# Digital Signature in Automatic Email Processing: A Customer Case Study

Authors:
Francesco Gennai
Network Systems and Services team leader, ISTI-CNR, Pisa, Italy
Marina Buzzi
Technologist, IIT-CNR, Pisa, Italy
Revised by Giovanni Vischio, HP Italy, OpenVMS Ambassador

## Overview

Today many services can be requested over the network by sending an electronic form or message to the service provider. If the data is coded in syntax that can be read by computers, it is possible to automate its interpretation, elaboration, and storage, thus speeding up data processing and reducing human error.

A digital signature can be associated with Internet messages to guarantee sender authentication, message integrity, and nonrepudiation of origin. The verification process for incoming digitally-signed messages is usually performed by the email client on behalf of the end user. However, if a digital signature is applied to data that is subjected to automatic elaboration in order to maintain the above-mentioned benefits, it may be convenient to automate the verification process as well. Our idea is to implement the verification process in the email server rather than the client.

This paper describes our experience in designing and implementing software to automate the verification of digitally-signed messages and web forms in order to simplify the registration of Internet domains under the .IT Top Level Domain.

## Introduction

Verification of incoming digitally signed messages is a critical operation that is usually performed by the receiver using an email client (receiving agent). If the verification process is unsuccessful (that is, the "from" header field does not match any of the email addresses present in the associated certificate), the client alerts the user. However, the user plays a fundamental role in the entire process by doing the following:

- Correlating the message content with the sender (usually a relationship exists between sender and content)
- Authorizing the client to upload or remove certificates of trusted Certification Authorities (CAs)
- Configuring the client to retrieve the CAs' Certificate Revocation Lists (CRLs) to ensure the correctness of the verification process.

In system design, it is very important to evaluate the purpose as well as the context in which the digital signature is applied. It is obvious that the automation of the verification process is applicable only within certain contexts. Unless a semantic relationship exists between data and sender that

only the receiver can understand, the verification process can be transferred to automatic systems, which can process a large number of messages in a single time unit.

The Message Verify (MV) system has been designed to simplify the registration of Internet domains under the .IT Top Level Domain (TLD), performed by the Italian Registration Authority[1] (RA).

Usually ISPs (or maintainers) register domain names on behalf of third parties (organizations, associations, individuals, and so forth) and also take charge of technical maintenance duties such as managing DNS for the requested domain. Each domain name registration requires a two-step procedure:

- Fill out and send a letter of assumption of responsibility (LAR) for the use of the domain, signed by the registrant (using fax or surface mail)
- Send one electronic mail[2] message containing technical and administrative data needed for the registration (domain name, organization, administrative contact, and so forth).

The email can be automatically generated by filling out a web form or manually composed as a text file with well-established fields conforming to a predefined syntax (for example, name:value). The elaboration of these data is automatic. The software for data processing is triggered upon message reception. It performs user access control and verifies syntactical and technical correctness of data (for example, it checks DNS configuration). If any control fails, an automatic error notification is created and sent to the sender. User authentication is performed by means of a password assigned to the ISP. This password permits the user to access the web form for the domain registration. Alternatively, the password can be ciphered and included in the plain text message. With the same password, an ISP can view the status of its' ongoing registrations using web interfaces.

The MV system introduces the use of digital signature instead of user/password authentication. The use of digitally signed messages offers strong user authentication (compared to the user/password mechanism) and guarantees data integrity as well as nonrepudiation of origin, which is fundamental for proving data trustworthiness in case of legal problems (for example, between the RA and its customers).

An alternative to signed emails is the use of digitally signed web forms. Both technologies are valid. Using an email server, which is a "store and forward" service rather than an interactive service like the web, allows the masking of any processing delay and accidental server unavailability from the user. However, web service offers the user more awareness of the session progress (for example, data transmission and elaboration).

Our study embraces both the design and development of an automatic system for verifying digitally-signed messages and web forms as well as the activation of security mechanisms to maintain an adequate degree of operational security and reliability.

