PKI4IPSEC Working Group
Internet Draft
Draft-ietf-pki4ipsec-profile-reqts-01.txt
July 19, 2004
Expires January 19, 2005

Chris Bonatti, IECA
Sean Turner, IECA
Gregory Lebovitz, Netscreen

Requirements for an IPsec Certificate Management Profile

Status of this Memo

Abstract

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

1 Introduction

This document enumerates requirements for PKC management interaction among different IPsec VPN products and PKI products in order to better enable large scale, PKI-supported IPsec VPN deployments. Requirements for both the IPsec and the PKI products are discussed. The goal is to create a set of requirements from which a profile document will be derived. The specification will clarify the transactions necessary between the VPN System and the PKI System that enable the deployment of easily manageable, easily scalable VPNs. When implemented, the specification will enable improved interoperability between IPsec and PKI products. 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.

Within IPsec VPNs, the PKI supports authentication of IPSec Peers through digital signatures during security association establishment using IKE. The protocol and PKI operational usages are considered in order to define a common, single set of methods (which forces interoperability) between PKI Systems and VPN Systems for large-scale deployments. The requirements address the entire lifecycle for PKI usage within IPsec transactions: pre-authorization of PKC issuance, enrollment process (PKC request and retrieval), PKC renewals and changes, revocation, validation and repository lookups. They enable a VPN Operator to:

- Authorize individual or batches of PKC issuances based on locally defined criteria, and do so from the VPN Administration point.

- Provision PKI-based user or machine identity to IPsec Peers, on a large scale. Provision means the IPsec Peer ends up with a valid public and private key pair and PKC based on the IETF Public Key

Infrastructure X.509 (PKIX) PKC profile from [CERTPROFILE]. These are used in the IKE negotiation for tunnel setup.

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

- Establish automatic renewal for PKCs, or changes.

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

The desired outcome is that both IPSec and PKI vendors create interoperable products to enable such scalable deployments, and do so as quickly as possible. For example, an VPN Operator should be able to use any conforming IPsec implementation of the certificate management profile with any conforming PKI vendor's implementation to perform the VPN rollout and management as described below.

The certificate management profile will also clarify and constrain existing PKIX and IPsec standards and protocols for easier understanding and the limiting of complexity in deployment. Some new elements are identified that may require either a new protocol, or changes or extensions to an existing protocol, especially in the area of bulk authorization for PKC issuance. The document introduces the idea of a VPN Administration function (Admin) within the VPN System. This VPN Administration function bears great responsibility for the task of managing pre-authorization for PKC issuance and of distributing the results between the VPN System and the PKI System.

1.1 Scope

The solution described in this document focuses on the requirements for the interaction between the VPN Systems and the PKI Systems. The internals of the operation of these systems are beyond scope.

The solution focuses on the needs of large-scale rollouts, i.e. VPNs including hundereds or thousands of managed VPN gateways or VPN remote access clients. 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.

There is no intention to discuss all or other PKI issues here. The scope is limited to requirements for easing and enabling scalable IPsec with PKI deployments.

The requirements strive to meet eighty percent of the market needs for large-scale deployments. 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.

1.2 Non-Goals

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

The specification for the communication method and transactions between VPN Administration function and IPSec Peers is up to vendor implementation and therefore is not expected to be included in the certificate management profile. Such a protocol may be standardized at a later date to enable interoperability between VPN Administration function stations and IPsec Peers from different vendors, but is far beyond the scope of this current effort, and will be considered

opaque by the certificate management profile.

1.3 Definitions

VPN System
The VPN System is comprised of the VPN Admininistration 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 and issue PKCs and provide revocation information about those 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.

IPsec Peer (Gateway or Client)
For the purposes of this document, an IPsec Peer, or simply "Peer", is any IPsec System that communicates IKE and IPsec to another Peer in order to create a secure tunnel 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 IPsec System can pass traffic with no IPsec protection, and can add IPsec protection to chosen traffic streams.

(VPN) Admin

The function of the VPN System that manages and distributes policy to Peers and who interacts with the PKI System to define policy for PKC provisioning for the VPN connections. See Section 2.1.1 below for more details.

End Entity
An end entity is the entity or subject that a PKC exists to authenticate. The end entity is the one entity that will finally use a private key associated with a PKC to sign data. In this document, the end entity is also an IPsec Peer.

Community Realm
A community realm is the set of IPsec Peers and VPN Administration function that operate under a common policy, and PKI authorizations.

PKC Renewal
The acquisition of a new PKC (often accompanied by a new key) due to the expiration of an existing PKC. Renewal occurs prior to the expiration of the existing PKC to avoid any connection outages.

PKC Change
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 or identifying information of the end entity.

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 trusted by one or more users to create and assign 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 time stamped list identifying revoked PKCs that is signed by a CA and made freely available in a public repository. Peers

retrieve the CRL to verify that a PKC being presented to them as identity in an IKE transaction has not been revoked.

CRL Distribution Point (CDP)
The CDP extension in a PKC identifies the location from which end entities should retrieve CRLs to perform local validity checking.

Authority Info Access (AIA)
The AIA extension in a PKC 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

Though this document is not an Internet Draft, we use the convention that 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 architecture picture is presented. Last, the process of the interaction between the two Systems for large-scale deployment is described.

2.1 VPN System

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

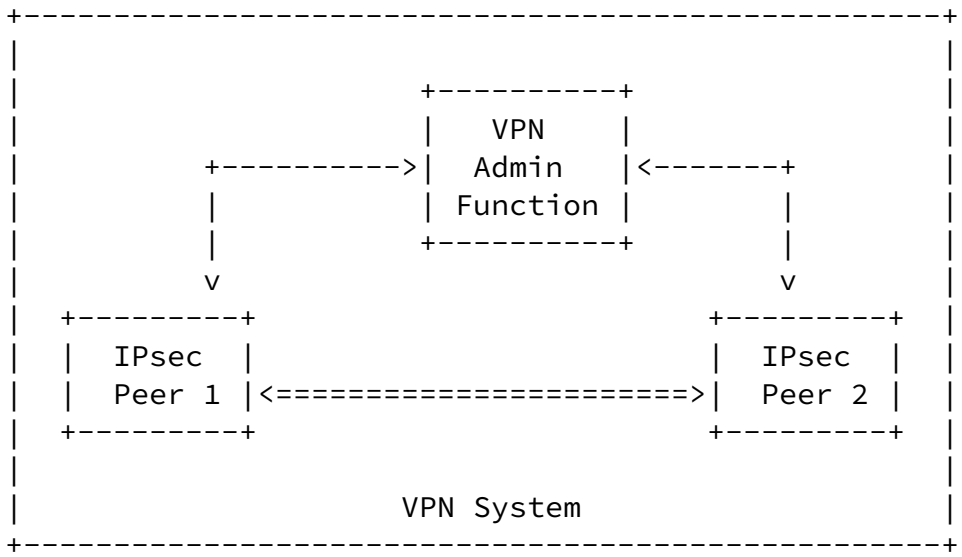


Figure 1: VPN System

2.1.1 IPsec Peer(s)

The Peers are two entities between which the Operator requires an IPsec tunnel establishment. Two Peers are shown in Figure 1, but implementations MAY support an actual number in the hundreds or thousands. The Peers could be either gateway-to-gateway, remote-access-host-to-gateway, or a mix of both. The Peers authenticate themselves in the IKE negotiation using digital signatures through 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 solution. The Admin is a centralized function. It defines the VPN System policy and informs the PKI and Peers how it wants each to enforce that policy. One main role defined here is that Admin specifies to the PKI the contents and use parameters of the credentials the PKI will issue, or at least references a template or policy-set for a Peer or set of Peers. In this way Admin MAY perform many RA-like functions, for example authorization of PKC issuance and revocation.

