網路安全的理論與實務

楊中皇 著

第十五章 OpenCA憑證管理中心

http://crypto.nknu.edu.tw/textbook/

第十五章 OpenCA憑證管理中心

- OpenCA介紹
- OpenCA計畫
- OpenCA的安裝與設定
- OpenCA的使用

前言

- 瞭解了數位憑證、數位簽章以及PKI之後，可以得知要建置一個安全、可信賴的網路環境，讓電子訊息在傳遞與交換過程中，能確保訊息的：
  - 身分鑑別(Authentication)
  - 資料完整 (Integrity)
  - 不可否認性 (Non Repudiation)
  - 私密性 (Private) 等四大需求
- 需依賴利用PKI技術所架設的憑證管理中心 (Certificate Authority，簡稱CA)，來建立值得信賴的數位憑證及數位簽章並強化網路應用程式的安全性。
- OpenCA憑證管理中心便是希望能用最簡便的方式，節省最大的成本來建置符合PKI的憑證管理中心

OpenCA介紹

- OpenCA最早的版本在1998年誕生，而正式發展是在1999年開始
- OpenCA計畫是爲了提供PKI研究及發展相關元件的架構而產生
- 其發展計畫是一個合作致力於發展強健而全方位的開放原始碼憑證管理中心，讓強力的密碼學能實現在網路上所使用的通訊協定上
- OpenCA是架構在多項開放原始碼計畫上，支援OpenCA的開放原始碼軟體有：OpenLDAP、OpenSSL、Apache Project、Apache mod_ssl，來完成憑證系統的實現。這個發展計畫的主要目的有二：
  - 研究強化安全基模，確保最安全的模型用於憑證管理中心
  - 發展最易於安裝及管理的憑證管理中心

OpenCA介紹(續)

- OpenCA最初的構想主要包含三部份
  - (1)以Perl爲主的網頁介面
  - (2)以OpenSSL作爲後端的密碼技術
  - (3)利用Database來儲存管理使用者的所有資訊
- 即使現在OpenCA的複雜性更高了，這三個部份至今仍是OpenCA的根本，幾乎所有的操作都能經由網頁介面來進行
- 目前有六個預設的操作介面，而且使用者可依需求的來建立網頁介面。OpenCA也想爲PKI建立組織化基礎建設，而資料庫儲存關於用戶的必要資訊，如憑證簽署的請求、憑證、憑證廢止請求和CRLs

