

Brian Korver
Xythos Software
Eric Rescorla
RTFM, Inc.

INTERNET-DRAFT

<[draft-ietf-ipsec-pki-profile-04.txt](#)>

Feb 2004 (Expires Jul 2004)

The Internet IP Security PKI Profile of IKE/ISAKMP and PKIX

Status of this Memo

This document is an Internet-Draft and is in full conformance with all provisions of [Section 10 of RFC2026](#). Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as ``work in progress.''

To learn the current status of any Internet-Draft, please check the

``[1id-abstracts.txt](#)'' listing contained in the Internet-Drafts Shadow Directories on [ftp.is.co.za](#) (Africa), [nic.nordu.net](#) (Europe), [munnari.oz.au](#) (Pacific Rim), [ftp.ietf.org](#) (US East Coast), or [ftp.isi.edu](#) (US West Coast).

Abstract

ISAKMP and PKIX both provide frameworks that must be profiled for use in a given application. This document provides a profile of ISAKMP and PKIX that defines the requirements for using PKI technology in the context of IPsec. The document complements protocol specifications such as IKE, which assume the existence of public key certificates and related keying materials, but which do not address PKI issues explicitly. This document addresses those issues.

Table of Contents

1	Introduction	4
2	Terms and Definitions	5
3	Profile of IKE/ISAKMP	5
3.1	Identification Payload	5
3.1.1	ID_IPV4_ADDR and ID_IPV6_ADDR	7
3.1.2	ID_FQDN	8
3.1.3	ID_USER_FQDN	8

3.1.4	ID_IPV4_ADDR_SUBNET, ID_IPV6_ADDR_SUBNET, ID_IPV4_A...	9
3.1.5	ID_DER_ASN1_DN	9
3.1.6	ID_DER_ASN1_GN	10
3.1.7	ID_KEY_ID	10
3.1.8	Selecting an Identity from a Certificate	10
3.1.9	Transitively Binding Identity to Policy	10
3.2	Certificate Request Payload	10
3.2.1	Certificate Type	11
3.2.2	X.509 Certificate - Signature	11
3.2.3	Certificate Revocation List (CRL)	11
3.2.4	Authority Revocation List (ARL)	11
3.2.5	PKCS #7 wrapped X.509 certificate	12
3.2.6	Presence or Absence of Certificate Request Payloads	12
3.2.7	Certificate Requests	12
3.2.7.1	Specifying Certificate Authorities	12
3.2.7.2	Empty Certificate Authority Field	12
3.2.8	Robustness	13
3.2.8.1	Unrecognized or Unsupported Certificate Types	13
3.2.8.2	Undecodable Certificate Authority Fields	13
3.2.8.3	Ordering of Certificate Request Payloads	13
3.2.9	Optimizations	13
3.2.9.1	Duplicate Certificate Request Payloads	13
3.2.9.2	Name Lowest 'Common' Certification Authorities	13
3.2.9.3	Example	14
3.3	Certificate Payload	14
3.3.1	Certificate Type	14
3.3.2	X.509 Certificate - Signature	15
3.3.3	Certificate Revocation List (CRL)	15
3.3.4	Authority Revocation List (ARL)	15
3.3.5	PKCS #7 wrapped X.509 certificate	15
3.3.6	Certificate Payloads Not Mandatory	15
3.3.7	Response to Multiple Certificate Authority Proposals	16
3.3.8	Using Local Keying Materials	16
3.3.9	Robustness	16

3.3.9.1	Unrecognized or Unsupported Certificate Types	16
3.3.9.2	Undecodable Certificate Data Fields	16
3.3.9.3	Ordering of Certificate Payloads	16
3.3.9.4	Duplicate Certificate Payloads	17
3.3.9.5	Irrelevant Certificates	17
3.3.10	Optimizations	17
3.3.10.1	Duplicate Certificate Payloads	17
3.3.10.2	Send Lowest 'Common' Certificates	17
3.3.10.3	Ignore Duplicate Certificate Payloads	17
3.3.11	Hash Payload	18
4	Profile of PKIX	18
4.1	X.509 Certificates	18
4.1.1	Versions	18
4.1.2	Subject Name	18

Korver, Rescorla

[Page 2]Intern

4.1.2.1	Empty Subject Name	18
4.1.2.2	Specifying Non-FQDN Hosts in Subject Name	18
4.1.2.3	Specifying FQDN Host Names in Subject Name	19
4.1.2.4	EmailAddress	19
4.1.3	X.509 Certificate Extensions	19
4.1.3.1	AuthorityKeyIdentifier	19
4.1.3.2	SubjectKeyIdentifier	20
4.1.3.3	KeyUsage	20
4.1.3.4	PrivateKeyUsagePeriod	20
4.1.3.5	Certificate Policies	20
4.1.3.6	PolicyMappings	20
4.1.3.7	SubjectAltName	21
4.1.3.7.1	dNSName	21
4.1.3.7.2	iPAddress	21
4.1.3.7.3	rfc822Name	21
4.1.3.8	IssuerAltName	21
4.1.3.9	SubjectDirectoryAttributes	22
4.1.3.10	BasicConstraints	22
4.1.3.11	NameConstraints	22
4.1.3.12	PolicyConstraints	22
4.1.3.13	ExtendedKeyUsage	22
4.1.3.14	CRLDistributionPoints	23
4.1.3.15	InhibitAnyPolicy	23
4.1.3.16	FreshestCRL	23
4.1.3.17	AuthorityInfoAccess	23
4.1.3.18	SubjectInfoAccess	23
4.2	X.509 Certificate Revocation Lists	24
4.2.1	Multiple Sources of Certificate Revocation Informati... 24	
4.2.2	X.509 Certificate Revocation List Extensions	24

4.2.2.1	AuthorityKeyIdentifier	24
4.2.2.2	IssuerAltName	24
4.2.2.3	CRLNumber	24
4.2.2.4	DeltaCRLIndicator	24
4.2.2.4.1	If Delta CRLs Are Unsupported	25
4.2.2.4.2	Delta CRL Recommendations	25
4.2.2.5	IssuingDistributionPoint	25
4.2.2.6	FreshestCRL	25
5	Configuration Data Exchange Conventions	25
5.1	Certificates	26
5.2	Public Keys	26
5.3	PKCS#10 Certificate Signing Requests	26
6	Security Considerations	26
6.1	Identity Payload	26
6.2	Certificate Request Payload	27
6.3	Certificate Payload	27
6.4	IKE Main Mode	27
7	Intellectual Property Rights	27
8	IANA Considerations	27

Korver, Rescorla

[Page 3]Intern

9	Normative References	27
10	Informational References	28
11	Acknowledgements	28
12	Author's Addresses	28

[1](#). Introduction

IKE [[IKE](#)] and ISAKMP [[ISAKMP](#)] provide a secure key exchange mechanism for use with IPsec [[IPSEC](#)]. In many cases the peers authenticate using digital certificates as specified in PKIX [[PKIX](#)]. Unfortunately, the combination of these standards leads to an underspecified set of requirements for the use of certificates in the context of IPsec.