It is important to note that, within this document, 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 may be performed on a stand-alone work station, on a gateway, on an administration software component, etc. It is also possible for the Admin function to be one in the same as the gateway or client device or software. They are represented in the architectural diagram below as different functions, but they need not be different physical entities. As such, 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 will be the place where Certificate Policy (CP) (see RFC 3647) for use in the VPN is defined, not the PKI. In VPN Systems the Operator chooses to strengthen the VPN by using PKI; PKI is a bolt-on to the VPN System. The PKC characteristics and contents are a function of the local security policy the VPN serves to enforce. Therefore the Operator will configure policy and contents for PKCs in the Admin, and apply those templates to groups of IPsec Peers.

- It will interact directly with the PKI System to initiate authorization for end entity PKCs by sending the parameters and contents for those PKCs, or by referring to a template or policy-set on the PKI. (Such templates would likely have been created in conjunction with the Operator.) It will receive back from the PKI identification values and authorization codes to be used in the PKC requests for each of the pre-authorized PKCs.

- It will deliver instructions to the IPsec Peers, and the Peers will carry out those instructions. An example of such an instruction is an IKE policy configuration. Therefore, the communication mechanism between the Admin and the IPsec Peers MUST be private, authenticated and employ integrity checks. The contents of some such instructions will be defined below. However, the communication mechanism will be handled completely within the VPN System and is out of the scope of this document (see Scope, Section 1.1 above).

The Admin MUST be reachable by the Peers. Most implementations will meet this requirement by ensuring the Peer can connect to the Admin from anywhere on the network or Internet. However, communication between the Admin and Peer may not necessarily be "on-line". It may, in some environments, be "moving media," i.e. the configuration or data may be loaded on to a floppy disk or other media and physically moved to the IPsec Peer. This reality should be considered when requirements are defined, and when supporting networks are architected.

2.2 PKI System

The PKI System, as depicted in Figure 2, may be set up and operated

by the Operator (in-house), may be provided by third party PKI providers to which connectivity is available at the time of provisioning (managed PKI service), or may be integrated with the VPN product.

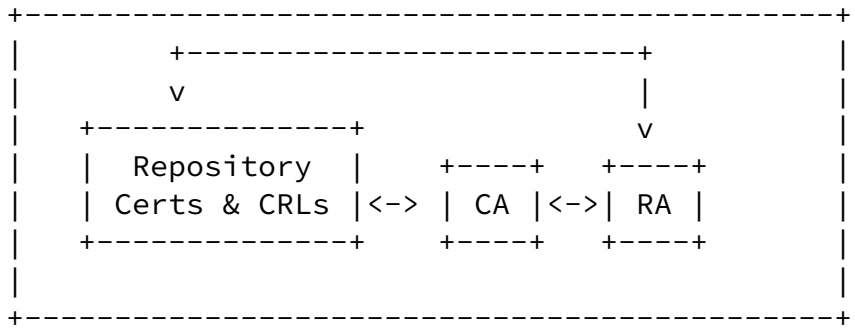


Figure 2: PKI System

This framework assumes that all components of the VPN will obtain PKCs from a single PKI community. An IPsec Peer MAY 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.

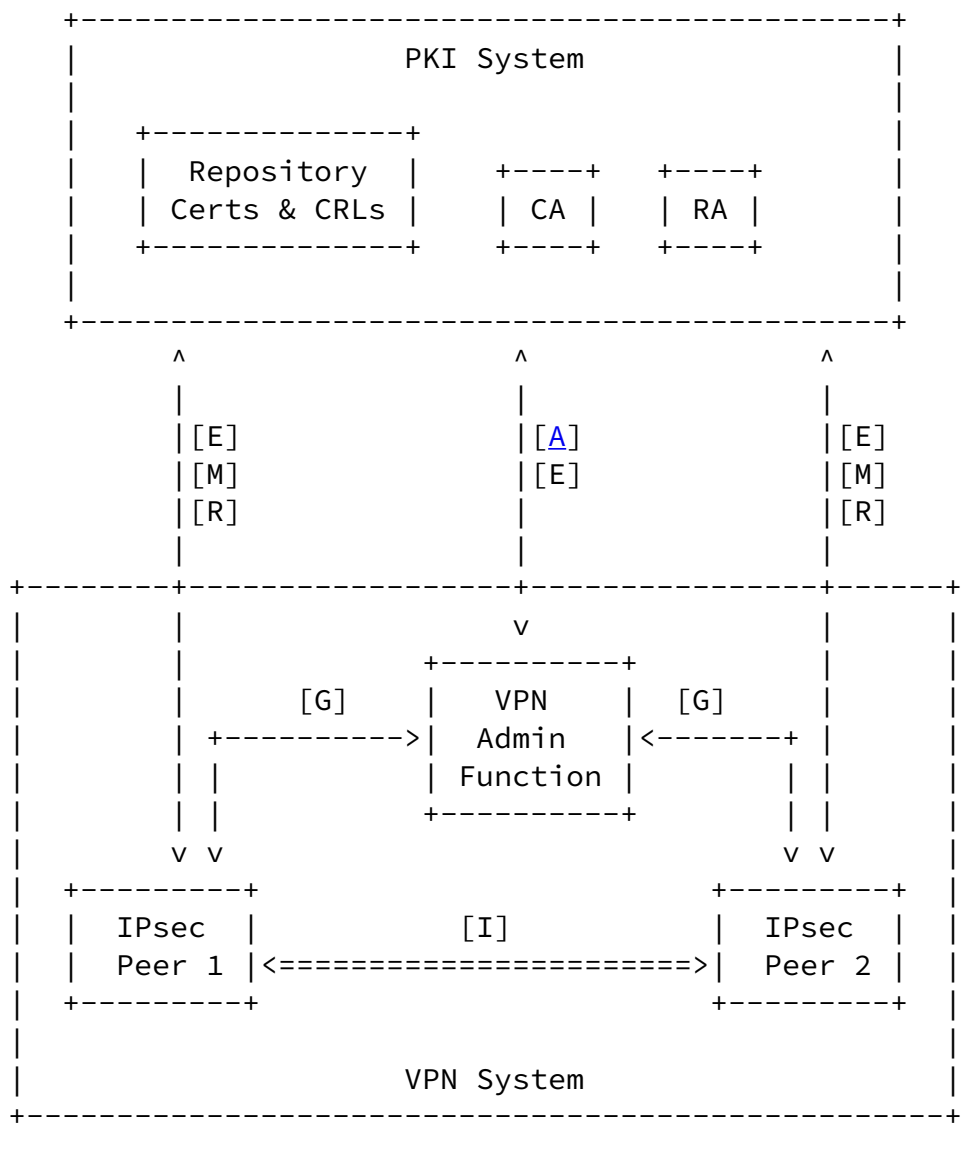
The PKI System will contain a mechanism for handling Admin's authorization requests and PKC enrollments. These mechanisms are referred to as the RA. The PKI System contains a Repository used by the Peers to look up each other's PKCs. Last, the PKI System contains the core function of a CA that uses a public and private key pair and signs PKCs.

The PKI System SHOULD be built so that lookups resolve directly and completely at the URL indicated in a CDP, or AIA. The PKI ought to 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.

