to work in an environment with RAs is strongly suggested. Implementation of such support is strongly suggested as this permits the delegation of substantial administrative interaction onto an RA rather than at the CA.
CAs MUST perform at least minimal checks on all public keys before issuing a certificate. At a minimum a check for syntax would occur with the POP operation. Additionally CAs SHOULD perform simple checks for known bad keys such as small subgroups for DSA-SHA1 and DH keys [SMALL-SUB-GROUP] (Zuccherato, R., “Methods for Avoiding the "Small-Subgroup" Attacks on the Diffie-Hellman Key Agreement Method for S/MIME,” March 2000.) or known bad exponents for RSA keys.
CAs MUST enforce POP checking before issuing any certificate. CAs MAY delegate the POP operation to an RA for those cases where 1) a challenge/response message pair must be used, 2) an RA performs escrow of a key and checks for POP in that manner or 3) an unusual algorithm is used and that validation is done at the RA.
CAs SHOULD implement both the DH-POP Proof-of-Possession as defined in [DH-POP] (Prafullchandra, H. and J. Schaad, “Diffie-Hellman Proof-of-Possession Algorithms,” June 2000.) Section 4 and the challenge-response POP controls id-cmc-encryptedPOP and id-cmc-decryptedPOP.

---

9. Security Considerations

TOC

This document uses [CMC-STRUCT] (Schaad, J. and M. Myers, “Certificate Management Messages over CMS,” August 2006.) and [CMC-TRANS] (Schaad, J., Myers, M., Liu, X., and J. Weinstein, “CMC Transport,” December 2004.) as building blocks to this document. The security sections of those two documents are included by reference.

Knowledge of how an entity is expected to operate is vital in determining which sections of requirements are applicable to that entity. Care needs to be taken in determining which sections apply and fully implementing the necessary code.

Cryptographic algorithms have and will be broken or weakened. Implementers and users need to check that the cryptographic algorithms listed in this document make sense from a security level. The IETF from time to time may issue documents dealing with the current state of the art. Two examples of such documents are [SMALL-SUB-GROUP] (Zuccherato, R., “Methods for Avoiding the "Small-Subgroup" Attacks on the Diffie-Hellman Key Agreement Method for S/MIME,” March 2000.) and [HASH-ATTACKS] (Hoffman, P. and B. Schneier, “Attacks on Cryptographic Hashes in Internet Protocols,” November 2005.).

10. IANA Considerations

TOC

There are no IANA considerations in this document.

11. Acknowledgements

TOC

The authors and the Working Group are grateful for the participation of Xiaoui Lui and Jeff Weinstein in helping to author the original versions of this document.

The authors would like to thank Brian LaMacchia for his work in developing and writing up many of the concepts presented in this document. The authors would also like to thank Alex Deacon and Barb Fox for their contributions.

12. References

TOC

12.1. Normative References

TOC

[CMC-STRUCT]	Schaad, J. and M. Myers, “Certificate Management Messages over CMS,” draft-ietf-pkix-2797-bis-05.txt , August 2006.
[CMC-TRANS]	Schaad, J., Myers, M., Liu, X., and J. Weinstein, “CMC Transport,” Work In Progress , December 2004.
[CMS]	

	Housley, R., “Cryptographic Message Syntax (CMS),” RFC 3852, July 2004.
[CMS-AES]	Schaad, J., “Use of the Advanced Encryption Standard (AES) Encryption Algorithm in Cryptographic Message Syntax (CMS),” RFC 3565, July 2003.
[CMS-ALG]	Housley, R., “Cryptographic Message Syntax (CMS) Algorithms,” RFC 3370, August 2002.
[CMS-DH]	Rescorla, E., “Diffie-Hellman Key Agreement Method,” RFC 2631, June 1999.
[CRMF]	Schaad, J., “Internet X.509 Certificate Request M2essage Format,” RFC 4211, September 2005.
[CMS-RSA-OAEP]	Housley, R., “Use of the RSAES-OAEP Key Transport Algorithm in the Cryptographic Message Syntax (CMS),” RFC 3560, July 2003.
[CMS-RSA-PSS]	Schaad, J., “Use of the RSA PSS Signature Algorithm in CMS,” Work In Progress , December 2003.
[DH-POP]	Prafullchandra, H. and J. Schaad, “Diffie-Hellman Proof-of-Possession Algorithms,” RFC 2875, June 2000.
[MUST]	Bradner, S., “Key words for use in RFCs to Indicate Requirement Levels,” RFC 2119, BCP 14, March 1997.
[RSA-256]	Schaad, J., Kaliski, B., and R. Housley, “Additional Algorithms and Identifiers for RSA Cryptography for use in the Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile,” RFC 4055, June 2005.
[AES-WRAP]	Schaad, J. and R. Housley, “Advanced Encryption Standard (AES) Key Wrap Algorithm,” RFC 3394, September 2002.

12.2. Informational References

TOC

[PKCS10]	Nystrom, M. and B. Kaliski, “PKCS #10: Certification Request Syntax Specification v1.7,” RFC 2986, November 2000.
[SMALL-SUB-GROUP]	Zuccherato, R., “Methods for Avoiding the "Small-Subgroup" Attacks on the Diffie-Hellman Key Agreement Method for S/MIME,” RFC 2785, March 2000.
[HASH-ATTACKS]	Hoffman, P. and B. Schneier, “Attacks on Cryptographic Hashes in Internet Protocols,” RFC 4270, November 2005.

Authors' Addresses

TOC

	Jim Schaad
	Soaring Hawk Consulting
	PO Box 675

	Gold Bar, WA 98251
Phone:	(425) 785-1031
Email:	jimsch@nwlink.com
	Michael Myers
	TraceRoute Security, Inc.
Email:	mmyers@fastq.com

Full Copyright Statement

TOC

Intellectual Property