## The MV System

The MV system was designed to be transparently placed between the message's arrival and the automatic elaboration of its content. Transparency requires that the pre-existing legacy software for automatic elaboration remain unchanged. This basic requirement was the first constraint on the system design. Our system automatically verifies the digital signature associated with an incoming

[1] The Italian Registration Authority at `http://www.nic.it/`

[2] The email must be sent from the provider/maintainer within ten working days from the date the RA received the LAR; otherwise the domain name becomes available for a new assignment.

message before transmitting data to the process for automatic elaboration. If the verification is successful, the request is accepted and elaborated; otherwise it is rejected and a notification is automatically sent to the sender.

First we studied the automation of the digital signature's verification process. This required studying integration of Public Key Cryptography Technology with the electronic mail system and understanding security service mechanisms based on cryptography.

Another fundamental question was how to carry out the function performed by the user interacting with the client software (that is, to choose trusted CAs and download certificates or CRLs). To maintain control over the verification process, all functions the user performs by means of an email client must be assigned under the control of a system administrator using a simple web interface.

To perform the verification process, the receiving agent extracts a great deal of information from the message: the cryptographic algorithms applied for signing the message (one-way hash function, encryption algorithm), the sender certificate, and the CA certificate chain (if present). The verification environment is quite complex. This can be explored in detail by following activities and studying documents by the IETF working group: S/MIME Mail Security. However the verification process can be logically divided into two main functions:

- Recognize message MIME parts containing protected data. The RFC 1847 "Security Multi-parts for MIME: Multipart/Signed and Multipart/Encrypted" specifies how to apply security service to MIME body parts. It adds two new content types: Multipart/Signed and Multipart/Encrypted. Both of these content types contain two body parts: one for the protected data and another for the control information necessary to remove the protection. RFC 2630 describes the Cryptographic Message Syntax used to digitally sign, digest, authenticate, and encrypt messages, and the RFC 2633 defines the application/pkcs7-mime and application/pkcs7-signature MIME types used to transport S/MIME signed messages.

- Apply the verification process to the extracted MIME parts. Mechanisms are needed to retrieve and validate certificates. The RFC 2632 specifies basic rules to be implemented by receiving agents to correctly verify a signed message. In addition, the RFC 3280 outlines the format and semantics of certificates and certificate revocation lists for the Internet PKI. RFC 2560 describes procedures for processing certification paths in the Internet environment.

## Operative Environment

The operative environment consists of a 2-node SCSI cluster sharing an external disk array. Each node is a COMPAQ AS 800 5/500 (1 GB of RAM, 4GB internal disk) running OpenVMS Version 7.3-1. The external storage is a Digital RAID7000 configured as follows:

- 2 RAID5 string of 6 x 36.4 GB
- 1 RAID5 string of 3 x 18.2 GB
- 1 RAID5 string of 6 x 9.1 GB

The MV system is composed of numerous software modules written in DCL, and utilizes the cryptography libraries of OpenSSL (v. 0.9.6d, the openssl.org standard distribution). It interacts with the PMDF electronic email/FAX system (v. 6.2, http://www.process.com/) and the HTTP OSU Web server (v. 3.9c, Ohio State University Web Server) running on both the cluster nodes.

The OpenSSL toolkit has been fundamental for system development. OpenSSL offers libraries for developing cryptographic software and for managing PKI objects (certificate manipulation, basic CRL manipulation, basic CA management, signing, encrypting, decrypting, verifying, and so forth) as well as line-mode tools. By using these commands, the MessageVerify system performs most controls including signature verification, control of valid paths to trusted CAs, verification of accessibility and validity of CRLs, control of expiration, and revocation of certificates.

## The Architecture

The digital signature process works as follows:

- A user sends a digitally-signed message to a mailbox configured for automatic elaboration of message content.
- A unique identifier is assigned to the incoming message.
- The digital signature is verified.
- If verification is successful, the signed part is extracted and sent to the elaboration process.
- If verification fails because of a temporary error (for example, a CRL is not available), the message is queued for future attempts until an established threshold is reached (see Configuration Parameters).
- If verification fails because of a permanent error (for example, no digital signature is present), an error notification is automatically sent to the sender and the system administrator.

The system is composed of software modules, databases, and web interfaces for administration and control. Each module, which can be executed in parallel on each node of the cluster, performs one specialized function. Figure 1 shows the logical scheme of the Message Verify system, where processes (that is, modules in execution) are represented by rectangles and databases are represented by ellipses.

