Project Dploy
Draft Requirements

Lebovitz (NetScreen), et al
Editor
March 2002

Requirements for Large Scale PKI-Enabled VPNs
<draft-dploy-requirements-00.doc>

Abstract

Project Dploy was initiated by IPsec product developers in an effort to work better with the PKI vendors to create interoperable tools needed for large scale, PKI-enabled IPsec VPN deployments. Impediments to such large deployments today include their complexity and operational difficulty, as well as a lack of interoperable methods, multiple competing protocols, and some missing components. To this end IPsec product developers will document requirements for IPsec product interactions with PKI products for such large scale deployments. The requirements will be used to identify the protocols and profiles (or create new ones as needed) that enable the IPsec and PKI vendors to create interoperable, scalable deployment and management tools for PKI-enabled VPNs. The desired outcome is that interoperable products would be created by both IPsec and PKI vendors to enable such deployments, and created as quickly as possible. Project Dploy will not address all or other PKI issues here; the scope is limited to requirements for easing and enabling scalable PKI-enabled IPsec deployments.

Table of Contents

1 Introduction

This document enumerates requirements for 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 specification 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 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 certificate issuance, enrollment process (certificate request and retrieval), certificate renewals and changes, revocation, validation and repository lookups. They enable a VPN Operator to:

- authorize batches of certificate issuances based on locally defined criteria, and do so from the VPN Admin point

- provision PKI-based user and/or machine identity to VPN peers, on large scale. Provision means the IPsec peer ends up with a valid public/private key and PKIX certificate that is used in the VPN tunnel setup.

- set the corresponding gateway and/or client authorization policy for remote access and site-to-site connections

- establish automatic renewal for certificates, or changes

- ensure timely revocation information is available for certificates used in IKE exchanges

Project Dploy was initiated by IPsec product developers in an effort to better work with the PKI vendors to create interoperable tools needed for large scale, PKI-enabled IPsec VPN build-outs by customers. One main reason for failure in this area to date has been due to the lack of definition of a centralized policy function within the VPN system. Even where such products exist, they definitely do not speak to the PKI today. Dploy defines this centralized function, and the transactions between it and the PKI system. Other impediments to such large deployments today are the complexity, difficulty, lack of interoperable methods, multiple competing protocols/methods, and some missing components, all of which are addressed within these requirements.

To date, SCEP and "cut-and-paste" techniques are more commonly used to accomplish end entity certificate acquisition for IPsec VPN usage, but are better suited to small VPN deployments, and are out of scope for this solution. Instead, a robust certificate management scheme is needed to empower operators in large scale deployment and management efforts. A separate document containing the business case justifying this need is “draft-dploy-bizcase-00.rtf” [BIZ].

The desired outcome is that interoperable products would be created by both IPsec and PKI vendors to enable such scalable deployments, and do so as quickly as possible. For example, an IPsec operator should be able to use any conforming IPsec vendor’s implementation of the final Dploy specification with any conforming PKI vendor’s implementation to perform the VPN rollout and management as described below. Such standards and profiles do not exist today, Project Dploy aims to create them.

These requirements will be used to identify a specific protocol that may be leveraged to accomplish such large-scale deployments. The specification will also profile existing PKIX and IPsec standards/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/extensions to an existing protocol, especially in the area of bulk authorization for certificate issuance. The document introduces the idea of a VPN administration function (ADMIN) within the VPN system. This Admin function bears great responsibility for the task of managing pre-authorization for certificate issuance and of distributing the results between the VPN system and the PKI system.

1.1 Scope

The solution focuses on the needs of large-scale rollouts, i.e. VPNs including thousands of managed VPN gateways and/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.

We do not intend to discuss all or other PKI issues here. The scope is limited to requirements for easing and enabling scalable IPsec with PKI deployments.

Section 3.1.1 below describes a VPN Administrative function and its communication with the IPsec Peers in the VPN System. The steps for what should occur as a result of that communication is described in this document. (See the section below on non-goals).

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 certificate cross-certification will not be addressed

The case for when certificates are not sent in the IKE payload will not be addressed. This solution cannot be accomplished without a change to IKE v1. Such changes will not happen at present, as IKEv2 is in the works. It is recommended to consider this scenario in future iterations of Dploy (and maybe IKEv2).

