Efficient Certificate Status Handling within PKIs: an Application to Public Administration Services

Marco Prandini
DEIS – Department of Electronics, Computer and System Science
University of Bologna – Viale Risorgimento 2, 40136 Bologna, Italy
E-mail: mprandini@deis.unibo.it

Abstract

Public administrations show a strong interest in digital signature technology as a mean for secure and authenticated document exchange, hoping it will help reducing paper-based transactions with citizens. The main problem posed by this technology is with the necessary public-key infrastructure, and in particular with certificate status handling. This paper describes the definition and deployment of a web-based environment suitable for offering administrative services to citizens and for accepting authenticated documents from citizens. The best features of two different certificate status handling schemes, namely CRL and OCSP, have been exploited within this environment to obtain a good balance between security, timeliness and efficiency.

1. Introduction

As the free exchange of people and goods within Europe becomes effective, the efficiency required to public administrations must increase in order to meet the expectations of the new and enlarged market.

The adoption of electronic data processing (EDP) has already speeded up most of the internal procedures of public administrations, but the potential improvement of EDP can't be fully exploited until document hardcopies are needed at different moments during the process. Of course, this isn't a technological constraint, but a legal issue, mainly related to signature and archival operations.

During the last few years, powerful tools like digital signatures have become widely available. Their usage is currently being put under legal regulation, with the aim of giving digital signatures the same value of autographed ones, and of substituting digital archives for hardcopy.

Various Italian public administrations the author has worked with (Regione Emilia-Romagna, Comune di Modena, Comune di Cesena) are deploying experimental infrastructures to support the exchange of digitally signed documents. This paper discusses some technological issues arisen during these experiences.

2. Technology overview

A general definition of a digital signature can be stated as follows: a digital signature is a bit string computed by the signer over a digitally formatted document, which can both prove the integrity of the document and authenticate the signer.

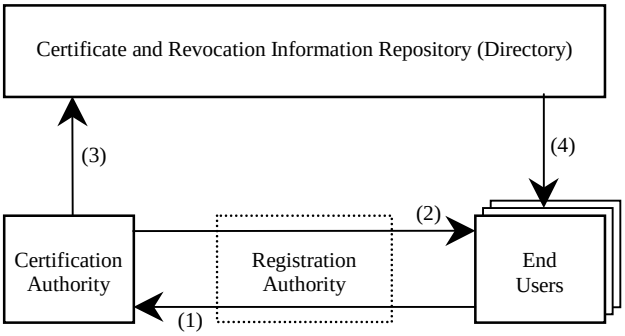
While these properties can be achieved in different ways, the most viable option these days seems to be *asymmetric cryptography* [1,2]. Within an asymmetric cryptosystem, each signer owns a pair of numbers, called *private key* and *public key*, used for, respectively, signing and verifying signatures. The first key must be kept strictly secret by its owner, while the second one should be made widely available.

Since legal and economic transactions occur between individuals or organizations, and not between "keys", the fundamental task of binding each public key to its owner is performed by Public-Key Infrastructures (PKIs). The entities of a PKI and their mutual relationships, described in detail in [3], are shown in Fig. 1.

When a user presents a request to a Registration Authority (RA) to become a PKI member, a certificate is issued containing the user identity (ascertained in a secure way) and the associated public key, together with some other relevant information such as the certificate issue and expiration dates. The certificate is signed by the Certification Authority (CA), so that its integrity is guaranteed, and published to the directory where it is made available to all PKI users. The CA's public key, which is required to verify any CA's signed document, is transferred to the new user via a secure channel.

It may be necessary to revoke a certificate before expiration of its validity period, commonly one year, due to events like private key compromise or changes in user identification data (e.g., affiliation). As a consequence,

proper use of a certificate implies not only expiration date verification, but also revocation checking.



(1) Users registration, certificates revocation and update requests
(2) Certification Authority's public key distribution
(3) Certificates and associated status information publication
(4) Certificates and associated status information distribution

Fig. 1 – Architectural model of a PKI.

The implementation of a certificate status handling scheme must be devised according to security, timeliness and efficiency requirements. A more precise description of these requirements within this context can be helpful:

Security: certificate status information, as known to the CA, should be made available to users without the possibility of alteration.

Timeliness: there should be the minimum possible delay between any certificate status change (e. g., certificate revocation) and its reflection on the replies to user queries.

Efficiency: the algorithms and protocols involved in status updating shouldn't be limiting factors for the secure and timely behavior of the infrastructure.

