PKI Enrollment Information

Status of this Memo

This document is an Internet-Draft and is subject to all provisions of Section 10 of RFC2026. 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/lid-abstracts.html

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

Comments or suggestions for improvement may be made on the "ietf-sacred" mailing list, or directly to the author.

Copyright Notice

Copyright (C) The Internet Society (1998). All Rights Reserved.

Abstract

This document describes the format of PKI enrollment information, which may be used by an RA/CA to enable automated end entity certification. The PKI enrollment information contains PKI parameters describing RA/CA certification policy requirements put on end entities during their enrollment for a public key certificate.

1. End Entity PKI Enrollment

End entities represent all users of PKI (individuals or software processes), who are issued public key certificates. The process of issuing a certificate for an end entity is known as end-entity PKI enrollment. Certificates are issued by a Certification Authority (CA), with or without the interaction of a Registration Authority (RA) [RFC2510].

An end entity PKI enrollment can be initiated at the RA/CA or at the end entity (EE).
When PKI enrollment is initiated at the RA or CA, the end entity eventually gets a token containing a private key and the corresponding certificate as issued by the CA. After receiving the token, the EE is capable of performing authenticated signing operations (e.g., S/MIME or SSL) without any further interaction with the RA/CA. The drawback of this solution is that non-repudiation of signatures cannot be guaranteed since the keys are generated by the RA/CA.

Alternatively, PKI enrollment can be initiated by the end entity and the enrollment process can be performed with or without the human interaction. An example of an end entity initiated PKI enrollment is a browser-based enrollment with the RA/CA providing HTML forms, requiring human interaction to populate the fields and complete the enrollment.

This draft proposes a solution for performing automated end entity PKI enrollment (without requiring human intervention). The automated end entity PKI enrollment is based on the assumption that the RA/CA configuration information is made available to the end entity prior to the initiation of the enrollment, and that the end entity is in possession of client software capable of interpreting the configuration information.

With this solution keys are being generated by the end entity, and non-repudiation is thus provided for. When there is a need for using multiple keys for different key usage (dual keys), with RA/CA generation and optionally backup of encipherment keys, then a mixed model can be used. The RA/CA is creating a token for an end entity with encipherment private key and corresponding certificate, and PKI enrollment information. After receiving the token, the end entity is able to initiate automated PKI enrollment for signature keys.

The PKI enrollment information can alternatively be used by the RA/CA to make its PKI parameters publicly available to end entities (and RAs) wishing to communicate with the RA/CA. This communication can involve more than just the initial PKI enrollment e.g., key update, certificate update, certificate revocation, etc.

Depending on the type of PKI enrollment information, it can contain only general RA/CA configuration information and be made publicly available to everyone, or it can be tailored with each end entity’s individual data and thus required to be available only for the particular end entity. In the former case, the PKI enrollment data may be placed at the RA’s or CA’s web page, and in the latter case it should be distributed to the EE in a secure way.

The PKI enrollment information may be stored on the end entity’s token in the Personal Security Environment (PSE) and handed to the end entity prior to PKI enrollment. PKCS#15 token format could be extended to define a placeholder for the PKI enrollment information [PKCS#15].

Kapidzic-Cicovic
INTERNET DRAFT PKI Enrollment Information June 2001

The format of the end entity token and its location after PKI enrollment (e.g., at a credential server) is not covered by this draft document.
2. PKI Enrollment Information

A PKI enrollment structure is composed of general RA/CA parameters and optionally individual end entity parameters:

PKI_EnrollmentInfo ::= SEQUENCE {
  generalParameters PKI_GeneralParameters,
  eeIndividualParameters PKI_EEIndividualParameters OPTIONAL
}

The fully automated end entity initiated PKI enrollment can be achieved only if the information required for authentication of the end entity is contained in the PKI enrollment information (eeIndividualParameters). Otherwise, some human intervention is still required during the enrollment in order to achieve authentication.