The specification for the communication method and transactions between Admin and peers is up to vendor implementation and therefore is not expected to be included in the Dploy Specification document. Such a protocol may be standardized at a later date to enable interoperability between Admin stations and IPsec Peers from different vendors, but is far beyond the scope of this current effort, and will be considered opaque to the Dploy specification.

1.3 Audience

This document has three basic audiences. The first audience is IPsec VPN vendors, who may use this specification to build interoperable VPN components that rely on PKI for authentication. The second audience is PKI vendors, who may use this specification to build interoperable PKI components that support VPN deployments as described in this specification. The third audience is VPN/PKI operational personnel, who may use this specification to select conforming equipment and configure that equipment appropriately.

1.4 Definitions

VPN System

The VPN System is comprised of the VPN Admin function (defined below), the IKE peers, and the communication mechanism between the Admin and the peers. VPN System is defined in more detail in section 3.1.

PKI System

The PKI System is the set of functions needed to authorize and issue certificates and provide revocation information about those certificates. PKI System is defined in more detail in section 3.2.

(VPN) Operator

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 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 Certificate provisioning for the VPN connections. See Section 3.1.1 below for more details.

End Entity

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

Certificate Renewal

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

Certificate Change

A special case of a renewal-like occurrence where a certificate needs to be changed prior to expiration due to some change in its subject's information, such as a transfer from the OU=Engineering to OU=Marketing, or the change of a last name at marriage that would effect Surname or CN fields.

Registration Authority

An optional entity 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 certificate.

Certificate Authority

An authority trusted by one or more users to create and assign public key certificates. It is important to note that the CA is responsible for the public key certificates during their whole lifetime, not just for issuing them.

Repository

An Internet-accessible server that stores and makes available for retrieval certificates and certificate revocation list

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 certificates that is signed by a CA and made freely available in a public repository. Peers retrieve the CRL to verify that a certificate being presented to them as identity in an IKE transaction has not been revoked.

Authority Info Access (AIA)

The authority information access extension indicates how to access CA information and services for the issuer of the certificate in which the extension appears. Information and services may include on-line validation services and CA policy data.

1.5 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 [RFC 2119].

1.6 Change History

This is the first public draft therefore no changes exist.

1.6.1 [Place Holder]

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 Admin. Peers are two entities between which the Operator requires an IPsec tunnel establishment. Two Peers are shown below, but implementations SHOULD support an actual number in the tens of 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.

The Admin MUST be reachable by the Peers. Most implementations will be 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 “online”. 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.

2.1.1 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 roll defined here is that Admin specifies to PKI the contents and use parameters of the credentials PKI will issue. In this way Admin MAY perform many RA-like functions, for example authorization of certificate 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/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 VPN Peers will vary widely from vendor to vendor. However some basic functions of the Admin are assumed.

- It will be the place where certificate policy 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 certificate 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 certificates 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 certificates by sending the parameters and contents for those certificates, likely derived from templates created on itself by the VPN Operator. It will receive back from the PKI identification numbers and authorization codes to be used in the requests for each of the pre-authorized certificates.

- It will deliver instructions to the IPsec Peers, and the Peers will carry out those instructions. An example of such an instruction is a policy configuration. Therefore, the communication mechanism between the Admin and the IPsec peers MUST be secure and authenticated. 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).

2.2 PKI System

The PKI system may be set up and operated by the VPN Operator (in-house), may be provided by 3rd party PKI providers to which connectivity is available at the time of provisioning (managed PKI service), or may be integrated with the VPN product.

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

The PKI System will contain a mechanism for handling Admin's authorization requests and certificate enrollments. These mechanisms are referred to as the Registration Authority (RA) below. It will also contain a mechanism that the Peers can use to validate each others' certificates, retrieving CRLs, called the Validation Authority. The PKI system contains a Repository used by the Peers to look up each others certificates. Last, the PKI System contains the core function of a Certificate Authority (CA) that uses a public/private key pair and signs certificates.

The PKI system SHOULD be built so that lookups resolve directly and completely at the URL indicated in a CRL Distribution Point, or Authority Info Access. The PKI should be built such that URL contents to do not contain referrals to other hosts or URLs, as such referral lookups will increase the time to complete the IKE negotiation, and may cause implementations to timeout.

