Cert'eM: Certification System Based on Electronic Mail Service Structure

J. Lopez, A. Mana, and J. J. Ortega, "Cert'eM: Certification System Based on Electronic Mail Service Structure", *Secure Networking (CQRE99)*, LNCS vol. 1740, pp. 109-118, 1999.
NICS Lab. Publications: https://www.nics.uma.es/publications

Javier Lopez, Antonio Mana, and Juan J. Ortega

Computer Science Department, E.T.S. Ingenieria Informatica
Universidad de Malaga, 29071 - Malaga. SPAIN
{jlm, amg, juanjose}@lcc.uma.es

Abstract. Public-Key Infrastructures are considered the basis of the protocols and tools needed to guarantee the security demanded for new Internet applications like electronic commerce, government-citizen relationships and digital distribution. This paper introduces a new infrastructure design, Cert'eM, a key management and certification system that is based on the structure of the electronic mail service and on the principle of near-certification. Cert'eM provides secure means to identify users and distribute their public-key certificates, enhances the efficiency of revocation procedures, and avoids scalability and synchronization problems. The system, developed and tested at the University of Malaga, was recently selected by RedIRIS, the National Research and Academic Network in Spain, to provide the public key service for its secure electronic mail.

1 Introduction

There is wide agreement on the immense potential of Internet, specially for exciting new applications like electronic commerce, government-citizen relationships and digital distribution, but a significant part of the users are still reluctant to use the network for financially or legally sensitive data due to the lack of security. The growth and performance of Internet are adversely affected by security issues and by the open design of the network itself. Thus, despite its enormous possibilities, the Internet has not yet become a common vehicle for those applications because it is still too easy to intercept, monitor and forge messages, and even impersonate users [1].

Several systems, such as *Kerberos* [2,3] have been proposed to protect communications over public networks using symmetric-key cryptography. Those systems are not easily scalable for large groups of users belonging to different organizations. However, some efforts have been accomplished to solve this problem [4,5,6].

On the other hand, public-key cryptography [7] seems to be well suited to satisfy the requirements of the Internet, and is fast becoming the foundation for those applications that require confidentiality and authentication in an open network.

The widespread use of a global public-key cryptosystem is complemented by a *Public-Key Infrastructure* (PKI), an efficient and trustworthy mean to manage

R. Baumgart (Ed.): CQRE'99, LNCS 1740, pp. 109–118, 1999.

public-key values. A PKI is a vital element because it enables the application of the cryptosystem to the exchange of sensitive information between parties that do not have a face to face interaction.

This paper introduces Cert'eM, a new key management and certification system based on the electronic mail service structure, and it is organized as follows: section 2 presents the system structure and operation; section 3 summarizes additional features that improve the efficiency of the system; section 4 describes the protocol used to access the key servers and, finally, section 5 presents concluding remarks.

2 Description of the System

The fundamental principles of Cert'eM can be summarized in the following design goals:

- to use a CAs architecture that satisfy the needs of near-certification so the trust can be based on whatever criteria is used in real life;
- to eliminate problems associated with the revocation procedures and simplify the validation of certificates;
- to avoid architectures that yield scalability problems;
- to avoid the synchronization problems associated to schemes that keep multiple copies of the keys and certificates; and
- to minimize the network traffic, specially that generated by management operations.

2.1 Structure

The main element in the hierarchy is the *Keys Service Unit* (KSU), which integrates both key certification and certificate management functions. Cert'eM uses a scheme with various KSUs operating over disjoint groups of users, conforming a predefined hierarchy.

Figure 1 shows the system structure. The KSU hierarchy defined by Cert'eM is parallel to the hierarchy of Internet domains. A relevant feature is that KSUs are associated to the corresponding e-mail offices.

As shown in figure 2, every KSU is managed by a *Certification Authority* (CA). Additionally, it contains a *database* to store the certified keys of its users; each user public-key certificate is stored exclusively in the database of his/her KSU. The third component in the KSU is the *key server*, which receives requests and delivers the certificates to the requesters. A key server also manages a certificate cache that keeps some of the external certificates recently received. The certificate cache, carefully designed, enhances the efficiency of the system without introducing any security risk. Furthermore, any CA can define its own cache policy according to its users needs.

Each CA can set restrictions to limit the users or KSUs allowed to access the server. This feature provides the CA with a useful tool to avoid abuse and to balance the workload between different servers.

