A Public Key Infrastructure for User Identification

Javier Lopez, Antonio Maña and Juan J. Ortega

Computer Science Department
E.T.S. de Ingenieria Informatica. University of Malaga e-mail: {jlm, amg, juanjose}@lcc.uma.es

Key words: Digital Certification, Public-Key Infrastructure, Identification Sytems, Key Service Unit, Internet Domains Hierarchy

Abstract: While there is wide agreement on the immense potential of Internet, its growth and performance are adversely affected by security issues. Despite its impressive size, scope and reach, the Internet has not yet become a common vehicle for many of these new possibilities. Progress in fields as electronic commerce and government-citizen relationships have been limited by the open design of the network itself. Today, Public-Key Infrastructures are the basis of the protocols and tools needed to guarantee the security demanded in those fields. Trust management and user identification are also important issues that remain unresolved. This paper introduces a key management and user identification system, named Cert’eM, that is based on the electronic mail service. Cert'eM provides important advantages over existing Public-Key Infrastructures and user identification proposals.

1. INTRODUCTION

Many users view the Internet as a universal communications medium that can replace telephone, television and radio [McCu96]. Although Internet is growing at explosive rates, it is still constrained by security issues. Public administration and commercial companies have adopted the technology in very limited aspects, essentially as an informational vehicle. Progress in the fields of electronic commerce and government-citizens relationships have been limited by the open design of the network itself [Tsuj96] [Ilpf97]. But today's technology provides means for the interception, monitoring and forging of messages, and even impersonation of users on the Internet and, consequently, people are reluctant to use the network for financially or legally sensitive data. The consequences of abuse, misuse, and failure can include: direct financial loss resulting from fraud; theft of valuable confidential information; loss of opportunity through disruption of service; unauthorised use of resources; loss of customer confidence or respect; and costs resulting from uncertainties [FoBa97]. Future widespread use of the Internet for these purposes may not be able to rely worldwide on current mid-scale solutions.

The need for user identification is evident in many public administration and electronic commerce applications. For example, there are multiple instances where two unknown officials in different branches of the public administration need to exchange some documents. This necessity is not adequately resolved by present systems.

Several systems, such as Kerberos [Kohl89][RFC1510] 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 [Davi95][Gane95][ScAt95].

On the other hand, public-key cryptography [DiHe76] seems to be well suited to satisfy the requirements of the Internet, and is fast becoming the foundation for those applications that require security and authentication in an open network.

The widespread use of a public-key cryptosystem requires a *Public-Key Infrastructure* (PKI), an efficient and trustworthy mean to manage public-key values. A PKI is a vital element because it provides confidentiality, integrity, authentication and non-repudiation services; that is, it ensures the security of electronic transactions, and the exchange of sensitive information between parties that do not have a face to face interaction. It is impractical and unrealistic to expect that each user in a large-scale network will have a previously established relationship with all other users. So, without a functioning infrastructure, public-key cryptography is only marginally more useful than traditional secret-key cryptography.

Certification of the public keys is the most important function of a PKI. A PKI user trusts the *Certification Authorities* (CAs). These entities issue *certificates*, computer-based records that attest to the connection of publics keys to identified subscribers (*identity certificates*), or that truly describe attributes of those subscribers (*attribute certificates*). To provide assurance as to the authenticity and integrity of the certificates, the CAs attach their own digital signatures to the certificates before storing them in a repository.

The second basic PKI process is certificate *validation*. The information signed by the issuer can change over time. A certificate user needs to be sure that all the data it contains are trustworthy and up to date. Two methods are used: *on-line*, if the user requests the CA to confirm the certificate validity each time it is used; and, o*ff-line*, if the CA includes a validity period in the certificate to let people know when a certificate expires.

A process related to validation is the certificate *revocation*. With public-key cryptography, the security of private keys is a problem. It is inevitable that someone's key will be lost o compromised, either through carelessness or a successful cryptanalytic attack. In addition, there are times –such as when a company goes out of business, or an employee quits, is fired or transferred to a new position– when a key may no longer be needed or used. Thus, there will be times when a key needs to be revoked before it expires.

The revocation problem is trivial in case of on-line validation; the CA simply states that the certificate is not valid. But, if the validation action is off-line, the problem is to notify people that they should no longer rely on a key. In this case, the common solution is to use a *Certificate Revocation List* (CRL), a database of certificates that have been revoked before their expiration date. This approach introduces a performance degradation factor and does not conform with the Annex II-b of the proposed Directive about digital signatures of the European Parliament [EU98]

In the design of a key management system that has to deal with trust in some way, 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). Trust is originated in the real world, based on whatever criteria are important to the application being implemented. That trust is then abstracted and mechanized using certificates. Any system that does not follow this basic rule is bound to fail when used for identification in applications where the real-world user is concerned and legal validity is necessary.

