PKI4IPSEC Working Group
Internet Draft   Chris Bonatti, IECA
draft-ietf-pki4ipsec-mgmt-profile-rqts-07.txt   Sean Turner, IECA
November 8, 2006   Gregory Lebovitz, Juniper
Expires May 8, 2007

Requirements for an IPsec Certificate Management Profile

Status of this Memo

Copyright Notice

Abstract

   This informational document describes and identifies the requirements
   for transactions to handle Public Key Certificate (PKC) lifecycle
   transactions between Internet Protocol Security (IPsec) Virtual
   Private Network (VPN) Systems using Internet Key Exchange (IKE)
   (versions 1 and 2) and Public Key Infrastructure (PKI) Systems. These
   requirements are designed to meet the needs of enterprise scale IPsec
   VPN deployments. It is intended that a standards track profile of a
   management protocol will be created to address many of these
   requirements.

Table of Contents

1   Introduction

This document contains requirements for a transaction-based
approach. Other models are conceivable, for example a directory-
centric approach, but their requirements are beyond the scope of
this document.

This document enumerates requirements for Public Key Certificate
(PKC) lifecycle transactions between different VPN System and PKI
System products in order to better enable large scale, PKI-enabled

IPsec deployments with a common set of transactions. Requirements for

both the IPsec and the PKI products are discussed. The requirements
are carefully designed to achieve security without compromising ease
of management and deployment, even where the deployment involves tens
of thousands of IPsec users and devices.

The requirements address transactions for the entire PKC lifecycle
for PKI-enabled VPN System: authorization (of PKC issuance),
generation (public-private key pair and PKC request), enrollment (PKC
request, PKC response, and confirmation), maintenance (rekey, renew,
update, revoke, and confirm), and repository lookups. These
transactions enable a VPN Operator to:

  - Use a VPN Administration function (Admin), which is introduced in
    this document, to manage PKC authorization and possibly act as
    the sole interface for the VPN System and the PKI System.

  - Authorize individual or batches of PKC issuances based on a pre-
    agreed template (i.e., both types of authorization requests
    refer to the pre-agreed template). These authorizations can
    occur either prior to the enrollment or in the same transaction
    as the enrollment.

  - Provision PKI-based user or machine identity to IPsec Peers, on a
    large scale.

  - Set the corresponding gateway or client authorization policy for
    remote access and site-to-site connections.

  - Establish policies for automatic PKC renewal, updates, or rekeys.

  - Ensure timely revocation information is available for PKCs used
    in IKE exchanges.

These requirements are intended to be used to profile a certificate
management protocol that the VPN System will use to communicate with
the PKI System. Note that this profile will be in another document.
The certificate management profile will also clarify and constrain
existing PKIX and IPsec standards to limit the complexity of
deployment. Some requirements may require either a new protocol, or
changes or extensions to an existing protocol.

The desired outcome of the requirements and profile documents is that
both IPsec and PKI vendors create interoperable products to enable
large-scale IPsec System deployments, and do so as quickly as
possible. For example, a VPN Operator should be able to use any
conforming IPsec implementation (VPN Admin or IPsec Peer) of the
certificate management profile with any conforming PKI vendor s
implementation to perform the VPN rollout and management.

1.1 Scope

The document addresses requirements on transactions between the VPN
Systems and the PKI Systems and between the VPN Administration and
IPsec Peers. The requirements strive to meet eighty percent of the
market needs for large-scale deployments (i.e., VPNs including
hundreds or thousands of managed VPN gateways or VPN remote access
clients). Environments will understandably exist in which large-scale
deployment tools are desired, but local security policy stringency
will not allow for the use of such commercial tools. The solution
will possibly miss the needs of the highest ten percent of stringency
and lowest ten percent of convenience requirements. Use cases will be
considered or rejected based upon this eighty percent rule. The needs
of small deployments are a stated non-goal, however service providers
employing the scoped solution and applying it to many smaller
deployments in aggregate may address them.

Gateway-to-gateway access and end-user remote access (to a gateway)
are both covered. End-to-end communications are not necessarily
excluded but are intentionally not a focus.

Only VPN-PKI transactions that ease and enable scalable PKI-enabled
IPsec deployments are addressed.

1.2 Non-Goals

The scenario for PKC cross-certification will not be addressed.

The protocol specification for the VPN-PKI interactions will not be
addressed.

The protocol specification for the VPN Administrator to Peer
transactions will not be addressed. These interactions are considered
vendor proprietary. These interactions may be standardized later to
enable interoperability between VPN Administration function stations
and IPsec Peers from different vendors, but are far beyond the scope
of this current effort, and will be described as opaque transactions
in this document.

The protocol specification for RA-CA, CA-Repository, and RA-
Repository interactions will not be addressed.

1.3 Definitions

VPN System
The VPN System is comprised of the VPN Administration function
(defined below), the IPsec Peers, and the communication mechanism
between the VPN Administration and the IPsec Peers. VPN System is
defined in more detail in section 2.1.

PKI System
The PKI System, or simply PKI, is the set of functions needed to
authorize, issue, and manage PKCs. PKI System is defined in more
detail in section 2.2.

(VPN) Operator
The Operator is the person or group of people that define security
policy and configure the VPN System to enforce that policy, with the
VPN Administration function.

IPsec Peer (Gateway or Client)
For the purposes of this document, an IPsec Peer, or simply "Peer",
is any VPN System component that communicates IKE and IPsec to
another Peer in order to create an IPsec Security Association for
communications. It can be either a traditional security gateway (with
two network interfaces, one for the protected network and one for the
unprotected network), or it can be an IPsec client (with a single
network interface). In both cases, the Peer can pass traffic with no
IPsec protection, and can add IPsec protection to chosen traffic
streams. See Section 2.1.1 for more details.

(VPN) Admin
The Admin is the VPN System function that interacts with the PKI
System to establish PKC provisioning for the VPN connections. See
Section 2.1.2 for more details.

End Entity
An end entity is the entity or subject that is identified in a PKC.
The end entity is the one entity that will finally use a private key
associated with a PKC to digitally sign data. In this document, an
IPsec Peer is certainly an end entity, but the VPN Admin can also
constitute an end entity. Note that end entities can have different
PKCs for different purposes (e.g., signature vs. key exchange, Admin-
functions vs. Peer-functions).

PKC Renewal
The acquisition of a new PKC with the same public key due to the
expiration of an existing PKC. Renewal occurs prior to the expiration
of the existing PKC to avoid any connection outages. A renewal
process can rely on the existing key pair to bootstrap authentication

for the new enrollment.

PKC Update
A special case of a renewal-like occurrence where a PKC needs to be
changed prior to expiration due to some change in its subject s
information. Examples might include change in the address, telephone
number, or name change due to marriage of the end entity. An update
process can rely on the existing key pair to bootstrap authentication
for the new enrollment.

PKC Rekey
The routine procedure for replacement of a PKC with a new PKC with a
new public key for the same subject name. A rekey process can rely
on the existing key pair to bootstrap authentication for the new
enrollment.

Registration Authority (RA)
An optional entity in a PKI System given responsibility for
performing some of the administrative tasks necessary in the
registration of end entities, such as confirming the subject s
identity and verifying that the subject has possession of the private
key associated with the public key requested for a PKC.

Certificate Authority (CA)
An authority in a PKI System that is trusted by one or more users to
create and sign PKCs. It is important to note that the CA is
responsible for the PKCs during their whole lifetime, not just for
issuing them.

Repository
An Internet-accessible server in a PKI System that stores and makes
available for retrieval PKCs and Certificate Revocation Lists (CRLs).

Root CA/Trust Anchor
A CA that is directly trusted by an end entity; that is, securely
acquiring the value of a Root CA public key requires some out-of-band
step(s). This term is not meant to imply that a Root CA is
necessarily at the top of any hierarchy, simply that the CA in
question is trusted directly.

Certificate Revocation List (CRL)
A CRL is a CA-signed, time stamped list identifying revoked PKCs and
made freely available in a repository. Peers retrieve the CRL to
verify that a PKC being presented to them as the identity in an IKE
transaction has not been revoked.

CRL Distribution Point (CDP)
The CDP is a PKC extension that identifies the location from which
end entities should retrieve CRLs to check status information.

   Authority Info Access (AIA)
   The AIA is a PKC extension that indicates how to access CA
   information and services for the issuer of the PKC in which the
   extension appears. Information and services may include on-line
   validation services and Certificate Policy (CP) data.

1.4 Requirements 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 [MUSTSHOULD].

2 Architecture

   This section describes the overall architecture for a PKI-supported
   IPsec VPN deployment. First, an explanation of the VPN System is
   presented. Second, key points about the PKI System are stated. Third,
   the VPN-PKI architecture is presented.

2.1 VPN System

   The VPN System consists of the IPsec Peers and the VPN Administration
   function, as depicted in Figure 1.

```
       +-----------------------------------------------------+
       |                                                     |
       |                   +----------+                      |
       |                   |   VPN    |                      |
       |      +---------->|   Admin   |<-------+             |
       |      |           | Function  |        |             |
       |      |           +----------+         |             |
       |      v                                v             |
       |  +---------+                     +---------+        |
       |  |  IPsec  |                     |  IPsec  |        |
       |  | Peer 1  |<====================>| Peer 2 |        |
       |  +---------+                     +---------+        |
       |                                                     |
       |                 VPN System                          |
       +-----------------------------------------------------+
```
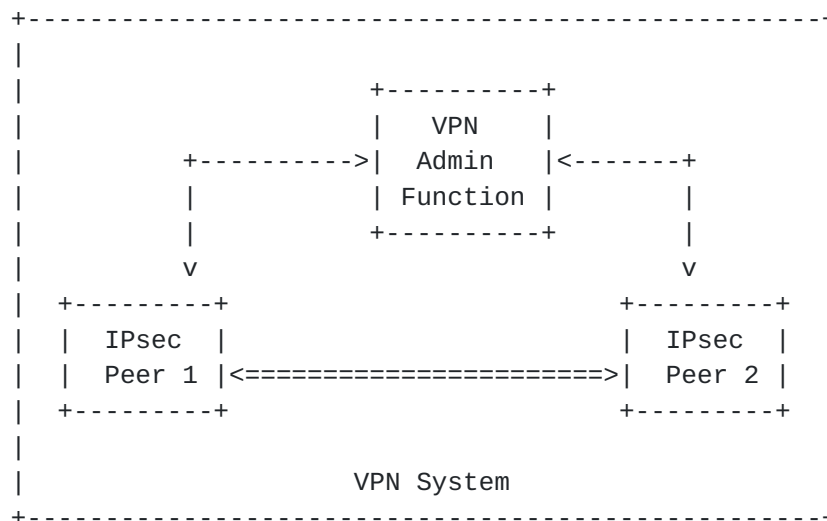
                        Figure 1: VPN System

2.1.1 IPsec Peer(s)

The Peers are two entities between which establishment of an IPsec
Security Association is required. Two Peers are shown in Figure 1,
but implementations can support an actual number in the hundreds or
thousands. The Peers can be gateway-to-gateway, remote-access-host-
to-gateway, or a mix of both. The Peers authenticate themselves in
the IKE negotiation using digital signatures generated with PKCs for
a PKI System.

