An LDAP/X.500 based distributed PGP Keyserver

First PGP Keyserver Manager Symposium
22.-23. May 2000, Utrecht

Peter Gietz
Peter.gietz@directory.dfn.de
Agenda

- PKI and Directory
  - X.500
  - LDAP

- PGP Keyserver
  - The current PKSD and its problems
  - New concepts

- Directory based PGP keyserver
  - Standardization process
  - Status
  - Objectclasses
PKI and Directory

The Burton Group:

Network Strategy Report, PKI Architecture, July 1997:

“... Customers should always consider PKI a directory-enabled set of services and infrastructure. Without directory services, PKI will be exponentially harder to implement and manage. Consequently, customers should’t deploy PKI widely without an accompanying directory plan”
Directory as Key Server Requirements

- Publishing medium for public keys and certificates
- Gets public keys from user
- Gets certificates from CA
- Documents revocation of keys/certificates (CRL)
- Documents status of a certificate at a specific time
What is Directory?

- X.500 Database standard
- Worldwide distributed data
- All data accessible worldwide
- Hierarchical organized data tree
- Objectoriented design (inheritage of objectclasses)
- Extensible data model - anything goes
Directory Information Tree (DIT)

DN: c=NL,o=University,cn=Mister X
LDAP

- Lightweight Directory Access Protocol
- Current version: 3
- IETF standard (RFC 2251-2256)
- Not anymore only access protocol, but a full blown client server system
- All Directory implementations have LDAP interface (X.500 products, Novell NDS, M$ Active Directory)
- Lots of client applications have LDAP interface (mail user agents, browser, PGP software)
PGP key server

- First only replication of pubring via email
- Marc Horowitz Keyserver (PKSD)
  - Started 1995
  - Own database backend
  - Email and HTTP interface
  - Operational model (add, revoke, etc.)
  - Net of server
  - Every server has all keys
  - Server synchronisation via email
PKSD Problems

- No distributed system: all keys on all server
- Permanent server synchronisation causes high bandwidth usage
- Chaos when one server is down (bouncing emails)
- No guarantee that a key is replicated on all server
- Not scalable
Problems of the Web of trust

- Most keys only selfsigned (=islands of trust)
- The web of trust is only existing for people belonging to certain inner circles
- Many users don’t know what they are signing
- Even at IETF Key signing parties there is no proof of identity
New concepts for PGP key server

- PKSD with enhanced backend (Open Keyserver from Highware)
- Keyserver based on DNSSec (www.ietf.org/html-charter/dnssec-charter.html)
- Keyserver based on LDAP (PGP Certificate Server from NAI)
LDAP PGP-Keyserver requirements

- Standardizes solution
  - data model
  - operational model
- Keys searchable by different criteria
- Certification path followable
- Key status retrievable
Process of standardization

• 1994 Draft from Roland Hedberg
• 1994 proprietary solution in Tübingen
• Both models fail to include more than one certificate in a person’s entry
• 1998 new initiative by DANTE
• DDS and University of Stuttgart take part in the discussion and announce an Internet Draft
• Roadmap: Draft in Summer 2000
Status of LDAP PGP key server

- PGP test server based on LDAP:
  - ldap://as.directory.dfn.de:11010/l=PGP Keyspace??sub?(cn=*)
  - http://as.directory.dfn.de:11011
- Policy for a service
- Definition of a data model for PGP
- Definition of a format for CAs to send certificates
- Software for storing and retrieving certificates
- A user can store his key into the server via WWW formular
The Directory Information Tree for PGP

PKI and Certs in LDAP

Peter Gietz, DFN Directory services
A PGP key displayed (1)

DFN-DIRECTORY, ROOT-CA-KEY (LowLevel: 1999–2000)  
<pgp-ca@directory.dfn.de>

UsedID:  
DFN-DIRECTORY, ROOT-CA-KEY (LowLevel: 1999–2000)  <pgp-ca@directory.dfn.de>

SchlüsselTyp:  
RSA

KeyID:  
094E0FC0

SchlüsselLänge:  
1024

Fingerprint:  
1B 22 93 13 07 3C 71 0F EA 60 C3 D1 12 33 14 03

Asci–Armored Key (base64 encoded)

-----BEGIN PGP PUBLIC KEY BLOCK-----
Version: 2.6.9a

---BEGIN DECRYPTED PGP PUBLIC KEY BLOCK------

PACKETSTART  

Packet:  

PacketType:  

PacketId:  

PacketData:  

PacketSignature:  

PacketSignatureAlgorithm:  

PacketSignatureData:  