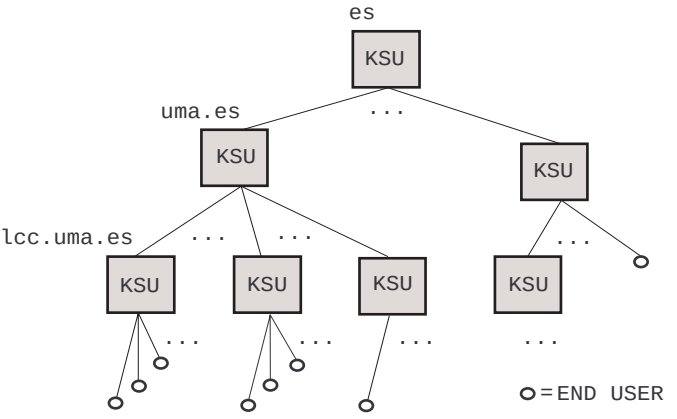

Fig. 1. Hierarchy of Cert'eM Nodes

The certified keys are managed solely by the corresponding CA; therefore, key updating and revocation are local operations without influence in the rest of the system.

We must underline that no *Certificate Revocation List* (CRL) is used in the design. The validation of a certificate is achieved using the *Validity Statement* (VS), a timestamp statement signed by the CA attesting that the certificate has not been revoked at the time of issuance of the VS. A certificate is considered *expired* if the validity period has finished. If a certificate has not expired we call it an *active* certificate. An active certificate is *valid* if it has not been revoked; therefore in order to validate active certificates the CA simply issues a VS.

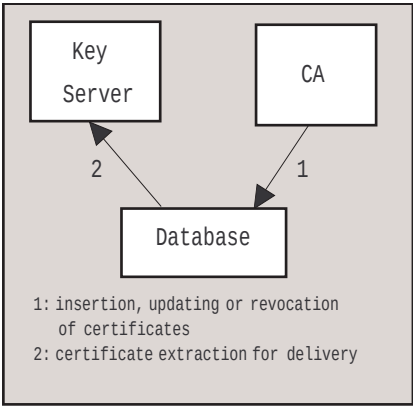

Fig. 2. KSU Components

2.2 System Operation

Cert'eM defines a special user, *ca@<domain>*, in every KSU in order to denote the correspondent CA. The certificate of any CA is stored exclusively in the database of its parent KSU. Exceptionally, the key of a CA located at any top-level domain is stored in the database of its own KSU, certified by the domain registering authority (e.g. *ICANN*). Keys distributed by any KSU are always certified by the corresponding CA; thus, in the subsequent discussion, we will use the terms 'key' and 'certificate' equivalently.

The logical structure of the data transmitted by a KSU in response to a certificate request is important in order to clarify the key distribution procedure. A certification response consists of two components:

- a X.509v3 certificate [8] containing, among other information, a serial number and the expected life of the certificate (the validity information);
- the VS signed by the CA, containing the certificate serial number and the time of issuance.

Other systems, like *SPKI/SDSI* [9,10], propose a similar mechanism called *one-time revalidation* (OTR). But for our purposes this solution is not convenient because it does not provide tools to limit the use of that "pre-validated" certificate in the future.

Therefore, in our scheme, the certificate does not need to be issued on-line; however, it still provides a good degree of security against attacks that try to use revoked certificates.

We describe now the sequence of actions that are carried out when any user (requester) wants to get the public key of another user (addressee). This process starts when the e-mail address of the last one is provided by the requester to his/her KSU, and this one, in turn, conducts the request to the addressee KSU, whose database contains the key. Such operation is easily done because the system can determine the KSU to be contacted from the email address provided.

Previous actions are showed in figure 3 (left). In this case, the figure depicts the information flow produced when user *Bob* (*bob@r.s.t*) requests the key of user *Alice* (*alice@x.y.z*). As shown, *Bob* requests Alice's key from his own KSU and this one directs the request to the KSU located at the *x.y.z* node. The response from Alice's KSU is then forwarded to *Bob*.

Bob must request the key from his KSU due to the access restrictions that other KSUs set, and also to take advantage of the certificate cache of his KSU. If considered, *Bob* can also request the certificate of *ca@x.y.z* from the KSU located at *y.z*, obtaining a new certificate that proves the authenticity of the first one. This is depicted in figure 3 (right). The ascending validation process can continue until a top-level node is reached. If no KSU is present at *y.z* (i.e. the domain does not support Cert'eM system), the key of *ca@x.y.z* is automatically requested from the parent node, that is, *z*. This allows Cert'eM to be used even in case of incomplete structures.

