

Creating a PKI infrastructure with OpenSSL

Written by *Iarno Pagliani* (ipagliani@homeworks.it) and *Alessandro Tani* (atani@homeworks.it)

- Published 14 September 2008 -

Computer security is a strategic factor in today day, every system and network administrator should, sooner or later, compare with it. This article will try to explain how to implement a [PKI](#) infrastructure with the program [OpenSSL](#) of [Debian](#) (Etch). Explain how to create digital certificates to secure communications with the SMTP, IMAP, POP3 and HTTP protocols, then we will see how to implement a hierarchy of Certification Authority (CA) and how to assign a digital certificate to a person so that he can digitally sign their e-mail messages and send encrypted messages.

Table of Contents

- [License](#)
- [Acronyms used](#)
- [PKI with OpenSSL](#)
 - [Generating Root Certification Authority with OpenSSL](#)
 - [Generation of certificates of HomeWorks Issuing CA](#)
 - [Installation of digital certificates in Firefox and Thunderbird](#)
 - [CRL generation of HomeWorks Root CA](#)
 - [Creating the HomeWorks Issuing CA](#)
 - [CRL generation of HomeWorks Issuing CA](#)
 - [How to enable the control of CRL in Firefox and Thunderbird](#)
 - [Create the digital certificate and private key of Postfix](#)
 - [Generate the digital certificate and private key of Courier](#)
 - [Generate the digital certificate and private key of Apache](#)
 - [Generate the digital certificate to sign an email message](#)
 - [Installation of certificates PKCS#12 on Thunderbird](#)
- [Conclusions](#)
- [Web Reference and Bibliography](#)

License

The article **Creating a PKI infrastructure with OpenSSL** written by *Alessandro Tani* e *Iarno Pagliani* is protected by license [Creative Commons Attribution-Noncommercial-Share Alike 2.5 Italy License](#)..

Acronyms used

During the reading of the article or publications that speak of Public Key Infrastructure, you may experience the following acronyms:

Acronyms	Description
AA	Attribute Authority
ABA	American Bar Association Digital Signature Guidelines
AIA	Authority Information Access
ASN	Abstract Syntax Notation One
CA	Certification Authority
CMC	Certificate Management Messages over CMS
CMS	Cryptographic Message Syntax
CPS	Certification Practice Statement
CRL	Certificate Revocation List
CSP	Cryptographic Service Provider
CSR	Certificate Signing Request
DER	Distinguished Encoding Rules
DSA	Digital Signature Algorithm
EDI	Electronic Data Interchange
LRA	Local Registration Authority
HSM	Hardware Security Module
IPRA	Internet Policy Registration Authority
ISP	Internet Service Provider

Acronyms	Description
OCSP	Online Certificate Status Protocol
OID	Object Identifier
PEM	Internet Privacy Enhanced Mail
PCA	Policy Certification Authorities
PKC	Public Key Certificate
PKI	Internet Public Key Infrastructure
RA	Registration Authority

PKI with OpenSSL

During this article we will see how to implement a PKI infrastructure based on the program OpenSSL of the Debian (Etch) distribution. To make it easier our exposure, we suppose to create a PKI infrastructure for a company called Home Works S.p.A. This company is based in Reggio Emilia, Italy, has an IT Department and is conducting its first internal PKI infrastructure. The purpose of this PKI infrastructure is:

- create digital certificates that will be used by people of company, to sign and encrypt their e-mail messages;
- create digital certificates to enable secure communications with the SMTP ([Postfix](#)), IMAP, POP3 ([Courier](#)) and HTTP ([Apache](#)) protocols.

Since the company Home Works SpA has a single headquarters of great size and has no foreign locations, we create a PKI structure based only on two levels, the **Root CA** (or rather *HomeWorks Root CA*) and only one subordinate CA, the **Issuing CA** (or rather *HomeWorks Issuing CA*). The **Root CA**, once created the digital certificate for the **Issuing CA**, will be turned off and will be turned on only once time a year to create its Certificate Revocation List (CRL). This Certificate Revocation List will be available at URL http://www.homeworks.it/crl/root_ca.crl.

The digital certificate of the **Root CA** will be valid for sixteen years (for certificate longer than eight years, you should use keys of length equal to or exceeding the 2048 bits), while the digital certificate of the **Issuing CA** will be valid for eight years. The digital certificates issued by the **Issuing CA** will have the following duration:

- digital certificates issued to applications or special accounts, will have a duration of four years;
- digital certificates issued to people, will have a duration of one year.

The CRL of **Issuing CA** will be valid for thirty days, so will be updated at least once time every month. The CRL of **Issuing CA** will be made available to the URL http://www.homeworks.it/crl/issuing_ca.crl

For conventions adopted by the authors, all files generated by the OpenSSL program will have extension **.pem**, to remember that these files are formatted according to the standard **PEM**. All public keys not yet signed digitally, will have the suffix **public_key_req**. All digital certificates (or public keys digitally

signed), will have the suffix **public_cert**. All private keys, will have the suffix **private_key**.

Generating Root Certification Authority with OpenSSL

First at all, we install, if it were not already installed, the program OpenSSL:

```
apt-get install openssl
```

Before proceeding, however, with the creation of HomeWorks Root CA, we must define the HomeWorks Root CA and HomeWorks Issuing CA, *Certification Practice Statement*. In other words, we should on the one hand indicate a document which reported the purpose of the certificates provided by HomeWorks Root CA and the HomeWorks Issuing CA, on the other require a [Object Identifier](#) to identify unambiguously the certificates provided by HomeWorks Root CA and the HomeWorks Issuing CA (for more information about the Certification Practice Statement, you can see [RFC3647](#)).

To obtain a [Object Identifier](#) with which uniquely identify digital certificates of HomeWorks Root and Issuing CA, we may do a request to the [IANA](#), filling out a [web application form](#). In the case of Home Works S.p.A. were provided the following data:

- **Organization Name:** Home Works S.p.A.
- **Organization Address:**
Via Max Born, 28
42100 Reggio Emilia (RE)
Emilia Romagna
Italy
- **Organization Phone:** 0522327124
- **Contact Name:** Tani Alessandro
- **Contact Address:**
Via Max Born, 28
42100 Reggio Emilia (RE)
Emilia Romagna
Italy
- **Contact Phone:** 0522327124
- **Contact Fax:**
- **Contact Email:** support@homeworks.it

To receive a [Object Identifier](#) it takes up to 60 days (although usually the application is processed in about one week). Once you got the object identifier, you should register the code on the site [www.oid-info.com](#) in this way will be easily accessible by those who are seeking information about the owner of object identifier. Home Works S.p.A. has obtained the object identifier: [1.3.6.1.4.1.31012](#). This code remains associated with the branch of the [Registration-Hierarchical-Name-Tree](#) corresponding to Home Works S.p.A. From the object identifier [1.3.6.1.4.1.31012](#) are then generated sub-identifiers that are the mission to define *Certificate Policy* on the various digital certificates, signed by HomeWorks Root CA and by HomeWorks Issuing CA. For choice of the authors of this article, it was decided to create the following sub-code:

- **1.3.6.1.4.1.31012.1.1** refers to the **Certification Practice Statement**;
- **1.3.6.1.4.1.31012.1.2** refers to the policies for management of digital certificates issued by Certification Authority (CA), or, in our example, on certificates provided by HomeWorks Root CA or by HomeWorks Issuing CA;
- **1.3.6.1.4.1.31012.2.1** refers to the policies for managing digital certificates used by people to digitally sign their e-mail messages;
- **1.3.6.1.4.1.31012.2.2** refers to the policies for managing digital certificates provided to people who need to digitally sign electronic documents;
- **1.3.6.1.4.1.31012.3.1** refers to the policies for managing digital certificates used by mail servers;

- **1.3.6.1.4.1.31012.3.2** refers to policies for managing digital certificates used by Web server, for secure communications between the client and the Web server itself;

At each of these identification codes will remain associated a document (in HTML format) explaining in detail the various policies with which digital certificates are generated and issued to people or applications.

The OpenSSL package of Debian provides a Perl script, `/usr/lib/ssl/misc/CA.pl`, to generate easily a Certification Authority. The script `/usr/lib/ssl/misc/CA.pl` refers to the configuration file `/etc/ssl/openssl.cnf` to proceed with the generation of the Certification Authority. The file `/etc/ssl/openssl.cnf` in the OpenSSL package, is an example, but just only an example, though fully functional, how should create both public and private keys, both certificates. To be able to create a genuine CA, we should modify the file `/etc/ssl/openssl.cnf`:

```
cp /etc/ssl/openssl.cnf /etc/ssl/openssl.cnf.default
vi /etc/ssl/openssl.cnf
```

We edit [/etc/ssl/openssl.cnf](#) as follows:

```
# File /etc/ssl/openssl.cnf
#
# Environment Settings
HOME          = .
RANDFILE      = $ENV::HOME/.rnd

#####
## Configuration Sections ##
#####

# OID Section
oid_section   = new_oids

# New OID for certificate
# For more information about OID visit the site:
# http://www.alvestrand.no/objectid/index.html
[ new_oids ]
# Short Name      = OID Number Code
HW-CPS            = 1.3.6.1.4.1.31012.1.1 # Certification Practice Statement
HW-CA-Cert        = 1.3.6.1.4.1.31012.1.2 # Subordinate CA Certificate
HW-MAIL-Cert      = 1.3.6.1.4.1.31012.2.1 # Mail Certificate
HW-CODE-Cert      = 1.3.6.1.4.1.31012.2.2 # Code Signature Certificate
HW-TLS-MAIL-Cert  = 1.3.6.1.4.1.31012.3.1 # Secure Communications Mail Ser
# Certificate
HW-TLS-WEB-Cert   = 1.3.6.1.4.1.31012.3.2 # Secure Communications Web Server
# Certificate

# Certificate Authority Section
[ ca ]
default_ca        = CA_default                # The default CA section

# Default CA configuration to sign a Certificate (Public Key)
[ CA_default ]
dir               = $ENV::CADIR                # Where everything is kept
certs             = $dir/certs                 # Where the issued certs are kept
crl_dir           = $dir/crl                   # Where the issued CRL are kept
database          = $dir/index.txt            # Database index file
unique_subject    = no # Allow creation of several ctificates with same
# subject
new_certs_dir     = $dir/newcerts              # Default place for new certs.
certificate        = $dir/root_ca_public_cert.pem # The CA Certificate (Public
# Key)
serial            = $dir/serial                 # The current serial number
crlnumber         = $dir/crlnumber             # CRL Serial Number
crl               = $dir/crl/root_ca.crl       # The current CRL
```

```

private_key          = $dir/private/root_ca_private_key.pem # The CA Private key
RANDFILE             = $dir/private/.rand                  # Private random number file
x509_extensions      = sub_ca_cert # The extensions to add to the certificate
                                # (Public Key)
name_opt             = ca_default      # Subject Name options
cert_opt             = ca_default      # Certificate field options
crl_extensions       = crl_ext # Certificate Revocation List (CRL) extensions
default_days         = 2920           # How long to certify for (8 years)
default_crl_days     = 365            # How long before next CRL (1 year)
default_md            = sha1           # Which Hash Functions to use
preserve             = no              # Keep passed DN ordering
policy               = policy_match    # Policy for CA only (this can be
                                # overridden by the "-policy" option)

# Extensions to add when Root CA creates an Subordinate Certificate CA (Public Key)
[ sub_ca_cert ]
basicConstraints      = CA:true
keyUsage              = critical, cRLSign, keyCertSign
subjectKeyIdentifier = hash
authorityKeyIdentifier = keyid, issuer
authorityInfoAccess   = caIssuers;URI:http://www.homeworks.it/ca/cainfo.html
crlDistributionPoints = URI:http://www.homeworks.it/crl/root_ca.crl
certificatePolicies   = ia5org,@HomeWorks_CPS,@HomeWorks_CA_policy

[ HomeWorks_CPS ]
policyIdentifier = HW-CPS
CPS.1           = "http://www.homeworks.it/ca/homeworks_cps.html"
userNotice.1    = @HomeWorks_CPS_Notice

[ HomeWorks_CPS_Notice ]
explicitText = "Home Works S.p.A. Certification Practice Statement"

[ HomeWorks_CA_policy ]
policyIdentifier = HW-CA-Cert
userNotice.2    = @HomeWorks_CA_Notice

[ HomeWorks_CA_Notice ]
explicitText = "Home Works S.p.A. CA Certificate Policy"

# CRL extensions
[ crl_ext ]
crlDistributionPoints = URI:http://www.homeworks.it/crl/root_ca.crl

# Requirement for a new Private Key
[ req ]
dir          = $ENV::CADIR # Where everything is kept
default_bits = 2048 # This specifies the default key size in bits
default_keyfile = $dir/private/new_private_key.pem # Default new Private
                                                    # Key file name (this
                                                    # can be

distinguished_name = req_distinguished_name # Distinguished Name of the
                                                    # subject of the certificate
attributes         = req_attributes
x509_extensions    = v3_ca

# Challenge password section
[ req_attributes ]
challengePassword      = A challenge password (between 6 and 20 characters)
challengePassword_min  = 6
challengePassword_max  = 20

# Version 3 certificate extensions for a new Root CA Certificate Self Signed
# (Public Key)
[ v3_ca ]
basicConstraints      = CA:true
keyUsage              = critical, cRLSign, keyCertSign
subjectKeyIdentifier = hash
authorityKeyIdentifier = keyid:always, issuer:always
authorityInfoAccess   = caIssuers;URI:http://www.homeworks.it/ca/cainfo.html

```

```

crlDistributionPoints    = URI:http://www.homeworks.it/crl/root_ca.crl
certificatePolicies      = ia5org,@HomeWorks_CPS

# Distinguished Name of the certification authority
[ req_distinguished_name ]
0.organizationName       = Organization Name (eg, company)
0.organizationName_default = Home Works S.p.A.
1.organizationName       = Internet Company Web Site
1.organizationName_default = http://www.homeworks.it
organizationalUnitName    = Organizational Unit Name (eg, section)
organizationalUnitName_default = HomeWorks IT Department
commonName               = Certification Authority Name (Common Name)
commonName_default       = HomeWorks Root CA
commonName_max           = 64
emailAddress             = Email Address (max 64 characters)
emailAddress_default     = support@homeworks.it
emailAddress_max         = 64
localityName             = Locality Name (eg, city)
localityName_default     = Reggio Emilia
countryName              = Country Name (2 letter code)
countryName_default      = IT
countryName_min          = 2
countryName_max          = 2
stateOrProvinceName      = State or Province Name (full name)
stateOrProvinceName_default = Italy
# SET-ex3                = SET extension number 3

#####
## Policy Sections ##
#####

# For the CA only
[ policy_match ]
organizationName      = match
organizationalUnitName = match
commonName            = supplied
emailAddress          = optional
localityName          = optional
stateOrProvinceName   = match
countryName           = match

# For every certificate (Public Key)
[ policy_anything ]
organizationName      = optional
organizationalUnitName = optional
commonName            = supplied
emailAddress          = optional
localityName          = optional
stateOrProvinceName   = optional
countryName           = optional

# End File

```

More generally, the lines:

```

...
HW-CPS          = 1.3.6.1.4.1.31012.1.1 # Certification Practice Statement
HW-CA-Cert      = 1.3.6.1.4.1.31012.1.2 # Subordinate CA Certificate
HW-MAIL-Cert    = 1.3.6.1.4.1.31012.2.1 # Mail Certificate
HW-CODE-Cert    = 1.3.6.1.4.1.31012.2.2 # Code Signature Certificate
HW-TLS-MAIL-Cert = 1.3.6.1.4.1.31012.3.1 # Secure Communications Mail Server
# Certificate
HW-TLS-WEB-Cert = 1.3.6.1.4.1.31012.3.2 # Secure Communications Web Server
# Certificate
...
authorityInfoAccess = caIssuers;URI:http://www.homeworks.it/ca/cainfo.html

```

```

crlDistributionPoints = URI:http://www.homeworks.it/crl/root_ca.crl
policyIdentifier      = ...
CPS.1                = "http://www.homeworks.it/ca/homeworks_cps.html"
explicitText          = "Home Works S.p.A. Certification Practice Statement"
explicitText          = "Home Works S.p.A. CA Certificate Policy"
...
crlDistributionPoints = URI:http://www.homeworks.it/crl/root_ca.crl
...
stateOrProvinceName_default = Italy
localityName_default        = Reggio Emilia
0.organizationName_default  = Home Works S.p.A.
organizationalUnitName_default = HomeWorks IT Department
commonName_default          = HomeWorks Root CA
emailAddress_default        = support@homeworks.it
...

```

should be changed, with information on the particular company which is in the process of achieving PKI infrastructure.

