Associating metrics to Certification Paths

Anas TARAH[1] and Christian HUITEMA[2]

[1] CERICS, rue Albert Einstein Sophia-Antipolis B.P. 148
06561 Valbonne, Cedex-France.
email: tarah@cerics.cerics.fr
[2] INRIA, Projet Rodeo Sophia-Antipolis B.P. 109
06561 Valbonne, Cedex-France
email: huitema@sophia.inria.fr

Abstract. This paper presents a part of our work on open systems' security in conformance with the *X509* framework. The Chimæra model tries to cover all *X509*'s lacks specially for what concerns *Certification Authorities* **CA**. Although our primary concern was the elaboration of a security scheme, we quickly met the need of a convenient distribution of CAs and the manipulation of both certificates and certification paths.
The main trends of the scheme are: the elaboration of the CA concepts, the elaboration of a communication protocol between these authorities by and the introduction of the evaluation notion of both certificates and *Certification Paths* **CP**. In the first part, A brief introduction to major security trends and mechanisms is given, then some implimentations and standards are cited. At this level, deficiencies of actual models and the need of more convenient scheme are shown. In the next part, main trends of the Chimæra model and its OSI environment are presented. We describe then a protocol for the exchange and evaluation of both *certificates* and CP, *Certification Paths*, hence ensuring a secure propagation of trust and knowledge over the network. Finally, the Added value of the given scheme is discussed in relation to certificate's establishment and revocation.

1 Introduction

Openness and security are inherently contradictory requirements, specially for heterogenous networks with various purposes. Interconnecting these networks in a secure and reliable manner has became a primer concern for both network operators and research centers. Several studies have been undertaken to cope with network's security holes, they led to standards, recommendations and schemes.

The *DoD* [1] published one of the first standards dealing with computer security. In a later step the International Standardization Organization *OSI* and the *CCITT* issued the *X.800* recommendation [10], as a complementary standard to the layered *OSI* model for Open Systems Interconnection [2]. One can also find some other security standards dedicated to specific *OSI* applications like the *X.509* authentication framework in *X.500*, and the security enhancements in *X.400*. All the above mentioned standards expose most of the security re-

Y. Deswarte et al. (eds.), *Computer Security - ESORICS 92*

quirements, services, functions and mechanisms. But they all have shortcomings which can be summarized by:

- all these standards specify, except X509, specify the integration of a security scheme within the concerned application, but they never specify the security scheme itself.
- never defining the accurate location of these services and functions in the system.

On the other hand, several practical systems have been developed to cope with network security. A promising one is *Kerberos* [11], developed within the MIT Athena project. It is essentially based on the trust attributed to a third part, an authentication server, within a given domain. The main drawbacks of this model is the difficulty of dealing with inter-domain authentication within an open system, inherent to symetric key cryptosystems. In the same context, another model is proposed for the security of the Internet based on a hierarchical key distribution between authentication servers. This scheme presents the same shortcomings of the precedent besides accumulating the security of the whole system over the security of the authentication server of the higher level of the hierarchical tree.

Based on the mentioned schemes and standards, the main security requirements are extracted and given in the following with the corresponding mechanism chosen for our security model. The main requirements for a secure open system can be restricted to the following five items:

- access control,
- peer authentication,
- communication integrity,
- non-repudiation, and
- communication privacy;

There are two well known mechanisms to insure access control. The first one consist on attributing name lists of allowed entities to every service. The second gives the allowed entities a privilege (token, ticket, ..) in order to obtain, use, the required service. Detailing these mechanisms is not the scope of this proposed model.

Peer authentication and communication integrity are related items since there is no use for authentication if communication integrity is not insured and vice versa. The digital signature mechanism [13] provides a suitable solution for these two requirements, and keeping a trace of this digital signature may be used as a proof to a judge for non-repudiation.

Encryption is the best way to insure the privacy of a communication through an entrusted network. The common problem with all cryptographic systems resides in key sharing and distribution. This problem can jointly be resolved if an authentication mechanism based on a public-key cryptosystem is used. This can be done, on the sender side, by encrypting the chosen session key with the public-key of the destinator and then with the private key of the sender [4].