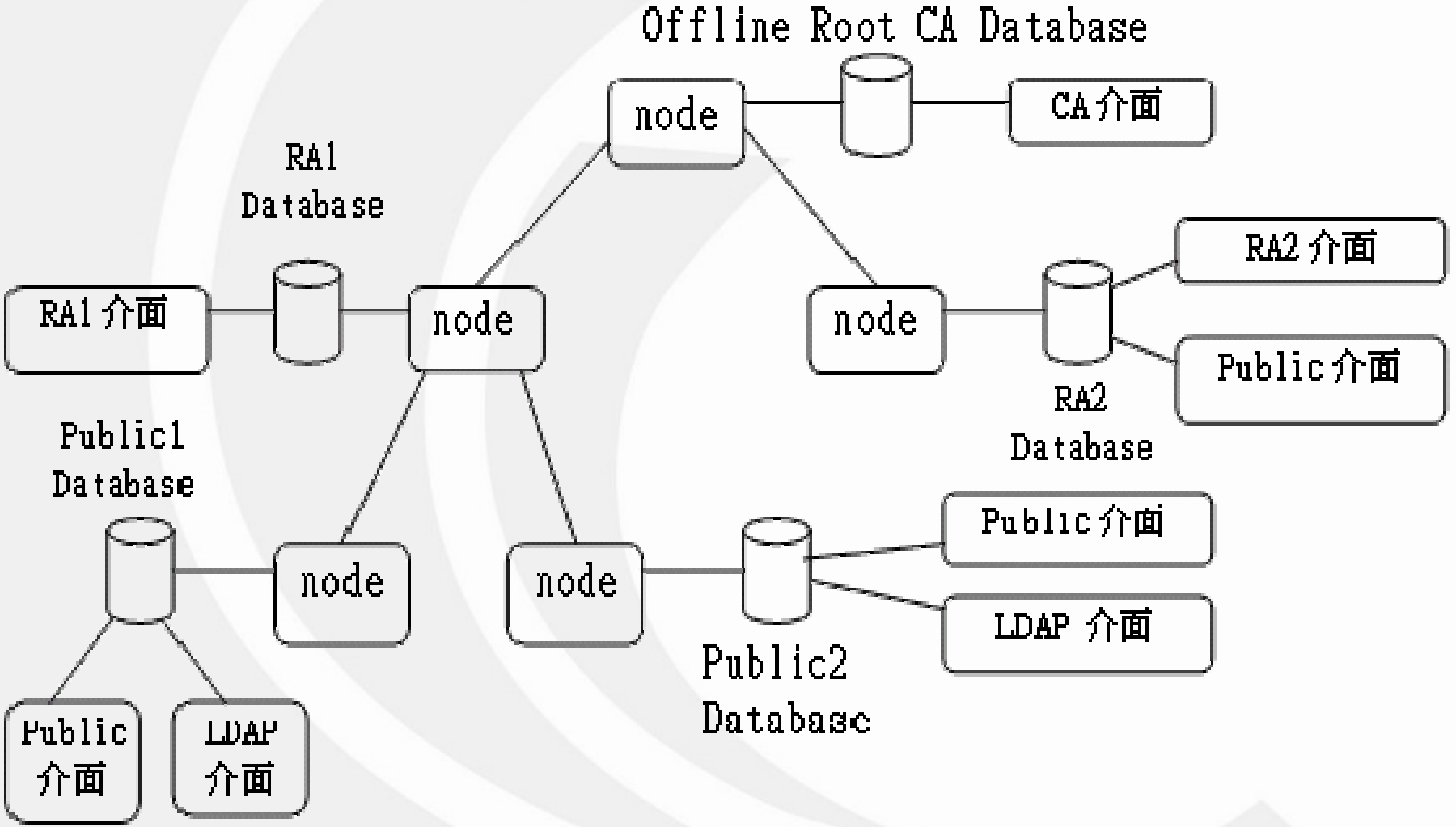
OpenCA所支援的功能

- Public介面 (Public interface)
- LDAP 介面 (LDAP interface)
- RA介面 (RA interface)
- CA介面 (CA interface)
- SCEP
- OCSP
- 網頁的IP過濾 (IP-filters for interfaces)
- 密碼登入 (Passphrase based login)
- 憑證登入 (包含智慧卡) (Certificate based login (incl. smartcards))
- 身份存取控制 (Role Based Access Control)
- 彈性的憑證主體 (flexible certificate subjects)
- 彈性的憑證延伸 (flexible certificate extensions)
- PIN廢止/註銷 (PIN based revocation)
- 數位簽章廢止/註銷 (digital signature based revocation)
- 憑證廢止清冊發佈 (CRL issuing)
- 即將過期憑證警告 (Warnings for expiring certificates)
- 幾乎支援所有的瀏覽器 (support for nearly every (graphical) browser)

OpenCA計畫

- 爲了要使用強力及具有彈性的OpenCA來建立PKI需先了解OpenCA的概觀及階層組織。它的基本概念是每一個X.509的PKI建設爲一個強化的階層組織。因此在建置時，必須建立分散式的PKI架構
- OpenCA在每一個資料庫之前都提供一個網頁式的管理介面稱爲節點 (Node) 進行存取，而在這樣的階層管理中其主幹就是信任中心
- 目前提供的網頁式介面有以下六種：
  - Node (做後端管理)
  - CA (憑證授權中心介面)
  - RA (註冊管理中心介面)
  - LDAP (目錄服務介面)
  - Pub(使用者公用介面)
  - SCEP (由Cisco發展制定的協定，提供虛擬網路VPN，如路由器、交換器之間的通訊，應如何傳送請求及憑證的協定)

完善的OpenCA憑證管理系統設計圖



使用者介面說明

- 節點 (Node)
  - 能管理資料庫及所有匯出和匯入的功能。而資料庫初始化也可經由這個介面操作，來建立OpenCA所需的資料表。
  - OpenCA所使用的資料庫本身是無法經由這個介面建立，您必須事先建立好一個全新的資料庫，並給予適當的管理資料庫權限才行。
  - 這個介面也提供備份和復原資料庫的功能，但不提供備份CA本身私鑰及憑證的機制，目前OpenCA團隊尚未找到備份私鑰的安全機制。
  - 匯出和匯入的功能可以由使用者自訂規則來進行向高層級或低層級進行同步化。這些也包含了被更動設定的物件及狀態

