J. Lopez, A. Mana, J. J. Ortega, and J. M. Troya, "Distributed Storage and Revocation in Digital Certificate Databases", *11th International Conference on Database and Expert Systems Applications (DEXA00)*, LNCS vol. 1873, pp. 929-938, 2000.
NICS Lab. Publications: https://www.nics.uma.es/publications

# DISTRIBUTED STORAGE AND REVOCATION OF DIGITAL CERTIFICATE DATABASES

Javier Lopez, Antonio Mana, Juan J. Ortega, Jose M. Troya
E. T. S. Ingenieria Informatica - University of Malaga
Campus de Teatinos, 29071 – Malaga (SPAIN)
{jlm, amg, juanjose, troya} @lcc.uma.es

# DISTRIBUTED STORAGE AND REVOCATION OF DIGITAL CERTIFICATE DATABASES

**Abstract:** Public-key cryptography is fast becoming the foundation for those applications that require security and authentication in open networks. But the widespread use of a global public-key cryptosystem requires that public-key certificates are always available and up-to-date. Problems associated to digital certificates management, like storage, retrieval, maintenance, and, specially, revocation, require special procedures that ensure reliable features because of the critical significance of inaccuracies. Most of the existing systems use a *Certificate Revocation List* (CRL), a database of certificates that have been revoked before their expiration date. The need to access CRLs in order to check certificate revocations becomes a performance handicap. Furthermore, they introduce a source of vulnerability in the whole security infrastructure, as it is impossible to produce a new CRL each time a revocation takes place. This paper introduces an alternative for the storage of digital certificates that avoids the use of CRLs. The system is designed to provide an distributed management of digital certificates by using Certification Authorities (CAs) that, while being part of a whole Public-Key Infrastructure (PKI), operate over local certificates databases. Communication protocols between local databases have been designed to minimize network traffic without a lack of security and efficiency.

## 1 INTRODUCTION

With the rapid diffusion of the Internet in recent years transactions via networks are becoming increasingly common. In order to promote network transactions further, it is essential to ensure electronic authentication. Electronic authentication plays a very important role in assuring the reliability of network transactions and, indeed, the network itself. If electronic authentication is provided in an inappropriate manner, reliability is brought into question.

Therefore, a mechanism must be put in place for confirming the authentication of network users and the content of communications. Public key cryptography [DiHe76] makes such mechanism, the digital signature, possible. By using digital signatures we can obtain authentication in the form of digital certificates.

A Certification Authority (CA) is a trusted entity that issues certificates. The most basic certificate (identity certificate) binds the user's name (or identifier) to the corresponding public key, to which the CA's digital signature is affixed.

Because the number of users in a system increase, the number of CAs increases too, conforming a Public Key Infrastructure (PKI). A PKI is the relying framework that

allows a wide deployment of public-key technology as it provides essential reliability for electronic communication between users that can not have a face-to-face relationship. Thus, by using a PKI, public key certificates management becomes possible, and a secure network environment can be established, enabling the use of security services (confidentiality, access control, integrity, authentication, and non-repudiation) for electronic transactions and for their supporting information technology applications.

A CA issues a certificate for a subject user such as an individual or a corporation, a affirming something about some principal [Ilpf97]. But, usually, the validity of this statement is limited in time; a certificate from my university that states that I teach "Computer Security" dated on 1993 will probably useless to access the records of this year's students. As the information that the certificate holds is subject to change, it is valid for a stated period of time. To avoid this type of problems certificates usually include a validity interval. This does not solve the problems completely because there are circumstances when a certificate needs to be invalidated before the expected validity interval expires. For example, if the subject user loses the private key corresponding to the public key given on the certificate, or has it stolen or compromised, or if there is a possibility of this having occurred, the certificate has to be invalidated. The CA must publish the fact of revocation in a way that is accessible to the user and other parties involved through publicly accessible networks, such as the Internet.

Therefore, we need mechanisms to invalidate (revoke) the certificate before it expires[1]. Typically, CA makes the revocation information known by using a Certification Revocation List (CRL).

Consequently, CRLs are basically repositories that identify certificates that have been withdrawn, canceled, compromised, or should not be trusted for other specified reasons. Because a CA cannot force the destruction of all copies of a certificate, anyone who plans to rely on it must check it against a current CRL to ensure its validity. As a result, a CA must maintain continuity and promptness in the provision of revocation services, so that the public will not be misled by revoked certificates.

