A Tool for Support of Key Distribution and Validity Certificate Check in Global Directory Service

Borka Jerman-Blažič,* Denis Trček, Tomaž Klobučar, Franc Bračun

Laboratory for Open Systems and Networks, Jožef Stefan Institute,
Jamova 39, Ljubljana, Slovenia

phone: 386 61 125 -91-99
fax: 386 61 219-385
e.mail:jerman-blazic@e5.ijs.si

Abstract: The problem of key exchange and strong authentication in the Directory services according to the X.500 Recommendation is addressed with certificates which are issued by well-known, trusted authorities known as certification authorities. Finding a path of certification assignment for certification validity checking and verification of the communication party public key is a rather complex task in the global Directory. This paper describes the development of a tool which addresses this problem and provides support for the end user and the certification authority in finding a path of certification assignment.

Keywords: security, open communications, key management, Directory services, certificate, certification authority

1. Introduction

Security services based on a cryptographic mechanism (e.g. encipherment) assume cryptographic keys to be distributed to the communication parties prior to secure communications taking place. This causes one of the most subtle problems when integrating cryptographic functions into networks, since key management has to guarantee that the right keys are available at the right time in the right place. In the Open Systems Interconnection Reference Model, Security Architecture, Part2 (1) key management is defined as "generation, storage, distribution, deletion, archiving and application of keys in accordance with security policy". The purpose of key management, therefore is to provide procedures for handling cryptographic keying material to be used in cryptographic mechanisms (symmetric or asymmetric).

Key management schemes generally depend on the type of keys to be distributed, on the given facilities (e.g. the properties of the specific environment) and on specific application. The most important considerations are the type of threats to be protected against, and the physical and architectural structure of the system. The most fundamental problem in key management is key distribution i.e. the problem of establishing and exchanging keying material whose origin, integrity, and, in the case of secret keys, confidentiality can be

guaranteed. The other important consideration is to provide an easy use of the security services for the average end user.

A key may be distributed either manually or automatically. Manually distributed keys are exchanged between parties through the use of a courier or by some other physical means. In a global network with open architecture, manual exchange of keys can not be considered a serious means of key distribution and exchange. Automatic distribution of keys typically involves the transmission of different data elements. A transaction is usually initiated by requesting a key from central facility (e.g. Key Distribution Centre). Cryptographic information is then exchanged between communicating parties for the transmission of either keying material or for authentication purposes.

A large variety of key distribution protocols can be found in the literature (2,3,4). Key management has also been addressed by different standardisation bodies (ISO, IETF) which has led to several national and international standards. Standards for other application areas, as well as base standards dealing with generic issues of key management, are in preparation (2,5,6). This paper deals with the key distribution mechanism used as defined in X.500 or ISO 9594 where the problem of key exchange and strong authentication is addressed with certificates. Certificates (7) are issued by a well-known, trusted authority known as a certification authority - CA. In a global open network one can expect a number of CAs making up a complex infrastructure. In such a system there is an obvious need for a tool which will help the end user find and check the validity of a partner's certificate. In this paper we describe the design and the implementation of a tool capable of path finding certificate assignments in a globally organised Directory system. The development of the tool was based on the infrastructure provided by the Directory services based on X.500 Recommendation. The tool design is general enough to be implemented in other emerging general directory services such as WHOIS ++ (8) where similar problems may appear.

2. The global infrastructure

Directory Standards (6,7) define a generic network service represented as a globally-distributed database called the Directory. The initial motivation of the Directory design was to provide OSI applications with a standard facility for performing "name-to-address" mappings such as obtaining an electronic mail address given a person's name or obtaining a presentation layer address given an application layer entity title. However, since the Directory can support a wide variety of data types and has powerful search capabilities it is suitable for a much broader set of applications. One of the many valuable application foreseen in the near future is the provision of security services such as the authentication of users and the exchange of certificates.