使用者介面說明(續)

- CA介面
  - 包含所有要建立一個憑證及憑證廢止清冊(CRL)的功能
  - 也包含所有要修改憑證及憑證廢止清冊的功能，也提供批次處理系統，讓憑證建立可以自動化
- RA介面
  - 可處理所有的申請要求，包含編輯申請、核准申請、建立智慧卡私鑰、刪除錯誤申請及電子郵件使用者等等
- LDAP介面
  - 是實現將LDAP管理獨立分離出來，因爲LDAP管理是必須的。如果要使用此項功能，則需要LDAP管理的權限。

使用者介面說明(續)

- 公用介面
  - 建立瀏覽器 (IE, Mozilla 1.1+ and Netscape Communicator and Navigator) 憑證簽署請求
  - 建立客戶個人的請求及私鑰 (如：KDE‘s konqueror 或不知如何建立私鑰或請求的系統管理者)
  - 接受由伺服器傳來的PEM格式的PKCS#10請求、註冊憑證、憑證廢止清冊(CRL)、搜尋憑證、測試瀏覽器上的憑證(IE, Mozilla 1.1+ and Netscape Communicator and Navigator 4.7)

OpenCA的安裝與設定

- OpenCA並不是一套完整的系統，它使用了多個開放原始碼的軟體，因此在進行OpenCA安裝前請確定電腦有安裝以下套件：
  - Perl 5.6.1以上
  - MySQL
  - OpenSSL 0.9.7以上
  - LDAP library
  - Perl module：XML-Parser
  - Apache+mod_ssl

MySQL的安裝

- 首先連線到Mysql官方網站，取得mysql。將下載的mysql-4.0.24.tar.gz置於/tmp目錄下，切換到/tmp目錄下將mysql-4.0.24.tar.gz解壓縮，會產生mysql-4.0.24這個資料夾。
  - # cd /tmp
  - # tar zxvf mysql-4.0.24.tar.gz

MySQL的安裝(續)

- 切換到mysql-4.0.24目錄下並執行configure程式它會產生必要的Makefile及相關檔案。再開始編譯及安裝MySQL程式
  - # cd mysql-4.0.24
  - # ./configure –prefix=/usr/local/mysql
  - /*進行mysql安裝的組態工作。完成組態工作
  - # make; make install
  - /*開始編譯及安裝

MySQL的安裝(續)

- 執行初始化MySQL資料庫〈第一次安裝才需執行這個步驟〉
  - # cd script
  - # ./mysql_install_db --user=mysql
- 進入MySQL資料庫，更改root帳號的密碼(預設mysql無設定密碼)
  - # mysql　/*進入資料庫
  - mysql > set password for root=password('openca');
  - /*設定root密碼爲openca，可以自行決定

MySQL的安裝(續)

- 下次進入mysql時需輸入密碼openca
  - # mysql –u root –p
  - Enter password：******　　/* 輸入密碼openca
  - /*或者用底下方式進入mysql
  - #mysql –u root –popenca　/*–p與密碼openca間不能有空白

MySQL的安裝(續)

- 進入mysql建立OpenCA需要用到的資料庫，並賦予openca帳號與mysql資料庫有完全權限
  - #mysql –u root –popenca
  - mysql> create database openca;　/*建立資料庫 openca
  - mysql> grant all privileges on openca .* to openca@localhost identified by "openca"; /*增加新使用者具完全的權限

MySQL的安裝(續)

- 測試資料庫，檢驗設定有無錯誤，關係到OpenCA存取資料
  - #mysql –u root –popenca
  - mysql> use openca;　/*使用資料庫openca
  - mysql> show tables;　/*顯示資料表，正常應該是空的empty set
  - mysql> exit;　/*離開mysql
- 如果系統裡已經裝有MySQL，可以跳過前面編譯安裝的過程

安裝OpenSSL

- 首先連線到OpenSSL官方網站，取得OpenSSL。將下載的openssl-0.9.7e.tar.gz置於/tmp目錄下，切換到/tmp目錄下將openssl-0.9.7e.tar.gz解壓縮，會產生openssl-0.9.7e這個資料夾
  - # cd /tmp
  - # tar zxvf openssl-0.9.7e.tar.gz

安裝OpenSSL(續)