2.1.2 VPN Administration Function (Admin)

This document defines the notion of a VPN Administration function,
hereafter referred to as Admin, and gives the Admin great
responsibility within the VPN System. The Admin is a centralized
function used by the Operator to interact with the PKI system to
establish PKI policy (e.g., algorithms, key lengths, lifecycle
options, and PKC fields) for groups of IPsec Peers. The Admin also
authorizes PKC issuance and it can act as the Peer's PKI System
interface, which allows the Admin to perform many RA-like functions.

It is important to note that, within this document, the Admin is
neither a device nor a person; rather it is a function. Every large-
scale VPN deployment will contain the Admin function. The function
can be performed on a stand-alone workstation, on a gateway, or on an
administration software component. The Admin function can also be one
and the same as the gateway or client device or software. They are
represented in the architectural diagram as different functions, but
they need not be different physical entities. As such, the Admin s
architecture and the means by which it interacts with the
participating IPsec Peers will vary widely from implementation to
implementation. However, some basic functions of the Admin are
assumed.

   - It and not the PKI will define the Certificate Policy (CP)
     [FRAME] for use in a VPN System. The PKC's characteristics and
     contents are a function of the CP. In VPN Systems, the Operator
     chooses to strengthen the VPN by using PKI; PKI is a bolt-on to
     the VPN System. The Operator will configure local security
     policy in part through the Admin and its authorized PKI-enabled
     Peers.

   - It will interact directly with the PKI System to initiate
     authorization for end entity PKCs by sending the parameters and
     contents for individual PKCs or batches of PKCs based on a pre-
     agreed template (i.e., both types of authorization requests
     refer to the pre-agreed template). Templates will be agreed in

an out-of-band mechanism by the VPN Operator and the PKI

       Operator. It will receive back from the PKI a unique tuple of
       authorization identifiers and one-time authorization tokens that
       will authorize Peers to request a PKC.

     - It will deliver instructions to the IPsec Peers, and the Peers
       will carry out those instructions (e.g., Admin passes Peer
       information necessary to generate keys and PKC request).

2.2 PKI System

   The PKI System, as depicted in Figure 2, can be set up and operated
   by the Operator (in-house), be provided by third party PKI providers
   to which connectivity is available at the time of provisioning
   (managed PKI service), or be integrated with the VPN product.

```
        +-----------------------------------------------+
        |          +------------------------+           |
        |          v                        |           |
        |    +--------------+                v           |
        |    |  Repository  |    +----+   +----+         |
        |    | Certs & CRLs |<-> | CA |<->| RA |         |
        |    +--------------+    +----+   +----+         |
        |                                               |
        +-----------------------------------------------+
```
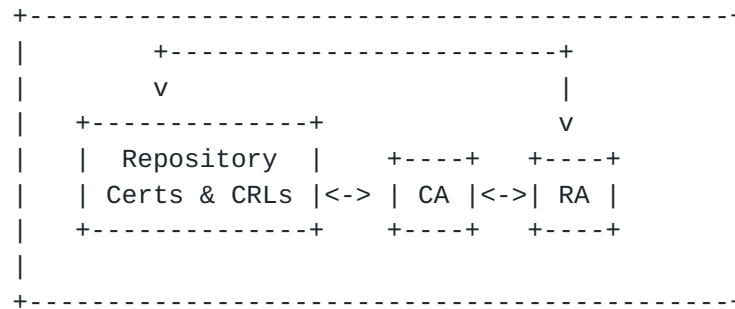
                      Figure 2: PKI System

   This framework assumes that all components of the VPN obtain PKCs
   from a single PKI community. An IPsec Peer can accept a PKC from a
   Peer that is from a CA outside of the PKI community, but the auto
   provision and life cycle management for such a PKC or its trust
   anchor PKC fall out of scope.

   The PKI System contains a mechanism for handling Admin s
   authorization requests and PKC enrollments. This mechanism is
   referred to as the Registration Authority (RA). The PKI System
   contains a Repository for Peers to retrieve each other s PKCs and
   revocation information. Last, the PKI System contains the core
   function of a CA that uses a public and private key pair and signs
   PKCs.

2.3 VPN-PKI Interaction

   The interaction between the VPN System and the PKI System is the key
   focus of this requirements document, as shown in Figure 3. It is
   therefore sensible to consider the steps necessary to set up, use and

manage PKCs for one Peer to establish an association with another
Peer.

```
      +-------------------------------------------------+
      |                    PKI System                   |
      |                                                 |
      |    +--------------+                             |
      |    |  Repository  |     +----+    +----+        |
      |    | Certs & CRLs |     | CA |    | RA |        |
      |    +--------------+     +----+    +----+        |
      |                                                 |
      +-------------------------------------------------+
          ^                    ^                  ^
          |[G]                 |[A]               |[G]
          |[E]                 |[G]               |[E]
          |[L]                 |[E]               |[L]
          |[R]                 |[R]               |[R]
          |                    |[L]               |
      +------+-----------------+------------------+-------+
      |      |                 v                  |      |
      |      |          +----------+              |      |
      |      | [G][E][L][R]|   VPN  |[G][E][L][R] |      |
      |      | +--------->|  Admin  |<----------+ |      |
      |      | | |        | Function |          | |      |
      |      | | |        +----------+          | |      |
      |      v v                              v v       |
      |   +---------+                    +---------+    |
      |   |  IPsec  |        [I]         |  IPsec  |    |
      |   |  Peer 1 |<=====================>| Peer 2 |  |
      |   +---------+                    +---------+    |
      |                                                 |
      |                   VPN System                    |
      +-------------------------------------------------+
```
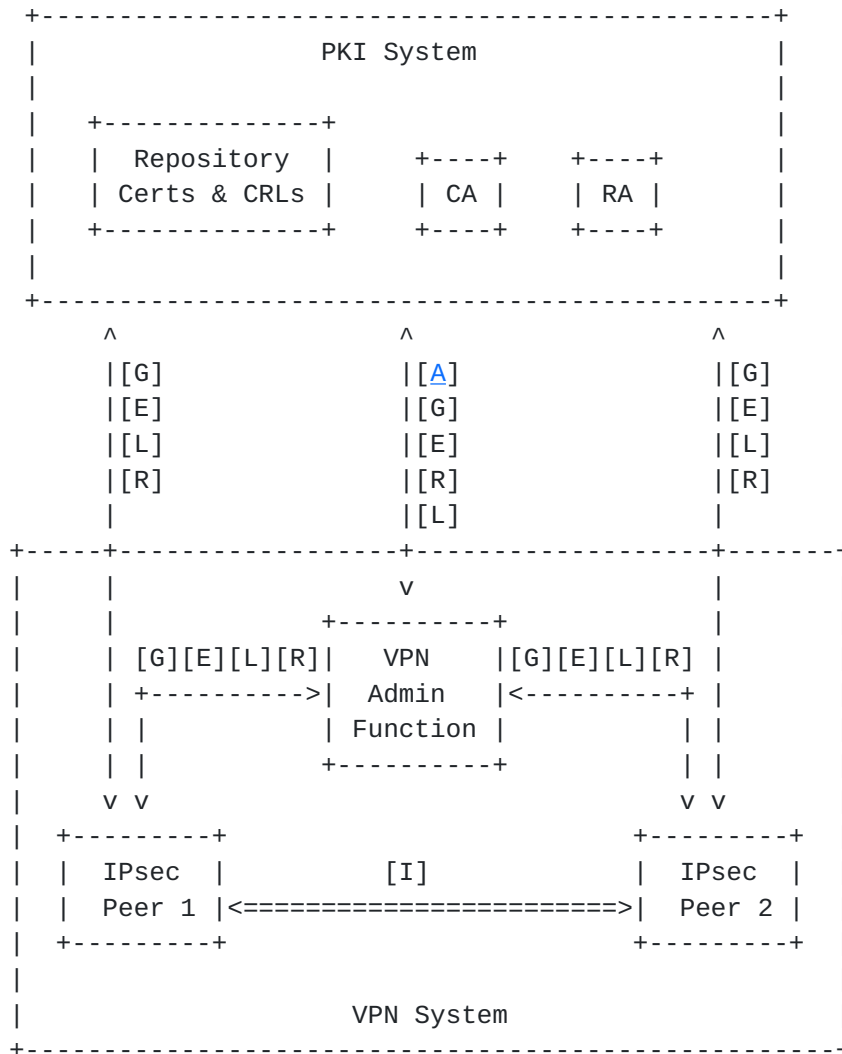
    [A] = Authorization: PKC issuance
    [G] = Generation: Public key, and private key, and PKC request
    [E] = Enrollment: Sending PKC request, verifying PKC response, and
          confirming PKC response
    [I] = IKE and IPsec communication
    [L] = Lifecycle: Rekey, renewal, update, revocation, and
          confirmation
    [R] = Repository: Posting and lookups

        Figure 3. Architectural Framework for VPN-PKI Interaction

    Requirements for each of the interactions, [A], [G], [E], [L], and
    [R], are addressed in paragraphs 3.2-3.6. However, only requirements
    for [A], [E], [L], and [R] will be addressed by the certificate

management profile. Requirements for [I] transactions are beyond the

   scope of this document. Additionally, the act of certification (i.e.,
   binding the public key to the name) is performed at the CA and is not
   shown in the Figure.

3 Requirements

3.1 General Requirements

3.1.1 One Protocol

   The target profile, to be based on this requirements document, MUST
   call for ONE PROTOCOL or ONE USE PROFILE for each main element of the
   [A], [E], [L], and [R] interactions. It is a specific goal to avoid
   multiple competing protocols or profiles to solve the same
   requirement whenever possible to reduce complexity and improve
   interoperability.

   Meeting some of the requirements may necessitate the creation of a
   new protocol or new extension for an existing protocol; however, the
   latter is much preferred.