Hence, using digital signature coupled with a public-key cryptosystem seems to be the good approach towards secure open system. But, in order to use a public-key cryptosystem one must first adopt an efficient and secure key management and certification mechanism.

A certification mechanism and a key management scheme, serviced for the digital signature, form the basis of the *Chimæra* security model [14], which is presented in the next section.

2 Chimæra

The first idea of this scheme came within the development of *PIZARRO* [8], an *X.500* developed by the Rodeo project at INRIA. The first step was, in conformance with the authentication frame *X.509* [9] to implement a strong authentication procedure between a DUA and a DSA. Primary works were concentrated on local authentication of users. This led us to the development of an experimental cryptographic infrastructure with key management mechanism, and to the elaboration of the *X509* framework in order to bypass some of its defects [12].

At this stage, the idea of a model for open system security came up using an extended version of CAs, Certification Authorities,. This extension mainly concerns the establishment of a certification path, CP, between CAs. The management these certification attributes (certificates, cross-certificates, CP...) needs the use of a database. This database must be at the same time secure and distributed to ensure the openness of the system. All actual models present weakness in managing this distributed data base , if used. *The Directory* seems to present all the characteristics needed for a secure database, the most obvious advantages of its use can be summarized by:

- *X.500* is a distributed database with controlled and authenticated access;
- it is conceived to contain all the network entities (users and applications' servers);
- since it contains all informations about all entities of the system, the CA can use it for name resolution;
- the use of the *Directory* has the important advantage of being within the OSI context.

For these reasons Chimæra use the *Directory* as a database for the management of certificates, cross-certificates and CP. It also uses a local data base, *SMIB*, for the local security policy and for keeping informations about interacting certification authorities, i.e. trust-values.

So, knowing that every entity has an entry in the *Directory* containing its certificate and pointing to its own certification authority, the major attention is given to the problem of key management and cross-certification. In fact, like all public-key based security schemes, the whole work is concentrated on the key management problem. In the Chimæra model we divided this mechanism in two parts. The first treats all problems related to the manipulation of private-keys.

The second concerns the certification of public-keys to the whole system; this led us to elaborate Certification authority and define a new protocol permitting the exchange of trust between CAs. These points are explained in the next paragraphs.

2.1 Secret-Key Management

An important asset of our model is the double protection of secret-keys:

- they are encrypted with their owner password, or PIN[3];
- and stored in a secure place such as a magnetic card or a protected file in the owner homedir.

Hence in order to use this secret key the user must supply his password locally on his working station to decrypt his private key, allowing the encryption of his digital signature. On the other hand, the user will not give his password, PIN, each time he desires to sign a message, and he will neither keep his secret key stored decrypted in his environment or in a file on the workstation: this could compromise the key on which depends highly the security of this model. The Chimæra model delegates the management of the decrypted private-key to a "signing-server", built locally on the user machine, which is responsible of signing its owner's messages with his private-key for a given period of time. Note that the model suppose that workstations insure a sufficient degree of protection to their users. All demands from a user to his signing-server are done locally and are encrypted with a session key in order to, firstly insure privacy and secondly prove user's identity to the signing server, since the session key is only known to the concerned user. When the signing server is firstly called, it asks the users password, then it decrypts the user's private-key and keep it for future demands for messages signature and proof of the pretended identity. The user and his signing-server will be communicating on the same machine, using a session-key known only to the user and to his own 'signing-server'. Thus Chimæra prevents tokens circulation on the network. When a server receives a demand for a signature, it first verifies the originator of the request by decrypting it. If verification succeed the server is sure that the appropriate key has been used which means that the request has been sent by the corresponding owner, then the server encrypts the received MIC[4] with the owner's private key and send it back to the user. Now the user receives back his digital signature performed by his secret-key and can send it with his message to destination. As mentioned before this signing-server has a lifetime, after this lifetime the user must give again his password in order to reactivate the server, new session key will be set by the server and given to the user for future identity proof requests.

[3] Personal Identity Number

[4] Message Integrity Check, which is included in the digital signature

2.2 Public-Key Certification

As already mentioned, the *X.509* standard recommends the use of a CP, allowing to verify the authenticity of the public-keys of a peer CA. It is formed of a sequence of cross-certificates issued by intermediate **CAs**. As given in *X.509*, a CA is responsible for certifying public-keys for all entities of the system (e.g. users, applications, CAs).

