Creazione di un'infrastruttura PKI con OpenSSL

Scritto da ***Iarno Pagliani*** (ipagliani@homeworks.it) ed ***Alessandro Tani*** (atani@homeworks.it)

- Pubblicato il 15 Giugno 2008 -

La sicurezza informatica è uno dei fattori strategici con cui oggi giorno, ogni amministratore di sistema e di rete si deve, prima o poi, confrontare. In questo articolo cercheremo di spiegare come realizzare un'infrastruttura PKI col programma OpenSSL della Debian (Etch). Illustreremo come creare i certificati digitali per rendere sicuri i collegamenti ai servizi SMTP, IMAP, POP3 e HTTP; vedremo poi come realizzare una struttura gerarchica di Certification Authority (CA) e come assegnare un certificato digitale ad un persona affinchè questi possa firmare digitalmente i propri messaggi di posta elettronica e farsi mandare dei messaggi criptati.

Licenza

Acronimi utilizzati

Nel corso dell'articolo o leggendo delle pubblicazioni che parlano di Public Key Certificate, si potrebbero incontrare i seguenti acronimi:

Acronimo	Descrizione
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

Acronimo	Descrizione
CSP	Cryptographic Service Provider
CSR	Certificate Signing Request
DER	Distinguished Encoding Rules
DSA	Digital Signature Algorithm
EDI	Electronic Data Interchang
LRA	Local Registration Authority
HSM	Hardware Security Module
IPRA	Internet Policy Registration Authority
ISP	Internet Service Provider
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

Un po' di teoria

Col termine **Public Key Infrastructure**, o più brevemente PKI, s'intende un sistema composto

logicamente dai seguenti componeneti:

- **Digital Certificates** (ovvero *chiavi pubbliche firmate digitalmente*);
- **Certification Authorities** (o più brevemente **CA**);
- **Certificate Revocation Lists** (o più brevemente **CRL**);

Un'infrastruttura di PKI permette di rendere sicure le seguenti operazioni:

- L'invio e la ricezione di e-mail.
- **Secure Web Comunication**. I server che offrono servizi Internet, come ad esempio i server di posta elettronica, possono autenticare i client (usando i certificati digitali dei client) e fornire comunicazioni confidenziali e criptate (utilizzando i certificati digitali del server).
- **Secure Web Site**. I siti Web possono utilizzare i certificati dei client per autenticare gli utenti e per controllare i loro diritti e permessi per accedere alle risorse dei siti Web.
- **Digital Signing of Software Files**.
- **Local Network Smart Card Authentication**. Il processo di logon via Kerberos può far uso dei certificati digitali e delle *chiavi private* imagazzinate in una *Smart Card* per autenticare un utente di rete quando questi tenta di fare logon.
- **Remote Access Smart Card Authentication**. I server che eseguono il *Routing and Remote Access Services* possono utilizzare i certificati e le *chiavi private* imagazzinate nelle *Smart Card* per autenticare gli utenti di rete quando questi eseguono il logon.
- **IPSec Authentication**.
- **Encrypting File System Recovery Agent**.
- **EAP-TLS Computer Authentication**. Attraverso il protocollo EAP-TLS risulta possibile stabilire quali postazioni di lavoro possono accedere o meno ad una rete.

Alla base del concetto di PKI, c'è il processo di criptazione denominato **Crittografia a Chiave Pubblica**. Questo processo di criptazione verte sull'idea di creare una coppia univoca di *chiavi pubbliche* e *private*. La *chiave pubblica* va resa nota al maggior numero possibile di persone (o sistemi), mentre la *chiave privata*, deve essere assolutamente confidenziale, nota solamente al *proprietario* della *chiave privata*. Affinchè però il sistema funzioni, si deve avere la *ragionevole certezza* che la *chiave pubblica* che si sta tentando di utilizzare sia effettivamente la *chiave pubblica* della persona o dell'applicazione con cui si desidera comunicare in modo sicuro. Per garantire questa *ragionevole certezza*, entrambe le parti, sia chi fornisce la *chiave pubblica* sia chi la utilizza, convengono di affidarsi ad una *terza parte*, la **Certification Authority** (o più brevemente **CA**), la quale ha il compito di *garantire* che la *chiave pubblica* affidata alla tale persona o alla tale applicazione sia effettivamente la *chiave pubblica* di quella persona o di quell'applicazione. Per stabilire questa *garanzia*, la **CA** provvede ad apporre, sulla *chiave pubblica*, un suo *sigillo*, od in altri termini, a *firmare digitalmente la chiave pubblica*. In questo modo, le parti coinvolte nel processo di comunicazione sicura, saranno in grado di individuare il *sigillo* della **CA** e quindi di avere una *ragionevole fiducia* che la *chiave pubblica* sia effettivamente autentica. Una *chiave pubblica* che è stata *firmata digitalmente*, prende il nome di **certificato digitale**. Ad ogni CA corrisponde uno ed uno solo *sigillo* e per ogni CA non ci possono essere due sigilli identici. Il *sigillo*, altri non è che la *chiave privata* della CA stessa.

Gli algoritmi con cui possono venire generate le coppie di *chiavi pubblico/private*, devono garantire la fondatezza delle seguenti due affermazioni:

- nota la chiave pubblica, non si può risalire da questa alla chiave privata e viceversa;
- cifrando un messaggio (o flusso di dati) con una soltando delle due chiavi, solamente l'altra chiave è in grado di decifrarlo.

Quando almeno una delle due affermazioni di sopra non è più vera, si dice che l'algoritmo di criptazione è stato violato e pertanto i certificati digitali basati su tale algoritmo non sono più validi. I certificati digitali che non sono più validi devono venire inseriti in una apposita lista chiamata **Certificate Revocation Lists** (o più brevemente **CRL**). Le CRL, affinchè siano utili, devono essere rese pubbliche, ovvero consultabili facilmente da coloro che voglio assicurarsi che un dato certificato digitale sia ancora valido. Da qualche anno, è stato reso disponibile un apposito *protocollo*, lo Online Certificate Status Protocol (o più brevemente **OCSP**), per facilitare la consultazione delle CRL (l'implementazione del protocollo OCSP non sarà oggetto di questo articolo). Uno degli algoritmi più utilizzati per generare le coppie di chivi pubbliche/private è l'algoritmo di Ron Rivest, Adi Shamir, e Leonard Adleman (**RSA**), pubblicato nel 1977 al MIT.

Ad ogni **CA** resta associata una coppia univoca di *chiavi pubbliche/private*. Questa coppia di chiavi pubblico/private, viene utilizzata per **firmare digitalmente** i certificati garantiti dalla **Certification Authority** (si osservi che il compito principale ci una CA, non è *fornire* le coppie di chiavi pubblico/private, ma di *garantire* l'autenticità delle chiavi pubbliche, fermo restando che una CA, è *anche* in grado di realizzare le coppie di chiavi pubblico/private). Col termine **firmare digitalmente** s'intende una procedura matematica di criptazione con la quale le chiavi pubbliche vengono criptate con la chiave privata di una data CA. Sfruttando la chiave pubblica della CA risulta possibile decriptare il certificato digitale, in questo modo si ha la certezza che la chiave pubblica è stata criptata e quindi garantita da quella CA.

Per **Digital Certificates** (Certificato Digitale) intenderemo tutte quelle *chiavi pubbliche* che sono state firmate digitalmente da una **Certification Authority**. All'interno di un certificato digitale vengono poi inserite una serie di informazioni cha hanno il compito di chiarire i seguenti aspetti:

- qual'è stata la CA che ha generato il certificato digitale;
- come è fatta la chiave pubblica da cui il certificato digitale ha avuto origine;
- qual'è l'utilizzo che se ne deve fare del certificato digitale;
- dove reperire la CRL che corrisponde alla CA che ha rilasciato il certificato digitale.

Per stabilire come le informazioni di sopra debbano venire inserite all'interno di un certificato digitale, sono state create delle opportune regole internazionali, queste regole vanno sotto il nome di **ITU-T Recommendation X.509**. Al momento in cui questo articolo viene scritto, esistono **tre versioni** di possibili certificati digitali, a seconda del tipo di informazioni che ciascun certificato contiene. Per scelta degli autori, in questo articolo verranno esposti solamente certificati conformi alla **terza versione**.

Grazie all'utilizzo delle chiavi pubblico/private e di opportuni algoritmi di criptazione a chiave simmetrica, denominati **Hash Funtions**, risulta possibile criptare in modo sicuro o un flusso di dati (*streaming*) o un file. Gli **Hash Funtions** sono particolari procedure matematiche che consentono di rendere *inalterabile* un flusso di dati o il contenuto di un file. Per raggiungere questo livello di affidabilità, il flusso di dati o il contenuto del file vengono opportunamente *criptati* secondo delle opportune procedure matematiche, il risultato di questa procedura di criptazione prendere il nome di **Message Digest**. Ad ogni flusso di dati o ad ogni file criptato corrisponde uno e uno solo **Message**

Digest, pertanto i **Message Digest** sono in corrispondenza *biunivoca* con i documenti originali da cui il **Message Digest** ha avuto origine. Questo fa in modo che se qualcuno o qualcosa tenta di alterare il documento originale (sia esso un file od un flusso di dati), il **Message Digest** risultante è completamente diverso da quello che si è ottenuto in precedenza dal documento originale stesso! In questo modo si può comprendere se un documento originale è stato manipolato o meno. Due fra i più usati **Hash Funtions** sono: Message Digest 5 (**MD5**) e Secure Hash Algorithm-1 (**SHA-1**).

Viene chiamato **Cryptographic Service Provider** (o più brevemente **CSP**) quel *software* che si preoccupa di generare la coppia *chiave pubblica* e *chiave privata* che vengono utilizzate dalle varie **Certification Authorities**. Nel nostro caso, il **CSP** sarà il programma OpenSSL che si trova all'interno di una distribuzione Debian (*Etch*).