For simplicity we will assume that the user who will administer the CA is the user `root` (it should be noted that given the delicate role of the servers that host the CA, delegate administrative tasks to a user who is not `root` is a better choice), therefore we modify the file `/root/.profile` (settings that are going to illustrate are valid, however, for any user):

```
vi /root/.profile
```

We add the following environment variables relating to Home Works Root CA:

```

# CA Settings
CADIR=/usr/lib/ssl/misc/CA
OPENSSL_CONF=/etc/ssl/openssl.cnf
export CADIR OPENSSL_CONF

```

We enable the new configuration of the `/root/.profile` file:

```
source /root/.profile
```

By default, the command `/usr/lib/ssl/misc/CA.pl` creates *public keys* that are valid for three years. Three years may seem a reasonable value in many circumstances, but to avoid creating unnecessary administrative burdens, we change the file `/usr/lib/ssl/misc/CA.pl` so as to create digital certificates for a period of sixteen years (that is, 5840 days):

```

cp /usr/lib/ssl/misc/CA.pl /usr/lib/ssl/misc/CA.pl.default
chmod 644 /usr/lib/ssl/misc/CA.pl.default
vi /usr/lib/ssl/misc/CA.pl

```

The following lines:

```

$CADAYS="-days 1095";    # 3 years
$CATOP="./demoCA";
$CAKEY="cakey.pem";
$CAREQ="careq.pem";
$CACERT="cacert.pem";

```

should become respectively:

```
$CADAYS="-days 5840";    # 16 years
```

```
$CATOP="./CA";
$CAKEY="root_ca_private_key.pem";
$CAREQ="root_ca_public_key_req.pem";
$CACERT="root_ca_public_cert.pem";
```

We make a copy of the file `/usr/lib/ssl/misc/CA.pl` if, during the process of updating the OpenSSL package, can be accidentally overwritten:

```
cp /usr/lib/ssl/misc/CA.pl /usr/lib/ssl/misc/CA.pl.backup
chmod 644 /usr/lib/ssl/misc/CA.pl.backup
```

We are ready to create the Home Works Root CA (having altered in a way appropriate the file `/etc/ssl/openssl.cnf`, for many fields relating to the creation of HomeWorks Root CA public and private key pairs, you can leave the default values). If nothing is specified within the various fields, then you will be left the default value, which is listed in the square brackets. To leave the default value you should press the *Enter* button:

```
cd /usr/lib/ssl/misc/
./CA.pl -newca

CA certificate filename (or enter to create) <-- Press Enter
Making CA certificate ...
Generating a 2048 bit RSA private key
.....+++
.....+++
writing new private key to './CA/private/root_ca_private_key.pem'
Enter PEM pass phrase: homeworks
Verifying - Enter PEM pass phrase: homeworks
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Organization Name (eg, company) [Home Works S.p.A.]: Home Works S.p.A.
Internet Company Web Site [http://www.homeworks.it]: http://www.homeworks.it
Organizational Unit Name (eg, section) [HomeWorks IT Department]: HomeWorks IT Department
Certification Authority Name (Common Name) [HomeWorks Root CA]: HomeWorks Root CA
Email Address (max 64 characters) [support@homeworks.it]: support@homeworks.it

Locality Name (eg, city) [Reggio Emilia]: Reggio Emilia
State or Province Name (full name) [Italy]: Italy
Country Name (2 letter code) [IT]: IT

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password (between 6 and 20 characters) []: homeworks
Using configuration from /etc/ssl/openssl.cnf
Enter pass phrase for ./CA/private/root_ca_private_key.pem: homeworks
Check that the request matches the signature
Signature ok
Certificate Details:
  Serial Number:
    ed:c9:28:8f:fa:00:58:6e
  Validity
    Not Before: May 18 13:49:58 2008 GMT
    Not After : May 14 13:49:58 2024 GMT
  Subject:
    organizationName = Home Works S.p.A.
    organizationName = http://www.homeworks.it
    organizationalUnitName = HomeWorks IT Department
    commonName = HomeWorks Root CA
    emailAddress = support@homeworks.it
    localityName = Reggio Emilia
    stateOrProvinceName = Italy
    countryName = IT
  X509v3 extensions:
    X509v3 Basic Constraints:
```

```

CA:TRUE
X509v3 Key Usage: critical
Certificate Sign, CRL Sign
X509v3 Subject Key Identifier:
09:30:BB:26:5A:05:C3:83:6E:DE:47:4E:FF:50:2C:23:0B:44:C8:D0
X509v3 Authority Key Identifier:
keyid:09:30:BB:26:5A:05:C3:83:6E:DE:47:4E:FF:50:2C:23:0B:44:C8:D0
DirName:/C=IT/ST=Italy/O=Home Works S.p.A./O=http://www.homeworks.it/OU=HomeWorks IT
Department/CN=HomeWorks Root CA/emailAddress=support@homeworks.it
serial:ED:C9:28:8F:FA:00:58:6E

Authority Information Access:
CA Issuers - URI:http://www.homeworks.it/ca/cainfo.html

X509v3 CRL Distribution Points:
URI:http://www.homeworks.it/crl/root_ca.crl

X509v3 Certificate Policies:
Policy: HW-CPS
CPS: http://www.homeworks.it/ca/homeworks_cps.html
User Notice:
Explicit Text: Home Works S.p.A. Certification Practice Statement

Certificate is to be certified until May 14 13:49:58 2024 GMT (5840 days)

Write out database with 1 new entries
Data Base Updated

```

In the compilation of these fields, please note that the field **Certification Authority Name (Common Name)** can not contain values string longer than 64 characters. Since this is a Root CA, the field **Common Name** should contain a value that describe the role of the CA, in our example: **HomeWorks Root CA**. That is the Root CA of the Home Works S.p.A company.

We check that the folder `/usr/lib/ssl/misc/CA` has been created:

```

ls -al /usr/lib/ssl/misc

drwxr-xr-x 6 root root 4096 2008-03-22 22:46 CA
-rwxr-xr-x 1 root root 5875 2007-03-22 22:41 CA.pl
-rw-r--r-- 1 root root 5872 2008-02-05 23:35 CA.pl.backup
-rw-r--r-- 1 root root 5875 2008-02-05 22:44 CA.pl.default
-rwxr-xr-x 1 root root 3758 2007-09-28 22:49 CA.sh
-rwxr-xr-x 1 root root 119 2007-09-28 22:49 c_hash
-rwxr-xr-x 1 root root 152 2007-09-28 22:49 c_info
-rwxr-xr-x 1 root root 112 2007-09-28 22:49 c_issuer
-rwxr-xr-x 1 root root 110 2007-09-28 22:49 c_name

```

If there is the folder `/usr/lib/ssl/misc/CA` means that the process of creating the CA is successful. The folder `/usr/lib/ssl/misc/CA` has the following structure:

```

tree CA/

CA/
|-- certs
|-- crl
|-- crlnumber
|-- index.txt
|-- index.txt.attr
|-- index.txt.old
|-- newcerts
|  |-- EDC9288FFA00586E.pem
|-- private
|  |-- root_ca_private_key.pem
|-- root_ca_public_cert.pem
|-- root_ca_public_key_req.pem
'-- serial

4 directories, 9 files

```

Where:

- `/usr/lib/ssl/misc/CA/root_ca_public_cert.pem` is the *digital certificate* of the HomeWorks Root CA.
- `/usr/lib/ssl/misc/CA/root_ca_public_key_req.pem` is the *public key* of the HomeWorks Root CA (or CSR).
- `/usr/lib/ssl/misc/CA/private/root_ca_private_key.pem` is the *private key* of the HomeWorks Root CA.
- The folder `/usr/lib/ssl/misc/CA/certs` contains all *certificates* provided by the HomeWorks Root CA.
- The folder `/usr/lib/ssl/misc/CA/private` contains all *private keys* of the various certificates provided.
- The folder `/usr/lib/ssl/misc/CA/crl` contains the updated copy of the *Certificate Revocation List* of the HomeWorks Root CA.

We fix the permissions of the folder `$CADIR/private`:

```
chmod g-rwx,o-rwx $CADIR/private
```

We monitor that the *digital certificate* of CA has been generated correctly:

```
openssl x509 -text -noout -in $CADIR/root_ca_public_cert.pem
```

We monitor that the *private key* of CA has been generated correctly:

```
openssl rsa -noout -text -in $CADIR/private/root_ca_private_key.pem
```

We convert the public key of CA in the format compatible with Windows systems and Mac OS X:

```
openssl x509 -in $CADIR/root_ca_public_cert.pem -out  
$CADIR/root_ca_public_cert_windows_format.der -outform DER
```

At this point, on every workstations that will use certificates issued by HomeWorks Root CA, should be imported the certificate `/usr/lib/ssl/misc/CA/root_ca_public_cert_windows_format.der` if the workstations has Windows or Mac OS X operating system. These digital certificates, which identify the HomeWorks Root CA, will be available to the URL <http://www.homeworks.it/ca/cainfo.html>.

Before proceeding with the creation of HomeWorks Issuing CA's public/private key pairs, we create the folders `/usr/lib/ssl/misc/CA/ext` and `/usr/lib/ssl/misc/CA/request`. The folder `/usr/lib/ssl/misc/CA/ext` will be used to contain all *extensions* to the configuration file `/etc/ssl/openssl.cnf`, while the folder `/usr/lib/ssl/misc/CA/request` will be used to contain all new public keys which will be digitally signed (CSR):

```
md $CADIR/ext  
md $CADIR/request
```

Generation of certificates of HomeWorks Issuing CA

Once the HomeWorks Root CA is operational, we can proceed with the creation of HomeWorks Issuing CA's certificates. We start with the creation of HomeWorks Issuing CA's public/private key pairs (`req` option allows the creation of public keys comply with the specific [PKCS#10 X.509 Certificate Signing Request](#)):

```
openssl req -new -nodes -keyout $CADIR/private/issuing_ca_private_key.pem -out
```

```

$CADIR/request/issuing_ca_public_key_req.pem

Generating a 2048 bit RSA private key
.....+++
...+++
writing new private key to '/usr/lib/ssl/misc/CA/private/issuing_ca_private_key.pem'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Organization Name (eg, company) [Home Works S.p.A.]: Home Works S.p.A.
Internet Company Web Site [http://www.homeworks.it]: http://www.homeworks.it
Organizational Unit Name (eg, section) [HomeWorks IT Department]: HomeWorks IT Department
Certification Authority Name (Common Name) [HomeWorks Root CA]: HomeWorks Issuing CA
Email Address (max 64 characters) [support@homeworks.it]: support@homeworks.it
Locality Name (eg, city) [Reggio Emilia]: Reggio Emilia
Country Name (2 letter code) [IT]: IT
State or Province Name (full name) [Italy]: Italy

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password (between 6 and 20 characters) []: homeworks

```

The option `-nodes` allow to do not specify the *pass phrase* with which then use the certificate, in this way it will be easier to create the CRL associated with HomeWorks Issuing CA and more generally certificates issued by this CA.

We monitor that the public key of Issuing CA, `$CADIR/request/issuing_ca_public_key_req.pem`, has been created correctly:

```
openssl req -text -noout -in $CADIR/request/issuing_ca_public_key_req.pem
```

We create the certificate to be assigned to HomeWorks Issuing CA:

```

openssl ca -policy policy_anything -out $CADIR/certs/issuing_ca_public_cert.pem -infiles
$CADIR/request/issuing_ca_public_key_req.pem

Using configuration from /etc/ssl/openssl.cnf
Enter pass phrase for /usr/lib/ssl/misc/CA/private/root_ca_private_key.pem:
Check that the request matches the signature
Signature ok
Certificate Details:
  Serial Number:
    ed:c9:28:8f:fa:00:58:6f
  Validity
    Not Before: May 18 14:13:20 2008 GMT
    Not After : May 16 14:13:20 2016 GMT
  Subject:
    organizationName = Home Works S.p.A.
    organizationName = http://www.homeworks.it
    organizationalUnitName = HomeWorks IT Department
    commonName = HomeWorks Issuing CA
    emailAddress = support@homeworks.it
    localityName = Reggio Emilia
    stateOrProvinceName = Italy
    countryName = IT
  X509v3 extensions:
    X509v3 Basic Constraints:
      CA:TRUE
    X509v3 Key Usage: critical
      Certificate Sign, CRL Sign
    X509v3 Subject Key Identifier:
      CC:A7:3D:F0:35:F0:83:8E:5A:1F:D0:67:AD:E9:63:95:5F:3C:C4:74
    X509v3 Authority Key Identifier:
      keyid:AA:E1:35:E1:5D:E3:FE:87:55:CB:56:AD:97:C5:73:72:D4:EE:CB:AE

  Authority Information Access:

```

```

CA Issuers - URI:http://www.homeworks.it/ca/cainfo.html

X509v3 CRL Distribution Points:
  URI:http://www.homeworks.it/crl/root_ca.crl

X509v3 Certificate Policies:
  Policy: HW-CPS
    CPS: http://www.homeworks.it/ca/homeworks_cps.html;
    User Notice:
      Explicit Text: Home Works S.p.A. Certification Practice Statement
  Policy: HW-CA-Cert
    User Notice:
      Explicit Text: Home Works S.p.A. CA Certificate Policy

Certificate is to be certified until Mar 24 23:13:29 2016 GMT (2920 days)
Sign the certificate? [y/n]: y

1 out of 1 certificate requests certified, commit? [y/n] y
Write out database with 1 new entries
Data Base Updated

```

We monitor that the digital certificate of HomeWorks Issuing CA has been generated correctly:

```
openssl x509 -text -noout -in $CADIR/certs/issuing_ca_public_cert.pem
```

We convert the public key of HomeWorks Issuing CA in the format compatible with Windows systems and Mac OS X:

```
openssl x509 -in $CADIR/certs/issuing_ca_public_cert.pem -out
$CADIR/certs/issuing_ca_public_cert_windows_format.der -outform DER
```

On all workstations stations that will use certificates issued by HomeWorks Issuing CA, should be imported the certificate `/usr/lib/ssl/misc/CA/certs/issuing_ca_public_cert.pem` if the workstations has Linux or Unix as its operating system; certificate `/usr/lib/ssl/misc/CA/certs/issuing_ca_public_cert_windows_format.der` if the workstations has Windows or Mac OS X operating system. These digital certificates, which identify the HomeWorks Issuing CA, will be available to the URL <http://www.homeworks.it/ca/cainfo.html>.

It should be noted that certificates issued by HomeWorks Issuing CA are considered *reliable*, that is worthy of trust, if on all workstations that will have to use certificates issued by HomeWorks Issuing CA, will be installed both the certificate of **HomeWorks Root CA** and the certificate of **HomeWorks Issuing CA**.

Installation of digital certificates in Firefox and Thunderbird

As an example to assign the certificates of **HomeWorks Root CA** and of **HomeWorks Issuing CA** to a workstation, we take a workstation with Windows XP Professional, [Firefox](#) and [Thunderbird](#) installed. In this case we will have to take versions of certificates of **HomeWorks Root CA** and of **HomeWorks Issuing CA** in DER format:

- `/usr/lib/ssl/misc/CA/root_ca_public_cert_windows_format.der` is the digital certificate of **HomeWorks Root CA**;
- `/usr/lib/ssl/misc/CA/certs/issuing_ca_public_cert_windows_format.der` is the digital certificate of **HomeWorks Issuing CA**.

We suppose that our Windows workstation sample has downloaded in the **C:\Certificates** the two certificates above, downloading from the site <http://www.homeworks.it/ca/cainfo.html>.

To charge the two certificates in Firefox you can just proceed as follows:

- Firefox start;
- open the **Tools** menu and select **Options**;
- open the section **Advanced** and then the **Encryption** subsection;
- make sure they are selected entries **Use SSL 3.0** and **Use TLS 1.0**;
- press the button **View Certificates**;
- go through **Authorities** tab;
- press the button **Import**;
- import before the file **C:\Certificates\root_ca_public_cert_windows_format.der** and then the file **C:\Certificates\issuing_ca_public_cert_windows_format.der**;
- press the **OK** button to confirm the changes made to **Authorities** tab;
- in the window entitled **Options**, press the **OK** button;
- if you wish, at this point you can close Firefox.

In this way Firefox will consider reliable all certificates issued by **HomeWorks Issuing CA**. Similarly, to charge the two digital certificates of **HomeWorks Root CA** and **HomeWorks Issuing CA** in Thunderbird, you can proceed as follows:

- Thunderbird start;
- open the **Tools** menu and select **Options**;
- open the section **Advanced** and then the subsection **Certificates**;
- press the button **View Certificates**;
- go through **Authorities** tab;
- press the button **Import**;
- import before the file **C:\Certificates\root_ca_public_cert_windows_format.der** and then the file **C:\Certificates\issuing_ca_public_cert_windows_format.der**;
- press the **OK** button to confirm the changes made to **Authorities** tab;
- in the window entitled **Options**, press the **OK** button;
- if you wish, at this point you can close Thunderbird.

In this way Thunderbird consider reliable all certificates issued by **HomeWorks Issuing CA**.

If the PKI infrastructure you are making is public, or you want the certificates issued by your Issuing CA are accessible to as many people as possible, without anyone should download the Issuing and Root CA certificates to resolve the **Certification Path**, you may decide to ask to enter the CA of your PKI infrastructure, in the list of CA within Firefox and Thunderbird. To enter your digital certificates within Firefox and Thunderbird CA list by default, you should ensure that your PKI infrastructure implemented, meet the requirements indicated in the document [Mozilla CA Certificate Policy](#).