The *X.509* model still presents serious lacks in its proposed architecture and protocols. Let's have a glance on the most dangerous ones:

- Every CA on the system has a full ability of certifying any entity over the system, there is no restriction on the trust and the view attributed to a CA. So, any CA can wrongly certify (consciously or not) any entity, or CA, all over the networks.
- The proposed model has a blind trust in every received CP whoever are its co-certifiers. Hence any CP is trusted and used even if one of its co-certifiers is known to be subverted, since no evaluation is done over a received CP;
- The revocation list notion, proposed by *X.509*, applies only to certificates and not to CAs. This will not resolve the problem of information propagation, for instance in the case of a subverted CA.
- This recommendation doesn't treat the problem of CP establishment; this establishment is based on information exchange between CAs. But no protocol has been given for this purpose.

In order to circumvent these problems, several implementors have suggested to link closely a hierarchy of CA with the *X.500* hierarchy of names. This model simplifies the problem of defining the certification domain of a CA to all subnodes in the name hierarchy. But it also has several weak points:

- The security of any sub-tree depends on the security of the superior nodes. Hence any malicious action on a root node can compromise the whole structure.
- The structure supposes a complete and unique hierarchy. This is in contradiction with current practices, when the perception of the hierarchy depends on the type of activity. Moreover, there are many cases where one cannot find a legitimate *"master"* of a naming domain, e.g. of all the names within a country.
- This contradiction is very apparent in large multinational companies, whose networks are spread over several countries and regions.

It is useful to recall the definition and the utility of certification paths.

2.3 Importance of Certification Paths

As already mentioned, Chimæra uses the *Directory* as its distributed database. It is of interest at this stage to moot the importance of a certification path in such an environment.

In fact, the directory information tree, *DIT*, is divided into Directory Information Base, *DIB*, distributed over the whole system. Every DIB of the DIT is linked to a Directory Service Agent, *DSA*, which is responsible of:

- Administrating the local DIB and copies of other DSA's DIB.
- Interacting with other DSAs in order to ensure an ubiquitous directory service.
- Providing a directory interface to system's users.

In order to ensure an ubiquitous directory service and to minimize the interaction between different DSAs. a DSA may keep copies of other DSA's DIBs locally in his database.

The standard does not impose any measures for the security of a DSA and its DIB. So protection measures must be taken locally to ensure the integrity and the safety of the DIB. Meanwhile, the problem of secure exchange of copies between DSAs is still unsolved. This means that secret assets can not be kept in the DIB, which only contains informations of public nature.

For this reasons, a public key can not be obtained directly for its corresponding user's entry in the directory: an attacker could replace the information stored in the DIB by some forgery, hence compromising the system. Certification paths are used to cope with this problem: a CP [9] is formed of a sequence of CertificatePair, *CPR*, which are certificates between intermediate CAs. CPRs and CPs are built by CAs independently of the directory: they are then inserted in their owners' entry in the directory. Since the public-key of a local CA is published locally by administrators for all entities, these entities can verify the authenticity of the assets contained in the directory which are signed by their CA. By checking all elements in the CP, they can verify the key of their peers.

The specifications of these assets (CPs, CCs or Certificates) is given by:

```
Certificate         ::= SIGNED SEQUENCE {
            version                 [0] Version DEFAULT 1988,
            serialnumber            SerialNumber,
            signature               ALgorithmIdentifier,
            issuer                  Name,
            validity                Validity,
            subject                 Name,
            subjectPublicKeyInfo    SubjectPublicKeyInfo}

CertificatePair     ::= SEQUENCE {
            forward[0]              Certificate OPTIONAL'
            reverse[1]              Certificate OPTIONAL}

CertificationPath  ::= SEQUENCE {
            userCertificate         Certificate,
            theCACertificate        SEQUENCE OF CertificatePair OPTIONAL}
```

The CP scheme is nessary to ensure the security of public key systems. But CPs themselves can be forged if an attacker manages to subvert or misuse an intermediate CA. Protection and evaluation of CPs are thus extremely important. After this description on the role of a CP and the necessity of having a framework permitting a secure manipulation of these assets, let us see what Chimæra proposes.