Le CA hanno il compito di garantire che le coppie *chiavi pubbliche* e *chiavi private* affidate alle persone, o ai computer o a delle applicazioni siano affidabili ed autentiche. Le chiavi pubbliche vengono rilasciate sotto forma di **certificato digitale** e rese disponibili a tutti. Per assolvere questo compito, le CA vengono inseriti all'interno di un'apposita struttura gerarchica. Il legame che lega, ciascuna CA di una medesima struttura gerarchica, è la *fiducia*, ovvero le CA che si trovano nella parte più bassa della gerachia *credono* alle CA che si trovano nella parte più alta della gerarchia. La struttura di fiducia a cui le CA danno luogo è di tipo piramidale, ed in cima a questa piramide c'è la **Root Certification Authority** (o più brevemente *Root CA*). La Root CA è una *CA che crede a se stessa*, ovvero la coppia di chiavi pubbliche/private, che corrisponde alla Root CA, viene utilizzata per firmare digitalmente la chiave pubblica della CA stessa (*self-signed certificate*). Il legame di fiducia si manifesta nel fatto che le CA che si trovano nei livelli più bassi della scala gerarchica utilizzano coppie di chiavi pubbliche/private generate dalle CA che si trovano nel livello gerarchico immediatamente superiore al proprio. Per semplicità, comunque, non vengono generate strutture gerarchiche con più di tre livelli: al primo livello si trova la sola **Root CA**, nel secondo livello si trovano le **Certification Policy Certification Authority** (o più brevemente *Policy CA*), nel terza livello invece si trovano le **Issuing Certification Authority** (o più brevemente *Issuing CA*). Lo scopo della **Root CA** è quello di firmare (ed eventualmente fornire) le chiavi pubbliche a tutte le CA che compongono il secondo livello, ovvero le **Policy CA**. Compito delle **Policy CA** è quello di firmare (ed eventualmente fornire) le chiavi pubbliche a tutte le CA che compongono il terzo livello, ovvero le **Issuing CA**. Compito delle **Issuing CA** è quello di firmare (ed eventualmente fornire) le chiavi pubbliche degli utenti, dei computer o delle applicazioni.

Ma perchè realizzare una simile struttura di CA? Lo scopo ultimo di una simile infrastruttura è la sicurezza e l'affidabilità dei certificati digitali delle **Issuing CA**, tenendo presente che quest'ultime possono venire compromesse! Infatti, in una simile struttura gerarchica, la **Root CA** e le **Policy CA** non vengono minimamente coinvolte nel processo di rilascio dei certificati digitali da parte delle **Issuing CA**, per cui possono venire tranquillamente spente! Come si sà, una macchina spenta e scollegata dalla rete, gode del maggiore livello di sicurezza possibile contro le manomissioni da parte delle persone maleintenzionate. Inoltre, qualora una delle **Issuing CA** venisse compromessa, solamente i certificati digitali rilasciati da questa CA sarebbero da revocare, mentre l'intera archiettura sarebbe ancora valida, cioè le **Root CA** e le **Policy CA** non risulterebbero in alcun modo compromesse (erano spente!).

A livello pratico, si creano di solito strutture di CA a due livelli (composti solamente dalla **Root CA** e dalle **Issuing CA**), il terzo livello viene creato solamente se l'infrastruttura PKI di cui si ha bisogno è molto estesa a livello geografico. Si pensi ad esempio ad una multi-nazionale che ha sedi

in diverse nazioni e che per ciascuna nazione esistono diverse sedi assai affollate di personale. Se questa azienda dovesse dotarsi di una infrastruttura PKI, potrebbe scegliere di procedere come segue:

- La **Root CA** viene creata all'interno della rete della sede principale dell'azienda.
- Per ciascuna nazione, viene creata una **Policy CA**.
- Per ciascuna sede all'interno della stessa nazione, viene creata una **Issuing CA**.

Prende il nome di **Certificate Signing Request** la procedura con cui una CA genera la chiave pubblica che poi dovrà venire firmata digitalmente (ovviamente a questa chiave pubblica resterà associata, in modo univoco, una chiave privata).

Esistono infine, due formati con cui vengono creati i file che contengono le chiavi pubbliche e private. Il formato **PEM** ed il formato **DER**. Questi due formati sono molto simili, l'unica differenza è che nel formato **PEM** esiste una stringa di caratteri (tipicamente del tipo *"-----BEGIN CERTIFICATE REQUEST-----"* e *"-----END CERTIFICATE REQUEST-----"* o simili), denominata *intestazione*, con la quale viene identificata la parte del file che contiene o la chiave pubblica o la chiave privata. Nel formato **DER** queste stringhe di testo non esistono. Il formato **PEM** è il formato di default utilizzato dal programma **OpenSSL**.

Per maggiori informazioni sui certificati digitali e sulle infrastrutture PKI, si possono consultare i seguenti siti o documenti:

- X.509
- Certificate Revocation List (CRL)
- Public Key Cryptography Standards
- OpenSLL 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)

PKI con OpenSSL

Nel corso di questo articolo faremo vedere come realizzare una infrastruttura PKI basata sul programma OpenSSL che si trova all'interno di una distribuzione Debian (Etch). Per poter rendere più semplice la nostra esposizione, faremo finta di dover realizzare questa infrastruttura PKI per una azienda, chiamata Home Works S.p.A. Questa azienda ha sede in Reggio Emilia, ha un Dipartimento IT e sta realizzando la sua prima infrastruttura PKI interna. Scopo di questa PKI è quello di:

- creare i certificati digitali che dovranno venire utilizzati dalle persone della società, per firmare e criptare i propri messaggi di posta elettronica;
- creare i certificati digitali per consentire delle comunicazioni sicure con i protocolli SMTP (Postfix), IMAP e POP3 (Courier) e HTTP (Apache)

Poiché l'azienda Home Works S.p.A ha un unica sede di grande dimensioni e non ha sedi estere, converremo di creare una struttura PKI basata solamente su due livelli, la **Root CA** (o meglio la *HomeWorks Root CA*) ed una sola **Issuing CA** (ovvero la *HomeWorks Issuing CA*). La **Root CA**, una volta creato il certificato digitale da fornire alla **Issuing CA**, verrà spenta e sarà accesa solamente una volta all'anno per creare la sua CRL. Questa CRL sarà disponibile all'URL http://www.homeworks.it/crl/root_ca.crl.

Il certificato digitale della **Root CA** sarà valido per sedici anni (per certificati di durata superiore agli otto anni conviene utilizzare chiavi di lunghezza pari o superiore ai 2048 bits), mentre il certificato digitale della **Issuing CA** sarà valido per otto anni. I certificati digitali rilasciati dalla **Issuing CA** avranno la seguente durata:

- i certificati digitali rilasciati alle applicazioni, avranno la durata di quattro anni;
- i certificati digitali rilasciati agli utenti, avranno la durata di un anno.

La CRL della **Issuing CA** avrà la durata di trenta giorni, pertanto dovrà venire aggiornata almeno una volta ogni mese. La CRL della **Issuing CA** sarà reperibile all'URL http://www.homeworks.it/crl/issuing_ca.crl

Per convenzioni adottate dagli autori, tutti i file generati dal programma OpenSSL avranno estensione **.pem**, per ricordare che questi file sono formattati secondo lo standard **PEM**. Tutte le chiavi pubbliche non ancora firmate digitalmente, conterrano il suffisso **public_key_req**. Tutti i certificati digitali (ovvero le chiavi pubbliche firmate digitalmente), conterrano il suffisso **public_cert**. Tutte le chiavi private, invece, conterrano il suffisso **private_key**.

Generazione della Root Certification Authority con OpenSSL

Per prima cosa installiamo, qualora non fosse già presente, il programma OpenSSL:

```
apt-get install openssl
```

Prima di procedere però con la creazione della HomeWorks Root CA, dobbiamo definire la Dichiarazione di Pratica di Certificazione (*Certification Practice Statement*) relativa alla HomeWorks Root CA ed alla HomeWorks Issuing CA. In altri termini, dobbiamo da un lato indicare un documento in cui viene riportato lo scopo dei certificati forniti dalla HomeWorks Root CA e dalla HomeWorks Issuing CA, dall'altro richiedere un identificativo numerico (*Object Identifier*) per identificare in modo univoco i certificati forniti dalla HomeWorks Root CA e dalla HomeWorks Issuing CA (per maggiori informazioni sulla Dichiarazione di Pratica di Certificazione, si può consultare la RFC3647).

Per poter ottenere un identificativo numerico (*Object Identifier*) univoco con cui identificare i certificati digitali della HomeWorks Issuing CA, ci si può rivolgere allo IANA, compilando un apposito modulo elettronico. Nel caso della Home Works S.p.A. sono stati forniti i seguenti dati:

- **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

Per ricevere un identificativo numerico (*Object Identifier*) ci vogliono al massimo 60 giorni (anche se di solito la pratica viene evasa in circa una settimana). Una volta ottenuto il codice identificativo (*Object Identifier*) bisogna provvedere a registrarlo sul sito www.oid-info.com in questo modo sarà facilmente consultabile da coloro che desiderano avere informazioni sul proprietario del codice identificativo. Nel caso della Home Works S.p.A. il codice identificativo ottenuto è: 1.3.6.1.4.1.31012. A questo codice resta associato il ramo del Registration-Hierarchical-Name-Tree corrispondente alla Home Works S.p.A. Dal codice identificativo 1.3.6.1.4.1.31012 vengono poi generati dei sottocodici identificativi che hanno il compito di definire le politiche sui vari certificati digitali (*Certificate Policy*) firmati dalla HomeWorks Root CA e dalla HomeWorks Issuing CA. Per scelta degli autori di questo articolo, si è deciso di creare i seguenti sottocodici:

- **1.3.6.1.4.1.31012.1.1** si riferisce alla **Dichiarazione di Pratica di Certificazione** (Certification Practice Statement);
- **1.3.6.1.4.1.31012.1.2** si riferisce alle politiche di gestione dei certificati digitali rilasciati alle Certification Authority (CA), ovvero, nel nostro esempio, ai certificati forniti dalla HomeWorks Root CA alla HomeWorks Issuing CA;
- **1.3.6.1.4.1.31012.2.1** si riferisce alle politiche di gestione dei certificati digitali utilizzati dalle persone per firmare digitalmente i propri messaggi di posta elettronica;
- **1.3.6.1.4.1.31012.2.2** si riferisce alle politiche di gestione dei certificati digitali forniti alle persone che hanno la necessità di firmare digitalmente i documenti elettronici;
- **1.3.6.1.4.1.31012.3.1** si riferisce alle politiche di gestione dei certificati digitali utilizzati dai server di posta elettronica;
- **1.3.6.1.4.1.31012.3.2** si riferisce alle politiche di gestione dei certificati digitali utilizzati dai server Web, per rendere sicure le comunicazioni fra la postazione client ed il server Web;

A ciascuno di questi codici identificativi resterà associato un apposito documento (in formato HTML) che spiega nel dettaglio le varie politiche con le quali i certificati digitali vengono generati e rilasciati alle persone o alle applicazioni.