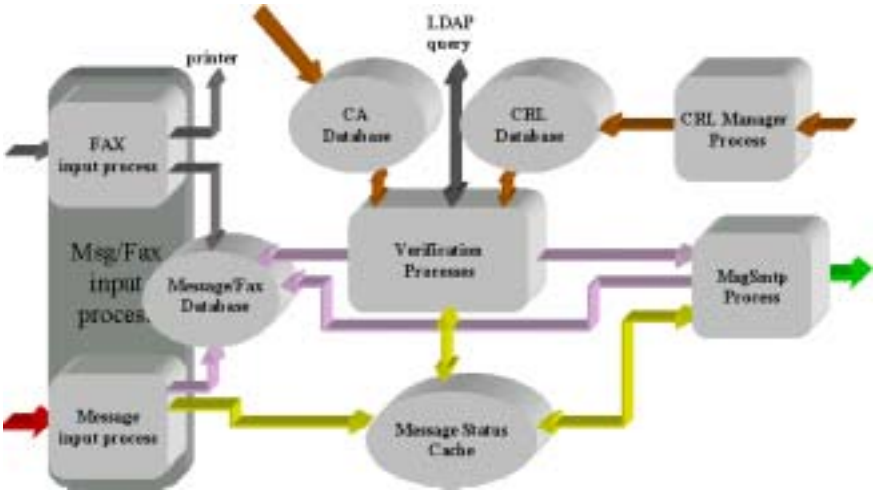


Figure 1. MV System's Logical Scheme

Modules perform the following functions:

- The Message Input process performs preprocessing of messages to extract data necessary for subsequent elaboration, synchronization between the MV system and fax-RA system (receiving LAR) to generate the unique identifier, and message queuing.

- The Verification process is the heart of the MV system. It verifies the signature's validity and adds four header fields to the message to specify the certificate Issuer Distinguished Name, the certificate Subject Distinguished Name, the certificate serial number, and the concatenation between the verification process return code and the message global identifier. In this way the stored messages keep information about the result of the verification operation.

- The MsgSmtp process composes the final message, removing the MIME part that includes the digital signature, and sends it toward the RA systems. Messages are processed in parallel on a 2-node cluster, which means they may be out of sequence after verification. The process reorders the messages in the correct temporal sequence, according to the message status present in the Message Status Cache, before sending them to the automatic elaboration process (see Figure 2).

- The CRL Manager process automates the management of the CRL's database and keeps it up to date. The module itself notifies the administrator of any errors or warnings (for example, the proximity of the CRL's expiration date).

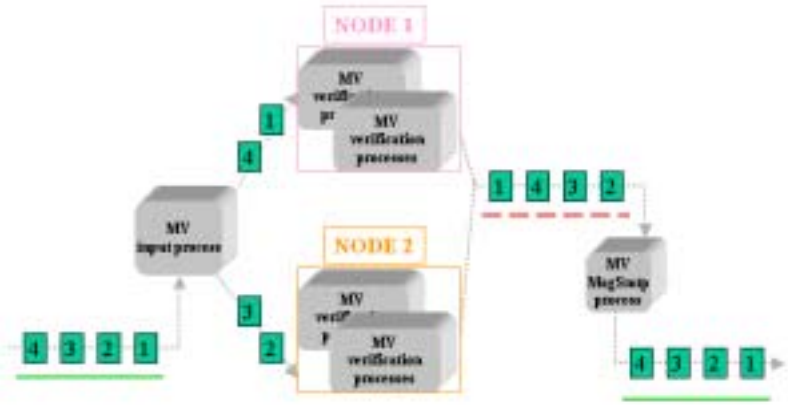


Figure 2. Message Flow

The MV system utilizes the following databases:

- Message/Fax Database
  The message/fax database includes both messages and requests received by means of fax, stored as postscript files. A global identifier is assigned to each message or fax entering the system (by the message or fax input processes), thus maintaining the temporal sequence of the requests; this step is an important factor in being able to resolve collisions on requests for the same domain name.

- CA Database

Certificate of trusted CAs are added to (or removed from) the CA database by a system administrator. The system can use certificates issued by any CA after its own and any sub-CA certificates have been loaded in the database. The loading can occur by one of the following means:

- HTTP session with the server web where the CA certificate is published
- File upload from the administrator's local disk

- CRL Database
  CRLs are automatically downloaded (using the Certificate's CRL Distribution Point field or one system configuration parameter) by the CRL Manager process (which uploads the local CRL database). LDAP query is possible but not yet implemented.