But all this process is not trivial; oppositely, it is complex. Checking the validity of a certificate is not straightforward as the user must open a network connection to the issuing authority, find the CRL, and submit the certificate for checking. That is the reason why the issue of CRLs and the certificate revocation management are becoming an increasing focus of attention.

In this paper we introduce an alternative solution to the use of CRLs because we consider that they are not efficient for most applications. In section 2 we analyze the problem of using CRLs as mechanisms for revocation. In section 3 a new approach for revocation is introduced. This method is connected to the operation of Cert'eM, a hierarchical certification system based on electronic mail addresses that has been developed in our University. Section 4 shows two applications that use this system, and, finally, section 5 presents conclusions and future work.

[1] A certificate has *expired* if its validity period has finished. Conventionally, a certificate that has not expired has been called *valid*. The previous discussion leads us to consider that this is not accurate and, consequently, we prefer to use the term *active* certificate to refer to a certificate that has not expired. An active certificate is *valid* if it has not been revoked.

## 2 CERTIFICATE REVOCATION LISTS

Most of PKI models use CRLs as the mechanism for certificate revocation. This method is defined by Recommendation X.509 [ISO88], which is the recognized standard for public key certificate formats. In this Recommendation, a CRL is defined as a time-stamped list identifying revoked certificates which is signed by a CA and made freely available. Each revoked certificate is identified in a CRL by its certificate serial number, a unique number for the certificate which is generated by the issuing CA and included in a certificate field.

The pull method of CRL distribution is the most common method that users employ to check the lists of revoked certificates. The CRL is not automatically distributed; on the contrary, users access to the list, that is periodically published by the CA. But one limitation of this method is that the time granularity of revocation is limited to the CRL issue period [FoBa97]. As next figure shows, significant time intervals can take place since a user manifest the intention to revoke the certificate until the CA is informed of the fact, from that moment until the new list is issued, and from that moment until the CRL modification reach final users (figure 1). During those periods, the risk of integrity in the system grows exponentially, and, certainly, it is not clear who holds responsibility in each of them.

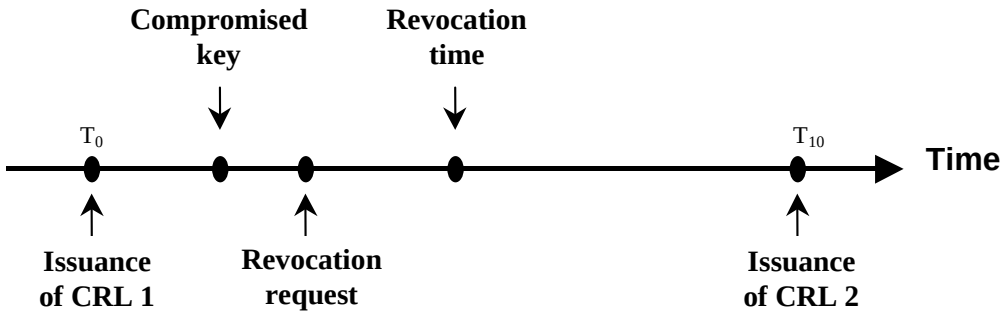


**Figure 1. Timeline of a certificate revocation process**

Another limitation is the size that the CRL can reach. If the database becomes very large, performance problems appear in terms of both communication overheads and processing overheads in certificate-using end entities. There are two elements whose simultaneous growth can make a CRL difficult to be managed. The first one is the rate at which revocations occurs, which is quite unpredictable, but that is clearly dependent on the size of the population of users covered. The second one is the certificates validity period, because once the CA introduces a certificate into the CRL, it is not deleted until it reaches its the expiration date.

The first element – number of users – is not easily controlled, but the second one seems to represent a possible solution as it depends on the certification policy established inside the PKI. Thus, decreasing the certificate validity period during the certificate creation process, the CRL becomes smaller (very small validity periods could even eliminate the need to issue CRLs). But, in this case, what is positive for CRL size is negative for the general performance of the system. The reason is that the use of small validity periods implies that certificates must be issued more frequently. Therefore, this is a costly and ineffective solution and can be used only in some very specific cases.

As modification of the validity period turns problematic, the solution for the uncontrolled growth of the CRL must be targeted to modify the number of users. But, as this number cannot be decreased, the solution is to distribute users, that is, to partition the revocation repository into CRL distribution points, with each point containing a disjoint group of revoked certificates.