3.1.2 Secure Transactions

   The target certificate management profile MUST specify the [A], [E],
   [L], and [R] transactions between VPN and PKI Systems. To support
   these transactions, the Admin and PKI MUST exchange policy details,
   identities, and keys. As such, the method of communication for [A],
   [E], and [L] transactions MUST be secured in a manner that ensures
   privacy, authentication, and message data integrity. The
   communication method MUST require that mutual trust be established
   between the PKI and the Admin. See paragraph 3.7.1. [R] transactions
   do not require authentication or message data integrity because the
   responses (i.e., PKCs and CRLs) are already digital signed. Whether
   [R] transactions require privacy is determined by the local security
   policy.

   The target certificate management profile will not specify [G]
   transactions; however, these transactions MUST be secured in a manner
   that ensures privacy, authentication, and message data integrity
   because these transactions are the basis for the other transactions.

3.1.3 Admin Availability

   The Admin MUST be reachable by the Peers. Most implementations will
   meet this requirement by ensuring Peers can connect to the Admin from
   anywhere on the network or Internet. However, communication between

the Admin and Peers can be "off-line". It can, in some environments,
be "moving media" (i.e., the configuration or data is loaded on to a
floppy disk or other media and physically moved to the IPsec Peers).
Likewise, it can be entered directly on the IPsec Peer via a User
Interface (UI). In this case, the Admin function is co-located on the
Peer device itself. Most requirements and scenarios in this document
assume on-line availability of the Admin for the life of the VPN
System.

3.1.3 PKI Availability

Availability is REQUIRED initially for authorization transactions
between the PKI and Admin. Further availability is required in most
cases, but the extent of this availability is a decision point for
the Operator. Most requirements and scenarios in this document assume
on-line availability of the PKI for the life of the VPN System.

Off-line interaction between the VPN and PKI Systems (i.e., where
physical media is used as the transport method) is beyond the scope
of this document.

3.1.4 End-User Transparency

PKI interactions are to be transparent to the user. Users SHOULD NOT
even be aware that PKI is in use. First time connections SHOULD
consist of no more than a prompt for some identification and pass
phrase, and a status bar notifying the user that setup is in
progress.

3.1.5 PKC Profile for PKI Interaction

A PKC used for identity in VPN-PKI transactions MUST include all the
[CERTPROFILE] mandatory fields. It MUST also contain contents
necessary to support path validation and certificate status checking.

It is preferable that the PKC profiles for IPsec transactions
[IKECERTPROFILE] and VPN-PKI transactions (in the certificate
management profile) are the same so that one PKC could be used for
both transaction sets. If the profiles are inconsistent then
different PKCs (and perhaps different processing requirements) might
be required. However, the authors urge that process on other aspects
of this standardization effort continue regardless of the status of
efforts to achieve PKC profile consensus.

3.1.5.1 Identity

   PKCs MUST support identifying (i.e., naming) Peers and Admins. The
   following name forms MUST be supported:

   - Fully-Qualified Domain Name (FQDN)
   - RFC 822 (also called USER FQDN)
   - IPv4 Address
   - IPv6 Address

3.1.5.2 Key Usage

   PKCs MUST support indicating the purposes for which the key (i.e.,
   digital signature) can be used. Further, PKCs MUST always indicate
   that relying parties (i.e., Peers) need to understand the indication.

3.1.5.3 Extended Key Usage

   Extended Key Usage (EKU) indications are not required. The presence
   or lack of an EKU MUST NOT cause an implementation to fail an IKE
   connection.

3.1.5.4 Revocation Information Location

   PKCs MUST indicate the location of CRL such that any Peer who holds
   the PKC locally will know exactly where to go and how to request the
   CRL.

3.1.6 Error Handling

   The protocol for the VPN-PKI transactions MUST specify error handling
   for each transaction. Thorough error condition descriptions and
   handling instructions will greatly aid interoperability efforts
   between the PKI and VPN System products.

3.2 Authorization

   This section refers to the [A] elements labeled in Figure 3.

3.2.1 One Protocol

   One protocol MUST be specified for these Admin to PKI (RA/CA)
   interaction. This protocol MUST support privacy, authorization,

   authentication, and integrity. PKCs for authorization of the Admin
   can be initialized through an out-of-band mechanism.

   The transport used to carry the authorization SHOULD be reliable
   (TCP).

   The protocol SHOULD be as lightweight as possible.

3.2.2 Bulk Authorization

   Bulk authorization MUST be supported by the certificate management
   profile. Bulk authorization occurs when the Admin requests of the PKI
   that authorization be established for several different subjects with
   almost the same contents. A minimum of one value (more is also
   acceptable) differs per subject. Because the authorizations may occur
   before any keys have been generated, the only way to ensure unique
   authorization identifiers are issued is to have at least one value
   differ per subject.

   Authorization can occur prior to a PKC enrollment request, or the
   authorization and the PKC enrollment request can be presented to the
   PKI at the same time. Both of these authorization scenarios MUST be
   supported.

   A bulk authorization SHOULD occur in one single connection to the PKI
   (RA/CA), with the number of subjects being one or greater.
   Implementations SHOULD be able to handle one thousand subjects in a
   batch authorization.

3.2.3 Authorization Scenario

The authorization scenario for VPN-PKI transactions involves a two-
step process: an authorization request and an authorization
response. Figure 4 shows the salient interactions to perform
authorization transactions.

```
    +--------------+     +----------------------+
    |  Repository  |     |         CA/RA        |
    +--------------+     +----------------------+
                                    ^
                                    | 1
                                 2  |
                                    v
                              +-------+
                              | Admin |
                              +-------+



       +------------------+       +--------+
       |      IPsec        |      | IPsec  |
       |      Peer 1       |      | Peer 2 |
       +------------------+       +--------+
```
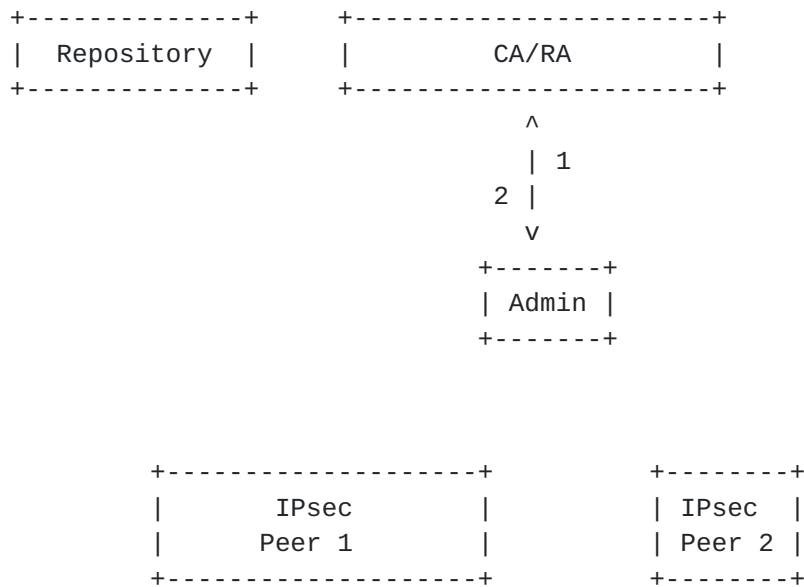
Figure 4. Authorization Transactions

1) Authorization [A]. Admin sends a list of identities and PKC
contents for the PKI System to authorize enrollment. See paragraph
3.2.4.

2) Authorization Response [A]. The PKI returns a list of unique
authorization identifiers and one-time authorization tokens to be
used for the enrollment of each PKC (1). Response may indicate
success, failure, or errors for any particular authorization. See
paragraph 3.2.5.

3.2.4 Authorization Request

3.2.4.1 Specifying Fields within the PKC

The Admin authorizes individual PKCs or batches of PKC issuances
based on a pre-agreed template. This template is agreed by the VPN
Operator and PKI Operator and is referred to in each authorization
request. This allows the authorization requests to include the
minimal amount of information necessary to support a VPN System.

The Admin can send the PKI System the set of PKC contents that it
wants the PKI to issue to a group of IPsec Peers. In other words, it
tells the PKI System, "if you see a PKC request that looks like this,
from this person, process it and issue the PKC."

Requirements for PKC fields used in IPsec transactions are specified
in [IKECERTPROFILE].

Requirements for PKC fields used in VPN-PKI transactions are
specified in paragraph 3.1.5.

3.2.4.2 **Authorizations for Renewal, Update, and Rekey**

When the VPN Operator and PKI Operator pre-agree on a template, they
MUST also agree on the local policy regarding PKC renewal and PKC
update. These are:

  - Admin MUST specify if automatic renewals are allowed, that is,
    the Admin authorizes the PKI to process a future renewal for the
    specified Peer PKC.

  - Admin MUST specify if PKC update is allowed, that is, the Admin
    authorizes the PKI to accept a future request for a new PKC with
    changes to non-key-related fields.

If a PKC renewal is authorized, the Admin MUST further specify:

  - Who can renew, that is, can only the Admin send a renewal request
    or can the Peer send a request directly to the PKI, or either.

  - Specify at how long before the PKC expiration date the PKI will
    accept and process a renewal (i.e., N% of validity period, or
    the UTC time after which renewal is permitted).

If PKC update is authorized, the Admin MUST further specify:

  - The aspects of non-key-related fields that are changeable.

  - The entity that can send the PKC Update request, that is, only
    the Admin, only the Peer, or either.

  - Specify at how long before the PKC expiration date the PKI will
    accept and process an update (i.e., N% of validity period, or
    the UTC time after which update is permitted).

A new authorization by the Admin is REQUIRED for PKC rekey. No
parameters of prior authorizations need be considered.

3.2.4.3 Other Authorization Elements

The Admin MUST have the ability to specify the format for the
authorization ID and one-time authorization token. The one-time
authorization token SHOULD be unique per authorization ID. The more
randomness that can be achieved in the relationship between an
authorization ID and its one-time authorization token, the better.
The one-time authorization token MUST be in UTF8 format to avoid
incompatibilities that may occur due to international characters. It
MUST support normalization as in [CERTPROFILE]. The Admin MUST have
the ability to constrain the UTF8 character set.

There MUST be an option to specify a validation period for the
authorization ID and its one-time authorization token. If such a
validation period is set, any PKC requests using this authorization
ID and one-time authorization token that arrive at the PKI outside of
the validation period MUST be dropped and the event logged.

The Protocol SHOULD consider what happens when Admin requested
information conflicts with PKI settings such that the Admin request
cannot be issued as requested (e.g., Admin requests validation period
= 3 weeks and CA is configured to only allow validation periods = 1
week). Proper conflict handling MUST be specified.

3.2.4.4 Cancel Capability

Either the Admin or the Peer can send a cancel authorization message
to PKI. The canceling entity MUST provide the authorization ID and
one-time authorization token in order to cancel the authorization. At
that point, the authorization will be erased from the PKI, and a log
entry of the event written.