ISAKMP references PKIX but in many cases merely specifies the contents of various messages without specifying their syntax or semantics. Meanwhile, PKIX provides a large set of certificate mechanisms which are generally applicable for Internet protocols, but little specific guidance for IPsec. Given the numerous underspecified choices, interoperability is hampered if all implementors do not make similar choices, or at least fail to account for implementations

which have chosen differently.

This profile of the ISAKMP and PKIX frameworks is intended to provide an agreed-upon standard for using PKI technology in the context of IPsec by profiling the PKIX framework for use with ISAKMP and IPsec, and by documenting the contents of the relevant ISAKMP payloads and further specifying their semantics.

In addition to providing a profile of ISAKMP and PKIX, this document attempts to incorporate lessons learned from recent experience with both implementation and deployment, as well as the current state of related protocols and technologies.

Material from ISAKMP and PKIX is not repeated here, and readers of this document are assumed to have read and understood both documents. The requirements and security aspects of those documents are fully relevant to this document as well.

This document is organized as follows. [Section 2](#) defines special terminology used in the rest of this document, [Section 3](#) provides the profile of IKE/ISAKMP and [Section 4](#) provides the profile of PKIX. [Section 5](#) covers conventions for the out-of-band exchange of keying materials for configuration purposes.

This document is being discussed on the pki4ipsec@icsalabs.com mailing list.

Korver, Rescorla

[Page 4]Intern

[2](#). Terms and Definitions

Except for those terms which are defined immediately below, all terms used in this document are defined in either the PKIX, ISAKMP, or DOI [\[DOI\]](#) documents.

* Peer source address: The source address in packets from a peer. This address may be different from any addresses asserted as the "identity" of the peer.

* FQDN: Fully qualified domain name.

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](#) [[RFC2119](#)].

[3.](#) Profile of IKE/ISAKMP

[3.1.](#) Identification Payload

The Identification (ID) Payload is used to indicate the identity that the agent claims to be speaking for. The receiving agent can then use the ID as a lookup key for policy and whatever certificate store or directory that it has available. Our primary concern in this document is to profile the ID payload so that it can be safely used to generate or lookup policy. IKE mandates the use of the ID payload in Phase 1.

The [[DOI](#)] defines the 11 types of Identification Data that can be used and specifies the syntax for these types. These are discussed below in detail.

The ID payload requirements in this document cover only the portion of the explicit policy checks that deal with the Identity Payload specifically. For instance, in the case where ID does not contain an IP address, checks such as verifying that the peer source address is permitted by the relevant policy are not addressed here as they are out of the scope of this document.

Implementations SHOULD populate ID with identity information that is contained within the end entity certificate. This enables recipients to use ID as a lookup key to find the peer end entity certificate. The only case where implementations MAY populate ID with information that is not contained in the end entity certificate is when ID contains the peer source address (a single address, not a subnet or range). This means that implementations MUST be able to map a peer source address to a peer end entity certificate, even when the certificate does not contain that address. The exact method for performing this mapping is out of the scope of this document.

Because implementations may use ID as a lookup key to determine which policy to use, all implementations MUST be especially careful to verify the truthfulness of the contents by verifying that they correspond to some keying material demonstrably held by the peer. Failure to do so may result in the use of an inappropriate or insecure policy. The following sections describe the methods for performing this binding.

The following table summarizes the binding of the Identification Payload to the contents of end-entity certificates and of identity information to policy.

ID type	Support for send	Correspond PKIX Attrib	Cert matching	SPD lookup rules
IP*_ADDR	MUST [1]	SubjAltName ipAddress	MUST [2]	MUST [3]
FQDN	MUST [1]	SubjAltName dNSName	MUST [2]	MUST [3]
USER_FQDN	MUST [1]	SubjAltName rfc822Name	MUST [2]	MUST [3]
DN	MUST [1]	Entire Subject, bitwise compare	MUST [2]	MUST support lookup on any combination of C, CN, O, or OU
IP range	MUST NOT	n/a	n/a	n/a
KEY_ID	MUST NOT	n/a	n/a	n/a

[1] = MUST be able to send based on local configuration.

[2] = The ID in the ID payload MUST match the contents of the corresponding field (listed) in the certificate exactly, with no other lookup. The matched ID MAY be used for SPD lookup, but is not required to be used for this.

[3] = MUST be able to support exact matching in the SPD, but MAY also support substring or wildcard matches.

When sending an IPV4_ADDR, IPV6_ADDR, FQDN, or USER_FQDN, implementations MUST be configurable to send the same string as

Korver, Rescorla

[Page 6]Intern

appears in the corresponding SubjectAltName attribute. Recipients MAY use wildcards to do the SPD matching.

When sending a DN as ID, implementations MUST send the entire DN in ID. Recipients MAY perform SPD lookup based on some combination of C, CN, O, OU. Implementations MUST at a minimum be configurable to match on any combination of those 4 attributes. Implementations MAY support matching using other DN attributes in any combination, including the entire DN.

3.1.1. ID_IPV4_ADDR and ID_IPV6_ADDR

Implementations MUST support either the ID_IPV4_ADDR or ID_IPV6_ADDR ID type. These addresses MUST be stored in "network byte order," as specified in [\[RFC791\]](#). The least significant bit (LSB) of each octet is the LSB of the corresponding byte in the network address. For the ID_IPV4_ADDR type, the payload MUST contain exactly four octets [\[RFC791\]](#). For the ID_IPV6_ADDR type, the payload MUST contain exactly sixteen octets [\[RFC1883\]](#). When comparing the contents of ID with the ipAddress field in the subjectAltName extension for equality, binary comparison MUST be performed.

Implementations MUST verify that the address contained in ID is the same as the peer source address. If the end entity certificate contains address identities, then the peer source address must match at least one of those identities. If either of the above do not match, this MUST be treated as an error and security association setup MUST be aborted. This event SHOULD be auditable. In addition, implementations MUST allow administrators to configure a local policy that requires that the peer source address exist in the certificate. Implementations SHOULD allow administrators to configure a local

policy that does not enforce this requirement.