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. Figure 4 (below) illustrates the information flow associated with the steps initial PKC generation relative to the architecture diagram. Figure 5 (below) illustrates the information flow associated with the steps PKC renewal relative to the architecture diagram. Figure 6 (below) illustrates the information flow associated with the steps PKC renewal relative to the architecture diagram. For simplicity only the steps associated with IPsec Peer 1 are shown.



[A] = Authorization of PKC issuance and revocation
[G] = Generation of public and private key pair, PKC request
[E] = Enrollment (request and retrieval)
[I] = IKE and IPsec communication
[M] = Maintenance: validation, revocation, repository lookups
[R] = Renewal (and changes)

Figure 3. Architectural Framework for VPN-PKI Interaction

2.3.1 New PKC

The steps of the VPN-PKI interaction are summarized here for generating a new PKC. The letters refer to Figure 3. The numbers refer to Figure 4. The detailed requirements are described below in Section 3. Note that there are a number of architectul options available and that the most common architecture is depecited in Figure 4; IPsec Peer generated Keys and IPsec Peer generated PKC Request. Other architectural options are discussed in Section 3.

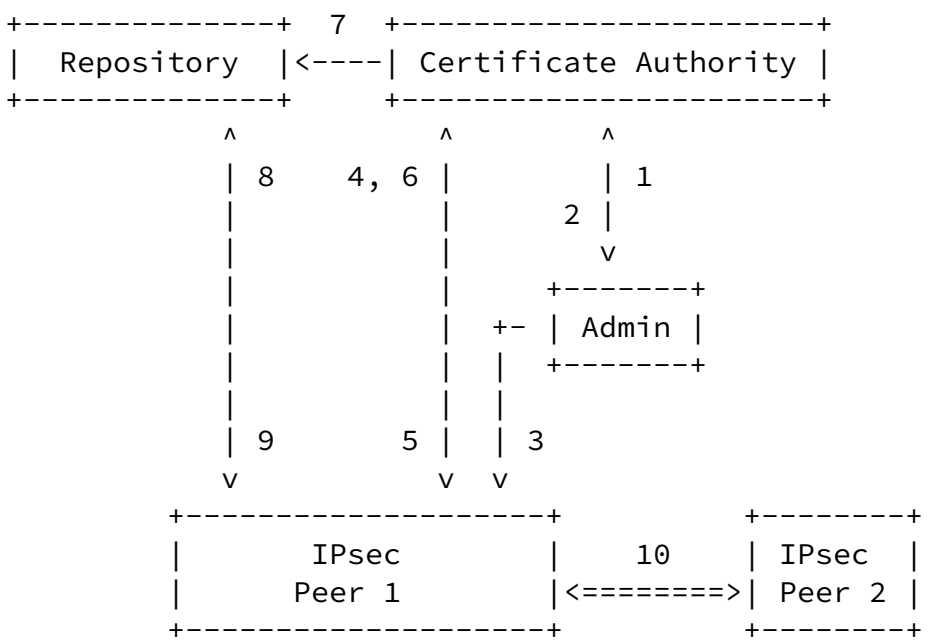


Figure 4. VPN-PKI Interaction Steps:
IPsec Peer Generates Keys and PKC Request,
Enrolls Directly with PKI

1) Authorization [A]. Admin sends a list of IDs and PKC contents for the PKI System to authorize enrollment. The PKI returns a list of unique identifiers and one-time tokens to be used for the enrollment of each PKC. Other PKC usage policy is also set at this time, for example parameters for renewals or changes, key lengths, etc. The amount of information that the Admin communicates to the PKI about how it wants the PKCs built could be very small, perhaps just a reference to a template already existing in the PKI System. Likewise it could be very large, with several fields being specified along with their contents. [EDITOR'S NOTE: We need some work on this line of thought.]

2) Authorization Response [A]. The PKI System acknowledges the authorizations provided in (1). Response may indicate success or failure for any particular authorization.

3) Generate Keys and PKC Request [G]. The Admin communicates with the Peer to either give it information so that it can generate a public and private key pair and PKC request and send the request directly to the PKI.

4) Enrollment [E]. The IPsec Peer requests a PKC from the PKI, providing the generated public key. The IPsec Peer generates the key pair and PKC request.

5) Enrollment Response [E]. The PKI responds to the enrollment request sent in (4), providing either the new PKC that was generated or a suitable error indication.

6) Enrollment Confirmation. Peer must positively acknowledge receipt of new PKC.

7) PKC Posting. The newly-generated PKC for IPsec Peer 1 is posted to the repository.

8) Maintenance [M]. The IPsec Peer accesses the PKI to support look-up of PKCs for other IPsec Peers, certification path validation, and revocation checking. This step consists of sending requests for specific PKCs or CRLs, or requests for the PKI System to perform validation checks.

9) Maintenance Response [M]. The PKI responds to the maintenance request sent in (7), providing either the requested PKC or CRL, indicating the validity status of a PKC, or indicating an error condition.

10) IKE/IPsec Communication [I]. The Peers communicate authenticated by the PKCs they received from the PKI.

2.3.2 Renewal PKC

The steps of the VPN-PKI interaction are summarized here for rewal PKCs. The letters refer to Figure 3. The numbers refer to Figure 5. The detailed requirements are described below in Section 3. Note that there are a number of architectul options available and that the most common architecture is depecited in Figure 4; IPsec Peer generated Keys and IPsec Peer generated PKC Request. Other architectural options are discussed in Section 3.

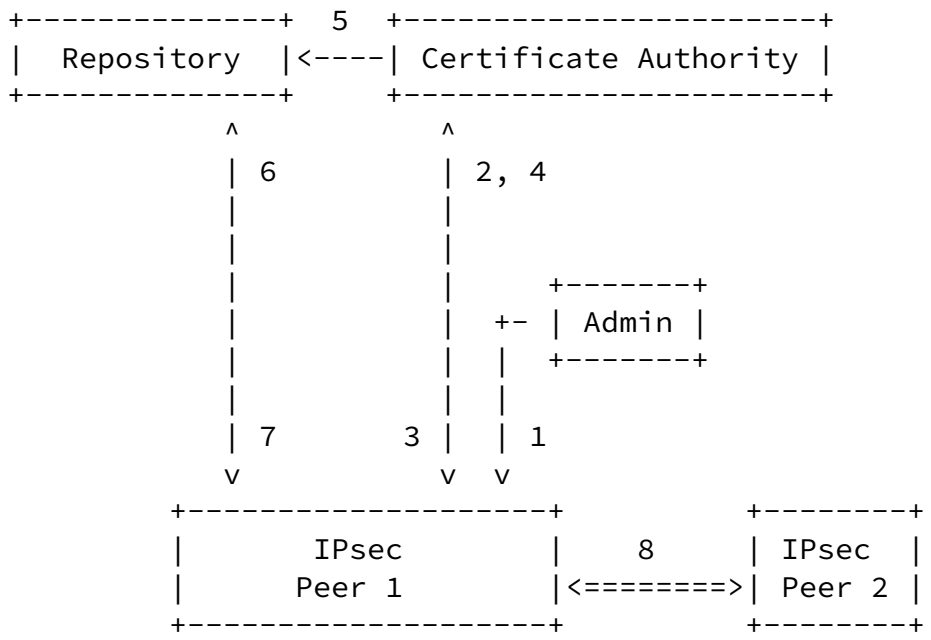


Figure 5. VPN-PKI Interaction Steps: Renewal by IPsec Peer 1

1) Rekey or Renewal Initiation. The Admin communicates renewal or change instructions to the Peers. Renewal may also be signalled to the PKI (not shown), particularly if authorization changes are necessary. Initiation of this process by the Admin enables IPsec Peers to automatically generate renewal or change requests as needed with minimal user burden, and for those requests to be immediately granted by the PKI System.