- 切換到openssl-0.9.7e目錄下並執行config程式它會產生必要的Makefile及相關檔案。再開始編譯、測試及安裝OpenSSL程式
- # cd openssl-0.9.7e
- # ./config /*預設應該會將在/usr/local/ssl下，也可指定--prefix=/usr/local/openssl
- # make; make test; make install /*開始編譯測試及安裝

安裝Apache及mod_ssl

- 首先連線到Apache及Mod_ssl官方網站，取得Apache、Mod_ssl。
- 將下載的apache-1.3.33.tar.gz、mod_ssl-2.8.22-1.3.33.tar.gz置於/tmp目錄下，切換到/tmp目錄下將apache-1.3.33.tar.gz、mod_ssl-2.8.22-1.3.33.tar.gz解壓縮，會產生apache-1.3.33、mod_ssl-2.8.221.3.33這個資料夾。
  - # cd /tmp
  - # tar zxvf apache-1.3.33.tar.gz ； tar zxvf mod_ssl-2.8.22-1.3.33.tar.gz

安裝Apache及mod_ssl(續)

- 切換到mod_ssl目錄下並執行configure程式它會產生必要的Makefile及相關檔案。再切換到apache下編譯、測試及安裝Apache程式
  - # cd mod_ssl-2.8.22-1.3.33
  - # ./configure --with-apache=../apache_1.3.33 --with-ssl=../openssl-0.9.7e
  - # cd ../apache-1.3.33　/*切換到apache目錄下
  - # make; make certificate; make install　/*開始編譯及安裝

安裝Apache及mod_ssl(續)

- 編輯httpd.conf增加以下設定字串，加在</VirtualHost>之下(續下頁)

```
# OpenCA Mods
# CA Aliases
Alias     /ca/              /usr/local/openca/httpd/htdocs/ca/
Alias     /ca-node/         /usr/local/openca/httpd/htdocs/ca-node/
ScriptAlias /cgi-bin/ca/       /usr/local/openca/httpd/cgi-bin/ca/
ScriptAlias /cgi-bin/ca-node/  /usr/local/openca/httpd/cgi-bin/ca-node/
# OpenCA Mods
# RA Aliases
Alias     /ra/              /usr/local/openra/httpd/htdocs/ra/
Alias     /pub/             /usr/local/openra/httpd/htdocs/pub/
Alias     /ra-node/         /usr/local/openra/httpd/htdocs/ra-node/
ScriptAlias /cgi-bin/ra/       /usr/local/openra/httpd/cgi-bin/ra/
ScriptAlias /cgi-bin/pub/      /usr/local/openra/httpd/cgi-bin/pub/
ScriptAlias /cgi-bin/ra-node/  /usr/local/openra/httpd/cgi-bin/ra-node/
# OpenCA Mods
<Directory "/usr/local/openca/httpd/cgi-bin/">
    AllowOverride None
    Options ExecCGI
    Order allow,deny
    Allow from all
</Directory>
<Directory "/usr/local/openra/httpd/cgi-bin/">
    AllowOverride None
    Options ExecCGI
    Order allow,deny
    Allow from all
</Directory>
```

安裝Apache及mod_ssl(續)

```
<Directory "/usr/local/openca/httpd/htdocs/">
    AllowOverride None
    Options FollowSymLinks Indexes
    Order allow,deny
    Allow from all
</Directory>
<Directory "/usr/local/openra/httpd/htdocs/">
    AllowOverride None
    Options FollowSymLinks Indexes
    Order allow,deny
    Allow from all
</Directory>
# OpenCA Mods
# adding dir to symlinks following for cert retrieval
# not totally clear WHY openca puts a symlink here, but it did.
<Directory "/usr/local/openra/httpd/cgi-bin/pub">
    AllowOverride None
    Options FollowSymLinks Indexes
    Order allow,deny
    Allow from all
</Directory>
```

安裝Apache及mod_ssl(續)

- 啓動ssl web serv
  - #/usr/local/apache/bin/apachectl startssl /*啓動https
  - #/usr/local/apache/bin/apachectl start /*啓動web server
  - #/usr/local/apache/bin/apachectl restart /*重新啓動web server
  - #/usr/local/apache/bin/apachectl stop /*停止web serverer，startssl 開啓ssl功能，使用https協定