3 Certification within Chimæra

3.1 CP Establishment

Chimæra proposes the basis of a scheme for the establishment of a certification path, and more generally a protocol for the exchange of trust-values in order to evaluate the established CP, see figure 1. In this figure, the CA 'A' needs to establish a CP towards 'B', a remote CA. The problem resides in how to establish this CP without a direct cross-certificate to 'B'. Hence the necessity of communication between intermediate CAs, who can certify 'B', and 'A'.

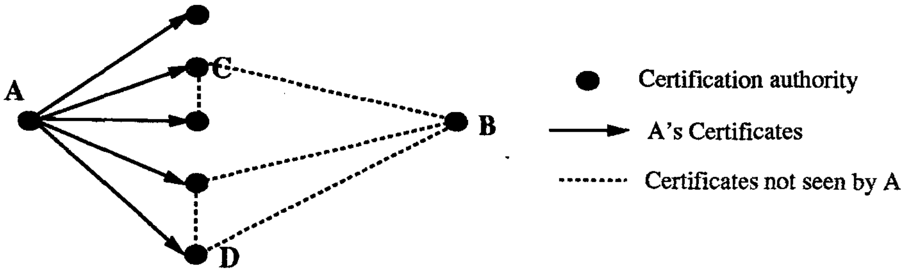

Fig. 1. An example of communication necessity between CAs

Our model considers the CA to be an active entity on the system providing two main services which are:

- Certifying public keys for local entities,
- communicating with other CAs in order to establish new connections (certification paths) to remote CAs.

In order to clearly explain this proposed communication protocol between CAs, we consider the interconnection of several networks as an example of an open system. These elementary networks, or subnetworks, form the granularity of the system. Several subnetworks may be attached to form one *Area* ,or more generally a *Domain*. The proposed model, first supposes that every granular network has its own, single and well-known CA. This CA is distinguished locally by its name. Secondly, one supposes that every CA has certificates for all its local entities and at least one cross-certificate to another CA in the outer world.

The main advantage of such a view is that connections (cross-certificates) need not follow a fixed architecture, they can be bilateral, vertical or hierarchical.

Based on such a view of the system, Chimæra distinguishes three types of certifications:

- Certification within one local network which is the trivial case knowing that there is one CA in the network. This CA has a complete knowledge of all entities connected to the network. Hence certification between local entities can take place easily.
- Certification within an area, a Domain[5], where several CAs, corresponding to the different subnetworks, cooperate and they must exchange their views (certificates).
- Inter-domain certification which treats the certification problem between entities of different domains. This case is separated from the previous one because of the limited number of CAs having out-of-domain communication facilities (i.e certificates for an out-of-domain CA).

Let us first consider the case of certification within a Domain, the example of figure 2. Alice receives a signed message from Bob. She now asks her local CA, CAessi, to certify Bob's public-key. After looking at Bob's entry in the *Directory* and reading his certificate, CAessi find that Bob's certifier is CAinria. But the only CPs that CAessi owns are to CAcerics and CAcma. That is why CAessi diffuses a message to all known CAs asking for a CP towards CAinria. Upon reception of the request of CAessi, CAcerics and CAcma look if they have already obtained a CP to CAinria. If this is the case, they send this CP back to CAessi certified by their secret-keys, and the authentication procedure of Alice can take place.

Otherwise, if CAcerics or CAcma don't have any CP towards CAinria, they just reiterate the request to all their known CAs except the one who sent it. One can project this scheme from another angle, supposing that a CA will propagate its CPs to its known CAs. Thus there is no need for issuing requests, because all CP are propagated to all the CAs in the area as soon as they are constructed.

If the same politic is applied for certification over the whole system, the bandwidth is rapidly exhausted. That is why Chimæra distinguished between intra-domain and inter-domain certification.

For the inter-domain certification and as shown in the figure 3, only a limited number of CAs have an access to the exterior world. Therefore these special **CAs** have a supplementary task of communicating with the outer world. They possess CPs concerning their own domain and others concerning boundering CAs of other domains. Hence as for certification within one domain, a CA has the task of affording, if possible, a certification path of a given target or forwarding the request to other communicating **CAs**. Only these boundering CAs can forward requests to the outer world and more precisely to neighboring CAs. In order to optimize the use of the system Chimæra also supposes the exchange

[5] Note that the notion of domain used in this article may be the same one used within the routing context.