2.3 Architecture Diagram

```
+---------------------------------------------+
|                   PKI                       |
|                                             |
|   +--------------+                          |
|   |  Repository  |    +----+   +----+       |
|   | Certs & CRLs |    | CA |   | RA |       |
|   +--------------+    +----+   +----+       |
|                                             |
+---------------------------------------------+
      ^                  ^                ^
      |                  |                |
      |[E]               |[A]             |[E]
      |[M]               |[E]             |[M]
      |[R]               |                |[R]
      |                  v                |
      |             +----------+          |
      |      [G]    |   VPN    |  [G]     |
      | +---------->|  Admin   |<-------+ |
      | |           | Function |        | |
      | |           +----------+        | |
      v v                               v v
+---------+                         +---------+
|   VPN   |           [I]           |   VPN   |
|  Peer 1 |<=======================>|  Peer 2 |
+---------+                         +---------+

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

Figure 1. Architectural Framework for VPN-PKI interaction

2.4 The VPN-PKI interaction

The steps of the VPN-PKI interaction are summarized here. The detailed requirements are described below in Sect X.

1) Authorization [A]. Admin sends a list of IDs and certificate contents for the PKI systems to authorize enrollment, and the PKI returns a list of unique identifiers and one-time tokens to be used for the enrollment of each certificate. Other certificate usage policy is also set at this time, for example parameters for renewals or changes, key lengths, etc.

2) Generate public and private key pair and certificate request [G]. The Admin communicates with the Peer to either give it information so that it can
generate a key pair and certificate request and send the request directly to the PKI, or to give them a key pair and certificate request that the Peers can send to the PKI. In some cases, the Peers MAY generate the key pair and certificate locally, then send the certificate request back to the VPN Admin Station who will send the request to the PKI System for them (more about these options will be mentioned in section 4.3 and 4.4 below)

3) Enrollment - Request and retrieve certificate [E]. The Peers of the Admin request their certificates from the PKI and receive them back from the PKI.

4) IKE/IPsec communication [I]. The Peers communicate authorized by the certificates they received from the PKI.

5) Maintenance: revocation lookups, revocations, and repository lookups [M]. During IPsec communication, the Peers communicate with the PKI to validate each others certificates, such as by retrieving CRLs. The Peers also look up each other's certificates using HTTP. Each Peer or the Admin can also communicate with the PKI to revoke a certificate if keys have been compromised.

6) Renewals and changes [R]. Admin station communicates renewal or change instructions to the Peers and also to the PKI, enabling the Peers to automatically generate renewal/change requests as needed, and for those requests to be immediately granted by the PKI System.

3 Requirements

3.1 General requirements

3.1.1 One Protocol

Dploy will create a specification for PKI-enabled IPsec VPNs. This specification 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/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 products MUST implement the ONE PROTOCOL or ONE USE PROFILE that is specified for a given requirement.

3.1.2 Secure Transactions

Dploy largely specifies 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.

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 online availability of the PKI system and Admin for the life of the VPN. Light consideration has been given to environments where physical media is used as the transport method.

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 certificate 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, one of the key objectives of the Dploy effort.

3.2 One Protocol for Certificate Management

One of the major goals of Dploy is to support a single certificate transaction protocol. There are three protocols, Simple Certificate Enrollment Protocol (SCEP), Certificate Management protocol using CMS (CMC) and Certificate Management Protocol (CMP). Although SCEP is close to a de facto standard in that the large majority of IPsec VPN vendors and many PKI vendors support it, SCEP was designed to be an interim step and is not considered as a candidate. CMC and CMP both had proponents in attendance for their particular protocol to be considered for certificate transactions between the VPN administration station or VPN client and the PKI components.

With regard to the PKI vendors themselves, the major PKI vendors appear to be divided between the support of CMC and CMP protocols. There are some PKI implementations with CMP available. Also there has been private interoperability testing between one PKI vendor and one IPsec vendor with CMC. The PKI industry is split, though more have CMP implementations that CMC today.

Based on the technical discussions regarding the completeness of each protocol, there is a consensus that neither protocol meets all the technical requirements of Dploy. Each protocol will require modifications and extensions in order to meet the requirements. Dploy thought it better to edit an already close solution than to create yet another protocol, both for ease and time to market. There appears to be workable solutions to the requirement deficiencies in either protocol. Neither protocol is considered overwhelmingly better from a technical standpoint. It was determined that the CMC protocol specification will need to add definition in its transaction process, error handling, and add transactions for renewal and change issuance types.

