Library

IBM SecureWay White Paper

Public Key Infrastructure: The PKIX Reference Implementation Project (aka Jonah)

Public Key Infrastructure (PKI) services are becoming important general purpose tools for authentication, authorization, encryption and key management for digital signatures in a variety of Internet/intranet applications, including secure messaging and electronic commerce. With projected growth of business to business internet commerce going from $8 billion in 1997 to $327 billion by 2002 it is crucial that the infrastructure be in place to provide an open, secure, interoperable, and integrateable environment.

A PKI is the set of security services that enable the use and management of public-key cryptography and certificates, including key, certificate, and policy management. Electronic commerce and communication on the Internet has heightened the focus on PKI products and services, and they are becoming important tools for a variety of applications. Application encryption and authentication using Secure Sockets Layer (SSL), e-mail using Secure/Multimedia Internet Mail Extensions (S/MIME), Virtual Private Networks using ISAKMP Key Exchange (IKE) and code-signing services for both Java and ActiveX all specify the use of PKI which underscores the need.

The standard format required for the certificate is X.509v3. The X509 adoption rate is perceived to be lagging. The reality is that there has been a lot of talk and, although there is a standard for the actual certificate, there is no deployed workable PKI standard to manage these certificates.

A PKI is only useful to the degree it is integrated with an application. People are reluctant to integrate with applications until they believe they know what will be the eventual standard, and that it is available to them. Today, several vendors (including Lotus and IBM) are going their own ways, trying to follow standards, interoperate, and provide the necessary functionality. However, there is nothing stable to build on.

Technologically, Lotus with the Domino PKI is ahead of the industry. The PKI used in Domino was developed prior to the standards. Even though Lotus is aggressively adding in support for standards like X.509 and SSL, the PKI is proprietary which means we like other vendors are struggling to make our products interoperate with Entrust, Verisign, Netscape and Microsoft, just as they are struggling to make their products interoperate with each other.

Today, because there are no agreed upon standards to assure interoperability, it is highly unusual for two products to interoperate since there is no common infrastructure to test with. Unfortunately, because there is no standard PKI implementation, every product must be tested with every other product to ensure interoperability. This is time consuming and affects product time to market.

IETF

To address these issues the IETF, Internet Engineering Task Force, has formed the Public-key Infrastructure (X.509) (pkix) (http://www.ietf.org) working group. The IETF is an international

organization of network designers, operators, vendors, and researchers who continuously work to improve Internet architecture and the smooth operation of the Internet.

The task of the working group is to develop the Internet standards needed to support an X.509-based PKI. The goal of this PKI will be to facilitate the use of X.509 certificates in multiple applications which make use of the internet and to promote interoperability between different implementations choosing to make use of X.509 certificates. The resulting PKI is intended to provide a framework which will support a range of trust/hierarchy environments and a range of usage environments

To understand how this will come about, you need to understand the process.

The working group is actively working to create a standard. So, the question becomes "What is the standard?"

An internet standard is a specification (e.g. a written description) that is stable and well-understood, is technically competent, has multiple, independent, and interoperable implementations with substantial operational experience, enjoys "rough" consensus, has code that works, and is recognizably useful in some or all parts of the Internet.

The process for creating an Internet standard is that a specification undergoes a period of development and several iterations of review by the Internet community. At that point it is adopted as a standard by the appropriate body and is published. During the development of a specification, draft versions of the document are made available for informal review and comment and are placed in the IETF's "Internet-Drafts" directory.

Each distinct version of an Internet standards-related specification is published as part of the "Request for Comments" (RFC) document series. RFCs that document Internet Standards form the 'STD' subseries of the RFC series. When a specification has been adopted as a Internet Standard, it is given the additional label "STDxxx" but it keeps its RFC number and its place in the RFC series.