Il pacchetto OpenSSL della Debian mette a disposizione uno script Perl, `/usr/lib/ssl/misc/CA.pl`, per generare in modo semplice delle Certification Authority. Lo script `/usr/lib/ssl/misc/CA.pl` fa riferimento al file di configurazione `/etc/ssl/openssl.cnf` per procedere con la generazione della Certification Authority. Il file `/etc/ssl/openssl.cnf` presente nel pacchetto OpenSSL, costituisce un esempio, ma solamente un esempio, seppur perfettamente funzionante, su come si dovrebbero

creare sia le coppie di chiavi pubbliche e private sia i certificati. Per poter procedere alla creazione di una vera e propria CA, bisogna modificare opportunamente il file */etc/ssl/openssl.cnf*:

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

Modificare il file /etc/ssl/openssl.cnf come segue:

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

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

# OID Section
oid_section   = new_oids

# New OID for certificate (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 Mail Server Cert
HW-TLS-WEB-Cert  = 1.3.6.1.4.1.31012.3.2 # Secure Web Server Cert

# 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
crl_dir                 = $dir/crl
database                = $dir/index.txt # Database index file
unique_subject          = no
new_certs_dir           = $dir/newcerts
certificate             = $dir/root_ca_public_cert.pem
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
RANDFILE                = $dir/private/.rand # Private random number file
x509_extensions         = sub_ca_cert
name_opt                = ca_default     # Subject Name options
cert_opt                = ca_default     # Certificate field options
crl_extensions          = crl_ext        # CRL exstensions
default_days            = 2920           # How long to certify for
default_crl_days        = 365            # How long before next CRL
default_md              = sha1           # Which Hash Funtions to use
```

```
preserve                = no            # Keep passed DN ordering
policy                  = policy_match

# Extensions to add when Root CA creates an Subordinate Certificate CA
[ 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 exstensions
[ crl_ext ]
crlDistributionPoints   = URI:http://www.homeworks.it/crl/root_ca.crl

# Requirement for a new Private Key
[ req ]
dir                     = $ENV::CADIR
default_bits            = 2048
default_keyfile         = $dir/private/new_private_key.pem
distinguished_name      = req_distinguished_name
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 exstensions for a new Root CA Certificate Self Signed
[ 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
```

Più in generale, le parti:

```
...
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 Mail Server Cert
```

```
HW-TLS-WEB-Cert  = 1.3.6.1.4.1.31012.3.2 # Secure Web Server Cert
...
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
...
```

Vanno modificate con le informazioni relative alla particolare azienda a cui si sta provvedendo a realizzare l'infrastruttura PKI.

Per semplicità supporremo che l'utente che amministrerà la CA sia l'utente `root` (va ricordato che dato il ruolo delicato che hanno i server che ospitano le CA, conviene delegare i compiti amministrativi della CA ad un utente diverso da `root`), pertanto modifichiamo il file */root/.profile* (le impostazioni che andremo ad illustrare sono valide, però, per qualunque utente):

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

Aggiungiamo le seguenti variabili d'ambiente relative alla CA:

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

Carichiamo la nuova configurazione del file */root/.profile*:

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

Per impostazione predefinita, il comando */usr/lib/ssl/misc/CA.pl* crea *chiavi pubbliche* che hanno la validità di tre anni. Tre anni possono sembrare un valore ragionevole in molte circostanze, ma per evitare di creare inutili carichi amministrativi, modificheremo il file */usr/lib/ssl/misc/CA.pl* di modo da creare certificati digitali della durata di sedici anni (ovvero, 5840 giorni):

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

Le seguenti righe:

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

Diventano rispettivamente:

```
$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";
```

Eseguiamo una copia del file */usr/lib/ssl/misc/CA.pl* qualora, durante i processi di aggiornamento del pacchetto OpenSSL, possa venire accidentalmente sovrascritto:

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

Creiamo l'infrastruttura della CA (avendo modificato in modo opportuno il file */etc/ssl/openssl.cnf*, per molti campi relativi alla creazione della chiave pubblica e privata della HomeWorks Root CA, si possono lasciare i valori predefiniti), se non si specifica nulla all'interno dei vari campi che compongono le chiavi pubbliche/private e si preme il pulsante *Enter*, allora viene lasciato il valore predefinito, che è riportato tra le parentesi quadrate:

```
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/cakey.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
```

Nella compilazione di questi campi, si tenga presente che il campo **Certification Authority Name (Common Name)** non può contenere valori stringa più lunghi di 64 caratteri. Trattandosi di una Root CA, il campo **commonName** dovrebbe contenere un valore che riassume il ruolo della CA, nel nostro esempio: **HomeWorks Root CA**. Ovvero la Root CA della società Home Works S.p.A.

Verifichiamo che la cartella `/usr/lib/ssl/misc/CA` sia stata creata:

```
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.originale
-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
```

Se compare la cartella `/usr/lib/ssl/misc/CA` vuol dire che il processo di creazione della CA è andato a buon fine. La cartella `/usr/lib/ssl/misc/CA` presenta la seguente struttura:

```
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
```

Dove:

- `/usr/lib/ssl/misc/CA/root_ca_public_cert.pem` è il *certificato digitale* della HomeWorks Root CA.
- `/usr/lib/ssl/misc/CA/root_ca_public_key_req.pem` è la *chiave pubblica* della HomeWorks Root CA (ovvero il CSR).
- `/usr/lib/ssl/misc/CA/private/root_ca_private_key.pem` è la *chiave privata* della HomeWorks Root CA.
- La cartella `/usr/lib/ssl/misc/CA/certs` contiene tutti i *certificati* forniti dalla HomeWorks Root CA.
- La cartella `/usr/lib/ssl/misc/CA/private` contiene tutte le *chiavi private* dei vari certificati forniti.
- La cartella `/usr/lib/ssl/misc/CA/crl` contiene la copia aggiornata della *Certificate Revocation List* della HomeWorks Root CA.

Sistemiamo i permessi della cartella `$CADIR/private`:

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

Controlliamo che il *certificato digitale* della CA sia stato generato correttamente:

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

Controlliamo che la *chiave privata* della CA sia stata generata correttamente:

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

Convertiamo la chiave pubblica della CA nel formato compatibile con i sistemi Windows e Mac OS X:

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

A questo punto, su tutte le postazioni client che utilizzeranno i certificati rilasciati dalla HomeWorks Root CA, andrebbe importato il certificato `/usr/lib/ssl/misc/CA/root_ca_private_key.pem` se la postazione ha come sistema operativo Linux od Unix; il certificato `/usr/lib/ssl/misc/CA/root_ca_public_cert_windows_format.der` se ha come sistema operativo Windows o Mac OS X. Questi certificati digitali, che identificano la HomeWorks Root CA, saranno reperibili all'URL http://www.homeworks.it/ca/cainfo.html.

Prima di procedere con la creazione delle coppie di chiavi pubbliche/private da fornire alla HomeWorks Issuing CA, creiamo le cartelle `/usr/lib/ssl/misc/CA/ext` e `/usr/lib/ssl/misc/CA/request` La cartella `/usr/lib/ssl/misc/CA/ext` avrà il compito di contenere tutte le *estensioni* al file di configurazione `/etc/ssl/openssl.cnf`, mentre la cartella `/usr/lib/ssl/misc/CA/request` conterrà tutte le nuove chiavi pubbliche che dovranno poi venire firmate digitalmente (CSR):

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

Generazione dei certificati della HomeWorks Issuing CA

Una volta che la HomeWorks Root CA è operativa, possiamo procedere con la creazione dei certificati della HomeWorks Issuing CA. Iniziamo con la crezione della coppia di chiavi pubblica/privata da assegnare alla HomeWorks Issuing CA (l'opzione `req` consente la creazione di chiavi pubbliche conformi con la specifica 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
```

L'opzione `-nodes` consente di non specificare la *parola di accesso* con cui poi utilizzare il certificato, in questo modo risulterà più semplice provvedere a creare la CRL associata alla HomeWorks Issuing CA e più in generale i certificati rilasciati da questa CA.

Controlliamo che la chiave pubblica della Issuing CA, *$CADIR/request/issuing_ca_public_key_req.pem*, sia stata creata correttamente:

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

Generiamo il certificato da assegnare alla 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
```

Controlliamo che il certificato digitale della HomeWorks Issuing CA sia stato generato correttamente:

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

Convertiamo la chiave pubblica della HomeWorks Issuing CA nel formato compatibile con i sistemi Windows e 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
```

Su tutte le postazioni client che utilizzeranno i certificati rilasciati dalla HomeWorks Issuing CA, andrebbe importato il certificato `/usr/lib/ssl/misc/CA/certs/issuing_ca_public_cert.pem` se la postazione ha come sistema operativo Linux od Unix; il certificato `/usr/lib/ssl/misc/CA/certs/issuing_ca_public_cert_windows_format.der` se ha come sistema operativo Windows o Mac OS X. Questi certificati digitali, che identificano la HomeWorks Issuing CA, saranno reperibili all'URL http://www.homeworks.it/ca/cainfo.html.

Si osservi che affinchè i certificati rilasciati dalla HomeWorks Issuing CA vengano considerati *attendibili*, cioè degni di fiducia, è bene che su tutte le postazioni che dovranno utilizzare i certificati rilasciati dalla HomeWorks Issuing CA, siano installati sia il certificato della **HomeWorks Root CA**, sia il certificato della **HomeWorks Issuing CA**.

Installazione dei certificati digitali in Firefox e Thunderbird

Come esempio per assegnare i certificati della **HomeWorks Root CA** e della **HomeWorks Issuing CA** ad un client, prendiamo una postazione Windows XP con installato Firefox e Thunderbird. In questo caso dovremo prendere le versioni dei certificati della **HomeWorks Root CA** e della **HomeWorks Issuing CA** in formato **DER**:

- `/usr/lib/ssl/misc/CA/root_ca_public_cert_windows_format.der` è certificato digitale della **HomeWorks Root CA**;
- `/usr/lib/ssl/misc/CA/certs/issuing_ca_public_cert_windows_format.der` è certificato

digitale della **HomeWorks Issuing CA**.

Supporremo che la nostra postazione Windows di esempio abbia scaricato nella cartella **C:\Certificati** i due certificati di sopra, scaricandoli dal sito http://www.homeworks.it/ca/cainfo.html.

Per caricare i due certificati in Firefox basta procedere come segue:

- avviare Firefox;
- aprire il menù **Tools** e selezionare la voce **Options**;
- aprire la sezione **Advanced** e poi la sottosezione **Encryption**;
- controllare che siano selezionate le voci **Use SSL 3.0** e **Use TLS 1.0**;
- premere il pulsante **View Certificates**;
- andare nella sezione **Authorities**;
- premere il pulsante **Import**;
- importare prima il file **C:\Certificati\root_ca_public_cert_windows_format.der** e poi il file **C:\Certificati\issuing_ca_public_cert_windows_format.der**;
- premere il pulsante **OK** per confermare le modifiche apportate alla sezione **Authorities**;
- nella finestra dal titolo **Options**, premere il pulsante **OK**;
- se lo si desidera, a questo punto si può chiudere Firefox.