- Message Status Cache

The message status cache is used for greater efficiency and separates process information from that related to fax and messages. It maintains temporary information on the message's status as the message undergoes elaboration.

Management and control of the system is performed using web interfaces.

## Notifications and Errors

The verification process can fail for several reasons, including the following:

- The "from" header field does not match any of the email addresses present in the certificate (*subjectAltName* field and *emailAddress* attribute of distinguished Name field).
- A path to a trusted root CA does not exist.
- It is not possible to retrieve a CRL.
- CRL is not valid.
- CRL is expired.
- Certificate is expired.
- Certificate is revoked.

Failure can be temporary or permanent. For temporary failures (for example, CRL is not available) the system makes a fixed number of retries until it reaches a threshold that is configured through the administrative interface. Both notifications and errors are inserted into the *Subject* header field and into other internal message fields (From:, Global ID:, Message ID:, Previous Message ID:, Error-type:, Number of retries:, Cert Issuer:, Cert Subject:).

## System Extension

Recently, a new component (Form Input Process) was added to the system to accept signed web forms that contain the domain registration data. Using a Netscape browser (4.04 or higher), it is possible to apply a digital signature to a web form. Netscape V4.04 and higher include a JavaScript method (crypto.sign.Text) that enables the user to apply a digital signature to input data. Conforming to the PKCS7 and to the cryptographic message syntax (which in part extends the

PKCS7), Netscape utilizes the external signature modality. The resulting signed-data structure does not contain the original data, which must be dealt with separately. In this case, the session is interactive and the user receives acknowledgement of the signature verification process along with a unique identifier, which can be used to access the state of the request.

To synchronize the unique identifier among messages, fax, and web forms (in a cluster environment) required rewriting the algorithm of allocation of the identifier generators (msgid, faxid and globalid). The new algorithm can manage any level of conflict of access to the identifier generator. When there is a conflict in resource allocation, it is simpler and correct to generate an error message to the interactive user (by web access) rather than interfere with the messaging or fax processes. Based on these considerations, the algorithm can favor allocation requests arriving from message or fax processes by "interrupting" the concurrent CGI/HTTP process.

## Web Interfaces

The MV system can be fully configured and monitored using the web. The user or administrator interface requires user authentication using one of the following access modes:

- Normal login (for operators) furnishes access to the message/fax archive, which is useful for domain name modification, monitoring, or removal, and for supporting help desk activities. It permits searches based on unique a identifier or date, visualization and reprinting of faxes and messages, and access to verification process results.

- Privileged login allows the system administrator to do the following:
  - Add and remove CA certificates.
  - Manage the CA certificates database, and display certificate fields.
  - Display the log file of the CRL manager process in short and extended format.
  - Configure and monitor the system.

  The configuration interface has been designed to be highly configurable to adapt easily to changes in organization policy.

## Related Work and Security Issues

The problem of reducing the cost of bulk cryptographic operations (especially public key operations, which are computationally expensive) has been treated in numerous papers. Two approaches are widely studied: remotely keyed encryption and server-aided cryptography.

- Remotely keyed encryption efficiently achieves encryption by sharing the computational load between a fast but untrusted device and a slow, trusted device. Blaze introduced remotely keyed encryption schemes (RKESs) to support cryptographic applications that require high bandwidth, such as encryption of online multimedia contents, in a secure, low bandwidth, smart card environment (where users' private keys never leave the smart card). In such schemes, the load of communication and computation required of the smart card is independent from the input size [Blaze96]. Further studies to define the security of other RKESs [Lucks97, Blaze98, Lucks99] and other applications have been accomplished. In particular, Weis applied remotely keyed encryption in a java card environment [Weis 2000].

