A Flexible Management Framework for Certificate Status Validation

ANTONIO CORRADI[1], REBECCA MONTANARI[1], CESARE STEFANELLI[1], DIANA BERBECARU[2], ANTONIO LIOY[2] and FABIO MAINO[2]

1.Dip. Di Elettronica, Informatica e Sistemistica, Università di Bologna, Italy: 2.Dip. Di Automatica e Informatica,Politecnico di Torino, Italy

Key words: Security, Public Key Infrastructure, Certificate Status Validation, Mobile Agents

Abstract: Public key cryptography is widely recognised as the technology to develop effective authentication, integrity, confidentiality and non-repudiation services. The provision of public key-based security services for complex and large scale organisations requires a Public Key Infrastructure (PKI) in charge of securely managing cryptographic keys/certificates. An essential PKI service is the certificate status validation (CSV) system that supports the publishing and the consistent usage of certificate status information for wide range of applications. Several CSV solutions, such as Certificate Revocation Lists or the On-line Certificate Status Protocol, are available, but none can meet the requirements for all applications, in particular of timeliness and performance. The lack of a comprehensive CSV solution calls for the development of a flexible framework that can integrate all available validation mechanisms and permit the selection of alternative validation strategies, depending on application requirements. The paper describes this framework that provides PKI users with a flexible, dynamic and transparent CSV support. In addition, the paper claims that the framework flexibility, dynamicity and transparency can greatly benefit from the adoption of the Mobile Agent (MA) technology because it exhibits the same intrinsic features, by presenting an MA-based prototype for CSV.

The original version of this chapter was revised: The copyright line was incorrect. This has been corrected. The Erratum to this chapter is available at DOI: 10.1007/978-0-387-35515-3_53

S. Qing et al. (eds.), *Information Security for Global Information Infrastructures*
© IFIP International Federation for Information Processing 2000

1. INTRODUCTION

Computer networks offer significant business opportunities but greatly increase the risk of exposing the system to security breaches [1] that can make it unreliable and/or unusable and can make its services unavailable.

Public key cryptography has demonstrated its effectiveness in achieving scalable confidentiality, integrity, authentication, and non-repudiation services, defying many of the threats posed by the deployment of open networks. However, the new possibilities stemming from public-key cryptography have reinforced the need for sophisticated systems to manage key lifecycle in complex organisations, from the initial secure distribution of keys to their successive update, suspension or revocation.

The Public Key Infrastructure (PKI) is the emerging technology to support the use of public key cryptography in large scale networks by providing automatic and scalable key/certificate management [2]. PKI services can enable authentication, authorisation, encryption and digital signature in a variety of Internet/Intranet applications, including secure messaging and electronic commerce. The importance of PKIs in the provision of commercial, educational and personal services is widely recognised and cannot be neglected.

However, some significant issues in the development of large scale PKIs are still to be solved. One particularly debated aspect in PKI design is the mechanism for validating certificate status information. Several solution have been proposed, including the Certificate Revocation List scheme (CRLs) [3], the On-line Certificate Status Protocol (OCSP) [4] or the Certificate Revocation Trees (CRT) [5], and all of them presents advantages and disadvantages subject to discussion.

None of the mechanisms currently used for certificate status validation (CSV) can alone meet the timeliness and performance requirements of all applications. The requirements can greatly vary depending on the type of transaction the certificate is to be used for, or on the amount of latency and computational load the involved parties are willing to tolerate. For example, high value financial transactions could require strict real-time support for certificate status validation, while low value commercial applications could tolerate higher latency [6], [7]. An additional complexity in CSV is solution heterogeneity in PKI environments. It is very unlikely that all PKI organisations employ the same validation mechanism, whereas it is more realistic to assume that they deploy different methods to embody their management and security policies.

Only the integration and support of different CSV mechanisms seems a proper answer to different application requirements and to PKI heterogeneity. Along this guideline, few solutions permit the inclusion and enabling of all major validation mechanisms into custom applications [8].

This paper does not present another method for CSV. Rather it proposes a framework to provide a flexible integration of solutions and a coherent CSV management in wide application contexts and in heterogeneous PKI environments. The framework is designed to accommodate the specific application trade-off between timeliness and resource usage by permitting dynamic selection of the CSV methods, and to provide final users with automatic and transparent CSV support to overcome PKIs heterogeneity.

