THE Open GROUP

Business Scenario: Directory in the Key Management Infrastructure

Issue 1

Issue 1, July 2001

Business Scenario:
Directory in the Key Management Infrastructure

Published by The Open Group
Apex Plaza, Forbury Road
Reading Berkshire
RG1 1AX, UK

www.opengroup.org

Open Group Document Number. W011
UK ISBN 1-85912-222-1
US ISBN 1-931624-02-X

> *What a difference a few more bucks for first-rate architecture make to everyone and everything it impacts.* - Malcolm Forbes

Management Summary

The goal of this business scenario is to identify the standards needed in commercial off-the-shelf directory products to realize the Key Management Infrastructure, to enable:

- Design and manufacture by product vendors, and
- Procurement by customers.

While a key management infrastructure is needed for cryptographic technologies other than public key, this scenario concentrates on the key management infrastructure for PKI.

There are strong indications that standards are needed. A pattern is developing of disjoint PKI communities; PKI can be used within, but not between, them. Off-the-shelf applications that use PKI are not using it consistently or correctly. Both of these problems are symptomatic of a lack of good standards.

But there are difficulties. This scenario identifies four major issues affecting the deployment of PKI. Until those issues are resolved – or at least better understood – it will be hard to define effective standards.

Nevertheless, this scenario does identify some standardization requirements for improving the use of Directory in the Key Management Infrastructure. They are for

- Transactionality
- Updatability
- Schema Extensibility
- Signed Directory Contents.

These requirements are in part (but only in part) the subject of existing standardization work.

Standardization in those areas will not address the basic problem of retrieval of PKI information from directories by applications. This problem must be solved for applications to use PKI consistently and correctly. There are already some standards in this area but they are not fully effective. The recommendation for immediate action arising from this scenario is to develop guidelines on the use of existing standards. These guidelines should cover

- Information Publication
- Directory Policy.

This scenario does not address requirements for Key management interfaces other than those involving Directories. Scenarios for such requirements should be developed by a body with wider scope than the Directory Interoperability Forum.

Contents

Figures

Business Scenario Problem Description

Background of the Scenario

Cryptography – the science of codes and ciphers - is widely used in today's information society. It is used not only to keep things secret, but also to tamper-proof information, to enable people and organizations to prove who they are, and to guarantee authorship by digital signature.

Cryptography requires keys. As its use increases, people and organizations need keys for more and more purposes. Their storage and management is a problem. This problem is limiting the deployment of cryptography today.

Electronic directories are potentially an important part of the solution. The ITU X.500-series recommendations define the basic model for an electronic directory. Those recommendations also define directory communications protocols, but access to directories today most commonly uses the Lightweight Directory Access Protocol (LDAP) defined by the IETF.

Purpose of the Scenario

The goal of this business scenario is to identify the standards needed in commercial off-the-shelf directory products to realize the Key Management Infrastructure, to enable

- Design and manufacture by product vendors, and
- Procurement by customers.

It is not enough just to define standards. Their interpretation must be commonly agreed if they are to be effective. As well as identifying areas where new standards are needed, the scenario aims to identify areas where standards exist but are not working.

The scenario considers the whole Key Management problem, but focuses especially on the parts of it that can be solved using directory services. It describes the business drivers and technical environment for key management. It describes the business processes and actors (both human and computer) that use it. It identifies issues for deployment of PKI. It states operational requirements for directory services in the Key Management Infrastructure. It then outlines a technical architecture, in terms of the standards that define interfaces between components. Finally, it discusses the requirements for standardization to meet the goals of the scenario.

Development of the Scenario

The scenario was developed by the Security Working Group of the Directory Interoperability forum, a part of The Open Group. There was significant input from the Security and eCommerce program of The Open Group and from the Trust Infrastructure for Europe (TIE) project of the European Commission. The main work was done at a Workshop held in Reading, UK on 25-26 January, 2001. There was further input from the Directory Interoperability Forum meeting on February 7.

Representatives of the main kinds of organization involved with key management were at the Workshop. The specific organizations were:

- Critical Path, a directory system vendor
- Hewlett-Packard, a computer system supplier
- The National Security Agency (NSA) of the USA
- The Open Group, a consortium dedicated to open systems and standards
- Shell International, a multi-national corporation
- Utimaco, a supplier of cryptographic technology
- ViaCode, the public Certificate Authority arm of the UK Post Office.

Thanks are due to those organizations and to their representatives for contributing the time and effort to make the workshop a success.

The work built on the previously developed business scenario for the Directory-Enabled Enterprise, which is part 1 of The Open Group White Paper: Assuring Interoperability for the Directory-Enabled Enterprise. That business scenario included use of PKI to control access to enterprise systems, services and information. This scenario assumes that use of PKI, among others. It develops the role of the Key Management Infrastructure in more detail.

Objectives

The Key Management Infrastructure is needed to support cryptographic systems, to provide secure reliable distribution of information over public networks.

Today, this mainly means supporting the Public Key Infrastructure (PKI). There are other cryptographic systems that must be considered, including some with symmetric keys. However, the main focus of this scenario is on PKI.

The primary business drivers in this scenario are for cryptography. If there were no need to use cryptography, there would be no need for Key Management. Given that cryptography is to be used, the Key Management Infrastructure – including Directory Services – must support it in a way that is as cost-effective and time-effective as possible.

Business Drivers for Cryptography

Management of Risk

Cryptography provides

- Certainty of identity (of person, or role, or application),
- Certainty of message content (integrity), and
- Confidentiality of message content.

This is particularly important when using public networks.

The most compelling examples of risks from lack of confidentiality come from the military sphere. For example, reading British naval communications helped the German submarines find British Atlantic convoys in 1940, and sink ships. Avoiding being sunk may not be a business driver, but it is certainly a strong motivator of another kind.

Examples of use of cryptography to minimize risk in commercial operations can be found in the Business Scenario for the Directory-Enabled Enterprise.

For a commercial organization, PKI can enable it to transfer some risk to a trusted third party (TTP). This helps it to manage the trade-off between the cost and the risk. It can employ just enough control to make the risk manageable.

Cost

Cryptography is the most cost-effective way of managing risk related to confidentiality of messages and of the key itself. It can also reduce cost by replacing paper-based methods of information storage and management.

Protecting Brand Value

The value of a brand may be damaged or destroyed by loss of confidence. Credit card companies do not publicize credit-card fraud, because to do so would discourage people from using the cards and lower their brand value. This may apply to an individual business or to a whole business sector.

For example, there was a recent case where customers of Barclay's Internet Banking service were able to see other customers' financial details; this dented confidence not only in that particular service, but also in Internet banking as a whole.

The need to preserve confidence applies to government as well as to commercial organizations.

Business Enabling

The ability reliably to use a trusted third party may enable two parties to do business where they otherwise would not be prepared to do so.

Conforming with Regulation

Legislation or other regulation may require certain information – such as health or personnel records - to be kept confidential. Where such information is transmitted over public networks, confidentiality can only be assured by cryptography.

Controlled Co-operation

There are many instances where organizations that would normally compete wish to co-operate. For example, they may form a consortium (such as the Automotive Exchange) or a joint venture.

In the last 10 years, all US military engagements have been in coalition, implying a need for co-operation.

Co-operation requires sharing of information, including in many cases directory information. For example, in NATO it is proposed to merge different elements of national and NATO directories into one shared Alliance directory.

Using cryptographic techniques, an organization can give its co-operating competitors controlled access to its systems and information.

Controlled Access

Cryptographic techniques enable organizations to provide controlled access to their information, systems and services, not only when co-operating with their peers, but during normal business operation, for example with on-line customers.

The Key Management Infrastructure

The functions of the Key Management Infrastructure include:

- Key Generation – the process of creating keys. It can be complex and technical.
- Derivation - of individual keys from a master key.
- Registration – the official association of a key with a person, organization, or other entity.

- Certificate creation – a certificate provides evidence of an association between a key and a person, organization, or other entity.
- Certificate installation - the installation process effectively puts the certificate into use.
- Storage and distribution – keys and certificates must be stored securely but available as and when needed for use.
- Status maintenance – certificates can be revoked. There may also be requirements for temporary suspension. The status of a certificate must be maintained and made available to users.
- Deregistration - breaks an association between a key and an entity. The opposite of the registration process. The certificate is removed from the active database and is no longer used for new transactions, but can still be used in connection with old transactions, for example to verify a signature.
- Archive - when a key has reached end of usage lifetime, there may still be a long-term need for decipherment and verification of material that it has been used to encrypt or protect, and for verifying digital signatures.
- Destruction – at the end of its archive period, a key is actually destroyed.
- Audit enabling – the infrastructure must provide an audit trail by which significant events can be traced in case of security breaches.

(See ISO 11770 - Parts 1, 2 and 3 for the standard definition of the Key Management Infrastructure.)

The Role of Directory

The Directory must be able to selectively and securely distribute key material to those authorized to receive it. This includes people both inside and outside the organization that owns the material. The material includes status and other information related to certificates, as well as the certificates themselves. In addition to operational aspects, there are management aspects to consider, particularly of the cryptographic capabilities of the system.

Access

It must be easy to find the right certificate. For example, Shell has many organizations around the world that manage cash. If a clerk has to deal with different banks with different certificates, how does he or she know which certificate to use to transfer cash to a particular bank?

The Key Management Infrastructure must be global.