- Server-aided cryptography refers to the ability to aid small and resource-limited devices in expensive computation. Jakobsson introduced this term

while studying how to decrease the local computational cost by transforming a heavy task into a large set of small subtasks carried out by a set of external servers, and performing security analysis of the proposed scheme [Jakobsson01]. By using this scheme he generated inexpensive server-aided batch signatures, showing improved efficiency for groups greater then 20. Recently, Ding applied server-aided cryptography in smart but resource-limited devices such as PDAs, cell phones, and palm pilots [Ding02]. In particular, Ding generated server-aided signature (SAS) while implementing fast certificate revocation, which is fundamental for the accuracy of the verification process. This approach is based on using a partially trusted server, which acts as a security mediator (SEM) executing the main load of the computation. Two half signatures, independent but interrelated, are generated by the SEM and the user to create a SAS signature. A relevant characteristic of this scheme is its online nature permitting fast revocation of signing capabilities, which limits damage from potential compromises. This scheme appears to be suitable for many mobile infrastructures. However, centralization of the architecture and incompatibility of the SAS signature with other signature types, make it impractical at the moment. (SEM is a single point of failure if it is subjected to DoS attack, performance bottleneck, and so forth.)

Berson et al. proposed providing cryptographic operations as a network service [Berson01]. Authors designed and built a centralized cryptoserver equipped with hardware cryptographic accelerators to provide public key operations to clients using their internal network. Because the cost of the accelerator is shared by a large number of clients, this system allows clients to benefit from hardware speedups while reducing CPU load and the cost of a single cryptographic operation. This approach differs from the previous ones because it implies that the client must trust the cryptoserver with knowledge of his private key. This delegation mechanism practically simplifies the design of the user-server interaction, although it poses a security risk of private key storage and protection. Berson observed that outsourcing cryptography inherently raises questions about the trustworthiness of the computation. However, cryptographic operations can be split into two categories: those requiring knowledge of the client's private key (that is, decryption and digital signature), and the other needing only the client's public key (publicly distributed by means of his certificate), which are encryption and verification of digital signature. The first category poses security issues (of secret key storage and protection) and requires more attention than the second category. Thus, instead of building a complete cryptoserver, it is simpler to use a subset of cryptographic operations (encryption and digital signature verification) and limit automation to these. In this case, assuming that the communication between the client and server is secure, the cryptographic operations have the same degree of trustworthiness as performing the computation locally [Berson01].

Our approach is similar to Berson's but is less ambitious. The MV system automates only one cryptographic function, which does not require the use of secret keys. Although our system has this limitation, it is simpler, cheaper (being based on free cryptographic software), and standard – compared to previously discussed schemes – and is interoperable with any type of signature. Our goal is to improve the quality of service offered to a large set of widespread users. In our scenario, automation of the verification process is possible and appropriate because signed data do not contain any semantic information that only the receiver is able to understand. Therefore, the main security issue of the MV system is its integrity. If a system is violated, the entire service can be compromised. For example, if the legacy software receives input data that bypasses authentication, it makes a registration invalid.

In the 2001 annual security report, CERT affirms: "… the ability to attack a system depends on the 'global' security of the Internet." For example, a DoS attack takes advantage of other weak points in the network. This implies that at the moment there is no way to implement a totally secure policy, but it is important to create a strategy and use combined technologies to fend off intruder attacks.

According to this approach, in order to protect the system, different contra-measures have been applied to build protection levels with the following features:

*Software design*:

- The system is designed to be failure-tolerant and robust.
- The MV system and the legacy software are within the same security perimeter.

*Operating environment:*

- Managing network services in an OpenVMS[3] environment reduces the recurring security problems typical of other platforms and enhances compatibility between different versions of software. The benefits of using an OpenVMS system equate to those of Unix and Windows systems, with the notable advantage of being one of the least-attacked systems on the Internet today.
- The system is carefully configured to include access control, logging of user sessions, auditing of security-relevant events, ensuring that all features and services that are not explicitly required are disabled, and so forth.

*Network:*

Protection of the network perimeter includes the following features:

- Blocking the transfer of executable files on the Internet gateway or email server
- Configuring servers to disable features and services that are not explicitly required
- Configuring routers and firewalls to enable traffic only to authorized servers and ports
- Running and maintaining updated anti-virus software
- In addition, Intrusion Detection Systems, which conduct analyses of live network traffic, and Honeypots (usually single systems that emulate systems, known services and vulnerabilities, or create jailed environments) can contribute to monitoring network and system activities and detecting attacks.

Be sure to read security bulletins and promptly apply software patches to the operating system and applications.