CRL generation of HomeWorks Root CA

To complete the creation of HomeWorks Root CA, we have to generate the *Certificate Revocation List* (or more briefly *CRL*) associated with HomeWorks Root CA. To create the CRL, will create a digital certificate that will be then revoked. For simplicity we will call this certificate with the name of `$CADIR/certs/crl_public_cert.pem`. So, first we create the public/private key pairs:

```
openssl req -new -nodes -keyout $CADIR/private/crl_private_key.pem -out
$CADIR/request/crl_public_key_req.pem

Generating a 2048 bit RSA private key
.....+++
.+++
writing new private key to '/usr/lib/ssl/misc/CA/private/crl_private_key.pem'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
```

If you enter '.', the field will be left blank.

```
Organization Name (eg, company) [Home Works S.p.A.]: Home Works S.p.A.
Internet Company Web Site [http://www.homeworks.it]: http://www.homeworks.it
Organizational Unit Name (eg, section) [HomeWorks IT Department]: HomeWorks IT Department
Certification Authority Name (Common Name) [HomeWorks Root CA]: CRL of HomeWorks Root CA
Email Address (max 64 characters) [support@homeworks.it]: support@homeworks.it
Locality Name (eg, city) [Reggio Emilia]: Reggio Emilia
Country Name (2 letter code) [IT]: IT
State or Province Name (full name) [Italy]: Italy
```

Please enter the following 'extra' attributes
to be sent with your certificate request

A challenge password []:<-- **Premi Invio**

An optional company name []:<-- **Premi Invio**

Then we digitally sign the public key `$CADIR/request/crl_public_key_req.pem`:

```
openssl ca -policy policy_anything -out $CADIR/certs/crl_public_cert.pem -infiles
$CADIR/request/crl_public_key_req.pem
```

At the end we revoke the certificate `$CADIR/certs/crl_public_cert.pem`:

```
openssl ca -revoke $CADIR/certs/crl_public_cert.pem
Using configuration from /etc/ssl/openssl.cnf
Enter pass phrase for /usr/lib/ssl/misc/CA/private/root_ca_private_key.pem: homeworks
Revoking Certificate EDC9288FFA005870.
Data Base Updated
```

After this, we create the CRL:

```
openssl ca -gencrl -out $CADIR/crl/root_ca.crl
Enter pass phrase for /usr/lib/ssl/misc/CA/private/root_ca_private_key.pem: homeworks
```

We convert the CRL to DER format:

```
openssl crl -in $CADIR/crl/root_ca.crl -out $CADIR/crl/root_ca.crl -outform DER
```

At this point we have to make public the CRL, to make it accessible through the URL http://www.homeworks.it/crl/root_ca.crl The CRL will be generated once a year (as specified by the parameter `default_crl_days` of the configuration file `/etc/ssl/openssl.cnf`), possibly a week before its expiry. Since the HomeWorks Root CA will be largely off, not being involved in the generation of certificates, the task of creating a CRL will be performed in manual mode.

It should be noted that applications usually are not able to control, in automatic way, the CRL. We should therefore [configure in a timely manner](#) so that each application controls the CRL associated with the CA who provided the certificate (in our article, the CA, will be the HomeWorks Root CA and HomeWorks Issuing CA).

Rename the certificate `$CADIR/certs/crl_public_key_req.pem`, its public key `$CADIR/private/crl_public_cert.pem` and its corresponding private key, `$CADIR/private/crl_private_key.pem`, adding the suffix `.revoked`:

```
mv $CADIR/certs/crl_public_cert.pem $CADIR/certs/crl_public_cert.pem.revoked
mv $CADIR/request/crl_public_key_req.pem $CADIR/request/crl_public_key_req.pem.revoked
mv $CADIR/private/crl_private_key.pem $CADIR/private/crl_private_key.pem.revoked
```

Creating the HomeWorks Issuing CA

Once generated the private key of HomeWorks Issuing CA and its digital certificate, we can proceed to

finalize the HomeWorks Issuing CA. For obvious security reasons, it is good that the HomeWorks Issuing CA will be hosted on a server other than hosting the HomeWorks Root CA (in fact, the server hosting the HomeWorks Root CA should be switched off and only turned on to make or maintenance on the system, or generate the CRL of HomeWorks Root CA). The private key of HomeWorks Issuing CA and its digital certificate, should be moved, in a safe manner, from the server that hosting HomeWorks Root CA, to the server that hosting HomeWorks Issuing CA (usually the use of a USB stick should be more than good).

As done before for the HomeWorks Root CA, we suppose, for simplicity, that the administrator user of the HomeWorks Issuing CA will be `root` user (again what we said about HomeWorks Root CA is true about HomeWorks Issuing CA, given the delicacy of the role of CA, the administrative tasks of CA should be delegated to someone other than `root` user).

First we change the file `/root/.profile`:

```
vi /root/.profile
```

We add the following environment variables relating to HomeWorks Issuing CA:

```
# CA Settings
CADIR=/usr/lib/ssl/misc/CA
OPENSSL_CONF=/etc/ssl/openssl.cnf
export CADIR OPENSSL_CONF
```

We load the new configuration files `/root/.profile`:

```
source /root/.profile
```

On the server hosting the HomeWorks Issuing CA we should proceed to create the following structure folders and files:

```
/usr/lib/ssl/misc/CA
|-- root_ca_public_cert.pem
|-- root_ca_public_cert_windows_format.der
|-- issuing_ca_public_cert.pem
|-- issuing_ca_public_cert_windows_format.der
|-- certs
|-- crl
|-- crlnumber
|-- ext
|-- index.txt
|-- newcerts
|-- oid
|-- private
|  |-- issuing_ca_private_key.pem
|-- request
`-- serial
```

To create the structure indicated we proceed as follows:

```
md $CADIR
md $CADIR/certs
md $CADIR/crl
md $CADIR/ext
md $CADIR/newcerts
md $CADIR/oid
md $CADIR/private
md $CADIR/request
chmod g-rwx,o-rwx $CADIR/private
echo '84D2C38B199FEA83' > $CADIR/serial
```

```
echo '01' > $CADIR/crlnumber
> $CADIR/index.txt
```

To generate the random numbers to put in the file `$CADIR/serial` we used the function **Random Number Generator** of the site www.dnsstuff.com. We copy in the structure just created the files

`issuing_ca_private_key.pem`, `issuing_ca_public_cert.pem`, `root_ca_public_cert_windows_format.der`, `root_ca_public_cert.pem` e `issuing_ca_public_cert_windows_format.der`, taking care of:

- copy the file `issuing_ca_public_cert_windows_format.der` and `issuing_ca_public_cert.pem` in the folder `/usr/lib/ssl/misc/CA`;
- copy the file `issuing_ca_private_key.pem` in the folder `/usr/lib/ssl/misc/CA/private`
- copy the file `root_ca_public_cert_windows_format.der` and `root_ca_public_cert.pem` in the folder `/usr/lib/ssl/misc/CA`;

We join the digital certificate of HomeWorks Root CA with the digital certificate of HomeWorks Issuing CA, we get the *global certificate* of the PKI infrastructure realized:

```
cat $CADIR/root_ca_public_cert.pem $CADIR/issuing_ca_public_cert.pem >
$CADIR/global_ca_public_cert.pem
```

Just as we did for HomeWorks Root CA, we customize the file `/etc/ssl/openssl.cnf`:

```
cp /etc/ssl/openssl.cnf /etc/ssl/openssl.cnf.originale
vi /etc/ssl/openssl.cnf
```

We modify the file </etc/ssl/openssl.cnf> as follow:

```
# File /etc/ssl/openssl.cnf
#
# Environment Settings
HOME          = .
RANDFILE      = $ENV::HOME/.rnd

#####
##  Configuration Sections  ##
#####

# OID Section
oid_section   = new_oids

# New OID for certificate
# For more information about OID visit the site: http://www.alvestrand.no/objectid/
# index.html
[ new_oids ]
# Short Name          = OID Number Code
HW-CPS                = 1.3.6.1.4.1.31012.1.1 # Certification Practice Statement
HW-CA-Cert             = 1.3.6.1.4.1.31012.1.2 # Subordinate CA Certificate
HW-MAIL-Cert          = 1.3.6.1.4.1.31012.2.1 # Mail Certificate
HW-CODE-Cert          = 1.3.6.1.4.1.31012.2.2 # Code Signature Certificate
HW-TLS-MAIL-Cert      = 1.3.6.1.4.1.31012.3.1 # Secure Communications Mail Server
                        # Certificate
HW-TLS-WEB-Cert       = 1.3.6.1.4.1.31012.3.2 # Secure Communications Web Server
                        # Certificate

# Certificate Authority Section
[ ca ]
default_ca          = CA_default          # The default CA section

# Default CA configuration to sign a Certificate (Public Key)
[ CA_default ]
dir                 = $ENV::CADIR          # Where everything is kept
certs               = $dir/certs          # Where the issued certs are kept
crl_dir             = $dir/crl            # Where the issued CRL are kept
```

```

database                = $dir/index.txt          # Database index file.
unique_subject          = no # Creation of several certificates with same subject.
new_certs_dir           = $dir/newcerts           # Default place for new certs.
certificate              = $dir/issuing_ca_public_cert.pem # The CA Certificate (Public
                                                                # Key)
serial                  = $dir/serial              # The current serial number
crlnumber                = $dir/crlnumber # The current CRL number must be commented out
                                                                # to leave a V1 CRL
crl                      = $dir/crl/issuing_ca.crl # The current CRL
private_key              = $dir/private/issuing_ca_private_key.pem # The CA Private key
RANDFILE                 = $dir/private/.rand      # Private random number file
x509_extensions          = email_cert # The extensions to add to the email certificate
                                                                # (Public Key)
name_opt                 = ca_default              # Subject Name options
cert_opt                 = ca_default              # Certificate field options
crl_extensions           = crl_ext # Certificate Revocation List (CRL) extensions
default_days             = 1460                   # How long to certify for (4 years)
default_crl_days         = 30                     # How long before next CRL (1 month)
default_md                = sha1                   # Which Hash Functions to use.
preserve                 = no                      # Keep passed DN ordering
policy                   = policy_anything         # Default policy

# Extensions to add when CA signs an eMail Security Certificate (Public Key)
[ email_cert ]
basicConstraints          = CA:false
nsComment                 = "eMail Signing Encryption Certificate"
nsCertType                = email
keyUsage                  = critical, nonRepudiation, digitalSignature,
                            keyEncipherment, dataEncipherment
extendedKeyUsage          = emailProtection
subjectKeyIdentifier      = hash
authorityKeyIdentifier    = keyid, issuer:always
authorityInfoAccess       = caIssuers;URI:http://www.homeworks.it/ca/cainfo.html
crlDistributionPoints     = URI:http://www.homeworks.it/crl/issuing_ca.crl
certificatePolicies        = ia5org,@HomeWorks_CPS,@HomeWorks_eMail_CA_Policy

[ HomeWorks_CPS ]
policyIdentifier          = HW-CPS
CPS.1                    = "http://www.homeworks.it/ca/homeworks_cps.html"
userNotice.1             = @HomeWorks_CPS_Notice

[ HomeWorks_CPS_Notice ]
explicitText              = "Home Works S.p.A. Certification Practice Statement"

[ HomeWorks_eMail_CA_Policy ]
policyIdentifier          = HW-MAIL-Cert
userNotice.2             = @HomeWorks_eMail_CA_Notice

[ HomeWorks_eMail_CA_Notice ]
explicitText              = "Home Works S.p.A. Signature and Encryption Mail Certificate
                            Policy"

# CRL extensions
[ crl_ext ]
crlDistributionPoints     = URI:http://www.homeworks.it/crl/issuing_ca.crl

# Requirement for a new Private Key
[ req ]
dir                      = $ENV::CADIR # Where everything is kept
default_bits              = 1024       # This specifies the default key size in bits
default_keyfile            = $dir/private/new_private_key.pem # Default new Private Key
                                                                # file name
distinguished_name        = req_distinguished_name_email # Distinguished Name of the
                                                                # email certificate
attributes                = req_attributes # Certificate Version 3 extensions
x509_extensions           = email_cert # The extensions to add to the CA certificate

# Distinguished Name of the eMail Security Certificate
[ req_distinguished_name_email ]

```

```

name                = First Name (eg, Alessandro)
name_max            = 24
surname             = Surname (eg, Tani)
surname_max         = 64
0.organizationName  = Organization Name (eg, your company)
0.organizationName_default = Home Works S.p.A.
1.organizationName  = Internet Company Web Site
1.organizationName_default = http://www.homeworks.it
organizationalUnitName = Organizational Unit Name (eg, your department)
organizationalUnitName_default = HomeWorks IT Department
commonName          = Person Name (Common Name)
commonName_max      = 64
emailAddress         = Email Address (max 64 characters)
emailAddress_default = support@homeworks.it
emailAddress_max     = 64
localityName         = Locality Name (eg, city)
localityName_default = Reggio Emilia
stateOrProvinceName = State or Province Name (full name)
stateOrProvinceName_default = Italy
countryName          = Country Name (2 letter code)
countryName_default  = IT
countryName_min      = 2
countryName_max      = 2
# SET-ex3            = SET extension number 3

# Challenge password section
[ req_attributes ]
challengePassword      = A challenge password (between 6 and 20 characters)
challengePassword_min = 6
challengePassword_max = 20

# Version 3 Extensions to add to a subordinate CA certificate
[ sub_ca_cert ]
basicConstraints      = CA:false
subjectKeyIdentifier = hash
keyUsage              = nonRepudiation, digitalSignature, keyEncipherment
authorityInfoAccess   = caIssuers;URI:http://www.homeworks.it/ca/cainfo.html
crlDistributionPoints = URI:http://www.homeworks.it/crl/issuing_ca.crl
certificatePolicies    = ia5org,@HomeWorks_CPS,@HomeWorks_CA_policy

# These extensions should be added when creating a proxy certificate
[ proxy_cert_ext ]
basicConstraints      = CA:false
subjectKeyIdentifier = hash
authorityKeyIdentifier = keyid, issuer:always
proxyCertInfo         = critical, language:id-ppl-anyLanguage, pathlen:3,
                        policy:policy_anything

#####
## Policy Sections ##
#####

# For the CA only
[ policy_match ]
organizationName      = match
organizationalUnitName = match
commonName            = supplied
emailAddress          = optional
localityName          = optional
stateOrProvinceName   = match
countryName           = match

# For every certificate (Public Key)
[ policy_anything ]
name                  = optional
surname              = optional
organizationName      = optional
organizationalUnitName = optional
commonName            = supplied

```

```

emailAddress          = optional
localityName          = optional
stateOrProvinceName  = optional
countryName           = optional

# End File

```

More generally, the parties:

```

...
authorityInfoAccess    = caIssuers;URI:http://www.homeworks.it/ca/cainfo.html
crlDistributionPoints  = URI:http://www.homeworks.it/crl/issuing_ca.crl
policyIdentifier       = ...
CPS.1                 = "http://www.homeworks.it/ca/issuing_ca_cps.html";
userNotice.1          = @HomeWorks_Issuing_CA_Notice
explicitText           = "HomeWorks Issuing CA Certification Practice Statement"
explicitText           = "Home Works S.p.A. Secure Communications Mail Server
                        Certificate Policy"

...
stateOrProvinceName_default = Italy
localityName_default       = Reggio Emilia
0.organizationName_default = Home Works S.p.A.
organizationalUnitName_default = HomeWorks IT Department
commonName_default        = HomeWorks Issuing CA
emailAddress_default       = support@homeworks.it
...

```

should be changed, with information on the particular company which is in the process of achieving PKI infrastructure.

In implementing the configuration file `/etc/ssl/openssl.cnf` was assumed that the HomeWorks Issuing CA will provide, during his normal life, many more certificates to digitally sign and encrypt e-mail messages, compared to all other types of certificates that HomeWorks Issuing CA is able to deliver. Therefore, the configuration file proposed generates this kind of *preferential* certificates.

CRL generation of HomeWorks Issuing CA

To complete the creation of HomeWorks Issuing CA, we have to generate the *Certificate Revocation List* (CRL) associated with it. To create the CRL, we will create a digital certificate that will be then revoked. For simplicity we will call this certificate with the name of `$CADIR/certs/crl_public_cert.pem`. So, first we create the public/private key pairs:

```

openssl req -new -nodes -keyout $CADIR/private/crl_private_key.pem -out
$CADIR/request/crl_public_key_req.pem

Generating a 1024 bit RSA private key
.....++++++
.....++++++
writing new private key to '/usr/lib/ssl/misc/CA/private/crl_private_key.pem'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
First Name (eg, Alessandro) []: Alessandro
Surname (eg, Tani) []: Tani
Organization Name (eg, your company) [Home Works S.p.A.]: Home Works S.p.A.
Internet Company Web Site [http://www.homeworks.it]: http://www.homeworks.it
Organizational Unit Name (eg, your department) [HomeWorks IT Department]: HomeWorks IT

```

Department

Person Name (Common Name) []: **CRL of HomeWorks Issuing CA**
Email Address (max 64 characters) [support@homeworks.it]: **support@homeworks.it**
Locality Name (eg, city) [Reggio Emilia]: **Reggio Emilia**
State or Province Name (full name) [Italy]: **Italy**
Country Name (2 letter code) [IT]: **IT**

Please enter the following 'extra' attributes
to be sent with your certificate request

A challenge password []:<-- **Premi Invio**

An optional company name []:<-- **Premi Invio**

We digitally sign the public key `$CADIR/request/crl_public_key_req.pem`:

```
openssl ca -out $CADIR/certs/crl_public_cert.pem -infile  
$CADIR/request/crl_public_key_req.pem
```

We revoke the certificate `$CADIR/certs/crl_public_cert.pem`:

```
openssl ca -revoke $CADIR/certs/crl_public_cert.pem  
Revoking Certificate 84D2C38B199FEA83.  
Data Base Updated
```

We create the CRL of HomeWorks Issuing CA:

```
openssl ca -gencrl -out $CADIR/crl/issuing_ca.crl
```

We convert the CRL to DER format:

```
openssl crl -in $CADIR/crl/issuing_ca.crl -out $CADIR/crl/issuing_ca.crl -outform DER
```

To publish the CRL, we make it accessible through the URL http://www.homeworks.it/crl/issuing_ca.crl
The CRL will be generated once a month (according to what is specified by the parameter `default_crl_days` of the configuration file `/etc/ssl/openssl.cnf`), possibly a week before its expiry. If you want you can automate the creation of CRL using a special script properly scheduled. For example, a possible script could be:

```
#!/bin/sh  
#  
# We create the CRL  
openssl ca -gencrl -out $CADIR/crl/issuing_ca.crl  
#  
# We convert the CRL from the PEM format to the DER format  
openssl crl -in $CADIR/crl/issuing_ca.crl -out $CADIR/crl/issuing_ca.crl -outform  
DER
```

We rename the certifiante `$CADIR/certs/crl_public_cert.pem`, the public key `$CADIR/request/crl_public_key_req.pem` and its private key, `$CADIR/private/crl_private_key.pem`, adding the suffix `.revoked`:

```
mv $CADIR/certs/crl_public_cert.pem $CADIR/certs/crl_public_cert.pem.revoked  
mv $CADIR/request/crl_public_key_req.pem $CADIR/request/crl_public_key_req.pem.revoked  
mv $CADIR/private/crl_private_key.pem $CADIR/private/crl_private_key.pem.revoked
```

At this point, we can enable applications that make use of digital certificates authenticated by HomeWorks Issuing CA to [check periodically](#) the CRL provided by HomeWorks Issuing CA.