This paper presents a key management and user identification system based on electronic mail. In order to introduce the system, we review, in section 2, the problem of identification. Section 3 resumes the outcome of the analysis of the most relevant Internet PKIs. This analysis was significant to set the design goals of the new system. The goals and a general description of the system are discussed in section 4. Finally, section 5 summarizes the conclusions.

2. USER IDENTIFICATION

2.1 The problem of user identification

There are many different proposals dealing with user identification and public-key management. Let us take first a deeper look at the problem of identification. *Identity* is defined as “the condition of being a specified person” or “collective aspect of the set of characteristics by which a thing is definitively recognizable or known”. But identities are not intrinsically restricted to one per physical person [Clar97]. A person may adopt different identities at various times during a life-span, and some individuals maintain several at once. Therefore identity is a vague concept and is not a useful start point for our purposes. Oppositely, *identification,* has a clearer definition: “identification of people and things is the process of recognizing or choosing them because they have a particular quality”. The definition of *identify* includes “something that identifies you makes you easy to recognize because it makes you different in some way”.

In the real world identification is achieved in different ways; for example a person can be identified using a document (v.g. the passport), a business can be identified by its physical location, etc. On the other hand, computers do not usually identify their users. The (absolute) identification of the real-world person using a particular computer is not necessary for many computer systems (particularly for those not connected to an open network). Instead, the relative identification is enough; i.e. the computer does not need to know your legal name; it simply needs to verify that the person sitting at the keyboard is authorized to access the system. We can categorize identification systems in four groups [GaSp97]:

Password-based systems. In these systems the user knows some information that no one else knows. When the user is requested, he provides all or part of the information to the verifier. The classical username/password schemes continue to be the most widely used identification systems because of their simplicity and ease of implementation. Some other examples of this kind of systems include credit card and cellular phone PINs.

Important properties of these systems are transitivity and not exclusiveness. Once a user discloses the secret information to another user, this one is undistinguishable from the first one and acquires the same capabilities; thus, the system will identify both users as the same.

Problems associated to these systems when used for real-world identification are:

- The password (or the verifying information) has to be stored in the computer prior to any access and it has to be provided using some secure channel.
- In case of remote access, the password can be intercepted in the way to the destination computer. If this is the case, any attacker that learns the password can impersonate the owner.
- Passwords are frequently forgotten so there has to be some mechanism to establish a new password without knowledge of the previous one. This introduces a serious security breach.
- The users usually select "bad quality" passwords (easily guessed passwords).
- Users share their passwords.

Physical tokens. A token is a physical object the user carries with him that somehow proves his identity. The most typical examples are access cards that grant access to restricted areas and smart cards. Physical tokens are transitive but also exclusive because when a user gives the token to another user this one gains full capabilities of use. Simultaneous identification of both of them is not possible.

The problems these systems presents are:

- The token does not really prove user's identity. Anyone who gets possession of the token is positively identified.
- If the token is lost or damaged the system will not be able to identify the user.
- Today, most kind of tokens can be easily copied or forged.

Biometrics. These systems use data extracted from some biologic characteristic of the real-world user (iris image, fingerprints, voice, handwriting, etc.) using a device that is not likely to produce the same value for different persons. Oppositely to the previous types this scheme is not transitive and, therefore, exclusive. There are many problems associated to this scheme:

- The biometric profile of a user has to be stored in the computer before the user can be identified.
- The devices used to extract the biometric characteristics of the user are expensive.
- The devices are also vulnerable and special protection is required to avoid sabotage or fraud.

Biometrics can be a reliable way to establish user identity but, because of the many problems that they introduce, they are not widely used. Due to the problems associated to the biometric devices, biometrics are not well suited for Internet user identification (although they can be very useful for other applications like physical access control). To get a better understanding of why this is so let us introduce two definitions:

- *Sense*: the semantic value of a phrase or name; meaning; connotation; an essential property of a thing.
- *Reference*: the truth value of a phrase; the syntactic or direct specific value of a phrase or name; denotation.

Biometrics will still only provide references, not sense. So biometrics still needs trust to link sense to reference, like any other certification system. Biometrics systems are not self-secure.

Location. Several companies have announced identification systems based on the *Global Positioning System* (GPS). These systems have the same drawbacks as the biometrics; they are expensive and exposed to threats related to the base equipment. There are devices (designed for military use to defeat missile navigation systems) that can forge satellite GPS signals.