After the cancellation has been verified (a Cancel, Cancel ACK, ACK
type of a process is REQUIRED to cover a lost connections scenario),
the PKI will accept a new authorization request with the exact same
contents as the canceled one, except that the identifier MUST be new.
The PKI MUST NOT process duplicate authorization requests.

Note that if the PKI has already issued a PKC associated with an
authorization, then cancellation of the authorization is not
possible and the authorization request SHOULD be refused by the PKI.
Once a PKC has been issued it MUST be revoked in accordance with
clause 3.6.

3.2.5 Authorization Response

If the authorization request is acceptable, the PKI will respond to
the Admin with a unique authorization identifier per subject
authorization requested and a one-time authorization token per
authorization ID. See paragraph 3.2.4.3 for additional authorization
ID and one-time authorization token requirements.

The PKI can alter parameters of the authorization request submitted
by the Admin. In that event, the PKI MUST return all the contents of
the authorization request (as modified) to the Admin with the
confirmation of authorization success. This will allow the Admin to
perform an "operational test" to verify that the issued PKCs will
meet its requirements. If the Admin determines that the modified
parameters are unacceptable, then the authorization should be
cancelled in accordance with clause 3.2.4.4.

After receiving a bulk authorization request from the Admin, the PKI
MUST be able to reply YES to those individual PKC authorizations that
it has satisfied and NO or FAILED for those requests that cannot be
satisfied, along with sufficient reason or error codes.

A method is REQUIRED to identify if there is a change in PKI setting
between the time the authorization is granted and PKC request occurs,
and what to do about the discrepancy.

3.2.5.1 Error Handling for Authorization

Thorough error condition descriptions and handling instructions MUST
be provided to the Admin for each transaction in the authorization
process. Providing such error codes will greatly aid interoperability
efforts between the PKI and IPsec products.

3.3 Generation

This section refers to the [G] elements labeled in Figure 3.

Once the PKI System has responded with authorization identifiers and
authorization tokens (see paragraph 3.2), and this information is
received at the Admin, the next step is to generate public and
private key pairs and to construct PKC requests using those key
pairs. The key generations can occur at one of three places,
depending on local requirements: at the IPsec Peer, at the Admin, or
at the PKI. The PKC request can come from either the IPsec Peer, a
combination of the Peer and the Admin, or not at all.

3.3.1 **Generation Method 1: IPsec Peer Generates Key Pair, Constructs
     PKC Request, and Signs PKC Request**

   This option will be used most often in the field. This is the most
   secure method for keying, as the keys are generated on the end entity
   and the private key never leaves the end entity. However, it is the
   most computationally intensive for the Peer as it must be "ASN.1
   aware" to support generating and digitally signing the PKC request.

```
      +--------------+     +-----------------------+
      |  Repository  |     |          CA/RA        |
      +--------------+     +-----------------------+




                           +-------+
                    +------>| Admin |
                    |       +-------+
                    |
                    | 1
                    V
         +--------------------+          +--------+
      2  |       IPsec        |          | IPsec  |
         |       Peer 1       |          | Peer 2 |
         +--------------------+          +--------+
```

                 Figure 5. Generation Interactions:
          IPsec Peer Generates Key Pair and Constructs PKC Request

   1) Opaque transaction. Admin sends authorization identifier, one-time
   authorization token, and any other parameters needed by the Peer to
   generate the PKC request, including key type and size.

   2) Generation [G]. Peer receives authorization identifier, one-time
   authorization token, and any parameters. Peer generates key pair and
   constructs PKC request.

   Steps prior to these can be found in paragraph 3.2. The next step,
   enrollment, can occur either directly between the Peer and PKI (see
   paragraph 3.4.5) or through the Admin (see paragraph 3.4.6).

3.3.2 **Generation Method 2: IPsec Peer Generates Key Pair, Admin
     Constructs PKC Request, Admin Signs PKC Request**

This option also supports IPsec Peer generation of key pair, but
removes the requirement for the Peer to be ASN.1 aware because it
does not have to construct or digitally sign the PKC request. The
drawback is that the key pair does need to be provided to the Admin.

```
    +--------------+     +----------------------+
    |  Repository  |     |        CA/RA         |
    +--------------+     +----------------------+




                    3 +-------+
                +------>| Admin | 4
                |       +-------+
                |
                | 1
                V
    +--------------------+       +--------+
  2 |       IPsec         |       | IPsec  |
    |       Peer 1        |       | Peer 2 |
    +--------------------+       +--------+
```
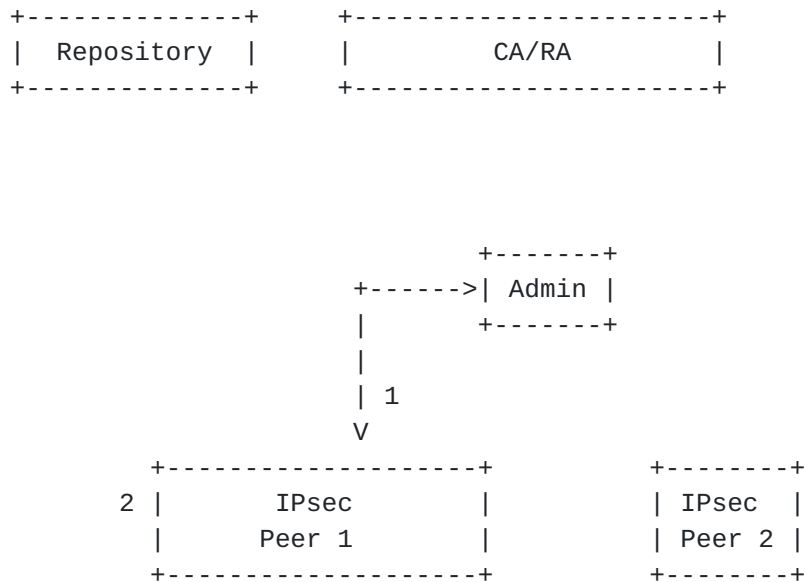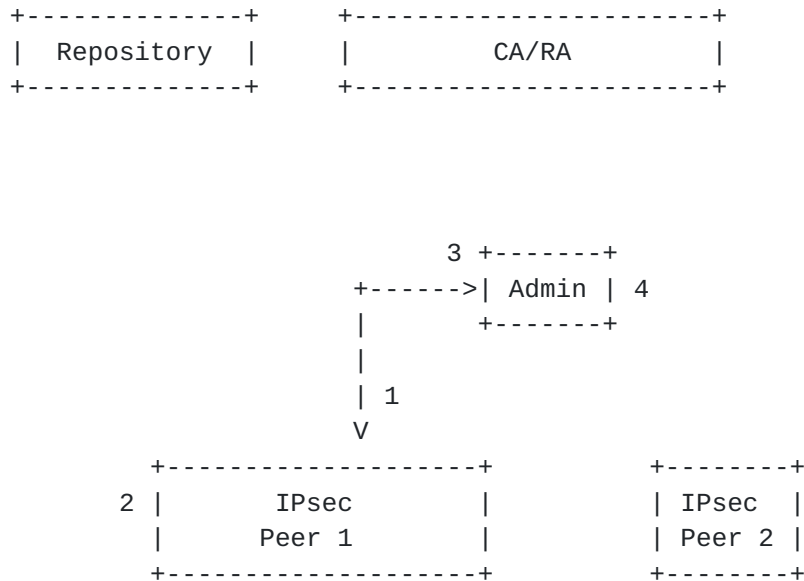
              Figure 6. Generation Interactions:
        IPsec Peer Generates Key Pair, Admin Constructs PKC Request

1) Opaque transaction. Admin sends command to Peer to generate key
pair, based on parameters provided in the command.

2) Generation [G]. Peer generates key pair.

3) Opaque transaction [G]. Peer returns key pair to Admin.

4) Generation [G]. Admin constructs and digitally signs PKC request.

Steps prior to these can be found in paragraph 3.2. The next step,
enrollment, occurs through the Admin (see paragraph 3.4.7).

3.3.3 Generation Method 3: Admin Generates Key Pair, Constructs PKC Request, and Signs PKC Request

   This option exists for deployments where Peers cannot generate their
   own key pairs. Some examples are for PDAs and handsets where to
   generate an RSA key would be operationally impossible due to
   processing and battery constraints. Another case covers key recovery
   requirements, where the same PKCs are used for other functions in
   addition to IPsec, and key recovery is required (e.g. local data
   encryption), therefore key escrow is needed off the Peer. If key
   escrow is performed then the exact requirements and procedures for it
   are beyond the scope of this document.

```
      +--------------+     +-----------------------+
      |  Repository  |     |         CA/RA         |
      +--------------+     +-----------------------+




                          +-------+
                          | Admin | 1
                          +-------+



      +--------------------+        +--------+
      |       IPsec        |        | IPsec  |
      |       Peer 1       |        | Peer 2 |
      +--------------------+        +--------+
```
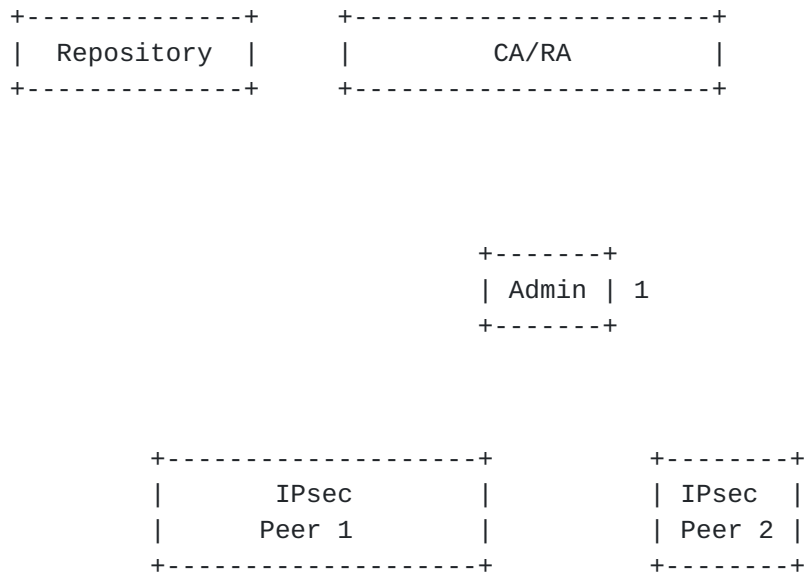
             Figure 7. Generation Interactions:
       Admin Generates Key Pair and Constructs PKC Request

   1) Generation [G]. Admin generates key pair, constructs PKC request,
   and digitally signs PKC request.