Implementations MAY use the IP address found in the header of packets received from the peer to lookup the policy, but such implementations MUST still perform verification of the ID payload. Although packet IP addresses are inherently untrustworthy and must therefore be independently verified, it is often useful to use the apparent IP address of the peer to locate a general class of policies that will be used until the mandatory identity-based policy lookup can be performed.

For instance, if the IP address of the peer is unrecognized, a VPN gateway device might load a general "road warrior" policy that specifies a particular CA that is trusted to issue certificates which contain a valid rfc822Name which can be used by that implementation

Korver, Rescorla

[Page 7]Intern

to perform authorization based on access control lists (ACLs) after the peer's certificate has been validated. The rfc822Name can then be used to determine the policy that provides specific authorization to access resources (such as IP addresses, ports, and so forth).

As another example, if the IP address of the peer is recognized to be a known peer VPN endpoint, policy may be determined using that address, but until the identity (address) is validated by validating the peer certificate, the policy MUST NOT be used to authorize any IPsec traffic. Whether the address need appear as an identity in the certificate is a matter of local policy, and SHOULD be configurable by an administrator.

As a general comment, however, it may be easier to spoof the contents of an ID payload than it is to spoof a peer source address because the peer source address must exist on the route to the peer, while ID can contain essentially random identification information. Implementations MUST validate the Identity Data provided by a peer, but implementations MAY wish to favor unauthenticated peer source addresses over an unauthenticated ID for initial policy lookup.

[3.1.2.](#) ID_FQDN

Implementations MUST support the ID_FQDN ID type, generally to support host-based access control lists for hosts without fixed IP addresses. However, implementations SHOULD NOT use the DNS to map the FQDN to IP addresses for input into any policy decisions, unless that mapping is known to be secure, such as when [\[DNSSEC\]](#) is employed. When comparing the contents of ID with the `dnsName` field in the `subjectAltName` extension for equality, caseless string comparison MUST be performed. Substring, wildcard, or regular expression matching MUST NOT be performed.

Implementations MUST verify that the identity contained in the ID payload matches identity information contained in the peer end entity certificate, in the `subjectAltName` extension. If there is not a match, this MUST be treated as an error and security association setup MUST be aborted. This event SHOULD be auditable.

[3.1.3.](#) ID_USER_FQDN

Implementations MUST support the ID_USER_FQDN ID type, generally to support user-based access control lists for users without fixed IP addresses. However, implementations SHOULD NOT use the DNS to map the FQDN portion to IP addresses for input into any policy decisions, unless that mapping is known to be secure, such as when [\[DNSSEC\]](#) is employed. When comparing the contents of ID with the `rfc822Name` field in the `subjectAltName` extension for equality, caseless string

comparison MUST be performed. Substring, wildcard, or regular expression matching MUST NOT be performed.

Implementations MUST verify that the identity contained in the ID payload matches identity information contained in the peer end entity certificate, in the `subjectAltName` extension. If there is not a match, this MUST be treated as an error and security association setup MUST be aborted. This event SHOULD be auditable.

[3.1.4.](#) ID_IPV4_ADDR_SUBNET, ID_IPV6_ADDR_SUBNET, ID_IPV4_ADDR_RANGE, ID_IPV6_ADDR_RANGE

As there is currently no standard method for putting address subnet or range identity information into certificates, the use of these ID types is currently undefined. Implementations MUST NOT generate these ID types.

Note that work in [\[SBGP\]](#) for defining blocks of addresses using the certificate extension identified by

id-pe-ipAddrBlock OBJECT IDENTIFIER ::= { id-pe 7 }

is experimental at this time.

[3.1.5](#). ID_DER_ASN1_DN

Implementations MUST support receiving the ID_DER_ASN1_DN ID type. Implementations MAY generate this type. Implementations which generate this type MUST populate the contents of ID with the Subject Name from the end entity certificate, and MUST do so such that a binary comparison of the two will succeed. For instance, if the certificate was erroneously created such that the encoding of the Subject Name DN varies from the constraints set by DER, that non-conformant DN MUST be used to populate the ID payload: in other words, implementations MUST NOT re-encode the DN for the purposes of making it DER if it does not appear in the certificate as DER. Implementations MUST NOT populate ID with the Subject Name from the end entity certificate if it is empty, as described in the "Subject" section of PKIX.

Implementations MUST verify that the identity contained in the ID payload matches identity information contained in the peer end entity certificate, in the Subject Name field. If there is not a match, this MUST be treated as an error and security association setup MUST be aborted. This event SHOULD be auditable.

[3.1.6.](#) ID_DER_ASN1_GN

Implementations MUST NOT generate this type.

[3.1.7.](#) ID_KEY_ID

The ID_KEY_ID type used to specify pre-shared keys and thus is out of scope.

[3.1.8.](#) Selecting an Identity from a Certificate

Implementations MUST support certificates that contain more than a single identity. In many cases a certificate will contain an identity such as an IP address in the subjectAltName extension in addition to a non-empty Subject Name.

Which identity an implementation chooses to populate ID with is a local matter. For compatibility with non-conformant implementations, implementations SHOULD populate ID with whichever identity is likely to be named in the peer's policy. In practice, this generally means IP address, FQDN, or USER-FQDN.

[3.1.9.](#) Transitively Binding Identity to Policy

In the presence of certificates that contain multiple identities, implementations SHOULD NOT assume that a peer will choose the most appropriate identity with which to populate ID. Therefore, when determining the appropriate policy, implementations SHOULD select the most appropriate identity to use from the identities contained in the certificate.

For example, imagine that a peer is configured with a certificate that contains both a non-empty Subject Name and an dNSName. Independent of which identity is used to populate ID, the host implementation MUST locate the proper policy. For instance, if ID contains the peer Subject Name, then the peer end entity certificate may be found using the Subject Name as a key. Once the certificate has been located and then validated, the dNSName in the certificate can be used to locate the appropriate policy. In other words, the Subject Name is used to find the certificate, the certificate contains the dNSName, and the dNSName is used to lookup policy.

[3.2.](#) Certificate Request Payload

The Certificate Request (CERTREQ) Payload allows an ISAKMP implementation to request that a peer provide some set of

Korver, Rescorla

[Page 10]Intern

certificates or certificate revocation lists. It is not clear from ISAKMP exactly how that set should be specified or how the peer should respond. We describe the semantics on both sides.

[3.2.1.](#) Certificate Type

The Certificate Type field identifies to the peer the type of certificate keying materials that are desired. ISAKMP defines 10 types of Certificate Data that can be requested and specifies the syntax for these types. For the purposes of this document, only the following types are relevant:

- * X.509 Certificate - Signature
- * Certificate Revocation List (CRL)
- * Authority Revocation List (ARL)
- * PKCS #7 wrapped X.509 certificate

The use of the other types:

- * X.509 Certificate - Key Exchange
- * PGP Certificate
- * DNS Signed Key
- * Kerberos Tokens
- * SPKI Certificate
- * X.509 Certificate - Attribute

are out of the scope of this document.