In questo modo Firefox riterrà attendibili tutti i certificati rilasciati dalla **HomeWorks Issuing CA**. Analogamente, per caricare i due certificati digitali della **HomeWorks Root CA** e della **HomeWorks Issuing CA** in Thunderbird, si può procedere come indicato di seguito:

- avviare Thunderbird;
- aprire il menù **Tools** e selezionare la voce **Options**;
- aprire la sezione **Advanced** e poi la sottosezione **Certificates**;
- premere il pulsante **View Certificates**;
- andare nella sezione **Authorities**;
- premere il pulsante **Import**;
- importare prima il file **C:\Certificati\root_ca_public_cert_windows_format.der** e poi il file **C:\Certificati\issuing_ca_public_cert_windows_format.der**;
- premere il pulsante **OK** per confermare le modifiche apportate alla sezione **Authorities**;
- nella finestra dal titolo **Options**, premere il pulsante **OK**;
- se lo si desidera, a questo punto si può chiudere Thunderbird.

In questo modo Thunderbird riterrà attendibili tutti i certificati rilasciati dalla **HomeWorks Issuing CA**.

Volendo, se l'infrastruttura PKI che si sta realizzando è pubblica, ovvero si desidera che i certificati rilasciati dalle Issuing CA siano fruibili al maggior numero di persone senza che questi provvedano a scaricare in autonomia i certificati digitali necessari per risolvere il *percorso di certificazione* (**Certification Path**) dei certificati forniti dalle Issuing CA, si può decidere di inserire le CA che compongono l'infrastruttura PKI, all'interno dell'elenco delle CA predefinite all'interno di Firefox e Thunderbird. Per poter inserire dei certificati digitali all'interno di Firefox a Thunderbird in modo predefinito, si deve fare in modo che l'infrastruttura PKI realizzata, soddisfi ai requisiti indicati nel documento Mozilla CA Certificate Policy.

Generazione della CRL della HomeWorks Root CA

Per portare a termine la creazione della HomeWorks Root CA, non resta che provvedere a generare la *Certificate Revocation List* (o più brevemente *CRL*) associata alla HomeWorks Root CA. Per creare la CRL, verrà creato un certificato digitale che poi sarà revocato. Per semplicità chiameremo questo certificato col nome di *$CADIR/certs/crl_public_cert.pem*. Pertanto, per prima cosa creiamo la coppia di chiavi pubblica/privata:

```
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
```

Firmiamo digitalmente la chiave pubblica *$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
```

Revochiamo il certificato *$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:
Revoking Certificate EDC9288FFA005870.
Data Base Updated
```

Generiamo la 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
```

Convertiamo la CRL nel formato DER:

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

A questo punto non resta che rendere pubblica la CRL provvedendo a renderla raggiungibile tramite lo URL http://www.homeworks.it/crl/root_ca.crl La CRL dovrà venire generata una volta all'anno (in base a quanto specificato dal parametro `default_crl_days` del file di configurazione */etc/ssl/openssl.cnf*), possibilmente una settimana prima della sua scadenza. Poiché la HomeWorks Root CA sarà per lo più spenta, non essendo coinvolta nella generazione dei certificati, l'operazione di generazione della CRL dovrà venire eseguita in modo manuale.

Si osservi che di solito le applicazioni non sono in grado di controllare, in modo automatico, le CRL. Bisogna pertanto configurare in modo opportuno ciascuna applicazione affinchè controlli la CRL associata alla CA che ha fornito il certificato (nel nostro aritcolo, tale CA, saranno la HomeWorks Root CA e la HomeWorks Issuing CA).

Rinominiamo il certificato *$CADIR/certs/crl_public_key_req.pem*, la sua chiave pubblica *$CADIR/private/crl_public_cert.pem* e la sua corrispondente chiave privata, *$CADIR/private/crl_private_key.pem*, aggiungendo il suffisso `.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
```

Creazione della HomeWorks Issuing CA

Una volta generata la chiave privata della HomeWorks Issuing CA ed il suo certificato digitale, si può procedere all'attivazione della HomeWorks Issuing CA. Per ovvi motivi di sicurezza, è bene che la HomeWorks Issuing CA sia un server diverso da quello che ospita la HomeWorks Root CA (infatti, il server che ospita la HomeWorks Root CA andrebbe spento ed acceso solamente per fare o manutenzione sul sistema, oppure per generare la CRL della HomeWorks Root CA). Pertanto si deve provvedere a spostare, in modo sicuro, la chiave privata ed il certificato della HomeWorks Issuing CA, dal server su cui è operativa la HomeWorks Root CA, al server su cui è operativa la HomeWorks Issuing CA (di solito l'utilizzo di una chiavetta USB va più che bene).

Sul server che ospita la HomeWorks Issuing CA vanno apportate delle modifiche al sistema del tutto simili a quelle effettuate per la HomeWorks Root CA. Anche in questo caso supporremo, per semplicità, che l'utente che amministrerà la CA sia l'utente `root` (anche in questo caso vale quanto detto per la HomeWorks Root CA, ovvero, data la delicatezza del ruolo di CA, i compiti amministrativi della CA andrebbero delegati ad un utente diverso da `root`).

Per prima cosa modifichiamo il file */root/.profile*:

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

Aggiungiamo le seguenti variabili d'ambiente relative alla CA:

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

Carichiamo la nuova configurazione del file */root/.profile*:

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

Sul server che ospita la HomeWorks Issuing CA si deve procedere a creare la seguente struttura di cartelle e di file:

```
/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
```

Per creare la struttura indicata procediamo come segue:

```
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
```

Per generare dei numeri casuali da mettere nel file *$CADIR/serial* si è utilizzato la funzione **Random Number Generator** del sito www.dnsstuff.com. Copiamo nella struttura appena creata i file *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*, avendo cura di:

- copiare i file *issuing_ca_public_cert_windows_format.der* e *issuing_ca_public_cert.pem* nella cartella */usr/lib/ssl/misc/CA*;
- copiare il file *issuing_ca_private_key.pem* nella cartella */usr/lib/ssl/misc/CA/private*

- copiare i file *root_ca_public_cert_windows_format.der* e *root_ca_public_cert.pem* nella cartella */usr/lib/ssl/misc/CA*;

Uniamo il certificato digitale della HomeWorks Root CA col certificato della HomeWorks Issuing CA, ottenendo il *certificato globale* dell'architettura PKI realizzata:

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

Esattamente come fatto per la HomeWorks Root CA, personalizziamo il file */etc/ssl/openssl.cnf*:

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

Modifichiamo il file /etc/ssl/openssl.cnf come segue:

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

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

# OID Section
oid_section    = new_oids

# New OID for certificate (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 Mail Server Cert
HW-TLS-WEB-Cert  = 1.3.6.1.4.1.31012.3.2 # Secure Web Server Cert

# 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
new_certs_dir      = $dir/newcerts     # Default place for new certs.
certificate        = $dir/issuing_ca_public_cert.pem # The CA Certificate
serial             = $dir/serial       # The current serial number
crlnumber          = $dir/crlnumber
crl                = $dir/crl/issuing_ca.crl # The current CRL
```

```
  private_key       = $dir/private/issuing_ca_private_key.pem
  RANDFILE          = $dir/private/.rand # Private random number file
  x509_extensions   = email_cert
  name_opt          = ca_default         # Subject Name options
  cert_opt          = ca_default         # Certificate field options
  crl_extensions    = crl_ext # Certificate Revocation List (CRL) exstensions
  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 Funtions 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 exstensions
  [ 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
  distinguished_name = req_distinguished_name_email
  attributes         = req_attributes # Certificate Version 3 extensions
  x509_extensions    = email_cert # The extentions 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
  ubjectKeyIdentifier    = 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
```

Più in generale, le parti:

```
  ...
  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
  ...
```

Vanno modificate con le informazioni relative alla particolare azienda a cui si sta provvedendo a realizzare l'infrastruttura PKI.

Nel realizzare il file di configurazione `/etc/ssl/openssl.cnf` si è presupposto che la HomeWorks Issuing CA fornirà, durante il suo normale ciclo di vita, molti più certificati per firmare digitalmente e criptare i messaggi di posta elettronica, rispetto a tutti gli altri tipi di certificati che la HomeWorks Issuing CA è in grado di erogare. Pertanto, il file di configurazione proposto, genera in modo *preferenziale* questo tipo di certificati.

Generazione della CRL della HomeWorks Issuing CA

Per portare a termine la creazione della HomeWorks Issuing CA, non resta che provvedere a generare la *Certificate Revocation List* ad essa associata. Per creare la CRL, verrà creato un certificato digitale che poi sarà revocato. Per semplicità chiameremo questo certificato col nome di `$CADIR/certs/crl_public_cert.pem`. Pertanto, per prima cosa creiamo la coppia di chiavi pubblica/

privata:

```
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
```

Firmiamo digitalmente la chiave pubblica `$CADIR/request/crl_public_key_req.pem`:

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

Revochiamo il certificato `$CADIR/certs/crl_public_cert.pem`:

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

Generiamo la CRL della HomeWorks Issuing CA:

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

Convertiamo la CRL nel formato DER:

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

Non resta che rendere pubblica la CRL provvedendo a renderla raggiungibile tramite lo URL http://www.homeworks.it/crl/issuing_ca.crl La CRL dovrà venire generata una volta al mese (in base a quanto specificato dal parametro `default_crl_days` del file di configurazione `/etc/ssl/openssl.cnf`), possibilmente una settimana prima della sua scadenza. Volendo si può

automatizzare la creazione della CRL ricorrendo ad un apposito script opportunamente pianificato. Ad esempio, un possibile script potrebbe essere:

```
  #!/bin/sh
  #
  # Creiamo la CRL
  openssl ca -gencrl -out $CADIR/crl/issuing_ca.crl
  #
  # Convertiamo la CRL dal formato PEM al formato DER
  openssl crl -in $CADIR/crl/issuing_ca.crl -out $CADIR/crl/issuing_ca.crl
-outform DER
```

Rinominiamo il certificato *$CADIR/certs/crl_public_cert.pem*, la chiave pubblica *$CADIR/request/crl_public_key_req.pem* e la sua corrispondente chiave privata, *$CADIR/private/crl_private_key.pem*, aggiungendo il suffisso `.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
```

A questo punto si può procedere ad abilitare le applicazioni che faranno uso dei certificati digitali autenticati dalla HomeWorks Issuing CA a controllare periodicamente la CRL fornita dalla HomeWorks Issuing CA.