2.1 PKI Enrollment Information - General Parameters

PKI enrollment information general parameters structure contains a version number and several optional RA/CA parameters. Their detailed description is given in sections 2.1.1 to 2.1.7.

PKI_GeneralParameters ::= SEQUENCE {
  Version             [0]   EXPLICIT Version DEFAULT v1,
  cAResponderAddress  [1]   PKI_Entity OPTIONAL,
  rAResponderAddress  [2]   PKI_Entity OPTIONAL,
    -- either rA or parentCA should be present
  pkiProtocol         [3]   SEQUENCE OF -- all supported by RA/CA
    PKI_Protocol OPTIONAL,
  popMethod           [4]   SEQUENCE OF -- all supported by RA/CA
    PKI_POP OPTIONAL,
  certTemplate        [5]   CertTemplate OPTIONAL,
    -- the presence of a part of CertTemplate means that the
    -- RA/CA accepts this part to be populated by the EE
    -- (the actual value is not important)
  keyGenerators       [6]   SEQUENCE OF -- as required by RA/CA
    KeyGenerator OPTIONAL,
  minKeySize          [7]   SEQUENCE OF -- as required by RA/CA
    KeySizeRequirement OPTIONAL,
  maxKeySize          [8]   SEQUENCE OF -- as required by RA/CA
    KeySizeRequirement OPTIONAL,
  keyTypes            [9]   SEQUENCE OF -- all supported by RA/CA
    KeyTypeRequirement OPTIONAL
}

2.1.1 Responder Address

The responder address parameter specifies the address of the RA/CA responder where certification requests and other PKI messages are to be sent for processing by the RA/CA. The address is to be provided as a uniformResourceIdentifier choice of GeneralName [RFC2459], thus specifying the transport protocol as well as the address of the responder (e.g. "http://www.myCA.com/responder");
PKI_Entity ::= SEQUENCE {
  name        [0] Name  OPTIONAL,  -- EE, RA or CA DN
  altNames    [1] GeneralNames  OPTIONAL -- e.g., e-mail for EE,
                  -- URL or IP of the RA/CA
} -- either one of name and altNames or both must be present

GeneralNames ::= SEQUENCE SIZE (1..MAX) OF GeneralName

GeneralName ::= CHOICE {
  otherName                       [0]     OtherName,
  rfc822Name                      [1]     IA5String,
  dNSName                         [2]     IA5String,
  x400Address                     [3]     ORAddress,
  directoryName                   [4]     Name,
  ediPartyName                    [5]     EDIPartyName,
  uniformResourceIdentifier       [6]     IA5String,
  iPAddress                       [7]     OCTET STRING,
  registeredID                    [8]     OBJECT IDENTIFIER
}

2.1.2 PKI Protocol

There are a number of existing standards for PKI enrollment. The
most usually used ones are PKCS #10 certification requests [PKCS10]
with PKCS #7 certification responses [PKCS7], PKIX CMP [RFC2510] and
PKIX CMC [RFC2797]. The RA/CA may use this parameter to express its
preferred PKI protocol for end entities to use when enrolling for
certificates.

PKI_Protocol ::= ENUMERATED {
  PKIX_CMPv1      (0),   -- RFC 2510
  PKIX_CMPv2      (1),   -- 2510 bis
  PKCS10_PKCS7    (2),
  PKIX_CMC        (3)    -- RFC 2797
}

2.1.3 POP Method

There are different ways for an end entity to supply proof of
possession of the private key corresponding to the public key
contained in the certification request. This parameter specifies the
private key proof-of-possession method as required by the CA
[RFC2511].

PKI_POP ::= ENUMERATED {
  ImplicitPOP     (0),
  ExplicitPOP     (1)
}

2.1.4 Certificate Template

The CA certification policy dictates which fields of the end entity
certificate are to be populated. For some certificate fields it is
the end entity that should provide the value in the certification
request (e.g., key usage, subject alternative name, etc.).