Information in the Directory is stored in entries. The set of entries in the Directory is known as the Directory Information Base (DIB). Each entry holds information about a single object, e.g. a person, an organisation or a service. Entries are organised hierarchically within the Directory to form a tree, in much the same way as files are

organised in a hierarchical file system. This tree is known as a Directory Information Tree (DIT). The content of each entry is defined by the attribute types which indicate how associated attribute values are to be interpreted. Attribute types are: commonName, surname and e-mail address. Borka Jerman-Blazic is an attribute value for commonName and jerman-blazic@ijs.si is an e-mail address. A subset of each entry's attribute values are distinguished to form a relative distinguished name (RDN). RDNs are used to distinguish entries with a common parent, much as file names are used to distinguish files within a given directory. Each entry in the Directory is uniquely named by the sequence of RDNs labelling the entries on the path from the root to the entry itself. This sequence of names is called the distinguished name (DN) of the entry and corresponds to an absolute path name in a hierarchical file system. The root entry of the DIT has no attributes. The entries immediately beneath a given entry are referred to as subordinates of that entry. The naming authority for an entry decides how its subordinates are named.

Many different organisations are expected to provide entries in the Directory. To facilitate the distribution of the Directory service, the Directory is described as a set of processes called Directory System Agents (DSA) which communicate with each other using the Directory System Protocol (DSP). A Directory user is a person or application process that accesses the Directory. Each Directory user has an associated Directory entry. The DN of this Directory entry is the user's "userid" when accessing the Directory. Users must access the Directory through a process called a Directory User Agent (DUA). The DUA communicates with the Directory using the Directory Access Protocol (DAP). Communication between the DUA and the Directory is only enabled if the DUA can provide the user's Distinguished Name. After the validation of the user identity by the Directory, the DUA is allowed to issue a request to the Directory service. The identity of the user can be verified by either simple or strong authentication

In simple authentication a user provides a password and the DN of his Directory Entry. The password is compared with one stored in the named Directory Entry and if they match the user is allowed to issue requests. If the DSA being accessed in the simple authentication process does not contain the user's Directory entry, then that DSA must issue a request to the appropriate DSA asking if the provided password matches the one in the user's Directory entry. Simple authentication is based on the assumption that DSAs are trustworthy. With different organisations administering different DSAs, this is an unacceptable assumption.
The other approach, i.e. strong authentication, is briefly described in the text that follows.

3. Public key cryptography and strong authentication

Strong authentication described in X.509 is based on public key cryptography (6). In a public key cryptosystem each user has a public and a private or secret key. These keys have the following properties:

- It is computationally infeasable to derive one key from the other without additional (secret) information;
- Data encrypted with one of the keys can only be decrypted using the corresponding key.

It is assumed that the users of the Directory keep their private keys confidential. Given this assumption, if data can be decrypted by a particular user public key, then that user must have encrypted it. The users convince a DSA that they know their private keys through submission of two copies of their DN: one encrypted with their private key and the other unencrypted. The DSA obtains the user's public key from the Directory, uses it to decrypt the encrypted name and then compares it with the unencrypted name. If they match, the DSA assumes that the user knows the private key and hence is who he claims to be.

The scheme seems very simple and sufficiently secure, but in fact it is not; firstly, the DSA has to trust that the public key obtained from the Directory is actually the user's public key and secondly there is no protection against reply. A malicious user can make a copy of the information sent and use it later to masquerade as the user. The first problem is addressed with certificates. A certificate consists of a collection of information, including the user's distinguished name and public key, and a digital signature of this information produced with the private key of some well known trusted certification authority (CA). Digital signatures are obtained in a two step process. Firstly, the data is compressed using a one-way hash function which transforms the information into a string of fixed length, then this so-called message digest is encrypted with the secret key. Because the digital signatures are unforgeable, certificates can be stored in the Directory. Given a user 's name, a DSA can securely obtain the user's public key by fetching a user certificate from the Directory and using the CA's public key to verify the certificate, i.e. verify the digital signature.

The reply problem is addressed with a "serial number". In the on-going communication with encrypted data a serial number is associated along with the DSA's name. Each time a user connects to a particular DSA a different serial number must be supplied. The serial numbers are not issued in any particular order. The proliferation of serial numbers is limited by the time validity of the certificate and thus the user’s public key. Users need only to be sure that in the given period of time there does not exist more than one unexpired certificate in a communication token with the same serial number.