[3.2.2.](#) X.509 Certificate - Signature

This type requests that the end entity certificate be a signing certificate. Implementations that receive CERTREQs which contain this ID type in a context in which end entity signature certificates are not used SHOULD ignore such CERTREQs.

[3.2.3.](#) Certificate Revocation List (CRL)

ISAKMP does not support Certificate Payload sizes over approximately 64K, which is too small for many CRLs. For this and other reasons, implementations SHOULD NOT generate CERTREQs where the Certificate Type is "Certificate Revocation List (CRL)". Upon receipt of such a CERTREQ, implementations MAY ignore the request.

[3.2.4.](#) Authority Revocation List (ARL)

Implementations SHOULD NOT generate CERTREQ payloads with this type. Recipients of this type SHOULD treat it as synonymous with the CRL type.

[3.2.5.](#) PKCS #7 wrapped X.509 certificate

This ID type defines a particular encoding (not a particular certificate), some current implementations may ignore CERTREQs they receive which contain this ID type, and the authors are unaware of any implementations that generate such CERTREQ messages. Therefore, the use of this type is deprecated. Implementations SHOULD NOT require CERTREQs that contain this Certificate Type. Implementations which receive CERTREQs which contain this ID type MAY treat such payloads as synonymous with "X.509 Certificate - Signature".

[3.2.6.](#) Presence or Absence of Certificate Request Payloads

When in-band exchange of certificate keying materials is desired, implementations MUST inform the peer of this by sending at least one

CERTREQ. An implementation which does not send any CERTREQs during an exchange SHOULD NOT expect to receive any CERT payloads.

[3.2.7. Certificate Requests](#)

[3.2.7.1. Specifying Certificate Authorities](#)

Implementations MUST generate CERTREQs for every peer trust anchor that local policy explicitly deems trusted during a given exchange. Implementations MUST populate the Certificate Authority field with the Subject Name of the trust anchor, populated such that binary comparison of the Subject Name and the Certificate Authority will succeed.

Upon receipt of a CERTREQ where the Certificate Type is "X.509 Certificate - Signature", implementations MUST respond by sending each certificate in the chain from the end entity certificate up to and including the certificate whose Issuer Name matches the name specified in the Certificate Authority field. Implementations MAY send other certificates.

Note, in the case where multiple end entity certificates may be available, implementations SHOULD resort to local heuristics to determine which end entity is most appropriate to use. Such heuristics are out of the scope of this document.

[3.2.7.2. Empty Certificate Authority Field](#)

Implementations MUST NOT generate CERTREQs where the Certificate Type is "X.509 Certificate - Signature" with an empty Certificate Authority field, as this form is explicitly deprecated. Upon receipt of such a CERTREQ from a non-conformant implementation, implementations SHOULD send just the certificate chain associated

with the end entity certificate, not including any CRLs or the certificates that would be needed to validate those CRLs.

Note that PKIX prohibits certificates with an empty issuer name field.

[3.2.8](#). Robustness

[3.2.8.1](#). Unrecognized or Unsupported Certificate Types

Implementations MUST be able to deal with receiving CERTREQs with unsupported Certificate Types. Absent any recognized and supported CERTREQs, implementations MAY treat them as if they are of a supported type with the Certificate Authority field left empty, depending on local policy. ISAKMP [Section 5.10](#) "Certificate Request Payload Processing" specifies additional processing.

[3.2.8.2](#). Undecodable Certificate Authority Fields

Implementations MUST be able to deal with receiving CERTREQs with undecodable Certificate Authority fields. Implementations MAY ignore such payloads, depending on local policy. ISAKMP specifies other actions which may be taken.

[3.2.8.3](#). Ordering of Certificate Request Payloads

Implementations MUST NOT assume that CERTREQs are ordered in any way.

[3.2.9](#). Optimizations

[3.2.9.1](#). Duplicate Certificate Request Payloads

Implementations SHOULD NOT send duplicate CERTREQs during an exchange.

[3.2.9.2](#). Name Lowest 'Common' Certification Authorities

When a peer's certificate keying materials have been cached, an implementation can send a hint to the peer to elide some of the certificates the peer would normally respond with. In addition to the normal set of CERTREQs that are sent specifying the trust anchors, an

implementation MAY send CERTREQs containing the Issuer Name of the relevant cached end entity certificates. When sending these hints, it is still necessary to send the normal set of CERTREQs because the hints do not sufficiently convey all of the information required by the peer. Specifically, either the peer may not support this optimization or there may be additional chains that could be used in this context but will not be specified if only supplying the issuer

of the end entity certificate.

No special processing is required on the part of the recipient of such a CERTREQ, and the end entity certificates will still be sent. On the other hand, the recipient MAY elect to elide certificates based on receipt of such hints.

ISAKMP mandates that CERTREQs contain the Subject Name of a Certification Authority, which results in the peer always sending at least the end entity certificate. This mechanism allows implementations to determine unambiguously when a new certificate is being used by the peer, perhaps because the previous certificate has just expired, which will result in a failure because the needed keying materials are not available to validate the new end entity certificate. Implementations which implement this optimization MUST recognize when the end entity certificate has changed and respond to it by not performing this optimization when the exchange is retried.

[3.2.9.3](#). Example

Imagine that an implementation has previously received and cached the peer certificate chain TA->CA1->CA2->EE. If during a subsequent exchange this implementation sends a CERTREQ containing the Subject Name in certificate TA, this implementation is requesting that the peer send at least 3 certificates: CA1, CA2, and EE. On the other hand, if this implementation also sends a CERTREQ containing the Subject Name of CA2, the implementation is providing a hint that only 1 certificate needs to be sent: EE. Note that in this example, the fact that TA is a trust anchor should not be construed to imply that TA is a self-signed certificate.

[3.3. Certificate Payload](#)

The Certificate (CERT) Payload allows the peer to transmit a single certificate or CRL. Multiple certificates are transmitted in multiple payloads. However, not all certificate forms that are legal in PKIX make sense in the context of ISAKMP or IPsec. The issue of how to represent ISAKMP-meaningful name-forms in a certificate is especially problematic. This memo provides a profile for a subset of PKIX that makes sense for IKE/ISAKMP.