One of the major influences in the decision making process is an IPsec Vendor’s experience with and understanding of the current industry practice of a PKCS#10 request followed by a PKCS#7 response. Most IPsec VPN vendors have a SCEP implementation for their PKI product, which has provided experience with PKCS #10, PKCS#7, and SCEP request formats. This experience and basic understanding by the developers provide a base for migration to CMC, which uses the previously mentioned protocols and formats. CMP requires a completely new development effort, as the basic formats utilized for request and retrieval are different. Along the same line is that having implemented SCEP means the company possesses an appropriate toolkit. Also there are several PKCS#10 and PKCS#7 toolkits freely available. CMP requires a different toolkit. It is not a simple matter for a company to change toolkits, and this was considered as well. Therefore it is believed that it would be easier and faster for the IPsec vendors to implement CMC.

For these reasons the majority of IPsec VPN vendors represented at Dploy state a preference for CMC. Some had already begun projects supporting CMC, and there were a few IPsec VPN vendors with CMP implementations already.

The consensus recommendation is to use CMC as the one certificate transaction protocol.

3.3 Authorization Transactions

3.3.1 Bulk Authorization

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 certificate enrollment request (in which case it is a "pre-authorization"), or within the same connection.

3.3.2 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 VPN Administration to RA/CA interaction.

The PKI responds to the VPN Administration station with Authorization identifiers (maybe serial numbers or such) and a corresponding key 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 and authorization.

3.3.3 Admin Authorization Requests to PKI

3.3.3.1 Specifying Fields within the Certificate

Components of the Certificate Template that the VPN Admin MAY send to the PKI to authorize the eventual creation of certificates include:

* DN fields
* Any number of locally defined CNs with their contents
* Validation Period of the Certificate
* Renewal parameters (i.e. N% of validity period, and certificate overlap duration in N [measurement? Minutes? Hours? %?], or just let it expire)
* Any of SubjAlt 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. NOTE: The specification should define who will handle the CDP contents. Suggest the PKI, not Admin, but further research is needed.

* Size of CRL the client can support. This may be a function of the template being used for this set of Peers.
* Not Valid before
* Not Valid After

3.3.3.2 Authorizations for Renewal and Change

When the Admin sends its authorization 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 renewal for the specified end-entity certificate.

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

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

* Whether or not a new key must be used for the new certificate.

* 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 certificate expiration date the PKI will accept and process a renewal.

* Length of time (if ever) after PKI receives end entity peer confirmation (see 4.4.8 below) that the old certificate 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 4.4.8 below) that the old certificate is revoked, and removed from repository.

3.3.3.3 Other authorization elements

If CDP is flagged as required in the authorization request, then the Admin will also specify the method (i.e. HTTP or LDAP, though HTTP is the Dploy specified mechanism)

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 COMMUNITY REALM for the certificate to the PKI. COMMUNITY REALM is an important component in provisioning that allows the Admin to specify for the Peer various elements of the certificate'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/export for easily synchronizing the COMMUNITY REALM lists between the Admin and PKI systems.

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. Now what? Proper conflict handling MUST be specified.)

3.3.4 Cancel Capability

Admin can send a cancel authorization message to PKI. 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.

3.3.5 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 certificate 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 certificates 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 certificate contents that may occur due to a change in PKI configuration right in the middle of a batch pre-authorization request.

After receiving a bulk authorization request from the Admin, the PKI must be able to reply YES to those individual certificate authorizations that it can satisfy and NO or FAILED for those requests that

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

3.3.6 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, one of the key objectives of the Dploy effort.

3.4 Key Generation and Certificate 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/private key pairs and to construct certificate 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 certificate constructions MAY occur at either the IPsec Peer or a combination of the Peer and the Admin.