In many applications, the security requirement is considered of primary concern. To this end, the only secure method currently recognized in the Internet Society standard track, namely the *Certificate Revocation List* (CRL) scheme [3], is exploited. Under this scheme a CA periodically timestamps, signs and sends to the directory a list of (the serial numbers of) all revoked certificates, together with the corresponding date and time of revocation. This very simple solution, which prevents any hostile intervention by the directory (except, of course, denial of service), has the advantage of requiring a single cryptographic operation both by the CA at each update and by the user at each check. It exhibits, however, a serious drawback, since the status of a certificate is verifiable by a user only by obtaining the complete, comprehensive list of all revoked certificates. Moreover, since the list authentication is performed off-line by the CA, timeliness is somewhat limited.

In other applications, the timeliness requirement is considered of primary concern. To this end, the *On-line Certificate Status Protocol* (OCSP) [4], derived from the original proposal of the *Real-Time Certificate Status Protocol* [5], is exploited. Within an OCSP-based system certificate status authentication is delegated to a *responder*, usually integrated in the directory, i.e. it isn't directly guaranteed by the CA signature, but by a signature produced with a key that the CA and/or users trust. The drawback of this solution is that the responder, can be attacked through the network or perform malicious alteration of information sent to users, without CA and users can promptly realize it. Proper behavior of an OCSP-based system is therefore possible if and only if the directory is trusted.

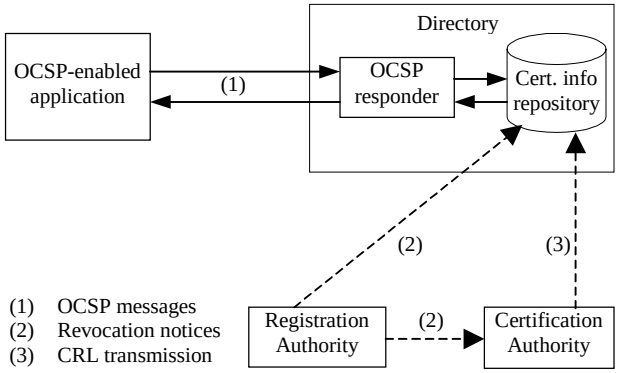


Fig. 2 – Architectural model of a CRL/OCSP-based system

The idea behind the work reported in this paper is that the combined use of OCSP and CRL (Fig. 2) may provide a better overall performance. OCSP, in fact, makes available explicit, concise and timely updated information regarding each single certificate. CRL may conveniently add periodic guarantee of authentication and unforgeability.

3. Efficiency of the CRL/OCSP architectural model

The efficiency of a PKI relying on both CRL and OCSP for certificate status handling can be conveniently estimated before effective deployment, in order to evaluate the impact of PKI operations over the existing information system. In this section, both communications traffic between PKI entities and computational load over each entity are analytically expressed as functions of the relevant PKI parameters. Particular emphasis is placed on the computational load deriving from application of each scheme, a matter which only recently has received the attention it deserves [6]. The parameters usually

considered of primary concern in similar contexts are related to:

i) certificate status generation and publication:
   - number of cryptographic operations performed by the CA or by the directory to update the status of all certificates;
   - amount of data transferred from the CA to the directory at each update;
ii) certificate status checking:
   - amount of data transferred from the directory to a user;
   - number of cryptographic operations performed by a user.

These parameters have been computed under the usual assumption that a PKI works in stationary conditions as regards the number of revoked certificates, that is, within any certificate status updating period, the average number R of new revocations is balanced by as many revocation removals. R is simply given by:

$$R = \frac{N \cdot P}{365T}$$

where, according to NIST notation [7], N indicates the total number of certificates handled within a PKI, P the revoked certificates fraction, and T the daily number of certificate status updates.

3.1. Computational load and traffic deriving from the CRL protocol

(a) CA computational load evaluation

CRL is essentially (ignoring the header specifying the CA identity, the timestamp, and the adopted signing algorithm) a signed list of $N \cdot P$ (revoked certificate serial number, revocation date and time) pairs. Each pair, according to NIST estimates, can be represented with 68 bits: 20 for the serial number and 48 for the revocation date and time. A list therefore consists of $l_{info} = 68N \cdot P$ bit. At each update, the list must be signed with the usual process: a message digest function is first applied, then the resulting hash is signed with the CA's private key. The daily computational load of the CA is therefore given by:

$$L_{CA} = T\,(l_{info} \cdot L_{hash} + L_{signature})$$

With the aim of expressing the overall load of an operation which involves very different cryptographic algorithms, two symbols have been introduced: $L_{signature}$, which represents the computation time needed to perform a signature operation, and L_{hash}, which represents the equivalent marginal load of a 1-bit message digest, computed as the inverse of the function's bit rate [8].