We claim that the development of the framework can greatly benefit from the exploitation of the Mobile Agent (MA) technology that can help in providing the required degree of flexibility, dynamicity and transparency because it possesses the same intrinsic properties [9]. In addition, MAs are designed to overcome the limits of the traditional client/server (C/S) model, thus permitting to improve the timeliness, efficiency and scalability of current C/S based certificate status validation mechanisms. The paper describes the implementation of a MA-based flexible prototype for certificate status validation, built on top of the MA system called SOMA (**S**ecure and **O**pen **M**obile **A**gent) that permits an easy design of the framework services by providing appropriate levels of security and interoperability with existing PKI systems [10].

2. A FRAMEWORK FOR CSV

The lack of a unique and comprehensive solution to handle CSV motivates the development of a framework that can flexibly integrate all available mechanisms and can support the certificate validation in a dynamic and transparent way in all application scenarios and in heterogeneous PKI environments.

We propose a framework for CSV in PKI environments composed of the following coordinated distributed services:

- the *configuration service* that supports PKI administrators in setting up, updating and integrating different CSV mechanisms;
- the *repository service* that is in charge of maintaining a description and a reference to the CSV mechanisms currently available in its PKI;
- the *discovery service* that is in charge of retrieving from the repository service the information about the mechanisms for CSV on behalf of PKI end-users;
- the *validation service* that is responsible for supporting end-users in the certificate checking phase either in the case of pull and push certificate distribution model [2], according to user requirements;

- the *preference service* in charge of helping PKI end-users in the specification/enforcement of their preferences and their dynamic maintenance. Preferences can include requirements for specific CSV tailored to user application context, or to the maximum latency or to the processing power of the current terminal of a user (laptop, personal data assistant, PC or workstation). Moreover, preferences can specify actions to perform in case of an invalid certificate, from the issuing of a user warning to the logging of the event for future reference.

We claim that the framework should be developed according to several guidelines. *Flexibility* is the basic design criterion to satisfy in order to enable the selection of alternative validation strategies on the basis of application and management requirements. For this reason, the framework should integrate all available CSV mechanisms and should support both pull and push certificate distribution models. Another primary requirement is *dynamicity*. Because the availability of the CSV service is a necessary and central condition for PKI liability, the framework should guarantee the service availability while incorporating new CSV mechanisms. A final but fundamental consideration is to meet an adequate level of *transparency*. Transparency is an essential property for a PKI to be effective. The framework should allow final users to ignore the underlying available CSV mechanisms and should relieve them of the effort of dealing with heterogeneity of solutions.

3. MOBILE AGENTS FOR MANAGING CSV

The framework for CSV can be implemented by using different distributed programming technologies. The C/S model could be adopted while a less traditional choice could lead to explore the use of programming model based on some forms of mobility, such as the Code on Demand, remote Evaluation, and Mobile Agent cases [9].

In our opinion, the realisation of the framework can significantly benefit from the adoption of the MA technology to achieve the needed degree of transparency, dynamicity and flexibility because the paradigm itself is designed according to the same properties [11]. MAs can be effectively exploited to optimise CSV, to support flexible and dynamic integration between different validation mechanisms and to deal with PKI heterogeneity on behalf of their responsible users. Mobile agents are software entities that are designed to autonomously roam networks by transferring themselves (code and state together) to find and to locally process the information they have to operate upon, thus reducing network traffic and latency [11].

A primary advantage of adopting MAs is the possibility to achieve the *flexibility* requirement of the CSV framework. MAs can be used to distribute certificate status information by providing efficient pull and push distribution. In particular, MA facilitates the development of push distribution models because they are themselves designed according to the so-called Internet push model.

In addition, the possibility to dynamically extend MAs functionalities without interrupting their execution, permits to answer the *dynamicity* requirement of the CSV framework. For instance, MAs could be exploited to dynamically update the CSV protocols without CSV service suspension.

Moreover, the autonomy property of mobile agents can address the *transparency* requirement of the framework. The introduction of MAs autonomously acting on behalf of PKI end-users could reduce the effort required to complete a CSV by transparently retrieving all the necessary information. The use of MAs can also permit to deal with the heterogeneity of CSV mechanisms. MAs could move to target PKIs and there establish "channels" based on proprietary CSV protocols without requiring ad-hoc configuration of user applications.