All of the former identification systems are well suited for small communities of users and, thus, we could call them private identification systems. But Internet is a very large community of users and many of them have no previous relationship.

Many interests need to be balanced when considering the design of identification schemes. In order to guarantee secure user identification for applications that deal with real-world users in a worldwide environment and over an open network, such as Internet, previous identification systems are not applicable. In the real world we have ways to establish the identity of a person (Passports, National Identity Cards, Driver Licenses, etc). Recalling from our previous definitions, if we want to securely identify Internet users we should find “something that makes users easy to recognize” or “something that makes users different in some way” that is usable in the digital world. Cert’eM uses the combination of the e-mail address and the cryptographic key of the user (which is in short a large number) to make him different. This combination fulfils most of the desired criteria for human identifiers [Clar97].

2.2 Legal aspects

The European Parliament proposal on a common framework for electronic signatures [EC98] establishes the requirements for the validity of electronic signatures and the certification services. It takes special care to protect the privacy of the users while allowing real-world identification in cases allowed by the law. Annex II: “Requirements for certification service providers” contains some of the most important issues regarding certification. It states that certification service providers must:

- operate a prompt and secure revocation service;

- verify by appropriate means the identity and capacity to act of the person to which a qualified certificate is issued;
- use trustworthy systems, and use electronic signature products that ensure protection against modification of the products so that they can not be used to perform functions other than those for which they have been designed; they must also use electronic signature products that ensure the technical and cryptographic security of the certification processes supported by the products;
- record all relevant information concerning a qualified certificate for an appropriate period of time, in particular to provide evidence of certification for the purposes of legal proceedings. Such recording may be done electronically;
- inform consumers before entering into a contractual relationship in writing, using readily understandable language and a durable means of communication, of the precise terms and conditions for the use of the certificate, including any limitations on the liability, the existence of a voluntary accreditation and the procedures for complaints and dispute settlement.

Later amendments state that the certification service must support the use of pseudonyms, and that public authorities access to the real-world identity when lawfully allowed must be provided. Cert'eM conforms with these requirements.

3. ANALYSIS OF EXISTING PKIS

Many proposals have been introduced as PKIs for a general use in the Internet, but only a few have achieved an extensive use. This section summarizes their essential features.

A PKI security policy must establish which entities can be Certification Authorities. Some of these policies propose, as Cert'eM does, that only those entities satisfying certain restrictive conditions must work as CAs. Within this group of PKIs, the location of the Authorities inside the infrastructure is considered a basic characteristic. Some systems use a general hierarchy where each CA certifies the CA located in the immediately upper node in the structure (parent node), and all the CAs in the nodes directly below (children nodes), producing a tree-like structure. This configuration originates *certification paths*. Directly related to a PKI of this type there exists a special CA, the *Root Authority*. This Authority is located in the root of the structure and not only performs CA functions, but it also establishes a global policy (or Certification Practice Statement) for the overall infrastructure; that is, it is responsible for the policy development and coordination in the system.

The certification relationships are not limited to parent and children. A CA can certify another CA located in a different branch of the hierarchy, producing a *cross-certification*. Cross certificates allows greater flexibility and short certification paths. They require only one certificate signature verification in addition to the user certificate verification that must always be done. But problems arise when the number of cross-certificates is high. In that case, this feature does not yield a viable architecture for global scale PKIs.

On those systems that are intended to provide user identification functions, there is usually a *Registration Authority* (RA) that is responsible for the registering and authorization of access to the system.

Some Internet PKI proposals, as PEM (Privacy Enhanced Mail [RFC1422] [RFC1424]), use the top-down hierarchy. It is similar to the general one, except that CAs only certify their children nodes and the Root Authority is the source of all certification paths. There are three types of PEM certification authorities:

– *Internet Policy Registration Authority* (IPRA): This authority, operated by the Internet Society, is the root of the PEM certification hierarchy. All certification paths start with it.

– *Policy Certification Authorities* (PCAs): PCAs are at level 2 of the hierarchy, each PCA being certified by the IPRA. Distinct PCAs aim to satisfy different user needs.

– *Certification Authorities* (CAs): CAs are at level 3 of the hierarchy and can also be at lower levels. Those at level 3 are certified by PCAs.

The identification issues are not specified by the PEM RFCs. PEM simply proposes that every PCA must establish and publish a statement of its policy with respect to certifying users or subordinate certification authorities.