   Steps prior to these can be found in paragraph 3.2. The next step,
   enrollment, occurs through the Admin (see paragraph 3.4.8).

   Note that separate authorizations step are still of value even though
   the Admin is the also performing the key generation. The PKC
   template, Subject fields, SubjectAltName fields and more are part of
   the request, and must be communicated in some way from the Admin to
   the PKI. Instead of creating a new mechanism, the authorization
   schema can be reused. This also allows for the feature of role-based
   administration, where Operator 1 is the only one allowed to have the
   Admin function pre-authorize PKCs, but Operator 2 is the one doing
   batch enrollments and VPN device configurations.

3.3.4 Method 4: PKI Generates Key Pair

This option exists for deployments where end entities cannot generate
their own key pairs and the Admin function is a minimal
implementation. The PKI and Admin pre-agree to have the PKI generate
key pairs and PKCs. This is, in all likelihood, the easiest way to
deploy PKCs, though it sacrifices some security since both the CA and
the Admin have access to the private key. However, in cases where key
escrow is required, this may be acceptable. The Admin effectively
acts as a proxy for the Peer in the PKC enrollment process.

```
   +--------------+     +-----------------------+
   |  Repository  |     |         CA/RA         | 1
   +--------------+     +-----------------------+




                         +-------+
                         | Admin |
                         +-------+



      +--------------------+        +--------+
      |        IPsec       |        | IPsec  |
      |        Peer 1      |        | Peer 2 |
      +--------------------+        +--------+
```
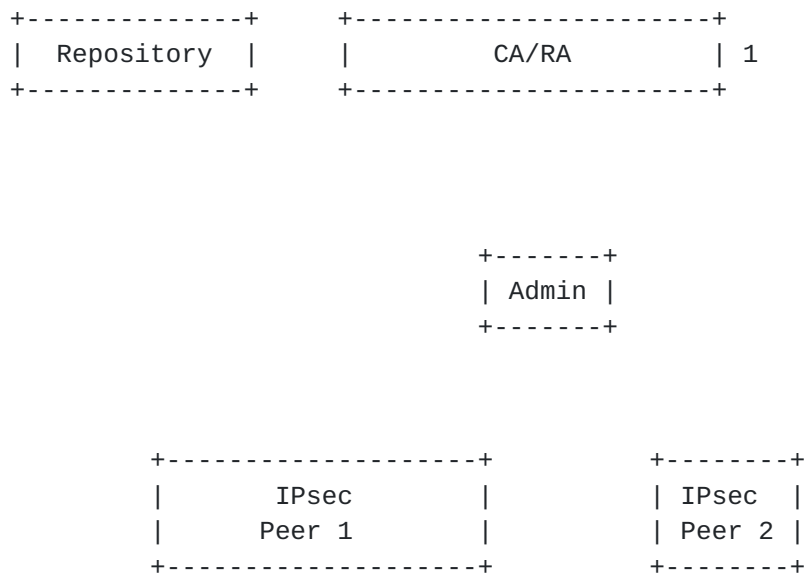
                Figure 8. Generation Interactions:
      IPsec Peer Generates Key Pair, Admin Constructs PKC Request

1) Generation [G] The PKI generates the key pair.

Steps prior to these can be found in paragraph 3.2. The next step,
enrollment, occurs through the Admin (see paragraph 3.4.9).

3.3.5 Error Handling for Generation

Thorough error condition descriptions and handling instructions MUST
be provided for each transaction in the key generation and PKC
request construction process. Providing such error codes will greatly
aid interoperability efforts between the PKI and IPsec products.

Error conditions MUST be communicated to the Admin regardless of who
generated the key or PKC request.

[3.4](#) Enrollment

   This section refers to the [E] elements labeled in Figure 3.

   Regardless of where the keys were generated and the PKC request
   constructed, an enrollment process will need to occur to request that
   the PKI issue a PKC and the corresponding PKC be returned.

   The protocol MUST be exactly the same regardless of whether the
   enrollment occurs from the Peer to the PKI or from the Admin to the
   PKI.

[3.4.1](#) One protocol

   One protocol MUST be specified for enrollment requests, responses,
   and confirmations.

[3.4.2](#) On-line protocol

   The protocol MUST support enrollment that occurs over the Internet
   and without the need for manual intervention.

[3.4.3](#) Single Connection with Immediate Response

   Enrollment requests and responses MUST be able to occur in one on-
   line connection between the Admin on behalf of the Peer or the Peer
   itself and the PKI (RA/CA).

[3.4.4](#) Manual Approval Option

   Manual approval of PKC enrollments is too time consuming for large
   scale implementations, and is therefore not required.

[3.4.5](#) Enrollment Method 1: Peer Enrolls to PKI Directly

   In this case, the IPsec Peer only communicates with the PKI after
   being commanded to do so by the Admin. This enrollment mode is
   depicted in Figure 9 and the letters in the following description
   refer to Figure 3. Prior authorization (see paragraph 3.2) and
   generation (see paragraph 3.3.1) steps are not shown.

   Most IPsec Systems have enough CPU power to generate a public and
   private key pair of sufficient strength for secure IPsec. In this
   case, the end entity needs to prove to the PKI that it has such a key
   pair; this is normally done by the PKI sending the end entity a
   nonce, which the end entity signs and returns to the Admin along with
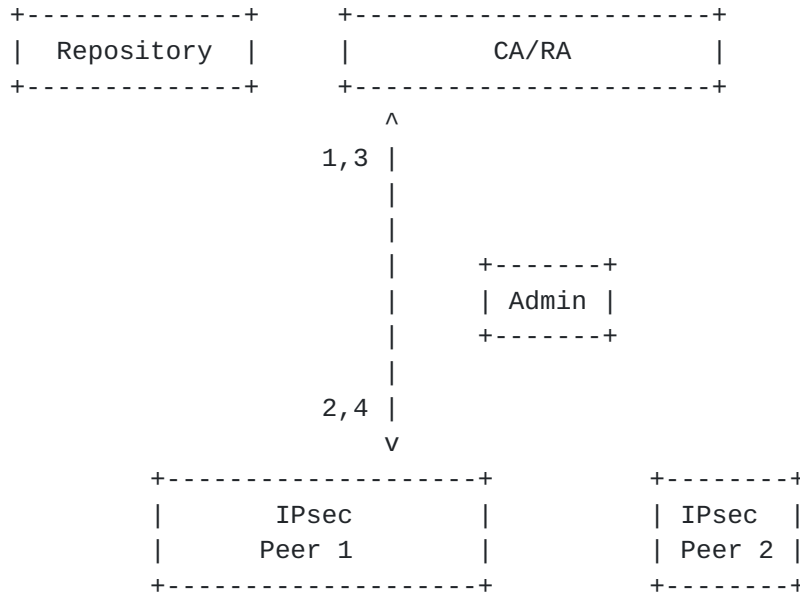   the end entity s public key.

        +--------------+   +----------------------+
        | Repository | | CA/RA |
        +--------------+   +----------------------+
                                ^
                           1,3 |
                                |
                                |
                                | +-------+
                                | | Admin |
                                | +-------+
                                |
                           2,4 |
                                v
           +--------------------+   +--------+
           | IPsec | | IPsec |
           | Peer 1 | | Peer 2 |
           +--------------------+   +--------+

             Figure 9. VPN-PKI Interaction Steps:
            IPsec Peer Generates Keys and PKC Request,
                    Enrolls Directly with PKI

   1) Enrollment Request [E]. The IPsec Peer sends PKC requests from the
   PKI, providing the generated public key.

   2) Enrollment Response [E]. The PKI responds to the enrollment
   request, providing either the new PKC that was generated or a
   suitable error indication.

   3) Enrollment Confirmation [E]. Peer positively acknowledges receipt
   of new PKC.

   4) Enrollment Confirmation Receipt [E]. PKI sends enrollment
   confirmation receipt back to the Peer.

3.4.6 Enrollment Method 2a: Peer Enrolls through Admin

In this case, the IPsec Peer has generated the key pair and the PKC
request, but does not enroll directly to the PKI System. Instead, it
automatically sends its request to the Admin, and the Admin redirects
the enrollment to the PKI System. The PKI System does not care where
the enrollment comes from, as long as it is a valid enrollment. Once
the Admin receives the PKC response, it automatically forwards it to
the IPsec Peer.

Most IPsec Systems have enough CPU power to generate a public and
private key pair of sufficient strength for secure IPsec. In this
case, the end entity needs to prove to the Admin that it has such a
key pair; this is normally done by the Admin sending the end entity a
nonce, which the end entity signs and returns to the Admin along with
the end entity s public key.

This enrollment mode is depicted in Figure 10 and the letters in the
following description refer to Figure 3. Prior authorization (see
paragraph 3.2) and generation (see paragraph 3.3.1) steps are not
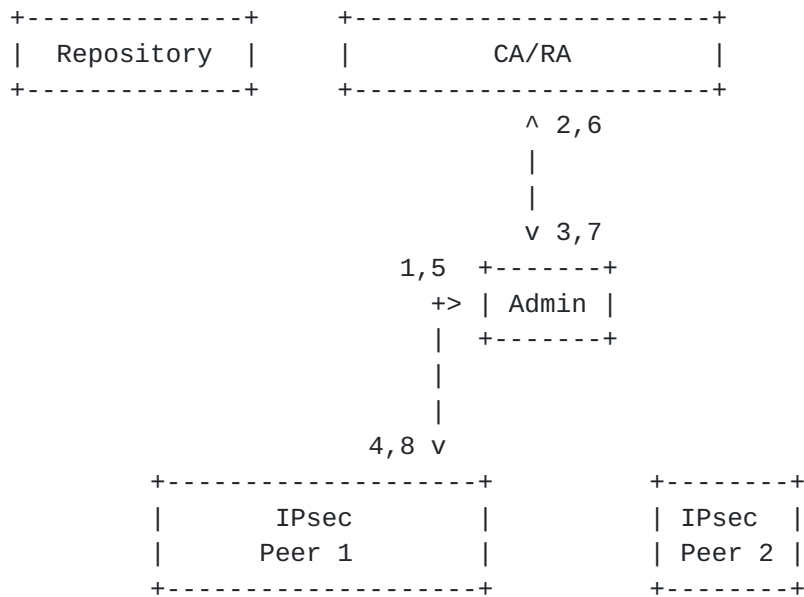shown.

```
     +--------------+     +----------------------+
     |  Repository  |     |        CA/RA         |
     +--------------+     +----------------------+
                                   ^ 2,6
                                   |
                                   |
                                   v 3,7
                          1,5  +-------+
                            +> | Admin |
                            |  +-------+
                            |
                            |
                      4,8 v
        +--------------------+           +--------+
        |       IPsec        |           | IPsec  |
        |       Peer 1       |           | Peer 2 |
        +--------------------+           +--------+
```

Figure 10.   VPN-PKI Interaction Steps:
IPsec Peer Generates Keys and PKC Request,
Enrolls Through Admin

1) Opaque Transaction [E]. The IPsec Peer requests a PKC from the
Admin, providing the generated public key.