Where access to keys and certificates is authorized, that access must be available to a wide range of applications, possibly using various protocol front-ends. For example, when the U.S. Department of Defense (DoD) interconnects with the civil agencies of the Federal Government through the Federal Bridge CA, and then to their respective trading partners, there are many different profiled protocols involved.

Lifetime

Keys and certificates must be stored over their whole life and must not be corrupted, lost or destroyed.

Elements of the U.S. Government have a 30-year retention rule. Under UK law, Board minutes must be kept in perpetuity - so some classes of documents will need to be kept in perpetuity. This implies that the keys that can verify them must be kept forever. (Or they must be kept at least until the need for verification or decryption no longer applies. For example, when encrypted information ceases to be confidential an un-encrypted version could be created for further archive).

Security

Keys and certificates must be protected absolutely from tampering in storage and transmission.

Some keys, for example, PKI private keys, must be protected absolutely from unauthorized reading.

Directories store various items of information, such as addresses, in addition to keys and certificates. The presence of a certificate in association with other information will sometimes in itself be information that must be protected.

In some cases there will also be a need to protect information that can be obtained not from a single directory entry but by aggregating a number of entries.

Views of Environments and Processes

Business Environment

Organizations

The Key Management Infrastructure is used by a wide range of organizations, including commercial organizations, government and public services, the military, service providers, and individuals.

An organization may have a number of internal divisions, each dealing separately with other organizations. For example, internal to the US DoD are the CINCS, armed services, agencies, and DoD internal agencies. They deal with other US federal government agencies, which have their own directories and PKI infrastructures. In addition, DoD agencies deal with allied partners, for example in a coalition / tactical environment.

Organizations and their divisions often have separate departments that must co-operate in a transaction, such as Legal, Sales, Purchasing, Finance, Production, Logistics (Transport, Storage), Operations (including support).

In any transaction, there will be other parties concerned as well as those directly involved. They include:

- PKI service providers
- Trusted third parties (TTPs)
- Network service providers
- Regulators (eg. government).

Some organizations will set up internal service-provider departments; others will use external service providers. The decision will depend on a cost / risk trade-off.

Roles

There is an important distinction between identity and role. A certificate binds an identity to a key, but the role of the person with that identity may change according to the context of the communication. For example, an employee has a role in his or her organization, and is also an individual with private activities.

Roles are generally not standardized, and even where different organizations use the same roles, they will typically have different names for them.

When an individual with a role in an organization signs a document, there are two aspects to the signature: the signature of the individual, and the signature in the role on behalf of the organization.

Can an individual have multiple identities? This is not impossible, but the practice in the organizations in the workshop was to say that each individual has a single, unique identity. This might be associated with physical characteristics (retina image, etc.)

An individual can have several roles. For example, an individual may need to be able to sign things in a legal sense in any one of a number of different roles, such as managing

director and finance director. Also, individuals can share roles. This is common in 24-hour operation.

Sometimes an organization will deploy PKI only internally, and can then have well-defined roles. In other cases they will use it for relations with other organizations and will then have to share part of their PKI with those organizations. Use of shared parts of the PKI will imply sharing other information: for example, if you give your signature during a business transaction, your identity becomes known.

Constraints on PKI Use

A PKI-user organization will not be able to define its own way of working and use it for all its business dealings.

Users may express a preference for business partners to use a particular service provider or technology vendor, but ultimately they will have no choice. There may be requirements from particular partners or in particular countries. Users will potentially have to deal with certificates from many different PKI platforms (Entrust, Baltimore etc) and from many different service providers using those platforms. There will be considerable variation between these offerings.

Regulations of various kinds may affect organizations' ability to use PKI. For example, Shell needs to recognize and follow US export control restrictions wherever it operates.

Technical Environment

The technical environment includes the general-purpose computing and networking devices and software in common use, and the Internet. Examples of these devices and software are Windows PCs and UNIX servers. Figure 1, taken from the Directory-Enabled Enterprise Business Scenario, illustrates the enterprise technical environment.

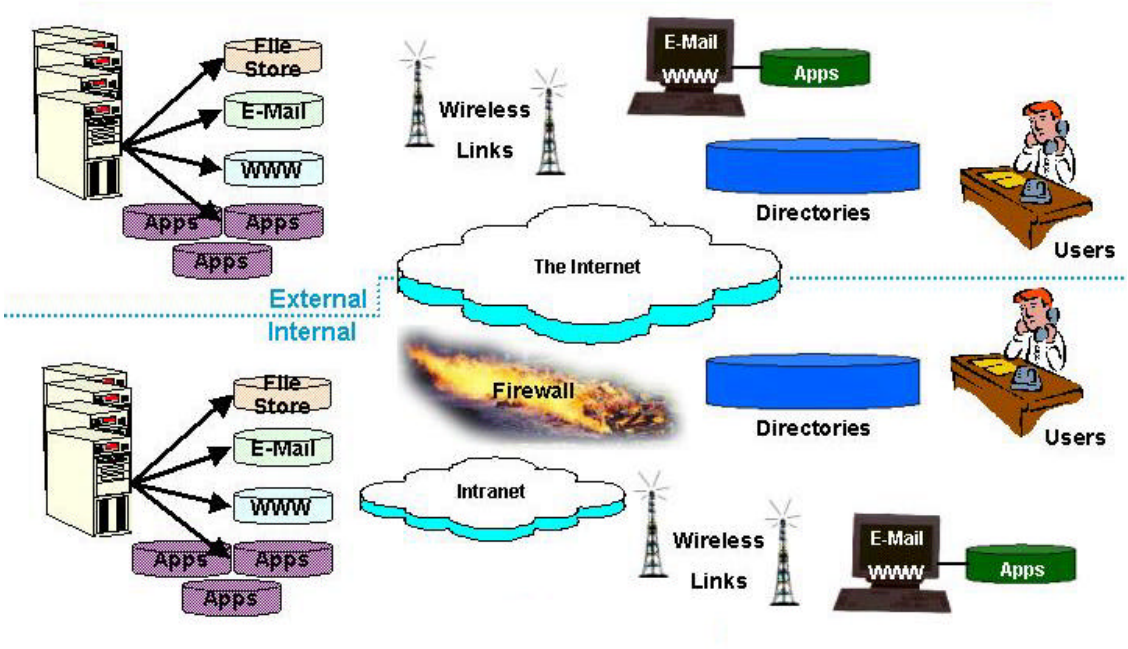

Figure 1 – The Enterprise Technical Environment

See the Directory-Enabled Enterprise Business Scenario for a more detailed description of the technical environment for use of directory by an organization.

More particularly, for this Business Scenario, the technical environment also includes cryptographic hardware and software.

Two main kinds of cryptographic systems can be distinguished:

- **Symmetric Key** - in which the same key is used to decode a message as is used to encode it, and
- **Public Key** – in which a message encoded by one key is decoded by another, associated but different, key.

(Note that public key systems generally include elements of symmetric key usage. Randomly generated symmetric keys are often used on a newly established secure association between two parties, or in the case of an encrypted message part for asynchronous messaging, and are passed between parties using public key techniques.)

While a Key Management Infrastructure is needed by systems of both kinds, their requirements are different. In particular, unless the keys are stored in encrypted form in the directory, there are more stringent access control requirements necessary in a symmetric key system to ensure that only members of the cryptonet are allowed to retrieve the key material. Asymmetric key relies on the private key being protected

close to the entity that owns it and the public key being publicly available, for example through the directory.

The requirements considered in this version of this Business Scenario are for public key systems. The extension of this scenario to cover symmetric key systems and their storage and access needs with respect to the directory service is for further study.

Deployment of public key systems requires technology, standards, and institutions that collectively form part of the Public Key Infrastructure (PKI). This infrastructure is not yet completely defined, but the basic concepts such as Certificate, Certification Authority (CA), Registration Authority (RA), Certificate Revocation List (CRL), etc., are commonly agreed.

This scenario assumes knowledge of the concepts of the PKI. For further information, refer to ITU-T Recommendation X.509, "Information Technology - Open Systems Interconnection: The Directory", Authentication Framework," IETF RFC 2527: Internet X.509 Public Key Infrastructure Certificate Policy and Certification Practices Framework," and IETF RFC 2459: "Internet X.509 Public Key Infrastructure, Certificate and CRL Profile".

Process Descriptions

There are a large number of different kinds of business transaction. Examples described in the Directory-Enabled Enterprise Business Scenario are Shipping Logistics, Joint Venture Partnership Negotiations, Tendering Auction, Enterprise E-mail, and Gas Price Change Management.

Each kind of business transaction has a defined set of roles, which are filled by individuals. These individuals use computer applications (e-mail, etc) in order to conduct the transaction. If PKI services are used, those applications must be PKI-enabled.

Within the context of this business scenario, it is impossible to exhaustively analyse all business transactions. Two example transactions have been chosen, as being representative of the general case.

(A further scenario was identified but not developed in detail. This was application-to-application file transfer, such as orders to banks for direct debits or payments.)

The examples considered are both business-to-business transactions, and assume that one business wishes to procure equipment that can be supplied by a number of other businesses. The procurement is to be on the best commercial terms, working over the Internet. In the first example, the transaction is conducted by e-mail. In the second example, it is conducted using a web portal.

In addition to developing these examples of the use of PKI, the workshop analyzed the key registration process.

Procurement via E-mail

The organization requiring the equipment advertises its requirement, for example by publishing a procurement specification on the Web. The remainder of the transaction is

conducted by e-mail. Interested parties respond and negotiate with the customer. The customer selects a supplier and places a contract.