In the PEM system, every entity has a distinguished name, based in the standard X.500 [ISO88], which presumes the existence of a global directory that provides a format to name every individual, organization, computer, etc. One decade later, the solution of the X.500 directory has not reached its global implementation. Moreover, the IAB has recently considered a few protocols as not useful because they have failed to catch on, even though they have been available for some time [RFC2316]. PEM is included in that group.

Based on the work of PEM, the IETF PKIX Working Group has proposed an infrastructure [PKIX97] that covers automatic identification, authentication, access control and authorization functions using the X.509 V.3 certificates [ISO96]. The draft papers elaborated by this group have not been adopted as standards yet because some implementation issues are not definitely closed. The basic element in this scheme is called *repository* a system or collection of distributed systems that store certificates and CRLs and serves as a means of distributing these certificates and CRLs to end entities.

Other important systems based on a top-down hierarchy are the United States Federal PKI [Chok94] [NIST96], the Canadian Federal PKI [CSE98] and the Australian Government one [OGIT98].

The recent extensions to the *Domain Name System* (DNS) [RFC1101] establish another proposal that allows authentication through digital signatures. Its name is Secure-DNS [RFC2065]. These extensions describe a hierarchic PKI, integrated in the DNS database by adding it a group of registers that contain the public keys of the users in the domain. But, as we will see later, name servers expose several problems to store public keys because quite often DNS can not be tightly coupled with its users and therefore the link between real-world users and keys cannot be guaranteed (Article 8.2 in [EU98]).

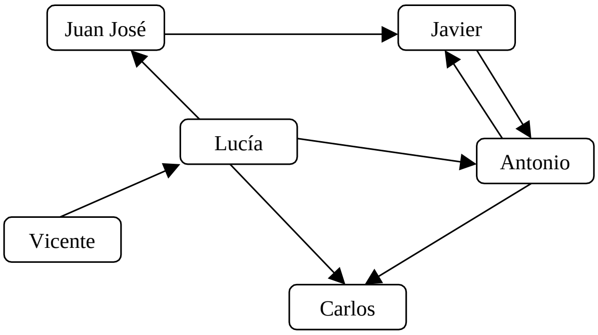

Figure 1. PGP Web of Trust Example

There are other security policies that do not have a fixed PKI structure. In those policies every user can act as a CA, with full autonomy to assign trust. The most important example of this type of unstructured PKIs is Pretty Good Privacy (PGP) [Zimm95], where users built their confidence on other users certificates. Therefore, a *web of trust* is created between users, as showed in figure 1. The system provides a high flexibility and easy deployment because users issue their own certificates. This is the best option for the communication among a closed set of persons, as a group of friends or the workers of a firm.

A very important problem of PGP is that no entity is responsible if (or when) something goes wrong -somebody is not who he claims to be-, not even those users that signed the impostor's key. The use of PGP in a commercial situation is difficult and may not adequately protect the business interests involved, as they usually need to be guaranteed in well-defined contracts with loss responsibilities and fines. There are some problems associated to key administration too; for instance, revocation process is problematic, since multiple users may sign the same public key. Further, PGP does not scale well in size due to the high number of certificates needed for a global communication, and due to the length of the certification paths. Likewise, it does not scale well in time because of the maintenance problems of the CRLs. Again, within a circle of close friends this is not important. PGP is a good example of a private identification system.

Finally, if quantitative trust metrics are added to the arcs of the certification graphs, the task of finding the “best” (most trusted) path between two users is a NP-complete problem [Kent97].

Other proposals as SPKI (Simple Public Key Infrastructure) [SPKI98a], and SDSI (Simple Distributed Security Infrastructure) [RiLa97] are similar to the previous one in the sense that no global PKI is used. The main objective of both proposals is to provide mechanisms to support security in a wide range of Internet applications that require the use of public key certificates and the ability to access them. They intend to produce a certificate structure and operating procedures to meet the needs of the Internet community for trust management.

Both schemes share the idea that every subject must be unequivocally identified by a number, the public key, and not by a common name (as the X.500-based proposals). A certificate can be created and signed up by any user due to the lack of formal notion of CAs. Actually, both proposals are merging [SPKI98b].

4. DESCRIPTION OF THE SYSTEM

4.1 Design goals

When a new system is going to be designed, some decisions must be taken in order to define the system capabilities and the equilibrium between the design goals. To delimit the scope of the system we had to decide whether it would be an identification system or a trust-management system. There are two concepts that need to be reviewed: validity and trust. A user's key is valid if it is possible to determine that it belongs to that specific user, this introduces the concept of identification of users. A key is trusted if there is confidence in some fact about the corresponding user. A definition of trust is: "that essential to a communication channel but cannot be transferred from a source to a destination using only that channel. Trust is a very wide concept [Gerc98] and the complete concept is not needed in most applications, also there are some specific solutions being developed in this aspect [SPKI98b] therefore the system presented is restricted to identification, which is a special case of trust.