In version 3 of X.509 [ITU97] the concept of distribution point is extended from previous versions, allowing those points to be established depending, not only on the subject type (final user or CA), but on the reason of revocation. Furthermore,

certification policy allows each list to be issued at different times. The X.509 v.3 also introduces the concept of incremental CRL, or Δ-CRL. According to this concept, a CA does not need to issue new periodic versions of a CRL; it just needs to issue the modifications (Δ) from the last version.

Our consideration is that all these possibilities concerning the pull method represents a collection of too complicated solutions, and shows that this method is far from the solution that most of real PKI applications demand.

There is a less used method for CRL distribution, the push method. In this method, the CA sends the revocation lists to the users periodically, and does not introduce it into a repository as in the pull method. Such broadcasts are accomplished via protected communication means such as secure e-mail or a protected transaction protocol. The major advantage of this approach is that important revocations can be distributed very quickly, without the time granularity delay problem inherent to the periodic revocation list approach.

However, there are two potential problems with this approach. First is the requirement for a protected distribution method to ensure that CRLs reach their intended destinations. The protection of the distribution method represents an overload for the system. Second is the massive amount of traffic generated in order to notify all revocations. This problem could be solved if the broadcast is restricted; that is, if it is possible to establish, at the beginning, which revocations are broadcasted and who are the intended recipients. But this scheduling becomes impossible inside a large PKI.

Therefore, neither pull nor push methods are good options to solve generation, management and distribution of CRLs inside a PKI. We consider that the concept of

CRL itself represents a drawback, and that the best solution is that one in which the knowledge of a revoked certificate is immediately available to users without a lack of performance in the system. This idea has been followed in the design of our certification system, Cert'eM, when certificate revocation problem has been faced. We have considered this problem as priority in the design process, and has had a big influence in the rest of the PKI components.

# 3 DISTRIBUTED STORAGE AND REVOCATION

## 3.1 First Consideration: Certifying On-line

If a certificate is not included in a CRL then all the user knows is that the certificate was not revoked when the CRL was issued, but in most cases this is not enough. The reason for this is that CRL-based systems provide negative proofs (proofs of the negative validity but they give no evidence of the positive validity). What a user usually needs is a positive proof of the validity of the certificate, or even better a proof of the status of the certificate; we call this proof a *validity statement* (VS).

As a consequence we affirm that, for uses other than historical (i.e. knowing that the certificated was once issued), the requirement of previous possession of the certificate does not represent any advantage in order to obtain confidence in the information contained in that certificate, on the contrary it introduces serious obstacles to the certificate management and use procedures. We believe that an online certificate server can solve the certificate validity problem more efficiently than CRL-based systems (after all, the CRL retrieval requires an online request).

## 3.2 Second Consideration: Distributing Contents

Most of the systems that deal with the storage of digital certificates are based in centralized schemes. Some of them replicate the contents in different servers to distribute the requests among them, thus introducing synchronization problems.

Other systems do not provide storage for the certificates (this is done by the user) but do provide storage for the certificate revocations (using CRLs). These schemes, like those based in the X.509v3 standard, do not conform a distributed database (in this case, of certificate revocations) but separate, unrelated revocation lists. The revocation point is established by the CA at the time of issuance of the certificate and there is no standard way to balance the load between servers, or distributing the CRL among servers. Besides, the process of obtaining the appropriate CRL and all the subsequent Δ-CRLs just to know if an active certificate is still valid is quite complicated. As these CRLs contain the revocation of the certificates of many users, most of the information received in this case is completely useless for the requestor.

These solutions are very inefficient. The most efficient approach is to distribute the contents of the database of certificates among a series of servers according to some established distribution criteria. A good distribution scheme should fulfill the following properties:

- An algorithm exists that applied to the known (key) data of the certificate (not the complete certificate), unambiguously identifies the server that contains it, and
- The scheme must distribute the certificates in a balanced manner (i.e. the number of certificates stored in each server must be proportional to the capacity of the server).

## 3.3 Third Consideration: Distinguishing Names

One of the basic fields in almost all digital certificates is the *distinguished name* (DN). The DN is a unique identifier of the certificate. Sometimes, this DN is globally unique. For the purpose of using the DN in a distributed certificate storage scheme a globally unique DN is optimal for property one. A DN that is related to the logical location of the certificate is also desirable because it would help to fulfill the second property.