2) Renewals and Changes [R]. The IPsec Peer requests renewal or

change of an existing PKC. Rekey MAY also occur depending upon policy constraints. The renewal or change request will either be provided in (10) above, or will be generated by the IPsec Peer.

3) Renewal/Change Response [R]. The PKI responds to the renewal or change request sent in (11), providing either the new PKC that was generated or a suitable error indication.

4) Enrollment Confirmation. Peer must positively acknowledge receipt of new PKC.

5) PKC Posting. The newly-generated PKC for IPsec Peer 1 is posted to the repository.

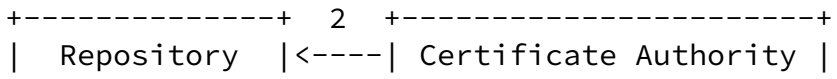
6) Maintenance [M]. The IPsec Peer accesses the PKI to support look-up of PKCs for other IPsec Peers, certification path validation, and revocation checking. This step consists of sending requests for specific PKCs or CRLs, or requests for the PKI System to perform validation checks.

7) Maintenance Response [M]. The PKI responds to the maintenance request sent in (7), providing either the requested PKC or CRL, indicating the validity status of a PKC, or indicating an error condition.

8) IKE/IPsec Communication [I]. The Peers communicate authenticated by the PKCs they received from the PKI.

2.3.3 Revocation

The steps of the VPN-PKI interaction are summarized here for generating a new PKC. The letters refer to Figure 3. The numbers refer to Figure 6. The detailed requirements are described below in Section 3.



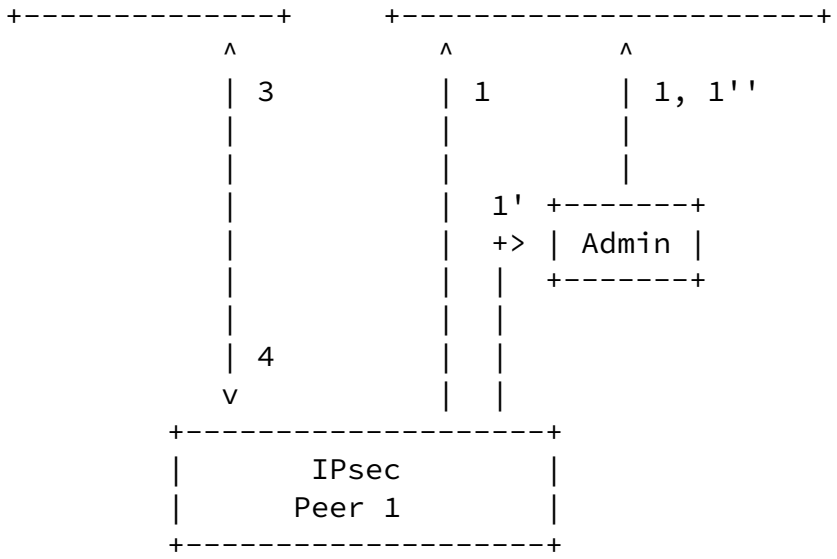


Figure 6. VPN-PKI Interaction Steps: Revocation

1) Revocation. The IPsec Peer or Admin requests revocation of IPsec Peer 1's PKC directly from the PKI.

1') Revocation. The IPsec Peer requests revocation of their PKC through admin.

1'') Revocation. The Admin forwards IPsec Peer 1's PKC revocation request to PKI.

2) CRL Posting. The newly-generated CRL revoking IPsec Peer 1's PKC is posted to the repository.

3) Maintenance [M]. The IPsec Peer accesses the PKI to support look-up of CRL.

4) Maintenance Response [M]. The PKI responds to the maintenance request sent in (3), providing either the requested CRL, indicating the validity status of a PKC, or indicating an error condition.

3 Requirements

3.1 General Requirements

3.1.1 One Protocol

This target profile will call for ONE PROTOCOL or ONE USE PROFILE for each main element of the requirements. It is a specific goal to avoid multiple protocols or profiles to solve the same requirement whenever possible so as 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.

Conforming implementations MUST implement the ONE PROTOCOL or ONE USE PROFILE that is specified for a given requirement.

3.1.2 Secure Transactions

The target profile will specify the transactions for certificate management between VPN and PKI Systems and their components, as needed to ease large scale VPN deployment and management. Specifically, Admin and PKI will transmit between themselves policy details, identities, and keys. As such, the method of communication for these transactions MUST be secured in a manner that ensures privacy, authentication, message data integrity and non-repudiation. This method will require that mutual trust be established between the PKI and the Admin.

[EDITOR'S NOTE: Need to perhaps elaborate on "policy details" above.]

3.1.3 PKI Availability

Central availability is required initially for authorization transactions between the PKI and Admin. Further availability will be required in most cases, but is a decision point for the Operator. Most requirements and scenarios below assume on-line availability of the PKI and Admin for the life of the VPN.

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 need not even be aware that PKI is in use. First time connections need 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 Error Handling

The PKC transaction protocol for the PKI and VPN System transactions

MUST specify error handling for each transaction. Thorough error condition descriptions and handling instructions will greatly aid interoperability efforts between the PKI and IPsec products.

3.2 Authorization Transactions

3.2.1 Bulk Authorization

Bulk authoriztion 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 field (more is also acceptable) MUST differ per subject. Because the authorization may occur before any keys have been generated, the only way to determine one authorization from another for the purpose of issuing unique identifiers is by having at least one field differ.

The authorization MAY occur prior to the event of a PKC enrollment request (in which case it is a "pre-authorization"), or within the same connection.

3.2.2 Protocol Preferences for Authorization

A single connection per multiple transactions. It is preferred that the setup for all subjects in an authorization batch occurs in one single connection to the RA/CA, with the number of subjects being one or greater. Implementations should be able to handle tens of thousands at a time.

ONE protocol must be specified for these Admin to RA/CA interaction.

The PKI responds to the Admin station with Authorization identifiers (maybe serial numbers or such) and a corresponding pre-authorization key (not to be confused with the public and private key pair) for each identifier.

It is preferred that the transport used to carry the pre-authorization be reliable (TCP).

The protocol should be as lightweight as possible.

A method for securing the communication between the Admin and the PKI MUST be defined, including privacy, authorization, and integrity.

PKCs and authorization of the Admin may need to be initialized by physical rather than on-line means.

3.2.3 Admin Authorization Requests to PKI

3.2.3.1 Specifying Fields within the PKC

The VPN may send the PKI System the set of PKC contents that make up a PKC template that it wants the PKI to use. 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." Likewise, such a template

may have already been defined on the PKI System, and the Admin may simply reference it.

In the former case, the elements that the Admin MAY send to the PKI to authorize the eventual creation of PKCs include:

- DN fields
- Any number of locally defined CNs with their contents [EDITOR'S NOTE: this is difficult to do. We may need to say just one CN.]
- Validation Period of the PKC
- Renewal parameters (i.e., N% of validity period, and PKC overlap duration in N [EDITOR'S NOTE: Should consider other factors. Measurement? Minutes? Hours? Percentage?], or just let it expire)
- Any of SubjAltName fields
- Key type
- Key length
- Any of the extension fields (Key usage, extended key usage, Policy constraints, etc.)
- Require a CDP be filled in by the PKI in issuance. The specification should define who will handle the CDP contents. Suggest the PKI, not Admin, but further research is needed.