The e-mail messages sent are tamper-proofed by adding hash digests and in some cases are signed. They may also be encrypted to ensure confidentiality; this case will be addressed below after the more basic case involving just hashing and signing has been described.

Figure 2 shows the process of sending a secure e-mail message.

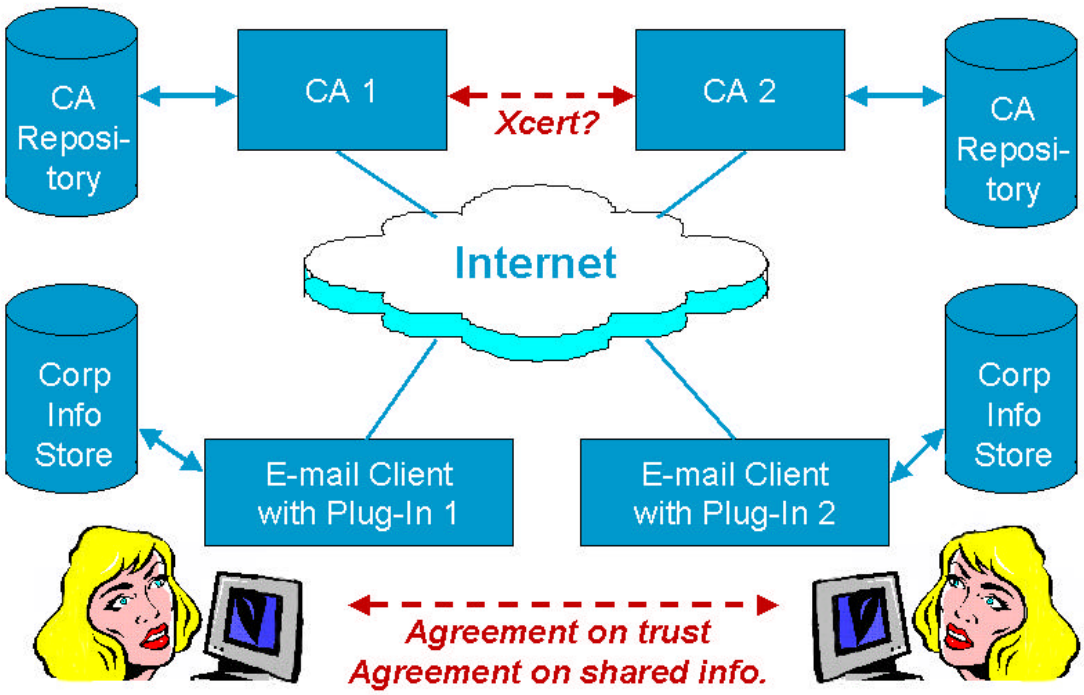

Figure 2 – Sending a Secure E-Mail Message

The message is sent, by a user who subscribes to the services of CA1, to a user who subscribes to the services of CA2. CA1 and CA2 may or may not have cross-certified.

Signature

Where signature is required, the sender's key must be used. The sender packs the message in S/MIME format, and includes with it a certificate and a certificate path to validate the certificate. (If the sender does not include the complete validation path then the receiver must re-construct it. The issues involved in path re-construction are not dealt with in this Scenario.) The sender's certificate and the certificates in the certificate path may be stored in the sender's corporate directory or in the directory of CA1.

The recipient unpacks the message and checks the whole certificate path for possible revocation. The revocation information could be kept in the directory in a CRL, or could be retrieved by OCSP. The information is provided by CA1, and is retrieved either directly from CA1's directory or OCSP server, or indirectly via a third party acting for CA1. (The figure assumes the simplest case – that the information is held in CA1's directory).

Packing of the message at the sending end, and unpacking / validation at the receiving end, are done by the sender's and the recipient's e-mail clients.

This process raises questions that at present have no satisfactory standard answers:

- A certificate path should not simply be accepted. A certificate in the path could have been revoked. How is checking to be done in a standard way? Currently, using off-the-shelf browsers, revocation CRL updates can be obtained but only manually.
- Validating a certificate path does not in itself enable the recipient to trust the message. The recipient must establish trust in the certificate at the head of the certification path. How? Root certificates are preinstalled in current off-the-shelf browsers (which is not a satisfactory trust model) or are manually imported requiring user intervention for trust decisions.
- Even when the whole certificate path, including its root, is validated, the recipient needs to establish that the sender has authority to send the message. For example, the fact that the sender is Mr. W. Mitty may be definitely proved, but is he really MegaCorp's chief purchasing officer with authority to place that two-billion-dollar contract? How is the sender's role and level of authorization to be proved? An off-the-shelf, standards-based, infrastructure for privilege management does not yet exist.

These questions are discussed further in the PKI Issues section.

Sending a message in this way implies some kind of pre-existing relationship between the parties, so that the recipient knows how to validate the certificate path and how to establish role and authorization. This relationship is most likely to be between organizations (in an Enterprise use case) rather than between parties engaged in current communication.

An approach that has been suggested in the past is to look the sender up in his or her corporate directory and obtain his or her certificate from there. However, for many organizations, this is not normal current practice.

Confidentiality

In addition to hashing and signing a message, the sender may wish to encrypt it to ensure confidentiality. In a public key system, a message is normally encrypted using the recipient's public key. The recipient can then decrypt it using the corresponding private key. (A message encrypted using the sender's private key could be decrypted by anyone with the sender's public key, not just by the recipient.)

In many organizations, the practice is for people to have separate public / private key pairs for signing and encryption.

The sender may obtain the recipient's public encryption key in several ways. The most convenient, especially where there has been no prior communication between the parties, is to retrieve it from the recipient's directory entry, either in the recipient's corporate directory or in the directory of the recipient's CA.

This raises an important question: how does the sender know where to look up the recipient's key? It might be in the directory of the recipient's CA, but how does the sender know what CA that is and what is the recipient's DN in that CA's directory?

There is a further issue: the CA is making some information about the sender public. How much information is it desirable or necessary to make public in this way? For example, inclusion of people's e-mail addresses in public directories could lead to those people being subjected to junk mail (spamming).

Procurement via Web Portal

In this case the transaction is conducted on a website. The customers and potential suppliers are registered as users of the website.

Figure 3 shows this situation.

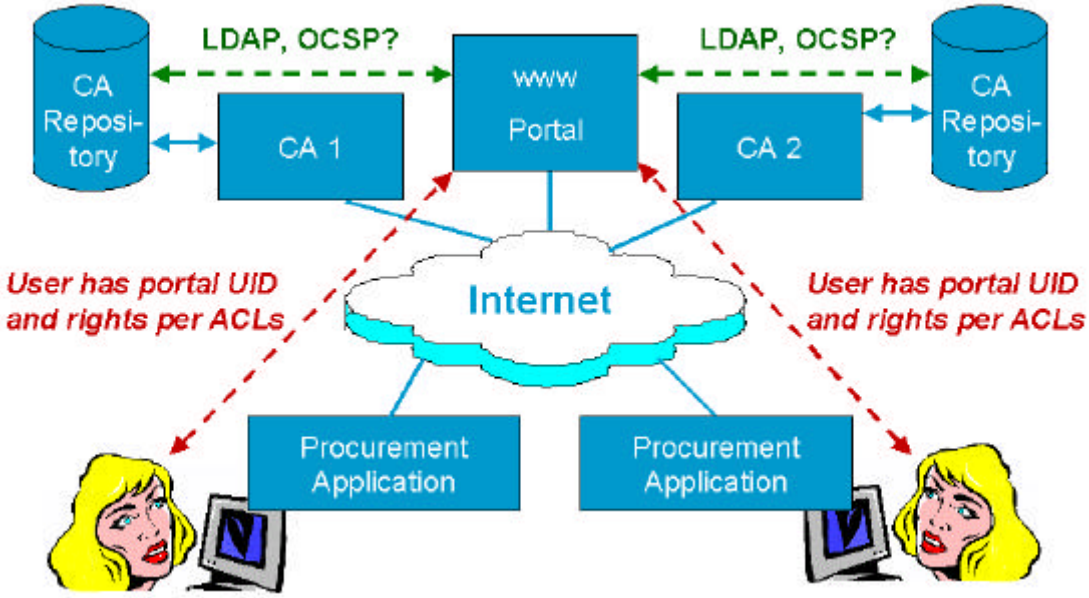

Figure 3 – Using a Procurement Portal

The purchasing organization in this case does not issue a procurement specification, but searches the web and finds a supplier organization from which it wishes to purchase. The supplier organization uses a web portal through which it publishes a catalog and allows purchase.

This system is commonly employed for web shopping without use of certificates (other than the web server's certificate which is used to establish an SSL connection). However, certificates could be used in conjunction with a portal.

In this example, we assume that certificates are used and that they are stored on hardware tokens (eg. smart cards), since it is the identity of the individual that is important, not that of the workstation.

The user has an application that reads the certificate from the smart card and transmits it to the portal. The portal has a mechanism to handle the transaction, perhaps a shopping-cart mechanism. There have to be existing relationships between the vendor and the portal and between the purchaser and the portal, but not necessarily between the vendor and the purchaser.

CA1 and CA2 are involved because they issued the vendor's and the purchaser's certificates, and the portal may need to interface to the CAs to validate them, by checking for revocation and establishing that the CA's certificate that signs them is valid.

Registration

Introduction

Registration is the official association of a key with a person, organization, or other entity. Keys are needed not only by human users but also by devices and application processes.

There can be very large numbers of keys. Shell has a single internal directory with 100,000 entries. The DoD deals with volumes of 20 million devices, not including mobile and wireless, on top of 3 or 4 million individuals.