[3.3.1. Certificate Type](#)

The Certificate Type field identifies to the peer the type of certificate keying materials that are included. ISAKMP defines 10 types of Certificate Data that can be sent and specifies the syntax for these types. For the purposes of this document, only the following types are relevant:

Korver, Rescorla

[Page 14]Intern

- * X.509 Certificate - Signature
- * Certificate Revocation List (CRL)
- * Authority Revocation List (ARL)
- * PKCS #7 wrapped X.509 certificate

The use of the other types:

- * X.509 Certificate - Key Exchange
- * PGP Certificate
- * DNS Signed Key
- * Kerberos Tokens
- * SPKI Certificate
- * X.509 Certificate - Attribute

are out of the scope of this document.

[3.3.2. X.509 Certificate - Signature](#)

This type specifies that Certificate Data contains a certificate used for signing, whether an end entity signature certificate or a CA signature certificate.

[3.3.3.](#) Certificate Revocation List (CRL)

This type specifies that Certificate Data contains an X.509 CRL.

[3.3.4.](#) Authority Revocation List (ARL)

This type specifies that Certificate Data contains an X.509 CRL that applies only to CA certificates. Recipients of this type MAY treat it as synonymous with the CRL type.

[3.3.5.](#) PKCS #7 wrapped X.509 certificate

This type defines a particular encoding, not a particular certificate type. Implementations SHOULD NOT generate CERTs that contain this Certificate Type. Implementations which violate this requirement SHOULD note that this is a single certificate as specified in ISAKMP. Implementations SHOULD accept CERTs that contain this Certificate Type.

[3.3.6.](#) Certificate Payloads Not Mandatory

An implementation which does not receive any CERTREQs during an exchange SHOULD NOT send any CERT payloads, except when explicitly configured to proactively send CERT payloads in order to interoperate with non-compliant implementations. In this case, an implementation MAY send the certificate chain (not including the trust anchor)

associated with the end entity certificate. This MUST NOT be the default behavior of implementations.

Implementations which are configured to expect that a peer must

receive certificates through out-of-band means SHOULD ignore any CERTREQ messages that are received.

Implementations that receive CERTREQs from a peer which contain only unrecognized Certification Authorities SHOULD NOT continue the exchange, in order to avoid unnecessary and potentially expensive cryptographic processing.

[3.3.7.](#) Response to Multiple Certificate Authority Proposals

In response to multiple CERTREQs which contain different Certificate Authority identities, implementations MAY respond using an end entity certificate which chains to a CA that matches any of the identities provided by the peer.

[3.3.8.](#) Using Local Keying Materials

Implementations MAY elect not to use keying materials contained in a given set of CERTs if preferable keying materials are available. For instance, the contents of a CERT may be available from a previous exchange or may be available through some out-of-band means.

[3.3.9.](#) Robustness

[3.3.9.1.](#) Unrecognized or Unsupported Certificate Types

Implementations MUST be able to deal with receiving CERTs with unrecognized or unsupported Certificate Types. Implementations MAY discard such payloads, depending on local policy. ISAKMP [Section 5.10](#) "Certificate Request Payload Processing" specifies additional processing.

[3.3.9.2.](#) Undecodable Certificate Data Fields

Implementations MUST be able to deal with receiving CERTs with undecodable Certificate Data fields. Implementations MAY discard such payloads, depending on local policy. ISAKMP specifies other actions which may be taken.

[3.3.9.3.](#) Ordering of Certificate Payloads

Implementations MUST NOT assume that CERTs are ordered in any way.

[3.3.9.4](#). Duplicate Certificate Payloads

Implementations MUST support receiving multiple identical CERTs during an exchange.

[3.3.9.5](#). Irrelevant Certificates

Implementations MUST be prepared to receive certificates and CRLs which are not relevant to the current exchange. Implementations MAY discard such extraneous certificates and CRLs.

Implementations MAY send certificates which are irrelevant to an exchange. One reason for including certificates which are irrelevant to an exchange is to minimize the threat of leaking identifying information in exchanges where CERT is not encrypted. It should be noted, however, that this probably provides rather poor protection against leaking the identity.

Another reason for including certificates that seem irrelevant to an exchange is that there may be two chains from the Certificate Authority to the end entity, each of which is only valid with certain validation parameters (such as acceptable policies). Since the end entity doesn't know which parameters the relying party is using, it should send the certs needed for both chains (even if there's only one CERTREQ).

[3.3.10](#). Optimizations

[3.3.10.1](#). Duplicate Certificate Payloads

Implementations SHOULD NOT send duplicate CERTs during an exchange. Such payloads should be suppressed.

[3.3.10.2](#). Send Lowest 'Common' Certificates

When multiple CERTREQs are received which specify certificate authorities within the end entity certificate chain, implementations MAY send the shortest chain possible. However, implementations SHOULD always send the end entity certificate. See [section 3.2.9.2](#) for more discussion of this optimization.

[3.3.10.3](#). Ignore Duplicate Certificate Payloads

Implementations MAY employ local means to recognize CERTs that have been received in the past, whether part of the current exchange or not, for which keying material is available and may discard these duplicate CERTs.

[3.3.11](#). Hash Payload

IKE specifies the optional use of the Hash Payload to carry a pointer to a certificate in either of the Phase 1 public key encryption modes. This pointer is used by an implementation to locate the end entity certificate that contains the public key that a peer should use for encrypting payloads during the exchange.

Implementations SHOULD include this payload whenever the public portion of the keypair has been placed in a certificate.

[4](#). Profile of PKIX

[4.1](#). X.509 Certificates

[4.1.1. Versions](#)

Although PKIX states that "implementations SHOULD be prepared to accept any version certificate", in practice this profile requires certain extensions that necessitate the use of Version 3 certificates for all but self-signed certificates used as trust anchors. Implementations that conform to this document MAY therefore reject Version 1 and Version 2 certificates in all other cases.

[4.1.2. Subject Name](#)

[4.1.2.1. Empty Subject Name](#)

Implementations MUST accept certificates which contain an empty Subject Name field, as specified in PKIX. Identity information in such certificates will be contained entirely in the SubjectAltName extension.

[4.1.2.2. Specifying Non-FQDN Hosts in Subject Name](#)

Implementations which desire to place host names that are not intended to be processed by recipients as FQDNs (for instance "Gateway Router") in the Subject Name MUST use the commonName attribute.

While nothing prevents an FQDN, USER-FQDN, or IP address information from appearing somewhere in the Subject Name contents, such entries MUST NOT be interpreted as identity information for the purposes of matching with ID or for policy lookup.

[4.1.2.3. Specifying FQDN Host Names in Subject Name](#)

Implementations MUST NOT populate the Subject Name in place of

populating the `dnsName` field of the `SubjectAltName` extension.

[4.1.2.4. EmailAddress](#)

As specified in PKIX, implementations **MUST NOT** populate `DistinguishedNames` with the `EmailAddress` attribute.