How to enable the control of CRL in Firefox and Thunderbird

Usually applications that use digital certificates, are unable to verify, in an automatic way, the CRL associated with each digital certificate. To enable periodic monitoring of CRL, we must manually change the configuration of these application. We shall see below how to enable the control of CRL in programs like [Firefox](#) and [Thunderbird](#). Referring to architecture PKI achieved in this article, the CRLs of the HomeWorks Root CA and HomeWorks Issuing CA are located respectively in the following file (it is worth noting that before they can import the CRL, we need to [install digital certificates](#) associated with HomeWorks Root CA and HomeWorks Issuing CA):

- **CRL of HomeWorks Root CA:** http://www.homeworks.it/crl/root_ca.crl
- **CRL of HomeWorks Issuing CA:** http://www.homeworks.it/crl/issuing_ca.crl

To enable the control of CRL on the programme Firefox just do the following recipe:

- Firefox start;
- open the **Tools** menu and select **Options**;
- open the section **Advanced** and then the **Encryption** subsection;
- make sure they are selected entries **Use SSL 3.0** and **Use TLS 1.0**;
- press the button **Revocation Lists**;
- to add a CRL to control press the button **Import**;
- fill the field **Import CRL from**, one at a time, with the following values:
 - http://www.homeworks.it/crl/root_ca.crl
 - http://www.homeworks.it/crl/issuing_ca.crl
- once inserted one of the values above, press the **OK**. To confirm the loading of CRL press the button **Yes**;
- select the item **Enable Automatic Update for this CRL**. Select then the item **Update Day(s) before next Update date** and set as the number of days before expiry, the value **3**. Press the **OK** button to confirm;
- repeat the procedure indicated both for the CRL of HomeWorks Root CA, and for the HomeWorks Issuing CA;
- press **OK** to close the window entitled **Manage CRLs**;
- returned to the window entitled **Options**, press the button **Verification**;
- check that the item is selected **Do not use OCSP for certificate validation**, alternatively you can select the item **Use OCSP to validate only certificate that specify an OCSP service URL**;
- press **OK** to confirm the choice made;
- close the window entitled **Options** by pressing the **OK** button;
- if you wish, at this point you can close Firefox.

In this way Firefox will monitor regularly the CRL of HomeWorks Root CA and HomeWorks Issuing CA. Similarly, for Thunderbird just do the following:

- Thunderbird start;
- open the **Tools** menu and select **Options**;
- open the section **Advanced** and then the subsection **Certificates**;
- press the button **Revocation Lists**;
- to add a CRL to control press the button **Import**;
- fill the field **Import CRL from**, one at a time, with the following values:
 - http://www.homeworks.it/crl/root_ca.crl
 - http://www.homeworks.it/crl/issuing_ca.crl
- once inserted one of the values above, press the **OK** button;
- to confirm the loading of CRL press the button **Yes**;
- select the item **Enable Automatic Update for this CRL**.
- to set the mode control CRL just imported, press the **Settings** button;
- check that the item is selected **Enable Automatic Update for this CRL**. Select then the item **Update Day(s) before next Update date** and set as the number of days before expiry, the value **3**.

- Press the button **OK** to confirm;
- repeat the procedure indicated both for the CRL of HomeWorks Root CA, and for the HomeWorks Issuing CA;
- Press **OK** to close the window titled **Manage CRLs**;
- returned to the window titled **Options**, press the button **Verification**;
- check that the item is selected **Do not use OCSP for certificate validation**, alternatively, you can select the item **Use OCSP to validate only certificate that specify an OCSP service URL**;
- press **OK** to confirm the choice made;
- close the window titled **Options** pressing the **OK** button;
- if you wish, at this point you can close Thunderbird.

Create the digital certificate and private key of Postfix

Once you have created HomeWorks Issuing CA you can proceed with the generation of public/private key pairs to be assigned to the various applications. We proceed with the creation of the couple [Postfix](#)'s public and private keys. For choice of the authors, all application public/private keys will last long four years. In the generation of public and private key pairs, we must be careful to insert in the **Common Name** the FQDN of email server (which is, in our case, *mail.homeworks.it*). We generate the configuration file to create the application digital certificates, [\\$CADIR/ext/app_req.ext](#):

```
touch $CADIR/ext/app_req.ext
vi $CADIR/ext/app_req.ext
```

we insert the following text:

```
# File /usr/lib/ssl/misc/CA/ext/app_req.ext
#
# Environment Settings
HOME          = .
RANDFILE      = $ENV::HOME/.rnd

#####
## Configuration Sections ##
#####

[ req ]
dir                  = $ENV::CADIR
default_bits         = 1024
default_keyfile      = $dir/private/new_app_private_key.pem
default_days         = 1460
default_md           = sha1
distinguished_name   = req_distinguished_name_app
attributes          = req_attributes

# Distinguished Name of the eMail Security Certificate
[ req_distinguished_name_app ]
0.organizationName   = Organization Name (eg, your company)
0.organizationName_default = Home Works S.p.A.
1.organizationName   = Internet Company Web Site
1.organizationName_default = http://www.homeworks.it
organizationalUnitName = Organizational Unit Name (eg, your department)
organizationalUnitName_default = HomeWorks IT Department
commonName           = FQDN host name (Common Name)
commonName_max       = 64
emailAddress          = Email Address (max 64 characters)
emailAddress_default  = support@homeworks.it
emailAddress_max      = 64
localityName         = Locality Name (eg, city)
localityName_default  = Reggio Emilia
```

```

stateOrProvinceName          = State or Province Name (full name)
stateOrProvinceName_default  = Italy
countryName                  = Country Name (2 letter code)
countryName_default          = IT
countryName_min              = 2
countryName_max              = 2
# SET-ex3                    = SET extension number 3

# Challenge password section
[ req_attributes ]
challengePassword            = A challenge password (between 6 and 20 characters)
challengePassword_min        = 6
challengePassword_max        = 20

# End File

```

Now we are ready to generate the Postfix's public/private key pairs:

```

openssl req -new -nodes -keyout $CADIR/private/postfix_private_key.pem -out
$CADIR/request/postfix_public_key_req.pem -config $CADIR/ext/app_req.ext

Generating a 1024 bit RSA private key
.....++++++
.+++++
writing new private key to '/usr/lib/ssl/misc/CA/private/postfix_private_key.pem'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Organization Name (eg, your company) [Home Works S.p.A.]: Home Works S.p.A.
Internet Company Web Site [http://www.homeworks.it]: http://www.homeworks.it
Organizational Unit Name (eg, your department) [HomeWorks IT Department]: HomeWorks IT
Department
FQDN host name (Common Name) []: mail.homeworks.it
Email Address (max 64 characters) [support@homeworks.it]: postmaster@homeworks.it
Locality Name (eg, city) [Reggio Emilia]: Reggio Emilia
State or Province Name (full name) [Italy]: Italy
Country Name (2 letter code) [IT]: IT

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []: homeworks

```

We check that *Postfix's* public key has been generated correctly:

```

openssl req -text -noout -in $CADIR/request/postfix_public_key_req.pem

```

Before proceeding with the generation of the Postfix's certificate, we must create the file with extensions X.509 to be applied to the certificate. Therefore we will create the file [\\$CADIR/ext/mail_server_x509_cert.ext](#):

```

touch $CADIR/ext/mail_server_x509_cert.ext
vi $CADIR/ext/mail_server_x509_cert.ext

```

we insert the following text:

```

# File /usr/lib/ssl/misc/CA/ext/mail_server_x509_cert.ext

basicConstraints              = CA:false
nsComment                    = "Mail Server Certificate"
nsCertType                   = server, client
keyUsage                     = critical, digitalSignature, keyEncipherment
extendedKeyUsage              = serverAuth, clientAuth

```

```

subjectKeyIdentifier    = hash
authorityKeyIdentifier = keyid, issuer:always
authorityInfoAccess     = caIssuers;URI:http://www.homeworks.it/ca/cainfo.html
crlDistributionPoints   = URI:http://www.homeworks.it/crl/issuing_ca.crl
certificatePolicies     = ia5org,@HomeWorks_CPS,@HomeWorks_Mail_Server_CA_Policy

[ HomeWorks_CPS ]
policyIdentifier = 1.3.6.1.4.1.31012.1.1
CPS.1           = "http://www.homeworks.it/ca/homeworks_cps.html"
userNotice.1    = @HomeWorks_CPS_Notice

[ HomeWorks_CPS_Notice ]
explicitText     = "Home Works S.p.A. Certification Practice Statement"

[ HomeWorks_Mail_Server_CA_Policy ]
policyIdentifier = 1.3.6.1.4.1.31012.3.1
userNotice.2    = @HomeWorks_Mail_Server_CA_Notice

[ HomeWorks_Mail_Server_CA_Notice ]
explicitText     = "Home Works S.p.A. Secure Communications Mail Server Certificate
Policy"

# End File

```

Now we can proceed to create the Postfix's certificate:

```

openssl ca -policy policy_anything -out $CADIR/certs/postfix_public_cert.pem -extfile $CADIR/
ext/mail_server_x509_cert.ext -infiles $CADIR/request/postfix_public_key_req.pem

```

Using configuration from /etc/ssl/openssl.cnf

Check that the request matches the signature

Signature ok

Certificate Details:

Serial Number:

84:d2:c3:8b:19:9f:ea:84

Validity

Not Before: May 24 22:38:24 2008 GMT

Not After : May 23 22:38:24 2012 GMT

Subject:

organizationName = Home Works S.p.A.

organizationName = http://www.homeworks.it

organizationalUnitName = HomeWorks IT Department

commonName = mail.homeworks.it

emailAddress = postmaster@homeworks.it

localityName = Reggio Emilia

stateOrProvinceName = Italy

countryName = IT

X509v3 extensions:

X509v3 Basic Constraints:

CA:FALSE

Netscape Comment:

Mail Server Certificate

Netscape Cert Type:

SSL Client, SSL Server

X509v3 Key Usage:

Digital Signature, Non Repudiation, Key Encipherment, Data Encipherment

X509v3 Extended Key Usage: critical

TLS Web Server Authentication, TLS Web Client Authentication

X509v3 Subject Key Identifier:

05:57:24:DB:4C:D8:18:0C:C8:35:99:30:1F:55:C5:FF:99:E2:F7:CD

X509v3 Authority Key Identifier:

keyid:CC:A7:3D:F0:35:F0:83:8E:5A:1F:D0:67:AD:E9:63:95:5F:3C:C4:74

DirName:/C=IT/ST=Italy/O=Home Works S.p.A./O=http://www.homeworks.it/OU=HomeWorks IT
Department/CN=HomeWorks Root CA/emailAddress=support@homeworks.it

serial:F0:27:8F:E6:31:7D:C5:D7

Authority Information Access:

CA Issuers - URI:http://www.homeworks.it/ca/cainfo.html

X509v3 CRL Distribution Points:

URI:http://www.homeworks.it/crl/issuing_ca.crl

```
X509v3 Certificate Policies:
Policy: 1.3.6.1.4.1.31012.1.1
CPS: http://www.homeworks.it/ca/homeworks_cps.html;
User Notice:
    Explicit Text: Home Works S.p.A. Certification Practice Statement
Policy: 1.3.6.1.4.1.31012.3.1
User Notice:
    Explicit Text: Home Works S.p.A. Secure Communications Mail Server Certificate
Policy

Certificate is to be certified until May 23 22:38:24 2012 GMT (1460 days)
Sign the certificate? [y/n]: y

1 out of 1 certificate requests certified, commit? [y/n] y
Write out database with 1 new entries
Data Base Updated
```

We check if the *Certification Path* of the certificate just created is valid:

```
openssl verify -CAfile $CADIR/root_ca_public_cert.pem -untrusted
$CADIR/issuing_ca_public_cert.pem $CADIR/certs/postfix_public_cert.pem
/usr/lib/ssl/misc/CA/certs/postfix_public_cert.pem: OK
```

OR:

```
openssl verify -CAfile $CADIR/global_ca_public_cert.pem $CADIR/certs/postfix_public_cert.pem
/usr/lib/ssl/misc/CA/certs/postfix_public_cert.pem: OK
```

We check if the certificate is an *SSL Server Certificate*:

```
openssl verify -purpose sslserver -CAfile $CADIR/global_ca_public_cert.pem
$CADIR/certs/postfix_public_cert.pem
/usr/lib/ssl/misc/CA/certs/postfix_public_cert.pem: OK
```

We check the content of the Postfix's certificate:

```
openssl x509 -text -noout -in $CADIR/certs/postfix_public_cert.pem

Certificate:
Data:
  Version: 3 (0x2)
  Serial Number:
    84:d2:c3:8b:19:9f:ea:84
  Signature Algorithm: sha1WithRSAEncryption
  Issuer: C=IT, ST=Italy, L=Reggio Emilia, O=Home Works S.p.A., O=http://www.homeworks.it,
OU=HomeWorks IT Department, CN=HomeWorks Issuing CA/emailAddress=support@homeworks.it
  Validity
    Not Before: May 24 22:38:24 2008 GMT
    Not After : May 23 22:38:24 2012 GMT
  Subject: O=Home Works S.p.A., O=http://www.homeworks.it, OU=HomeWorks IT Department,
CN=mail.homeworks.it/emailAddress=postmaster@homeworks.it, L=Reggio Emilia, ST=Italy, C=IT
  Subject Public Key Info:
    Public Key Algorithm: rsaEncryption
    RSA Public Key: (1024 bit)
      Modulus (1024 bit):
        00:bf:f6:7c:be:ff:dd:da:29:84:39:a8:f6:4b:af:
        08:fa:27:9f:92:d0:de:ab:26:36:70:66:c2:e4:ad:
        6c:05:d6:21:44:4e:2a:d9:b3:8a:24:47:04:42:67:
        8f:52:de:28:54:c8:ec:5a:58:dd:36:ac:06:fd:18:
        6a:29:46:2a:6a:3c:99:15:aa:f1:7b:f5:94:de:41:
        77:44:f0:f7:b9:a7:fe:8e:57:be:e9:14:26:e6:41:
        36:9d:6e:a6:b4:83:fc:ff:93:c7:3f:82:94:98:26:
        9e:61:4d:3c:07:48:68:a1:46:d1:0e:c9:5b:77:7e:
        e5:58:2e:18:e2:74:4c:ef:59
      Exponent: 65537 (0x10001)
  X509v3 extensions:
    X509v3 Basic Constraints:
      CA:FALSE
    Netscape Comment:
      Mail Server Certificate
```

```

Netscape Cert Type:
  SSL Client, SSL Server
X509v3 Key Usage: critical
  Digital Signature, Non Repudiation, Key Encipherment, Data Encipherment
X509v3 Extended Key Usage:
  TLS Web Server Authentication, TLS Web Client Authentication
X509v3 Subject Key Identifier:
  05:57:24:DB:4C:D8:18:0C:C8:35:99:30:1F:55:C5:FF:99:E2:F7:CD
X509v3 Authority Key Identifier:
  keyid:CC:A7:3D:F0:35:F0:83:8E:5A:1F:D0:67:AD:E9:63:95:5F:3C:C4:74
  DirName:/C=IT/ST=Italy/O=Home Works S.p.A./O=http://www.homeworks.it/OU=HomeWorks IT
Department/CN=HomeWorks Root CA/emailAddress=support@homeworks.it
  serial:F0:27:8F:E6:31:7D:C5:D7

Authority Information Access:
  CA Issuers - URI:http://www.homeworks.it/ca/cainfo.html

X509v3 CRL Distribution Points:
  URI:http://www.homeworks.it/crl/issuing_ca.crl

X509v3 Certificate Policies:
  Policy: 1.3.6.1.4.1.31012.1.1
    CPS: http://www.homeworks.it/ca/homeworks_cps.html;
    User Notice:
      Explicit Text: Home Works S.p.A. Certification Practice Statement
  Policy: 1.3.6.1.4.1.31012.3.1
    User Notice:
      Explicit Text: Home Works S.p.A. Secure Communications Mail Server Certificate
Policy

Signature Algorithm: sha1WithRSAEncryption
8b:d5:ca:3d:fa:8c:30:0a:9c:db:c7:1b:43:64:63:5f:7c:e4:
70:7d:b3:4a:88:48:de:a2:ff:ad:fb:c5:8c:38:f5:4e:73:7a:
25:33:e4:1e:f5:b1:10:de:4b:4f:d7:13:84:67:ac:b1:3d:1f:
91:1b:95:e3:a7:9a:23:a0:32:b8:d5:7c:2b:26:d5:d7:b0:a5:
a4:bb:5d:52:c7:f2:f7:8c:9a:16:8c:a7:84:46:03:70:08:84:
96:18:b5:e2:3c:f8:f6:86:39:43:16:49:97:e3:91:78:92:f3:
10:88:bd:6b:38:29:ce:00:83:7e:2d:df:a8:dd:1a:78:b4:a4:
65:59

```

So the Postfix's private key and digital certificate are these files:

- `$CADIR/private/postfix_private_key.pem` is the Postfix's *private key*;
- `$CADIR/certs/postfix_public_cert.pem` is the Postfix's *digital certificate*.