3.4.1 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 certificate request, including key type and size. Recall that the mechanism for how this information is communicated from the Admin to the Peer is opaque to the Dploy specification.

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

3.4.2 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 certificate 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 certificate generation.

3.4.3 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 certificates 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.

The Admin will generate the key pair, construct the certificate request, and enroll on behalf of the Peer. Once the certificate has been retrieved, the keys and certificate will be sent to the Peer using a secure method, such as PKCS12 (see Appendix A for a discussion of PKCS12).

What is the value for pre-authorization when the Admin is the one doing the key generation? 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 certificates, but Operator2 is the one doing batch enrollments and VPN device configurations.

3.4.4 Trust Anchor Certificate Acquisition

The root certificate MUST arrive on the peer via one of two methods:

(a) Peer can get the root certificate 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 via the enrolment protocol.

3.4.5 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, one of the key objectives of the Dploy effort.

3.5 Enrollment (Sending Request and Certificate Retrieval)

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

The protocol MUST be exactly the same regardless of whether the enrolment occurs from the Peer to the PKI or from the Admin to the PKI (as seen below in sections 4.4.5 through 4.4.7).

3.5.1 One protocol

One protocol MUST be specified for both request and retrieval.

3.5.2 On-line protocol

The protocol MUST supports automated enrollment that occurs over the Internet (as is done today by SCEP and others) and without the need for manual intervention.

3.5.3 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 certificate match, and the Authorization identifier and key are accurate, the certificate is generated immediately, with no need for manual intervention.

The PKI makes the certificate available immediately for retrieval, or possibly sends the certificate to the end entity as a response in the request/retrieval exchange.

3.5.4 Manual Approval Option

The option/ability to queue and manually approve certificate 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 certificate 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.

3.5.5 Enrolment 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 enrolment, 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.

3.5.6 Enrolment Method 2: IPsec Peer Enrolls to PKI through Admin

In this case, the IPsec Peer has generated the key pair and the certificate 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 enrolment to the PKI system. The PKI System does not care where the enrolment comes from, as long as it is a valid enrolment. Once the Admin retrieves the certificate, 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/private key pair of sufficient strength for secure IPsec. In this case, the end entity needs to prove to the VPN Admin that they have such a key pair; this is normally done by the VPN Admin sending the end entity a nonce, which the end entity signs and returns to the VPN Admin along with the end entity's public key.

3.5.7 Enrolment 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 [RFC 2459]. 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 certificate. Again, the mechanism for the Peer to Admin communication is opaque.

3.5.8 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 4.6). 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.

3.5.9 Confirmation Handshake

Any time a new certificate 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 certificate, loaded it, and can use it effectively in an IKE exchange. This requirement exists so that:

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

* A revocation may be invoked if the certificate 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 certificate in IKE connections. The PKI MUST complete all processing necessary to enable the end entity's operational use of the new certificate (for example, writing the certificate to the repository) before sending the confirmation acknowledgement. The PKI MUST also issue a revoke on the original certificate before sending the confirmation ACK (see section 4.X). The end entity peer MUST NOT begin using the certificate until the PKI's confirmation acknowledgement has been received.

3.5.10 Failure Cases

Thorough error condition descriptions and handling instructions are required for each transaction in the enroll/retrieval process. Providing such error codes will greatly aid interoperability efforts between the PKI and IPsec products, one of the key objectives of the Dploy effort.

The Dploy specification must clarify what happens if the request/retrieval fails for some reason. The following cases will be covered:

* Admin/Peer cannot send the request
* Admin/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 certificate
* The PKI system generated the certificate, but could not send it
* The PKI sent the certificate, but the requestor (Admin or Peer) did not receive it

* The Requestor (Admin/Peer) received the certificate, but could not process it due to incorrect contents, or other certificate-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 certificate 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 certificate again.

3.6 Certificate profile for an IPsec Certificate