Come abilitare il controllo delle CRL in Firefox e Thunderbird

Di solito le applicazioni che utilizzano i certificati digitali, non sono in grado di verificare, in modo automatico, le CRL associate a ciascun certificato digitale. Per poter abilitare il controllo periodico delle CRL, bisogna modificare manualmente la configurazione di queste applicazioni. Faremo vedere di seguito come abilitare il controllo delle CRL nei programmi Firefox e Thunderbird. Facendo riferimento all'architettura PKI realizzata in questo articolo, le CRL relative alla HomeWorks Root CA ed alla HomeWorks Issuing CA si trovano rispettivamente nei seguenti file (vale la pena osservare che prima di poter importare le CRL, bisogna installare i certificati digitali associati alla HomeWorks Root CA ed alla HomeWorks Issuing CA):

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

Per abilitare il controllo delle CRL sul programma Firefox basta procedere come indicato di seguito:

- avviare Firefox;
- aprire il menù **Tools** e selezionare la voce **Options**;
- aprire la sezione **Advanced** e poi la sottosezione **Encryption**;
- controllare che siano selezionate le voci **Use SSL 3.0** e **Use TLS 1.0**;
- premere il pulsante **Revocation Lists**;
- per aggiungere una CRL da controllare premere il pulsante **Import**;

- riportare nel campo **Import CRL from**, uno alla volta, i seguenti valori:
  - http://www.homeworks.it/crl/root_ca.crl
  - http://www.homeworks.it/crl/issuing_ca.crl
- una volta inserito uno dei valori di sopra, premere il pulsante **OK**. Per confermare il caricamento della CRL premere il pulsante **Yes**;
- selezionare la voce **Enable Automatic Update for this CRL**. Selezionare poi la voce **Update Day(s) before next Update date** ed impostare come numero di giorni prima della scadenza il valore **3**. Premere il pulsante **OK** per confermare;
- ripetere la procedura indicata sia per la CRL della HomeWorks Root CA, sia per la HomeWorks Issuing CA;
- premere **OK** per chiudere la finestra dal titolo **Manage CRLs**;
- tornati alla finestra dal titolo **Options**, premere il pulsante **Verification**;
- controllare che sia selezionata la voce **Do not use OCSP for certificate validation**, in alternativa si può selezionare la voce **Use OCSP to validate only certificate that specify an OCSP service URL**;
- premere **OK** per confermare la scelta adottata;
- chiudere la finestra dal titolo **Options** premendo il pulsante **OK**;
- se lo si desidera, a questo punto si può chiudere Firefox.

In questo modo Firefox provvederà a controllare con regolarità le CRL della HomeWorks Root CA e della HomeWorks Issuing CA. Analogamente, per Thunderbird basta procedere come indicato di seguito:

- avviare Thunderbird;
- aprire il menù **Tools** e selezionare la voce **Options**;
- aprire la sezione **Advanced** e poi la sottosezione **Certificates**;
- premere il pulsante **Revocation Lists**;
- per aggiungere una CRL da controllare premere il pulsante **Import**;
- riportare nel campo **Import CRL from**, uno alla volta, i seguenti valori:
  - http://www.homeworks.it/crl/root_ca.crl
  - http://www.homeworks.it/crl/issuing_ca.crl
- una volta inserito uno dei valori di sopra, premere il pulsante **OK**;
- per confermare il caricamento della CRL premere il pulsante **Yes**;
- selezionare la voce **Enable Automatic Update for this CRL**.
- per impostare la modalità di controllo della CRL appena importata, premere il pulsante **Settings**;
- controllare che sia selezionata la voce **Enable Automatic Update for this CRL**. Selezionare poi la voce **Update Day(s) before next Update date** ed impostare come numero di giorni prima della scadenza il valore **3**. Premere il pulsante **OK** per confermare;
- ripetere la procedura indicata sia per la CRL della HomeWorks Root CA, sia per la HomeWorks Issuing CA;
- premere **OK** per chiudere la finestra dal titolo **Manage CRLs**;
- tornati alla finestra dal titolo **Options**, premere il pulsante **Verification**;
- controllare che sia selezionata la voce **Do not use OCSP for certificate validation**, in alternativa si può selezionare la voce **Use OCSP to validate only certificate that specify an OCSP service URL**;
- premere **OK** per confermare la scelta adottata;
- chiudere la finestra dal titolo **Options** premendo il pulsante **OK**;

- se lo si desidera, a questo punto si può chiudere Thunderbird.

Generiamo il certificato digitale e la chiave privata di Postfix

Una volta creata la HomeWorks Issuing CA si può procedere con la generazione delle coppie di chiavi pubbliche/private da assegnare ai vari applicativi. Procediamo pertanto con la creazione della coppia di chiavi pubbliche e private di Postfix. Per scelta degli autori, tutte le chiavi pubbliche/private degli applicativi avranno durata pari a quattro anni. Nella generazione della coppia di chiavi pubbliche e private bisogna stare attenti ad inserire nel campo **Common Name** il nome FQDN del server di posta elettronica (che nel nostro caso supporremo essere *mail.homeworks.it*). Creaimo il file di configurazione dei certificati, $CADIR/ext/app_req.ext che dovranno venire assegnati alle applicazioni:

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

Inseriamo il testo seguente:

```
  # 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
```

Creiamo la coppia di chiavi pubbliche/private da assegnare a Postfix:

```
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
```

Controlliamo che la *chiave pubblica* di Postfix sia stata generata correttamente:

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

Prima di procedere con la generazione del certificato da assegnare a Postfix, dovremo creare il file con le estensioni X.509 da applicare al certificato stesso. Pertanto provvediamo a creare il file $CADIR/ext/mail_server_x509_cert.ext:

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

Inseriamo il testo seguente:

```
  # 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
```

Dopo di che procediamo a creare il certificato da assegnare a Postfix:

```
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
```

Verifichiamo se il *Certification Path* del certificato appena creato sia valido:

```
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
```

Oppure:

```
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
```

Verifichiamo se il certificato è un *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
```

Controlliamo il contenuto del certificato di Postfix:

```
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
```

Pertanto la coppia certificato digitale e chiave privata di Postfix, è formata dai file:

- `$CADIR/private/postfix_private_key.pem` è la *chiave privata* di Postfix;
- `$CADIR/certs/postfix_public_cert.pem` è il *certificato digitale* di Postfix.

Rendiamo disponibili a Postfix il certificato e la chiave privata che gli sono stati assegnati:

- il certificato di Postfix, `$CADIR/certs/postfix_public_cert.pem`, va copiato nella cartella `/etc/postfix/certs/` sul server dove Postfix è installato;
- la chiave privata di Postfix, `$CADIR/private/postfix_private_key.pem`, va copiata nella cartella `/etc/postfix/certs/` sul server dove Postfix è installato. L'accesso alla chiave privata di Postfix va limitato al solo utente `root`;
- il certificato globale della HomeWorks Root CA e HomeWorks Issuing CA, `$CADIR/global_ca_public_cert.pem`, va copiata nella cartella `/etc/postfix/certs/` sul server dove Postfix è installato;

Idealmente:

```
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/
```

Nella configurazione che adotteremo, imporremo a Postfix di considerare *attendibile* solamente la HomeWorks Issuing CA, ovvero, di ritenere validi solamente i certificati generati dalla HomeWorks Issuing CA. Pertanto le modifiche da apportare al file di configurazione di Postfix, `/etc/postfix/main.cf`, sono, lato SMTP server (ovvero quando Postfix si comporta come un server SMTP):

```
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
```

mentre lato SMTP client (ovvero quanto Postfix si comporta come un SMTP client) imporremo a Postfix di considerare attendibili tutti i certificati generati da CA pubbliche (l'elenco delle CA pubbliche da considerare attendibili si trova all'interno del 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
postconf -e smtp_tls_loglevel = 1
```

La configurazione relativa al protocollo Transport Layer Security (TLS) del file */etc/postfix/main.cf* dovrebbe apparire come:

```
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
...
```

Poichè nel processo di autenticazione TLS fra due server SMTP si possono verificare dei problemi, conviene creare un apposito file che inibisca l'utilizzo del protocollo TLS verso certi server SMTP. Creiamo a tale scopo il file */etc/postfix/deny_tls_per_domains*:

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

Inseriamo il seguente testo:

```
# 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
```

Creiamo la mappa di Postfix del file */etc/postfix/deny_tls_per_domains*:

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

Aggiungiamo la seguente riga al file di configurazione di Postfix, */etc/postfix/main.cf*:

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

Verifichiamo se la configurazione di Postfix risulta corretta, simulando una connessione TLS:

```
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
VQQGEwJJVDEOMAwGA1UECBMFSXRhbHkxFjAUBgNVBAcTDVJlZ2dpbyBFbWlsaWEx
GjAYBgNVBAoTEUhvbWUgV29ya3MgUy5wLkEuMSAwHgYDVQQLExdIb21lV29ya3Mg
SVQgRGVwYXJ0bWVudDEdMBsGA1UEAxMUSG9tZVdvcmtzIElzc3VpbmcgQ0ExIzAh
BgkqhkiG9w0BCQEWFHN1cHBvcnRAaG9tZXdvcmtzLml0MB4XDTA4MDMyOTAyMDgy
M1oXDTEyMDMyODAyMDgyM1owgbcxCzAJBgNVBAYTAklUMQ4wDAYDVQQIEwVJdGFs
eTEWMBQGA1UEBxMNUmVnZ2lvIEVtaWxpYTEaMBgGA1UEChMRSG9tZSBXb3JrcyBT
LnAuQS4xIDAeBgNVBAsTF0hvbWVXb3JrcyBJVCBEZXBhcnRtZW50MRowGAYDVQQD
ExFtYWlsLmhvbWV3b3Jrcy5pdDEmMCQGCSqGSIb3DQEJARYXcG9zdG1hc3RlckBo
b21ld29ya3MuaXQwgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBALGJI5ZBWk5V
/14bTChuPh8f3R91UW1bEUJze7IxWIXaSqmS9W2CnPXZ03MQmp0uMA/p/BnzaqZk
```

```
KoVowPid99+S3l0EmYLv0tbZ2ckVtHbbi9wP3ZXRHZUfZoa7ALhDoBrWH5TjyN6Q
nZ07tiqlE8PHCY55lbMRkbZW378fQ7z1AgMBAAGjggG/MIIBuzAJBgNVHRMEAjAA
MBEGCWCGSAGG+EIBAQQEAwIGwDALBgNVHQ8EBAMCBPAwHQYDVR0lBBYwFAYIKwYB
BQUHAwEGCCsGAQUFBwMCMB0GA1UdDgQWBBRBtnfg2nht4wSKlMDF6uYzT12U6TCB
0QYDVR0jBIHJMIHGgBSsR7OnfSaM+wdWc7ZiHQ/raI6T16GBoqSBnzCBnDELMAkG
A1UEBhMCSVQxDjAMBgNVBAgTBUl0YWx5MRowGAYDVQQKExFIb21lIFdvcmtzIFMu
cC5BLjEgMB4GA1UECxMXSG9tZVdvcmtzIElUIERlcGFydG1lbnQxGjAYBgNVBAMT
EUhvbWVXb3JrcyBSb290IENBMSMwIQYJKoZIhvcNAQkBFhRzdXBwb3J0QGhvbWV3
b3Jrcy5pdIIJANgUMb8Nfv2KMD8GCCsGAQUFBwEBBDMwMTAvBggrBgEFBQcwAoYj
aHR0cDovL3d3dy5ob21ld29ya3MuaXQvY2FpbmZvLmh0bWwwOwYDVR0fBDQwMjAw
oC6gLIYqaHR0cDovL3d3dy5ob21ld29ya3MuaXQvY3JsL2lzc3VpbmdfY2EuY3Js
MA0GCSqGSIb3DQEBBQUAA4GBAEm7cTPDfILe6tbHIwDMH+tY8s3KM2wFxdE10iAu
mXINBE6t5AshDdghHw/vjmWGPnt2Wh6mcGlckdrtXhwtal6q2Wgbf/1Z7PDfGBA3
KOt1t+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:CC5AACC3AC41023245CA2FAC724E08445A021CA53D5CCEAAC071FA936C723DF623CEDB72421AAC22CDC3
D71FE8E6CC58
    Key-Arg : None
    Start Time: 1207172342
    Timeout : 300 (sec)
    Verify return code: 0 (ok)
---
220 born.homeworks.it ESMTP Postfix (Debian/GNU)
```