We make available to Postfix its certificate and its private key just created:

- the certificate of Postfix, `$CADIR/certs/postfix_public_cert.pem`, should be copied to the folder `/etc/postfix/certs/` on the server where Postfix is installed;
- the private key of Postfix, `$CADIR/private/postfix_private_key.pem`, should be copied to the folder `/etc/postfix/certs/` on the server where Postfix is installed. Restrict the access to the private key to only root user;
- the global certificate for HomeWorks Root CA and HomeWorks Issuing CA, `$CADIR/global_ca_public_cert.pem`, should be copied to the folder `/etc/postfix/certs/` on the server where Postfix is installed;

For example:

```

cp $CADIR/private/postfix_private_key.pem /etc/postfix/certs/
chmod 600 /etc/postfix/certs/postfix_private_key.pem
cp $CADIR/certs/postfix_public_cert.pem /etc/postfix/certs/
cp $CADIR/global_ca_public_cert.pem /etc/postfix/certs/

```

In the configuration that we adopt, we configure Postfix to consider *reliable* only the HomeWorks Issuing CA, and, to consider only valid certificates generated by HomeWorks Issuing CA. Therefore we change the Postfix configuration file, `/etc/postfix/main.cf`, on the SMTP server sections (or when Postfix behaves as an SMTP server):

```
postconf -e smtpd_tls_CAfile = /etc/postfix/certs/global_ca_public_cert.pem
postconf -e smtpd_tls_cert_file = /etc/postfix/certs/postfix_public_cert.pem
postconf -e smtpd_tls_key_file = /etc/postfix/certs/postfix_private_key.pem
postconf -e smtpd_use_tls = yes
postconf -e smtpd_tls_session_cache_database = btree:${queue_directory}/smtpd_scache
postconf -e smtpd_tls_session_cache_timeout = 3600s
postconf -e smtpd_tls_auth_only = no
postconf -e smtpd_tls_loglevel = 1
postconf -e smtpd_tls_received_header = yes
postconf -e tls_random_source = dev:/dev/urandom
```

While on the SMTP client section (or as Postfix behaves as an SMTP client) we configure Postfix to consider reliable all public certificates (the list of public CA to be considered reliable is located within the file `/etc/ssl/certs/ca-certificates.crt`):

```
ln -s /etc/ssl/certs/ca-certificates.crt /etc/postfix/certs/ca-certificates.crt
postconf -e smtp_use_tls = yes
postconf -e smtp_tls_note_starttls_offer = yes
postconf -e smtp_tls_CAfile = /etc/postfix/certs/ca-certificates.crt
postconf -e smtp_tls_session_cache_database = btree:${queue_directory}/smtp_scache
postconf -e smtp_tls_session_cache_timeout = 3600s
smtp_tls_loglevel = 1
```

The configuration about Transport Layer Security (TLS) located within the file `/etc/postfix/main.cf` should look like:

```
cat /etc/postfix/main.cf
```

```
...
# TLS parameters (Server Side, from this SMTP Server to Mail Client)
smtpd_tls_CAfile = /etc/postfix/certs/global_ca_public_cert.pem
smtpd_tls_cert_file = /etc/postfix/certs/postfix_public_cert.pem
smtpd_tls_key_file = /etc/postfix/certs/postfix_private_key.pem
smtpd_use_tls = yes
smtpd_tls_session_cache_database = btree:${queue_directory}/smtpd_scache
smtpd_tls_auth_only = no
smtpd_tls_loglevel = 1
smtpd_tls_received_header = yes
smtpd_tls_session_cache_timeout = 3600s
tls_random_source = dev:/dev/urandom

# See /usr/share/doc/postfix/TLS_README.gz in the postfix-doc package for
# information on enabling SSL in the smtp client.

# TLS parameters (Client Side, from this SMTP Server to another SMTP Server)
smtp_use_tls = yes
smtp_tls_note_starttls_offer = yes
smtp_tls_CAfile = /etc/postfix/certs/ca-certificates.crt
smtp_tls_session_cache_database = btree:${queue_directory}/smtp_scache
smtp_tls_session_cache_timeout = 3600s
smtp_tls_loglevel = 1
...
```

As in the authentication process between two TLS SMTP server, sometimes we may experience some problems, for these situations we advance create a special file that inhibit the use of TLS protocol to certain SMTP server. We create for this purpose the file `/etc/postfix/deny_tls_per_domains`:

```
vi /etc/postfix/deny_tls_per_domains
```

We insert the following text:

```
# File /etc/postfix/deny_tls_per_domains
#
# Insert the DNS domain should to be denied to use client-side TLS
```

```
#
# Example: mybusinessdomain.com                               None
#
# DNS domain name                                           Action (None)
#
gmail.com                                                    None
# End File
```

We create the hash map of the file, `/etc/postfix/deny_tls_per_domains`:

```
postmap hash:/etc/postfix/deny_tls_per_domains
```

We insert the follow line in the file, `/etc/postfix/main.cf`:

```
postconf -e smtp_tls_per_site = hash:/etc/postfix/deny_tls_per_domains
```

and check the correct configuration, of the file `/etc/postfix/main.cf`, simulating a TLS connection:

```
openssl s_client -starttls smtp -CAfile /etc/postfix/certs/global_ca_public_cert.pem -connect localhost:25

CONNECTED(00000003)
depth=2 /C=IT/ST=Italy/O=Home Works S.p.A./OU=HomeWorks IT Department/CN=HomeWorks Root
CA/emailAddress=support@homeworks.it
verify return:1
depth=1 /C=IT/ST=Italy/L=Reggio Emilia/O=Home Works S.p.A./OU=HomeWorks IT
Department/CN=HomeWorks Issuing CA/emailAddress=support@homeworks.it
verify return:1
depth=0 /C=IT/ST=Italy/L=Reggio Emilia/O=Home Works S.p.A./OU=HomeWorks IT
Department/CN=mail.homeworks.it/emailAddress=postmaster@homeworks.it
verify return:1
---
Certificate chain
0 s:/C=IT/ST=Italy/L=Reggio Emilia/O=Home Works S.p.A./OU=HomeWorks IT
Department/CN=mail.homeworks.it/emailAddress=postmaster@homeworks.it
i:/C=IT/ST=Italy/L=Reggio Emilia/O=Home Works S.p.A./OU=HomeWorks IT Department/CN=HomeWorks
Issuing CA/emailAddress=support@homeworks.it
1 s:/C=IT/ST=Italy/L=Reggio Emilia/O=Home Works S.p.A./OU=HomeWorks IT
Department/CN=HomeWorks Issuing CA/emailAddress=support@homeworks.it
i:/C=IT/ST=Italy/O=Home Works S.p.A./OU=HomeWorks IT Department/CN=HomeWorks Root
CA/emailAddress=support@homeworks.it
2 s:/C=IT/ST=Italy/O=Home Works S.p.A./OU=HomeWorks IT Department/CN=HomeWorks Root
CA/emailAddress=support@homeworks.it
i:/C=IT/ST=Italy/O=Home Works S.p.A./OU=HomeWorks IT Department/CN=HomeWorks Root
CA/emailAddress=support@homeworks.it
---
Server certificate
-----BEGIN CERTIFICATE-----
MIIErzCCBBigAwIBAgIJANgUMb8Nfv2RMA0GCSqGSIb3DQEBBQUAMIG3MQswCQYD
VQQGEwJJVDEOMAwGA1UECBMFsXRhbHkxFjAUBgNVBACITDVJlZ2dpbyBFbWlsaWEEx
GjAYBgNVBAoTEUhhbWVudG9yV29ya3MgUy5wLkEuMSAwHgYDVQQLEXdIb21lV29ya3Mg
SVQgRGVwYXJ0bWVudDEdMBsGA1UEAxMUSG9tZVdvcmtzIElzc3VpbmcgQ0ExIzAh
BgkqhkiG9w0BCQEFwFHN1cHBvcnRAaG9tZVdvcmtzLml0MB4XDTA4MDMyOTAYMDgy
Ml0xZDTEyMDMyODAYMDgyMl0wgbcxZCZAJBgNVBAYTAklUMQ4wDAYDVQQIEwVJdGFs
eTEWMBQGA1UEBxMNUmVnZ21vIEVtaWxpYTEaMBGGA1UEChMRSG9tZSBXb3JrcyBT
LnAuQS4xIDAeBgNVBASTF0hbWVXb3JrcyBJVCBEZXBhcnRtZW50MRowGAYDVQQDD
ExFtYWlsLmhhbWVudG9yV29ya3MgUy5pZDEmMCQGCsGSIb3DQEJARYXcG9zdG1hc3RlckBo
b21ld29ya3MuaXQwZ28wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBALGJI5ZBwk5V
/14bTChuPh8f3R91UW1bEUJze7IxWIXaSqms9W2CnPXZ03MQmp0uMA/p/BnzaqZk
KoVowPid99+S3l0EmYlv0tbZ2ckVtHbbi9wP3ZXRHZUfZoa7ALhDoBrWH5TjyN6Q
nZ07tiqlE8PHCY55lbMRkbZW378fQ7z1AgMBAAEggG/MIIBuzAJBgNVHRMEAjAA
MBEGCWCsSAGG+EIBAQQEAwIGwDALBgNVHQ8EBAMCBPAwHQYDVR0lBBYwFAYIKwYB
BQUHAWEGCCsGAQUFBwMCMB0GA1UdDgQWBBrBtbnfg2nht4wSK1MDF6uYzT12U6TCB
0QYDVR0jBIHJMIHGgBSsR7OnfSaM+wdWc7ZiHQ/raI6T16GBoqSBnzCBnDELMaK
A1UEBhMCSVQxQjAMBgNVBAGTBUI0YXx5MRowGAYDVQQKEwFtY21lIFdvcmtzIFMu
cC5BLjEgMB4GA1UECxMUSG9tZVdvcmtzIElUIERlclCGFydG1lbnQxGjAYBgNVBAMT
EUhhbWVXb3JrcyBSb290IENBMSMwIQYJKoZIhvcNAQkBFhRzdXBwb3J0QGhvbWV3
b3Jrcy5pdIIJANgUMb8Nfv2KMD8GCCsGAQUFBwEBBDMwMTAvBggrBgEFBQcwAoYj
aHR0cDovL3d3dy50b21ld29ya3MuaXQvY2FpbmZvLmhhbWVudG9yV29ya3MgUy5p
ZDEmMCQGCsGSIb3DQEJARYXcG9zdG1hc3RlckBob21ld29ya3MuaXQvY3JsL2lzc3VpbmdfY2EuY3Js
```

```

MA0GCSqGSIb3DQEBAQUAA4GBAE7cTPDfIle6tbHIwDMH+tY8s3KM2wFxdE10iAu
mXINBE6t5AshDdghHw/vjmWGPnt2Wh6mcGlckdrtXhwtal6q2Wgbf/1Z7PDfGBA3
K0t1t+vxSL00Nm4FeO+MwRu7W4mbKqW0UaZzzDhOp80b1exSP5E/fZS1rD5Cx2PB
L7tG
-----END CERTIFICATE-----
subject=/C=IT/ST=Italy/L=Reggio Emilia/O=Home Works S.p.A./OU=HomeWorks IT
Department/CN=mail.homeworks.it/emailAddress=postmaster@homeworks.it
issuer=/C=IT/ST=Italy/L=Reggio Emilia/O=Home Works S.p.A./OU=HomeWorks IT
Department/CN=HomeWorks Issuing CA/emailAddress=support@homeworks.it
---
No client certificate CA names sent
---
SSL handshake has read 3899 bytes and written 326 bytes
---
New, TLSv1/SSLv3, Cipher is DHE-RSA-AES256-SHA
Server public key is 1024 bit
Compression: NONE
Expansion: NONE
SSL-Session:
    Protocol : TLSv1
    Cipher : DHE-RSA-AES256-SHA
    Session-ID: 101D8C076F036B0AE5A8F9483BBB6009382228B280D648EC0F9853E24CE02EB8
    Session-ID-ctx:
    Master-
Key:CC5AACC3AC41023245CA2FAC724E08445A021CA53D5CCEAAC071FA936C723DF623CEDB72421AAC22CDC3D71FE
8E6CC58
    Key-Arg : None
    Start Time: 1207172342
    Timeout : 300 (sec)
    Verify return code: 0 (ok)
---
220 born.homeworks.it ESMTP Postfix (Debian/GNU)

```

We close the connection test `openssl s_client -starttls smtp -CAfile /etc/postfix/certs/root_ca_private_key.pem -connect localhost:25:`

```

quit

221 2.0.0 Bye
read:errno=0

```

If in the result of the connection TLS simulated there is the message `Verify return code: 0 (ok)`, then it means that the configuration of Postfix is correct. At this point you can configure e-mail clients so that they can connect on SMTP-TLS to Postfix server.

Generate the digital certificate and private key of Courier

[Courier](#) is one of the most popular Open Source program. Between its features there is the possibility of realize IMAP, POP3, IMAP-SSL and POP3-SSL connections. The basic configuration of [Courier](#) allows you to automatically create digital certificates necessary for establishing IMAP-SSL and POP3-SSL connections, but if you want to integrate the communication process Transport Layer Security (TLS) with a PKI infrastructure, you should proceed to the generation of own certificates to be allocated to [Courier](#). In our example, the CA predisposed to the generation of certificates is the HomeWorks Issuing CA. [Courier](#) can use the certificates generated by a CA, if the certificates meets the following conditions:

- certificates used by the programme Courier must call as follows:
 - `/etc/courier/imapd.pem` IMAP-SSL connection ;
 - `/etc/courier/pop3d.pem` POP3-SSL connection;
- both certificates are the union of digital certificate and the corresponding private key assigned to Courier;
- both certificates must contain the Diffie-Hellman parameters.

We start with the generation of public/private key pairs to assign to Courier (in our example, Courier is hosted on the same server on whose was installed Postfix on previous section, *mail.homeworks.it*):

```
openssl req -new -nodes -keyout $CADIR/private/courier_private_key.pem -out
$CADIR/request/courier_public_key_req.pem -config $CADIR/ext/app_req.ext

Generating a 1024 bit RSA private key
...+++++
...+++++
writing new private key to '/usr/lib/ssl/misc/CA/private/courier_private_key.pem'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Organization Name (eg, your company) [Home Works S.p.A.]: Home Works S.p.A.
Internet Company Web Site [http://www.homeworks.it]: http://www.homeworks.it
Organizational Unit Name (eg, your department) [HomeWorks IT Department]: HomeWorks IT
Department
FQDN host name (Common Name) []: mail.homeworks.it
Email Address (max 64 characters) [support@homeworks.it]: postmaster@homeworks.it
Locality Name (eg, city) [Reggio Emilia]: Reggio Emilia
State or Province Name (full name) [Italy]: Italy
Country Name (2 letter code) [IT]: IT

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password (between 6 and 20 characters) []: homeworks
```

We create the Courier's digital certificate (note that to create this digital certificate we are using the option `-notext`):

```
openssl ca -policy policy_anything -notext -out $CADIR/certs/courier_public_cert.pem -extfile
$CADIR/ext/mail_server_x509_cert.ext -infiles $CADIR/request/courier_public_key_req.pem

Using configuration from /etc/ssl/openssl.cnf
Check that the request matches the signature
Signature ok
Certificate Details:
  Serial Number:
    84:d2:c3:8b:19:9f:ea:85
  Validity
    Not Before: May 25 14:45:46 2008 GMT
    Not After : May 24 14:45:46 2012 GMT
  Subject:
    organizationName = Home Works S.p.A.
    organizationName = http://www.homeworks.it
    organizationalUnitName = HomeWorks IT Department
    commonName = mail.homeworks.it
    emailAddress = postmaster@homeworks.it
    localityName = Reggio Emilia
    stateOrProvinceName = Italy
    countryName = IT
  X509v3 extensions:
    X509v3 Basic Constraints:
      CA:FALSE
    Netscape Comment:
      Mail Server Certificate
    Netscape Cert Type:
      SSL Client, SSL Server
    X509v3 Key Usage: critical
      Digital Signature, Key Encipherment
    X509v3 Extended Key Usage:
      TLS Web Server Authentication, TLS Web Client Authentication
    X509v3 Subject Key Identifier:
      A2:29:CF:7E:99:E8:3F:1A:48:A4:68:25:4D:26:DB:0A:CD:72:CE:B0
    X509v3 Authority Key Identifier:
      keyid:CC:A7:3D:F0:35:F0:83:8E:5A:1F:D0:67:AD:E9:63:95:5F:3C:C4:74
      DirName:/C=IT/ST=Italy/O=Home Works S.p.A./O=http://www.homeworks.it/OU=HomeWorks IT
      Department/CN=HomeWorks Root CA/emailAddress=support@homeworks.it
```

```

serial:F0:27:8F:E6:31:7D:C5:D7

Authority Information Access:
  CA Issuers - URI:http://www.homeworks.it/ca/cainfo.html
X509v3 CRL Distribution Points:
  URI:http://www.homeworks.it/crl/issuing_ca.crl
X509v3 Certificate Policies:
  Policy: HW-CPS
    CPS: http://www.homeworks.it/ca/homeworks_cps.html;
  User Notice:
    Explicit Text: Home Works S.p.A. Certification Practice Statement
  Policy: HW-TLS-MAIL-Cert
  User Notice:
    Explicit Text: Home Works S.p.A. Secure Communications Mail Server Certificate
Policy

Certificate is to be certified until Apr 1 23:35:40 2012 GMT (1460 days)
Sign the certificate? [y/n]: y

1 out of 1 certificate requests certified, commit? [y/n] y
Write out database with 1 new entries
Data Base Updated

```

We check if the Courier's *certificate* has been generated correctly:

```
openssl x509 -text -noout -in $CADIR/certs/courier_public_cert.pem
```