A certificate 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 Dploy specification).

It is preferable that the certificate profiles for IPsec and S/MIME were the same so that one certificate could be used for both protocols. However, failure to achieve this requirement in the Dploy specification MUST NOT hold up the Dploy standardization effort.

3.6.1 Identity Usage

The portion of the certificate to be used by the Peer for identity verification will be either the X.500 Distinguished Name (DN) within the Subject Name contents or a specific field within the Extension SubjectAlternativeName (per [RFC 2407] 4.6.2.1 Identification Type Values). Usage descriptions for each follow.

If one of these items will be used for identity, they MUST be included in the SubjectAltName
* FQDN
* RFC 822 (also called USER FQDN)
* IPv4 Address
* IPv6 Address
While these data strings may also exist in the DN, they will not be looked for in the DN, only in SubjectAltName.

3.6.2 Chaining

The Peers must validate the chain. The contents necessary in the certificate to allow this will be enumerated in the Dploy specification document.

The peer MAY have the ability to construct the chain itself, however Admin MUST be able to supply peers with the trust anchor and any chaining certificates necessary.

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

3.6.3 KeyUsage

The certificate's KeyUsage digialSignature bit [Son-of-2459] MUST be flagged on.

3.6.4 Extended KeyUsage

EKU's are not required for Dploy. 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, your 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.

3.6.5 Pointer to Revocation Checking

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

The location and method for either a CRL Distribution Point (CDP) and an Authority Info Access (AIA) [RFC 2459] MUST be included in the certificate. 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.

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

3.7 Certificate Renewals and Changes

In order to allow for continued certificate usage, a new certificate will need to be issued for an end-entity before the end entity's currently held certificate expires. A renewal is defined as a new certificate issuance with the same SubjectName and SubjectAlternativeName contents as an existing certificate for the same end entity before expiration of the end entity's current certificate. A change is defined as a new certificate issuance with an altered SubjectName and/or SubjectAlternativeName for the same end-entity before expiration of the end-entity's current certificate. Renewals and changes are variants of a request/retrieval scenario with unique operational and management requirements.

Once the PKI has issued a certificate for the end entity Peer, the Peer MUST be able to either contact the PKI directly or through the Admin for any subsequent renewals and/or changes. The PKI MUST support either case.

It is desired that a renew/change request contain an element that identifies the request as either type=renewal, or type=change. This element MUST be specified in the final Dploy specification. This will allow for better management, logging and auditing of certificate management.

When sending a renew/change request, the entire contents of the certificate 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/renew/change cases will make for easier implementations.

The renew/change request MUST be signed by the private key of the old certificate. 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 certificate with another end entity's identity.

Whether or not a new key is used for the new certificate in a renew/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.

The new certificate resulting from a renew/change will be retrieved in band, using the same mechanism as a new request/retrieval.

For the duration of time after a renew/change has been processed and before PKI has received confirmation of the peer's successful receipt of the new certificate (as described above in section 4.4.8), both certificates – 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 certificate while the renewal process occurs.

In the case where new keys were generated for a renew/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 certificate can occur once all connections that used the old certificate have expired.

After the renew/change occurs, the question now exists for the PKI of what to do about the old certificate. If the old certificate is to be made unusable, the PKI will need add it to the revocation list and removed from the repository. The decision about if the old certificate should be made unusable is a decision of local policy. and the Admin will need to specify this parameter during the authorization phase. In this case Admin MUST also specify during authorization the length of time after the PKI receives the end entity peer's confirmation (of receipt of the certificate) that will pass before the old certificate is made unusable.

If a certificate has been revoked, it MUST NOT be allowed a renew or change.

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

3.7.1 Renew Request for a New Certificate (before expiry)

VPN Operators may choose to force renewals for several reasons:

* To enforce an automated "clean up" of unused certificates 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 4.2.3.2 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 certificate unusable

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

3.7.2 Change Request for a New Certificate

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 certificate 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 certificate's period of validity