安裝OpenCA

- 首先連線到OpenCA官方網站，取得OpenCA
- 將下載的openca-0.9.2.1.tar.gz置於/tmp目錄下，切換到/tmp目錄下將openca-0.9.2.1.tar.gz解壓縮，會產生openca-0.9.2.1這個資料夾
  - # cd /tmp
  - # tar zxvf openca-0.9.2.1.tar.gz

安裝OpenCA(續)

- 切換到openca-0.9.2.1目錄下並執行configure程式它會產生必要的Makefile及相關檔案。再開始編譯、測試及安裝OpenCA程式，首先安裝RA

```
# cd openca-0.9.2.1
#./configure \
--prefix=/usr/local/openra \ (指定RA目錄)
--with-openssl-prefix=/usr/local/ssl \ (指定openssl位置)
--with-module-prefix=/usr/local/openra/modules \ (指定perl模組位置)
--with-node-prefix=ra-node \ (指定RA伺服器節點管理)
--with-web-host=localhost \ (指定伺服器名稱可使用IP或domain name)
--with-httpd-user=nobody \ (指定web資料夾使用者)
--with-httpd-group=nobody \ (指定web資料夾群組)
--with-httpd-fs-prefix=/usr/local/openra/httpd \ (指定web資料夾位置)
--with-engine=no \
--enable-ocspd \ (啟動線上回應)
--enable-dbi \ (啟動資料庫)
--enable-rbac \
--with-hierarchy-level=ra (安裝層級)
# make；make install-online  /* install-online 同時會安裝ra，ldap，pub，scep和node
# make clean  /*清除剛才編譯時所產生的暫存檔，避免接下來編譯CA部份時發生錯誤。
```

安裝OpenCA(續)

- OpenCA有6個不同元件，分別爲CA，RA，LDAP，PUB，NODE和SCEP，接下來安裝CA

```
# cd openca-0.9.2.1
#./configure \
--prefix=/usr/local/openca \ (指定RA目錄)
--with-openssl-prefix=/usr/local/ssl \ (指定openssl位置)
--with-module-prefix=/usr/local/openca/modules \ (指定perl模組位置)
--with-node-prefix=ra-node \ (指定RA伺服器節點管理)
--with-web-host=localhost \ (指定伺服器名稱可使用IP或domain name)
--with-httpd-user=nobody \ (指定web資料夾使用者)
--with-httpd-group=nobody \ (指定web資料夾群組)
--with-httpd-fs-prefix=/usr/local/openca/httpd \ (指定web資料夾位置)
--with-engine=no \
--enable-ocspd \ (啟動線上回應)
--enable-dbi \ (啟動資料庫)
--enable-rbac \
--with-hierarchy-level=ca (安裝層級)
# make；make install-offline  /* install-offline 同時會安裝ca和node
```

安裝OpenCA(續)

```
<!--general options--->
ca_organization--->CA       /* 設定組織名稱
ca_locality--->KH           /* 設定地區名稱
ca_country--->TW            /* 設定國家代碼
service_mail_account--->yourname@domain.net  /* 管理者email
<!—database configuration--->
dbmodule ---> DBI
db_type ---> mysql          /* 設定資料庫類別
db_name---> openca          /* 設定mysql 的資料庫名稱
db_host ---> localhost
db_port ---> 3306
db_user ---> openca         /* 登入mysql 的帳號
db_passwd ---> openca       /* 登入帳號的通行碼
<!—dataexchange configuration---> /* 設定CA與RA資料交換
0.no dataexchange configure the default 此項預設啟動，加註解符號<!--  --!>，使其失效。
1.the node acts as CA only  /*若為CA就啟動(就是註解拿掉)
2.the node acts as RA only  /*若為RA就啟動(就是註解拿掉)
<!-- these are the devices for the default dataexchange ---> /* 設定資料交換方向
<option>
<name>dataexchange_device_up</name>
<value>/usr/local/openra/OpenCA/var/tmp/ra-up</value>/* RA 的config.xml設定
</option>
<option>
  <name>dataexchange_device_down</name>
  <value>/usr/local/openra/OpenCA/var/tmp/ra-down</value>
</option>
<option>
  <name>dataexchange_device_local</name>
  <value>/usr/local/openra/OpenCA/var/tmp/ra-local</value>
</option>
```

接下來進行組態設定，修改RA的config.xml(路徑在/usr/local/openra/OpenCA/etc)，CA作法亦同

安裝OpenCA(續)