The certificate template parameter provides a way for the CA to
specify which fields in the certificate request template are required to be populated by the end entity.

For example, the certificate template parameter may contain the following fields:
- Subject DN (to be present in the request or not),
- Subject alternative names (e.g., e-mail),
- Key usage,
- Extended key usage,
- Other extensions (e.g., private).

The encoding of the certificate template parameter is as CertTemplate specified in [RFC2511].

The presence of a field in the certificate template means that its value is to be provided by an end entity in the certification request. The actual value that is encoded in the certificate template of the PKI enrollment info is to be ignored.

2.1.5 Key Generators

The CA certification policy may require certain type of keys to be generated at the RA or CA, e.g., encipherment keys for backup purposes. Some other types of keys should preferably be generated by the end-entity, e.g., signature keys in order to provide for non-repudiation. Alternatively encipherment keys may be generated at the end entity side and sent to the RA/CA for backup purposes.

An RA/CA may use the parameter Key Generators in order to provide this information about its certification policy to end entities.

PKI_EntityType ::= ENUMERATED {
  EE     (0),
  RA     (1),
  CA     (2)
}

KeyGenerator ::= SEQUENCE {
  keyUsage        KeyUsage,     -- only one value at a time
  entityType      PKI_EntityType,  -- RA, CA or EE
  backupRequired  BOOLEAN DEFAULT FALSE
  -- if TRUE and keys generated by EE,
  -- the private key MUST be sent
}

2.1.6 Supported Key Sizes

The RA/CA may specify the minimal key size that is acceptable for a specific key usage according to the CA certification policy. Furthermore, the RA/CA may have an upper limit on the length of the key it is capable of handling. The KeySizeRequirement structure may be used to provide this information to end entities.

KeySizeRequirement ::= SEQUENCE {
  keyUsage        KeyUsage,     -- only one value at a time

2.1.7 Supported Key Types

The RA/CA may be capable of handling only certain key algorithm types for a specific key usage, e.g. only DSA keys for digital signature and only Diffie-Hellman keys for key agreement (alternatively, only RSA key type for all key usages). The KeyTypeRequirement structure may be used to provide this type of information to end entities.

```plaintext
KeyTypeRequirement ::= SEQUENCE {
  keyUsage        KeyUsage,           -- only one value at a time
  keyType         OBJECT IDENTIFIER   -- algorithm identifier
}
```

2.2 PKI Enrollment Information - End Entity Individual Parameters

The individual end entity parameters are to be encoded in PKI_EEIndividualParameters structure.

```plaintext
PKI_EEIndividualParameters ::= SEQUENCE {
  endEntityNamingInfo      PKI_Entity,
  sharedSecret             BIT STRING   OPTIONAL
}
```

The RA/CA policy may be such that it requires end entities to populate certification requests with the same DN and alternative names as registered at the RA/CA. If this is the case, end entity naming info parameter may be used to provide this information to each end entity in the RA/CA's domain.

In order to provide authenticity of the initial EE certification requests, the RA/CA usually share a secret with each end entity. The shared secret parameter may be used to communicate this information to end entities.

3. Security Considerations

The PKI Enrollment Information needs to be provided to end entities in secure way only if it contains end entity individual data required for authentication of the end entity during certification. In case that these parameters are not included in the PKI Enrollment Information, no extra security measures need to be taken.

4. References

[RFC2459] Housley, R., Ford, W., Polk, W. and D. Solo "Internet


5. Author's Address

Nada Kapidzic Cicovic
Entegrity Solutions
164 74 Kista
Sweden

Phone: +46 8 477 77 37
EMail: nada@entegrity.com

Appendix A: ASN.1 Module using 1988 Syntax

PKIEnr??? (iso(1) ... ???)