There have to be procedural policies defined for registration. A device-registration procedure could involve people opening cabinet doors, etc. The procedure for registering a person might require showing a passport or giving a sample of DNA.

The procedural policies that a CA has for registering people and devices are important. Relying parties look at the policy to decide whether they can take the risk that someone is who they say he is. When another CA cross-certifies, it must look at the policy and consider whether its user population has a reasonable level of assurance. There are also liability issues.

Users' certificates are often stored on hardware tokens – smart cards – that they can hold and use when and where needed. A Smart Card may need to store several certificates. In addition to any internally issued logon / authentication certificate, the Smart Card may also have to store two or more public TTP certificates for signing and encryption. For example, the DoD's current requirement is for the hardware token to be able to hold a minimum of three certificates; signature, key establishment and, if appropriate, an application-specific certificate. The need for smart cards to store multiple certificates is currently both a capacity and a functionality issue.

The Registration Process

The process of registering a key is shown in Figure 4.

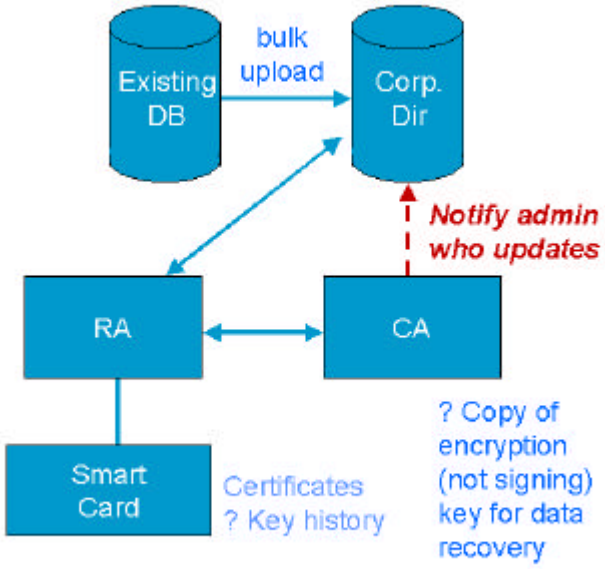

Figure 4 – Key Registration

Registration is performed by an RA using a registration system. This communicates with the CA and gives it details of key registrations.

An RA may perform registration tasks for multiple CAs. Such an RA will have to register communities with more than one CA. That could force it to use multiple registration applications rather than one. There would be a registration application for each CA, and the RA would have to select the appropriate one in each case.

The DoD has a registration system called the Real-Time Automated Personnel Identification System (RAPIDS), with operators around the world (there should be one within an hour's drive of every user location). Each user has a certificate, issued when he or she goes on civilian duty or active service. Not every individual needs to have the encryption capability, but everyone must have a unique ID. Each user has a directory entry identified by a Distinguished Name formed by appending a unique user identity to the Common Name. When the DoD user has an identity (signature) certificate issued, the subject DN is based on the DN in the Directory system. This certificate is typically not subsequently placed into the directory; rather it is only placed on the hardware token.When the DoD user receives his or her email address after registration, it may occur several weeks after the initial issuance of the identity (signature) certificate. If this is the case, the user returns to a RAPIDS system operator to have the Key Establishment certificate added to the hardware token in their possession.

In Shell's case, the plan is to allocate an email address first, and establish a bare record prior to the holder joining, then on the day of joining create the smart card.

The common pattern is as follows. A new user's directory entry is created by the owning agency / department or by HR at, or before, the point when the user joins the organization. When the user joins, he or she is given a smart card. The card initially contains at least an identity certificate. It can be updated when more information is available.

Details of existing members of the organization can be bulk-uploaded to the directory from HR records etc. prior to certificates being issued to those people.

Maintenance after Registration

The owning agency or department – not HR – typically maintains certificate status indirectly through the CA. It is important to keep information ownership at local level as far as possible.

When someone leaves the organization, it must be reported to the CA because they are the ones that must generate the addition to the CRL. The CA then posts the new CRL in the Directory.

There must be a proper procedure for this. For example, local managers might not be given the power to remove a certificate because this could be subject to abuse.

When a new CRL is issued, the certificates that it revokes should be removed from the Directory. Giving the CA access to the directory to do this is difficult – it requires access control at named attribute level, and many directories do not have this. The alternative is a cumbersome procedure in which the CA notifies the information owners, who do the update.

Even though a certificate has been revoked, the keys must be kept so that documents encrypted or signed by them can be decrypted and verified. Shell keeps a "graveyard" of revoked keys. If key history is important, a CA may keep copies of every key created so that they can be retrieved by government or by key recovery facility.

But there are confidence issues here. To keep the trust of its subscriber community, a public CA may decide not to hold any copies of encryption keys. Or there can be a compromise, where the key history is kept but in a zero knowledge form, so that it is only available to the end user, in cases where the card is lost or accidentally destroyed.

Actors and Their Roles and Responsibilities

Figure 5 summarizes the actors in the Key Management Infrastructure scenario.

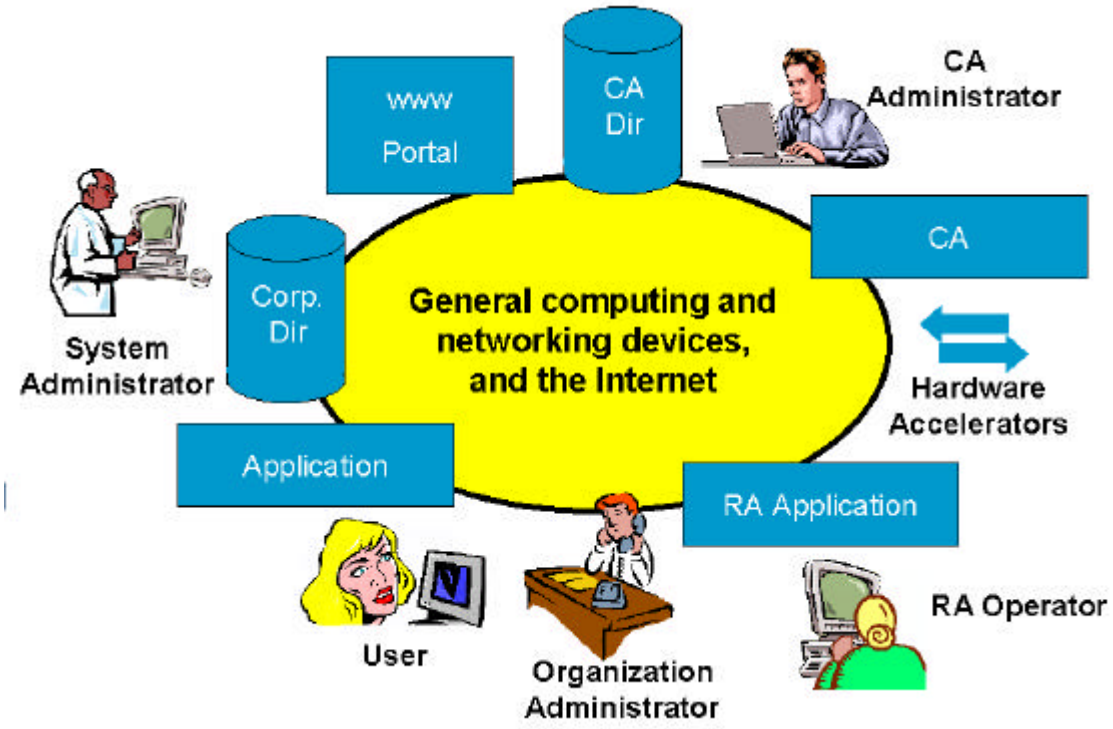

Figure 5 – The Actors

Human Actors and Roles

Actor	Role
User	• Sends e-mail • Receives e-mail • Uses web • Encrypts data, signs data, decrypts data, verifies signature • Interacts with applications (with no authentication, with simple authentication via passwords, or with strong authentication) • Selects / installs applications / downloaded software • Holds smart card • Holds, manages and protects private keys • Provides key when needed • Registers him / herself • Changes key status, recovers key history, and does other KM activities
Privileged User	• User role, plus . . . • Specific role / privileges depend on organization, eg. DoD "Comsec-Custodian" • Semantics are organization-specific – but will they remain so? • Some general roles, eg. "Company Officer" role identified in some statutory returns - identified as a named person, with named role - role might be in person's certificate - or there could be a role certificate - or name could be mapped to role internal to organization. • Privileges may be granted to people outside as well as inside the organization.

Actor	Role
Organization Administrator	• Handles billing information • Distributes management information • Obtains evidence (eg from access history) in case of dispute • Manages relationships with partners, CAs, etc. • Defines policy - Security policy - Certificate policy - Information dissemination (directory access) policy • Enforces policy.
System Administrator	• Authenticates self • Manages roles, responsibilities and access controls • Manages directory: schema design, knowledge references, shadowing / replication, directory security, misuse detection, directory contents, data-protection registration (e.g. In Europe) • Administers credentials: obtains directory, device and application process credentials, loads them into the directory and other devices, monitors key expiration and handles re-keying, handles any failures of cryptographic modules, or instances of keys being compromised, etc. • Manages hardware accelerators: set up (may need to be done in presence of auditors), monitor for faults, reconfigure when keys change, decommission / key destruction • Manages charging and billing • Different specific manager roles will have responsibility for different aspects of the above in different organizations.