* Once the Change is completed, and the new certificate is confirmed, the old certificate 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 certificate 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 3.3.3.2 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 certificate 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 certificate'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).

3.7.3 Error Handling for Renewal/Change

Thorough error condition descriptions and handling instructions are required for each transaction in the renew/change process. Providing such error codes will greatly aid interoperability efforts between the PKI and IPsec products, one of the key objectives of the Dploy effort.

3.8 Finding certificates 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 certificates sent during the IKE transaction.

All certificates 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/PAT)
* The protocol can easily perform some query against a remote directory on a specific ID element that was given to it in a standard certificate field

A consensus decision has chosen HTTP as the MUST protocol for repository lookups. HTTP's has a much wider implementation than LDAP, greater speed, relative ease of administration, and greater scalability (though there were also proponents for LDAP, because most PKI repositories are LDAP today, and LDAP lookups can be more complex as needed).

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

Certificates MAY have extendedKeyusage to help identify the proper certificate for IPsec in HTTP and in peer's cache, though the default behavior is to not use them. See the above section on extendedKeyusage.

IPsec peers MUST be able to resolve both DNS and act as an HTTP client at the time of starting up so they can perform the certificate lookups.

IPsec peers should cache certificates 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 certificates are not sent in IKE is a stated non-goal of the Dploy effort at this time.

3.8.1 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, one of the key objectives of the Dploy effort.

3.9 Revocation Action

The peer MUST be able to initiate revocation for its own certificate. In this case the revocation request MUST be signed by the peer's current key pair for the certificate 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 certificate for which it authorized the creation. The Admin will identify itself to the PKI by use of its own certificate; it MUST sign any revocation request to the PKI with the private key from its own certificate. The PKI MUST have the ability to configure Admin(s) with revocation authority, as identified by its certificate. Any certificate authorizations must specify if said certificate may be revoked by the Admin (see section 3.3.3.2 for more details).

The Dploy specification MUST identify the one protocol and/or transaction within a protocol to be used for both peer and Admin initiated revocations.

Below are guidelines for revocation in specific Dploy transactions:

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

* AFTER CHANGE, BEFORE EXPIRATION: The PKI MUST be responsible for the certificate revocation during a change transaction. PKI MUST revoke the certificate after receiving the confirm notification from the peer, and before sending the confirm-ack to the peer. The peer MUST NOT revoke its own certificate in this case.

3.10 Revocation Checking and Status Information

The PKI system MUST provide a mechanism whereby peers can check the revocation status of certificates that are presented to it for IKE identity. The mechanism should allow for access to extremely fresh revocation information. Dploy has chosen CRLs as that mechanism. Operators are RECOMMENDED to refresh CRLs as often as logistically possible.

All CRL lookups MUST be performed through HTTP. The reasons for preferring HTTP (over LDAP, for example) are HTTP's much wider implementation, greater speed, relative ease of administration, and greater scalability.

All certificates used in IKE MUST have cRLDistributionPoint and authorityInfoAccess fields populated with valid URLs. This will allow all recipients of the certificate 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 certificate as valid identity in the exchange and need not perform any check.

If the revocation of a certificate is used as the only means of deactivation of access authorization for the VPN 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.).

3.10.1 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, one of the key objectives of the Dploy effort.

3.11 Statistics and Monitoring

* Number of Authorizations requested
* Number of Authorization granted
* Number of Authorizations with completed enrollments
* Number of Pending Authorizations

*****More brainstorming needed here

3.12 Intra-IKE Considerations

3.12.1 Certificate Request payloads

Certificate Request payloads are to be used only in environments where VPN peers have been configured to have single point of trust. When cross certification or trust lists are used, number of CertReq's gets large. In these scenarios implementations are advised not to issue Certificate Requests but to rely on path validation to handle the problem.

3.12.2 Sending certificates in IKE payloads