PacketSignatureSignature:  

PacketSignatureSignatureAlgorithm:  

PacketSignatureSignatureData:  

PacketSignatureExpiration:  

PacketSignatureExpirationAlgorithm:  

PacketSignatureExpirationData:  

PacketSignatureExpirationSignature:  

PacketSignatureExpirationSignatureAlgorithm:  

PacketSignatureExpirationSignatureData:  

PacketSignatureExpirationExpiration:  

PacketSignatureExpirationExpirationAlgorithm:  

PacketSignatureExpirationExpirationData:  

PacketSignatureExpirationExpirationSignature:  

PacketSignatureExpirationExpirationSignatureAlgorithm:  

PacketSignatureExpirationExpirationSignatureData:  

PacketSignatureExpirationExpirationExpiration:  

PacketSignatureExpirationExpirationExpirationAlgorithm:  

PacketSignatureExpirationExpirationExpirationData:  

PacketSignatureExpirationExpirationExpirationSignature:  

PacketSignatureExpirationExpirationExpirationSignatureAlgorithm:  

PacketSignatureExpirationExpirationExpirationSignatureData:  

PacketSignatureExpirationExpirationExpirationExpiration:  

PacketSignatureExpirationExpirationExpirationExpirationAlgorithm:  

PacketSignatureExpirationExpirationExpirationExpirationData:  

PacketSignatureExpirationExpirationExpirationExpirationSignature:  

PacketSignatureExpirationExpirationExpirationExpirationSignatureAlgorithm:  

PacketSignatureExpirationExpirationExpirationExpirationSignatureData:  

PacketSignatureExpirationExpirationExpirationExpirationExpiration:  

PacketSignatureExpirationExpirationExpirationExpirationExpirationAlgorithm:  

PacketSignatureExpirationExpirationExpirationExpirationExpirationData:  

PacketSignatureExpirationExpirationExpirationExpirationExpirationSignature:  

PacketSignatureExpirationExpirationExpirationExpirationExpirationSignatureAlgorithm:  

PacketSignatureExpirationExpirationExpirationExpirationExpirationSignatureData:  

PacketSignatureExpirationExpirationExpirationExpirationExpirationExpiration:  

PacketSignatureExpirationExpirationExpirationExpirationExpirationExpirationAlgorithm:  

PacketSignatureExpirationExpirationExpirationExpirationExpirationExpirationData:  

PacketSignatureExpirationExpirationExpirationExpirationExpirationExpirationSignature:  

PacketSignatureExpirationExpirationExpirationExpirationExpirationExpirationSignatureAlgorithm:  

PacketSignatureExpirationExpirationExpirationExpirationExpirationExpirationSignatureData:  

PacketSignatureExpirationExpirationExpirationExpirationExpirationExpirationExpiration:  

PacketSignatureExpirationExpirationExpirationExpirationExpirationExpirationExpirationAlgorithm:  

PacketSignatureExpirationExpirationExpirationExpirationExpirationExpirationExpirationData:  

PacketSignatureExpirationExpirationExpirationExpirationExpirationExpirationExpirationSignature:  

PacketSignatureExpirationExpirationExpirationExpirationExpirationExpirationExpirationSignatureAlgorithm:  

PacketSignatureExpirationExpirationExpirationExpirationExpirationExpirationExpirationSignatureData:  

PacketSignatureExpirationExpirationExpirationExpirationExpirationExpirationExpirationExpiration:  

PacketSignatureExpirationExpirationExpirationExpirationExpirationExpirationExpirationExpirationAlgorithm:  

PacketSignatureExpirationExpirationExpirationExpirationExpirationExpirationExpirationExpirationData:  

PacketSignatureExpirationExpirationExpirationExpirationExpirationExpirationExpirationExpirationSignature:  

PacketSignatureExpirationExpirationExpirationExpirationExpirationExpirationExpirationExpirationSignatureAlgorithm:  

PacketSignatureExpirationExpirationExpirationExpirationExpirationExpirationExpirationExpirationSignatureData:  

PacketSignatureExpirationExpirationExpirationExpirationExpirationExpirationExpirationExpirationExpiration:  

PacketSignatureExpirationExpirationExpirationExpirationExpirationExpirationExpirationExpirationExpirationAlgorithm:  

PacketSignatureExpirationExpirationExpirationExpirationExpirationExpirationExpirationExpirationExpirationData:  

PacketSignatureExpirationExpirationExpirationExpirationExpirationExpirationExpirationExpirationExpirationSignature:  