- 若是CA，其dataexchange部份的設定爲

```
<!-- these are the devices for the default dataexchange ---> /* 設定資料交換方向
<option>
  <name>dataexchange_device_up</name>
  <value>/usr/local/openca/OpenCA/var/tmp/ca-up</value>/* CA 的config.xml設定
 </option>
 <option>
  <name>dataexchange_device_down</name>
  <value>/usr/local/openca/OpenCA/var/tmp/ca-down</value>
</option>
<option>
  <name>dataexchange_device_local</name>
  <value>/usr/local/openca/OpenCA/var/tmp/ca-local</value>
</option>
```

安裝OpenCA(續)

- 設定 /usr/local/openra/OpenCA/etc/access_control下的pub.xml.template修改成可接受任何通信協定，以及金鑰長度不受限制

<protocol> .* </protocol>
/*此處若設定為ssl，代表只能以https連線，設成.*則代表http或https均能連線。
<symmetric> keylength </symmetric>
/*此處代表金鑰密碼長度，一般是修改成 <symmetric>0</symmetric>
<login type>none，取消登入密碼限制，也可以設定成passwd
<map_role>no (配合上述login type使用)

安裝OpenCA(續)

- 其他ra-node.xml.template、ra.xml.template也比照上述設定，若想要增加安全性，在<login type>的選項須設成passwd，且<map_role>要設成yes
- 而/usr/local/openca/OpenCA/etc/access_control下的ca-node.xml.template、ca.xml.template也是根據您的需求來設定
- 啓動前先到/usr/local/openca及/usr/local/openra下將var目錄下面的資料夾權限設定成777。並將會讀取到的conf檔改成644
  - # cd /usr/local/openca; chmod –R 777 var
  - # cd /usr/local/openra; chmod –R 777 var
  - /*設定var資料夾權限
  - # cd /usr/local/openca/etc/servers ; chmod –R 644 *
  - # cd /usr/local/openra/etc/servers ; chmod –R 644 *
  - /*將conf檔權限設成644

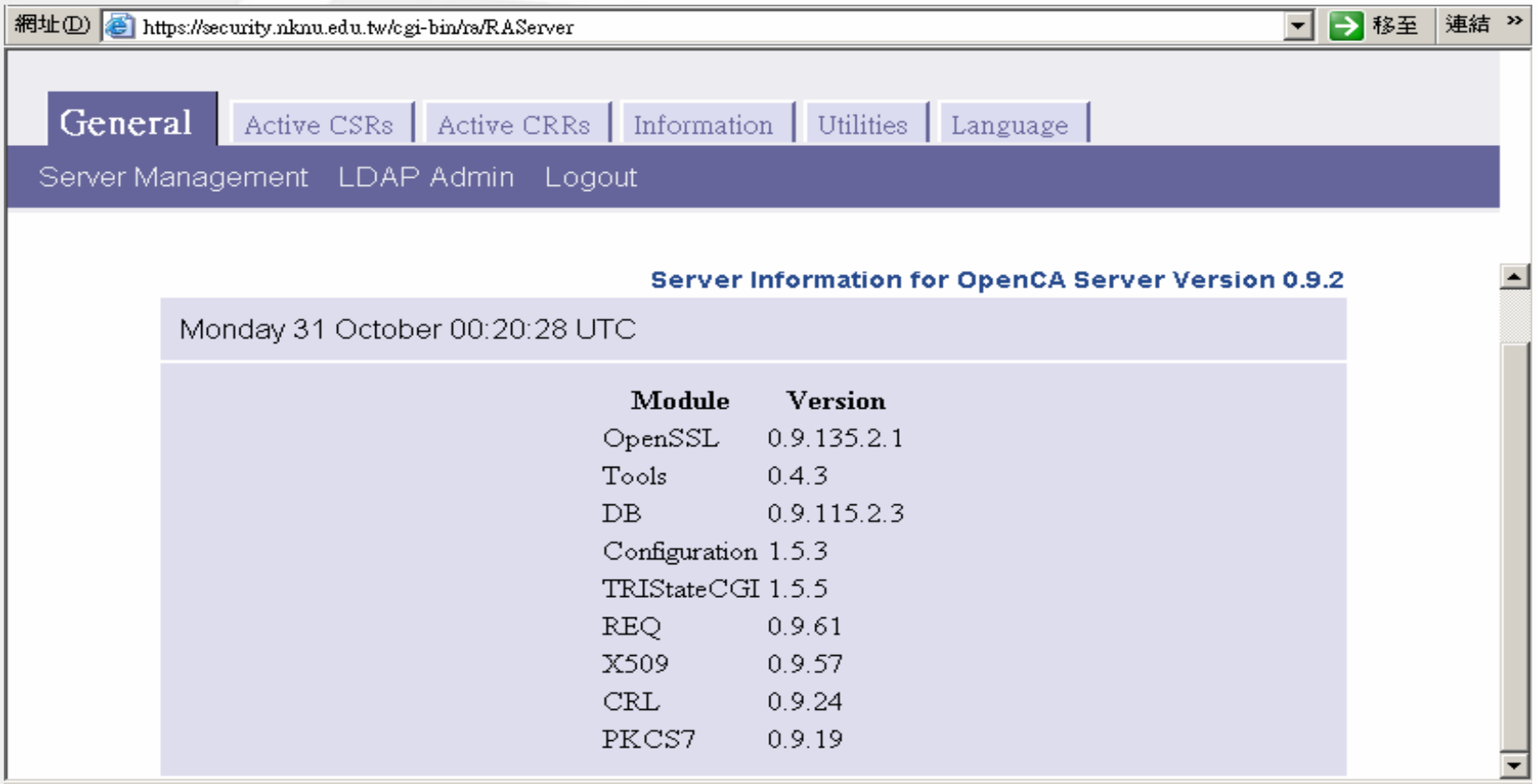
安裝OpenCA(續)