We check if the Certification Path of the just created certificate is valid:

```
openssl verify -CAfile $CADIR/global_ca_public_cert.pem $CADIR/certs/courier_public_cert.pem
/usr/lib/ssl/misc/CA/certs/courier_public_cert.pem: OK
```

We check if the Courier's digital certificate, is an SSL Server Certificate:

```
openssl verify -purpose sslserver -CAfile $CADIR/global_ca_public_cert.pem
$CADIR/certs/courier_public_cert.pem
/usr/lib/ssl/misc/CA/certs/courier_public_cert.pem: OK
```

We join the Courier's digital certificate with its private key:

```
cat $CADIR/private/courier_private_key.pem $CADIR/certs/courier_public_cert.pem >
$CADIR/certs/courier_cert.pem
```

We insert Diffie-Hellman parameters at the end of the just created certificate (this may take a some minutes):

```
openssl dhparam 1024 >> $CADIR/certs/courier_cert.pem
```

The final result should be:

```
cat $CADIR/certs/courier_cert.pem
```

```

-----BEGIN RSA PRIVATE KEY-----
MIICXgIBAAKBgQDY2iG5Y03Fz/D2UwVO3hDu5vcu67PoiUZhSLZGufkFnxau2Imx
V6l6Xs8FJXGWpiPxgm/1FTvgqH1bLVJwVcAr54ESyfSys6WXV2jzghpTLKtME6QF
WUcun2+jDbotXWp4MrUBj fesKrpj 9R8JKi/XxRMzoQTyl/YRb0wXwW06ZQIDAQAB
AoGAPJuy40rC+O+mbGJF0IY2e18oZP/Rt8NuXVBiSaA+3nhZcaLp0RwsLRyEhe6y
MaXb0+td+UTnCGJvLuWa7fS5kcfWrBzh11HlrtzM104AaVkjZgbilG57EdwCouR
wEDoUQ4HL55MoFbglDgCVHwHuRu21mjuSdngefBpNiQBcECQQD7TUIA9fo/zr1E
IC2h+KyV3Lv/1XQ6DRpBmdDJkNVly78dGmWD1tbk3M1gy7Nd3D83aU6NvLyX1144
MdLeDCO1AkeA3Of+9+Y+3TujcFr1qX6Awvmq00qZRC0fp3iEaTGpcEup4zd+/Y9x
5VknNtTxXe9vtR8krG5XMR/0HdN/7y1J8QJBAOSJTg0xpXOBvFqIKPez/sALDa2L
oTdp0wb1qzqjzG3W7Oa6qrdLGgLoCp6MoYIqWhM6YYXkrl4oLjdMmEf3IkECQQCE
eOdwt/V47Bu98/4f74m94sTrQnAY70ptPpuBDdQDUIyHgq831T7C/50qBZbc8wo1

```

```
PoDamqzU+8mD3WJ6BtmxAkEA5NvFW499+rD8TmhUj3MQzmORfUhUnrd+hm+RKZHM
JwtONAn7TO8LgaNmDS/tePv6+ztQ2vV2rswIID6Zu7voLw==
-----END RSA PRIVATE KEY-----
-----BEGIN CERTIFICATE-----
MIIErzCCBBigAwIBAgIJANgUMb8Nfv2SMA0GCSqGSIb3DQEBBQUAMIG3MQswCQYD
VQQGEwJJVDEOMAwGA1UECBMFSXRhbHkxYjFjAUBgNVBAcTDVJlZ2dpbyBFbWlsawEx
GjAYBgNVBAoTEUhhbWUgV29ya3MgUy5wLkEuMSAwHgYDVQQLExdIb21lV29ya3Mg
SVQqRGVwYXJ0bWVudDEdMBsGA1UEAxMUSG9tZVdvcmtzIElzc3VpbmcgQ0ExIzAh
BgkqhkiG9w0BCQEFwFHNlcHBvcnRAAG9tZVdvcmtzLml0MB4XDTA4MDQwMjIzMzU0
MfoXDTEyMDQwMTIzMzU0MFowgbcxZzAJBgNVBAYTAklUMQ4wDAYDVQQIEWVJdGFs
eTEWMBQGA1UEBxMNUmVnZ21vIEVtaWxpYTEaMBGGA1UEChMRSG9tZSBXb3JrcyBT
LnAuQS4xIDAeBgNVBASATF0hvbWVXb3JrcyBJVCBEZXBhcnRtZW50MR0wGAYDVQQD
ExFtYWlsLmhhbWV3b3Jrcy5pdDEmMCQGCsGSIb3DQEJARYXcG9zdG1hc3RlcBo
b21ld29ya3MuaXQwgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBANjaIb1jTcXP
8PZTBU7eEO7m9y7rs+iJRmFitka5+QWfFq7YibFXqXpezWUlcZamI/GCb/UVO+Co
fVstUnBvWcVngRLJ9LKzpZdXaPOCGLMspMwTpAVZRY6fb6MNUildangytQGN96wq
umPlHwkqL9fFEzOhBPKX9hFvTBfBbTPlAgMBAAGjggG/MIIBuzAJBgNVHRMEAjAA
MBEGCWCsAGG+EIBAQQEAwIGDALBgNVHQ8EBAMCBPAwHQYDVR01BBYwFAYIKwYB
BQUHAWEGCCsGAQUFBwMCMCB0GA1UdDgQWBbT108U4nVC09c2hcDT6tG/Mtnx2oDCB
0QYDVR0jBIHUMIHGGBSsR7OnfSaM+wdWc7ZiHQ/raI6T16GBoqSBnzCBnDELMakG
A1UEBhMCSVQxZjAMBgNVBAgTBUL0YwX5MR0wGAYDVQQKEXFib21lIFdvcmtzIFMu
cC5BLjEgMB4GA1UECzMxSG9tZVdvcmtzIElUIERlcGFydG1lbnQxGjAYBgNVBAMT
EUhhbWVXb3JrcyBSb290IENBMSMwIQYJKoZIhvcNAQkBFhRzdXBwb3J0QGhvbWV3
b3Jrcy5pdIIJANgUMb8Nfv2KMD8GCCsGAQUFBwEBBDMwMTAvBggrBgEFBQcwAoYj
aHR0cDovL3d3dy5ob21ld29ya3MuaXQvY2FpbmZvLmh0bWwwOwYDVR0fBDQwMjAw
cC6gLIYqaHR0cDovL3d3dy5ob21ld29ya3MuaXQvY3JsL21zc3VpbmdfY2EuY3Js
MA0GCSqGSIb3DQEBBQUAA4GBABilcK2sE5PrPkM1AnalyPEGTLnODRfSpRhGP/a4
4Cniqm1/htMZAJTgBPh+TC4Z2FteOTAHIfK3jqoHH8AL9UBfP7+swgygAyX4rJmv
Q+HUPPMTfwm3aj8NsNce0jJY1A8+/t4XfX/cOM3yrQzkOVb2/zERXUjhcvOTXwMB
sQ40
-----END CERTIFICATE-----
-----BEGIN DH PARAMETERS-----
MIGHAoGBAN/PC6aWXnCGNG/wnWcFxtEdym0+TLUBb24Xgmtm/n9TAR7++/zUtj9
3Bj98/I4byWk4CCj7cvl6uIA6hRt14HD1qEc2vOo9Purz40zZnXrKPnCDyWGG0EO
aBZlI897f3HjXBat45IchLDIGgO7lR4ekXG5FmRzaU+rQE7V/SEzAgEC
-----END DH PARAMETERS-----
```

At this point we can assign these certificate to Courier:

- the Courier's certificates, `$CADIR/certs/courier_cert.pem`, should be copied to the folder `/etc/courier/`, on the server where Courier is installed, with the name of `imapd.pem` and `pop3d.pem` respectively;
- the access to certificates `/etc/courier/imapd.pem` and `/etc/courier/pop3d.pem` should be limited only to daemon user.

For example:

```
cp $CADIR/certs/courier_cert.pem /etc/courier/imapd.pem
cp $CADIR/certs/courier_cert.pem /etc/courier/pop3d.pem
chmod 0600 /etc/courier/imapd.pem
chmod 0600 /etc/courier/pop3d.pem
chown daemon /etc/courier/imapd.pem
chown daemon /etc/courier/pop3d.pem
```

We restart demons of Courier related to IMAP-SSL e POP3-SSL protocols:

```
/etc/init.d/courier-imap-ssl restart
/etc/init.d/courier-pop-ssl restart
```

We check if the connection through IMAP and POP3-SSL protocols run properly. To check the POP3-SSL protocol type you can execute:

```
openssl s_client -CAfile $CADIR/global_ca_public_cert.pem -connect mail.homeworks.it:995
```

```
CONNECTED(00000003)
depth=2 /C=IT/ST=Italy/O=Home Works S.p.A./OU=HomeWorks IT Department/CN=HomeWorks Root
CA/emailAddress=support@homeworks.it
verify return:1
depth=1 /C=IT/ST=Italy/L=Reggio Emilia/O=Home Works S.p.A./OU=HomeWorks IT
Department/CN=HomeWorks Issuing CA/emailAddress=support@homeworks.it
verify return:1
depth=0 /C=IT/ST=Italy/L=Reggio Emilia/O=Home Works S.p.A./OU=HomeWorks IT
Department/CN=mail.homeworks.it/emailAddress=postmaster@homeworks.it
verify return:1
---
Certificate chain
0 s:/C=IT/ST=Italy/L=Reggio Emilia/O=Home Works S.p.A./OU=HomeWorks IT
Department/CN=mail.homeworks.it/emailAddress=postmaster@homeworks.it
i:/C=IT/ST=Italy/L=Reggio Emilia/O=Home Works S.p.A./OU=HomeWorks IT Department/CN=HomeWorks
Issuing CA/emailAddress=support@homeworks.it
---
Server certificate
-----BEGIN CERTIFICATE-----
MIIF1DCCBlygAwIBAgIJAITSw4sZn+qIMA0GCSqGSIb3DQEJBQQUAMIG3MQswCQYD
VQQGEwJJVDEOMAwGA1UECBMFsXRhbHkxFjAUBgNVBACTDVJlZ2dpbyBFbWlsaWEx
GjAYBgNVBAoTEUhhbWUgV29ya3MgUy5wLkEuMSAwHgYDVQQLExdIb21lV29ya3Mg
SVQqRGVwYXJ0bWVudDEdMBsGA1UEAxMUSG9tZVdvcmtzIElzc3VpbmcgQ0ExIzAh
BgkqhkiG9wOBCQEFHFN1cHBvcnRAAG9tZXZxcmtzLml0MB4XDTA4MDUwMzE0MjEw
N1oXDTEyMDUwMjE0MjEwN1owgbcxCzAJBgNVBAYTAklUMQ4wDAYDVQQIEVJdGFs
eTEWMBQGA1UEBxMNUmVnZ21vIEVtaWxpYTEaMBGGA1UEChMRSG9tZSBXb3JrcyBT
LnAuQS4xIDAeBgNVBAstF0hvbWVXb3JrcyBJVCBEZXBhcnRtZW50MR0wGAYDVQQD
ExFtYWlsLmhvbWV3b3Jrcy5pdEmMCQGCsGSIb3DQEJARYXcG9zdG1hc3RlckBo
b21ld29ya3MuaXQwgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBALWZkmHfE0TF
pTdhUJsvbXFaty9CCUaVjHdSR/NPAGJs1xC6pamh2FfBNchVjelrxR5TiRpKneUI
Nc12qSuqXjdXt+N5LtJ8RY19pamwTyvZU02GW8qd/JhMla/ff7ZadrhRf2lrs7QI
7///6Xdl/v9IB4eY10lQrV6MJ03jY99AgMBAAGjggJjMIICXzAJBgNVHRMEAjAA
MBEGCWCGSAGG+EIBAQQEAwIGwDALBgNVHQ8EBAMCBPAwHQYDVR0lBBYwFAYIKwYB
BQUHAEGCCSGAQUBwMCMBOGA1UdDgQWBBQRPiCzc32TqWZfPQDtfvGbeEQCeODCB
QQYDVR0jBIHJMIHGgBSLG4DX0xig6QVQHP36Q2GQ/3+tRaGBoqSBnzCBnDELMaG
A1UEBhMCSVQxQjAMBgNVBAgTBu10Yw5MR0wGAYDVQQKEXFib21lIFdvcmtzIFMu
cC5BLjEgMB4GA1UECxmXSG9tZVdvcmtzIElUIERlcGFydG1lbnQxGjAYBgNVBAMT
EUhhbWVXb3JrcyBSb290IENBMSMwIQYJKoZIhvcNAQkBFhRzdXBwb3J0QGhvbWV3
b3Jrcy5pdIIJAIP0ryKAjVmbMD8GCCSGAQUBwEBBDMwMTAvBggrBgEFBQcwAoYj
aHR0cDovL3d3dy5ob21ld29ya3MuaXQvY2FpbmZvLmh0bWwOwYDVR0fBDQWwJAw
oC6gLIYqAHR0cDovL3d3dy5ob21ld29ya3MuaXQvY3JsL2lzc3VpbmdfY2EuY3Js
MIGBgNVHSAEgZkwgZYwgZMGDCsGAQQBg9Sg30BATCBGjA7BggrBgEFBQcCARYv
aHR0cDovL3d3dy5ob21ld29ya3MuaXQvY2EvaXNzdWluZl9jYV9jCHMuaHRtbDsw
QwYIKwYBBQUHAgIwNxo1SG9tZVdvcmtzIElzc3VpbmcgQ0EgQ2Vydg1maWNhdGlv
biBQcmFjdGJjZSBTdGF0ZW1lbnQwDQYJKoZIhvcNAQEFBQADggEBAH/h8+uQp+KX
0JvurS+4iIyJhMS60X4Hz/snbuTEZnJmbVRNM+OaZdV1G9enGLJ8iwhghyJvMJOI
JrYlWmcxd5SYYGmrAiGSSSvbpVg7M+g1I/AEa4gJraiOoiybBfWz5p18eIfveBNT
G+OA7YeFPDd6G+INTbtIRXsqCe3L63D/bl4oV5rgKKYOC+jnZW8TTCwLgOJ2p
buYq1+5nmqwdtw49weoXaLui0gQYxVFkg8Dq2KmDZkDB3guXbD9J4f3y8bzclAHS
laTE7L80s9Ba/Vxv/u02eXXCh2MpfDyCoQdNLOrQi+lySiFIRaYJPM3qIBHeBYHy
+NoEpj1Sags=
-----END CERTIFICATE-----
subject=/C=IT/ST=Italy/L=Reggio Emilia/O=Home Works S.p.A./OU=HomeWorks IT
Department/CN=mail.homeworks.it/emailAddress=postmaster@homeworks.it
issuer=/C=IT/ST=Italy/L=Reggio Emilia/O=Home Works S.p.A./OU=HomeWorks IT
Department/CN=HomeWorks Issuing CA/emailAddress=support@homeworks.it
---
No client certificate CA names sent
---
SSL handshake has read 1658 bytes and written 316 bytes
---
New, TLSv1/SSLv3, Cipher is AES256-SHA
Server public key is 1024 bit
Compression: NONE
Expansion: NONE
SSL-Session:
    Protocol : TLSv1
    Cipher : AES256-SHA
    Session-ID: ACE31A085ACDFADB6505B6C1CCB0B5754CF812D908A875F6BE1B403838E7BAA6
    Session-ID-ctx:
    Master-Key:
AD7FDC44B7F107F42807100F6BD64D7B9D73CF837CE614DD66E50FD79465B0B08110137894564BD5B390D2AB06491
9F5
    Key-Arg : None
    Start Time: 1209826648
    Timeout : 300 (sec)
```

```
Verify return code: 0 (ok)
---
+OK Hello there.
```

If the test report the message `Verify return code: 0 (ok)` it means that the connection is successful. We stop the test connection:

```
quit
+OK Better luck next time.
closed
```