2) Enrollment [E]. The Admin forwards the enrollment request to the
PKI.

3) Enrollment Response [E]. The PKI responds to the enrollment
request, providing either the new PKC that was generated or a
suitable error indication.

4) Opaque Transaction [E]. The Admin forwards the enrollment response
back to the IPsec Peer.

5) Opaque Transaction [E]. Peer must positively acknowledge receipt
of new PKC back to the Admin.

6) Enrollment Confirmation [E]. Admin forwards enrollment
confirmation back to the PKI.

7) Enrollment Confirmation Receipt [E]. PKI sends enrollment
confirmation receipt back to the Admin.

8) Opaque Transaction [E]. Admin forwards PKI's enrollment
confirmation receipt back to the Peer.

3.4.7 Enrollment Method 2b: Peer Enrolls Through Admin

In this case, the IPsec Peer has generated the key pair, but the PKC
request is constructed and signed by the Admin. The PKI System does
not care where the enrollment comes from, as long as it is a valid
enrollment. Once the Admin retrieves the PKC, it then automatically
forwards it to the IPsec Peer along with the key pair.

Some IPsec Systems do not have enough CPU power to generate a public
and private key pair of sufficient strength for secure IPsec. In this
case, the Admin needs to prove to the PKI that it has such a key
pair; this is normally done by the PKI sending the Admin a nonce,
which the Admin signs and returns to the PKI along with the end
entity s public key.

This enrollment mode is depicted in Figure 11 and the letters in the
following description refer to Figure 3. Prior authorization (see
paragraph 3.2) and generation (see paragraph 3.3.2) steps are not
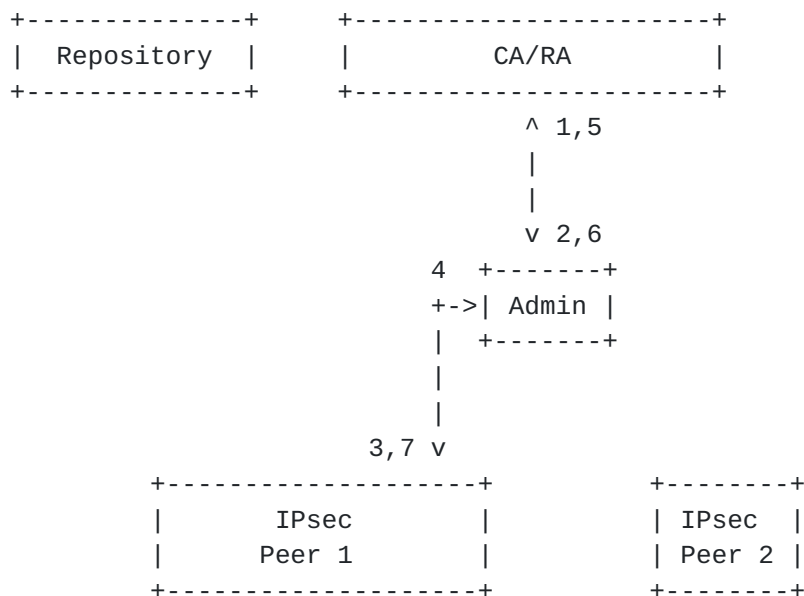shown.

```
      +--------------+       +----------------------+
      |  Repository  |       |         CA/RA        |
      +--------------+       +----------------------+
                                        ^ 1,5
                                        |
                                        |
                                        v 2,6
                              4  +-------+
                              +->| Admin |
                              |  +-------+
                              |
                              |
                          3,7 v
          +-------------------+          +--------+
          |       IPsec       |          | IPsec  |
          |       Peer 1      |          | Peer 2 |
          +-------------------+          +--------+
```

Figure 11.   VPN-PKI Interaction Steps:
IPsec Peer Generates Keys, Admin Constructs and
Signs PKC Request, Enrolls Through Admin

1) Enrollment [E]. The Admin requests a PKC from the PKI, providing
the generated public key.

2) Enrollment Response [E]. The PKI responds to the enrollment
request, providing either the new PKC that was generated or a
suitable error indication.

3) Opaque Transaction [E]. The Admin forwards the enrollment response
back to the IPsec Peer.

4) Opaque Transaction [E]. Peer positively acknowledge receipt of new
PKC back to the Admin.

5) Enrollment Confirmation [E]. Admin forwards enrollment
confirmation back to the PKI.

6) Enrollment Confirmation Receipt [E]. PKI sends enrollment
confirmation receipt back to the Admin.

7) Opaque Transaction [E]. Admin forwards PKI's enrollment
confirmation receipt back to the Peer.

3.4.8 Enrollment Method 3a: Admin Authorizes and Enrolls Directly to PKI

In this case, the Admin generates the key pair, PKC request, and
digitally signs the PKC request. The PKI System does not care where
the enrollment comes from, as long as it is a valid enrollment. Once
the Admin retrieves the PKC, it then automatically forwards it to the
IPsec Peer along with the key pair.

Some IPsec Systems do not have enough CPU power to generate a public
and private key pair of sufficient strength for secure IPsec. In this
case, the Admin needs to prove to the PKI that it has such a key
pair; this is normally done by the PKI sending the Admin a nonce,
which the Admin signs and returns to the PKI along with the end
entity s public key.

This enrollment mode is depicted in Figure 12 and the letters in the
following description refer to Figure 3. Prior authorization (see
paragraph 3.2) and generation (see paragraph 3.3.3) steps are not
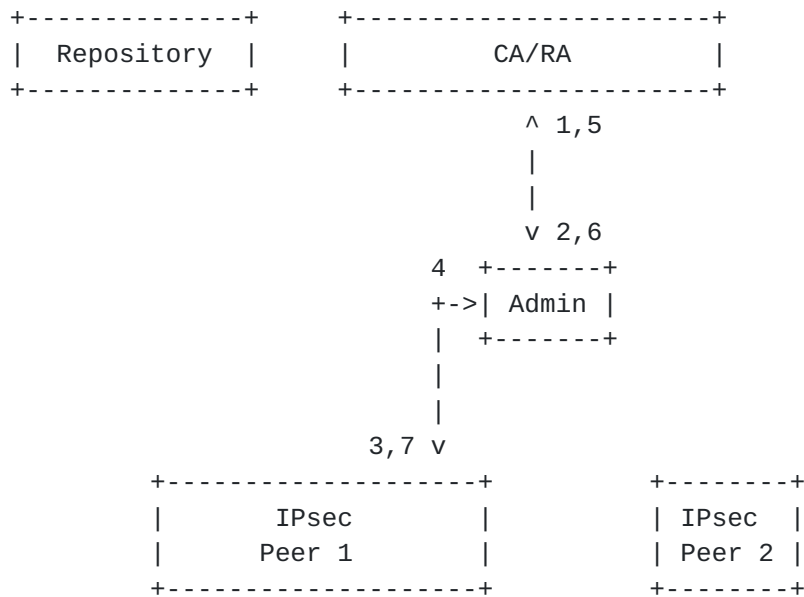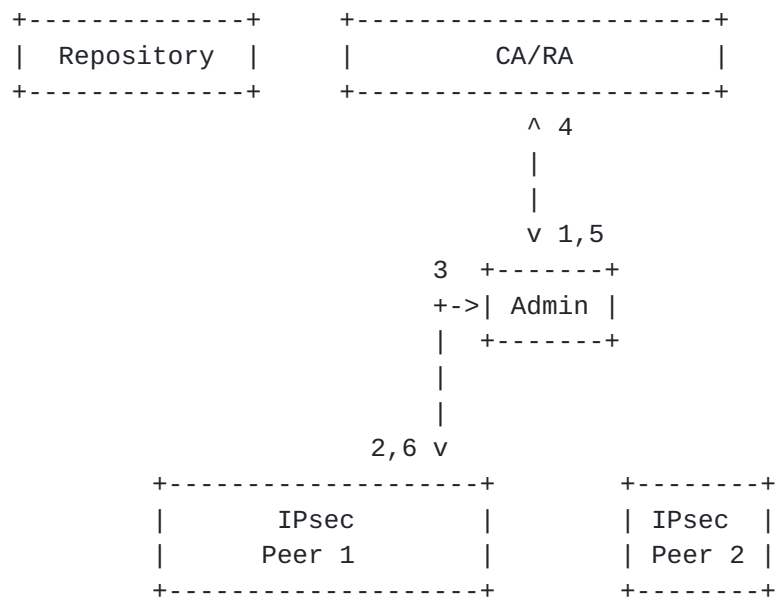shown.

```
      +--------------+      +----------------------+
      |  Repository  |      |         CA/RA        |
      +--------------+      +----------------------+
                                      ^ 1,5
                                      |
                                      |
                                      v 2,6
                            4  +-------+
                            +->| Admin |
                            |  +-------+
                            |
                            |
                       3,7 v
         +--------------------+         +--------+
         |       IPsec        |         | IPsec  |
         |       Peer 1       |         | Peer 2 |
         +--------------------+         +--------+
```

            Figure 12.   VPN-PKI Interaction Steps:
      Admin Generates Keys, PKC Request, and Enrolls Directly
                         with PKI

   1) Enrollment [E]. The Admin requests a PKC from the PKI, providing
   the generated public key.

   2) Enrollment Response [E]. The PKI responds to the enrollment
   request, providing either the new PKC that was generated or a
   suitable error indication.

   3) Opaque Transaction [E]. The Admin forwards the enrollment response
   back to the IPsec Peer, along with the keys.

   4) Opaque Transaction [E]. Peer positively acknowledge receipt of new
   PKC back to the Admin.

   5) Enrollment Confirmation [E]. Admin forwards enrollment
   confirmation back to the PKI.

   6) Enrollment Confirmation Receipt [E]. PKI sends enrollment
   confirmation receipt back to the Admin.

   7) Opaque Transaction [E]. Admin forwards PKI's enrollment
   confirmation receipt back to the Peer.

3.4.9 **Enrollment Method 3b: Admin Authorizes and Enrolls Directly to
       PKI**

   In this instance, the PKI and Admin have previously agreed to have
   the PKI generate key and certificates when the PKI receives an
   authorization request. The PKI returns to the IPsec Peer through the
   Admin, the final product of a key pair and PKC. Again, the mechanism
   for the Peer to Admin communication is opaque.