4. ONE MA-BASED PROTOTYPE FOR CSV

The proposed infrastructure for CSV has been realised by exploiting the underlying facilities of the MA system called SOMA [10]. SOMA is a platform for the design, implementation and deployment of distributed applications in several application scenarios in open, global, and untrusted environments, such as the Internet.

4.1 The SOMA environment

SOMA provides a hierarchy of locality abstractions that helps in modeling different schemes of interconnected networks, ranging from simple LANs to architectures composed of several LANs variously interconnected by bridges, routers, gateways, and firewalls. Any node owns at least one place that constitutes the agent execution environment. Several places can be grouped into a domain abstraction that corresponds to a network locality. In each domain, a default place hosts a gateway to perform inter-domain functionality and to maintain domain-specific runtime information.

The SOMA programming framework is designed according to a layered architecture built on top of the Java Virtual Machine (JVM). The basic layer provides a rich set of facilities to simplify the deployment of MA-based services, such as the *naming* facility to associate entities with globally unique identifiers, the *migration* facility to support agent mobility, the *communication* facility to provide tools for coordination and communication between possibly mobile entities, the *monitoring* facility to observe the state of local resources and service. On top of the basic layer, the interoperability facility exploits the CORBA technology [12] to achieve the full integration with other CORBA-compliant environments, while the security facility makes available a large range of security mechanisms and tools, to permit different trade-offs between performances and security levels.

In particular, the *security* facility provides tools for protecting both places and mobile agents. Authentication of incoming mobile agents is based on standard certificates and on a public key infrastructure; authorisation extends the Java standard mechanisms for access control. Secrecy and integrity for agent/message transfer in SOMA exploits standard solutions for securing the communication layer. In particular, SOMA employs Transport Layer Security (TLS) protocol [13] that can support a set of different cryptographic algorithms to provide channel integrity, authentication, and secrecy. The protection of mobile agents from their execution environment has required the development of specific protocols that can suit different application scenarios. The goal of the protocols is to detect any possible modification of the data collected by agents executing in an untrusted domain [14]. One solution, called Trusted Third Party protocol, requires the presence of a centralised entity that offers a trusted and secure environment to agents in need of performing cryptographic functions. In this case, after any visit to any untrusted site, an agent should go to the centralised entity site to validate its computation: this increases the number of hops in the agent path, but permits to checkpoint the agent state in a safe place. An alternative solution, called Multiple-Hops, achieves agent integrity without the need of any trusted centralised entity. Each host must provide a short proof of its agent computation, which is stored in the agent state. Each proof is cryptographically linked with the ones computed at the previous sites so that it is impossible to modify an intermediate proof without modifying also all successive ones. When the agent moves back to its owner, the integrity of the "chain" of cryptographic proofs is verified permitting to the owner to detect any integrity violation. Further details about the SOMA programming framework and its implementation are presented elsewhere [10] and are out of the scope of this paper.

4.2 The SOMA-based CSV Prototype

The locality abstractions of the SOMA system can easily model complex PKI infrastructures that can describe very different organisations, from strictly hierarchical to fully distributed [15]. Any PKI environment is represented by a different SOMA domain and each PKI component

(Certification Authority, CSV server, end-user) can be associated with one SOMA place. SOMA domains can be organised into hierarchical or distributed architectures depending on PKI organisational and security policies.

Within each SOMA domain the framework for CSV is implemented by several coordinated agents, anyone is in charge of specific management tasks. We distinguish between stationary and mobile agents: a stationary agent does not change its allocation, while a mobile agent is not bound to the system where it has started execution and can be exploited for information retrieval and updating.

As shown in Figure 1, the configuration service of the framework can exploit mobile Configuration Agents (CoA) for the dynamic installation/upgrade of CSV protocols. CoAs can be sent to CSV servers to analyse their configuration state and perform installation/upgrade steps accordingly.

