Foreword

IT security is mandatory for everybody. While security concerns had been widely addressed in network related subsystems, archiving software, on the contrary, is often only locked beneath perimetral defenses.

We do believe that archived data deserve special care: documents and data base info should be accessed only at the right level of authorization, and nobody, notably internal people like system administrators, should be able to bypass the security wall provided by the application itself. MOSTcrypt, an optional module of the MOSTdoc archiving system, tries to address these needs, not only providing row (field, actually) level encryption on the database and on the documents, but also using public key (RSA) techniques whenever possible.

Smart Cards, Hardware Security Modules (HSMs), or even a pin protected keystore on the filesystem are the infrastructure available on the server and on the Web client, on which MOSTcrypt is built upon.

The same technology is used for different purposes:

- Database and document encryption
- Document signing
- Strong Authentication

MOSTcrypt Structure

Crypt features are available both on the server and on the client side.

On the MOSTdoc server there is the full range of routines that deal with Certification Authorities, Time Stamping Authorities, HSM signing, pkcs7 file handling and so on. Many activities can be safely performed on the server

On the client side we provide a complete implementation, too.

Database and document Encryption

First of all, every application record in MOSTdoc (i.e. every table row holding data and pointers to one or more documents), can have one or more ACL records. ACL stands for Access Control List, and each ACL record holds the rights granted to a user (or a group of users) for a particular database row.

ACL records are also used to host signature data

In order to provide our best, security features are available both on the server and on the client.

On the client, a java applet is able to interact with the smart card, and to perform basic tasks, like computing a hash, and signing it. Data can now safely travel towards the server, where a complete implementation of RSA functionalities is able to verify the data and archive it in the best way.

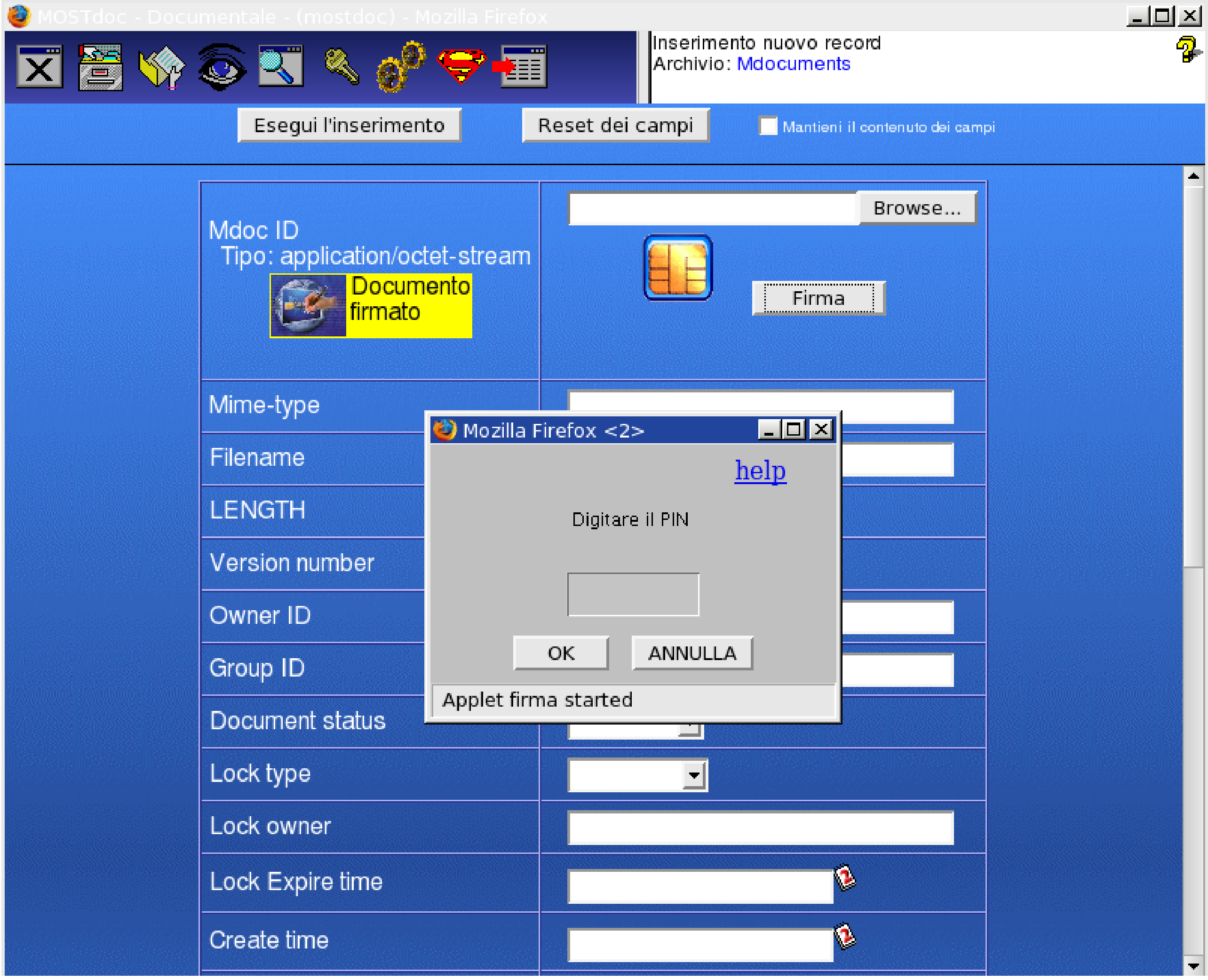
This kind of security is applied even on the data base fields, that are DES encrypted, and then crypted with the public key of the client

An additional signature is performed using an escrow key. This allows a disaster recovery of the data, provided that the escrow private key is available.

Access grants to the the record (or the document) can be added or revoked, using a web interface

Document signature

Document signing is available both as a standalone program, or just using an applet that interacts with the certificates and with the MOSTdoc server


MOSTdoc - Documentale - (mostdoc) - Mozilla Firefox
Esegui l'inserimento
Reset dei campi
Mantieni il contenuto dei campi
Browse...
Firma
Mozilla Firefox <2>
help
OK
ANNULLA

Inserimento nuovo record
Archivio: Mdocuments

Mdoc ID
Tipo: application/octet-stream
Documento firmato

Mime-type

Filename

LENGTH

Version number

Owner ID

Group ID

Document status

Lock type

Lock owner

Lock Expire time

Create time

Digitare il PIN

Applet firma started

Authentication

MOSTdoc users can be authenticated using a simple applet that interacts with the smart card (or the java keystore), and sends an encrypted message to the server, where a Public Key Infrastructure is used to verify the challenge string

Legal storage

According to the Italian law, some official documents can be stored in a digital way, destroying the original paper. The process, named ***Conservazione Sostitutiva,*** computes the hash of every file, and signs with one or more certificates the file holding the hashes and the metadata of the documents. Timestamping the signed file ends the process. These items are usually stored in a worm device, but the same level of trust can be obtained from a standard ISO 9660 image on the file system. Tamperproof confidence is based on the process that produces the information, not on the media that holds the information.

This process can be fully automated, accessing a HSM from the server, or can be integrated with a manual signature, using a smart card on the client side. the latter is useful also when a double signature is needed.