Chiudiamo la connessione di test aperta dal comando `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
```

Se nel risultato della connessione TLS simulata compare il messaggio `Verify return code: 0 (ok)`, allora vuol dire che la configurazione di Postfix è corretta. A questo punto si possono configurare i client di posta elettronica di modo che possano collegarsi via TLS a Postfix, ovvero al server SMTP.

Generiamo il certificato digitale e la chiave privata di Courier

Courier è uno dei più popolari programmi di posta elettronica. Tra le sue caratteristiche c'è la possibilità di realizzare connessioni IMAP e POP3 sicure, IMAP-SSL e POP3-SSL. La configurazione base di Courier consente di creare in automatico i certificati digitali necessari per stabilire le connessioni IMAP-SSL e POP3-SSL, se però si vuole integrare il processo di comunicazione Transport Layer Security (TLS) all'interno di una infrastruttura PKI, si deve procedere alla generazione dei certificati da assegnare al programma Courier, dalle CA che compongono l'infrastruttura PKI. Nel nostro esempio, la CA predisposta alla generazione dei certificati è la HomeWorks Issuing CA. Affinchè, però, il programma Courier possa utilizzare i certificati generati da una CA, questi devono soddisfare alle seguenti condizioni:

- i certificati utilizzati dal programma Courier si devono chiamare nel seguente modo:
  - `/etc/courier/imapd.pem` per la connessione IMAP-SSL;
  - `/etc/courier/pop3d.pem` per la connessione POP3-SSL;
- entrambi i certificati sono l'unione del certificato digitale e della corrispondete chiave privata assegnata a Courier.

Iniziamo quindi con la generazione della coppia chiave pubblica/privata da assegnare a Courier (per semplicità espositiva, supporremo che Courier si trovi ad operare sullo stesso server su cui era installato Postfix, ovvero *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
```

Generiamo il certificato digitale da assegnare a Courier (si osservi che per questo certificato digitale

viene utilizzata l'opzione `-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
```

Controlliamo che il *certificato digitale* di Courier sia stato generato correttamente:

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

Verifichiamo se il Certification Path del certificato appena creato è valido:

```
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
```

Verifichiamo se il certificato digitale assegnato a Courier, è un 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
```

Concateniamo la chiave privata col certificato di Courier:

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

Inseriamo i parametri di Diffie-Hellman (questa operazione può richiedere qualche minuto):

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

Il risultato finale dovrebbe essere:

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

```
-----BEGIN RSA PRIVATE KEY-----
MIICXgIBAAKBgQDY2iG5Y03Fz/D2UwVO3hDu5vcu67PoiUZhSLZGufkFnxau2Imx
V6l6Xs8FJXGWpiPxgm/1FTvgqH1bLVJwVcAr54ESyfSys6WXV2jzghpTLKTME6QF
WUcun2+jDbotXWp4MrUBjfesKrpj9R8JKi/XxRMzoQTyl/YRb0wXwW06ZQIDAQAB
AoGAPJuy40rC+O+mbGJF0IY2e18oZP/Rt8NuXVBiSaA+3nhZcaLp0RwsLRyEhe6y
MaXb0+td+UTnCGJvLuWa7fS5kcfWrBzh11HlrtzM1O4AaVkqjZgbilG57EdwCouR
wEDoUQ4HL55MoFbglDgCVHWaHuRu21mjuSdngefBpNiQBcECQQD7TUIA9fo/zrlE
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
VQQGEwJJVDEOMAwGA1UECBMFSXRhbHkxFjAUBgNVBAcTDVJlZ2dpbyBFbWlsaWEx
GjAYBgNVBAoTEUhvbWUgV29ya3MgUy5wLkEuMSAwHgYDVQQLExdIb21lV29ya3Mg
SVQgRGVwYXJ0bWVudDEdMBsGA1UEAxMUSG9tZVdvcmtzIElzc3VpbmcgQ0ExIzAh
BgkqhkiG9w0BCQEWFHN1cHBvcnRAaG9tZXdvcmtzLml0MB4XDTA4MDQwMjIzMzU0
MFoXDTEyMDQwMTIzMzU0MFowgbcxCzAJBgNVBAYTAklUMQ4wDAYDVQQIEwVJdGFs
eTEWMBQGA1UEBxMNUmVnZ2lvIEVtaWxpYTEaMBgGA1UEChMRSG9tZSBXb3JrcyBT
LnAuQS4xIDAeBgNVBAsTF0hvbWVXb3JrcyBJVCBEZXBhcnRtZW50MRowGAYDVQQD
ExFtYWlsLmhvbWV3b3Jrcy5pdDEmMCQGCSqGSIb3DQEJARYXcG9zdG1hc3RlckBo
```

```
b21ld29ya3MuaXQwgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBANjaIbljTcXP
8PZTBU7eEO7m9y7rs+iJRmFItka5+QWfFq7YibFXqXpezwUlcZamI/GCb/UVO+Co
fVstUnBVwCvngRLJ9LKzpZdXaPOCGlMspMwTpAVZRy6fb6MNui1dangytQGN96wq
umP1HwkqL9fFEzOhBPKX9hFvTBfBbTplAgMBAAGjggG/MIIBuzAJBgNVHRMEAjAA
MBEGCWCGSAGG+EIBAQQEAwIGwDALBgNVHQ8EBAMCBPAwHQYDVR0lBBYwFAYIKwYB
BQUHAwEGCCsGAQUFBwMCMB0GA1UdDgQWBBT1O8U4nVC09c2hcDT6tG/Mtnx2oDCB
0QYDVR0jBIHJMIHGgBSsR7OnfSaM+wdWc7ZiHQ/raI6T16GBoqSBnzCBnDELMAkG
A1UEBhMCSVQxDjAMBgNVBAgTBUl0YWx5MRowGAYDVQQKExFIb21lIFdvcmtzIFMu
cC5BLjEgMB4GA1UECxMXSG9tZVdvcmtzIElUIERlcGFydG1lbnQxGjAYBgNVBAMT
EUhvbWVXb3JrcyBSb290IENBMSMwIQYJKoZIhvcNAQkBFhRzdXBwb3J0QGhvbWV3
b3Jrcy5pdIIJANgUMb8Nfv2KMD8GCCsGAQUFBwEBBDMwMTAvBggrBgEFBQcwAoYj
aHR0cDovL3d3dy5ob21ld29ya3MuaXQvY2FpbmZvLmh0bWwwOwYDVR0fBDQwMjAw
oC6gLIYqaHR0cDovL3d3dy5ob21ld29ya3MuaXQvY3JsL2lzc3VpbmdfY2EuY3Js
MA0GCSqGSIb3DQEBBQUAA4GBABi1cK2sE5PrPkM1AnalyPEGTLnODRfspRhGP/a4
4Cniqm1/htMZAJTgBPh+TC4Z2FteOTAHIfK3jqoHH8AL9UBfP7+swgygAyX4rJmv
Q+HUpPMTfwm3aj8NsNCe0jJYlA8+/t4XfX/cOM3yrQzkOVb2/zERXUhjcvoTXwMB
sQ4O
-----END CERTIFICATE-----
-----BEGIN DH PARAMETERS-----
MIGHAoGBAN/PC6aWXnCgNG/wnWcFxetEdym0+TLUBb24Xgmtm/n9TAR7++/zUtj9
3Bj98/I4byWk4CCj7cvl6uIA6hRt14HD1qEc2vOo9PUrz40zZnXrKPnCDyWGG0EO
aBZlI897f3HjXBaT45IchLDIGgO7lR4ekXG5FmRzaU+rqE7V/SEzAgEC
-----END DH PARAMETERS-----
```

A questo punto possiamo assegnare a Courier il suo certificato:

- il certificato di Courier, *$CADIR/certs/courier_cert.pem*, va copiato nella cartella */etc/courier/*, sul server dove Courier è installato, coi nome di *imapd.pem* e *pop3d.pem* rispettivamente;
- l'accesso ai certificati */etc/courier/imapd.pem* e */etc/courier/pop3d.pem* va limitato al solo utente `daemon`.

Idealmente:

```
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
```

Riavviamo i demoni di Courier relativi ai protocolli IMAP-SSL e POP3-SSL:

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

Verifichiamo che il collegamento tramite i protocolli IMAP-SSL ed POP3-SSL venga eseguito correttamente. Per controllare il protocollo POP3-SSL digitiamo:

```
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+qIMA0GCSqGSIb3DQEBBQUAMIG3MQswCQYD
VQQGEwJJVDEOMAwGA1UECBMFSXRhbHkxFjAUBgNVBAcTDVJlZ2dpbyBFbWlsaWEx
GjAYBgNVBAoTEUhvbWUgV29ya3MgUy5wLkEuMSAwHgYDVQQLExdIb21lV29ya3Mg
SVQgRGVwYXJ0bWVudDEdMBsGA1UEAxMUSG9tZVdvcmtzIElzc3VpbmcgQ0ExIzAh
BgkqhkiG9w0BCQEWFHN1cHBvcnRAaG9tZXdvcmtzLml0MB4XDTA4MDUwMzE0MjEw
N1oXDTEyMDUwMjE0MjEwN1owgbcxCzAJBgNVBAYTAklUMQ4wDAYDVQQIEwVJdGFs
eTEWMBQGA1UEBxMNUmVnZ2lvIEVtaWxpYTEaMBgGA1UEChMRSG9tZSBXb3JrcyBT
LnAuQS4xIDAeBgNVBAsTF0hvbWVXb3JrcyBJVCBEZXBhcnRtZW50MRowGAYDVQQD
ExFtYWlsLmhvbWV3b3Jrcy5pdDEmMCQGCSqGSIb3DQEJARYXcG9zdG1hc3RlckBo
b21ld29ya3MuaXQwgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBALWZkmHfE0TF
pTdhUJsvbXFaty9CCUaVjHdSR/NPAGJslxC6pamh2FfBNchVjelrxR5TiRpKneUI
Ncl2qSuqXjdXt+N5LtJ8RYi9pamwTyvZU02GW8qd/JhMla/ff7ZadrhRf2lrs7QI
7///6Xdl/v9IB4eYlOlQrvV6MJ03jY99AgMBAAGjggJjMIICXzAJBgNVHRMEAjAA
MBEGCWCGSAGG+EIBAQQEAwIGwDALBgNVHQ8EBAMCBPAwHQYDVR0lBBYwFAYIKwYB
BQUHAwEGCCsGAQUFBwMCMB0GA1UdDgQWBBQRPiCzc32TqWZfPQDtfVGbEQCeODCB
0QYDVR0jBIHJMIHGgBSLG4DX0xig6QVQHP36Q2GQ/3+tRaGBoqSBnzCBnDELMAkG
A1UEBhMCSVQxDjAMBgNVBAgTBUl0YWx5MRowGAYDVQQKExFIb21lIFdvcmtzIFMu
cC5BLjEgMB4GA1UECxMXSG9tZVdvcmtzIElUIERlcGFydG1lbnQxGjAYBgNVBAMT
EUhvbWVXb3JrcyBSb290IENBMSMwIQYJKoZIhvcNAQkBFhRzdXBwb3J0QGhvbWV3
b3Jrcy5pdIIJAIp0ryKAjVmbMD8GCCsGAQUFBwEBBDMwMTAvBggrBgEFBQcwAoYj
aHR0cDovL3d3dy5ob21ld29ya3MuaXQvY2FpbmZvLmh0bWwwOwYDVR0fBDQwMjAw
oC6gLIYqaHR0cDovL3d3dy5ob21ld29ya3MuaXQvY3JsL2lzc3VpbmdfY2EuY3Js
MIGhBgNVHSAEgZkwgZYwgZMGDCsGAQQBgc9Sg30BATCBgjA7BggrBgEFBQcCARYv
aHR0cDovL3d3dy5ob21ld29ya3MuaXQvY2EvaXNzdWluZ19jYV9jcHMuaHRtbDsw
QwYIKwYBBQUHAgIwNxo1SG9tZVdvcmtzIElzc3VpbmcgQ0EgQ2VydGlmaWNhdGlv
biBQcmFjdGljZSBTdGF0ZW1lbnQwDQYJKoZIhvcNAQEFBQADggEBAH/h8+uQp+KX
0JvurS+4iIyJhMS60X4Hz/snbuTEnZJmbVRNM+OaZdV1G9enGLJ8iwhghyjVmJ0I
JrYlWmcxd5SYYGmrAiGSSSvbpVg7M+g1I/AEa4gJraiOoiybBfWz5p18eIfveBNt
G+OA7WOGlYeFDd6G+INTbtIRXsqCe3L63D/bl4oV5rgKKYOC+jnZW8TTCwLgOJ2p
buYql+5nmqwdtw49weoXaLui0gQYxVFkg8Dq2KmDZkDB3guXbD9J4f3y8bZc1AHS
laTE7L80s9Ba/Vxv/u02eXXCh2MpfDyCoQdNLOrQi+lYSiFIRaYJPM3qIBHeBYHy
+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:
AD7FDC44B7F107F42807100F6BD64D7B9D73CF837CE614DD66E50FD79465B0B08110137894564BD5B390D2AB
064919F5
  Key-Arg : None
  Start Time: 1209826648
  Timeout : 300 (sec)
  Verify return code: 0 (ok)
---
+OK Hello there.
```

Se compare la sigla `Verify return code: 0 (ok)` vuol dire che il collegamento è andato a buon fine. Interrompiamo la connessione:

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

Per controllare il protocollo IMAP-SSL digitiamo:

```
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-----
MIIF1DCCBLygAwIBAgIJAITSw4sZn+qIMA0GCSqGSIb3DQEBBQUAMIG3MQswCQYD
VQQGEwJJVDEOMAwGA1UECBMFSXRhbHkxFjAUBgNVBAcTDVJlZ2dpbyBFbWlsaWEx
GjAYBgNVBAoTEUhvbWUgV29ya3MgUy5wLkEuMSAwHgYDVQQLExdIb21lV29ya3Mg
SVQgRGVwYXJ0bWVudDEdMBsGA1UEAxMUSG9tZVdvcmtzIElzc3VpbmcgQ0ExIzAh
BgkqhkiG9w0BCQEWFHN1cHBvcnRAaG9tZXdvcmtzLml0MB4XDTA4MDUwMzE0MjEw
N1oXDTEyMDUwMjE0MjEwN1owgbcxCzAJBgNVBAYTAklUMQ4wDAYDVQQIEwVJdGFs
eTEWMBQGA1UEBxMNUmVnZ2lvIEVtaWxpYTEaMBgGA1UEChMRSG9tZSBXb3JrcyBT
LnAuQS4xIDAeBgNVBAsTF0hvbWVXb3JrcyBJVCBEZXBhcnRtZW50MRowGAYDVQQD
ExFtYWlsLmhvbWV3b3Jrcy5pdDEmMCQGCSqGSIb3DQEJARYXcG9zdG1hc3RlckBo
b21ld29ya3MuaXQwgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBALWZkmHfE0TF
pTdhUJsvbXFaty9CCUaVjHdSR/NPAGJslxC6pamh2FfBNchVjelrxR5TiRpKneUI
Ncl2qSuqXjdXt+N5LtJ8RYi9pamwTyvZU02GW8qd/JhMla/ff7ZadrhRf2lrs7QI
7///6Xdl/v9IB4eYlOlQrvV6MJ03jY99AgMBAAGjggJjMIICXzAJBgNVHRMEAjAA
MBEGCWCGSAGG+EIBAQQEAwIGwDALBgNVHQ8EBAMCBPAwHQYDVR0lBBYwFAYIKwYB
BQUHAwEGCCsGAQUFBwMCMB0GA1UdDgQWBBQRPiCzc32TqWZfPQDtfVGbEQCeODCB
0QYDVR0jBIHJMIHGgBSLG4DX0xig6QVQHP36Q2GQ/3+tRaGBoqSBnzCBnDELMAkG
A1UEBhMCSVQxDjAMBgNVBAgTBUl0YWx5MRowGAYDVQQKExFIb21lIFdvcmtzIFMu
cC5BLjEgMB4GA1UECxMXSG9tZVdvcmtzIElUIERlcGFydG1lbnQxGjAYBgNVBAMT
EUhvbWVXb3JrcyBSb290IENBMSMwIQYJKoZIhvcNAQkBFhRzdXBwb3J0QGhvbWV3
b3Jrcy5pdIIJAIp0ryKAjVmbMD8GCCsGAQUFBwEBBDMwMTAvBggrBgEFBQcwAoYj
```

```
aHR0cDovL3d3dy5ob21ld29ya3MuaXQvY2FpbmZvLmh0bWwwOwYDVR0fBDQwMjAw
oC6gLIYqaHR0cDovL3d3dy5ob21ld29ya3MuaXQvY3JsL2lzc3VpbmdfY2EuY3Js
MIGhBgNVHSAEgZkwgZYwgZMGDCsGAQQBgc9Sg30BATCBgjA7BggrBgEFBQcCARYv
aHR0cDovL3d3dy5ob21ld29ya3MuaXQvY2EvaXNzdWluZ19jYV9jcHMuaHRtbDsw
QwYIKwYBBQUHAgIwNxo1SG9tZVdvcmtzIEl1c3VpbmcgQ0EgQ2VydGlmaWNhdGlv
biBQcmFjdGljZSBTdGF0ZW1lbnQwDQYJKoZIhvcNAQEFBQADggEBAH/h8+uQp+KX
0JvurS+4iIyJhMS60X4Hz/snbuTEnZJmbVRNM+OaZdV1G9enGLJ8iwhghyjVmJ0I
JrYlWmcxd5SYYGmrAiGSSSvbpVg7M+g1I/AEa4gJraiOoiybBfWz5p18eIfveBNt
G+OA7WOGlYeFDd6G+INTbtIRXsqCe3L63D/bl4oV5rgKKYOC+jnZW8TTCwLgOJ2p
buYql+5nmqwdtw49weoXaLui0gQYxVFkg8Dq2KmDZkDB3guXbD9J4f3y8bZc1AHS
laTE7L80s9Ba/Vxv/u02eXXCh2MpfDyCoQdNLOrQi+lYSiFIRaYJPM3qIBHeBYHy
+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:
AD7FDC44B7F107F42807100F6BD64D7B9D73CF837CE614DD66E50FD79465B0B08110137894564BD5B390D2AB
064919F5
  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.
```

Se compare la sigla `Verify return code: 0 (ok)` vuol dire che il collegamento è andato a buon fine. Interrompiamo la connessione:

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

A questo punto possiamo collegarci al server che ospita il programma Courier (ovvero, *mail.homeworks.it*) utilizzando i protocolli IMAP-SSL ed POP3-SSL.

Generiamo il certificato digitale e la chiave privata di Apache2

In generale, per il programma Apache2, vanno create tante coppie di chiavi pubbliche/private, quanti sono i siti sicuri ospitati dal server web su cui il programma Apache2 è installato. Per semplicità, in questo articolo, prenderemo solamente in considerazione gli URL *http://mail.homeworks.it* e *https://mail.homeworks.it* (ovvero la *webmail* associata al server di posta elettronica *mail.homeworks.it* della società Home Works S.p.A), per cui avremo bisogno di generare un'unica coppia di chiavi pubbliche/private. Per rendere la spiegazione ancora più semplice, supporemo che il programma che gestisce la *webmail* della società Home Works S.p.A, sia il programma SquirrelMail.