To check the IMAP-SSL protocol type you can execute:

```
openssl s_client -CAfile $CADIR/global_ca_public_cert.pem -connect mail.homeworks.it:993

CONNECTED(00000003)
depth=2 /C=IT/ST=Italy/O=Home Works S.p.A./OU=HomeWorks IT Department/CN=HomeWorks Root
CA/emailAddress=support@homeworks.it
verify return:1
depth=1 /C=IT/ST=Italy/L=Reggio Emilia/O=Home Works S.p.A./OU=HomeWorks IT
Department/CN=HomeWorks Issuing CA/emailAddress=support@homeworks.it
verify return:1
depth=0 /C=IT/ST=Italy/L=Reggio Emilia/O=Home Works S.p.A./OU=HomeWorks IT
Department/CN=mail.homeworks.it/emailAddress=postmaster@homeworks.it
verify return:1
---
Certificate chain
0 s:/C=IT/ST=Italy/L=Reggio Emilia/O=Home Works S.p.A./OU=HomeWorks IT
Department/CN=mail.homeworks.it/emailAddress=postmaster@homeworks.it
i:/C=IT/ST=Italy/L=Reggio Emilia/O=Home Works S.p.A./OU=HomeWorks IT Department/CN=HomeWorks
Issuing CA/emailAddress=support@homeworks.it
---
Server certificate
-----BEGIN CERTIFICATE-----
MIIF1DCCBLYgAwIBAgIJIAITSw4sZn+qIMA0GCSqGSIb3DQEBBQUAMIG3MQswCQYD
VQQGEwJVVDEOMAwGA1UECBMFsXRhbHkxFjAUBgNVBACTDVJlZ2dpbyBFbWlsaWEx
GjAYBgNVBAoTEUhhbWUgV29ya3MgUy5wLkEuMSAwHgYDVQQLEXdIb21lV29ya3Mg
SVQqRGVWYXJ0bWVudDEdMBsGA1UEAxMUSG9tZVdvcmtzIElzc3VpbmcgQ0ExIzAh
BgkqhkiG9w0BCQEFWHN1cHBvcnRAAG9tZXZxdvcmtzLml0MB4XDTA4MDUwMzE0MjEw
N1oXDTEyMDUwMjE0MjEwN1owgbGcxZzAJBgNVBAYTAklUMQ4wDAYDVQQIEwVJdGFs
eTEWMBQGA1UEBxMNUmVnZ21vIEVtaWxpYTEaMBGGA1UEChMRSG9tZSBXb3JrcyBT
LnAuQ0s4IDAeBgNVBAsTF0hvbWVXb3JrcyBJVCBEZXhcnRtZW50MRowGAYDVQQD
ExFtYWlsLmhhbWVWV3b3Jrcy5pdDEmMCQGCSqGSIb3DQEJARYXcG9zdG1hc3RlckBo
b21ld29ya3MuaXQwZ28wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBALWZkmHfE0TF
pTdhUJsvbXFaty9CCUaVjHdSR/NPAGJslxC6pamh2FfBNchVjelrxR5TiRpKneUI
Ncl2qSuqXjdXt+N5LtJ8Ryi9pamwTyvZU02GW8qd/JhMla/ff7ZadRhRf2lrs7QI
7///6Xd1/v9IB4eYl0lQrvV6MJ03jY99AgMBAAGjggJjMIICXzAJBgNVHRMEAjAA
MBEGCWCsAGG+EIBAQQEAWIGwDALBgNVHQ8EBAMCBPAwHQYDVROlBBYwFAYIKwYB
BQUHAWEGCCsGAQUFBwMCMCB0GA1UdDgQWBBQRPiCzc32TqWZfPQDtfvGbeQCEODCB
0QYDVROlBjBIHJMTIHGgBSLG4DX0xiG6QVQHP36Q2GQ/3+tRaGBoqSBnzCBnDELMaK
A1UEBhMCSVQxZjAMBgNVBAgTBu10YXx5MRowGAYDVQQKEwFib21lIFdvcmtzIFMu
cC5BLjEgMB4GA1UECwMUSG9tZVdvcmtzIElUIERlcGFydG1lbnQxGjAYBgNVBAMT
EUhhbWVWV3b3JrcyBSb290IENBMSMwIQYJKoZIhvcNAQkBFhRzdXBwb3J0QGhvbWV3
b3Jrcy5pdIIJAIP0ryKAjVmbMD8GCCsGAQUFBwEBBDMwMTAvBggrBgEFBQcAwYj
aHR0cDovL3d3dy5ob21ld29ya3MuaXQvY2FpbmZvLmh0bWwwOwYDVROlBDBQwMjAw
oC6gLIYqaHR0cDovL3d3dy5ob21ld29ya3MuaXQvY3JsL2lzc3VpbmdfY2EuY3Js
MIGhBgNVHSAEGzkwZyYwZGMGDCsGAQQBg9Sg30BATCBGjA7BggrBgEFBQcCARYv
aHR0cDovL3d3dy5ob21ld29ya3MuaXQvY2EvaXNzdWluZl9jYV9jchMUAHRtbDsw
QWYIKwYBBQUHAgIwNxo1SG9tZVdvcmtzIElzc3VpbmcgQ0EgQ2VydGlmawNhdGlv
biBQcmFjdGJlZSBSTdGF0ZW1lbnQwDQYJKoZIhvcNAQEFBQADggEBAH/h8+uQp+KX
0JvurS+4iIyJhMS60X4Hz/snbuTEnzJmbVRNM+OaZdVlG9enGLJ8iwhghyJmJ0I
JrYlWmcxd5SYYGmrAiGSSSVbPvg7M+glI/AEa4gJraiOoiybBfWz5p18eIfveBNT
G+OATWOG1YeFDd6G+INTbtIRXsqCe3L63D/b14oV5rgKKYOC+jnZW8TTCwLgOJ2p
buYq1+5nmqwdtw49weoXaLui0gQYxVfKfg8Dq2KmDZkDB3guXbD9J4f3y8bZc1AHS
laTE7L80s9Ba/Vxv/u02eXXCh2MpfDyCoQdNLorQi+1YSiFIRaYJPM3qIBHeBYHy
+NoEpjlSaqs=
-----END CERTIFICATE-----
subject=/C=IT/ST=Italy/L=Reggio Emilia/O=Home Works S.p.A./OU=HomeWorks IT
Department/CN=mail.homeworks.it/emailAddress=postmaster@homeworks.it
issuer=/C=IT/ST=Italy/L=Reggio Emilia/O=Home Works S.p.A./OU=HomeWorks IT
Department/CN=HomeWorks Issuing CA/emailAddress=support@homeworks.it
---
No client certificate CA names sent
```

```

---
SSL handshake has read 1658 bytes and written 316 bytes
---
New, TLSv1/SSLv3, Cipher is AES256-SHA
Server public key is 1024 bit
Compression: NONE
Expansion: NONE
SSL-Session:
  Protocol : TLSv1
  Cipher : AES256-SHA
  Session-ID: ACE31A085ACDFADB6505B6C1CCB0B5754CF812D908A875F6BE1B403838E7BAA6
  Session-ID-ctx:
  Master-Key:
AD7FDC44B7F107F42807100F6BD64D7B9D73CF837CE614DD66E50FD79465B0B08110137894564BD5B390D2AB06491
9F5
  Key-Arg : None
  Start Time: 1209826648
  Timeout : 300 (sec)
  Verify return code: 0 (ok)
---
* OK [CAPABILITY IMAP4rev1 UIDPLUS CHILDREN NAMESPACE THREAD=ORDEREDSUBJECT THREAD=REFERENCES
SORT QUOTA IDLE AUTH=PLAIN ACL ACL2=UNION XCOURIEROUTBOX=INBOX.Outbox] Courier-IMAP ready.
Copyright 1998-2005 Double Precision, Inc. See COPYING for distribution information.

```

If the test report the message `Verify return code: 0 (ok)` it means that the connection is successful. We stop the test connection:

```

1 logout
* BYE Courier-IMAP server shutting down
1 OK LOGOUT completed
closed

```

At this point we can configure MTA client to connect to the server that hosts the program Courier (that is, *mail.homeworks.it*) using the IMAP-SSL ed POP3-SSL protocols.

Generate the digital certificate and private key of Apache

In general, for the program [Apache](#), it need create as many public/private key pairs as are secure sites hosted by the web server [Apache](#). For simplicity, in this article, we will take into account only the URLs *http://mail.homeworks.it* and *https://mail.homeworks.it* (in our case this URLs is that of the *webmail* associated with e-mail server *mail.homeworks.it* of Home Works S.p.A), so we will need to generate a single pair of public/private keys. To make the explanation even easier, we will suppose that the program which manages the *webmail* will be [SquirrelMail](#).

We create public/private key pairs to be assigned to Apache, taking care to specify as **Common Name** the FQDN *mail.homeworks.it*:

```

openssl req -new -nodes -keyout $CADIR/private/mail_private_key.pem -out
$CADIR/request/mail_public_key_req.pem -config $CADIR/ext/app_req.ext

Generating a 1024 bit RSA private key
.....++++++
.....++++++
writing new private key to '/usr/lib/ssl/misc/CA/private/mail_private_key.pem'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Organization Name (eg, your company) [Home Works S.p.A.]: Home Works S.p.A.

```

```
Internet Company Web Site [http://www.homeworks.it]: http://www.homeworks.it
Organizational Unit Name (eg, your department) [HomeWorks IT Department]: HomeWorks IT Department
FQDN host name (Common Name) []: mail.homeworks.it
Email Address (max 64 characters) [support@homeworks.it]: webmaster@homeworks.it
Locality Name (eg, city) [Reggio Emilia]: Reggio Emilia
State or Province Name (full name) [Italy]: Italy
Country Name (2 letter code) [IT]: IT

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password (between 6 and 20 characters) []: homeworks
```

We check that the public key `$CADIR/request/mail_public_key_req.pem`, has been created correctly:

```
openssl req -text -noout -in $CADIR/request/mail_public_key_req.pem
```

Before proceeding with the generation of the Apache's certificate, we must create the file with extensions X.509 to be applied to certificates. Therefore we will create the file [\\$CADIR/ext/web_server_x509_cert.ext](#):

```
touch $CADIR/ext/web_server_x509_cert.ext
vi $CADIR/ext/web_server_x509_cert.ext
```

We insert the following text:

```
# File /usr/lib/ssl/misc/CA/ext/web_server_x509_cert.ext

basicConstraints          = CA:false
nsComment                 = "Web Server Certificate"
nsCertType                = server, client
keyUsage                  = critical, digitalSignature, keyEncipherment
extendedKeyUsage          = serverAuth, clientAuth
subjectKeyIdentifier      = hash
authorityKeyIdentifier    = keyid, issuer:always
authorityInfoAccess       = caIssuers;URI:http://www.homeworks.it/ca/cainfo.html
crlDistributionPoints     = URI:http://www.homeworks.it/crl/issuing_ca.crl
certificatePolicies       = ia5org,@HomeWorks_CPS,@HomeWorks_Web_Server_CA_Policy

[ HomeWorks_CPS ]
policyIdentifier = 1.3.6.1.4.1.31012.1.1
CPS.1           = "http://www.homeworks.it/ca/homeworks_cps.html"
userNotice.1    = @HomeWorks_CPS_Notice

[ HomeWorks_CPS_Notice ]
explicitText     = "Home Works S.p.A. Certification Practice Statement"

[ HomeWorks_Web_Server_CA_Policy ]
policyIdentifier = 1.3.6.1.4.1.31012.3.2
userNotice.2    = @HomeWorks_Web_Server_CA_Notice

[ HomeWorks_Web_Server_CA_Notice ]
explicitText     = "Home Works S.p.A. Secure Communications Web Server Certificate Policy"

# End File
```

We digitally sign the just generated public key:

```
openssl ca -policy policy_anything -out $CADIR/certs/mail_public_cert.pem -extfile
$CADIR/ext/web_server_x509_cert.ext -infiles $CADIR/request/mail_public_key_req.pem

Using configuration from /etc/ssl/openssl.cnf
Check that the request matches the signature
Signature ok
Certificate Details:
```

```

Serial Number:
 84:d2:c3:8b:19:9f:ea:86
Validity
 Not Before: May 25 15:21:07 2008 GMT
 Not After : May 24 15:21:07 2012 GMT
Subject:
 organizationName = Home Works S.p.A.
 organizationName = http://www.homeworks.it
 organizationalUnitName = HomeWorks IT Department
 commonName = mail.homeworks.it
 emailAddress = webmaster@homeworks.it
 localityName = Reggio Emilia
 stateOrProvinceName = Italy
 countryName = IT
X509v3 extensions:
 X509v3 Basic Constraints:
  CA:FALSE
 Netscape Comment:
  Web Server Certificate
 Netscape Cert Type:
  SSL Client, SSL Server
 X509v3 Key Usage:
  Digital Signature, Key Encipherment
 X509v3 Extended Key Usage:
  TLS Web Server Authentication, TLS Web Client Authentication, Netscape Server Gated
Crypto
 X509v3 Subject Key Identifier:
 11:C4:FA:AE:CA:FD:4B:42:60:B7:D9:30:26:F3:11:A7:CE:DB:FD:DA
 X509v3 Authority Key Identifier:
 keyid:8B:1B:80:D7:D3:18:A0:E9:05:50:1C:FD:FA:43:61:90:FF:7F:AD:45
 DirName:/O=Home Works S.p.A./O=http://www.homeworks.it/OU=HomeWorks IT
Department/CN=HomeWorks Root CA/emailAddress=support@homeworks.it/L=Reggio
Emilia/ST=Italy/C=IT
 serial:F0:27:8F:E6:31:7D:C5:D7

 Authority Information Access:
  CA Issuers - URI:http://www.homeworks.it/ca/cainfo.html

 X509v3 CRL Distribution Points:
  URI:http://www.homeworks.it/crl/issuing_ca.crl

 X509v3 Subject Alternative Name:
  DirName:/CN=webmail.homeworks.it
 X509v3 Certificate Policies:
  Policy: HW-CPS
  CPS: http://www.homeworks.it/ca/homeworks_cps.html;
  User Notice:
  Explicit Text: Home Works S.p.A. Certification Practice Statement

  Policy: HW-TLS-WEB-Cert
  User Notice:
  Explicit Text: Home Works S.p.A. Secure Communications Web Server Certificate
Policy

Certificate is to be certified until May 24 15:21:07 2012 GMT (1460 days)
Sign the certificate? [y/n]: y

1 out of 1 certificate requests certified, commit? [y/n] y
Write out database with 1 new entries
Data Base Updated

```

We check that the Apache digital certificate has been generated correctly:

```
openssl x509 -text -noout -in $CADIR/certs/mail_public_cert.pem
```

We check if the Certification Path of the certificate `$CADIR/certs/mail_public_cert.pem`, is correct:

```
openssl verify -CAfile $CADIR/global_ca_public_cert.pem $CADIR/certs/mail_public_cert.pem
/usr/lib/ssl/misc/CA/certs/mail_public_cert.pem: OK
```

and verify if the just created certificate is a *SSL Server Certificate*:

```
openssl verify -purpose sslserver -CAfile $CADIR/global_ca_public_cert.pem
$CADIR/certs/mail_public_cert.pem
/usr/lib/ssl/misc/CA/certs/mail_public_cert.pem: OK
```

Once generated the Apache's certificate, we proceed with configuration of Apache itself. The digital certificate, `$CADIR/certs/mail_public_cert.pem` and its private key, `$CADIR/private/mail_private_key.pem`, must be copied on the folder `/etc/apache2/ssl/` on the web server whose hosting the site `http://mail.homeworks.it`. For example:

```
cp $CADIR/private/mail_private_key.pem /etc/apache2/ssl/
cp $CADIR/certs/mail_public_cert.pem /etc/apache2/ssl/
```

We enable Apache to listen on the SSL port, that **443**:

```
vi /etc/apache2/ports.conf
```

We append the entry **Listen 443**:

```
Listen 80
Listen 443
```

We allow the following Apache modules:

```
a2enmod ssl
a2enmod rewrite
```

We change the settings for the *Virtual Host* corresponding to URL `mail.homeworks.it` (that is for the *webamil* program [SquirrelMail](#)):

```
cp /etc/squirrelmail/apache.conf /etc/squirrelmail/apache.conf.originale
vi /etc/squirrelmail/apache.conf
```

We modify the configuration file of the Virtual Host of Webmail, as follows:

```
...
# users will prefer a simple URL like http://mail.example.com
# will be redirected to URL like https://mail.example.com
<VirtualHost mail.homeworks.it:80>
    DocumentRoot /usr/share/squirrelmail
    ServerAdmin webmaster@homeworks.it
    ServerName mail.homeworks.it
    RewriteEngine on
    RewriteCond %{SERVER_PORT} ^80$
    RewriteRule ^(.*)$ https://%{SERVER_NAME}$1 [L,R]
    RewriteLog "/var/log/apache2/rewrite.log"
    RewriteLogLevel 2
</VirtualHost>

# users will prefer a simple URL like https://mail.example.com
<VirtualHost mail.homeworks.it:443>
    DocumentRoot /usr/share/squirrelmail
    ServerAdmin webmaster@homeworks.it
    ServerName mail.homeworks.it
    SSLEngine on
    SSLCertificateFile /etc/apache2/ssl/mail_public_cert.pem
    SSLCertificateKeyFile /etc/apache2/ssl/mail_private_key.pem
</VirtualHost>
...
```