Once the standard or set of standards has been determined to be viable, a reference implementation may be developed using the draft, or specification, or RFC. Often, reference implementations are developed based on the draft with the purpose of speeding up the standards process. Traditionally, the implementations are made available to other vendors and users through the internet and are free.

By moving from a paper copy to actual code provides the ability for developers to test the concept with a real product and insures that at final acceptance the specification delivers instructions for developing a viable product.

A way to think of this process is to think of a builder making the decision to build a development of new homes. The first thing they would do, assuming the funding is all taken care of, is to work with an architect to develop a blueprint. They will have several iterations of the plan (draft) until they finally decide on the final blueprint (specification). The blue print defines the style, materials used, etc. When the construction begins the builder will be using the blueprint as the guide. Once the 1st home is complete the builder will have made adjustments to the blueprint to insure the house meets specifications and is appealing to the customer. This house will be the model (reference implementation) that other builders may use to compare their architecture, or even copy to reduce the time it takes them to build a house.

IBM and PKIX

The IETF working group has been working on the PKIX specification since October 1995. To date,

no commercially available reference implementation exists.

In a joint effort, IBM/Lotus/Iris have joined forces and have made a commitment to provide our customers with the highest level of interoperability independent of platform or software package. We are providing leadership within the industry to advance internet computing and ultimately drive faster acceptance/implementation of ebusiness.

To deliver on this commitment, IBM/Lotus/Iris have developed a PKIX reference implementation that will enable the development of X.509-based PKIs. This reference implementation of the IETF's PKIX standards is being donated by IBM to the Internet community. The code is being hosted by the Massachusetts Institute of Technology (MIT), who will assume responsibility for code distribution and change control. The PKIX reference implementation source code will be downloadable from the Web site of MIT after the IETF meeting in August. MIT agreed to host the code because they have done this in the past, (as an example Kerberos) and because Jeff Schiller, who is the IETF Area Director also is responsible for the MIT network, has been working closely with us through the development process.

What is the PKIX reference implementation?
The PKIX Reference implementation supplied by IBM (developed by IBM, Lotus and Iris) is based on the IETF Public Key Infrastructure (PKIX) working group's Public Key Infrastructure (PKI) specifications. This code base will form the foundation of a standard PKI to be used throughout the IBM companies.

The implementation is based on the Common Data Security Architecture (CDSA) from Intel which has been accepted as a standard by The Open Group. CDSA is the standardized framework that can support multiple trust models, certificate formats, cryptographic algorithms and certificate repositories. By basing the reference implementation on a standardized framework we are providing the capability for the customer to choose between different trust models, different key algorithms, etc.

The IETF PKIX specifications cover the certificate life-cycle: enrollment and initial certification, key-pair update, certificate update, certificate and certificate revocation list (CRL) publication, and certificate revocation.

The reference implementations provided in August and December should be considered a basis to build a feature rich certificate authority (CA) which may be considered the cornerstone of the PKI. It is basic and is not comparable on a feature-to-feature basis with CA products that are available in the market today. The purpose of this CA is to provide a base that is designed using standards that developers can use to develop products with the goal of increasing interoperability. The CA's that are available today are proprietary and while they claim to be interoperable with other CAs they have been unable to demonstrate interoperability.

The standards describe basic requirements (minimum and must-have features) and optional fields where support of these optional fields is considered value add by the customers. IBM products will incorporate this baseline reference code then enhance it with value-added features requested by their customers.