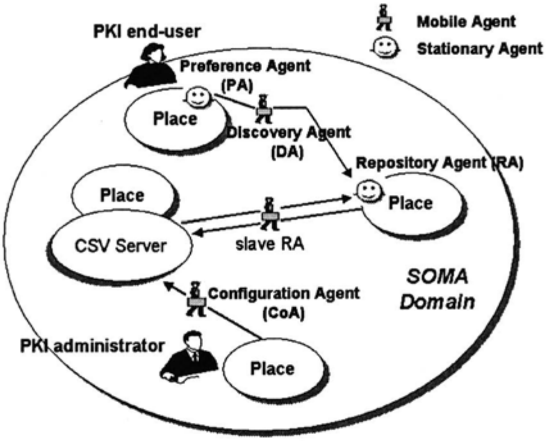


Figure 1. The SOMA based framework for certificate status management

In a SOMA domain at least one stationary Repository Agent (RA) provides access to the repository service of the framework. The RA may be designed according to the master/slave model: slave mobile RAs can be sent to CSV servers to observe configuration changes; at any detected change (introduction of new CSV protocols or protocol versioning update), the slaves can communicate it back to the master RA that updates accordingly the database storing the description of CSV mechanisms.

Discovery Agents (DA) are in charge of supporting the discovery service on behalf of PKI users: they can migrate to the node hosting RA execution to locally process CSV information, with the advantage of better network utilisation.

Also CSV servers can benefit from the exploitation of mobile agents. MAs can be exploited to observe certificate status information updates and, as soon as a change is identified, they can notify interested users depending on their preferences. A user could require to be notified as soon as a certificate status changes, while others only at predefined time intervals.

Finally, end-users can exploit Preference Agents (PA) that are in charge of coordinating the enforcement of user preferences. PAs can create DAs for the discovery of CSV methods in PKI environments and can delegate them for spawning other agents in charge of performing the CSV protocol steps. In addition, PAs can dynamically propagate to DAs preference updates and DAs can adapt their behaviour accordingly, without restarting their execution.

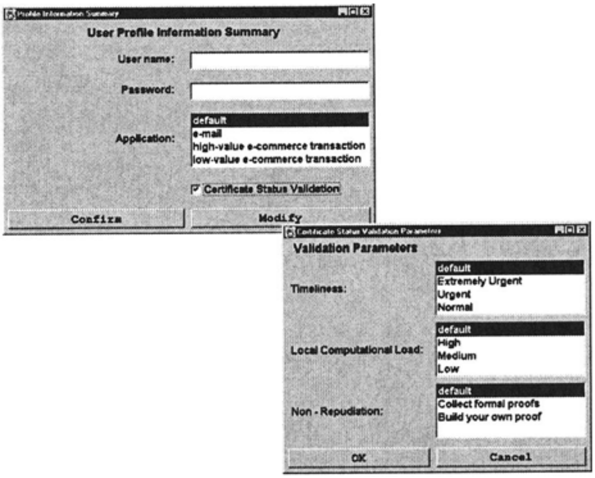
It is worth noticing that both stationary and mobile agents demand for an adequate level of security. Agents encapsulate critical management functions and intrinsically require strong protection both while migrating and executing in hosting nodes. In particular, the user preferences embedded within PA could require integrity: only its responsible users should modify this information, and the preferences passed by PAs to DAs should be unforgeable. If any tampering occurs, DA behaviour could be irremediably modified and PKI end-users damaged. We claim that the SOMA security facility described in section 4.1 can provide the required integrity mechanisms to build a secure infrastructure for CSV. In addition, fault tolerance techniques [16] can be exploited to guarantee agent availability in case of communication failures or attacks that may deny agents from roaming.

4.3 A Case Study

We have tested the SOMA-based prototype for CSV within the EuroPKI project [17]. The EuroPKI is an European wide PKI that has been set up within the ICE-TEL and ICE-CAR projects with the goal of providing the necessary support for the deployment of secure applications for Commerce, Public Administrations and Research Institutions. The EuroPKI intends to offer CSV services with openness and flexibility that does not bind the PKI users to a specific validation mechanism. For example, the PKI of the Politecnico di Torino provides its users with two CSV mechanisms: monthly issued CRLs (distributed via HTTP), and one OCSP responder. The other PKIs only provide CRLs generally issued once per month, or once a half-year for the root CA.

At system logon any PKI user is assigned with a single PA. The user, by employing the simple interface shown in Figure 2, can configure her PA according to the timeliness to be respected for any specific application, the tolerated computational overload, the required degree of non repudiability and the priorities to give to these requirements.