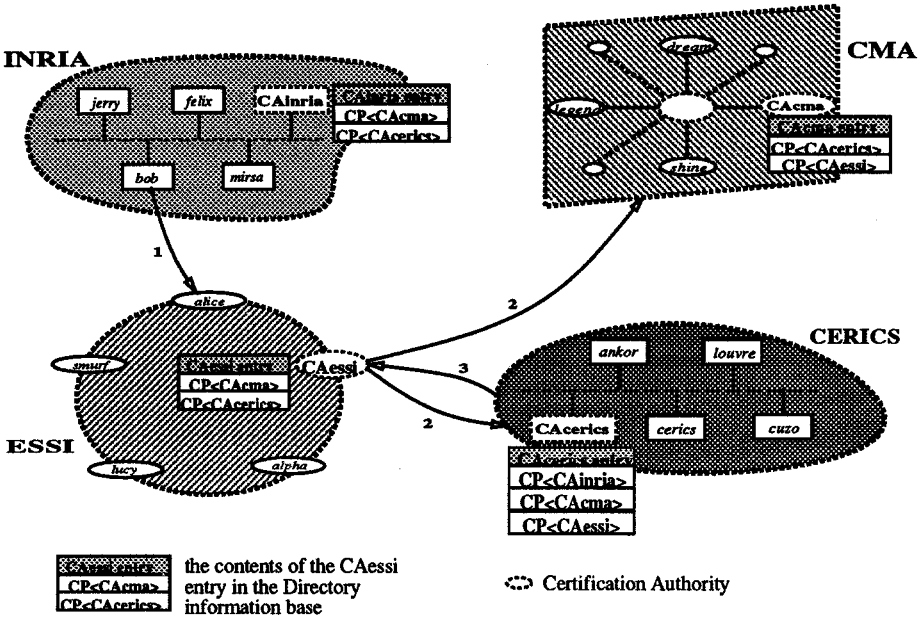

Fig. 2. Certification within a domain

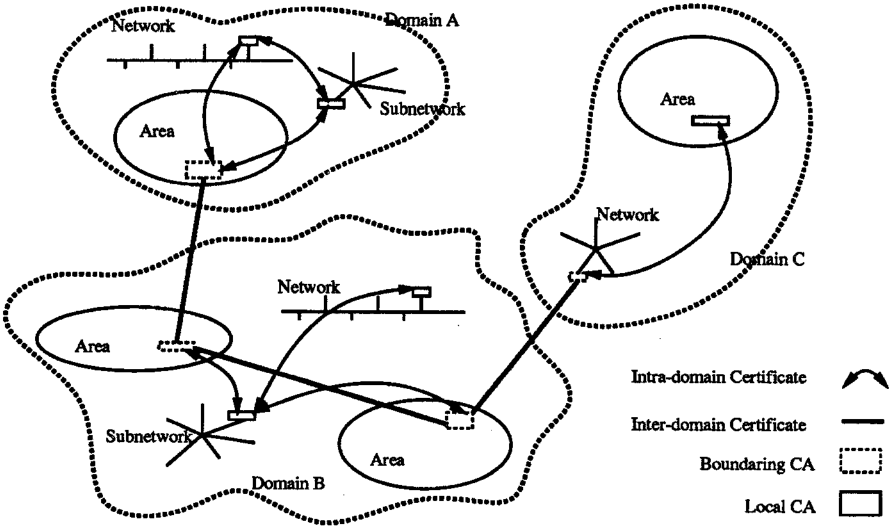

Fig. 3. Example of boundering CAs

of informations about the state of the system and more precisely about certification authorities. If a received information is useful locally, it is saved in the databases, *SMIB* or *the Directory*, else it is forwarded to other boundering CAs. These informations may be of three kinds:

- certification Paths which will be saved in the *Directory*;
- information concerning the definition of domains and their boundering CAs which is saved in the *SMIB*;
- information concerning the state of the system and the trust to be attributed to its different parts, this kind of information is also saved in the *SMIB*.