The draft specifications used to create the reference implementation code for August are:

Management Protocols:

draft-ietf-pkix-ipki3cmp	Internet X.509 Public Key Infrastructure Certificate Management Protocols
draft-ietf-pkix-crmf	Certificate Request Message Format
draft-ietf-pkix-ipki-part4	Internet X.509 Public Key Infrastructure Certificate Policy and Certificate Practices framework
draft-ietf-pkix-ipki-part1	Internet Public Key Infrastructure X.509 Certificate and CRL Profile
draft-ietf-pkix-ldapv2-schema	Internet X.509 Public Key Infrastructure LDAPv2 Schema

Later:

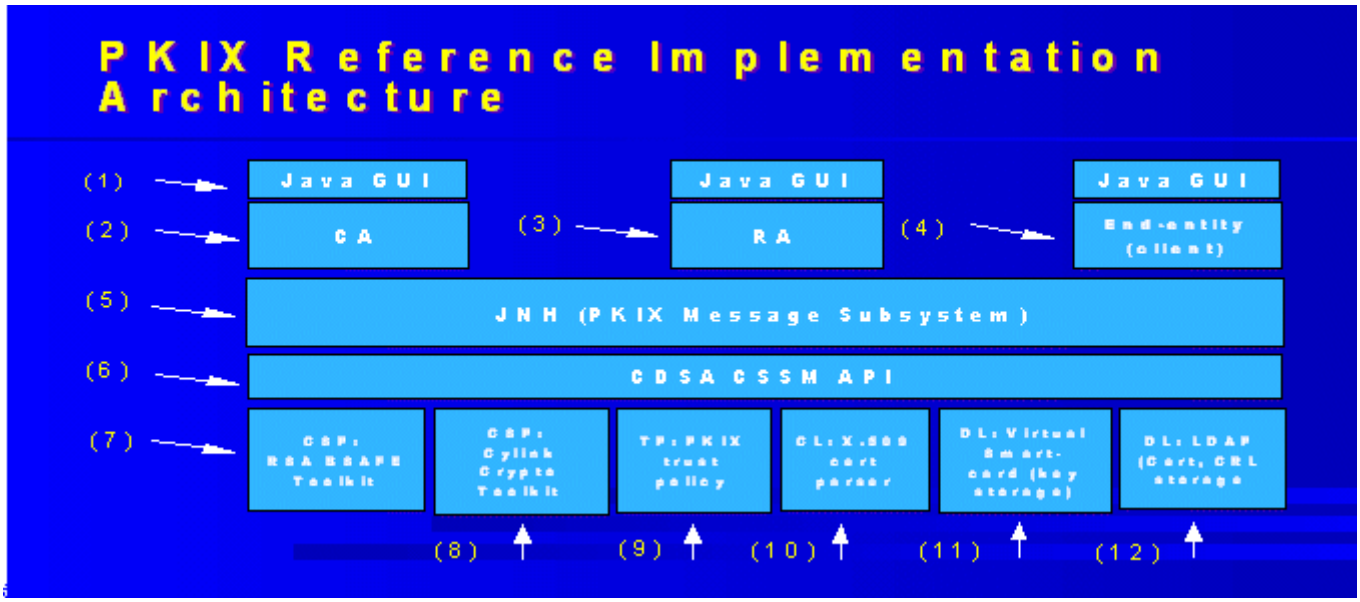
draft-ietf-smime-cert-04.txt	S/MIME Version 3 Certificate Handling
draft-ietf-pkix-cmc-00.txt	Certificate Management Messages over CMS

The specifications we chose to implement are considered the most mature and form the basic requirements of a CA. A complete list of the PKIX specifications is provided at the end of this document.

The PKI reference implementation includes the following components:

- Certification Authority (CA)
- Registration Authority (RA)
- Client (referred to in specification as End Entity - EE)
- Client PKI access library which is the directory and other repository for certificates and CRLs

The diagram below depicts the architecture for the PKIX reference implementation.



1. Java GUI - Graphical user interface designed using Java to insure portability across platforms.
2. The Certification Authority (CA) - creates and signs certificates, maintains certificate revocation lists (CRL), and provides an administrator interface for managing certificates and CRL's (authorize or refuse certificate and CRL operation). It will accommodate hierarchical models.