   This enrollment mode is depicted in Figure 13 and the letters in the
   following description refer to Figure 3. Prior authorization (see
   paragraph 3.2) and generation (see paragraph 3.3.4) steps are not
   shown.

```
        +--------------+      +----------------------+
        |  Repository  |      |        CA/RA         |
        +--------------+      +----------------------+
                                        ^ 4
                                        |
                                        |
                                        v 1,5
                               3  +-------+
                               +->| Admin |
                               |  +-------+
                               |
                               |
                          2,6 v
          +--------------------+        +--------+
          |       IPsec        |        | IPsec  |
          |       Peer 1       |        | Peer 2 |
          +--------------------+        +--------+
```

        Figure 13. VPN-PKI Interaction Steps:
        PKI Generates Keys,

   1) Enrollment Response [E]. The PKI responds to the authorization
   request sent, providing either the new PKC and public-private key
   pair that were generated or a suitable error indication.

   2) Opaque Transaction [E]. The Admin forwards the enrollment response
   back to the IPsec Peer, along with the keys.

   3) Opaque Transaction [E]. Peer positively acknowledge receipt of new
   PKC back to the Admin.

   4) Enrollment Confirmation [E]. Admin forwards enrollment
   confirmation back to the PKI.

   5) Enrollment Confirmation Receipt [E]. PKI sends enrollment
   confirmation receipt back to the Admin.

   6) Opaque Transaction [E]. Admin forwards PKI's enrollment
   confirmation receipt back to the Peer.

3.4.10 Confirmation Handshake

Any time a new PKC is issued by the PKI, a confirmation of PKC
receipt MUST be sent back to the PKI by the Peer or the Admin
(forwarding the Peer s confirmation).

Operationally, the Peer MUST send a confirmation to the PKI verifying
that it has received the PKC, loaded it, and can use it effectively
in an IKE exchange. This requirement exists so that:

  - The PKI does not publish the new PKC in the repository for others
    until that PKC is able to be used effectively by the Peer, and;

  - A revocation may be invoked if the PKC is not received and
    operational within an allowable window of time.

To assert such proof the Peer MUST sign a portion of data with the
new key. The result MUST be sent to the PKI. The entity that actually
sends the result to the PKI MAY be either the Peer (sending it
directly to the PKI) or Admin (the Peer would send it to Admin, and
Admin can in turn send it to the PKI).

The Admin MUST acknowledge the successful receipt of the
confirmation, thus signaling to the Peer that it may proceed using
this PKC in IKE connections. The PKI MUST complete all processing
necessary to enable the Peer s operational use of the new PKC (for
example, writing the PKC to the repository) before sending the
confirmation acknowledgement. The Peer MUST NOT begin using the PKC
until the PKI s confirmation acknowledgement has been received.

3.4.11 **Error Handling for Enrollment**

   Thorough error condition descriptions and handling instructions are
   REQUIRED for each transaction in the enrollment process. Providing
   such error codes will greatly aid interoperability efforts between
   the PKI and IPsec products.

   The profile will clarify what happens if the request and retrieval
   fails for some reason. The following cases MUST be covered:

     - Admin or Peer cannot send the request.

     - Admin or Peer sent the request but the PKI did not receive the
       request.

     - PKI received the request but could not read it effectively.

     - PKI received and read the request, but some contents of the
       request violated the PKI s configured policy such that the PKI
       was unable to generate the PKC.

     - The PKI System generated the PKC, but could not send it.

     - The PKI sent the PKC, but the requestor (Admin or Peer) did not
       receive it.

     - The Requestor (Admin or Peer) received the PKC, but could not
       process it due to incorrect contents, or other PKC-construction-
       related problem.

     - The Requestor failed trying to generate the confirmation.

     - The Requestor failed trying to send the confirmation.

     - The Requestor sent the confirmation, but the PKI did not receive
       it.

     - The PKI received the confirmation but could not process.

   In each case the following questions MUST be addressed:

     - What does Peer do?
     - What does Admin do?
     - What does PKI do?
     - Is Authorization used?

   If a failure occurs after the PKI sends the PKC and before the Peer
   receives it, then the Peer MUST re-request with the same
   authorization ID and one-time authorization token, and the PKI,

seeing the authorization ID and authorization token, MUST send the
PKC again.

Enrollment errors MUST be sent to the Admin regardless of entity that
generated the enrollment request.

3.5 Lifecycle

This section refers to the [L] elements labeled in Figure 3.

Once the PKI has issued a PKC for the end entity Peer, the Peer MUST
be able to either contact the PKI directly or through the Admin for
any subsequent renewals, updates, rekeys, or revocations. The PKI
MUST support either case for renewals, updates, and revocations.
Rekeys are Admin initiated therefore Peer initiated rekeys MUST be
transferred via the Admin.

3.5.1 One Protocol

One protocol MUST be specified for rekey, renew, and update
requests, responses, and confirmations. It MUST be the same protocol
as is specified in paragraph 3.4.

Revocation requests MAY use the same protocol as rekey, renew, and
update operations. Revocation requests MAY also occur via email,
telephone, Instant Messaging, etc.

3.5.2 PKC Rekeys, Renewals, and Updates

Renewals, updates, and rekeys are variants of a PKC enrollment
request scenario with unique operational and management requirements.

 - A PKC rekey replaces an end entity's PKC with a new PKC that has a
   new public key for the same SubjectName and SubjectAltName
   contents before the end entity s currently held PKC expires.

 - A PKC renewal replaces an end entity's PKC with the same public key
   for the same SubjectName and SubjectAlternativeName contents as
   an existing PKC before that PKC expires.

 - A PKC update is defined as a new PKC issuance with the same public
   key for an altered SubjectName or SubjectAlternativeName before
   expiration of the end entity s current PKC.

When sending renew, update, or rekey requests, the entire contents of
the PKC request needs to be sent to the PKI, not just the changed
elements.

The renew, update and rekey requests MUST be signed by the private
key of the old PKC. This will allow the PKI to verify the identity of
the requestor, and ensure that an attacker does not submit a request
and receive a PKC with another end entity s identity.

Whether or not a new key is used for the new PKC in a renew or update
scenario is a matter of local security policy, and MUST be specified
by the Admin to the PKI in the original authorization request. Re-
using the same key is permitted, but not encouraged. If a new key is
used, the update or renew request must be signed by both the old key
-- to prove the right to make the request -- and the new key -- to
use for the new PKC.

The new PKC resulting from a renew, update or rekey will be retrieved
in-band, using the same mechanism as a new PKC request.

For the duration of time after a renew, update, or rekey has been
processed and before PKI has received confirmation of the Peer s
successful receipt of the new PKC, both PKCs (the old and the new)
for the end entity will be valid. This will allow the Peer to
continue with uninterrupted IKE connections with the previous PKC
while the renewal, update, or rekey process occurs.

After the renewal, update or rekey occurs, the question now exists
for the PKI of what to do about the old PKC. If the old PKC is to be
made unusable, the PKI will need to add it to the revocation list,
removed from the repository; however this should only occur once all
connections that used the old PKC have expired. The decision about if
the old PKC should be made unusable is a decision of local policy.
Either the PKI or the Admin MUST specify this parameter during the
authorization phase. In this case, the specifying party, either the
Admin or the PKI, MUST also specify during authorization the length
of time after the PKI receives the end entity Peer s confirmation (of
receipt of the PKC) that will pass before the old PKC is made
unusable.

In the case where the new keys were generated for a renew or update
request and for rekey requests, once the Peer receives the
confirmation acknowledgement from the PKI, it is good practice for
the old key pair to be destroyed as soon as possible. Deletion can
occur once all connections that used the old PKC have expired.

If a PKC has been revoked, it MUST NOT be allowed a renewal, update
or rekey.

Should the PKC expire without renewal, update or rekey, an entirely
new request MUST be made.

3.5.2.1 Rekey Request

Admins manage rekeys to ensure uninterrupted use of the VPN by Peers
with new keys. Rekeys can occur automatically if the Admin is
configured to initiate a new authorization for the rekey.

Scenarios for rekey are omitted as they use the same scenarios used
in the original PKC enrollment from sections 3.2, 3.3, and 3.4.

3.5.2.1 Renew Request

Admins manage renewals to ensure uninterrupted use of the VPN by
Peers with the same key pair.

At the time of authorization, certain details about renewal
acceptance will be conveyed by the Admin to the PKI, as stated in
section 3.2.4.2. The renewal request MUST match the conditions that
were specified in the original authorization for:

   - Keys: New, existing, or either
   - Requestor: End entity Peer, Admin, either
   - Period: How soon before PKC expiry.
   - Time: Length of time before making the old PKC unusable.

If any of these conditions are not met, the PKI must reject the
renewal and log the event.

Scenarios for renewal are omitted as they use the same scenarios used
in the original PKC enrollment from sections 3.2, 3.3, and 3.4.

3.5.2.2 Update Request

An update to the contents of a PKC will be necessary when details
about an end entity Peer s identity change, but the Operator does not
want to generate a new PKC from scratch, requiring a whole new
authorization. For example, a gateway device may be moved from one
site to another. Its IPv4 Address will change in the SubjectAltName
extension, but all other information could stay the same. Another
example is an end user who gets married and changes the last name or
moves from one department to another. In either case, only one field
(the Surname or OU in the DN) need change.

An update differs from renewal and rekeys in a few ways:

  - A new key is not necessary

  - The timing of the update event is not predictable, as is the case
    with a scheduled renewal or rekey

  - The update request may occur at any time during a PKC s period of
    validity

  - Once the update is completed, and the new PKC is confirmed, the
    old PKC should cease to be usable, as its contents no longer
    accurately describe the subject

At the time of authorization, certain details about update acceptance
can be conveyed by the Admin to the PKI, as stated in section
3.2.4.2. The update request MUST match the conditions that were
specified in the original authorization for:

  - Keys: new or existing or either
  - Requestor: End entity Peer, Admin, either
  - The fields in the Subject and SubjectAltName that are changeable
  - Length of time before making the old PKC unusable

If any of these conditions are not met, the PKI MUST reject the
update and log the event.

If an update authorization was not made at the time of original
authorization, one can be made from Admin to the PKI at any time
during the PKC s valid life. When such an update is desired, Admin
must notify the PKI System that an update is authorized for the end
entity, and to expect it coming, and specify the new contents. Admin
then initiates the update request with the given contents in whatever
mechanism the VPN System employs (direct from end entity to PKI, from
end entity through Admin, or directly from Admin).

Scenarios for update are omitted as they use the same scenarios used
in the original PKC enrollment from sections 3.2, 3.3, and 3.4.