- 執行configure_etc.sh，此shell程式會依您剛所設定的config.xml及template檔來建立適合您的伺服器之組態檔
  - # cd /usr/local/openra/OpenCA/etc
  - # ./configure_etc.sh　/*設定openra的組態檔
  - # cd /usr/local/openca/OpenCA/etc
  - # ./configure_etc.sh　/*設定openca的組態檔

安裝OpenCA(續)

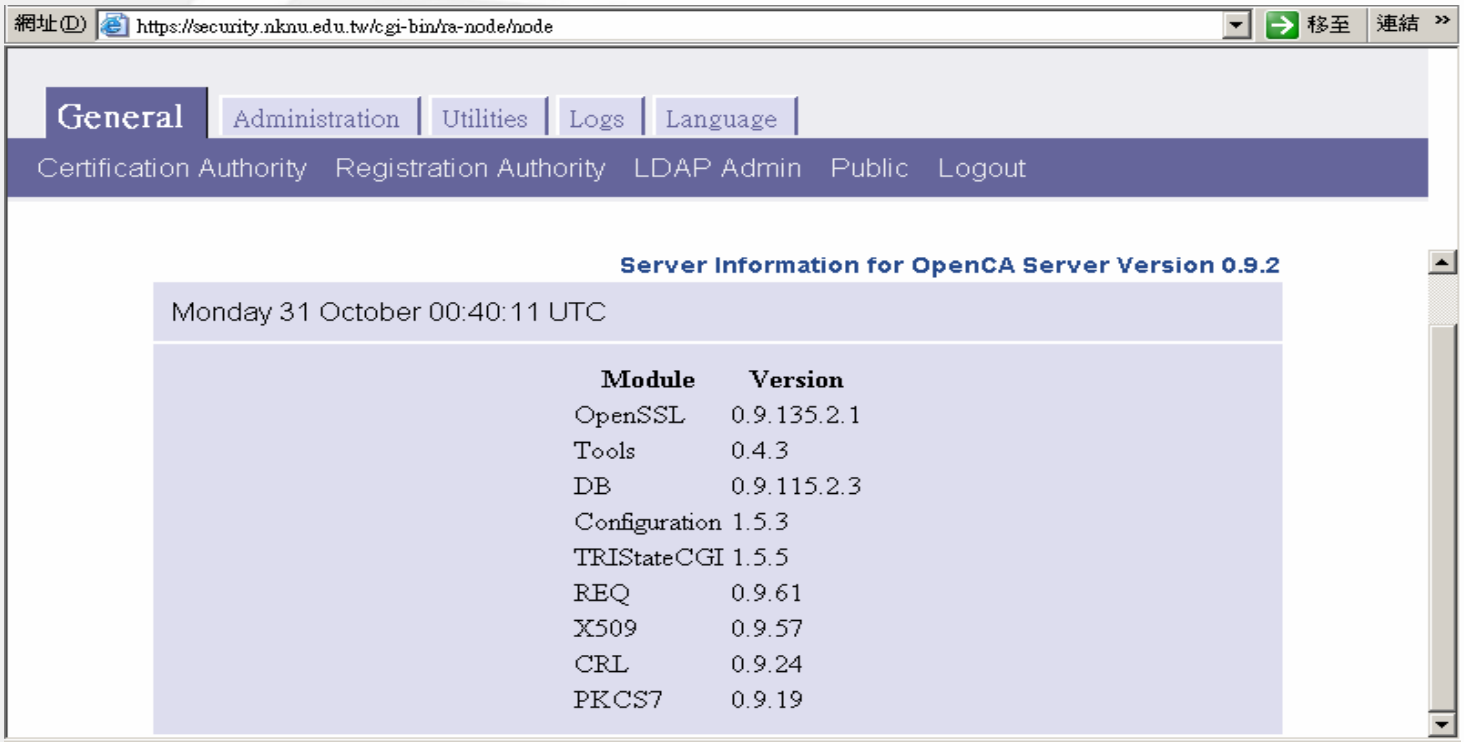
- 啓動openca及openra
  - # cd /usr/local/openra/OpenCA/etc
  - # ./openca_rc start　/*啓動openra。若要停止，將start改成stop
  - # cd /usr/local/openca/OpenCA/etc
  - # ./openca_rc start　/*啓動openca
- 小叮嚀：啓動CA伺服器前，請先檢查perl module有沒有裝到/usr/local/openca下，正常應該會看到/usr/local/openca/modules如果沒有直接複製RA的module檔過來，不然無法執行configure_etc.sh
  - # cd /usr/local/openra
  - # cp –R modules /usr/local/openca/