Once the analysis of the PKIs referred in the previous section was completed, the fundamental principles of Cert'eM were defined and the following goals were proposed:

- provide secure means to identify users and distribute their public keys;
- 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;
- eliminate problems associated with the revocation procedures and simplify the verification of certificates;
- avoid architectures that yield scalability problems;
- avoid the synchronization problems associated to schemes that keep multiple copies of the keys and certificates; and
- minimize the network traffic, specially that one generated by maintenance operations.

4.2 Structure

The mentioned analysis allowed us to conclude that, in order to obtain a satisfactory degree of security, only certain entities should be authorized to certify keys. Cert'eM proposes a scheme with various CAs operating independently over different groups of users.

Opposed to other designs, Cert'eM proposes a predefined hierarchy. The main element in the hierarchy is the *Keys Service Unit* (KSU), which integrates the key certification, maintenance and distribution functions. Figure 2 shows the basic structure of the system.

KSUs are associated to any e-mail office corresponding to an Internet domain, as it is shown for our local domain *lcc.uma.es*, where *es* corresponds to the RedIRIS central server. Therefore, the KSU hierarchy is parallel to the hierarchy of Internet domains [POLM97].

Every KSU is managed by a CA (Figure 3). Additionally, it contains a database to store the certified keys of its users. It must be emphasized that each user’s public key is stored exclusively in the database of his KSU. The third component is the key server, which receives requests and delivers the certificates.

Cert'eM does not use RA's, instead, this duty is managed by the CAs. Figure 3 shows that the certified keys are managed solely by the corresponding CA; therefore, key actualization and revocation are local operations that do not affect to the rest of the system.

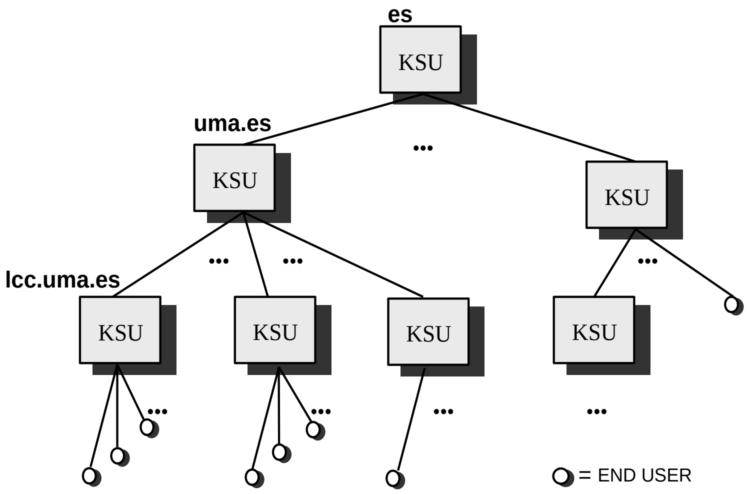

Figure 2. Hierarchy of Cert'eM Nodes

One of the advantages of the system is that, in case the private key is compromised or lost, the associated public key can be revoked or replaced without the knowledge of the private one. The user does not need to take any prevention measures for this circumstance. Furthermore, if the key to be changed or revoked belongs to a CA, there is no need to send new certificates and invalidate the previous ones, as is usual in most of the hierarchic systems. So, it is not necessary to define a protocol to notify the change and re-certify the keys of the users subordinated to that CA.

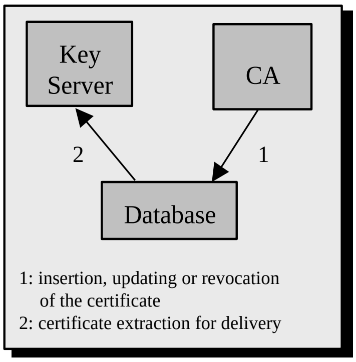
Key Server
CA
2
1
Database
1: insertion, updating or revocation of the certificate
2: certificate extraction for delivery

Figure 3. KSU Components

Frequently, systems that use CRLs or similar mechanisms (i.e., *On-line Certificate Status Protocol* -OCSP- [PKIX98], or *Suicidal Bureaus* -SBs- [Rive98]) incorporate solutions to minimize the number of accesses needed to verify a certificate, but these solutions are too artificial and inefficient. Therefore, avoiding the use of CRLs was considered one of the fundamental goals in the design of Cert'eM. In order to achieve a “CRL-free” design, users must not distribute their certificates. On the contrary, the certificates are kept in the database of the corresponding CA to be directly handled.