3. The Registration Authority (RA) - main function is to allow the CA to be operated as an off-line process, activated only when a person is present to authorize operations. It acts as an interface between the clients and the CA. When the CA is operated off-line the RA acts as an agent for the CA by buffering management requests/responses, performing cryptographic operations such as key generation or key archival on behalf of the client, authentication and issuing names to prospective PKI users, token distribution to client's , and archiving keys as required by CA policy or law. Splitting the CA and RA capability may be considered an additional security feature and a performance enhancement.

   It is designed to run either co-resident with the CA or on a separate machine. Depending on the environment and security policy the RA functionality may be implemented in the same system as the CA. A separate RA is not a requirement. The RA will have a user interface for monitoring CA operations and for controlling audit and archive settings.

   The RA will accept the PKCS10 or PKIX certificate request format. For August, we will support only PKIX, December will support both. Mail or TCP/IP protocols and diskette may be used to transmit requests to the CA.
4. Client - supplies the interface that a new user sees when they enroll with the PKI or whenever a change must be made to the user's personal security profile. Through this client the user issues requests for certificate generation, revocation, certificate lookup, certificate posting, and maintains the user's personal security profile. etc.

   For existing PKI-aware applications the client will also offer an interface for selected applications (e.g. Internet Explorer, Netscape Navigator) to load PKIX certificates into these applications. The certificate export functionality will be extensible so that support for additional specific applications can be easily added.
5. JNH (PKIX Message Subsystem) - High level API used by GUI for certificate management.
6. CDSA CSSM API - CDSA is the Common Data Security Architecture that provides a software security framework consisting of APIs designed to enhance security for applications. The Common Security Services Manager is a layer under CDSA that defines a certificate-based API for security services, and manages add-in security modules that provide services to applications.
7. CSP - Crypto Service Providers are add-in modules that perform cryptographic operations (e.g. encryption, decryption, digital signing, etc.)
8. TP - PKIX Trust Policy module that implements policies (rules or business practices). Examples of policies may be who can issue a certificate, if a certificate is issued by another CA, will another CA trust it? etc.
9. CL - Client PKI access library is a directory and other repository for certificates and CRLs. The PKI library offers application developers a way to access the PKI. There is no UI component.. Current Jonah implementation uses ASN1 class library.
10. DL - Data storage library for virtual smart card (key storage) PKCS-11interface
11. DL - Data storage library for LDAP (Certificate and CRL storage)

The Notes/Domino PKI will be be enhanced to encorporate the PKIX standards. Notes/Domino have already begun incorporating standards support (e.g. S/MIME, SSL, ETC.). Part of this integration will be to incorporate the Notes Trust model which provides support for hierarchical names within the reference implementation. This will allow customers time to migrate to the new industry standards while still ensuring the same secure business environment that Notes provides today. Features that are precursors to the full implementation will be incorporated into the R5, R5.1 and R.Next releases. Examples of these features include ability to issue X.509v3 certificates for native notes, CDSA lite, new user registration process, ability to share public and private keys with Netscape, IE, and Notes, and the Server key ring converted to server ID file.

Reference Implementation Deployment:
The PKIX reference implementation will be provided in 2 stages. The first, in August, is a snapshot and will not provide full functionality. As an example support for the DSS algorithm will be the only option supported for digital-signing but in December support for both DSS (no license required for freeware, license will be required for commercial use) and RSA (users will be required to have a RSA license) will be available.

August deliverables:

The August (snap shot) version will provide the following:

- Basic CA capability
- Basic RA capability
- Client
- Ability to create client certs extensions (such as key usage, basic constraints, name constraints (permitted subtree only), key IDs, policies)
- Ability to read certs
- Smartcard support (PKCS11) - software smartcard using PKCS11 interface. Ultimately by using PKCS11 other smartcards can be supported for storing private and personal information.

The policy of the CA, RA, and client is divided into four areas:

- Binding policy - this determines whether the binding between an attribute and public key in a certificate is considered locally trustworthy.
- Certificate policy - the policy information for certificate creation
- Certificate Revocation List policy - CRL creation, revocation, and management
- CA policy - configuration of security parameters for CA, RA, and client

Specific trust policies will include a Notes-like policy, where the trust hierarchy maps onto the Notes naming hierarchy.