DEFINITIONS EXPLICIT TAGS ::= 

BEGIN

-- EXPORTS ALL --

IMPORTS

Version, KeyUsage, Name, GeneralNames
FROM PKIX1Explicit88 (iso(1) identified-organization(3)

Kapidzic-Cicovic
INTERNET DRAFT PKI Enrollment Information June 2001

dod(6) internet(1) security(5) mechanisms(5) pkix(7)
id-mod(0) id-pkix1-explicit-88(1}}

CertTemplate
FROM PKIXCRMF {iso(1) identified-organization(3)
dod(6) internet(1) security(5) mechanisms(5) pkix(7)
id-mod(0) id-mod-crmf(5)}

-- Locally defined OIDs --

PKI_Entity ::= SEQUENCE {
    name        [0] Name  OPTIONAL, -- EE, RA or CA DN
    altNames    [1] GeneralNames  OPTIONAL -- e.g., e-mail for EE,
               -- URL or IP of the RA/CA
} -- either one of name and altNames or both must be present

PKI_EntityType ::= ENUMERATED {
    EE     (0),
    RA     (1),
    ...}
PKI_Protocol ::= ENUMERATED {
    PKIX_CMPv1      (0),   -- RFC 2510
    PKIX_CMPv2      (1),   -- 2510 bis
    PKCS10_PKCS7    (2),
    PKIX_CMC        (3)
}

PKI_POP ::= ENUMERATED {
    ImplicitPOP     (0),
    ExplicitPOP     (1)
}

KeyGenerator ::= SEQUENCE {
    keyUsage        KeyUsage,     -- only one value at a time
    entityType      PKI_EntityType,  -- RA, CA or EE
    backupRequired  BOOLEAN DEFAULT FALSE
        -- if TRUE and keys generated by EE,
        -- the private key MUST be sent
        -- to RA/CA for backup
}

KeySizeRequirement ::= SEQUENCE {
    keyUsage        KeyUsage,     -- only one value at a time
    size            INTEGER       -- bit size of the key
}

KeyTypeRequirement ::= SEQUENCE {
    keyUsage        KeyUsage,           -- only one value at a time
    keyType         OBJECT IDENTIFIER   -- algorithm identifier
}

PKI_GeneralParameters ::= SEQUENCE {
    Version             [0]   EXPLICIT Version DEFAULT v1,
    cAResponderAddress  [1]   PKI_Entity          OPTIONAL,
    rAResponderAddress  [2]   PKI_Entity          OPTIONAL,
        -- either rA or parentCA should be present
    pkiProtocol         [3]   SEQUENCE OF  -- all supported by RA/CA
        PKI_Protocol        OPTIONAL,
    popMethod           [4]   SEQUENCE OF  -- all supported by RA/CA
        PKI_POP             OPTIONAL,
    certTemplate        [5]   CertTemplate        OPTIONAL,
        -- the presence of a part of CertTemplate means that the
        -- RA/CA accepts this part to be populated by the EE
        -- (the actual value is not important)
    keyGenerators       [6]   SEQUENCE OF  -- as required by RA/CA
        KeyGenerator        OPTIONAL,
    minKeySize          [7]   SEQUENCE OF  -- as required by RA/CA
        KeySizeRequirement OPTIONAL,
    maxKeySize          [8]   SEQUENCE OF  -- as required by RA/CA
        KeySizeRequirement OPTIONAL,
    keyTypes            [9]   SEQUENCE OF  -- all supported by RA/CA
        KeyTypeRequirement OPTIONAL
PKI_EEIndividualParameters ::= SEQUENCE {
    endEntityNamingInfo     PKI_Entity,
    sharedSecret             BIT STRING  OPTIONAL
}

PKI_EnrollmentInfo ::= SEQUENCE {
    generalParameters         PKI_GeneralParameters,
    eeIndividualParameters    PKI_EEIndividualParameters  OPTIONAL
}

Kapidzic-Cicovic  9