PacketSignatureExpirationExpirationExpirationExpirationExpirationExpirationExpirationExpirationExpirationSignatureAlgorithm:  

PacketSignatureExpirationExpirationExpirationExpirationExpirationExpirationExpirationExpirationExpirationSignatureData:  

PacketSignatureExpirationExpirationExpirationExpirationExpirationExpirationExpirationExpirationExpirationExpiration:  

PacketSignatureExpirationExpirationExpirationExpirationExpirationExpirationExpirationExpirationExpirationExpirationAlgorithm:  

PacketSigna...
A PGP key displayed (2)

Zum Signaturschlüssel (Chain of Trust)

Signaturkey der DFN PKI

Erzeugungsdatum
1999-03-09

Verfallsdatum

Beschreibung
CA DFN Directory Services Deutschland

Name
CA DDSD

Nachname
CA DDSD

Mail-Adresse
ambix-pkisupport@directory.dfn.de

Der Schlüssel ist widerrufen:
Nein

Der Schlüssel ist invalidiert:
Nein

Status des Benutzers
CA

Erzeugungsmodus
CA

Revokation-Zertifikat bei zert. CA hinterlegt
Ja

PGP-Version
2.6.21

Verwendungszweck
Sign, Encrypt

Policy der zertifizierenden CA
http://www.directory.dfn.de/interns/ns/policy.html

DN der zertifizierenden CA
ou=CA DDSD, o=AMBIT, j=DFN, c=DE

xPGP: container-Version
Über diesen Link gelangen Sie direkt auf den gesamten Key-Server Datenbestand.

DFN-PCA, CERTIFICATION ONLY KEY (Low-Level: 1999-2000) <not-for-mail>

UserID
DFN-PCA, CERTIFICATION ONLY KEY (Low-Level: 1999-2000) <not-for-mail>

Schlüsseltyp
PSA

KeyID
F7B87B9D

Schlüssellänge
2048

FingerPrint
69 7D 72 74 B5 EC 3F 50 EA 7C AB E4 46 3F BS B2

Ascii-Armored Key (base64 encoded)

-----BEGIN DSS PUBLIC KEY BLOCK-----

---

---
Objectclasses for PGP 1: pGPKeyContainer

• must contain:
  • pGPKeyEntryName (name of the entry);
  • pGPKey (ASCII-armored key)
  • pGPUserId; pGPKeyID; pGPFingerPrint
  • pGPKeySize; pGPKeyType

• may contain:
  • pGPKeyCreateTime; pGPKeyExpireTime
  • pGPKeyRevoked (0=valid, 1=revoked)
  • pGPKeyUsage
  • pGPUserDN (DN of the directory entry of the person)
Objectclasses for PGP 2: cAInformationObject

- must contain:
  - cACertKeyLink / cACertKeyURL (DN / URL of the certifying key)
  - cADN / cAURL (DN / URL of the CA, or RA)
  - cAPolicy (URL of the CA’s policy)
  - cACRLDN / cACRLURL (DN / URL of the CA’s CRL)
Objectclasses for PGP 3: pGPServerInfo

- **must contain:**
  - cn (name of the entry, always cn=pGPServerInfo)
  - baseKeySpaceDN (DN of the PGP keyspace subtree)

- **may contain:**
  - basePendingDN (DN of the keyspace for yet pending keys)
Current problems

• PGP ServerInfo entry has to be directly underneath the root
• Current model is not similar to the X.509 Key storage model
• Will S/MIME win the race?
Addresses and Partners

• DFN Directory Services
  • http://www.directory.dfn.de
  • mailto:dirco@directory.dfn.de

• DFN PCA
  • http://www.cert.dfn.de/dfn-pca
  • mailto:dfnpca@pca.dfn.de

• University of Stuttgart CA
  • http://ca.uni-stuttgart.de
  • mailto:info@ca.uni-stuttgart.de

- Part of the OSI-Directory standard X.500
- Defines Data model, e.g.:
  - userCertificate; cACertificate
  - crossCertificatePair
  - certificateRevocationList
- Defines mechanisms for authentication
- Certificate includes DN of the user
- Certificate includes DN of the signing CA
X.509v3 (1997)

- New extension mechanism
- Predefined extensions:
  - Information about key: identifier, usage, ...
  - Policy information: certificate policy, ...
  - User and CA extensions: alternative name, ...
  - Certification path constraints
- Lots of people see X.509v3 as independent from X.500
  - Problem: hypothetical DNs
  - No proof of uniqueness