1. The Binding policy determines whether the attribute to public key binding in a certificate is considered locally trustworthy.

Within the Binding policy the plan is to support the following fields and basic extensions:

	Fields
Version	Version of encoded cert
Serial number	
Signature	Identifies algorithm used by CA to sign certificate and the actual signature
Issuer	Identifies signer of certificate
Validity	Period certificate valid
Subject	Entity associated with public key stored public key field
Subject public key info	Carries the public key and identifies the algorithm with which the key is used
	Extensions
Unique Identifier	Handles the possibility of reuse of subject and/or issuer names
Key usage	Defines purpose of the key
Basic constraints	Identifies whether the subject of the certificate is a CA and depth of certification path
Name constraints: permitted subtree only	Used within the CA, indicates name space location in certification path for subject names
Key IDs	
Policies	Indicates the purpose certificate issued and may be used
Policy constraints	For certs issued to CA's, constrains path validation

2. The Certificate policy contains the information for certificate creation. Within the certificate policy the plan is to support the following fields and basic extensions:

	Fields
Version	Version of encoded cert
Serial number	
Signature	Identifies algorithm used by CA to sign certificate
Issuer	Identifies signer of certificate
Validity	Period certificate valid
Subject	Entity associated with public key stored public key field
Subject public key info	Carries the public key and identifies the algorithm with which the key is used Extensions
Unique Identifier	Handles the possibility of reuse of subject and/or issuer names
Key usage	Defines purpose of the key
Basic constraints	Identifies whether the subject of the certificate is a CA and depth of certification path
Name constraints: permitted	
subtree	
Key IDs	
Policies	Indicates the purpose certificate issued and may be used

3. CRL policy defines the rules for CRL creation and management. This contains the information on the process for CRL creation and management and revocation policy. The CRL created will be based on Version 2 of the CRL specification.

Within the CRL policy the plan is to support the following fields and extensions:

	Fields
Signature	Identifies algorithms used my CA (DSS - August, RSA - December)
Issuer name	Identifies who signed and issued CRL
This update	Issue date of CRL
Next update	Date when next regular CRL will be issued
	Extensions
Authority key ID	Identifies particular public key used to sign a CRL
CRL number	Allows users to determine when one CRL supersedes another CRL

4. CA policy - is the configuration of the security parameters for the CA, RA, and client. Examples of the type of policies that are included are:

- Length of time a pre-registration request may sit in the RA queue.
- Length of time a pre-registration request may sit in the CA queue.
- What RAs are trusted by the calling CA .
- Choices of algorithms and key sizes allowed for particular key usages (for JNH_create_keypair).
- What operations require manual approval at CA and RA.
- Authorized operators of RA and CA.
- Crypto algorithm for password based encryption (for pre-registration).
- Directory (local data storage - string name)

December Deliverables

The main areas of functionality supplied within this release are Certification trust chain handling, directory technology and access protocol, directory user, certificate, and CRL schema, revocation check policy, and CRL issuance policy

Features planned for this release are:

- Key refresh (process for instance when certificate has expired and key has not)
- CA key Rollover (process for managing CA key expiration)
- Hierarchical CA (support for Trust hierarchy)
- Cross certification (ability to certify non Notes CA)
- Additional CRL extension support (e.g. reasons, invalidity date, etc.)
- Ability for client to specify a set of policy constraints
- Create x-key usage (ability to support extensions that are not listed in the X.509 specifications)
- Netscape interoperability

Future Deliverables

Features planned to be incorprated within future IBM/Lotus product offerings, or possibly supplied through IBM partners, and not available in the freeware PKIX Reference Implementation, may include:

- Additional Certificate extensions support (e.g. alternate name support, etc.)
- Key Escrow support
- Key Recovery
- Archiving
- Advanced auditing
- CA - generated keys where server generates key versus client generated (push model)Notary services
- Delegation
- Session authentication,
- OOTB support for hardware token

RFC'S covered within PKIX working group

Internet X.509 Public Key Infrastructure Certificate and CRL Profile
This specification provides information on the format and semantics of certificates and certificate revocation lists for the internet PKI.