(b) Directory incoming traffic evaluation

At each update the list is published on the directory. The deriving directory incoming traffic, in bit/day, is then given by:

$$T_{CA-DIR} = T(l_{info} + l_{signature})$$

where $l_{signature}$ indicates the number of bits involved in the representation of the signature.

(c) Directory computational load evaluation

In a CRL-based system, no cryptographic operations are requested to the directory in order to reply to a user query.

(d) Directory outgoing traffic evaluation

Each time a user needs to check a certificate status, the whole list has to be sent. The overall directory outgoing traffic, in bit/day, is then given by:

$$T_{DIR-U} = Q(l_{info} + l_{signature})$$

where Q is the daily number of user queries.

(e) User computational load

To perform a certificate status check, a user needs to search the CRL for the corresponding serial number and verify the signature on the list. It is reasonable to assume the latter contribution as the most relevant, so that the computational load can be estimated as:

$$L_U = l_{info} \cdot L_{hash} + L_{verification}$$

where $L_{verification}$ represents the computational load of a signature check operation.

3.2. Computational load and traffic deriving from the OCSP protocol

(a) CA computational load evaluation

OCSP does not rely on synchronous certificate status updates by the CA, that is, it doesn't involve any CA cryptographic operation or any communication between CA and directory.

(b) Directory incoming traffic evaluation

Each time a certificate status change is reported to the RA, immediate action is taken to directly inform the directory. This originates a negligible daily directory incoming traffic, given by:

$$T_{CA-DIR} = \frac{20N \cdot P}{365}$$

(c) Directory computational load evaluation

When a user issues a query about the status of one or more certificates, the OCSP responder performs a repository search to extract the selected up-to-date information. The signed reply contains a data section reporting, for each queried certificate, the target certificate identifier, the certificate status value, the response validity interval and, possibly, optional extensions. The reply contains also a header specifying the responder identity and version, and the adopted signing algorithm. The header size can no more be ignored with respect to the data section size as in the CRL case, since the status representation of a certificate calls for very few bits (possibly two bits only, discriminating whether the certificate is valid, revoked or unknown), and a reply most frequently deals with a single certificate. The overall size of the reply is therefore a fixed quantity, which can be indicated with $l_{OCSP\ reply}$. The resulting daily computational load can be estimated as:

$$L_{DIR} = Q(l_{OCSP\ reply} \cdot L_{hash} + L_{signature})$$

(d) Directory outgoing traffic evaluation

With the OCSP protocol, a user receives data about the certificate he wants to check. The deriving outgoing traffic for the directory, in bit/day, is:

$$T_{DIR-U} = Q(l_{OCSP\ reply} + l_{signature})$$

(e) User computational load evaluation

To perform a certificate status check, a user needs to verify the signature on the reply. The corresponding computational load is:

$$L_U = l_{OCSP\ reply} \cdot L_{hash} + L_{verification}$$

4. Applications and experimental results

The architectural model illustrated in this paper has been applied for the design of the PKI which will handle in the near future the public administrative services for the town of Modena (Italy). Following open standards, ensuring a high degree of interoperability, gathering as much awareness as possible during the development process, all have been primary design concerns dictated by the public administration. The implemented system is therefore based on well-established underlying architectures and protocols, on open-source software libraries, and on PKI components developed by the Italian participants to the ICE-TEL project [9]. Both the CRL and OCSP schemes have entered the testing phase to experimentally validate the foreseen performances.

It is important to notice that the main objective of this first deployment phase is not to build an all-purpose PKI for the citizens, but to evaluate the real benefits of substituting the traditional paper exchange with authenticated digital documents exchange between the citizens and the city administration offices. To this end, a test-bed has been created for submission of signed forms and certificate-based access control to services.

Working within this specific application field, it is possible to introduce a variant of the illustrated CRL/OCSP architecture. Since the public administration is at the same time both the entity in charge for certificate status updating and the only user which really needs high timeliness in status handling, the OCSP responder access could be restricted to the administration intranet, so gaining two important advantages:

- the responder can be placed in a secure network, where malicious attacks are much less probable;
- being the responder highly trusted, once a reply has been computed it can be cached for a longer time than it would normally be allowed to, easing the computational load associated with reply signing.