There is a special user called *CA@<domain>* (*CA@lcc.uma.es* in the example) in every KSU that denotes the correspondent CA. The certificate of any CA is stored in the database of its parent KSU (*uma.es* in the example). Finally, the keys of the root CAs, the CAs located in the top-level domains, are certified by the domain registering authority (f.e. ICANN)

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

Some similarities can be found between Cert'eM and the Secure-DNS proposal. Both use the Internet domain name hierarchy to find the location where a particular key is stored, but the Secure-DNS uses the Name Server files while Cert'eM uses the e-mail offices. This choice is based on the following reasons:

– Frequently, DNSs are not closely related to users; usually several domains share the same DNS, oppositely e-mail offices are tightly coupled with the users;
– DNSs use caching and lifetime mechanisms that could yield inaccurate or false information in some situations;
– The CA of a DNS may not have a direct knowledge of the users’ identities and, therefore, it is more vulnerable to impersonation.

– The design of the Secure-DNS does not provide mechanisms to determine if a malicious CA changes the keys of the local users.

4.3 System Operation

We describe now the sequence of actions that are carried out when any user (solicitor) wants to get the public key of another user (addressee). In such case, the e-mail address of the last one is provided by the solicitor.

Keys distributed by a KSU are always certified by the corresponding CA so, in the subsequent discussion, we will use the terms key and certificate equivalently. When the key is requested, the solicitor's KSU conducts the request to the addressee's KSU, whose database contains the key. This is easily done because the system can determine, from the email address provided, which KSU has to be contacted.

Figure 4a shows the information flow produced when user *Bob* (*bob@r.s.t*) requests the key of user *Alice* (*alice@x.y.z*). As shown in the figure, Bob requests Alice's key from his own KSU (step 1) and this one directs the request to the KSU located at the *x.y.z* node (step 2). The response from the addressee's KSU (step 3) is then forwarded to Bob (step 4). Bob must request the key from his KSU due to the access restrictions that other KSUs set, and also to take advantage of the key proxy of his KSU. The addition of key proxies to the KSUs is discussed later.

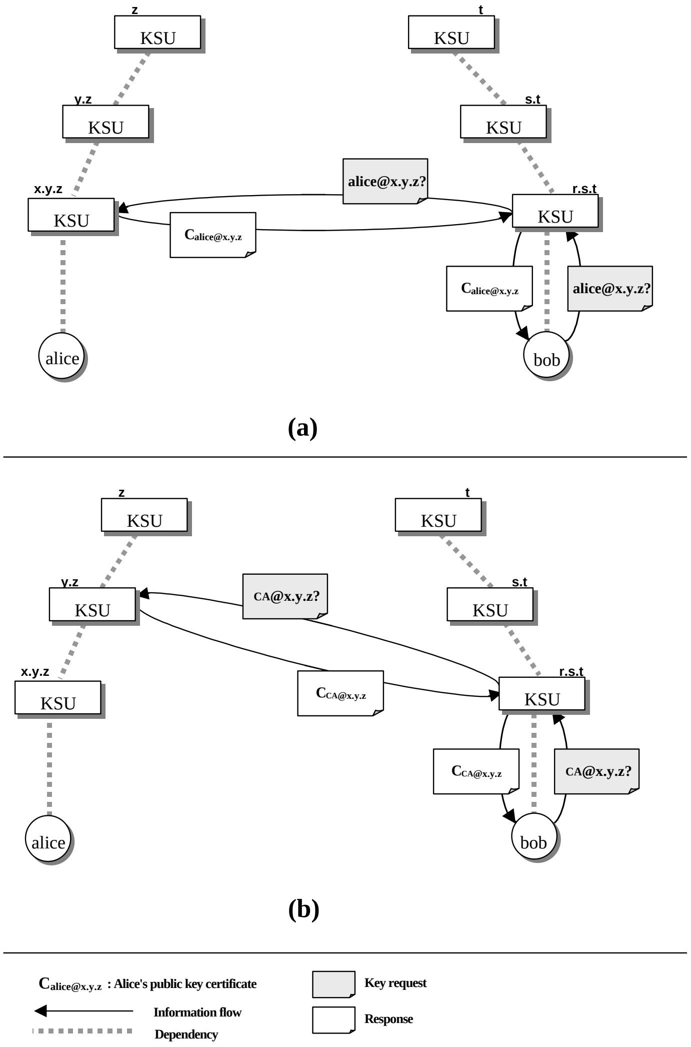

Figure 4. Key Request and Certificate Verification Information Flows

Afterwards, in case Bob needs to be more confident in the received key, he can request the key of *CA@x.y.z* from the KSU located at *y.z*, obtaining a new certificate (figure 4b). This guarantees Bob that Alice’s CA was not impersonated. The ascending verification process can continue until the 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*.