This model is, hence, based on a two level distribution of CAs, the kind of a CA is given be a new attribute of its entry in the directory. this attribute is called *CAClass*, and defined by:

```
CAClass ::= ENUMERATED { low , high }
```

The specification of the CA Communication Protocol between CAs is given by:

```
CAC-Protocol ::= CHOICE {
         [0] Request,
         -- Diffusing a request coming from low level CAs
         [1] Response,
         -- The result of a CertificationPath establishment
         [2] SEQUENCE OF Information
}
Request ::= SEQUENCE{
          Asset   CHOICE {
                  CP-entity Name,
                  -- A CertificationPath to a given entity is requested
                  CP-CA     Name,
                  -- A CertificationPair to a given CA is requested
                  Requestor Name,
                  -- This may help identifying the domain of the requestor
  }
           Control   ANY OPTIONAL
}
Response ::= SEQUENCE{
          Asset   CHOICE {
                   [0] CertificationPath,
           -- The CertificationPath to the requested entity
           [1] CertificationPair
           -- The CertificationPair to the requested CA
          }
  Control      ANY OPTIONAL
}


Information ::= SEQUENCE{
          Asset CHOICE {
                 [0] Certificate,
             -- A compromised Certificate
         [1] CertificationPair,
             -- A compromised CertificationPair
         [2] Name
             -- A compromised CA
          }
          Control ANY OPTIONAL
}
```

3.2 CP Evaluation

Chimæra does not have a blind confidence on every CA and every received certificates. For this reason, Chimæra proposes the attribution of a trust values to every CA and every asset (CP, CPR or certificate). This is important for two main reasons, which are:

- It constitutes an elegant alternative to the revocation list proposed in the *X.509*. In effect having a general idea about the system and knowing that a co-certifier of a CP has been subverted, is equivalent to the consultation of the revocation list.
- When several CPs are available for the same entity, the evaluation permits a parametrized choice of the CP to adopt.

So, even if all the co-certifiers, intermediate CAs, of a CP are nearly trusted, the requester is free to use this CP, hence considering it trusted, or to refuse it in trend with its global view of the system and the local security policy.

At this stage, and after exposing the necessity and the utility of the CP's evaluation, let us discuss some aspects of this evaluation procedure.

Recall that a CP is formed of a sequence of CAs, co-certifiers, and cross-certificates. A cross-certificate is a mutual certification of two neighboring co-certifiers, as depicted in figure 4.

The remaining problem is in adoption of a metric for the evaluation of a CP. As the evaluation is performed locally, this metric does not need to be universal. Its choice may depend on the security policy adopted in the local domain. Let's see some examples of possible policies:

- The trust degree of a CP depends on the trust in every intermediate CA and is equal to the minimum of trusts. So, any entrusted co-certifier rends the trust to nil.

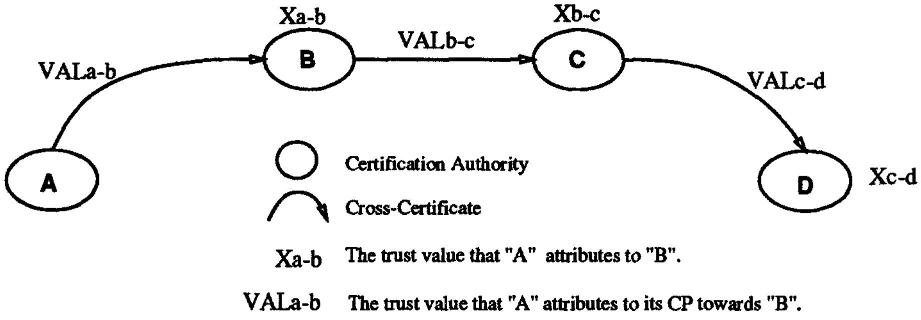
Xa-b
Xb-c
VALa-b
VALb-c
VALc-d
A
B
C
D
Xc-d
Certification Authority
Cross-Certificate
Xa-b The trust value that "A" attributes to "B".
VALa-b The trust value that "A" attributes to its CP towards "B".