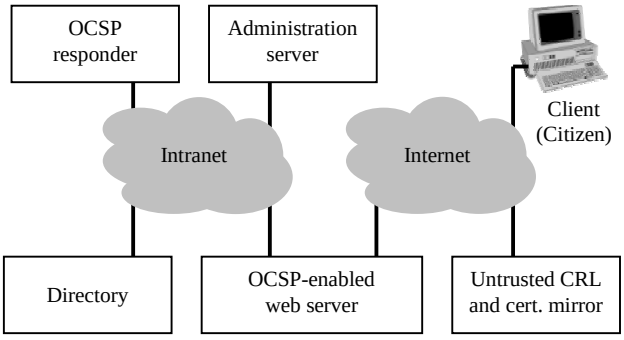


Fig. 3 – Architectural model of the implemented CRL/OCSP-based system

A possible architectural solution is represented in Fig. 3. The CA-authenticated information, such as certificates and CRLs, is both placed on the Intranet directory and mirrored on an Internet-accessible, untrusted directory,

thus allowing public, secure usage of certificates for transactions which don't need the timeliness provided by OCSP. The revocation notices, as soon as they are received by some PKI front-end (like RAs), are sent to both the CA for insertion in the CRL and the Intranet directory for updating the OCSP database.

The public administration offers its services by way of a secure web gateway, which performs client authentication and validates the client certificate status using both the CRL and an integrated OCSP client. A good timeliness/security tradeoff is obtained by tuning the frequency of CRL issuing. The gateway, in fact, caches a copy of the CRL each time it is updated, in order to get a secure, long-term reference list of revoked certificates. Each certificate that doesn't appear in the CRL is verified by means of the OCSP protocol, allowing real-time information to be retrieved.

A key step in the implementation of this gateway has been an ad-hoc extension to the well-known *mod_ssl* module [10] for the *Apache* web server [11]. It is worthwhile pointing out that this extension is candidate for definitive inclusion within *mod_ssl* in the near future. Of course, the resultant OCSP-enabled web server can be used in conjunction with every OCSP responder, being its application absolutely not restricted to the proposed secure-network-based architecture.

The prototypal system has undergone a testing phase under various realistic usage conditions, in order to evaluate the system behavior before full deployment. The same parameters analytically expressed in section 3 have been measured for different simulated values of the number of certificates handled in the PKI, of the update frequency, etc.. The results, as expected, were in good accordance with the analytically estimated figures.

Fig. 4 reports in graphical form two of the most interesting parameters, directory incoming and outgoing traffic, showing their dependence on the number of certificates handled in the PKI. It is evident how a big saving in the directory outgoing traffic can be achieved by exploiting OCSP instead of CRL for the greater part of the status checks. The directory incoming traffic is the sum of the CRL- and OCSP-induced traffic, which is only negligibly higher than the CRL-induced traffic alone. Again, there is a performance tradeoff as the traffic reduction is balanced by a computational load increase, due to the OCSP reply signing.

5. Conclusions

This work summarizes the experience gathered during a complex work, which involved deep study, application and extension of the concepts related to public-key certificate handling. The resulting architecture exploits the best features of two different certificate status handling schemes, namely CRL and OCSP, to obtain, within a peculiar citizens-to-administration communication model, a good balance between security, timeliness and efficiency.

Research studies currently being undertaken aim to devise schemes based on untrusted directories that make both the communication traffic, particularly the directory incoming traffic, and the overall computational load less dependent from the number of PKI users and certificates status update frequency. Interesting results seem to emerge from approaches exploiting OWA cryptographic primitives [12], and incremental cryptography techniques [13-16].