The application requirements drives the PA configuration: in some scenarios the promptness in providing certificate status information is the most important attribute, while in others is the liability associated with the provider of the status information, or the possibility to access status information on-line. On the contrary, if the user does not specify any CSV requirements, the PA is configured with default settings that reflect the certificate management policy of the user PKI.



User Profile Information Summary

User name:

Password:

Application: default

☑ Certificate Status Validation

Validation Parameters

Timeliness: default

Local Computational Load: default

Non - Repudiation: default

Figure 2. The Profile Agent Interface

As a final consideration, PAs can be easily plugged into all common applications by permitting transparent, efficient and interoperable CSV: when an application requires a certificate check, PAs are automatically invoked and the CSV process is performed according to the application CSV preferences.

5. CONCLUSIONS

The paper presents a framework for CSV that provides a flexible, dynamic, transparent, and interoperable environment to PKIs in need of offering services for large and heterogeneous user communities. The framework can benefit from the intrinsic characteristics of MAs and from the specific facilities of the SOMA system. We have implemented and tested a SOMA-based prototype for CSV within the EuroPKI infrastructure by providing users with a flexible common interface that allows them to specify and choose validation preferences in terms of timeliness, local computational overload, and non-repudiation. Future work will be aimed at implementing several plug-ins that permits common applications (e-mail and web client) to interact with our SOMA-based CSV system.

Acknowledgements

Investigation supported by the Italian "Consiglio Nazionale delle Ricerche" in the framework of the Project "Global Applications in the Internet Area: Models and Programming Environments" and by the University of Bologna (Funds for Selected Research Topics: "An integrated Infrastructure to support Secure Services").

Reference

[1] W. Stallings, *"Network and Internetwork Security: Principle and Practice"*, Prentice Hall, 1995.

[2] W. Ford, M. Baum, *"Secure Electronic Commerce"*, Prentice-Hall, 1996.

[3] R. Housley, et alii, RFC 2459, "Internet X.509 Public Key Infrastructure Certificate and CRL Profile", *Internet Engineering Task Force*, 1999.

[4] M. Myers, et alii, RFC 2560, "X509 Internet Public Key Infrastructure Online Certificate Status Protocol – OCSP", *Internet Engineering Task Force*, 1999.

[5] P. Kocher, "On Certificate Revocation and Validation", *Financial Cryptography*, Anguilla, 1998.

[6] M. Myers, "Revocation: Options and Challenges", *Financial Cryptography*, Anguilla, 1999.

[7] B. Fox, B. LaMacchia, "Online Certificate Status Checking in Financial Transactions: The Case for Re-issuance", *Financial Cryptography*, Anguilla, 1999.

[8] Valicert Validator Suite, http://www.valicert.com/html/validator_suite.html.

[9] A. Fuggetta, et alii, "Understanding Code Mobility", *IEEE Transactions on Software Engineering*, Vol. 24, No.5, 1998.

[10] P. Bellavista, et alii., "A Secure and Open Mobile Agent Programming Environment", *ISADS'99 – IEEE International Symposium on Anonymous Decentralized Systems*, Tokyo, 1999.

[11] D. Lange, M. Oshima, *"Programming and Deploying Java Mobile Agents with Aglets"*, 1998, Addison Wesley.

[12] Object Management Group, CORBA/IIOP Rev 2.2, OMG Document formal/98-07-01, 1998.

[13] T. Dierks, C. Allen, RFC 2246, "The TLS Protocol Version 1.0", *Internet Engineering Task Force*, 1999

[14] A. Corradi, et alii., "Mobile Agents Integrity for Electronic Commerce Applications", *Information Systems*, Elsevier, Vol. IS24, No.6, 1999.

[15] R. Perlman, "An Overview of PKI Trust Models", *IEEE Network*, Vol. 13, No.6, 1999.

[16] D. Johansen, et alii., "NAP: Practical Fault-Tolerance for Itinerant Computations", *ICDCS'99 – IEEE 19th International Conference on Distributed Computer Systems*, Austin (TX), 1999.

[17] D. Chadwick, A. Young, "Merging and Extending the PGP and PEM Trust Models - The ICE-TEL Trust Model", IEEE Network, Vol. 11, No.3, 1997.