Fig. 4. A Certification Path

- A composition law over the set of the trust degree attributed to the co-certifiers, may also distinguish two sorts of trust, the one given to co-certifiers and that attributed to certificates.
- and it may simply be the number of intermediate co-certifiers, supposing, hence, that all the CAs have the same degree of trust.

In the next paragraph we are going to iterate an evaluation algorithm on a graph representing a set of connected CAs, see figure 5. This evaluation algorithm is in fact the distributed routing algorithm [15].

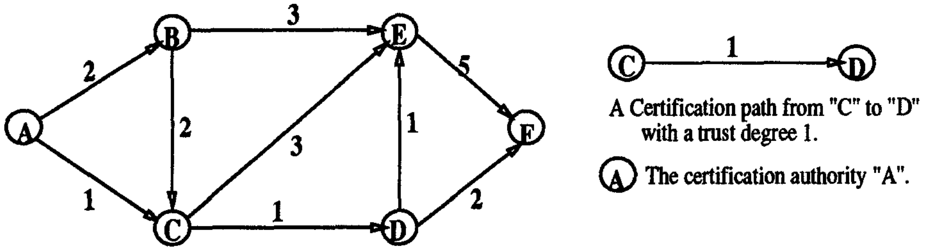
A
B
C
D
E
F
2
1
3
2
3
1
1
5
2
C
1
D
A Certification path from "C" to "D" with a trust degree 1.
A The certification authority "A".

Fig. 5. Example of an environment for the evaluation procedure

Before developing this example, let us describe the policy the concerned CA 'A'. This policy may be resumed by:

- Any CP is attributed to a trust value, this trust value is equal to ∞, when the CP is not established yet, or if the CP is considered to be insecure.
- CAs will add all received CPs to their database following a least cost strategy.
- All received CP are evaluated and stored in the database, and then forwarded to all neighboring CAs.

The evaluation function is chosen in a manner that:

$TV(CP_{A-F}) = \text{Min } (TV(CP_{A-I}) + TV(CP_{I-F}))$ Where:
I : an intermediate CA , I $\in$ { B,C,D,E }.
TV() : the trust value attributed to a certification path.

Initially, 'A' has only two CPs, CP_{A-B} and CP_{A-C}, with respectively their trust values 2 and 1. After information exchange and the application of the evaluation algorithm. One obtains new CP with new trust values as given in table 1.

Certification Path	*Trust Value*	*Intermediate CAs*
CP_{A-B}	2	None
CP_{A-C}	1	None
CP_{A-D}	2	C
CP_{A-B}	4	C,D
CP_{A-B}	5	C,D

Table 1. Table of all CPs known to 'A' after information exchange.

An important approach still unsolved is the propagation of metric values between CAs. In fact this propagation does not have any sense since the used metric is not universally adopted and differ from one domain to another. That is why the *Control* field of the CAC-Protocol is left as *ANY*.

We could however envisage to propagate "*characteristics*" of CAs, e.g. the trust they attach to their neighbours, which could be used by the local policy. In effect, Chimæra proposes to include a description of its local used metric besides the trust value in the Control field of the CAC-Protocol. This will allow a remote CA to convert this trust metric in accordance with both the local and the remote one. This approach is conform to that of policy requirements for inter-domain routing given by the RFC 1125 [17].

3.3 The Revocation

The use of Certificate Revocation Lists (CRLs) as defined in X.509 is one mean of propagating information relative to certificate revocation, though it is not a perfect mechanism. In particular, an X.509 CRL indicates only the age of the information contained in it; it does not provide any basis for determining if the list is the most current CRL available from a given CA. This subject is addressed in PEM [16], its proposed architecture establishes a format for a CRL in which not only the date of issue, but also the next scheduled date of issue is specified. Adopting this convention, when the next scheduled issue date arrives a CA will issue a new CRL, even if there are no changes in the list of entries. In this fashion each CA can independently establish and advertise the frequency with which CRLs are issued by that CA. Note that this does not preclude CRL issuance on a more frequent basis, e.g., in case of some emergency, but no system-wide mechanisms are architected for alerting users that such an unscheduled issuance has taken place. This scheduled CRL issuance convention allows users (UAs) to determine whether a given CRL is "out of date," a facility not available from the (1988) X.509 CRL format.