3.2.3.2 Authorizations for Renewal and Change

When the Admin sends its authorization request information it MUST also send information to the PKI about the local policy regarding renewal and changes. These are:

- Admin MUST specify if automatic renewals are allowed, that is, the Admin is presently authorizing the PKI to process a future renewal for the specified end entity PKC.

- Admin MUST specify if any changes are allowed, that is, the Admin is presently authorizing the PKI to accept a future request for a new PKC creation with some element of the Subject or SubjectAltName changed.

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

- Whether or not a new key must be used for the new PKC.

- Who can renew, i.e. can only the admin send a renewal request or can the end entity 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.

- Length of time (if ever) after PKI receives end entity Peer confirmation (see 3.4.8 and 3.6.1 below) that the old PKC is revoked, and removed from repository.

If change request is authorized, the Admin MUST further specify:

- The fields in the Subject and SubjectAltName that are changeable

- The entity that can send the change request, i.e. only the Admin, only the end entity, or either.

- Length of time (if ever) after PKI receives end entity Peer confirmation (see 3.6.1 below) that the old PKC is revoked, and removed from repository.

3.2.3.3 Other Authorization Elements

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

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

Ability to communicate the Community Realm for the PKC to the PKI. Community Realm is an important component in provisioning that allows the Admin to specify for the Peer various elements of the PKC's contents that the PKI will fill in, and are not defined by the Admin. It may be used to specify various local policy definitions. It also will be used to label different groups to have different CRLs (for example small CRLs with only gateways in the listing for use by Remote Access Peers, or large CRLs with all Remote Access Peers and gateways to be used by the Gateways). There will be a need for an import and export for easily synchronizing the Community Realm lists between the Admin and PKI System.

The Protocol should consider what happens when Admin requested information conflicts with PKI settings such that the Admin request cannot be issued as requested. (Ex: 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 Cancel Capability

Admin can send a cancel authorization message to PKI. [EDITOR'S NOTE: Should the Peer be able to send a cancel message as well?] Admin MUST provide the authorization ID and code 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 with the Admin (a Cancel, Cancel ACK, ACK type of a process is required to cover a lost connections scenario), the PKI will accept another Authorization request with the exact same contents as the canceled one. The PKI MUST NOT accept a second authorization request for the same identity [EDITOR'S NOTE: How do we decide what defines "identity"?] if one already exists.

3.2.5 PKI response to Admin

If the authorization is acceptable, the PKI will respond to the Admin with a unique identifier per subject authorization required and a one-time-authorization key per authorization ID. Strongly recommend the one-time-authorization key be unique per authorization ID. The more randomness that can be achieved in the relationship between an identifier and its key the better. The key MUST be in ASCII format to avoid incompatibilities that may occur due to international characters.

All the contents of the PKC that it intends to issue will be returned to the Admin. This will allow the Admin to perform an "operational test" to verify that the issued PKCs will meet its requirements.

For any request, the PKI cannot change any of the specified values in request within its response. We need to prevent a change in PKC contents that may occur due to a change in PKI configuration right in the middle of a batch pre-authorization request.

[EDITOR'S NOTE: what if the Admin sends a parameter that the PKI cannot fulfil, i.e. the parameter contradicts PKI policy? Would need to return an error code and description and refuse to authorize the enrollment.]

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

A method is needed 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.

3.2.6 Error Handling for Authorization Transactions

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

3.3 Key Generation and PKC Request Construction

Once the PKI System has responded with authorization identifiers and

keys, 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 MAY occur at one of two places, depending on local requirements: at the IPsec Peer or at the Admin. The PKC constructions MAY occur at either the IPsec Peer or a combination of the Peer and the Admin.

[EDITOR's NOTE: Should we have different arrow diagrams for each option? Option 1 is already depicted in Figure 4. Should we show the differences amongst the other three?]

3.3.1 IPsec Peer Generates Key Pair and Constructs Request

This case will be used most often in the field. This is the most secure method for keying; the keys are generated on the end entity and never leave the end entity.

The Admin will send the authorization identifier and authorization key to the end entity, the IPsec Peer. The Admin will also send any other parameters needed by the Peer to generate the PKC request, including key type and size. Recall that the mechanism for how this information is communicated from the Admin to the Peer is opaque.

Receiving the command and the necessary information from the Admin, the Peer will proceed to generate the key pair and construct the PKC request.

3.3.2 IPsec Peer Generates Key Pair, Admin Constructs Request

In this case, the Admin sends a command to the Peer to generate the key pair. The Admin then constructs the PKC request on behalf of the Peer, except for the signing. It sends the construction to the Peer for signing, and the Peer returns the signed request construction back to the Admin. The Admin then proceeds to enroll on behalf of the client.

The advantage of this solution is that the private key never leaves the IPsec Peer, but limits the amount the Peer must know and do regarding PKC generation.

3.3.3 Admin Generates Key Pair and Constructs Request

The use case exists for deployments where end entities 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 end entity station. If key escrow is performed then the exact requirements and procedures for it are beyond the scope of this document.

The Admin will generate the key pair, construct the PKC request, and enroll on behalf of the Peer. Once the PKC has been retrieved, the keys and PKC will be sent to the Peer using a secure method. The nature of this secure method is beyond the scope of this document.

Performing a separate pre-authorization step is still of value even though the Admin is the also performing the key generation. The Community Realm, Subject fields, SubjectAlt 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, we simply use the pre-authorize schema again. This also allows for the feature of role-based administration, where Operator1 is the only one allowed to have the Admin function pre-authorize PKCs, but Operator2 is the one doing batch enrollments and VPN device configurations.

3.3.4 PKI Generates Key Pair and Passes to Peer via Admin

TBD - [EDITOR'S NOTE: There is another use case here: PKI generates the key pair AND the PKC and simply hands it down to the Admin for installation into the Peer. This is, in all likelihood, the easiest way to deploy Certs, though sacrafices a bit in security. Do we just specify PKCS12 and try to create some requirements for how the Admin will say, "I need a cert for NNNNN," and how PKI will respond with the PKCS12?]

3.3.5 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.

3.3.6 Error Handling for Key Generation and Request Construction

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

3.4 Enrollment (Sending Request and PKC Retrieval)

Regardless of where the keys were generated and the PKC request constructed, an enrollment process will need to occur to request a PKC creation from the PKI and to retrieve that PKC.

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 (as seen below in sections 3.4.5 through 3.4.7).

3.4.1 One protocol

One protocol MUST be specified for both request and retrieval.

3.4.2 On-line protocol

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

3.4.3 Single Connection with Immediate Response

Request and retrieval MUST be able to occur in one on-line connection between the end entity and the PKI (RA/CA).

The end entity sends the request, attaching the Authorization identifier and key.

The RA/CA receives the request and uses the Authorization identifier and key to match it to the proper pre-authorization entry.

Since the contents of the PKC match, and the Authorization identifier and key are correct, the PKC is generated immediately, with no need for manual intervention or review on the PKI System before issuance.

The PKI makes the PKC available immediately for retrieval, or

possibly sends the PKC to the end entity as a response in the request or retrieval exchange.