[4.1.3. X.509 Certificate Extensions](#)

Conforming applications **MUST** recognize extensions which must or may be marked critical according to this specification. These extensions are: `KeyUsage`, `SubjectAltName`, and `BasicConstraints`.

Implementations **SHOULD** generate certificates such that the extension criticality bits are set in accordance with PKIX and this document. With respect to PKIX compliance, implementations processing certificates **MAY** ignore the value of the criticality bit for extensions that are supported by that implementation, but **MUST** support the criticality bit for extensions that are not supported by that implementation. That is, if an implementation supports (and thus is going to process) a given extension, then it isn't necessary to reject the certificate if the criticality bit is different from what PKIX states it must be. However, if an implementation does not support an extension that PKIX mandates be critical, then the implementation must reject the certificate.

implements	bit in cert	PKIX mandate	behavior
yes	true	true	ok
yes	true	false	ok or reject
yes	false	true	ok or reject
yes	false	false	ok
no	true	true	reject
no	true	false	reject
no	false	true	reject
no	false	false	ok

[4.1.3.1. AuthorityKeyIdentifier](#)

Implementations **SHOULD NOT** assume that other implementations support the `AuthorityKeyIdentifier` extension, and thus **SHOULD NOT** generate certificate hierarchies which are overly complex to process in the

absence of this extension, such as those that require possibly verifying a signature against a large number of similarly named CA

certificates in order to find the CA certificate which contains the key that was used to generate the signature.

[4.1.3.2.](#) SubjectKeyIdentifier

Implementations SHOULD NOT assume that other implementations support

the SubjectKeyIdentifier extension, and thus SHOULD NOT generate certificate hierarchies which are overly complex to process in the absence of this extension, such as those that require possibly verifying a signature against a large number of similarly named CA certificates in order to find the CA certificate which contains the key that was used to generate the signature.

[4.1.3.3.](#) KeyUsage

The meaning of the nonRepudiation bit is not defined in the context of IPsec, although implementations SHOULD interpret the nonRepudiation bit as synonymous with the digitalSignature bit. Implementations SHOULD NOT generate certificates which only assert the nonRepudiation bit.

See PKIX for general guidance on which of the other KeyUsage bits should be set in any given certificate.

[4.1.3.4.](#) PrivateKeyUsagePeriod

PKIX recommends against the use of this extension. The PrivateKeyUsageExtension is intended to be used when signatures will need to be verified long past the time when signatures using the private keypair may be generated. Since IKE SAs are short-lived relative to the intended use of this extension in addition to the fact that each signature is validated only a single time, the

usefulness of this extension in the context of IKE is unclear. Therefore, implementations MUST NOT generate certificates that contain the PrivateKeyUsagePeriod extension.

[4.1.3.5. Certificate Policies](#)

Many IPsec implementations do not currently provide support for the Certificate Policies extension. Therefore, implementations that generate certificates which contain this extension SHOULD mark the extension as non-critical.

[4.1.3.6. PolicyMappings](#)

Many implementations do not support the PolicyMappings extension.

[4.1.3.7. SubjectAltName](#)

Implementations SHOULD generate only the following GeneralName choices in the subjectAltName extension, as these choices map to legal ISAKMP Identity Payload types: rfc822Name, dNSName, or iPAddress. Although it is possible to specify any GeneralName choice in the ISAKMP Identity Payload by using the ID_DER_ASN1_GN ID type, implementations SHOULD NOT assume that a peer supports such functionality.

[4.1.3.7.1. dNSName](#)

This field MUST contain a fully qualified domain name. Implementations MUST NOT generate names that contain wildcards.

Implementations MAY treat certificates that contain wildcards in this field as syntactically invalid.

Although this field is in the form of an FQDN, implementations SHOULD NOT assume that this field contains an FQDN that will resolve via the DNS, unless this is known by way of some out-of-band mechanism. Such a mechanism is out of the scope of this document. Implementations SHOULD NOT treat the failure to resolve as an error.

[4.1.3.7.2. iPAddress](#)

Note that although PKIX permits CIDR [[CIDR](#)] notation in the "Name Constraints" extension, PKIX explicitly prohibits using CIDR notation for conveying identity information. In other words, the CIDR notation MUST NOT be used in the subjectAltName extension.

[4.1.3.7.3. rfc822Name](#)

Although this field is in the form of an Internet mail address, implementations SHOULD NOT assume that this field contains a valid email address, unless this is known by way of some out-of-band mechanism. Such a mechanism is out of the scope of this document.

[4.1.3.8. IssuerAltName](#)

Implementations SHOULD NOT assume that other implementations support the IssuerAltName extension, and especially should not assume that information contained in this extension will be displayed to end users.

[4.1.3.9. SubjectDirectoryAttributes](#)

The SubjectDirectoryAttributes extension is intended to contain privilege information, in a manner analogous to privileges carried in Attribute Certificates. Implementations MAY ignore this extension when it is marked non-critical, as PKIX mandates.

[4.1.3.10](#). BasicConstraints

PKIX mandates that CA certificates contain this extension and that it be marked critical. Implementations SHOULD reject CA certificates that do not contain this extension. For backwards compatibility, implementations may accept such certificates if explicitly configured to do so, but the default for this setting MUST be to reject such certificates.

[4.1.3.11](#). NameConstraints

Many implementations do not support the NameConstraints extension. Since PKIX mandates that this extension be marked critical when present, implementations which intend to be maximally interoperable SHOULD NOT generate certificates which contain this extension.

[4.1.3.12](#). PolicyConstraints

Many implementations do not support the PolicyConstraints extension. Since PKIX mandates that this extension be marked critical when present, implementations which intend to be maximally interoperable SHOULD NOT generate certificates which contain this extension.

[4.1.3.13](#). ExtendedKeyUsage

No ExtendedKeyUsage usages are defined specifically for IPsec, so if this extension is present and marked critical, use of this certificate for IPsec MUST be treated as an error unless the extension contains the anyExtendedKeyUsage keyPurposeID, which asserts that the certificate can be used for any purpose. Implementations MAY ignore this extension if it is marked non-critical. Implementations MUST NOT generate this extension in certificates which are being used for IPsec.

Note that a previous proposal for the use of three ExtendedKeyUsage values is obsolete and explicitly deprecated by this specification. For historical reference, those values were id-kp-ipsecEndSystem, id-kp-ipsecTunnel, and id-kp-ipsecUser.