Some similarities can be found between Cert'eM and the *Secure-DNS* proposal [11,12]. Both use the Internet domain name hierarchy to find the location

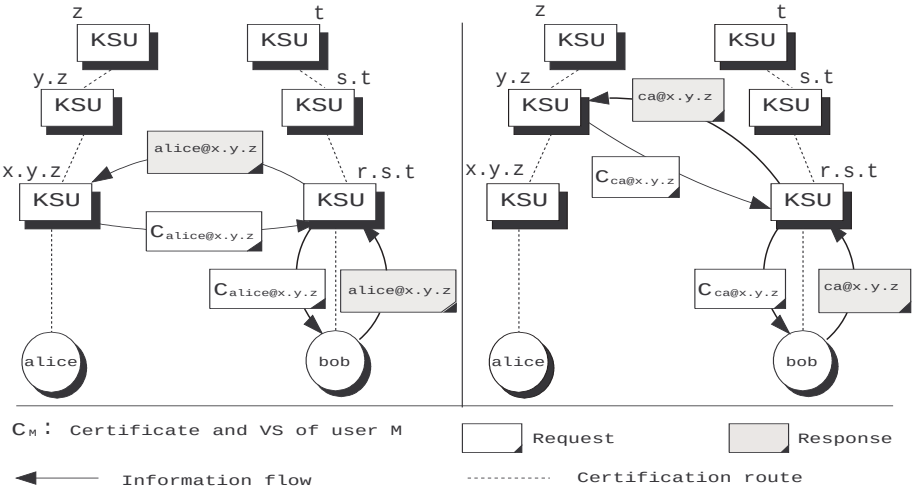

Fig. 3. Certificate Request (left) and Validation (right)

where a particular key is stored, but Secure-DNS uses the Name Server files while Cert'eM uses the e-mail offices. Our choice is based on the following reasons:

- Opposite to e-mail offices, it is usual that several domains share the same DNS; therefore, it is frequent that DNSs not closely related to users, and their CAs may not have direct knowledge of the users identities, being more vulnerable to impersonation.
- DNSs are intended to store information about domains, not about users. As a consequence, there is a registration procedure for a new domain but not for a new user of one of the registered domains. In fact, there is no need that a final user ever interacts with the DNS to get access to Internet, but users are forced to interact with e-mail offices to set up an e-mail account.
- DNSs use caching and lifetime mechanisms that could yield inaccurate or false information in some situations. This feature can be used to attack the system.

For these reasons the Secure-DNS scheme cannot guarantee the link between real-world users and keys (not conforming with article 8.2 in [13]).

3 Additional Features

One of the advantages of Cert'eM is that, in case the private key of a user is compromised or lost, the associated public key can be revoked or replaced without possessing the private one. This is possible because there is an entity (the CA), responsible for the maintenance of the database of certificates, which can perform a real-world user identification. Opposed to other systems that

require that the user generates a "suicidal note" to be used in case the key is compromised or lost [14], Cert'eM users do not need to take any prevention measures for this circumstance.

In case the key of a CA is changed, existing certificates must be discarded, and the CA must reissue all the certificates. Other systems need to notify this event to users and request old certificates in order to re-certify their keys and distribute the new certificates. In Cert'eM, any CA keeps the certificates of its users in a local database of the KSU, and there is no need to send new certificates and notify the invalidation of the previous ones. Consequently, the change of the CA key is transparent to users.

Usually, the need to check CRLs for certificate revocations becomes a performance handicap. For this reason, systems that use CRLs or similar mechanisms (e.g., On-line Certificate Status Protocol [15], or Suicidal Bureaus [14]) to invalidate certificates incorporate solutions to minimize the number of accesses needed to verify a certificate, but these solutions are sometimes artificial and not efficient. Therefore, avoiding the use of CRLs has been considered one of the priority goals in the design of Cert'eM.

In order to achieve a design that does not expose the problems of using CRLs while still retaining their benefits, all the information related to the certification of a specific user must be located and managed at the corresponding KSU. In case a CA decides to record certificate invalidation events, a *Local Invalidation Log* (LIL) can be managed locally. Notice that a LIL is completely different to a CRL because the LIL will be used exclusively by the CA.

When a user certificate needs to be invalidated (because his/her key has been lost or compromised, or because the CA has reasons to cease certifying the user) the CA simply deletes the certificate from its database and, if appropriate, stores the revoked certificate in a LIL. This procedure is simple, immediate, requires no communication and can provide proofs of the certificate revocations in case the CA needs those proofs.