Generiamo la coppia di chiavi pubbliche/private da assegnare al programma Apache2, avendo cura di specificare come **Common Name** il nome 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
```

Controlliamo che la chiave pubblica `$CADIR/request/mail_public_key_req.pem`, sia stata creata correttamente:

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

Prima di procedere con la generazione del certificato da assegnare ad Apache2, dovremo creare il file con le estensioni X.509 da applicare al certificato stesso. Pertanto provvediamo a creare il file $CADIR/ext/web_server_x509_cert.ext:

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

Inseriamo il testo seguente:

```
  # 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
```

Firmiamo digitalmente la chiave pubblica generata:

```
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
```

Controlliamo che il certificato digitale da assegnare ad Apache2 per la *webmail* sia stato generato correttamente:

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

Verifichiamo che il Certification Path del certificato `$CADIR/certs/mail_public_cert.pem` sia corretto:

```
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
```

Verifichiamo se il certificato sia un *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
```

Una volta generati i certificati da assegnare al programma Apache2, procediamo con la sua configurazione. Supporremo pertanto che il certificato digitale, `$CADIR/certs/mail_public_cert.pem` e la sua chiave privata, `$CADIR/private/mail_private_key.pem`, siano stati copiati sul server web che ospita il sito *http://mail.homeworks.it*, nella cartella `/etc/apache2/ssl/`, idealmente:

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

Per prima cosa mettiamo in ascolto il programma Apache2 sulla porta 443:

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

Aggiungiamo la voce Listen 443:

```
Listen 80
Listen 443
```

Abilitiamo i seguenti moduli di Apache2:

```
a2enmod ssl
a2enmod rewrite
```

Modifichiamo le impostazioni relative al *Virtual Host* corrispondente al URL *mail.homeworks.it* (come accennato in precedenza, supporremo che la *webamil* sia fornita dal programma SquirrelMail):

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

Modifichiamo il file di configurazione del Virtual Host della Webmail, come segue:

```
...
# 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>
...
```

Rendiamo disponibile il sito relativo alla Webmail:

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

Abilitiamo il sito relativo alla Webmail:

```
a2ensite squirrelmail.conf
```

Forziamo la riesecuzione di Apache2:

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

Controlliamo che tutto funzioni correttamente:

```
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
VQQGEwJJVDEOMAwGA1UECBMFSXRhbHkxFjAUBgNVBAcTDVJlZ2dpbyBFbWlsaWEx
GjAYBgNVBAoTEUhvbWUgV29ya3MgUy5wLkEuMSAwHgYDVQQLExdIb21lV29ya3Mg
SVQgRGVwYXJ0bWVudDEdMBsGA1UEAxMUSG9tZVdvcmtzIEl1zc3VpbmcgQ0ExIzAh
BgkqhkiG9w0BCQEWFHN1cHBvcnRAaG9tZXdvcmtzLml0MB4XDTA4MDQwNzAwMTA0
MloXDTEyMDQwNjAwMTA0MlowgbYxCzAJBgNVBAYTAklUMQ4wDAYDVQQIEwVJdGFs
eTEWMBQGA1UEBxMNUmVnZ2lvIEVtaWxpYTEaMBgGA1UEChMRSG9tZSBXb3JrcyBT
LnAuQS4xIDAeBgNVBAsTF0hvbWVXb3JrcyBJVCBEZXBhcnRtZW50MRowGAYDVQQD
ExFtYWlsLmhvbWV3b3Jrcy5pdDElMCMGCSqGSIb3DQEJARYWd2VibWFzdGVyQGhv
bWV3b3Jrcy5pdDCBnzANBgkqhkiG9w0BAQEFAAOBjQAwgYkCgYEAviI4YjyMdfmv
uPN9PCix76ip3xGzyAOtvIHTiGk7m+Zwn4wi2MGm4/iTQB8k6pgRZxWibj3/imei
I9kptc9MELKhwRdQkAe8Fp2Nsmek6e3gkZfvFWYp9lNqrE0Jkoq8kIir1r/ukvL9
T96622lDTvruHNYRHhf1bn1EJcL2GVcCAwEAAaOCAfgwggH0MAkGA1UdEwQCMAAw
EQYJYIZIAYb4QgEBBAQDAgZAMAsGA1UdDwQEAwIE8DAoBgNVHSUEITAfBggrBgEF
```

```
BQcDAQYIKwYBBQUHAwIGCWCGSAGG+EIEATAdBgNVHQ4EFgQUdu/VgFhHDDxGEUG2
YhNCu6KvMp0wgdEGA1UdIwSByTCBxoAUrEezp30mjPsHVnO2Yh0P62iOk9ehgaKk
gZ8wgZwxCzAJBgNVBAYTAklUMQ4wDAYDVQQIEwVJdGFseTEaMBgGA1UEChMRSG9t
ZSBXb3JrcyBTLnAuQS4xIDAeBgNVBAsTF0hvbWVXb3JrcyBJVCBEZXBhcnRtZW50
MRowGAYDVQQDExFIb21lV29ya3MgUm9vdCBDQTEjMCEGCSqGSIb3DQEJARYUc3Vw
cG9ydEBob21ld29ya3MuaXSCCQDYFDG/DX79ijA/BggrBgEFBQcBAQQzMDEwLwYI
KwYBBQUHMAKGI2h0dHA6Ly93d3cuaG9tZXdvcmtzLml0L2NhaW5mby5odG1sMDsG
A1UdHwQ0MDIwMKAuoCyGKmh0dHA6Ly93d3cuaG9tZXdvcmtzLml0L2NybC9pc3N1
aW5nX2NhLmNybDAsBgNVHREEJTAjpCEwHzEdMBsGA1UEAxMUd2VibWFpbC5ob21l
d29ya3MuaXQwDQYJKoZIhvcNAQEFBQADgYEAqGqEREGNX7bpMS1sX6Obt5v2j0pE
TFz06XquTEyBYdvnyJuFIF5h/gMcmX0qT7Ho/sGCu4l4qYYZhGzBYojk8dWVxHmg
B6zIx1lwuojUD+xgan/VvUEspKMPjwOgSwx5FRc7o0Glqlyvyxsr VLVqS+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:
729274FA1DAA1FC9EC5E80D61648E197975D30ED6EF9FB201DFED0C123DD7E8D3800CB3B1E66F5F3E267EC11
6B6A59FF
  Key-Arg : None
  Start Time: 1208012232
  Timeout : 300 (sec)
  Verify return code: 0 (ok)
---
```

Se compare la sigla `Verify return code: 0 (ok)` vuol dire che tutto è andato bene. Possiamo quindi interrompere il collegamento:

```
quit
closed
```

A questo punto si potrà collegarsi al sito *http://mail.homeworks.it* in modalità sicura (via HTTPS).

Generiamo il certificato digitale per firmare digitalmente un messaggio di posta elettronica

Per poter firmare digitalmente un messaggio di posta elettronica, bisogna utilizzare un certificato digitale nel formato PKCS#12. Per rendere la spiegazione più semplice, supporremo di creare prima un certificato digitale da assegnare a **Postmaster HomeWorks** che ha indirizzo email **postmaster@homeworks.it**, successivamente un certificato digitale da assegnare alla persona

Mario Rossi che supporremo avere indirizzo email **mrossi@homeworks.it**. Il certificato di **Postmaster HomeWorks** avrà la durata di *quattro anni*, mentre quello per **Mario Rossi** avrà la durata di *un solo anno*.

Procediamo a creare la prima coppia di chiavi pubbliche/private utilizzando la stessa procedura seguita sinora, avendo cura di specificare come *Common Name*, la sigla **Postmaster Home Works** e come *Email Address*, l'indirizzo **postmaster@homeworks.it**:

```
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
```

Firmiamo digitalmente la chiave pubblica appena creata:

```
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
```

Generiamo il certificato PKCS#12:

```
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
Verifying - Enter Export Password: homeworks
```

A questo punto non resta che importare il certificato `$CADIR/certs/postmaster_digital_sign_cert.pfx` all'interno del client di posta elettronica con cui si vogliono inviare le email firmate digitalmente da **Postmatser Home Works**.

Generiamo ora il certificato digitale da assegnare al Sig. Mario Rossi. Per prima cosa, procediamo

col creare la coppia chiave pubblica/privata del Sig. Rossi:

```
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
```

Firmiamo digitalmente la chiave pubblica appena generata, ricordando che il certificato digitale del Sig. Rossi dovrà avere la validità di un anno (`-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
```

Generiamo il certificato PKCS#12 di Mario Rossi:

```
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
```

Una volta realizzato il certificato PKCS#12 di Mario Rossi, non resta che fornire a Mario Rossi il suo certificato per firmare digitalmente i suoi messaggi di posta elettronica.

Installazione dei certificati PKCS#12 in Thunderbird

Per semplicità supporremo che sia **Postmaster HomeWorks**, sia il Sig. **Mario Rossi** utilizzino una postazione Windows e come client di posta elettronica il programma Thunderbird. Supporremo inoltre che i file **postmaster_digital_sign_cert.pfx** e **rossi_mario_digital_sign_cert.pfx** vengano

copiati, rispettivamente, all'interno della cartella **C:\Certificati**. Pertanto, per importare i certificati PKCS#12 citati, in Thunderbird basterà seguire le indicazioni riportate di seguito:

- avviare Thunderbird;
- aprire il menù **Tools** e selezionare la voce **Options**;
- aprire la sezione **Advanced** e poi la sottosezione **Certificates**;
- premere il pulsante **View Certificates**;
- andare nella sezione **Your Certificates**;
- premere il pulsante **Import**;
- importare il file **C:\Certificati\postmaster_digital_sign_cert.pfx** sulla postazione in cui opera **Postmaster HomeWorks**, il file **C:\Certificati\rossi_mario_digital_sign_cert.pfx** sulla postazione in cui opera **Mario Rossi**;
- premere il pulsante **OK** per confermare le modifiche apportate alla sezione **Your Certificates**;
- nella finestra dal titolo **Options**, premere il pulsante **OK**;
- aprire il menù **Tools** e selezionare la voce **Account Settings**;
- andare nella sezione dedicata o all'account **Postmaster HomeWorks** o all'account **Mario Rossi** ed aprire la voce **Security**;
- inserire, facendo uso del pulsante **Select**, il certificato **Postmaster HomeWorks Certificate** o **Mario Rossi Certificate**, a seconda dell'account su cui si opera, nelle sottosezioni **Digital Signing** ed **Encryption**;
- selezionare le voci **Digitaly sign message (by default)** e **Never (do not use encryption)**;
- premere il pulsante **OK** per confermare la configurazione adottata;
- provare ad inviare un messaggio e controllare che la firma digitale risulti corretta;
- se lo si desidera, a questo punto si può chiudere Thunderbird.

In questo modo Thunderbird invierà i messaggi di posta elettronica di **Postmaster HomeWorks** o di **Mario Rossi**, firmandoli digitalmente.

Conclusioni

La configurazione PKI che abbiamo proposto, basata sul programma OpenSSL, è adatta per una piccola realtà aziendale o per un piccolo ISP, che non debba generare più di uno o due certificati ogni mese. Qualora le esigenze aziendali richiedessero l'erogazione di molti certificati digitali al giorno, converrebbe ricorrere a sistemi PKI più sofisticati di quello esposto in questo articolo (soprattutto se si desidera automatizzare il rilascio dei certificati digitali stessi). Ad ogni modo, a presciendere da quale sia la soluzione adottata, i meccanismi che stanno dietro all'amministrazione di una infrastruttura PKI sono esattamente quelli riportati in questo articolo. Infine, le procedure riportate in questo articolo, possono venire facilmente automatizzate creando degli appositi script Bash, a titolo di esempio si può consultare l'articolo How to Set Up an OpenSSL TEST CA.