Actor	Role
CA Administrator	• Administrates PKI topology and policies • Performs routine registration and revocation tasks (but this could be completely automatic)
RA Operator (but RA could be completely automatic)	• Checks credentials • Uses RA application • Authenticates to RA application • Registers subordinate RAs • Processes user certification requests (sometimes RA generates keys, sometimes user generates the keys) • Loads user smart card into system • Generates tracking / audit records (where the application doesn't do it) • May be responsible for registration with multiple CAs (because of lack of cross-certification) – this probably implies use of multiple RA applications

Note that the *organization administrator* and *system administrator* roles are not specific roles but general classes of role that will correspond to different specific roles in any particular organization. They can include management as well as simply administrative roles. The loose distinction made here is between people with responsibility for management of the organization and people with responsibility for management of the systems used by the organization. But of course in some organizations there are people with both kinds of role.

Note also that the CA and RA administrator roles vary from one CA to another, and may depend on the PKI technology in use. For example, Baltimore has two main administrator roles, "CA operator", who administrates the PKI topology and policies and "RA Operator", who performs registration tasks, and the possibility to define customized CA operator roles with restricted rights, while Entrust has five default administrator roles and distinguishes between administrators using local tools in order to administrate the CA within a secure environment, and other administrators that are allowed to work remotely:

- The Master user, who operates the CA locally,
- The Security officer who works remotely but defines and configures PKI policies
- The PKI administrator: performs the day-to-day routine registration and revocation tasks
- The directory administrator, only responsible for the DIT administration (Tree management, access control management)
- The auditor, who only reviews audit logs.

Computer Actors and Roles

Actor	Role
Application	• Mail client or browser, or special-purpose, e.g. complex workflow application • Validate certificates • Perform encryption, decryption, signing, signature verification etc. as needed • May cache certificates and CRLs
Application KMI Functionality	• Can be provided: - by modules invoked via API – but it is dangerous to assume APIs present in wide range of clients - by plug-in – plug-in can handle differences between different technology suppliers, but there is trust issue if plug-in downloaded - By downloaded components (eg. Java applets) – but they need to be trusted – see TCPA www.tcpa.org • Advantages in hard coding as little as possible, also some devices may have size limitations implying need for download. • May cache CRLs etc, user must manage this. Application design is inconsistent in this respect. • Ideally, select appropriate certificate when the directory entry contains several – but directory-matching rules for this are not widely implemented, and most applications don't do it.

Actor	Role
RA Application	• Associates id of the entity (client) with a key • Establishes connection to directory to verify or generate name etc. • Establishes connection with smart card • Either instructs card to generate key pair and export public component or writes the key pair to the card • May instruct card to export (protected) private key for key establishment (Not signature use) • Establishes connection with CA • Sends to CA public key to create the certificate and other information to populate the certificate • May send to CA recovery information (eg private key) • If doesn't use smart card, may produce alternative token and give it to user.
CA (can be completely automatic)	• Receives requests to produce certificates • Processes those requests - just signs, or - produces key-pair and signs • Creates subordinate CAs • Potentially, performs whole range of KMI activities • Publishes certificates and CRLs • Protects private keys (some activities such as key generation may be off-line for this reason) • May have n-out-of-m control (for example, any three from 16 nominated operators) for particularly sensitive operations in some contexts • May do cross-certification (in some systems RAs do this, in others CAs do it, in yet others (eg. DoD) just one CA in entire organization does it).

Actor	Role
CA Repository (Directory)	• Directory enforces uniqueness on certificate naming (DN in subject field) • May hold: - CA specific information and information related to partner CAs - CA certs, cross- certificates - user encryption key certificates (there is no reason for the directory to hold user signing key certificates) - CRLs and ARLs (A-authority) - other user attributes, not necessarily for public view, eg. qualifying information (which must be kept for audit) and addresses etc. (but maintenance and access control of this information is hard, it's easier if certificate holders maintain this information) - key history (secure place, maybe not the directory) • The minimum information in the CA store is CA contact information, CRLs, ARLs, CA certificates, and cross-certification information. The rest, including all user certificates, can go in user organization stores. The CA directory doesn't need to know about these directories. • May provide audit log, optionally digitally signed • May provide billing information

Actor	Role
Corporate Information Store (Directories)	• May hold: - user certificates and public keys (encryption keys and certificates; there is no reason for the directory to hold user signing keys.) - additional user information, such as location, in case DN doesn't convey exactly who subject is (could eg. Include serial numbers to disambiguate people with same name, eg. John Smith 001234) - information about and certificates for devices, as well as people • Different views of information may be exposed internally, externally, and publicly • Access may depend not only on who you are but where you are accessing from • Information in directory is from multiple sources (HR and departments). Categorizing it for access control can be very expensive / labor-intensive. • May provide audit log, optionally digitally signed • May provide billing information
Hardware Accelerators	• Private key decryption is a compute-intensive operation. Servers handling multiple concurrent sessions benefit from addition of accelerators. • Needed at SSL level and also at directory-bind level • Problem faced by all large organizations • CA's signing key often kept in separate hardware, for performance and also security reasons. • Not standard products, but server vendors often recommend third-party products. Non-standard interfaces, implies high integration / admin overhead.

PKI Issues

The following issues were identified during the workshop as major factors affecting the deployment of PKI.

Certificate Path Validation

To decide whether to trust a certificate, for example when a secure e-mail message is received, the user's application (perhaps with input from the user) must validate the certification path by checking the certificates in it for revocation.

Checking for revocation means going back to the directory to validate the information that has been given in the certificate path. The directory must therefore be publicly accessible. But there may be directory information for the certificate holder that should not be made public. This can be handled by:

- Making part of the contents of the directory publicly accessible while restricting access to other parts, which requires a sophisticated access control model, or
- Having two directories - one public and one private – with the public directory containing a copy of that part of the private directory's contents that need to be public.

The processing required depends on the PKI technology in use. ViaCode as a CA, for example, currently use the Entrust interface; its customers' client software communicates with it and establishes the root signing key, the certificate revocation list, and the cross-certificate revocation list, and can access search book and address list information.

Off-the-shelf applications (such as e-mail clients) typically do not perform these operations in a satisfactory manner. Traversing a path can be difficult. Some applications do not bother doing any CRL checking.

Because of this, ViaCode, and possibly other CAs, will supply plug-ins that applications can use to do PKI processing.

Typically, different CA's use different PKI technology. The user will need a different plug-in for each CA. Note that this is not just his or her own CA: it is the CA of the sender of a message that will have the information that the recipient will need to validate the certificate path, not the recipient's CA.

Establishing Trust in the Root of the Certificate Path

To decide whether to trust a certificate, the user must also establish trust in the certificate at the root of the certificate path. In the example of procurement via E-mail (see Figure 2), the possibilities include the following.

- CA1 and CA2 are the same CA. In this case the certificate at the head of the certification path is probably signed by that CA's certificate, which is trusted by the receiver.
- CA1 and CA2 are different, but have cross-certified. In this case the certificate at the head of the certification path is probably CA1's certificate; the receiver can obtain from CA2 (for example, by directory look-up) a certificate that validates it.

- The receiver makes a decision to trust CA1 and obtains from CA1 its CA certificate. This could be done by e-mail, by looking it up in CA1s directory, or by some other means. One possibility is that CA1 publishes a thumbprint of its certificate in a national newspaper, which would enable the recipient to verify this certificate if it is included in the certificate path in the message.

The root certificate required to validate a cross certificate may not be available to the user. For example, in the DoD, the root certificate is not stored on every user's desktop; it is kept on hardware tokens.

Even where the certificate is available, finding it may present a problem. There is a lack of matching rules that would enable an application to go to a directory and select a particular certificate from several possible ones. Some applications present the first or the first few available certificates to the user for selection.

It will often be appropriate to trust different certificates in different contexts. This makes it hard to automate the process of deciding whether to trust a certificate; the choice may ultimately have to be made by the end user.

Cross-Certification

Cross-certification is the process by which a CA issues a certificate for a public key belonging to another CA. It is generally a reciprocal arrangement, and the second CA will also issue a certificate for a public key of the first CA.

Cross certification is the basic mechanism that enables organizations with different CAs to do business. An organization receiving a certificate issued by another organization will trust it if it is validated by a CA that has cross-certified with the recipient's CA.

The need for cross-certification can be avoided by the user deciding to trust particular CAs. For example, the current Microsoft and Netscape browsers allow the user to decide to trust a root CA such as Verisign, and will then accept all certificates that can be validated by that root. The underlying trust model of this approach is however unsatisfactory. A user should decide to trust another user, not a CA.

The user's perception is that cross certification is largely theoretical at the moment. There is cross certification between CAs that have a common owner. For example, ViaCode operates a number of CAs, and they are cross certified with each other. But there is little cross certification between CAs operated by different organizations.

There are a number of reasons for this.

- It is difficult to define the shared liability.
- The rewards for cross certification are not great, and there are commercial disadvantages in some cases. For example, if a CA with a user population of 10 million were to cross certify with a CA that had a user population of 1000, the small CA would have access to the larger CA's population, and could undercut it cost-wise because of being small.
- Cross certification implies sharing of some directory information. This is a commercial issue – the information can be of considerable value – and also a privacy issue.