- Draft version ready (May 11, 2000)
  - Press release: http://www.itu.int/ITU-T/itu-t_news/
    sg7_x509_press.htm
- Includes verification of certificate chains with CAs from different domains and hierarchies
- Enhancements in the area of certificate revocation
- New features in attribute certificates (AC)
- Defines usage of ACs for access control and authorization
Applications of X.509 certificates

- Certificate based security on different levels:
  - Network Layer:
    - IPSec (Internet Protocol Security)
  - Transport Layer:
    - SSL (Secure Socket Layer) =
    - TLS (Transport Layer Security)
  - Application Level:
    - S/MIME (Secure Multipurpose Internet Mail Extensions)
      v3: patent free algorithms, mailing list support
    - PGP (Pretty Good Privacy), since version 6
IETF WG PKIX

• Defines an Internet PKI on basis of X.509 certificates
• Supports the following IETF security protocols:
  • S/MIME
  • TLS (=SSL)
  • IPSec
• Status:
  • 9 RFCs
  • 21 Internet Drafts
• Overview: <draft-ietf-pkix-roadmap-05.txt>
PKIX and certificate profiles

- RFC 2459 redrafted: <draft-ietf-pkix-new-part1-00.txt> defines:
  - Certificate (X.509v3 standard fields and standard extensions plus one private extension for authority information access, for e.g. validation service)
  - CRL (X.509v2 standard fields, and standard extensions)
  - Certificate path validation process, basic and extending

- <draft-ietf-pkix-acx509prof-02.txt> defines:
  - Attribute certificate profile for standard fields and extensions
  - additional attribute types
  - Attribute certificate validation
  - revocation
PKIX and certificate profiles (contd.)

- `<draft-ietf-pkix-qc-03.txt>` defines:
  - Qualified Certificate
    - as prescribed by some governmental laws
    - owner is natural person
    - unmistakable identity
    - only non-repudiation as key usage
    - ...

PKI and Certs in LDAP

Peter Gietz, DFN Directory services
PKIX LDAPv2 schema

- RFC 2587 “Internet X.509 Public Key Infrastructure LDAPv2 Schema”, defines:
  - Objectclass pkiUser with attribute userCertificate
  - Objectclass pkiCA with attributes cACertificate, certificateRevocationListList, authorityRevocationListList, crossCertificatePair
  - Objectclass cRDLDistributionPoint with attributes cn, certificateRevocationListList, authorityRevocationListList, deltaRevocationListList
  - Objectclass deltaCRL with attribute deltaRevocationListList
PKIX operational protocols

• LDAPv2: RFC 2559 defines:
  • LDAP repository read
  • LDAP repository search

• LDAPv3: <draft-ietf-pkix-ldap-v3-01.txt> defines:
  • Which v3 features are needed in PKIX
  • attributeCertificate
  • certificate matching rules

• FTP/HTTP: RFC 2585

• Limited Attribute Certificate Acquisition Protocol (LAAP) <draft-ietf-pkix-laap-00.txt>
PKIX and certificate validation

- **Simple Certification Verification Protocol (SCVP)**
  <draft-ietf-pkix-scvp-01.txt>
  - Client can offload certificate validation to a dedicated (trusted) server (validity of certificate and certification path)

- **Online Certificate Status Protocol (OCSP) RFC 2560**
  - Determination of current status of a certificate without the use of CRLs
  - Question contains cert id and time
  - Answer contains: “revoked”, “notRevoked” or “unknown”

- **OCSP Extension** <draft-ietf-pkix-ocspx-00.txt>
  - Allows client to delegate processing of certificate acceptance functions to a trusted server
LDAP work on X.509: Data model

- LDAP Object Class for Holding Certificate Information <draft-greenblatt-ldap-certinfo-schema-02.txt>
  - Introduces Objectclass certificateType
  - enables client to retrieve only those certificates that it really wants
  - contains attributes: typeName, serialNumber, issuer, validityNotBefore, validityNotAfter, subject, subjectPublicKeyInfo, certificateExtension, otherInfo
LDAP work on X.509: TLS

• LDAPv3 Extension for Transport Layer Security
  <draft-ietf-ldapext-ldapv3-tls-06.txt>
    • Extended request/response for Start TLS operation

• Authentication Methods <draft-ietf-ldapext-authmeth-04.txt>
  • Includes (as SHOULD) certificate-based authentication with TLS
  • Client uses Start TLS operation
  • Server requests client certificate
  • Client sends certificate and performs a private key based encryption
  • Client and server negotiate ciphersuite with encryption algorithm
  • Server checks validity of certificate and its CA
  • Client binds with SASL “EXTERNAL” mechanism