There are two possible schemes that are based in the use of DN. The first scheme is based in the *Domain Name System* (DNS) structure [RFC1101]. The recent DNS security extensions establish another proposal that allows authentication through digital signatures. Its name is Secure-DNS [RFC2065] [RFC2137]. These extensions describe a hierarchic PKI, integrated into the DNS database by adding a set of registers called *RR* registers. The public key of the CA of a zone is recorded in the *SIG RR* register and the public keys of the users of this domain are recorded in *KEY RR* registers, certified by the corresponding CA. But 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 (therefore not conforming with article 8.2 in [EC98]).

The second scheme is to use the e-mail service structure as the base for the distribution of the certificates in the different servers. This was the choice we took for the design of Cert'eM [LMOT99]. Cert'eM is a multi-hierarchical scheme that is based in the use of e-mail addresses as distinguished names and in the location of certificate distribution points in each e-mail office. the main element in the hierarchy is the *Keys Service Unit* (KSU), which integrates certification and management functions. Cert'eM

proposes a scheme with various KSUs operating over disjoint groups of users, conforming a predefined hierarchy.

Figure 2 shows the system's structure. The KSU hierarchy defined by Cert'eM is parallel to the hierarchy of Internet domains. The KSUs are associated to the corresponding e-mail offices.

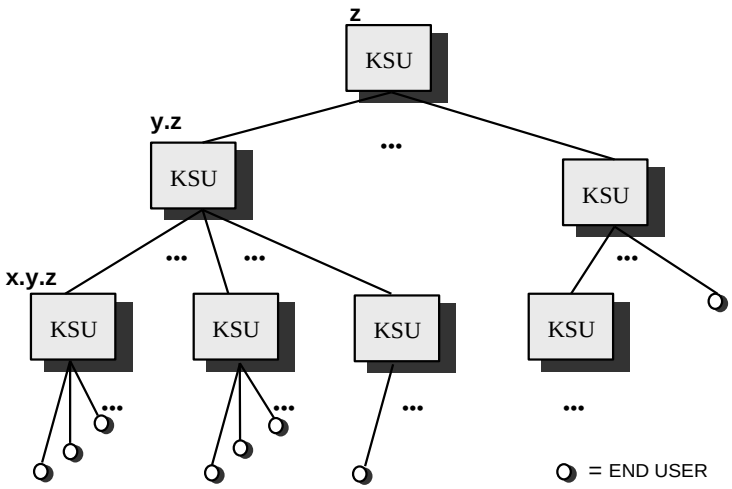


**Figure 2. Hierarchy of Cert'eM Nodes**

Every KSU is managed by a CA (Figure 3). Additionally, it contains a portion of the certificate 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/her KSU. The third component is the key server, which receives requests and delivers the certificates. The key server manages a certificate proxy that keeps some of the recently received external certificates. The certified keys are managed solely by the corresponding CA; therefore, key updating and revocation are local operations that do not affect the rest of the system.

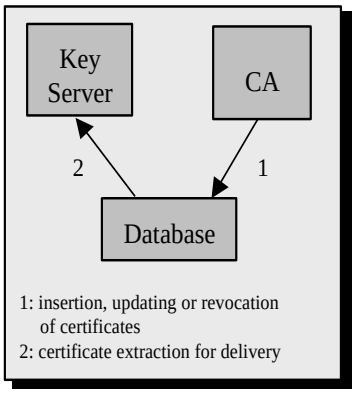


**Figure 3. KSU Components**

We want to emphasize that no CRL is used in the system. Instead the validation of certificates is achieved using the *Validity Statement* (VS), a timestamp statement signed by the CA attesting the status of the certificate at the time of issuance of the VS; therefore in order to validate active certificates the CA simply issues a VS.

To achieve a design that does not expose the mentioned problems of the use of CRL while still retaining their benefits, we impose the following restrictions:

- All the information related to the certification of a specific user must be located and managed at the corresponding KSU. Therefore, in case a CA decides to record certificate invalidation events, a *Local Invalidation Log* (LIL) can be managed locally. Notice that the LIL is completely different to CRLs. The LIL will be used exclusively by the CA.

- Users must not distribute their certificates. On the contrary, the certificates must be kept in the database of the corresponding CA and distributed by their KSU.

When a user's 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.

When 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 in old certificate reuse.

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. 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, 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 are not useful any more and the CA must reissue all the certificates. Other systems need to notify users and request old certificates in order to re-certify their keys and distribute these new certificates. In our proposal there is no need to send new certificates and invalidate the previous ones, because all the certificates of the users of a KSU are kept in a local database. Thus, the change of the CA key (and hence, of the certificates issued by that CA) is transparent to users.