- Defining what information to share and how to share it may be a complex matter. The CAs' directories will typically contain both certificates and other information. It may not all need to be visible to both customer populations.
- When a CA cross-certifies, it is effectively varying the terms of its agreements with its customers. This requires advance notice, and possibly discussion. ViaCode, for example, give their customers 30 days notice of a decision to cross certify.
- If a CA's key became compromised, it would probably not want to tell more people than it had to. It would certainly have to tell any CAs with which it had cross certified.
- Most of the people who make the final decision on whether to cross-certify are not technical people. They are used to gentlemen's agreements. They are not equipped to make a technical decision of this nature without training.

These reasons apply to large organizations operating their own CAs, as well as to public CAs. In the DoD, the need for cross certification is reduced by having one CA (at the "Federal Bridge") that cross certifies with all the others, rather than having pair-wise cross certification between CAs.

Because of the restricted amount of cross-certification that is in place, four different kinds of PKI use are emerging.

1. Use within single company.
2. Use within a closed group (e.g. NATO, which will have a single root CA)
3. Use in the financial world (via Identrus)
4. Use among the customers of a public operator, such as ViaCode.

Use within a single company or closed user group, or among the customers of a public operator, does not require cross certification. Cross-certification - but only with other financial bodies - is a condition of entry to Identrus.

Lack of effective cross certification is a major barrier to effective use of PKI for many organizations.

Establishing Roles and Authorization

When a digital signature is given as authorization for a stage in a transaction (eg. payment), and the signature is validated by a certificate path, and the path is checked successfully, and the certificate at the root of the path is verified, the identity of the person giving the signature is assured. But it is still necessary to be sure that he or she has the authority to sign off that stage on behalf of his or her organization.

PKI does not fully address this issue and was never meant to do so.

Different organizations have different approaches to validating authority in business partners. For example, the DoD has a database in which corporations can be registered. The registration process checks corporate viability, who the chief officers are, and so on. Signature authorization levels are not yet stored in the database – this is currently on the "wish list".

Attribute certificates could provide part of the solution to this problem. (An attribute certificate certifies certain attributes of the subject, such as organizational role, in addition to the subject's identity.) The DoD has piloted them. But authorization attributes (for example) are not standardized. Also, attribute certificates are difficult to maintain as people's roles and responsibilities change. For example, Shell's world expert on some particular topic such as corrosion management or horizontal drilling may move from one job to another, but will still remain the world expert on that topic.

One possibility is that businesses could define their own extended attributes reflecting particular roles in their organizations. There are some advantages to this but, as soon as certificates become richer, the chances of interoperability recede. There would need to be agreement between PKI vendors on how extended attributes are used, as well as bilateral agreements between businesses.

Use of such certificates would mean greater complexity in the applications that use them. The applications would need to pay heed to what is in the certificate. Currently, applications are quite dumb in this respect.

The DoD proposes to have a signature certificate that covers nationality and employment affiliation. This avoids people having to trawl the personnel database to validate transactions. Employment affiliation has five separate elements. It must not be too complex, as reissuing 4 million certificates regularly would be a nightmare. There has to be some reliance on the integrity of the issuing infrastructure. There is mapping to roles but on a case-by-case basis. The role is exposed, but not all the privileges attached to that role.

The London Stock Exchange is considering an identity certificate that includes a role name, but no further role attributes. This should be sufficient for their purposes. But changing everyone's identity certificate whenever his or her role changes is not ideal.

Another possibility is to associate the certificate with the role rather than with the person. However, this implies key sharing, which is a very tedious process because of the stages and protection barriers required.

Requirements for Operation

This section describes the general requirements for operation of the Key management Infrastructure that were identified by the workshop. The requirements for standardization that arise from this scenario are discussed in the Requirements for Standardization section.

Availability

The infrastructure must be available across multiple time zones and in multiple national languages.

The requirement for worldwide availability 24 hours a day, 7 days a week, 365 days a year was established in the Directory-Enabled Enterprise Business Scenario.

Scalability

The infrastructure must cope with populations of tens of millions of keys (cf. the DoD, which needs 20 million keys for devices and applications, on top of the 3-4 million for people.)

Interoperability

All components in the solution must interoperate.

Manageability

There should be better management interfaces for management functions, in particular for administration of component management and audit information.

Auditability

There should be an audit log of key management activities. It is important to know that this has not been interfered with, so it must be digitally signed.

Better Audit production tools are needed.

Security

The Key Management Infrastructure is susceptible to a number of threats against which suitable countermeasures should be taken. These threats include:

- Disclosure of key material to unauthorized entities
- Modification of key material without authorization
- Unauthorized deletion of key material
- Incomplete destruction of keying material
- Unauthorized revocation
- Masquerade
- Delay in executing key management functions
- Misuse of keys - Use not authorized, e.g. key use after expiration, excessive key use.

See Annex A (informative) of ISO / IEC 11770-1.

Key generation and storage of private keys must be highly secure against unauthorized write and read access. Losing keys, or being unable to use them because they are compromised, is very expensive. In the DoD implementation, there is an "air gap" between the key-generation device and the rest of the system. ViaCode stores private signing keys in a separate secure store, not the directory. Security requirements extend beyond the infrastructure to the applications. In some cases applications cache CRLs. This can be dangerous. There is no consistency across applications in how they manage their caches. In some cases the user has to manage the process. It is safer to use ARLs (Authority Revocation Lists) rather than just CRLs, as ARLs are issued more promptly.

Where application plug-ins are downloaded, the user must be certain that they can be trusted. This means that the plug-in has to be signed, by a certificate that is already trusted in the infrastructure.

Accounting

There may be a need for accounting to support charging for some uses of the Key Management Infrastructure. For example, Identrus charges for certificate validations, theirs is not just a subscription-based business model.

Non-Proliferation of CAs

An organization should have to manage no more than half a dozen CA relationships at any one time to cover the globe.

Access Control Information Tools

These tools are needed to help integrate an access-control policy into a directory schema. This would help to make the whole area of privilege management, including ways to communicate the information, to be standardized and more manageable.

Programmability of Hardware Accelerators

Hardware accelerators should be programmable to support multiple algorithms, as well as remote re-key or other key management functions.

Technology Architecture Model

The goal of this business scenario is to identify the standards needed in commercial off-the-shelf directory products to realize the Key Management Infrastructure, to enable design and manufacture by product vendors, and procurement by customers.

This section of the scenario describes the technology architecture as it is commonly agreed today, concentrating on the standards defining the interfaces between architectural components, in order to shed light on the areas where further standardization is needed, or where existing standards need to be made more effective.

What should be done to address the issues raised in this section is discussed in the final section of this scenario.

Constraints

The following inhibitors were identified in the workshop.

- Regulation (some mandatory, some voluntary)
- Social / cultural
- Existing practice and agreements
- Technology
- Legislation differences
- Unaware users
- Lack of standards

Architecture Overview

Figure 6 illustrates a typical situation in which a CA and an RA provide certificate-management services to a client, using a directory to store information.

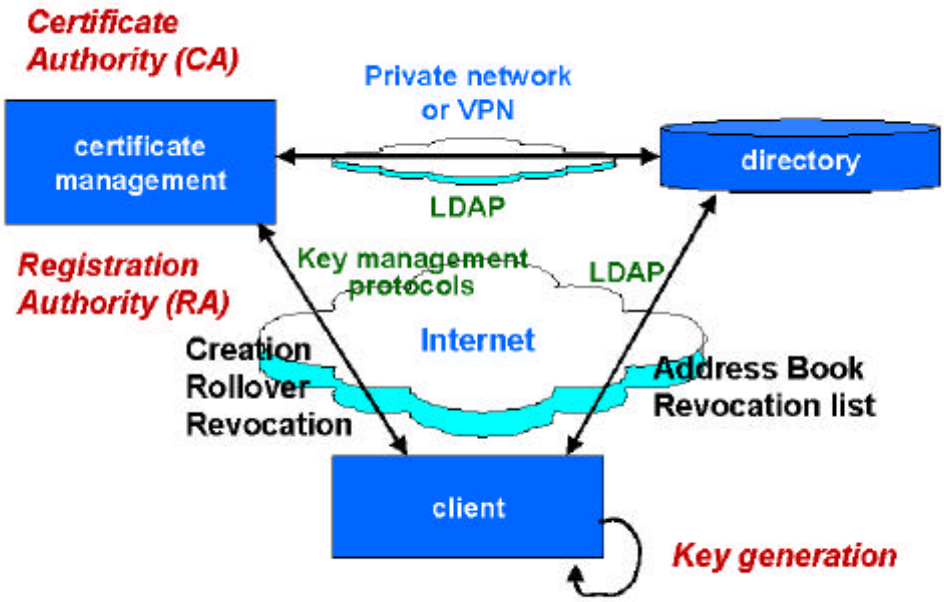

Figure 6 – Certificate Management Service Provision

The components of this model are:

- The directory
- The Client Application
- The CA Application, and
- The RA Application.

The functionality of these components is described under *Computer Actors* as *CA Repository*, *Application*, *CA*, and *RA Application*.

The interfaces between components fall into two main areas:

- Directory Access – the interfaces between the Directory and the Client Application, between the Directory and the CA Application, and between the Directory and the RA Application.
- Key Management – the interfaces between the Client Application, the CA Application, and RA Application.

Directory Access

Protocol Interfaces

The Lightweight Directory Access Protocol (LDAP) version 3 is the commonly accepted standard for directory access over the Internet. It can be used both for retrieval of

information from the directory (for example retrieval of certificates and CRLs to validate a certificate path) and storage of information in the directory (for example publication of certificates and CRLs).