The development of the Directory as a global service proceeds along two fronts: technical and administrative. On the technical side, a number of DSA and DUA implementations have been developed and they interoperate to different degrees. Most of the DSA implementations provide the majority of the specified features from the standard documents from 1993 with the exception of strong authentication. A wide variety of DUA implementations and interfaces are available for different operating environments and the same is true for DSAs. Probably the most widely used Directory implementation is QUIPU, developed as a part of ISODE (9). The only well known project implementing the security services (both in DUA and DSA) based on QUIPU software was PASSWORD (15). The pilot sites in PASSWORD (one of which was the Laboratory for Open Systems

and Networks at the Jozef Stefan Institute) have provided the facilities needed for the operation of CAs and for key management. A PASSWORD pilot site is capable of providing trusted key distributions for different application processes such as X.400, X.500 or PEM (Private Enhanced Mail) (13). Every pilot site in the project operates one or more CAs, creates and distribute keys and certificates, and cross-certifies other CAs, enabling users of different security domains to inter work, manage a certificate revocation list and monitor and audit the use of the security services. The certificates produced by the CAs are distributed in the Directory services. This infrastructure also enables distributed authentication service for several applications.

4. Discovering certification path in the global infrastructure

The global infrastructure of the Directory is expected to consist of many DSAs with numerous entries belonging to different CAs. All these CAs have their own certificates with “issuer” and “subject” fields which can be used as pointers to other CAs. Taking this into account, and by extracting the pointers and the CAs from the Directory Information Tree (DIT) we can build up an imaginary structure which we will use for a better understanding of the process of certificate validation and checking. This structure is represented in FIG.1

A typical problem for an end user wishing to send an X.400 or PEM encrypted message or to verify the digital signature of received mail will be to find out the other user’s certificate and its public key. In some cases the certificate found in the DIT together with the DN of the user will provide satisfactorily the requested user identification and authentication and thus the origin of data. However, verification of the complete path of certificate assignments to particular communication party will always be required, especially in applications such as EDI (3).

The hierarchical tree of CAs is not necessarily simply mapped to the tree or trees built up from the DIB. This is especially true if we consider cross-certificates between CAs. For this reason, the problem of certificate verification and validity checking in the Directory becomes a problem of path finding a particular certificate in an unlimited mesh of CAs. The structure represented in FIG.1 is intended as a simplified model for the explanation of the tool and the procedures applied.

Let us assume that the user A in the CA structure presented in FIG.1 needs to find out a path of certificate assignments to user E. The user A has to go through the following steps:

1. In the first step of the search, user A will find out that the immediate superior certificate assignment to user E was issued by CA named C6;
2. In the second step, user A will find in the entries for C4 and C6 in the "issuer-subject" fields that the next certification assignment was done by C2 and C3 respectively and in
3. the next step which can be (recursively) repeated the user will come to the final point of common trust. This will enable the full validity check of the public key of user E.

At first sight it might be concluded that neither the problem itself nor the procedure applied is particularly complex or tedious. However, if we consider the global infrastructure and the numerous CAs appearing in the Directory, together with the lack of automatic mapping or extracting of the CAs structures then the development of a tool to help the average user is certainly needed. This is also justified by the restrictions or administration procedures applied to an average user such as:

- administrative limitations within the Directory (time of search, amount of returned data),
- network failures,
- possible problems with different Directory Service Agents (DSAs with no knowledge of the CA structure).

Taking into account the above, a user tool has been developed. The tool is intended as a help to end users and to the application processes in the automatic search of paths of certificate assignments. The tool has been designated CA-Browsing System. The CA-Browser performs on-line search and finds certificate assignments as a path of certificates ordered according to the CA structure in the DIT.

5. The CA-Browsing System

The CA-Browsing System consists of the CA-Browsing Server and the CA-Browsing Client. It is open in terms of communication and internal architecture. It does not impose any specific security administrative solutions and is developed to allow the separation of technical from administrative matters. It can be used for support of end users, applications and certification authorities. The CA-Browsing System works in various environment, e.g. TCP/IP or OSI. It can be also applied in other security protocols dealing with key exchange such as NLSP (16).
The basic communication concept of the CA-Browsing System is presented in FIG.2.

5.1. The CA-Browsing Server

The internal architecture of the CA-Browsing Server is depicted in FIG.3. The CA-Browsing Server is made up of five modules:

- The first module binds to the system of Directory Service Agents (DSAs) and searches for CA entries in the Directory Information Base. This task is performed when the workload of the network is low (11) usually in an agreed time period (once a week for example).
- The second module manages the local database of the CAs and their relationship to each other. The local database is built up from information (CAs, certificates and revocation lists) provided by the DSA search module.
- The third module constructs the graph. The information about the CAs' relationships, certificates and revocation lists is stored in the graph nodes.