[4.1.3.14](#). CRLDistributionPoints

Receiving CRLs in band via IKE/ISAKMP does not alleviate the requirement to process the CRLDistributionPoints if the certificate being validated contains the extension and the CRL being used to validate the certificate contains the IssuingDistributionPoint extension. Failure to validate the CRLDistributionPoints/IssuingDistributionPoint pair can result in CRL substitution where an entity knowingly substitutes a known good CRL from a different distribution point for the CRL which is supposed to be used which would show the entity as revoked.

Implementations MUST support validating that the contents of CRLDistributionPoints match those of the IssuingDistributionPoint to prevent CRL substitution when the issuing CA is using them. At least one CA is known to default to this type of CRL use. See [section 4.2.2.5](#) for more information.

See PKIX docs for CRLDistributionPoints intellectual rights information. Note that both the CRLDistributionPoints and IssuingDistributionPoint extensions are RECOMMENDED but not REQUIRED by PKIX, so there is no requirement to license any IPR.

[4.1.3.15](#). InhibitAnyPolicy

Many implementations do not support the InhibitAnyPolicy extension. Since PKIX mandates that this extension be marked critical when present, implementations which intend to be maximally interoperable SHOULD NOT generate certificates which contain this extension.

[4.1.3.16](#). FreshestCRL

Implementations MUST NOT assume that the FreshestCRL extension will exist in peer extensions. Note that most implementations do not support delta CRLs.

[4.1.3.17](#). AuthorityInfoAccess

PKIX defines the AuthorityInfoAccess extension, which is used to indicate "how to access CA information and services for the issuer of the certificate in which the extension appears." Conformant implementations MAY support this extension.

[4.1.3.18](#). SubjectInfoAccess

PKIX defines the SubjectInfoAccess private certificate extension, which is used to indicate "how to access information and services for

the subject of the certificate in which the extension appears." This extension has no known use in the context of IPsec. Conformant implementations SHOULD ignore this extension when present.

[4.2](#). X.509 Certificate Revocation Lists

When validating certificates, implementations MUST make use of certificate revocation information, and SHOULD support such revocation information in the form of CRLs, unless non-CRL revocation information is known to be the only method for transmitting this information. Implementations MAY provide a configuration option to disable use of certain types of revocation information, but that option MUST be off by default.

[4.2.1](#). Multiple Sources of Certificate Revocation Information

Implementations which support multiple sources of obtaining certificate revocation information MUST act conservatively when the information provided by these sources is inconsistent: when a certificate is reported as revoked by one source, the certificate MUST be considered revoked.

[4.2.2.](#) X.509 Certificate Revocation List Extensions

[4.2.2.1.](#) AuthorityKeyIdentifier

Implementations SHOULD NOT assume that other implementations support the AuthorityKeyIdentifier extension, and thus SHOULD NOT generate certificate hierarchies which are overly complex to process in the absence of this extension.

[4.2.2.2.](#) IssuerAltName

Implementations SHOULD NOT assume that other implementations support the IssuerAltName extension, and especially should not assume that information contained in this extension will be displayed to end users.

[4.2.2.3.](#) CRLNumber

As stated in PKIX, all issuers conforming to PKIX MUST include this extension in all CRLs.

[4.2.2.4.](#) DeltaCRLIndicator

[4.2.2.4.1.](#) If Delta CRLs Are Unsupported

Implementations that do not support delta CRLs MUST reject CRLs which contain the DeltaCRLIndicator (which MUST be marked critical according to PKIX) and MUST make use of a base CRL if it is available. Such implementations MUST ensure that a delta CRL does not "overwrite" a base CRL, for instance in the keying material database.

[4.2.2.4.2](#). Delta CRL Recommendations

Since some implementations that do not support delta CRLs may behave incorrectly or insecurely when presented with delta CRLs, implementations SHOULD consider whether issuing delta CRLs increases security before issuing such CRLs.

The authors are aware of several implementations which behave in an incorrect or insecure manner when presented with delta CRLs. See [Appendix B](#) for a description of the issue. Therefore, this specification RECOMMENDS against issuing delta CRLs at this time. On the other hand, failure to issue delta CRLs exposes a larger window of vulnerability. See the Security Considerations section of PKIX for additional discussion. Implementors as well as administrators are encouraged to consider these issues.

[4.2.2.5](#). IssuingDistributionPoint

A CA that is using CRLDistributionPoints may do so to provide many "small" CRLs, each only valid for a particular set of certificates issued by that CA. To associate a CRL with a certificate, the CA places the CRLDistributionPoints extension in the certificate, and places the IssuingDistributionPoint in the CRL. The

distributionPointName field in the CRLDistributionPoints extension MUST be identical to the distributionPoint field in the IssuingDistributionPoint extension. At least one CA is known to default to this type of CRL use. See [section 4.1.3.14](#) for more information.

[4.2.2.6](#). FreshestCRL

Given the recommendations against implementations generating delta CRLs, this specification RECOMMENDS that implementations do not populate CRLs with the FreshestCRL extension, which is used to obtain delta CRLs.

[5](#). Configuration Data Exchange Conventions

Below we present a common format for exchanging configuration data. Implementations MUST support these formats, MUST support arbitrary

whitespace at the beginning and end of any line, MUST support arbitrary line lengths, and MUST support the three line-termination disciplines: LF (US-ASCII 10), CR (US-ASCII 13), and CRLF.

[5.1.](#) Certificates

Certificates MUST be Base64 encoded and appear between the following delimiters:

-----BEGIN CERTIFICATE-----

-----END CERTIFICATE-----

[5.2.](#) Public Keys

Implementations MUST support two forms of public keys: certificates and so-called "raw" keys. Certificates should be transferred in the same form as above. A raw key is only the SubjectPublicKeyInfo portion of the certificate, and MUST be Base64 encoded and appear between the following delimiters:

-----BEGIN PUBLIC KEY-----

-----END PUBLIC KEY-----

[5.3.](#) PKCS#10 Certificate Signing Requests

A PKCS#10 [[PKCS-10](#)] Certificate Signing Request MUST be Base64 encoded and appear between the following delimiters:

-----BEGIN CERTIFICATE REQUEST-----

-----END CERTIFICATE REQUEST-----

[6. Security Considerations](#)

[6.1. Identity Payload](#)

Depending on the exchange type, ID may be passed in the clear. Administrators in some environments may wish to use the empty Certification Authority option to prevent such information from leaking, at the possible cost of some performance, although such use is discouraged.

Korver, Rescorla

[Page 26]Intern

[6.2. Certificate Request Payload](#)

The Contents of CERTREQ are not encrypted in IKE. In some environments this may leak private information. Administrators in some environments may wish to use the empty Certification Authority option to prevent such information from leaking, at the cost of performance.

[6.3. Certificate Payload](#)

Depending on the exchange type, CERTs may be passed in the clear and therefore may leak identity information.