Once the revocation takes place, existing active certificates are not useful any more because no VS will be issued to make them valid. The use of the VS prevents attacks based on old certificate reuse.

3.1 User Identification

When designing a key management system that achieves secure user identification it is necessary to take into account the difference between the real world (where people, companies and computers are), and the Internet world (where names, keys and certificates are).

It must be pointed out that many of the identity certificates presently used by many schemes are based exclusively in a contact, through Internet, between the user and the CA. This is clearly unsatisfactory because the requester of a certificate will usually require some guarantee of the link between the identity of the user in the real world and his name in the Internet world. Therefore, in these certificates, trust is misinterpreted from the start.

The design of Cert'eM guarantees that a CA will only certify the keys of those users closed to it. Therefore, a formal identity verification procedure has been established to give a legal meaning to certification process [16]. Consequently, a link is established between the identity documents (valid in the real world), a distinguished name in the Internet world (the e-mail address) and a cryptographic key.

It has been described how Cert'eM uses the e-mail addresses to identify users. There are two common criticisms about the use of e-mail addresses as distinguished names. Firstly, it is claimed that the relationship between a person in the real world and an electronic mail address is not one-to-one because a user can have several e-mail accounts and different aliases. Besides, there are certain e-mail addresses that do not represent a single user but a group of them. Secondly, it is also claimed that, in some cases, the alias file can be modified without administrator or root permissions. Cert'eM has been designed to overcome these problems by isolating the certification management from the email account management.

4 Key Server Access Protocol

In this section we introduce the protocol that describes how both, individual users and other key servers, access a KSU. A TCP connection to the port 850 is used for Cert'eM service. The requests are represented in a Client/Server scenario, where individual users or key servers can play the client role; for instance, consider a request from user *bob@r.s.t* (client) to the KSU located at *r.s.t* (server), followed by a request from the KSU located at *r.s.t* (now client) to the KSU located at *x.y.z* (server). In the subsequent description C will be used to denote a generic client and S to denote a generic server.

4.1 Protocol Data

We will use the following data structures as part of the protocol:

<*clientID*>: Identification of the client.

<*userID*>: The e-mail address (with format <*name*>@<*domain*>) of the user whose key (certificate) is requested. Cert'eM uses the <*domain*> to determine in which KSU the key resides.

<*cert*>: An X.509v3 certificate containing among other information: the user identification (equivalent to <*userID*>), the user's public key, a certificate serial number that is unique for the issuing CA and the expected activity period life of the certificate. This record is kept in the KSU database, so there's no need to produce it online.

<*vs*>: A timestamp statement containing a certificate serial number, and the time of issuance of this <*vs*>, signed by the CA. It is used to guarantee that the certificate with that serial number was not revoked at the time of issuance. Opposite to the <*cert*> this record is produced online.

<*certID*>: Certificate identification consisting on the <*userID*> of the addressee user and the certificate serial number of the active certificate to be checked.
<*nack*>: Negative acknowledgement. It guarantees that there is no key associated to the <*userID*> requested.

4.2 Protocol Description

The protocol is structured in three phases: connection, transaction and termination.

Connection Phase

The connection is established with the following message:

C : HELLO [<clientID>]

where <*clientID*> is optional, depending on the particular KSU security policy to be implemented.

Each CA can set restrictions to limit the users or computers allowed to access the server. When a server receives this message, it checks whether or not <*clientID*> is allowed to establish the connection. Afterwards, the server sends one of the following messages as a response:

S : +OK – the client has permission
S : -ERR1 – the client host is not allowed
S : -ERR2 – the client is not allowed

Transaction Phase

When the connection is successfully established the client can start requesting keys. For this purpose the following message is used:

C: GET KEY <userID>

When the server receives the previous message the following situations can arise:

1. The requested <*domain*> coincides with the <*domain*> of S (i.e. the requested key belongs to a local user of S). The response is:
   S : KEY <cert> <vs>
   if the key was found, or
   S : -NSK <nack>; –no such key
   if the key was not found.
2. The requested <*domain*> does not correspond with that of S.
   a) The requested <*name*> is *ca*.
      i. If the <*domain*> of S corresponds to the parent of the requested <*domain*>, then the key should reside in the database of S; therefore, the case is managed as a local certificate request (case 1).
      ii. Otherwise, the key is requested from the KSU located at the upper node of <*domain*>. If there is no KSU in that node the request is redirected to the succeeding upper one until the top-level node or S are reached.

b) The requested <*name*> is not *ca*.
   i. If <*domain*> does not exist the server returns an error message:
      S: -ERR3
   ii. Otherwise, a new connection is established to request the key from the KSU located at <*domain*>. The result of this new request is forwarded to the requester.