LDAP version 3 is defined by the IETF in RFC 2251 and related RFCs. The Open Group LDAP 2000 Product Standard identifies the core LDAP v3 RFCs and The Open Brand for LDAP 2000 certifies conformance to them by directory servers. There are other RFCs covering extended LDAP features – for example RFC 2891: LDAP Control Extension for Server Side Sorting of Search Results. See the IETF ldapext Working Group web page at http://www.ietf.org/html.charters/ldapext-charter.html for further information.

The IETF is working on revised LDAPv3 specifications suitable for consideration as Draft Internet Standards. The work is being done in the IETF ldapbis working group; see http://www.ietf.org/html.charters/ldapbis-charter.html. While this will result in clarifications and revisions of the current standards for LDAP version 3, it is not expected to lead to major changes.

The facts that a certification program exists for LDAP version 3, and that the next version is planned to be just a minor revision, are indications of the stability and maturity of LDAP v3.

Although the access protocol is stable, there are still problems in use of Directory in the Key Management Infrastructure. These relate partly to the way directories are distributed and to the location of and interpretation of the information to be accessed via the protocol. These aspects are discussed in the following sections.

Directory Distribution

In practice, an organization's directory may be implemented on a number of different servers, which may be supplied by different vendors, and may even be of radically different kinds, as illustrated in Figure 7.

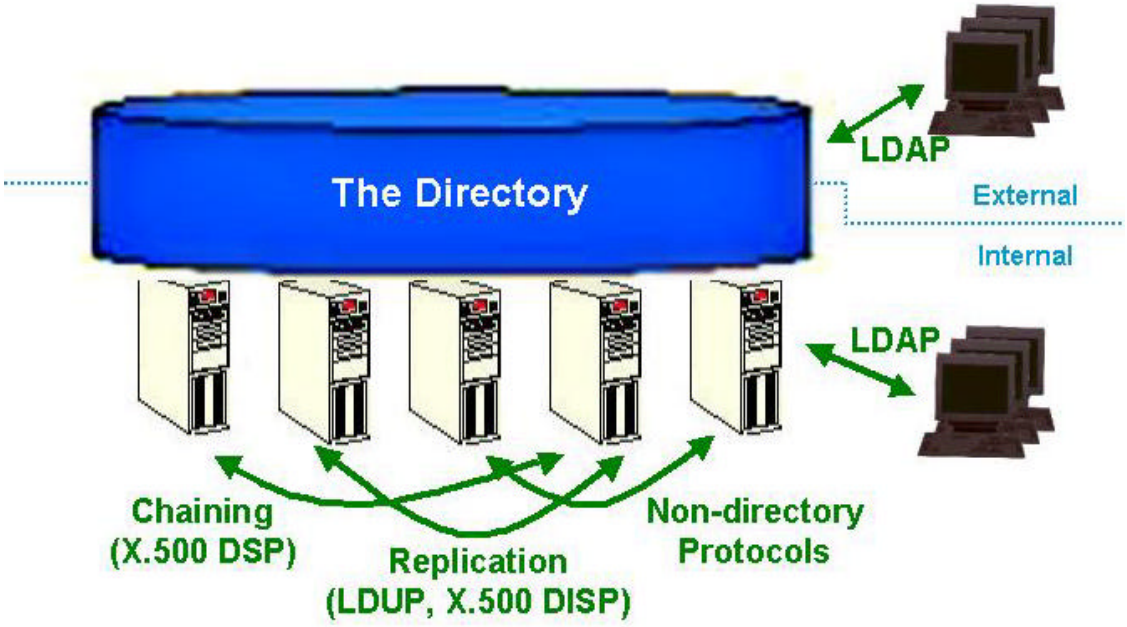

Figure 7 – Corporate Directory Architecture

The situation may be further complicated in the case of an organization that is a federation of divisions or other organizations, each of which has its own directory structure.

Figure 8 shows the proposed replication architecture for NATO, which is a prime example of this. The NATO DSA is proposed to be a mandatory shadowing partner for all the nations. It will receive and hold a copy of the information in all national Border DSAs and be configured to shadow it all out again to any nation. There may be bilateral shadowing agreements between nations as well.

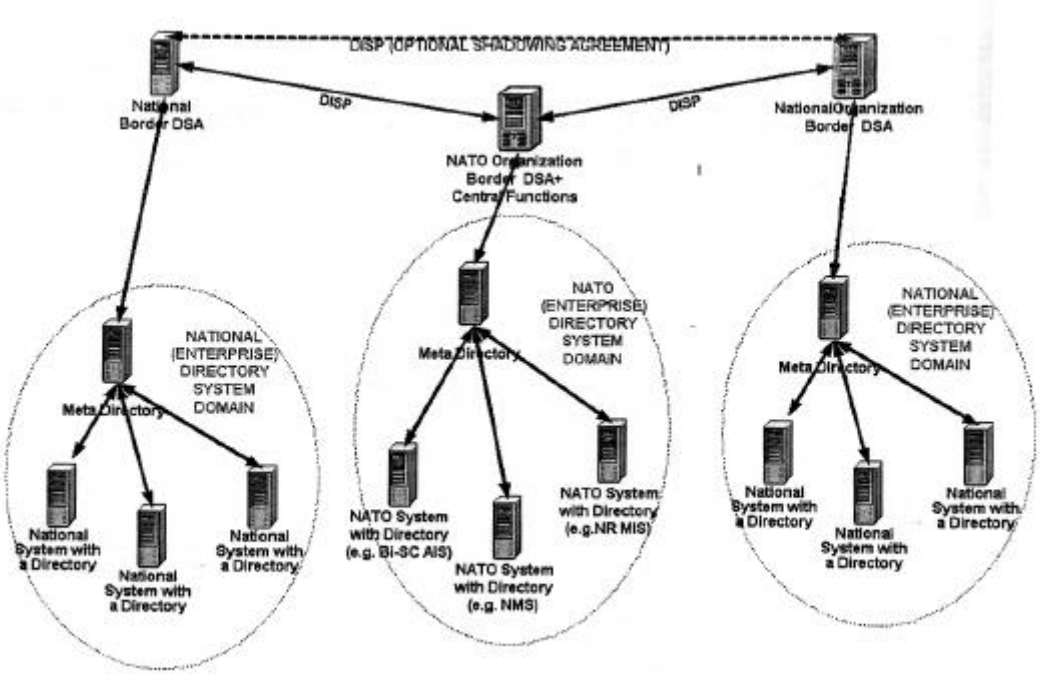

Figure 8 – NATO Replication Architecture

What information is stored where depends to some extent on the user organization. For example, in some cases certificates will be held in the user organizations' directories, in others they will be held in the CAs' directories. In the DoD, access to the certificate of someone local will be via a local directory but access to other certificates will be via a global directory if not replicated locally.

The right to access information may depend not only on the identity of the person requesting access but also on where they are located. For example, someone coming in via the Internet to an unclassified NSA website would see certain information through a firewall; if they can present credentials they will see more information; but to reach all authorized unclassified information, they must pass through additional layers of boundary protection.

Directory Information

The information stored in the directory and retrieved via LDAP includes Address Book and Revocation List information as shown in Figure 6 .

A more complete list is given under **Computer Actors and Roles**. The CA repository may include:

- Key history
- Certificates (CA and user)
- CRLs and ARLs
- Other user attributes, not necessarily for public view, e.g. qualifying information (which must be kept for audit) and addresses etc.

For security reasons, certain of this information (key history and signing keys, for example) may not be kept in the directory itself, although it may still be accessible through the directory, possibly using a metadirectory. Figure 9 shows the architecture for the DoD's PKI Global Directory Service (GDS) Directory, which illustrates this concept.

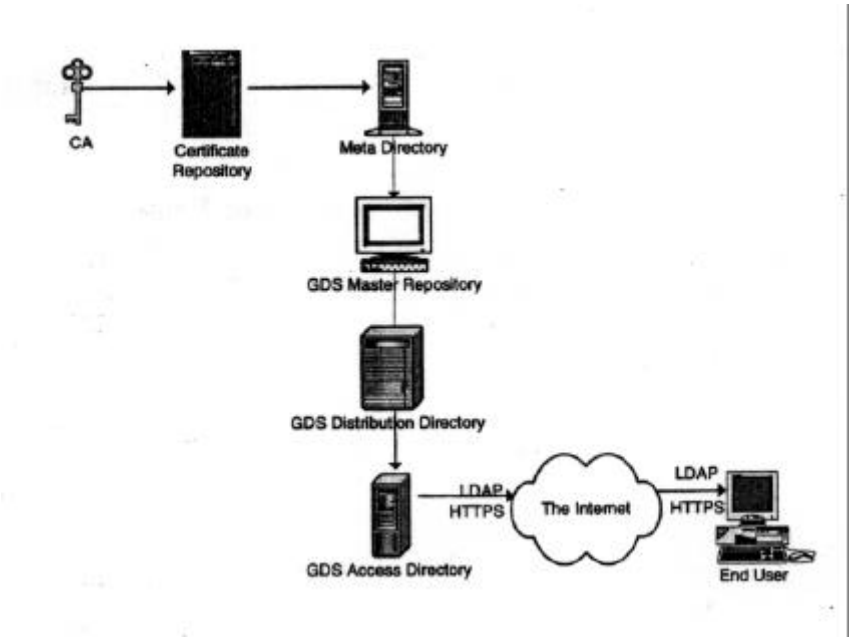

Figure 9 –Architecture for the PKI/GDS Directory

The basic directory model is defined in ITU-T Recommendation X.501. The public key and attribute certificate frameworks are defined in ITU-T recommendation X.509. These ITU-T recommendations form the foundation for the directory and PKI work of the IETF, as well as of the ITU-T.