[6.4. IKE Main Mode](#)

Implementations may not wish to respond with CERTs in the second message, thereby violating the identity protection feature of Main Mode IKE. ISAKMP allows CERTs to be included in any message, and therefore implementations may wish to respond with CERTs in a message

that offers privacy protection in this case.

7. Intellectual Property Rights

No new intellectual property rights are introduced by this document.

8. IANA Considerations

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

9. Normative References

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

[IKE] Harkins, D. and Carrel, D., "The Internet Key Exchange (IKE)", [RFC 2409](#), November 1998.

[IPSEC] Kent, S. and Atkinson, R., "Security Architecture for the Internet Protocol", [RFC 2401](#), November 1998.

[ISAKMP] Maughan, D., et. al., "Internet Security Association and Key Management Protocol (ISAKMP)", [RFC 2408](#), November 1998.

[PKCS-10] Kaliski, B., "PKCS #10: Certification Request Syntax Version 1.5", [RFC 2314](#), March 1998.

[PKIX] Housley, R., et al., "Internet X.509 Public Key

[RFC791] Postel, J., "Internet Protocol", STD 5, [RFC 791](#), September 1981.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), March 1997.

[10](#). Informational References

[CIDR] Fuller, V., et al., "Classless Inter-Domain Routing (CIDR): An Address Assignment and Aggregation Strategy", [RFC 1519](#), September 1993.

[DNSSEC] Eastlake, D., "Domain Name System Security Extensions", [RFC 2535](#), March 1999.

[RFC1883] Deering, S. and Hinden, R. "Internet Protocol, Version 6 (IPv6) Specification", [RFC 1883](#), December 1995.

[ROADMAP] Arsenault, A., and Turner, S., "PKIX Roadmap", [draft-ietf-pkix-roadmap-08.txt](#).

[SBGP] Lynn, C., Kent, S., and Seo, K., "X.509 Extensions for IP Addresses and AS Identifiers", [draft-ietf-pkix-x509-ipaddr-as-extn-00.txt](#)

[11](#). Acknowledgements

The authors would like to acknowledge the expired [draft-ietf-ipsec-pki-reg-05.txt](#) for providing valuable materials for this document. The authors would like to especially thank Greg Carter, Russ Housley, Steve Hanna, and Gregory Lebovitz for their valuable comments, some of which have been incorporated unchanged into this document.

[12](#). Author's Addresses

Brian Korver
Xythos Software, Inc.
One Bush Street, Suite 600
San Francisco, CA 94104
USA

Phone: +1 415 248-3800
EMail: brian@xythos.com

Eric Rescorla
RTFM, Inc.
2064 Edgewood Drive

Korver, Rescorla

[Page 28]Intern

Palo Alto, CA 94303
USA
Phone: +1 650 320-8549
EMail: ekr@rtfm.com

Copyright (C) The Internet Society (2004). All Rights Reserved.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to the Internet Society or other Internet organizations, except as needed for the purpose of developing Internet standards in which case the procedures for copyrights defined in the Internet Standards process must be followed, or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by the Internet Society or its successors or assigns.

This document and the information contained herein is provided on an "AS IS" basis and THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Acknowledgement

Funding for the RFC Editor function is currently provided by the Internet Society.

[Appendix A](#). Change History

* February 2004 (-04)

Minor editorial changes to clean up language

Deprecate in-band exchange of CRLs

Incorporated Gregory Lebovitz's proposal for CERT payloads:

Korver, Rescorla

[Page 29]Intern

"should deal with all the CRL, Intermediate Certs, Trust Anchors, etc OOB of IKE; MUST be able to send and receive EE cert payload; only real exception is Intermediate Certs which MAY be sent and SHOULD be able to be receivable (but in reality there are very few hierarchies in operation, so really it's a corner case); SHOULD NOT send the other stuff (CRL, Trust Anchors, etc) in cert payloads in IKE; SHOULD be able to accept the other stuff if by chance it gets sent, though we hope they don't get sent"

Incorporated comments contained in Oct 7, 2003 email from steve.hanna@sun.com to ipsec@lists.tislabs.com

Moved text from "Profile of ISAKMP" Background section to each payload section (removing duplication of these sections)

Removed "Certificate-Related Payloads in ISAKMP" section since it was not specific to IKE.

Incorporated Gregory Lebovitz's table in the "Identification Payload" section

Moved text from "binding identity to policy" sections to each payload section

Moved text from "IKE" section into now-combined "IKE/ISAKMP" section

ID_USER_FQDN and ID_FQDN promoted to MUST from MAY

Promoted sending ID_DER_ASN1_DN to MAY from SHOULD NOT, and receiving from MUST from MAY

Demoted ID_DER_ASN1_GN to MUST NOT

Demoted populating Subject Name in place of populating the dNSName from SHOULD NOT to MUST NOT and removed the text regarding domainComponent

Revocation information checking MAY now be disabled, although not by default

Aggressive Mode removed from this profile

★ June 2003 (-03)

Minor editorial changes to clean up language

Minor additional clarifying text

Removed hyphenation

Added requirement that implementations support configuration data exchange having arbitrary line lengths

* February 2003 (-02)

Word choice: move from use of "root" to "trust anchor", in accordance with PKIX

SBGP note and reference for placing address subnet and range information into certificates

Clarification of text regarding placing names of hosts into the Name commonName attribute of SubjectName

Added table to clarify text regarding processing of the certificate extension criticality bit

Added text underscoring processing requirements for CRLDistributionPoints and IssuingDistributionPoint

* October 2002, Reorganization (-01)

* June 2002, Initial Draft (-00)

[Appendix B](#). The Possible Dangers of Delta CRLs

The problem is that the CRL processing algorithm is sometimes written incorrectly with the assumption that all CRLs are base CRLs and it is assumed that CRLs will pass content validity tests. Specifically, such implementations fail to check the certificate against all possible CRLs: if the first CRL that is obtained from the keying material database fails to decode, no further revocation checks are performed for the relevant certificate. This problem is compounded by

the fact that implementations which do not understand delta CRLs may fail to decode such CRLs due to the critical DeltaCRLIndicator extension. The algorithm that is implemented in this case is approximately:

Korver, Rescorla

[Page 31]Intern

```
fetch newest CRL
check validity of CRL signature
if CRL signature is valid then
if CRL does not contain unrecognized critical extensions
and certificate is on CRL then
set certificate status to revoked
```

The authors note that a number of PKI toolkits do not even provide a method for obtaining anything but the newest CRL, which in the presence of delta CRLs may in fact be a delta CRL, not a base CRL.

Note that the above algorithm is dangerous in many ways. See PKIX for the correct algorithm.