In case a client already has an active certificate there's no need to request the complete certification information. The key check message is used in this case.

C: CHK KEY <certID>

To which the server responds:

S: VS <vs>

if the key is found and has not been revoked; otherwise, the request is carried out as a *GET KEY* request.

Termination Phase

This phase is meant to inform the server that the client has finished requesting keys. To do so the client sends this message:

C: EXIT

5 Conclusions and Future Works

Several PKIs have been proposed in the literature to meet the security needs of different network applications. This paper presents a new scheme, Cert'eM, a key management and certification system that is based on the structure of the electronic mail service and on the principle of near-certification. It provides secure means to identify users and distribute their certificates, eliminating problems associated to common revocation procedures, and simplifying the validation of certificates.

The system has been deployed for certified electronic mail in the University of Malaga, and presently services about forty thousand users distributed in more than twenty KSUs. Additionally, this system was recently selected by the National Research and Academic Network in Spain to provide the public key service for its secure electronic mail service, and is presently being tested by a restricted group of users, as the previous step to its distribution to the community of RedIRIS users. This is producing valuable information for future improvements.

Among the ongoing projects, we point out the utilization of Cert'eM in corporate extranets, as well as several applications in the University environment like computer system access controls and secure exchange of official documents.

References

1. U.K. Department of Trade and Industry, "Building Confidence in Electronic Commerce - A Consultation Document", March 1999.
2. J. Kohl, "The Use of Encryption in Kerberos for Network Authentication", *Advances in Cryptology, Proceedings of CRYPTO '89*, Springer-Verlag, 1989, pp. 35-43.
3. J. Kohl, B. Neuman, "The Kerberos Network Authentication Service (V5)", RFC 1510, 1993.
   `http://www.ietf.org/rfc/rfc1510.txt`
4. D. Davis, "Kerberos Plus RSA for World Wide Web Security", *First USENIX Workshop on Electronic Commerce*, 1995, pp. 185-188.
5. R. Ganesan, "Yaksha: Augmenting Kerberos with Public Key Cryptography", *Internet Society Symposium on Network and Distributed Systems Security*, IEEE Press, 1995, pp. 132-143.
6. J. Schiller, D. Atkins, "Scaling the Web of Trust: Combining Kerberos and PGP to Provide Large Scale Authentication", *USENIX Technical Conference*, 1995.
7. W. Diffie, M. Hellman, "New Directions in Cryptography". *IEEE Transactions on Information Theory*, IT-22, n. 6. 1976, pp. 644-654.
8. International Telecommunication Union, Itu-t Recommendation X.509. *Information Technology - Open Systems Interconnection - The Directory: Authentication Framework*, 1997.
9. C. Ellison, "SPKI Requirements", Internet draft, May 1999.
   `http://www.ietf.org/internet-drafts/draft-ietf-spki-cert-req-03.txt`
10. C. Ellison, W. Frantz, B. Lampson, R. Rivest, B. Thomas, T. Ylonen, "SPKI Certificate Theory", Internet draft, June 1999.
    `http://www.ietf.org/internet-drafts/draft-ietf-spki-cert-theory-05.txt`
11. D. Eastlake, "Domain Name System Security Extensions", RFC 2535, March 1999.
    `http://www.ietf.org/rfc/rfc2535.txt`
12. D. Eastlake, O. Gudmundsson, "Storing Certificates in the Domain Name System (DNS)", RFC 2538, March 1999.
    `http://www.ietf.org/rfc/rfc2538.txt`
13. European Commission, "Proposal for a European Parliament and Council Directive on a Common Framework for Electronic Signatures", COM(1998) 297 final, 1998.
    `http://www.ispo.cec.be/eif/policy/com98297.html`
14. R. Rivest, "Can we Eliminate Revocation Lists?", *Proceedings of the Second International Conference on Financial Cryptography, FC '98*, Springer-Verlag, 1998.
15. C. Adams, M. Myers, A. Malpani, R. Ankney, S. Galperin, "X.509 Internet Public Key Infrastructure Online Certificate Status Protocol - OCSP", Internet draft, April 1999.
    `http://www.ietf.org/internet-drafts/draft-ietf-pkix-ocsp-08.txt`
16. B. Wright, "Making Numbers Ceremonial: Signing Tax Returns with Personal Identification Numbers", personal communication, 1998.