More detailed elements of the directory information model are defined in ITU-T recommendations X.520 and X.521. They cover standard attribute types and object classes for use in directory schema. They too are referenced by further IETF work, as well as being standard definitions for X.500 directories. IETF RFC 2256 summarizes their application in the context of LDAP v3.

These standards do not completely define the way in which information is stored in directories. For example, people can be identified by the "Common Name" attribute, but this does not distinguish them uniquely. There may be more than one "John Doe" in the US Army. To obtain a unique identity, the DoD appends a random number to the

Common Name. Other organizations follow different approaches. For example, ViaCode allocate an anonymous name to each of their customers.

Telephone numbers provide another example of lack of format definition. There is a standard "Telephone Number" attribute, but no agreement on its form. (Does it, for example, include the Country Code? See RFC 2806 for the complexities involved, also the work in progress in the ITU-T on recommendation E.123 - Notation for National and International Telephone Numbers, E-mail Addresses and Web Addresses.)

The pkix working group of the IETF (see http://www.ietf.org/html.charters/pkix-charter.html) has produced some schema definitions for PKI-related data in RFC 2587. However, they are based on version 2, not version 3, of LDAP. Also, they do not define precisely how information should be stored; for example, they do not address the problem of how multiple certificates for the same person should be distinguished so that the correct one for any particular use can be found easily.

The pkix working group has also produced a certificate and CRL profile definition in RFC 2459. This RFC is however not uniformly implemented (this may just be a matter of the time needed for implementers to catch up with the standard).

The definition of the information retrieved via LDAP is thus an area where the standards are partly missing and where some of the standards that are there are currently not effective.

Because the way information is stored in directories is not fully and effectively standardized, it can be hard to write common applications that look up information in the directories.

However, the nature of much of this information, and the way it is inter-related, depends on the user organization. For example, a person could have one certificate or more than one certificate - the DoD has decided on separate signature and key management certificates because of key escrow issues. This implies that complete and universal standardization of this area may be neither achievable nor even desirable.

Key Management

The standards requirements for Key Management interfaces other than those involving Directory are not explored in this scenario.

The IETF pkix working group has produced some such standards, for example RFC 2510 which defines a set of certificate management protocols. There are also other formats in common use, such as the Public Key Cryptography Standards (PKCS) produced by RSA Laboratories (see http://www.rsasecurity.com/rsalabs/pkcs/) which are in part referred to by RFC 2510.

Exploration of the non-directory requirements for Key Management interfaces through business scenarios could be valuable, but such exploration should be undertaken by a group with wider scope than the Directory Interoperability Forum.

Requirements for Standardization

Transactionality

There should be some kind of notification through the protocol when a master-slave update succeeds, or in general in any instance of when one database provides information to another. The whole transaction should be successful, or else there should be rollback. Otherwise, there must be human-intensive checking processes, for example to verify that a CRL update was done successfully.

This requirement is somewhat at odds with the traditional distributed directory model of “loose consistency”, and there is therefore little attention to it in the directory standards community.

Updatability

In the context of a distributed, possibly multi-vendor, system, updating is not a simple matter. Updates must be able to be done reliably and in a timely manner. For example, the DoD has a policy that a change must be replicated throughout the system in 6 hours.

There are existing X.500 standards for replication of information in distributed directories. There is work in the IETF ldup working group (see http://www.ietf.org/html.charters/ldup-charter.html) on replication of information between LDAP servers. There is some commonality between these two activities, but their approaches are fundamentally quite different.

Schema Extensibility

It should be possible to extend schema to enable a variety of information to be imported into and stored in the directory. For example, an organization might have X.400 distribution lists and want to use them for key distribution. X.400 is much richer in attributes than SMTP. It should be possible, and easy, to extend an SMTP-based schema to allow input of the X.400 information.

Schema extensibility may be an implicit assumption of the LDAP community, but it is not formally stated as a requirement in the LDAP standards.

Signed Directory Contents

People will want to sign objects in the directory, to ensure that the data has not been interfered with or changed. This would apply to reasonably static, but important, data. For example, an automated key distribution facility might store keys signed by another key. The signature would be applied in the client and then stored with the data. It would also be useful to store some attributes in encrypted form.

The third edition 1997 of the X.500 standards covers signed attributes and attribute values and also encrypted attribute values. But the LDAP RFCs do not cover this, although some relevant work has been done by the IETF ldapext working group.

Directory Access by Applications

Applications today generally do not perform certificate validation properly. They should validate certificate paths of any length, checking for revocation.

That they do not is partly because there is not a commonly understood and agreed way of finding certificates and CRLs in the directory, or in some cases of finding the entry for the subject of a certificate. For example, if the certificate certifies an e-mail address then it is not clear how the application can find the matching directory entry.

These problems could be resolved by standards for how information should be stored in the directory and how applications should access it. But the definition of such standards is not an easy matter, for reasons discussed in this scenario. In particular:

- What information is stored and exactly how it is stored varies greatly from one user organization to another.
- A single, global, trust structure has not emerged. Instead, disjoint PKI communities are developing, and there is trust within them but not between them.

There are already standards in this area but they are not fully effective. What is needed at this point, rather than the definition of new standards, is development of guidelines for directory implementers and users on the use of existing ones.

The following requirements for guidelines were identified by the workshop.

Information Publication Guide

It is necessary to publish some information about people in the directory for certificate validation, and perhaps desirable to publish more information to enable e-business. But there are drawbacks to publishing too much information. For example it may lead to junk e-mail ("spam") or to headhunting by competitors.

Directory Policy Guide

A directory policy is in many ways like a certificate policy. There are many things to consider, including the form of Distinguished Names, and the use of dc-based or country-based (traditional X.500 style) naming.

(Tools to standardize mapping between dc-based and country-based naming schemes would be desirable, but it is not at this point clear how far a standardized mapping is possible. Development of directory policy guidelines could shed some light on this.)

Shell Information Services are developing a policy document: their "General Operating & Procedural Framework for the General Group Directory (GGD)". This is being written to document principles of and general procedures around the General Group Directory (GGD) in order to get a common understanding about:

- Roles, responsibilities and obligations with respect to the GGD and its content.
- How the scope of the GGD is controlled over time.
- The purpose of the GGD.
- The allowed use of the GGD and its content.

This document is an example of what the guidelines might be used to produce, and is also a good starting point for development of the guidelines themselves.

One possible approach to producing the guidelines would be to position (i) Directory Services and (ii) PKI services and (iii) KMI services against threat scenarios, showing how each area identifies security obligations within its functional scope that counters applicable threats. A Security Policy view of the system, including technical, social and business architectural policy areas could then be developed and the security policy obligations specific to directory services would emerge.

Referenced Documents

1. The Open Group Document W902: White Paper - Assuring Interoperability for the Directory-Enabled Enterprise.

   See especially Part 1: Business Scenario for the Directory-Enabled Enterprise. ISO / IEC 9594-8 / ITU-T Recommendation X.509: Information Technology - Open Systems Interconnection: The Directory, Authentication Framework

2. The Open Group Document X99DI: Product Standard – Directory – LDAP 2000

3. IETF RFC 2251: Lightweight Directory Access Protocol (v3)

4. IETF RFC 2256: A Summary of the X.500 (96) User Schema for use with LDAPv3

5. IETF RFC 2459: Internet X.509 Public Key Infrastructure, Certificate and CRL Profile.

6. IETF RFC 2510: Internet X.509 Public Key Infrastructure Certificate Management Protocols

7. IETF RFC 2527: Internet X.509 Public Key Infrastructure Certificate Policy and Certification Practices Framework

8. IETF RFC 2587: Internet X.509 Public Key Infrastructure LDAPv2 Schema

9. IETF RFC 2806: URLs for Telephone Calls

10. IETF RFC 2891: LDAP Control Extension for Server Side Sorting of Search Results

11. ISO / IEC 11770-1: 1996 Information technology - Security techniques - Key management - Part 1: Framework

12. ISO / IEC 11770-2: 1996 Information technology - Security techniques - Key management - Part 2: Mechanisms using symmetric techniques

13. ISO / IEC 11770-3: 1999 Information technology - Security techniques - Key management - Part 3: Mechanisms using asymmetric techniques

14. ITU-T Recommendation X.501 (2000) | ISO / IEC 9594-2:2001 Information Technology - Open Systems Interconnection - The Directory: Models

15. ITU-T Recommendation X.501 (2000) | ISO / IEC 9594-8:2001 Information Technology - Open Systems Interconnection - The Directory: Public-key and attribute certificate Framework

16. ITU-T Recommendation X.509 (2000) | ISO/IEC 9594-6:2001 Information Technology - Open Systems Interconnection - The Directory: Selected Attribute types

17. ITU-T Recommendation X.501 (2000) | ISO/IEC 9594-7:2001 Information Technology - Open Systems Interconnection - The Directory: Selected Object Classes

The Open Group publications can be obtained on-line from The Open Group Publications web pages at http://www.opengroup.org/publications/.

IETF RFCs can be obtained on-line from the IETF RFC Repository at http://www.ietf.org/rfc.html.

ISO Publications can be purchased on-line from the International Organization for Standardization via their on-line catalog at http://www.iso.ch/infoe/catinfo.html.

ITU-T Recommendations can be purchased on-line from the International Telecommunications Union via their on-line bookstore at http://www.itu.int/publications/bookstore.html.