4.4 User identification

It must be pointed out that many of the identity certificates presently used are based exclusively in a contact, through Internet, between the user and the certifier (once again not conforming [EU98]). This is clearly unsatisfactory because the acceptor of a certificate will usually require some proof 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 misled from the start.

The design of Cert'eM guarantees that a CA will only certify the keys of those users close to it. So, a formal identity verification procedure can be established (in accordance with the applicable law) in order to give the certification process a legal meaning [Wrig98]. Therefore, when the CA signs Alice's key, it guarantees that Alice’s identity has been successfully verified (for example, requiring the real-world identity documents). 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. This link, however, does not need to be available online, so for most uses the anonymity is preserved.

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 univocal because a user can have several e-mail accounts and different aliases. Besides, there are certain e-mail addresses that don't represent a single user but a group of them[Detw93]. Secondly, it is also claimed that, in some cases, the alias file can be modified without *administrator* or *root* permissions.

Cert'eM is not exposed to any of these disadvantages because it makes no distinction between an e-mail account and an alias. Let us review the possible cases in detail.

- *Single User Alias*: *Bob* wants to create an alias (e.g. *robert@r.s.t*) for his e-mail address, *bob@r.s.t*. This alias will be linked to his name in the real world and to a cryptographic key. The key can be the same one that he is already using in his existing e-mail account or, alternatively, he can choose a new one (there is no difference to the system). Therefore, the relation *registered address* → *real world user* is univocal.
- *Group Alias.* If a group of users want to register an alias for their addresses, they have to follow the registration procedure and, naturally, the responsibility of sharing the private key resides on them. Hence, although the relation <registered address, real world user> is still multiple, there is a finite well-defined group of users related to that address.

Regarding the second criticism, although a malicious user could change, under some circumstances, the address that is represented by an alias, no problem arises unless the registered user of the alias gives up his private key to the malicious one. It is clear that this user will not be able to certify and insert the new key in the KSU database.

5. CONCLUSIONS

A new system for public-key management and user identification in Internet has been introduced. It has been shown that most of the existing designs are not completely satisfactory, and how Cert'eM can solve the problems addressed, with additional advantages as the real-time revocation of keys (without the need of CRLs) and the ease of key maintenance. In addition to this, the system is easily used, transparent to the user and compatible with most of other existing key services, allowing the addition of future ones.

The system was recently adopted by the National Research & Academic Network in Spain (RedIRIS), and is presently being tested to provide the public-key service to RedIRIS users. This is producing a valuable information for future improvements.

Among other interesting applications that use Cert’eM as the public key distribution infrastructure, a certified electronic mail service is being deployed for the whole community (more that 40000 users) in the University of Malaga.

REFERENCES

[Chok94] Chokhani S. *Toward a National Public Key Infrastructure*. IEEE Communications Magazine, 1994, pp. 70-74.

[Clar97] Clarke R. *Human Identification in Information Systems: Management Challenges and Public Policy Issues*. First Published in Information Technology & People 7,4. pp. 6-37.1994. August 1997 version available online at http://www.anu.edu.au/people/Roger.Clarke/DV/HumanID.html

[CSE98] Communications Security Establishment. *Government of Canada Public Key Infrastructure - White Paper*, 1998.

[Davi95] Davis, D. *Kerberos Plus RSA for World Wide Web Security*. First USENIX Workshop on Electronic Commerce, 1995, pp. 185-188.

[Detw93] Detweiler, L. *Identity, Privacy and Anonimity on the Internet*. Available online at http://www.intac.com//man/faq/net-privacy/part1

[DiHe76] Diffie, W.; Hellman, M. *New Directions in Cryptography*. IEEE Transactions on Information Theory. IT-22, n. 6. 1976, pp. 644-654.

[EC98] European Commission. *Proposal for a European Parliament and Council directive on a common framework for electronic signatures*. COM(1998) 297 final. 1998. Available online at http://www.ispo.cec.be/eif/policy/com98297.html

[FoBa97] Ford, W.; Baum, M. *Secure Elecronic Commerce*. Prentice-Hall, 1997.

[Gane95] Ganesan, R. *Yaksha: Augmenting Kerberos with Public Key Cryptography*. Internet Society Symposium on Network and Distributed Systems Security. IEEE Press, 1995, pp. 132-143.