Internet X.509 Public Key Infrastructure Certificate Management Protocols
This specification describes a specific format for messages for X.509v3 certificates.

Internet X.509 Public Key Infrastructure Operational Protocols - LDAPv2
This specification address requirements to provide access to Public Key Infrastructure repositories for retrieving and managing PKI information. It is basedon the Lightweight Directory Acces Protocol (LDAP) v2 (RFC 1777)

Internet X.509 Public Key Infrastructure Certificate Policy and Certification Practices Framework
This specification presents a framework fo writers of certificate policies and certification practices with a comprehensive list of topics that may need to be covered. It is submitted as an Informational RFC.

Internet X.509 Public Key Infrastructure Representation of Key Exchange Algorithm (KEA) Keys in Internet X.509 Public Key Infrastructure Certificates
The Key Exchange Algorithm (KEA) is a classified algorithm for exchanging keys. This specification profiles the format and semanatics of fields in X.509v3 certificates containing KEA keys.

Internet X.509 Public Key Infrastructure Operational Protocols: FTP and HTTP
This specification covers the convention s for using File Transer Protocol (FTP) and Hypertext Transfer Protocol (HTTP) to obtain certificates and CRLs from PKI repositories.

X.509 Internet Public Key Infrastructure Online Certificate Status Protocol - OCSP
This specification defines a protocol useful in determining the current status of a digital certificate without requiring a CRL.

Internet X.509 Public Key Infrastructure Representation of Elliptic Curve Digital Signature Algorithm (ECDSA) Keys and Signatures in Internet X.509 Public Key Infrastructure Certificates
The Elliptic Curve Digital Signature Algorithm (ECDSA) is the elliptic curve version of the of the Digital Signature Algorithm key. The specification describes format and semantics of fields in X.509v3 certificates containing ECDSA keys.

Internet X.509 Certificate Request Message Format
This specification dscribes syntax used to convey a request for a certificate to a CA or RA to produce a X.509 certificate.

Internet X.509 Public Key Infrastructure Certificate Management Message Formats
This specification defines message formats used between a PKI client and PKI server or or service (where a PKI is defined to encompass the roles of Registration Authority and Certification Authority).

Internet X.509 Certificate Management Messages over CMS
This specification describes a an alternative format (to CMP listed above) for messages when using S/MIME Mandatory requirements are established which facilitate automated interoperability between a wide variety of PKI clients and servers or services.

Internet X.509 Public Key Infrastructure LDAPv2 Schema
The specification describes the minimal schema to support PKIX in an LDAPv2 environment. Only PKIX-specific components are specified here.

Internet Public Key Infrastructure Caching the Online Certificate Status Protocol
The Online Certificate Status Protocol [OCSP] specifies how a client process may obtain certificate status information online from a server.

WEB based Certificate Access Protocol-- WebCAP/1.0
This specification defines a set of methods, headers, and content-types for HTTP/1.1 to publish, retrieve X.509 certificates and Certificate Revocation Lists. This protocol also facilitates determining current status of a digital certificate without the use of CRLs.

Internet X.509 Public Key Infrastructure ENHANCED CRL DISTRIBUTION OPTIONS
This specification proposes enhancements to the X.509 CRL mechanism used to determine if a public-key certificate is valid or revoked. Examples of these enhancements include: reducing the need for unnecessarily fetching unchanged CRLs, improving timeliness, accommodating dynamic partitioning , as opposed to fixed partitioning; and better support of certificates in multiple environments with different CRL stores.