- The fourth module searches several optimal paths through the use of the Yen-Lawler algorithm (optimal in terms of the shortest sequence of certificates that are required for verification of certification assignment). This module also validates the certification paths that are found and checks them against the revocation lists of every node.
- The fifth module in the current pilot implementation is simply a standardised communication module, i.e. a UNIX socket. This module interacts with the CA-Browsing Clients which, on the other side, communicate with the users. The current implementation supports only 7-bit ASCII presentation.

The DSA access and search module binds the CA-Browsing system to the DSA. After successful binding it starts the search for CA objects. The easiest way to search the DIT for objects where the attribute type ObjectClass has a value CertificationAuthority is by sending a single search request that is applied to the base object, i.e. an object that represents country or locality, and all of its subordinates. Although the Directory has powerful search capabilities, too general search request to a Directory subtree with many entries is very time-consuming. Compared to other directory services (e.g. WHOIS++) the search in the DIT for attributes which are not part of DN requires a search of every corner of the DIT. In the course of the CA-Browser development we realised that, owing to these many limitations, such queries almost never get complete answers. For that reason, the search is performed in two steps.

In the first step, the search module looks for organisations in a particular country. This is done under the assumption that every CA in the DIT is located within an organisation. Sometimes it is necessary to search organisations according to the lexical order of their names because of the previously-mentioned limitations of DSAs (limited amount of returned data). Currently this approach is successful in that the search is relatively fast. In the second step the module looks for CAs within organisations.

In the first attempt at system design, a synchronous search was applied, i.e. no new request was issued to the Directory Service until a reply from the previous request had been received. The testing exercises showed that due to uncertain reachability of some sites in the network and some administrative limitations, the tool as a whole was too slow and almost unusable. In the second attempt, the search method was changed and the search was performed in an asynchronous way meaning that search requests, e.g. search requests for CAs in different organisations, were sent simultaneously without waiting for the reply to the previously-issued requests to appear. The different reachability of DSAs causes the sequence of the received replies to be different from the order in which the requests were issued. This approach makes the search more natural and logical. In cases where no data was obtained for a particular CA, then this CA was not added to the graph and classified as not accessible.

The search results, i.e. CAs, corresponding certificates and revocation lists, are stored in the local database. This data is later used for generation of a directed certification graph that is defined by two sets: the set of vertices and the set of directed lines, called arcs. The vertices of the graph are CAs, represented by their distinguished names extracted from the

certificates, more precisely from the issuer and the subject fields of the certificates. Two vertices, denoted by CA1 and CA2, are connected with an arc that goes from CA1 to CA2 if, and only if, there exists a certificate owned by CA2 and issued by CA1.

The certification graph is represented with the list of neighbours which is used in the Yen-Lawler algorithm for finding “k” shortest paths, where the number “k” is supplied by the user. The length of the path was chosen as a criterion for selecting a subset from all existing paths between two certification authorities. One may perhaps prefer other solutions, e.g. paths that contain particular CAs. This facility can be provided with the tool.

The current implementation of the tool constructs only one list of neighbours due to the low number of available CAs in the Directory service. With the increase of CAs in the Directory, a division of the directed graph will be needed. One natural division could be by country, by region or by certification policy. This will depend on the complexity of the Directory service in the future. If the Directory service grows in complexity and in the number of entries then the problem of many cross-certifications between organisations in different countries will need another approach. In that case a solution similar to a vector-distance routing algorithm could be applied. Servers in each country will be expected to build up a vector-distance table of the shortest paths from their "border" CAs to the "border" CAs in other countries. Such tables will be interchanged and updated at agreed time intervals.
Finally, the certification paths that have been found by the CA-Browser are validated and checked against the revocation lists.

5.2. The CA-Browsing Client

The CA-Browsing Client communicates with the server and enables user friendly interaction. Selected CAs can be entered in two ways:
1.By entering the distinguished name of a CA, e.g. c=SI@o=IJS@ou=E5@ou=IJS-CA.
2.By asking the system to list all CAs that have been found in a certain portion of the DIT, e.g. under some object with a distinguished name such as c=SI and then selecting the chosen CA.

When the system finds valid shortest paths from one CA to the other, it prints the data as DN sequences of corresponding nodes. The user, who also has the option of seeing the certificates that form the paths, can then choose the path that he trusts the most.

Example:

The user first enters the DN of the CA which has signed his certificate and then the DN in the issuer field of the certificate he is trying to verify, e.g.