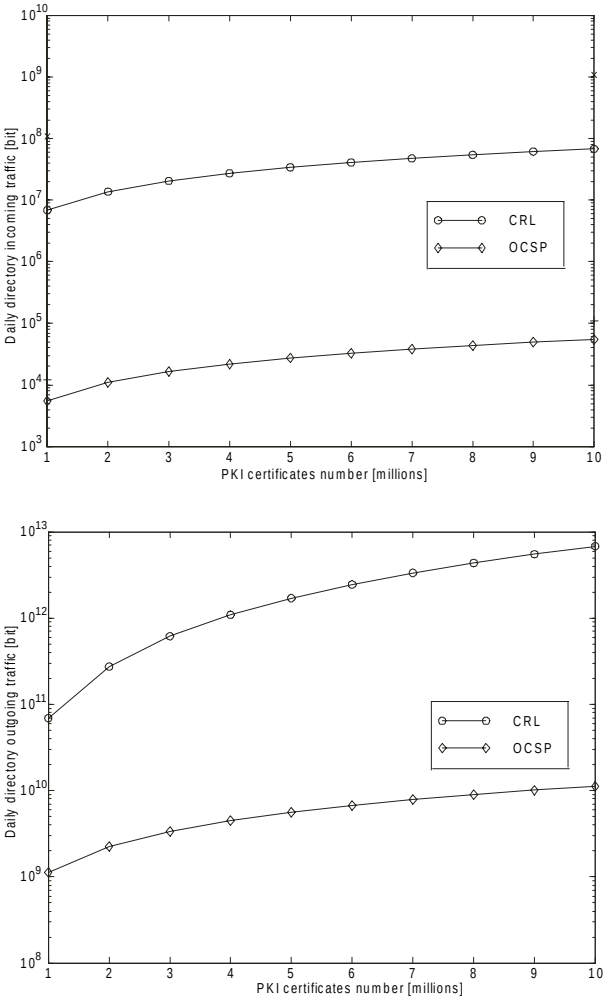


Fig. 4 – Directory traffic dependence on the number of certificates handled in the PKI

Acknowledgements

The author gratefully acknowledges the contribution of Modena city administration in developing

the prototype system. The author would also thank Prof. Eugenio Faldella, University of Bologna (Italy), and Mr. Giovanni Faglioni, IS consultant, for their support with several useful discussions, and Dr. Andrea Giacobazzi for the actual development of OCSP code within *mod_ssl* as part of his CS Engineering Master's thesis.

References

[1] W. Diffie, M. E. Hellman: *New directions in cryptography*, IEEE Transactions on Information Theory, v. IT-22, n. 6, Nov 1976, pp. 644-654.

[2] R. Rivest, A. Shamir, L. Adleman: *A method for Obtaining Digital Signatures and Public-key Cryptosystems*. Communications of the ACM, v. 21, n. 2, Feb. 1978, pp. 120-126.

[3] R. Housley, W. Ford, W. Polk, D. Solo: *RFC 2459 – Internet X.509 Public Key Infrastructure Certificate and CRL Profile*. The Internet Society, Jan. 1999 – http://www.rfc-editor.org/rfc/rfc2459.txt

[4] M. Myers, R. Ankney, A. Malpani, S. Galperin, C. Adams: *X.509 Internet Public Key Infrastructure Online Certificate Status Protocol – OCSP*. IETF, Sep. 1998 – ftp://ftp.ietf.org/internet-drafts/draft-ietf-pkix-ocsp-08.txt

[5] A. Malpani, C. Adams, R. Ankney, S. Galperin: *Internet Public Key Infrastructure Real Time Certificate Status Protocol – RCSP*. IETF, Mar. 1998 – ftp://ftp.ietf.org/internet-drafts/draft-malpani-rcsp-00.txt

[6] E. Faldella, M. Prandini: *Efficient Handling of Certificates within Public-Key Infrastructures.* Proceedings of the 3rd IMACS/IEEE International Multiconference on Circuits, Systems, Communications and Computers (CSCC'99), post-conference book, v. 2.

[7] NIST: *Public-key Infrastructure Study.* Gaithersburg, MD, April 1994.

[8] W. Dai: *Speed Comparison of Popular Crypto Algorithms* – http://www.eskimo.com/~weidai/benchmarks.html

[9] *Internetworking Public Key Certification Infrastructure for Europe (project programme)* http://www.darmstadt.gmd.de/ice-tel/programme/programme.html

[10] The *mod_ssl* web site: http://www.modssl.org

[11] The *Apache* web site: http://www.apache.org

[12] J. Benaloh, M. de Mare: *One-Way Accumulators: A Decentralized Alternative to Digital Signatures (Extended Abstract)*. Eurocrypt '93 Proceedings, pp. 274-285.

[13] M. Bellare, O. Goldreich, S. Goldwasser: *Incremental Cryptography: the Case of Hashing and Signing*. Crypto '94 Proceedings, Lecture Notes in Computer Science, v. 839, pp. 216-233, Springer-Verlag, 1994.

[14] M. Bellare, O. Goldreich, S. Goldwasser: *Incremental Cryptography and Application to Virus Protection*. Proceedings of the 27th Annual ACM Symposium on the Theory of Computing, pp. 45-56, 1995.

[15] M. Bellare, R. Guerin, P. Rogaway: *XOR MACs: New Methods for Message Authentication Using Finite Pseudorandom Functions.* Crypto '95 Proceedings, Lecture Notes in Computer Science, v. 963, pp. 15-29, Springer-Verlag, 1995.

[16] M. Fischlin: *Incremental Cryptography and Memory Checkers*. Eurocrypt '97 Proceedings, Lecture Notes in Computer Science, Vol.1233, pp.393-408, Springer-Verlag, 1997.