At this stage we moot the following question: what is the advantage of vouching certificates to a date later than the date of the next scheduled issue of a CRL. In fact, an alternative to this approach consists on the limitation of the lifetime of a certificate to the time used between to issuing of a CRL. Since neither a semantic nor a role has been established on the attribution of a lifetime to certificates, Chimæra proposes that a CA restricts the lifetime of its signed assets to the period of time between two productions of a CRL, which replaces the use of CRLs. This trend has the advantage of minimizing the number of read operations to the directory since a recipient has not, any more, to consult the CRL of the remote CA. This view has the inconvenient of increasing the charge of the CA because it has to recertify its assets more frequently. This does not have any repercussion on the system because a CA sign its assets off-line on an independant machine.

In this way, and by combining the limitation of the lifetime of signed assets and the use of evaluation coupled with the propagation of trust values, we obtain an elegent alternative to the revocation list mechanism.

4 Conclusion

In this paper, a new view of CAs is given with new functions and mechanisms. Similarities between routing and certification is demonstrated which allow the servicing of some routing concepts and mechanisms for certification purposes. A protocol has been also defined allowing cumminication between CAs, this new communication facility between CAs allowed the elaboration of new certification functions concerning the distribution of CAs and the manipulation of CPs.

These new concepts are integrated in Chimæra which is an authentication framework for the *OSI* environment. Chimæra does not integrate of these concepts in the directory DSP protocol, but it supposes on-line CAs reponsinbles of the acheivement of these concepts. Crucial tasks of CAs, like signing assets, are supposed to be done off-line or on separate machines. Finally, it seems of interest to couple routing with security functions in order to minimize the cost of these added services. In fact, many studies are undertoken to integrate security functions in the network layer of the OSI model, this prove the importance of this trend.

References

1. "Trusted Computer System Evaluation Criteria", DoD 5200.28-STD Department of Defense, USA, 1985
2. ISO 7498-2. Information Processing Systems - Open Systems Interconnection - Basic Reference Model - Part 2: Security Architecture.
3. ECMA Standard. Security in Open Systems. Data Elements and Service Definitions. Document Version: FINAL of July 1989 (Output of the 12th (Oslo) meeting).
4. R. L. RIVEST, A. SHAMIR, L. ADLEMAN. " A Method for Obtaining Digital Signature and Public Key Crypto-systems" Communications of the ACM February 1978 Vol 21, No2
5. Department of Defense Standard. Trusted Computer System Evaluation Criteria. "Orange Book". DOD 5200.28-STD of December 1985.
6. ISO-10021, Information Processing System - Text Communication - MOTIS (see also: CCITT-X.400-1988 Recommendations).
7. ISO-9594, Information Processing System - Open System Interconnection - The Directory (see also: CCITT-X.500 Recommendations).
8. C. Huitema, H. Afifi. "Solving Names Within X.500". INRIA Sophia-Antipolis, Technical report 1991.
9. ISO-9594, Information Processing System - Open System Interconnection - The Directory Part 8: Authentication Framework (see also: CCITT-X.509 Recommendations).
10. CCITT-X.800 Recommendations. Open System Interconnection, Security structure and application security architecture for open systems interconnection.
11. Jennifer G. Steiner, Clifford Neuman, Jeffrey I. Schiller. " Kerberos : An Authentication Service for Open Network Systems"
12. C. I'Anson, C.Mitchell. "Security Defects in CCITT Recommendation X.509 - The Directory Authentication Framework", ACM Computer Communication Review, VOL. 20, No. 2, April 1990, pp.30-34
13. A. Shamir, "Identity-based Cryptosystem and Signature Scheme". Advances in Cryptology: Proceedings of Crypto'84, Springer, Berlin 1985, pp. 47-53.
14. A. Tarah, C. Huitema, "CHIMÆRA: A Network Security Model". ESORICS 90, October 24-26, 1990, Toulouse, France.
15. A. Tarah, C. Huitema, "Certification and Routing Protocols", article submitted to the IPPS 92.
16. Kent (BBN). Network Working Group, "Privacy Enhancement for Internet Electronic Mail: Part II: Certificate-Based Key Management". Internet Draft.
17. D. Estrin., "Policy Requirements for Internet Administrative Domain Routing". Request For Comments 1125. November 1989.