[GaSp97] Garfinkel S.; Spafford G.*: Web Security and Commerce*. O'Reilly & Associates, Inc., 1997.

[Gerc98] Gerck, E.*: Towards Real-World Models of Trust: Reliance on Received Information*. MGC work document. Available online at http://www.mcg.org.br/trustdef.htm.

[Ilpf97] Ilpf Working Group on Certification Authority Practices. *The Role of Certification Authorities in Consumer Transactions*. Internet Law and Policy Forum, 1997.

[ISO88] ISO International Standard 9594. *Information Technology - Open Systems Interconnection Reference Model: The Directory*, 1988.

[ISO96] ISO/IEC JTC1/SC 21. *Draft Amendments DAM 4 to ISO/IEC 9594-2, DAM 2 to ISO/IEC 9594-6, DAM 1 to ISO/IEC 9594-7, and DAM 1 to ISO/IEC 9594-8 on Certificate Extensions*, 1996.

[Kent97] Kent, S. *How Many Certifications Authorities Are Enough?*. Public Forum on Certificate Authorities and Digital Signatures. U.S. Department of Commerce Technology Administration, 1997.

[Kohl89] Kohl, J. *The Use of Encryption in Kerberos for Network Authentication*. Advances in Cryptology - CRYPTO'89. LNCS 435. Springer, 1989, pp. 35-43.

[McCu96] McCurley, K. *Cryptography and the Internet: Lessons and Challenges*. Advances in Cryptology - ASIACRYPT'96. LNCS 1163. Springer, 1996, pp. 50-56.

[NIST96] National Institute of Standards and Technology. *A Public Key Infrastructure for U.S. Government Unclassified but Sensitive Applications*, 1996.

[OGIT98] Office of Government Information Technology. *Gatekeeper: A Strategy for Public-Key Technology Use in the Government*, 1998.

[PKIX97] PKIX Working Group Internet Draft. *Internet Public Key Infrastructure Certificate Policy and Certification Practices Framework*, 1997.

[PKIX98] PKIX Working Group Internet Draft. *X.509 Internet Public Key Infrastructure. Online Certificate Status Protocol – OCSP*, 1998.

[POLM97] Pino, L.; Ortega, J.; Lopez, J.; Maña, A. *A System For Public Keys Service In the Spanish Research & Academic Network*. VII Annual Conference of the Internet Society, 1997.

[RFC1101] Mockapetris, P.V. *DNS Encoding of Network Names and Other Types*, 1989.

[RFC1421] Linn, J., *Privacy Enhancement for Internet Electronic Mail: Part I: Message Encryption and Authentication Procedures*, 1993.

[RFC1422] Kent, S. *Privacy Enhancement for Internet Electronic Mail. Part II: Certificate-Based Key Management*, 1993.

[RFC1423] Balenson, D., *Privacy Enhancement for Internet Electronic Mail: Part III: Algorithms, Modes, and Identifiers*, 1993.

[RFC1424] Kaliski, B. *Privacy Enhancement for Internet Electronic Mail. Part IV: Key Certification and Related Services*, 1993.

[RFC 1510] Kohl, J.; Neuman, B.C. *The Kerberos Network Authentication Service (V5)*. 1993.

[RFC2065] Eastlake, D.; Kaufman, C. *Domain Name System Security Extensions*, 1997.

[RFC2316] Bellovin, S. *Report of the IAB Security Architecture Workshop*, 1998.

[RiLa96] Rivest, R.; Lampson, B. *SDSI – A Simple Distributed Security Infrastructure*, 1996.

[Rive98] Rivest, R. *Can we Eliminate Revocation Lists?*. Financial Cryptography 1998.

[ScAt95] Schiller, J. I.; Atkins, D. *Scaling the Web of Trust: Combining Kerberos and PGP to Provide Large Scale Authentication*. USENIX Technical Conference, 1995.

[SPKI98a] SPKI Working Group Internet Draft. *Simple Public Key Certificate*, 1998.

[SPKI98b] SPKI Working Group Internet Draft. *SPKI Certificate Theory*, 1998.

[Tsuj96] Tsujii S. *Electronic Money and Key Management from Global and Regional Points of View*. Advances in Cryptology - ASIACRYPT'96. LNCS 1163. Springer, 1996, pp. 173-184.

[Wrig98] Wright, B. *Making Numbers Ceremonial: Signing Tax Returns with Personal Identification Numbers*. Personal Communication, 1998.

[Zimm95] Zimmerman, P. *The Official PGP User’s Guide*. MIT Press, 1995.