1.CA: c=DE@o=GMD@ou=PASSWORD@ou=DE-CA
2.CA: c=SI@o=IJS@ou=E5@ou=IJS-CA

Number of paths: 1

The system displays the following shortest path:
1: c=DE@o=GMD@ou=PASSWORD@ou=DE-CA
2: c=SI@o=IJS@ou=E5@ou=SI-CA
3: c=SI@o=IJS@ou=E5@ou=IJS-CA

6. Future development

Further development of the tool and its use will depend on the future structure of CAs in the global network, types of available distributed Directory services and the intensity of the use of security services. Our current experiences have shown that :

- when the tool is applied in a general scheme of CA structure developed in line with X.500 Recommendation (12), the use of the tool is justified and very valuable;
- when the tool is used for PEM certification assignment if they are a subset of X.500 ("enriched" with a lot of cross-certificates), the system proves to be very useful too;
- when an "elementary" PEM-CA (13,14) structure is used with only forward certificates, such a system is not needed because one can deduce the only existing certification path from the distinguished name of the certification authority.

Currently only one server is running (at the Laboratory of Open Systems and Networks at the Jozef Stefan Institute). Considering the low number of CAs in the world (at the time of writing, there are approximately 84 CA entries in the pilot Directory services in 7 countries: GB, Germany, Ireland, France, Spain, Portugal, Slovenia, and under locality of Europe) this should not be a problem. Later, we expect CA-Browsing Servers to be installed in other countries, where the task will be carried on by the responsible organisation in the country concerned.

In the future we plan to upgrade the system in the following ways:
- by redesigning the system to run as a distributed application; such deployment will prevent the servers from overloading the X.500 Directory service (unwrapping a key certification path is an intensive operation and puts quite a heavy load on the system);
- by making it suitable for use in a graphical environment (Windows) e.g. development of a GUI.

7. Concluding remarks

Certification path finding in the Directory for verifying a user's public key is not a trivial task. A user oriented tool, called a CA-Browsing System has been developed to support different types of users (end users, certification authorities). The tool can promote the use of security services in a global open network. The world-wide Directory infrastructure according to X.500 is still in a testing phase. The same applies to the security services

offered over the global network. We expect the deployment of the system to continue along with the further wide-spreading of the Directory and security services offered to every user of the global network.

References

1.International Standard Organisation, Information Processing Systems, Open Systems Interconnection Reference Model - Security Architecture, ISO 7498-2, July 1988.
2.American National Standards Institute, Data Encryption Algorithm, ANSI X3.92, New York, 1981.
3.ANSI X12.42-198, EDI Security Structures and Cryptographic Service Message Transaction Set, 1989.
4.Rivest R.L., Shamir A., Adleman L., A Method for Obtaining Digital Signatures and Public Key Cryptosystems, Comm. ACM, Vol. 21, No. 2, February 1978, pp. 120-126.
5.W.Diffie, M.E.Hellman, New directions in Cryptography, IEE Trans. Inform. Techn., 22 (1976), 644-654.
6.CCITT Recommendation X.509, The Directory - Authentication Framework, 1989.
7.International Standard Organisation, Information Technology - OSI-The Directory, Authentication framework, ISO/IEC 9594-8, December 1988.
8. Deutsch, Peter, et. al., "Architecture of the WHOIS++ service", Internet Draft, July 1994.
9.S.E.Hardcastle Kille and P.Barker, The COSINE and Internet X.500 schema, RFC 1274, University College London, November 1991.
10.Rosenbaum, Using the Domain Name System To Store Arbitrary String Attributes, RFC 1464, May 1993.
11.CCITT, The Directory - Access Control, X.581, Recommendations of the X.500 Series, Helsinki, Sept. 1992 (unofficial draft version).
12.International Standard Organisation, The Directory - Overview of Concepts, Models and Services, ISO/IEC 9594-1, December 1988.
13.Internet Engineering Task Force, Privacy Enhancement for Internet Electronic Mail: Certificate Based Key Management, RFC 1422, February 1993.
14.Internet Engineering Task Force, Privacy Enhancement for Internet Electronic Mail, Key Certification and Related Services, RFC 1424, February 1993.
15.Luehe J., R2.6: User Requirements, Ver. 1.0, PASSWORD Project, November 1992.
16.ISO/IEC JTC/SC6, Telecommunications and Information Exchange Between Systems, Draft Network Layer Security Protocol, July 15, 1991.