# 4 EXAMPLE APPLICATIONS

Two different applications have been developed to test the system. The first application allows the secure communication of sensible information between the secretary of a university and the teachers. For example, it is used by the teachers to receive, fill, and send back the official evaluation records of the students, in the courses they teach. This process has many similarities with the sale of a document. Teachers picks documents between those that are automatically selected for them according to their corresponding certificate. The document container (called sales agent) enforces some steps to assist the teacher in filling the evaluations and to guarantee that the teacher information is error free.

The second application is designed to make exams using Internet [25]. This application is used to test the generation of client-specific applets and the use of certificates in a bigger community of users (It actually provides certification and key distribution services to more than 40,000 users in our university).

Both applications are based in the dynamic creation of specialized Java applets (called *sales agents*) that are responsible for the secure transport of the protected contents.

Figure 4 depicts a scheme of the dynamic applet creation process. The first step is the negotiation that is used to determine the terms and conditions (i.e. the contract) of the sale. This contract is then processed by the applet generator to produce a specific sales agent for that contract and user. This process needs access to trustworthy user identities and keys and is supported by the use of certificates that are managed, stored, distributed and revoked by Cert'eM servers.

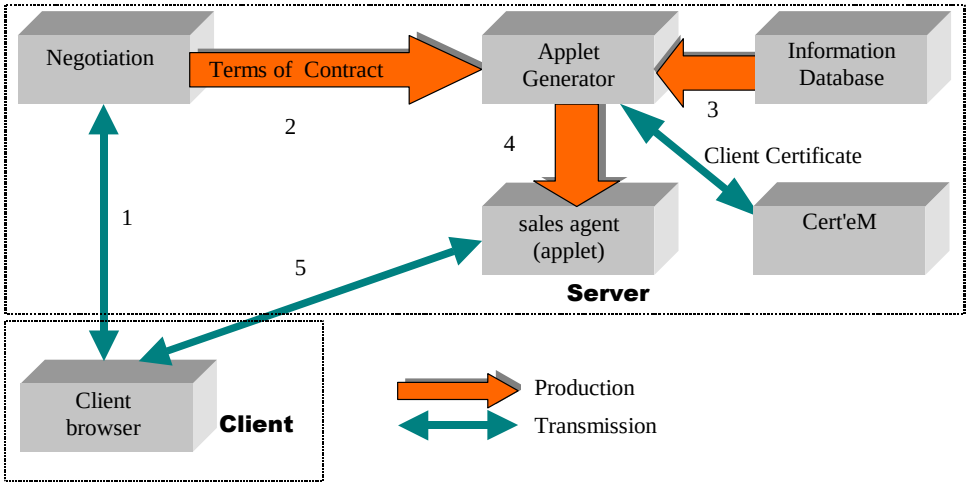


**Figure 4. Overview of the applet creation process**

## 5 CONCLUSIONS

Problems associated to digital certificates management, like storage, retrieval, maintenance, and, specially, revocation, require special procedures that ensure reliable features because of the critical significance of inaccuracies. Most of the existing systems use a Certificate Revocation List (CRL), a database of certificates that have been revoked before their expiration date.

In this paper we have introduced an alternative solution to the use of CRLs because we consider that they are not efficient for most applications. Moreover, we think that the concept of CRL itself represents a drawback, and that the best solution is that one in which the knowledge of a revoked certificate is immediately available to users without a lack of performance in the system.

This idea has been followed in the design of our certification system, Cert'eM, when certificate revocation problem has been faced. We have considered this problem as priority in the design process, and has had a big influence in the rest of the PKI components.

# REFERENCES


[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

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

[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.

[FoBa97] W. Ford, M. Baum, Secure Electronic Commerce, Prentice-Hall, 1997

[ITU97] International Telecommunication Union, *Itu-t recommendation x.509. Information technology – Open Systems Interconnection – The Directory: Authentication framework*, 1997.

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

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

[RFC2137] Eastlake, D. *Secure Domain Name System Dynamic Update*, 1997.

[LMOT99] Lopez, J.; Maña A.; Ortega, J. J.; Troya J. M. *Cert'eM: Certification System Based on Electronic Mail Service Structure.* Proceedings of CQRE'99. Springer-Verlag. LNCS 1740 (1999)

[MVL98] Maña, A.; Villalba, F.; López, J. *Secure Examinations Through The Internet*. Proceedings of Teleteaching'98, IFIP World Computer Congress, 1998.