Implementation MUST send certificates and they MUST be sent in last (encrypted) IKE main mode packets. Implementations are expected to send only their EE certificate. This is done because sending full certificate chains leads to large packets and fragmentation. Certificate chains are useful for the scenario where a CA re-key occurs. However, recent tests with several vendor devices that perform NAT have shown that they do not adequately handle UDP fragmentation, thus making the IKE exchange unreliable when passing through such NAT devices. Clearly the best course of action is for such vendors to improve their NAT implementations, but we cannot assume that, so we will be considerate to limit fragmentation. Therefore chaining information should be sent from Admin to the peer, out of band of the IKE negotiation between the two peers.

If implementation decides to send more than just EE certificate, it MUST send it's own EE certificate in first certificate payload. Recipient is not required to process any other than first certificate payload.

Implementation MUST use X.509 Signature(4) and PKCS7 type formatting for encoding certificate payloads, encoding each certificate as a separate payload.

IKE ID payload MUST be specified in a format that can be found from associated certificate (either DN or one of the SubjectAltName formats)

If Distinguished Name is used in ID payload, it has to be bitwise identical with DN specified in certificate. Implementations are not required to perform ASN.1 format conversions in order to determine similarity of two DNs.

To ease path construction, end-entity certificates will contain issuing CA access points in either the CRL Distribution Point (CDP) and AIA extensions, as specified in section 4.5.5 above.

3.12.3 Identity Handling

If DN's are used for identity, the entire DN will need to be considered in order to properly perform revocation and validation checking on the certificate. However, local security policy for an IKE receiver may not desire to define an exact match of the entire contents of its peer's DN for it policy's identification purposes. Such a policy may only want to know that (a) a certain field(s) within the DN matches a locally stored value, (b) the certificate is valid, and (c) the certificate has not been revoked. An example is an environment where certificates for remote access users are issued by a specified CA, call it CA1, with DNs that include, among other things, O=Company and OU=Department. Instead of holding each and every remote access users' full DN in the IPsec gateway for IKE identity checking, the gateway holds an SA definition for any peer presenting a valid, non-revoked certificate, issued by CA1, with O=Company and OU=Department. Therefore, a receiving peer MAY use either the whole DN or use specified fields of the DN for identity matching. If certain fields are specified in the local policy, then those fields MUST match what exists in the presented certificate's DN.

If either FQDN, RFC 822 NAME or IPv4 ADDRESS are to be used as identity then these fields MUST appear in SubjectAltName, as described below. While nothing prevents FQDN, RFC 822 NAME, or IPv4 ADDRESS information from appearing as string somewhere in the Subject Name contents, such entries there MUST NOT be interpreted as identity if the receiver is expecting any one of them – they MUST exist in the SubjectAltName section.

3.12.4 CRLs with IKE

Implementations MUST NOT rely on IKE certificate payloads for obtaining up-to-date CRL information. Implementations MUST have HTTP client functionality for retrieving CRL's. The reason for avoiding exchange of CRL's within the IKE negotiation is to prevent fragmentation of IKE packets.

4 Acknowledgements

Paul Hoffman – Sect 4.7 & 4.8, ASCII art, references, lots of edit suggestions, much other council and feedback
Leo Pluswick – Tireless organizing and facilitation, he kept it all moving, plus edits
Tim Polk – getting us all going and drafting the first specification, excellent technical advise
Jussi Kukkonen – Sect 4.10 and various other inputs
Hank Mauldin – Sect 5, Editing on Sect 4.7 & 4.8
Chris Wells – Editing on Sect 4.7 & 4.8
Thomas Hardjono (VeriSign) and Carlisle Adams (Entrust) – Both these gentleman have been extremely supportive. In fact, their positive feedback on the idea, and personal commitments to making the effort succeed, are what caused us to push forward in trying to achieve Dploy.
Michael Shieh – Constant technical advisor and idea man.

5 Participating IPsec Product Vendors

The following IPsec product vendors participated in creating the first draft of the Requirements Statement:

Check Point
Cisco Systems
Cylink
Microsoft
NetScreen Technologies
Network Associates
Nexsi
Nokia
SafeNet, Inc.
Secure Computing Corporation
SSH

6 Endorsing IPsec Product Vendors

The following IPsec product vendors endorse the requirements and specifications and intend to implement to them (Hopefully this list will be enlarged):

Alcatel

Cisco Systems
Cylink
Microsoft
NetScreen Technologies
Network Associates
Nexsi
Nokia
Secure Computing Corporation
SafeNet, Inc.
SSH

7 References

[BIZ] Lebovitz, G. M., "Project Dploy Business Case", March, 2002.

[RFC 2459] Housley, R., et. al. "Internet X.509 Public Key Infrastructure Certificate and CRL Profile", January 1999.

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

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

8 Appendix A – PKCS 12

PKCS 12 could be a secure and convenient method for containing and sending the key pair and certificate when these are generated off the IPsec Peer by the Admin. Note that PKCS 12 is not an Internet-Draft. More information on PKCS 12 can be found at XXXXXXX

Authors' Addresses

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

Paul Hoffman
paul.hoffman@vpnc.org
VPN Consortium

Hank Mauldin
hmauldin@cisco.com
Cisco Systems

Jussi Kukkonen
kukkonen@ssh.com
SSH Communications Security