3.4.4 Manual Approval Option

The optional capability to queue and manually approve PKC requests MUST exist within the protocol for those organizations that will not permit automation of credential issuing as described above. Likewise, polling to determine if request has been satisfied and to try to retrieve the PKC MUST exist within the protocol for those organizations that will not permit automation of credential issuing as described above.

End-entities and the PKI must disclose and agree upon which mode they will support (automated approval or manual approval) within the protocol.

3.4.5 Enrollment Method 1: Peer Enrolls to PKI Directly

The enrollment MAY occur in one of three fashions, and valid use cases exist for all three. First, and most straight forward, the Admin can instruct the IPsec Peer to execute an enrollment, telling it where to enroll, and providing any necessary parameters.

In this case the IPsec Peer only talks to the PKI after being commanded to do so by the Admin. Note that this enrollment mode is depicted in Figure 4.

3.4.6 Enrollment Method 2: IPsec Peer Enrolls to PKI 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 automatically performs 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 retrieves the PKC, it then automatically forwards it to the IPsec Peer, and the Peer can begin using it in security policy.

The communication of the request, retrieval, renewal, or change, can go directly from the end entity to the PKI, or be passed from end entity through the Admin to the PKI. In the latter case, the end

entity need not know how to do all the direct communication with the PKI; the function becomes focused in the Admin station. In either case, the format of messages should be identical regardless of who is sending the request.

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 they have 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.

The steps of the VPN-PKI interaction are summarized here for the IPSec Peer enrolling through the Admin. The letters refer to Figure 3. The numbers refer to Figure 7.

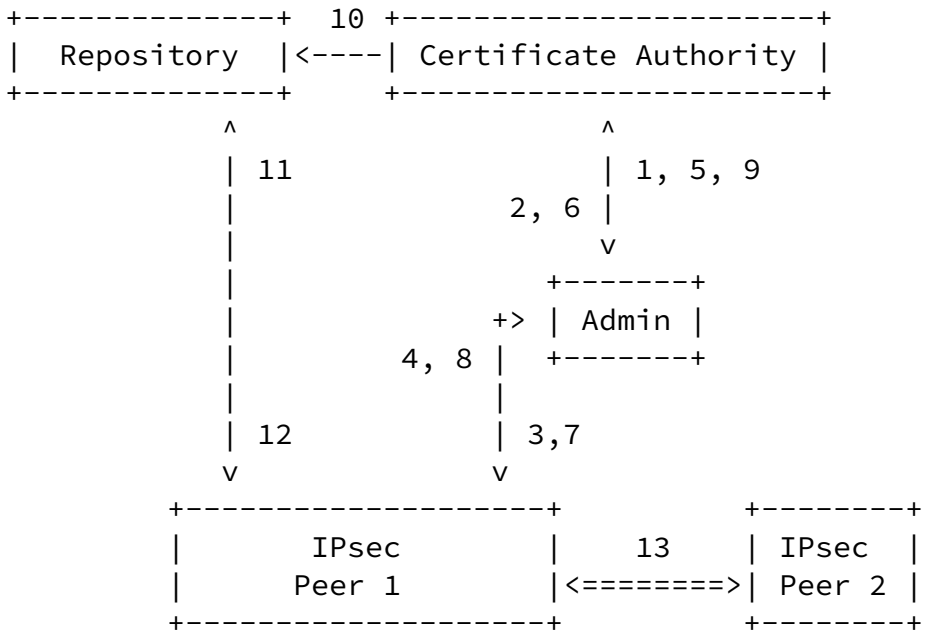


Figure 7. VPN-PKI Interaction Steps: IPsec Peer Generates Keys and PKC Request, Enrolls Through Admin

1) Authorization [A]. Admin sends a list of IDs and PKC contents for the PKI System to authorize enrollment. The PKI returns a list of unique identifiers and one-time tokens to be used for the enrollment of each PKC. Other PKC usage policy is also set at this time, for example parameters for renewals or changes, key lengths, etc. The amount of information that the Admin communicates to the PKI about how it wants the PKCs built could be very small, perhaps just a

reference to a template already existing in the PKI System. Likewise it could be very large, with several fields being specified along with their contents. [EDITOR'S NOTE: We need some work on this line of thought.]

2) Authorization Response [A]. The PKI System acknowledges the authorizations provided in (1). Response may indicate success or failure for any particular authorization.

3) Generate Keys and PKC Request [G]. The Admin communicates with the Peer to give it information so that it can generate a public and private key pair and PKC request and send the request back to the Admin.

4) Enrollment [E]. The IPsec Peer requests a PKC from the Admin, providing the generated public key.

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

6) Enrollment Response [E]. The PKI responds to the enrollment request sent in (5), providing either the new PKC that was generated or a suitable error indication.

7) Enrollment Response [E]. The Admin forwards the enrollment response back to the IPsec Peer.

8) Enrollment Confirmation. Peer must positively acknowledge receipt of new PKC back to the Admin.

9) Enrollment Confirmation. Admin forwards enrollment confirmation back to the PKI.

10) PKC Posting. The newly-generated PKC for IPsec Peer 1 is posted to the repository.

11) Maintenance [M]. The IPsec Peer accesses the PKI to support look-up of PKCs for other IPsec Peers, certification path validation, and revocation checking. This step consists of sending requests for specific PKCs or CRLs, or requests for the PKI System to perform validation checks.[EDITOR's NOTE û is the Admin going to the repository lookup for the IPsec Peer?]

12) Maintenance Response [M]. The PKI responds to the maintenance

request sent in (11), providing either the requested PKC or CRL, indicating the validity status of a PKC, or indicating an error condition.

13) IKE/IPsec Communication [I]. The Peers communicate authenticated by the PKCs they received from the PKI.

3.4.7 Enrollment Method 3: Admin Enrolls to the PKI Directly

In this instance, the Admin is performing a function similar to that of a Registration Authority (RA), as defined in [CERTPROFILE]. The Admin will have likely generated the key pair and constructed the request on behalf of the IPsec Peer. It proceeds to handle the entire enrollment directly with the PKI, and returns to the IPsec Peer the final product of a key pair and PKC. Again, the mechanism for the Peer to Admin communication is opaque.

The steps of the VPN-PKI interaction are summarized here for the Admin enrolling directly to the PKI. The letters refer to Figure 3. The numbers refer to Figure 8.