3.5.2.3 **Error Handling for Rekey, Renewal, and Update**

Thorough error condition descriptions and handling instructions are
required for each transaction in the renewal, update or rekey
process. Providing such error codes will greatly aid interoperability
efforts between the PKI and IPsec products.

3.5.2.4 Confirmation Handshakes

The confirmation handshake requirements are the same as in clauses
3.2, 3.3, and 3.4 except that depending on the Adminitrative policy
the PKI MUST also issue a revocation on the original PKC before
sending the confirmation response.

3.5.3 Revocation

The Peer MUST be able to initiate revocation for its own PKC. In this
case the revocation request MUST be signed by the Peer s current key
pair for the PKC it wishes to revoke. Whether the actual revocation
request transaction occurs directly with the PKI or is first sent to
Admin who proxies or forwards the request to the PKI is a matter of
implementation.

The Admin MUST be able to initiate revocation for any PKC issued
under a template it controls. The Admin will identify itself to the
PKI by use of its own PKC; it MUST sign any revocation request to the
PKI with the private key from its own PKC. The PKI MUST have the
ability to configure Admin(s) with revocation authority, as
identified by its PKC. Any PKC authorizations must specify if said
PKC may be revoked by the Admin (see section 3.2.3.2 for more
details).

The profile MUST identify the one protocol or transaction within a
protocol to be used for both Peer and Admin initiated revocations.

The profile MUST identify the size of CRL the client will be prepared
to support.

Below are guidelines for revocation in specific transactions:

  - AFTER RENEW, BEFORE EXPIRATION: The PKI MUST be responsible for
    the PKC revocation during a renew transaction. PKI MUST revoke
    the PKC after receiving the confirm notification from the Peer,
    and before sending the confirm-ack to the Peer. The Peer MUST
    NOT revoke its own PKC in this case.

  - AFTER UPDATE, BEFORE EXPIRATION: The PKI MUST be responsible for
    the PKC revocation during an update transaction. PKI MUST revoke
    the PKC after receiving the confirm notification from the Peer,
    and before sending the confirm-ack to the Peer. The Peer MUST
    NOT revoke its own PKC in this case.

3.6 Repositories

This section refers to the [R] elements labeled in Figure 3.

3.6.1 Lookups

The PKI System SHOULD be built so that lookups resolve directly and
completely at the URL indicated in a CDP or AIA. The PKI SHOULD be
built such that URL contents do not contain referrals to other hosts
or URLs, as such referral lookups will increase the time to complete
the IKE negotiation, and can cause implementations to timeout.

CDP MUST be flagged as required in the authorization request. The
method MUST also be specified: the HTTP method MUST be method; the
LDAP method MAY be supported.

The complete hierarchical PKC chain (except the trust anchor) MUST be
able to be searched in their respective repositories. The information
to accomplish these searches MUST be adequately communicated in the
PKCs sent during the IKE transaction.

All PKCs must be retrievable through a single protocol. The final
specification will identify one protocol as a "MUST", others MAY be
listed as "OPTIONAL".

The general requirements for the retrieval protocol include:

  - The protocol can be easily Firewalled (including NAT or PAT).

  - The protocol can easily perform some query against a remote
    repository on a specific ID element that was given to it in a
    standard PKC field.

Other considerations include:

  - Relative speed
  - Relative ease of administration
  - Scalability

Intermediate PKCs will be needed for the case of re-keying of the CA,
or a PKI System where multiple CAs exist.

PKCs MAY have extendedKeyusage to help identify the proper PKC for
IPsec, though the default behavior is to not use them (see 3.1.5.3).

IPsec Peers MUST be able to resolve Internet domain names and support
the mandatory repository access protocol at the time of starting up
so they can perform the PKC lookups.

   IPsec Peers should cache PKCs to reduce latency in setting up Phase
   1. Note that this is an operational issue, not an interoperability
   issue.

   The use case for accomplishing lookups when PKCs are not sent in IKE
   is a stated non-goal of the profile at this time.

3.6.2 Error Handling for Repository Lookups

   Thorough error condition descriptions and handling instructions are
   required for each transaction in the repository lookup process.
   Providing such error codes will greatly aid interoperability efforts
   between the PKI and IPsec products.

3.7 Trust

3.7.1 Trust Anchor PKC Acquisition

   The root PKC MUST arrive on the Peer via one of two methods:

   (a) Peer can get the root PKC via its secure communication with
   Admin. This requires the Peer to know less about interaction with the
   PKI.

   (b) Admin can command Peer to retrieve the root cert directly from
   the PKI. How retrieval of the root cert takes place is beyond scope,
   but is assumed to occur via an unauthenticated but confidential
   enrollment protocol.

3.7.2 Certification Path Validation

   The IPsec Peer MUST perform identity verification based on the fields
   of the PKC and parameters applicable to the VPN Security Association.
   The fields of the PKC used for verification MAY include either the
   X.500 Distinguished Name (DN) within the Subject Name, or a specific
   field within the Extension SubjectAltName (per [DOI] 4.6.2.1
   Identification Type Values). Usage descriptions for each follow.

   The Peers or a SCVP server MUST validate the certification path, as
   per RFC3280. The contents necessary in the PKC to allow this will be
   enumerated in the profile document.

   The Peer MAY have the ability to construct the certification path
   itself, however Admin MUST be able to supply Peers with the trust
   anchor and any chaining PKCs necessary. The Admin MAY ensure the

   template uses the AIA extension in PKCs as a means of facilitating
   path validation.

   DNS MUST be supported by the Peers in order to support resolving URLs
   present in CDPs and AIA extensions.

3.7.3 Revocation Checking and Status Information

   The PKI System MUST provide a mechanism whereby Peers can check the
   revocation status of PKCs that are presented to it for IKE identity.
   The mechanism should allow for access to extremely fresh revocation
   information. CRLs have been chosen as the mechanism for communicating
   this information. Operators are RECOMMENDED to refresh CRLs as often
   as logistically possible.

   A single mandatory protocol mechanism for performing CRL lookups MUST
   be specified by the final specification.

   All PKCs used in IKE MUST have cRLDistributionPoint and
   authorityInfoAccess fields populated with valid URLs. This will allow
   all recipients of the PKC to know immediately how revocation is to be
   accomplished, and where to find the revocation information. The AIA
   is needed in an environment where multiple layers of CAs exist and
   for the case of a CA key roll-over.

   IPsec Systems have an OPTION to turn off revocation checking. Such
   may be desired when the two Peers are communicating over a network
   without access to the CRL service, such as at a trade show, in a lab,
   or in a demo environment. If revocation checking is OFF, the
   implementation MUST proceed to use the PKC as valid identity in the
   exchange and need not perform any check.

   If the revocation of a PKC is used as the only means of deactivation
   of access authorization for the Peer (or user), then the speed of
   deactivation will be as rapid as the refresh rate of the CRL issued
   and published by the PKI. If more immediate deactivation of access is
   required than the CRL refreshing can provide, then another mechanism
   for authorization that provides more immediate access deactivation
   should be layered into the VPN deployment. Such a second mechanism is
   out of the scope of this profile. (Examples are Xauth, L2TP s
   authentication, etc.).

3.7.4 Error Handling in Revocation Checking and Certificate Path
           Validation

   Thorough error condition descriptions and handling instructions are
   required for each transaction in the revocation checking and path

validation process. Providing such error codes will greatly aid
interoperability efforts between the PKI and IPsec products.

4 Security Considerations

This requirements document does not specify a concrete solution, and
as such has no system-related security considerations per se.
However, the intent of the PKI4IPSEC WG was to profile and use
concrete protocols for certificate management (e.g., CMC, CMS,
CRMF). The individual security considerations of these protocols
should be carefully considered in the profiling effort.

In addition, this document allows significant flexibility in the
allocation of functions between the roles of Peer and Admin. This
functional allocation is crucial both to achieving successful
deployment, and to maintaining the integrity of the PKI enrollment
and management processes. However, much of the responsibility for
this allocation necessarily falls to product implementers and system
operators through the selection of applicable use cases and
development of security policy constraints. These factors must be
carefully considered to ensure the security of PKI4IPSEC certificate
management.

5 IANA Considerations

There are no known numbers which IANA will need to manage.

6 References

6.1 Normative References

None.

6.2 Informative References

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

[CERTPROFILE] Housley, R., et. al. "Internet X.509 Public Key
Infrastructure Certificate and Certificate Revocation List (CRL)
Profile", RFC 3280, April 2002.

[DOI] Piper, D., "Internet IP Security Domain of Interpretation for
ISAKMP", RFC 2407, November 1998.

[FRAME] Chokhani, S., Ford, W., Sabett, R., Merrill, C., Wu. S.,
"Internet X.509 Public Key Infrastructure: Certificate Policy and
Certificate Practices Framework", RFC 3647, November 2003.

[IKECERTPROFILE] Korver, B., The Internet IP Security PKI Profile
of IKEv1/ISAKMP, IKEv2, and PKIX ,draft-ietf-pki4ipsec-ikecert-
profile-11, 25 September 2006.

7 Acknowledgements

Editor s Address

    Chris Bonatti
    IECA, Inc.
    Bonattic(at)ieca.com

    Sean Turner
    IECA, Inc.
    Turners(at)ieca.com

    Gregory M. Lebovitz
    gregory.ietf(at)gmail.com

Intellectual Property Statement

   The IETF takes no position regarding the validity or scope of any
   Intellectual Property Rights or other rights that might be claimed
   to pertain to the implementation or use of the technology described
   in this document or the extent to which any license under such
   rights might or might not be available; nor does it represent that
   it has made any independent effort to identify any such rights.
   Information on the procedures with respect to rights in RFC
   documents can be found in BCP 78 and BCP 79.

   Copies of IPR disclosures made to the IETF Secretariat and any
   assurances of licenses to be made available, or the result of an
   attempt made to obtain a general license or permission for the use
   of such proprietary rights by implementers or users of this
   specification can be obtained from the IETF on-line IPR repository
   at http://www.ietf.org/ipr.

   The IETF invites any interested party to bring to its attention any
   copyrights, patents or patent applications, or other proprietary
   rights that may cover technology that may be required to implement
   this standard. Please address the information to the IETF at ietf-
   ipr@ietf.org.

Disclaimer of Validity

   This document and the information contained herein are provided on
   an "AS IS" basis and THE CONTRIBUTOR, THE ORGANIZATION HE/SHE
   REPRESENTS OR IS SPONSORED BY (IF ANY), THE INTERNET SOCIETY AND THE
   INTERNET ENGINEERING TASK FORCE DISCLAIM ALL WARRANTIES, EXPRESS OR
   IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF
   THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED
   WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.