We make the site `mail.homeworks.it` available:

```
ln -s /etc/squirrelmail/apache.conf /etc/apache2/sites-available/squirrelmail.conf
a2ensite squirrelmail.conf
```

and then we restart Apache2 daemon:

```
/etc/init.d/apache2 restart
```

We check that everything functions properly:

```
openssl s_client -CAfile $CADIR/global_ca_public_cert.pem -connect mail.homeworks.it:443

CONNECTED(00000003)
depth=2 /C=IT/ST=Italy/O=Home Works S.p.A./OU=HomeWorks IT Department/CN=HomeWorks Root
CA/emailAddress=support@homeworks.it
verify return:1
depth=1 /C=IT/ST=Italy/L=Reggio Emilia/O=Home Works S.p.A./OU=HomeWorks IT
Department/CN=HomeWorks Issuing CA/emailAddress=support@homeworks.it
verify return:1
depth=0 /C=IT/ST=Italy/L=Reggio Emilia/O=Home Works S.p.A./OU=HomeWorks IT
Department/CN=mail.homeworks.it/emailAddress=webmaster@homeworks.it
verify return:1
---
Certificate chain
 0 s:/C=IT/ST=Italy/L=Reggio Emilia/O=Home Works S.p.A./OU=HomeWorks IT
Department/CN=mail.homeworks.it/emailAddress=webmaster@homeworks.it
 i:/C=IT/ST=Italy/L=Reggio Emilia/O=Home Works S.p.A./OU=HomeWorks IT
Department/CN=HomeWorks Issuing CA/emailAddress=support@homeworks.it
---
Server certificate
-----BEGIN CERTIFICATE-----
MIIE5zCCBFCgAwIBAgIJANGUmB8Nfv2UMA0GCSqGSIb3DQEBBQUAMIG3MQswCQYD
VQQGEwJJVDEOMAwGA1UECBMFsXRhbHkxZjAUBgNVBAcTDVJlZ2dpbyBFbWlsaWEEx
GjAYBgNVBAoTEUhhbWVudWUgV29ya3MgUy5wLkEuMSAwHgYDVQQLExdIb21lV29ya3Mg
SVQqRGVwYXJ0bWVudDEdMBsGA1UEAxMUSG9tZVdvcmtzIElzc3VpbmcgQ0ExIzAh
BgkqhkiG9w0BCQEFWHN1cHBvcnRaaG9tZXZvcmtzLml0MB4XDTA4MDQwNzAwMTA0
Ml0XDTEyMDQwNjAwMTA0Ml0wgbYxZCZAJBgNVBAYTAklUMQ4wDAYDVQQIEWVJdGFs
eTEWFMBQGA1UEBxMNUmVnZ21vIEVtaWxpYTEaMBGGA1UEChMRSG9tZSBXB3JrcyBT
LnAuQS4xIDAeBgNVBAStF0hvbWVXb3JrcyBJVCBEZXhBcnRtZW50MR0wGAYDVQOD
ExFtYWlsLmhvbWV3b3Jrcy5pdDElMCMGCSqGSIb3DQEJARYWd2VibWFzdG9yQGHv
bWV3b3Jrcy5pdDCBnzANBgkqhkiG9w0BAQEFAAOBjQAwYkCgYEAviI4YjyMdfmv
uPN9PCix76ip3xGzyA0tvIHTiGk7m+Zwn4wi2MGm4/iTQB8k6pgRZxWibj3/imeI
I9kptc9MELKhWrdQkAe8Fp2Nsmek6e3gkZfvFWYp9lNqrE0Jkoq8kIir1r/ukvL9
T966221DTvruHNYRHhf1bn1EJcL2GVcCAwEAAaOCAfgwgGH0MAkGAlUdEwQCMAAw
EQYIJZiAYb4QgEBBAQDAgZAMAsGA1UdDwQEAwIE8DAoBgNVHSUEITAfBggrBgEF
BQcDAQYIKwYBBQUHAwIGCWCgsSAGG+EIEATAdBgNVHQ4EFgQUdu/VgFhHDDxGEUG2
YhNCu6KvMp0wgDEGA1UdIwSByTCBxoAUReezp30mjPsHVnO2Yh0P62iOk9ehgaKk
gZ8wgZwxCzAJBgNVBAYTAklUMQ4wDAYDVQQIEWVJdGFseTEaMBGGA1UEChMRSG9t
ZSBXB3JrcyBTLnAuQS4xIDAeBgNVBAStF0hvbWVXb3JrcyBJVCBEZXhBcnRtZW50
MR0wGAYDVQODEXFi21lV29ya3MgUm9vdCBDQTEjMCEGCSqGSIb3DQEJARYUc3Vw
cG9ydEBob21ld29ya3MuaXSCCQDYFDG/DX79ijA/BggrBgEFBQcBAQQzMDEwLWYI
KwYBBQUHMAKGI2h0dHA6Ly93d3cuG9tZXZvcmtzLml0L2NhaW5mb5odG1sMDsG
A1UdHwQ0MDIwMKAuoCyGKmh0dHA6Ly93d3cuG9tZXZvcmtzLml0L2Nybc9pc3N1
aW5nX2NhLmNybdAsBgNVHREJTApCEwHzEdMBsGA1UEAxMud2VibWFpbC5ob21l
d29ya3MuaXQwDQYJKoZIhvcNAQEFBQADgYEAqGqEREgNX7bpMS1sX6Obt5v2j0pE
TFz06XquTEyBYdvnyJuFIF5h/gMcmX0qT7Ho/sGCu4l4qYYZhGzBYojk8dWVxHmg
B6zIx1lwuojUD+xgan/VvUESpKMPjwOgSwx5FRc7o0G1qlvyxsrVLVqS+yZp6I6
0r2+a5uD5g6UykG=
-----END CERTIFICATE-----
subject=/C=IT/ST=Italy/L=Reggio Emilia/O=Home Works S.p.A./OU=HomeWorks IT
Department/CN=mail.homeworks.it/emailAddress=webmaster@homeworks.it
issuer=/C=IT/ST=Italy/L=Reggio Emilia/O=Home Works S.p.A./OU=HomeWorks IT
Department/CN=HomeWorks Issuing CA/emailAddress=support@homeworks.it
---
No client certificate CA names sent
---
SSL handshake has read 1823 bytes and written 316 bytes
---
New, TLSv1/SSLv3, Cipher is DHE-RSA-AES256-SHA
Server public key is 1024 bit
Compression: NONE
Expansion: NONE
SSL-Session:
  Protocol : TLSv1
  Cipher : DHE-RSA-AES256-SHA
```

```
Session-ID: 0E39A40FD14A7BA11381059930D5A8EB2737AF24F7EB19B04AD502B8E2E36C27
Session-ID-ctx:
Master-Key:
729274FA1DAA1FC9EC5E80D61648E197975D30ED6EF9FB201DFED0C123DD7E8D3800CB3B1E66F5F3E267EC116B6A5
9FF
Key-Arg : None
Start Time: 1208012232
Timeout : 300 (sec)
Verify return code: 0 (ok)
---
```

If the test return the message `Verify return code: 0 (ok)` means that everything went well. We can therefore break the connection:

```
quit
closed
```

Now you can connect to site <http://mail.homeworks.it> with the secure HTTP (HTTPS) protocol version.

Generate the digital certificate to sign an email message

To sign an email message, we need use a digital certificate in the format [PKCS#12](#). To make the explanation simple, we will create a digital certificate to be assigned to **Postmaster HomeWorks** with email address: **postmaster@homeworks.it**, and then a digital certificate to be assigned to the person **Mario Rossi** with email address email: **mrossi@homeworks.it**. The certificate of **Postmaster HomeWorks** will expired in *four years*, while that for **Mario Rossi** will be valid for only *one year*.

We proceed to create the public/private key pairs to assign to Postmaster HomeWorks and Mario Rossi, using the same procedure followed so far, taking care to specify how *Common Name*, the sentence **Postmaster Home Works** and for *Email Address*, we specify the email **postmaster@homeworks.it**, while for **Mario Rossi**, the *Common Name* and the *Email Address* will be **Rossi Mario** and **mrossi@homeworks.it** respectively. We begin to create the Postmaster's public/private key pairs:

```
openssl req -new -nodes -keyout $CADIR/private/postmaster_private_key.pem -out
$CADIR/request/postmaster_public_key_req.pem

Generating a 1024 bit RSA private key
.....+++++
.....+++++
writing new private key to '/usr/lib/ssl/misc/CA/private/postmaster_private_key.pem'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
First Name (eg, Alessandro) []: Postmaster
Surname (eg, Tani) []: HomeWorks
Organization Name (eg, your company) [Home Works S.p.A.]: Home Works S.p.A.
Internet Company Web Site [http://www.homeworks.it]: http://www.homeworks.it
Organizational Unit Name (eg, your department) [HomeWorks IT Department]: HomeWorks IT
Department
Person Name (Common Name) []: Postmaster HomeWorks
Email Address (max 64 characters) [support@homeworks.it]: postmaster@homeworks.it
Locality Name (eg, city) [Reggio Emilia]: Reggio Emilia
State or Province Name (full name) [Italy]: Italy
Country Name (2 letter code) [IT]: IT

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password (between 6 and 20 characters) []: homeworks
```

We digitally sign the public key just created:

```
openssl ca -policy policy_anything -out $CADIR/certs/postmaster_public_cert.pem -infiles
$CADIR/request/postmaster_public_key_req.pem

Using configuration from /etc/ssl/openssl.cnf
Check that the request matches the signature
Signature ok
Certificate Details:
  Serial Number:
    84:d2:c3:8b:19:9f:ea:87
  Validity
    Not Before: May 26 21:28:20 2008 GMT
    Not After : May 25 21:28:20 2012 GMT
  Subject:
    name = Postmaster
    surname = HomeWorks
    organizationName = Home Works S.p.A.
    organizationName = http://www.homeworks.it
    organizationalUnitName = HomeWorks IT Department
    commonName = Postmaster HomeWorks
    emailAddress = postmaster@homeworks.it
    localityName = Reggio Emilia
    stateOrProvinceName = Italy
    countryName = IT
  X509v3 extensions:
    X509v3 Basic Constraints:
      CA:FALSE
    Netscape Comment:
      eMail Signing Encryption Certificate
    Netscape Cert Type:
      S/MIME
    X509v3 Key Usage: critical
      Digital Signature, Non Repudiation, Key Encipherment, Data Encipherment
    X509v3 Extended Key Usage:
      E-mail Protection
    X509v3 Subject Key Identifier:
      57:E9:23:6E:F4:D6:3D:11:BD:37:81:02:04:1E:D4:02:04:55:C3:EB
    X509v3 Authority Key Identifier:
      keyid:AC:47:B3:A7:7D:26:8C:FB:07:56:73:B6:62:1D:0F:EB:68:8E:93:D7
      DirName:/O=Home Works S.p.A./O=http://www.homeworks.it/OU=HomeWorks IT
      Department/CN=HomeWorks Root CA/emailAddress=support@homeworks.it/L=Reggio
      Emilia/ST=Italy/C=IT
      serial:F0:27:8F:E6:31:7D:C5:D7

  Authority Information Access:
    CA Issuers - URI:http://www.homeworks.it/ca/cainfo.html

  X509v3 CRL Distribution Points:
    URI:http://www.homeworks.it/crl/issuing_ca.crl

  X509v3 Certificate Policies:
    Policy: HW-CPS
      CPS: http://www.homeworks.it/ca/homeworks_cps.html
    User Notice:
      Explicit Text: Home Works S.p.A. Certification Practice Statement
    Policy: HW-MAIL-Cert
    User Notice:
      Explicit Text: Home Works S.p.A. Signature and Encryption Mail Certificate Policy

Certificate is to be certified until May 25 21:28:20 2012 GMT (1460 days)
Sign the certificate? [y/n]: y

1 out of 1 certificate requests certified, commit? [y/n] y
Write out database with 1 new entries
Data Base Updated
```

Then we create the [PKCS#12](#) certificate :

```
openssl pkcs12 -export -in $CADIR/certs/postmaster_public_cert.pem -inkey
$CADIR/private/postmaster_private_key.pem -certfile $CADIR/global_ca_public_cert.pem -name
"Postmaster HomeWorks Certificate" -out $CADIR/certs/postmaster_digital_sign_cert.pfx

Enter Export Password: homeworks
```

and finally we [import the certificate](#) `$CADIR/certs/postmaster_digital_sign_cert.pfx` within the mail client with you want to send emails to be digitally signed as **Postmatser Home Works**.

We create the digital certificate to be assigned to Mr. Mario Rossi. We proceed to create Mr. Rossi's public/private key pairs for:

```
openssl req -new -nodes -keyout $CADIR/private/rossi_mario_private_key.pem -out
$CADIR/request/rossi_mario_public_key_req.pem

Generating a 1024 bit RSA private key
.....+++++
.....+++++
writing new private key to '/usr/lib/ssl/misc/CA/private/rossi_mario_private_key.pem'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
First Name (eg, Alessandro) []: Mario
Surname (eg, Tani) []: Rossi
Organization Name (eg, your company) [Home Works S.p.A.]: Home Works S.p.A.
Internet Company Web Site [http://www.homeworks.it]: http://www.homeworks.it
Organizational Unit Name (eg, your department) [HomeWorks IT Department]: HomeWorks IT
Department
Person Name (Common Name) []: Rossi Mario
Email Address (max 64 characters) [support@homeworks.it]: mrossi@homeworks.it
Locality Name (eg, city) [Reggio Emilia]: Reggio Emilia
State or Province Name (full name) [Italy]: Italy
Country Name (2 letter code) [IT]: IT

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password (between 6 and 20 characters) []: mariorossi
```

We digitally sign the public key just generated, remember that the digital certificate of Mr. Rossi should be valid for only one year (`-days 365`):

```
openssl ca -policy policy_anything -days 365 -out $CADIR/certs/rossi_mario_public_cert.pem
-infiles $CADIR/request/rossi_mario_public_key_req.pem

Using configuration from /etc/ssl/openssl.cnf
Check that the request matches the signature
Signature ok
Certificate Details:
  Serial Number:
    84:d2:c3:8b:19:9f:ea:88
  Validity
    Not Before: May 26 21:49:19 2008 GMT
    Not After : May 25 21:49:19 2009 GMT
  Subject:
    name = Mario
    surname = Rossi
    organizationName = Home Works S.p.A.
    organizationName = http://www.homeworks.it
    organizationalUnitName = HomeWorks IT Department
    commonName = Rossi Mario
    emailAddress = mrossi@homeworks.it
    localityName = Reggio Emilia
    stateOrProvinceName = Italy
    countryName = IT
  X509v3 extensions:
    X509v3 Basic Constraints:
      CA:FALSE
    Netscape Comment:
      eMail Signing Encryption Certificate
    Netscape Cert Type:
```

```

S/MIME
X509v3 Key Usage:
    Digital Signature, Non Repudiation, Key Encipherment, Data Encipherment
X509v3 Extended Key Usage:
    E-mail Protection
X509v3 Subject Key Identifier:
    EC:34:1A:1B:69:07:16:71:8B:C5:3C:95:ED:FE:BA:09:41:16:F9:C9
X509v3 Authority Key Identifier:
    keyid:AC:47:B3:A7:7D:26:8C:FB:07:56:73:B6:62:1D:0F:EB:68:8E:93:D7
    DirName:/O=Home Works S.p.A./O=http://www.homeworks.it/OU=HomeWorks IT
Department/CN=HomeWorks Root CA/emailAddress=support@homeworks.it/L=Reggio
Emilia/ST=Italy/C=IT

    serial:F0:27:8F:E6:31:7D:C5:D7

Authority Information Access:
    CA Issuers - URI:http://www.homeworks.it/ca/cainfo.html

X509v3 CRL Distribution Points:
    URI:http://www.homeworks.it/crl/issuing_ca.crl

X509v3 Certificate Policies:
    Policy: HW-CPS
        CPS: http://www.homeworks.it/ca/homeworks_cps.html
    User Notice:
        Explicit Text: Home Works S.p.A. Certification Practice Statement
    Policy: HW-MAIL-Cert
    User Notice:
        Explicit Text: Home Works S.p.A. Signature and Encryption Mail Certificate Policy

Certificate is to be certified until May 25 21:49:19 2009 (365 days)
Sign the certificate? [y/n]: y

1 out of 1 certificate requests certified, commit? [y/n] y
Write out database with 1 new entries
Data Base Updated

```

We create the Mr Rossi's [PKCS#12](#) certificate :

```

openssl pkcs12 -export -in $CADIR/certs/rossi_mario_public_cert.pem -inkey
$CADIR/private/rossi_mario_private_key.pem -certfile $CADIR/global_ca_public_cert.pem -name
"Mario Rossi Certificate" -out $CADIR/certs/rossi_mario_digital_sign_cert.pfx

Enter Export Password: mariorossi
Verifying - Enter Export Password: mariorossi

```

Once achieved the Mario Rossi's [PKCS#12](#) certificate, we deploy this certificate to Mario Rossi.

Installation of certificates PKCS#12 on Thunderbird

For simplicity, **Postmaster HomeWorks**, and Mr. **Mario Rossi** use a Windows XP Professional as Operating System and as e-mail client the program [Thunderbird](#). The files **postmaster_digital_sign_cert.pfx** and **rossi_mario_digital_sign_cert.pfx** being copied, inside of the folder **C:\Certificates**. To import the certificate PKCS#12, on [Thunderbird](#) you can just follow the recipe below:

- start Thunderbird;
- open the **Tools -> Options... -> Advanced -> Certificates -> Manage Certificates...**;
- open the tab named **Your Certificates**;
- Click on **Import**;
- import the file **C:\Certificates\postmaster_digital_sign_cert.pfx** on the location in which it operates **Postmaster HomeWorks**, the file **C:\Certificates\rossi_mario_digital_sign_cert.pfx** on the location in which it operates **Mario Rossi**;
- click **OK**;
- in the **Options** menu, click **OK**;
- open the **Tools -> Account Settings**;

- go in the section for the account or **Postmaster HomeWorks** or **Mario Rossi** and go to **Security** tab;
- go on **Select**, and insert the certificate **Postmaster HomeWorks Certificate** or **Mario Rossi Certificate**, depending on which account you operate, in subsections **Digital Signing** and **Encryption**;
- select **Digitally sign message (by default)** and **Never (do not use encryption)**;
- click **OK**;
- try to send a message and check that the digital signature is correct;
- if you wish, at this point you can close Thunderbird.

In this way Thunderbird send e-mails of **Postmaster HomeWorks** or **Mario Rossi**, digitally signed.

Conclusions

The PKI configuration proposed, based on the program [OpenSSL](#), is suitable for a small reality business or for a small ISP, which must not generate more than one or two certificates each month. If the business needs require the provision of more digital certificates per day, should be better to use PKI systems more sophisticated than that exhibited in this article (especially if you want to automate issuing digital certificates themselves). However, leaving this out from what is the solution adopted, the mechanisms behind the administration of a PKI infrastructure are exactly those reported in this article. Finally, the procedures described in this article, can be easily automated creating the appropriate Bash script, for example, you may consult the article [How to Set Up an OpenSSL TEST CA](#).

Web Reference and Bibliography

For more information on digital certificates and PKI infrastructure, you can read following sites or documents:

- [X.509](#)
- [Certificate Revocation List \(CRL\)](#)
- [Public Key Cryptography Standards](#)
- [OpenSSL X509v3 Configuration](#)
- [RFC3280 - Certificate and Certificate Revocation List \(CRL\) Profile](#)
- [Network Security with OpenSSL](#)
- [Planning for PKI: Best Practices Guide for Deploying Public Key Infrastructure](#)
- [Object Identifiers \(OID\)](#)

May be particularly useful the following books:

- *Ralf Hildebrandt, Patrick Koetter, **The book of Postfix State-Of-The-Art Message Transport*** (No Starch Press, ISBN: 1-59327-001-1);
- *Russ Housley, Tim Polk **Planning for PKI, Best Practices Guide for Deploying Public Key Infrastructure*** (Wiley Networking Council Series);