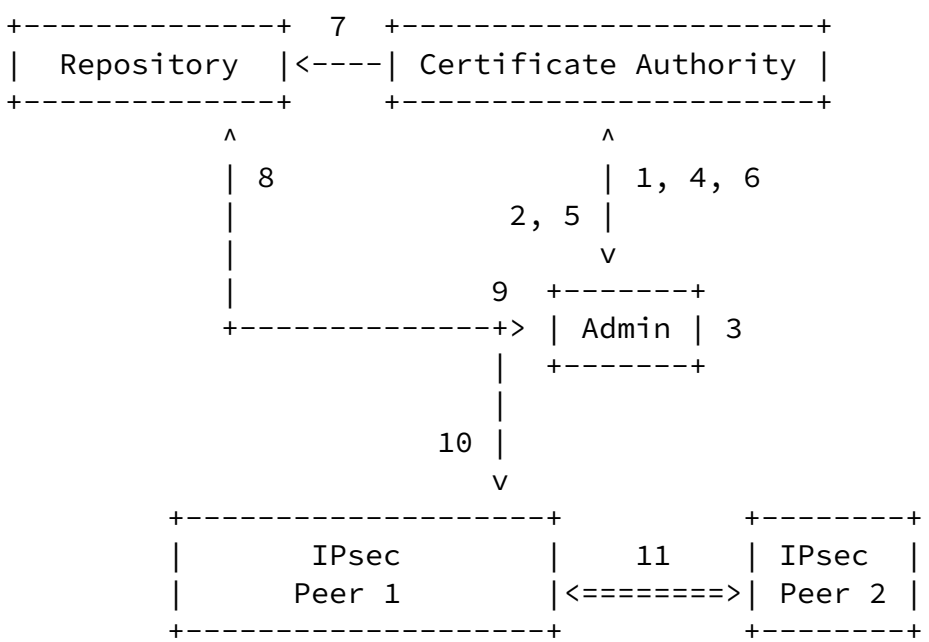


Figure 8. VPN-PKI Interaction Steps: Admin Generates Keys and PKC Request,

Admin Performs Enrollment

1) Authorization [A]. Admin sends a list of IDs and PKC contents for the PKI System to authorize enrollment. The PKI returns a list of unique identifiers and one-time tokens to be used for the enrollment of each PKC. Other PKC usage policy is also set at this time, for example parameters for renewals or changes, key lengths, etc. The amount of information that the Admin communicates to the PKI about how it wants the PKCs built could be very small, perhaps just a reference to a template already existing in the PKI System. Likewise it could be very large, with several fields being specified along with their contents. [EDITOR'S NOTE: We need some work on this line of thought.]

2) Authorization Response [A]. The PKI System acknowledges the authorizations provided in (1). Response may indicate success or failure for any particular authorization.

3) Generate Keys and PKC Request [G]. The Admin generates the public private key pair and PKC request.

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

5) Enrollment Response [E]. The PKI responds to the enrollment request sent in (4), providing either the new PKC that was generated or a suitable error indication.

6) Enrollment Confirmation. Admin must positively acknowledge receipt of new PKC back to the PKI.

7) PKC Posting. The newly-generated PKC for IPsec Peer 1 is posted to the repository.

8) Maintenance [M]. The Admin accesses the PKI to retrieve the new PKC. [EDITOR's NOTE û is the Admin going to the repository lookup for the IPsec Peer?]

9) Maintenance Response [M]. The PKI responds to the maintenance request sent in (8), providing the requested PKC, or indicating an error condition.

10) Admin sends newly generated PKC and private key to IPsec Peer.

11) IKE/IPsec Communication [I]. The Peers communicate authenticated by the PKCs they received from the PKI.

3.4.8 Enrollment Type Field

A field must exist in the request to specify the TYPE of request being made. Request types include new request, renew request, and change request (renewals and changes are discussed in detail in section 3.6). The type field is required for monitoring, logging and auditing purposes. They will help the Operator to know exactly what type of request was made so that suspicious activities, even if the request is denied, can be identified.

3.4.9 Confirmation Handshake

Any time a new PKC is issued by the PKI, a confirmation must be sent back to the PKI. This is true for first time issuances, renewals, and changes alike.

Operationally, the Peer MUST send a confirmation to the PKI verifying that the end entity 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 the end entity Peer that it may proceed using this PKC in IKE connections. The PKI MUST complete all

processing necessary to enable the end entity's operational use of the new PKC (for example, writing the PKC to the repository) before sending the confirmation acknowledgement. The PKI MUST also issue a revoke on the original PKC before sending the confirmation ACK (see section 4.X). The end entity Peer MUST NOT begin using the PKC until the PKI's confirmation acknowledgement has been received.

3.4.10 Failure Cases

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 must clarify what happens if the request and retrieval fails for some reason. The following cases will 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-key, and the PKI, seeing the ID and key, MUST send the PKC again.

3.5 PKC Profile for PKI Interaction

A PKC used for identity in IKE transactions MUST include all the X509v3 mandatory fields. It must also contain the minimal contents necessary for path validation and chaining (these items will be enumerated in the profile).

It is preferable that the PKC profiles for IPsec and certificate management were the same so that one PKC could be used for both protocols. If the profiles are inconsistent then different PKCs (and perhaps different processing requirements) might be required for certificate management transactions vs. IKE transactions. However, failure to achieve this requirement in the profile MUST NOT hold up the standardization effort.

3.5.1 Identity Usage

The IPsec Peer SHALL perform identity verification based on the fields of the PKC and parameters applicable to the VPN tunnel. 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 PKC field(s) that will be used for identity verification MUST be included in the PKC request by the Admin or the Peer. In addition to the DN, the following identity-related values may be included in the SubjectAltName:

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

While substrings of these identity values may also be present in elements of the DN, they will not be looked for in the DN, only in SubjectAltName.

3.5.2 Path Validation

The Peers must validate the certification path. 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 include the AIA extension in PKCs as a means of facilitating path validation.

DNS SHOULD be supported by the Peers in order to do certification path lookups, as well as those for revocation.

3.5.3 KeyUsage

The PKC's KeyUsage digialSignature bit [CERTPROFILE] MUST be flagged on.

[EDITOR'S NOTE: Shouldn't the non-repudiation bit also be required? It's in the stated requirements, and PKIX treats it separately. Also check whether the key exchange or key agreement bits should be required. These are employed by both CMC and IPsec.]

3.5.4 Extended Key Usage

EKU's are not required. The presence or lack of an EKU MUST NOT cause an implementation to fail an IKE connection.

Default behavior is to not check EKU. However, local security policy MAY check EKU, and if so the implementation SHOULD allow the acceptance or rejection based on the presence of each EKU. Those EKUs are defined as:

- serverAuth,
- clientAuth,

or an IKE specific EKU which are defined as one of the four currently issued IANA EKU's:

- IPsec user,

- IPsec computer,
- IPsec intermediate,
- IKE IPsec intermediate.

3.5.5 Pointer to Revocation Checking

The PKC contents must be constructed in a manner such that any Peer who hold the PKC locally will know exactly where to go and how to request the CRL.

The location and method for either a CDP or an AIA [CERTPROFILE] MUST be included in the PKC. Including such contents avoids the need to send the CRL to the Peer, and allows the receiving Peer to look up the CRL on their own.

PKCs MUST contain the full name of the CDP and AIA. Issuer-relative names are not considered sufficient.

3.6 PKC Renewals and Changes

In order to allow for continued PKC usage, a new PKC will need to be issued for an end entity before the end entity's currently held PKC expires. A renewal is defined as a new PKC issuance with the same SubjectName and SubjectAlternativeName contents as an existing PKC for the same end entity before expiration of the end entity's current PKC.

A change is defined as a new PKC issuance with an altered SubjectName or SubjectAlternativeName for the same end entity before expiration of the end entity's current PKC. Renewals and changes are variants of a PKC request scenario with unique operational and management requirements.

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 or changes. The PKI MUST support either case.

It is desired that a renew or change request contain an element that identifies the request as either type=renewal, or type=change. This

element MUST be specified in the profile. This will allow for better management, logging and auditing of certificate management.

When sending a renew or change request, the entire contents of the PKC request needs to be sent to the PKI, just as in the case of the original enrollment. Keeping the request format as similar as possible between new, renewal, and change cases will make for easier implementations; e.g. the format of the request is identitical except for a type=[renew | change] instead of type=new.

The renew and change 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 and change 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 change 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. [EDITOR'S NOTE: Is there a way to do this?]

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

For the duration of time after a renew or change has been processed and before PKI has received confirmation of the Peer's successful receipt of the new PKC (as described above in section 3.4.9), 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 process occurs.