Currently, a drawback of the proposed solution is the cost of certificate creation and distribution as well as help desk support. The solution is applied to ISPs, which manage nearly all domain name registrations[4]. This kind of automation will be suitable for everyone when PKI technology further penetrates e-society.

## Performance

Before buying expensive cryptographic hardware and software, it is important to evaluate the real load of the system or application to verify whether high performance is necessary. In our context, we operated under the following conditions and assumptions:

- In general, email-based applications are not required to be as fast as interactive applications such as web access.
- To test the system, the Italian Registration Authority (RA) set up its own CA and issued certificates only for a restricted set of ISPs, thus employing a low load of the MV system.

[3] The security protection provided by OpenVMS Operating System has been evaluated by the National Computer Security Center, receiving a C2 rating.

[4] Direct contract with final users (domain name requesters) are possible, but discouraged.

To evaluate system performance we undertook a real test in our operating environment. The test was split into two parts: verification, and message delivery to the software for automatic elaboration. In operating system terminology, a job unit can involve the execution of one or more modules. In our case the job includes all message elaboration between its arrival in the system and its exit. A script was set up to create and send a load of 2000 signed messages to the system.

Test results show that better performance can be achieved by activating a single job in each node of the cluster, with the total execution time of 1hour and 4 minutes. The activation of more jobs on the same node (for example, multiple processes) does not produce additional advantages, but provokes a slight increase in total execution time, probably due to competition of processes for access to resources. The average number of domain registrations per month in 2003 was 15,573.

Considering that our test was performed with the cluster fully operational, the results show that the system should be able to process the current monthly load in less than 9 hours. This implies that we do not need to invest in expensive cryptographic hardware and software.

## Summary

The Message Verify system was developed to permit automatic elaboration of digital signature in a flow of messages. Our experimentation revealed the following numerous advantages:

- Increased quality of service. Digital signature offers strong authentication (compared to the weak "user and password" mechanism), data integrity, and nonrepudiation of origin.
- Increased efficiency of service compared to signature verification using an email client.
- Transparency of service for RA operators. The service is user-friendly and training is not required.
- Reliability of the verification process and document management.
- Low cost, using free cryptographic software.

As an additional benefit, the use of certificates and secure mail has permitted users to become familiar with these technologies, which have wider application fields. Although the proposed solution implies the initial cost of certificate creation and dissemination, in the future, when all citizens have their own certificate to access numerous secure online services, this kind of tool can be used to obtain strong authentication of sensitive services (for example, accessing critical resources) directed to a large audience, such as that offered by public administrations and health departments.

## References

Berson, T. et al, 2001. Cryptography as a network service. Proceedings of Eighth Annual Network and distributed System Security Symposium. San Diego, USA. http://www.isoc.org/isoc/conferences/ndss/01/2001/INDEX.HTM.

Blaze, M. 1996. High-bandwidth encryption with low-bandwidth smartcards. Proceedings of the Fast Software Encryption Workshop. Lecture Notes in Computer Science. No. 1039, pp 33-40.

Blaze, M. et al, 1998. A formal treatment of remotely keyed encryption. Proceedings of EUROCRYPT98. Lecture Notes in Computer Science. No. 14039, pp 251-265.

Ding, X. et al, 2002. Experimenting with Server-Aided Signature. Proceedings of Ninth Annual Network and distributed System Security Symposium. San Diego, USA. http://www.isoc.org/isoc/conferences/ndss/02/proceedings/index.htm

Jakobsson, M. and Wetzel S., 2001. Secure Server-Aided Signature Generation. Proceedings of the Public Key Cryptography Conference, Cheju Island, Korea, pp. 383-401.

Lucks, S. 1999. Accelerated remotely keyed encryption. Proceedings of the Fast Software Encryption Workshop. Lecture Notes in Computer Science. No. 1636, pp 112-123.

Lucks, S. 1997. On the security of remotely keyed encryption. Proceedings of the Fast Software Encryption Workshop. Haifa, Israel, pp 219-229.

Weis, R. et al, 2000: Remotely Keyed Encryption with Java Cards: A Secure and Efficient Method to Encrypt Multimedia Streams. Proceedings of the IEEE International Conference on Multimedia and Expo (ICME). NY, USA, pp. 537-540.