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

After the renewal or change 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 and

removed from the repository. The decision about if the old PKC should be made unusable is a decision of local policy. Either the PKI or the Admin will need to 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.

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

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

3.6.1 Renew Request for a New PKC (before expiry)

Operators can choose to force renewals for several reasons:

- To enforce an automated "clean up" of unused PKCs that have not been specifically revoked

- To force re-keys

- To have manual review control over re-issuance.

In the latter case, automated renewals will likely not be used. In the former two cases automated renewal is a very attractive option.

At the time of authorization, certain details about renewal acceptance will be conveyed by the Admin to the PKI, as stated in section 3.2.3.2 above. The renewal 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
- Renewal Period
- 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.

3.6.2 Change Request for a New PKC

A change in contents 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.

A Change differs from a Renew in a few ways:

- A re-key is not necessary (though MAY be specified)

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

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

- Once the Change 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

- The existence of a "change" type allows for better logging and tracking of why the new issuance occurred, and why the old PKC was made unusable.

At the time of authorization, certain details about change acceptance MAY be conveyed by the Admin to the PKI, as stated in section 3.2.3.2

above. The change 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 renewal.

If a Change authorization was not made at the time of original authorization, one may be made from Admin to the PKI at any time during the PKC's valid life. When such a Change is desired, Admin must notify the PKI System that a chance is authorized for the end

entity, and to expect it coming, and specify the new contents. Admin then initiates the Change 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).

3.6.3 Error Handling for Renewal and Change

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

3.7 Finding PKCs in repositories

The complete hierarchical validation chain (except the trust point) 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 the above section on extendedKeyUsage.

IPsec Peers MUST be able to resolve Internet domain names and support the manadatory 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.

3.7.1 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.

3.8 Revocation Action

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 for which it authorized the creation. 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 CHANGE, BEFORE EXPIRATION: The PKI MUST be responsible for the PKC revocation during a change 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.

3.9 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 manadatory 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.).

3.9.1 Error Handling in Revocation Checking

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

4. Security Considerations

TBD

A References

A.1 Normative References

None

A.1 Non-Normative References

[STDPROCESS] Bradner, S., "The Internet Standards Process û Revision 3", BCP 9, RFC 2026, October 1996.

[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.

B. Acknowledgements

This draft is substantially based on a prior draft draft-dploy-requirements-00 developed by Project Dploy. The principle editor of that draft was Gregory M. Lebovitz (NetScreen Technologies). Contributing authors included Lebovitz, Paul Hoffman (VPN Consortium), Hank Mauldin (Cisco Systems), and Jussi Kukkonen (SSH Communications Security). Substantial editorial contributions were made by Leo Pluswick (ICSA), Tim Polk (NIST), Chris Wells (SafeNet), Thomas Hardjono(VeriSign), Carlisle Adams (Entrust), and Michael

Shieh (NetScreen).

Once brought to pki4ipsec, the following people made substantial contributions: [TBD] ...

C. Editor's Address

Chris Bonatti
IECA, Inc.
15309 Turkey Foot Road
Darnestown, MD 20878-3640 USA
bonattic@ieca.com

Sean Turner
IECA, Inc.
1421 T Street NW #8
Washington, DC 20009 USA
turners@ieca.com

Gregory M. Lebovitz
NetScreen Technologies, Inc.
gregory@netscreen.com

D. Summary of Requirements

TBD - EDITOR'S NOTE: Plan to add a summary table similar to those in RFCs 1122, 1123, and 2975. Table will briefly describe requirement, state the requirement level (i.e., "MAY", "SHOULD", "MUST", etc.), and cite the applicable paragraph in this draft.

E. Change History

2004-July Draft-bonatti-pki4ipsec-profile-reqts-01

It is submitted as an individual draft in order to meet a publication deadline though it has been accepted in to the working group. The following salient changes were introduced:

- A new Figure 1 was added in section 2.1 to depict just the VPN System.

- A new Figure 2 was added to depict 2.2 to depict just the PKI System.

- The old Figure 1 was moved to section 2.3.

- Section 2.3 was split in to three sections to depict the New PKC, Renewal, and Revocation. Also the text was modified to indicate that the pictures are only for IPsec Peers generating key pairs and requesting PKCs.

- Text and a Figure was added to Section 3.4.6 to show the architectural difference for IPsec Peers enrolling through an Admin.

- Text and a Figure was added to Section 3.4.7 to show the architectural difference for Admins performing the entire enrollment.

2004-January Draft-bonatti-pki4ipsec-profile-reqts-00

This is a revised requirements document based on the existing Project Dploy requirements draft. It adapts the revisions to adapt the Dploy requirements to the scope of the proposed charter for an IETF PKI4IPSEC WG. It is submitted as an individual draft in anticipation of formation of the WG. The following salient changes were introduced:

- Rewrote the abstract to focus on the document rather than the project.

- Rewrote and trimmed introduction to fit proposed scope of deliverable (2) from IETF PKI4IPSEC charter.

- Rewrote sentences throughout to genericize the document for the IETF and remove references to Project Dploy objectives.

- Removed reference to the Dploy Business Case.

- Removed the "Audience" subsection of the introduction because it was redundant with other aspects of the introduction, and unnecessary with the context of the proposed PKI4IPSEC WG.

- Added definition of Community Realm (used in 3.2.3.3) to the "Definitions" subsection.

- Added definition of CRL Distribution Points (CDP) and Authority Info Access (AIA) to the "Definitions" subsection.

- Restructured the "Architecture" section to bring the presentation of Figure 1 to the front to go along with the overview of the section, and to add a new step diagram to the "VPN-PKI Interaction" subsection.

- Added a new subsection 2.1.2 to describe the VPN peer. Text of the new subsection will be supplied in a subsequent draft.

- Added an editor's note to subsection 3.1.2 noting that further elaboration on the nature of "policy details" may be required.

- Subsection 3.2 was deleted to maintain the focus on generic requirements agreed in Minneapolis. Selection of specific protocols will be done in the deliverable (3) profile.

- Delete the requirement from 3.2.3.1 to include the maximum CRL size in the certificate template. This may need to be specified in the profile, but not be in the certificate itself.

- Revised 3.3.3 to to clarify that key escrow requirements and any key transport between the VPN admin and the peer are beyond scope.

- Adopted consistent spelling "enrollment" vs. "enrolment" throughout.

- Replaced instances of "and/or" and other slashed terminology with less ambiguous statements to clarify the requirements.

- Revised the text of 3.5.1 to clarify the proposed requirement in terms of SHALL and MAY terms.

- Retitled 3.5.2 as "Path Validation" instead of "Chaining".

- Added AIA extension as a MAY requirement in 3.5.2.

- Added an editor's note to subsection 3.5.3 to question whether additional keyUsage bits should be set in the certificate.

- Removed the requirement for HTTP support in favor of a requirement for a single mandatory protocol to be specified in the profile.

- Removed subsection on "Intra-IKE Considerations" as these should be dealt with in the existing deliverable (1) PKI profiles.

- Deleted existing sections 5 and 6 dealing with the partipating vendors in Project Dploy.

- Added new section 4 on "Security Considerations". Text of the new subsection will be supplied in a subsequent draft.

- Revised the "Acknowledgements" section to reflect this revision, and provide appropriate credit to Project DPloy.

- Normalized "References" section with the ID-Nits promulgated by the IESG.

- Added a stub for a proposed new Annex D to provide a requirements summary table. Content of the annex will be supplied in a subsequent draft.

2002-March Draft-dploy-requirements-00

- First public draft of the document released.