OpenCA的RA管理網頁



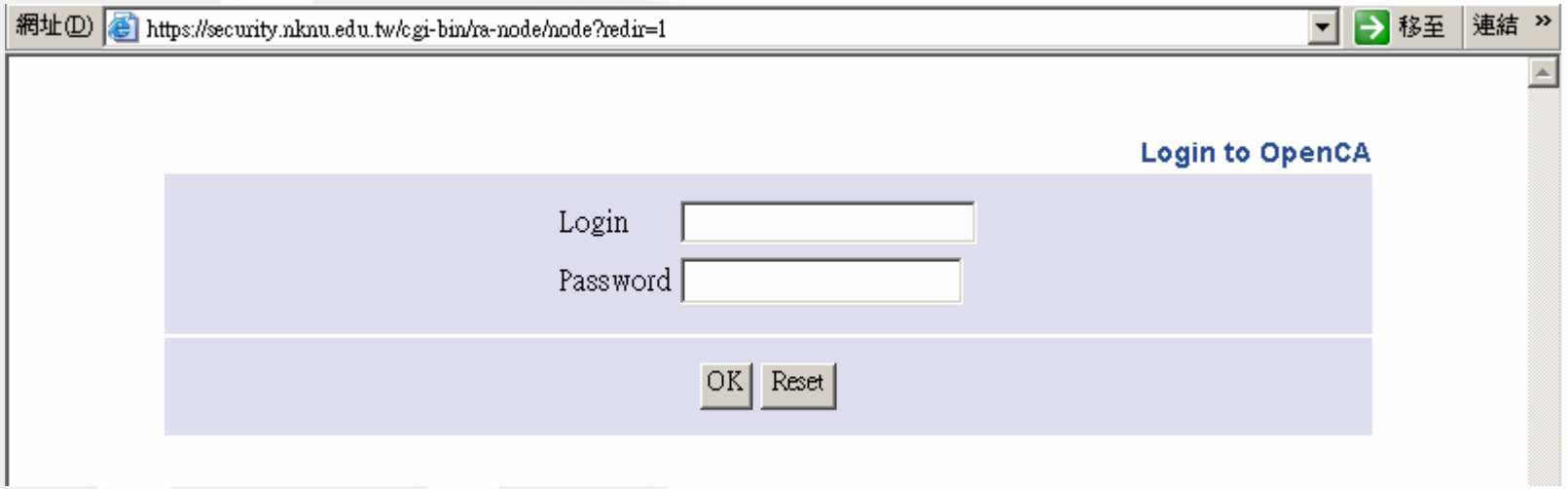
- 連線看看，使用瀏覽器連線到RA伺服器管理網頁，http://localhost/ra/，會看到一個RA的管理網頁，如上圖

RA Node的管理網頁



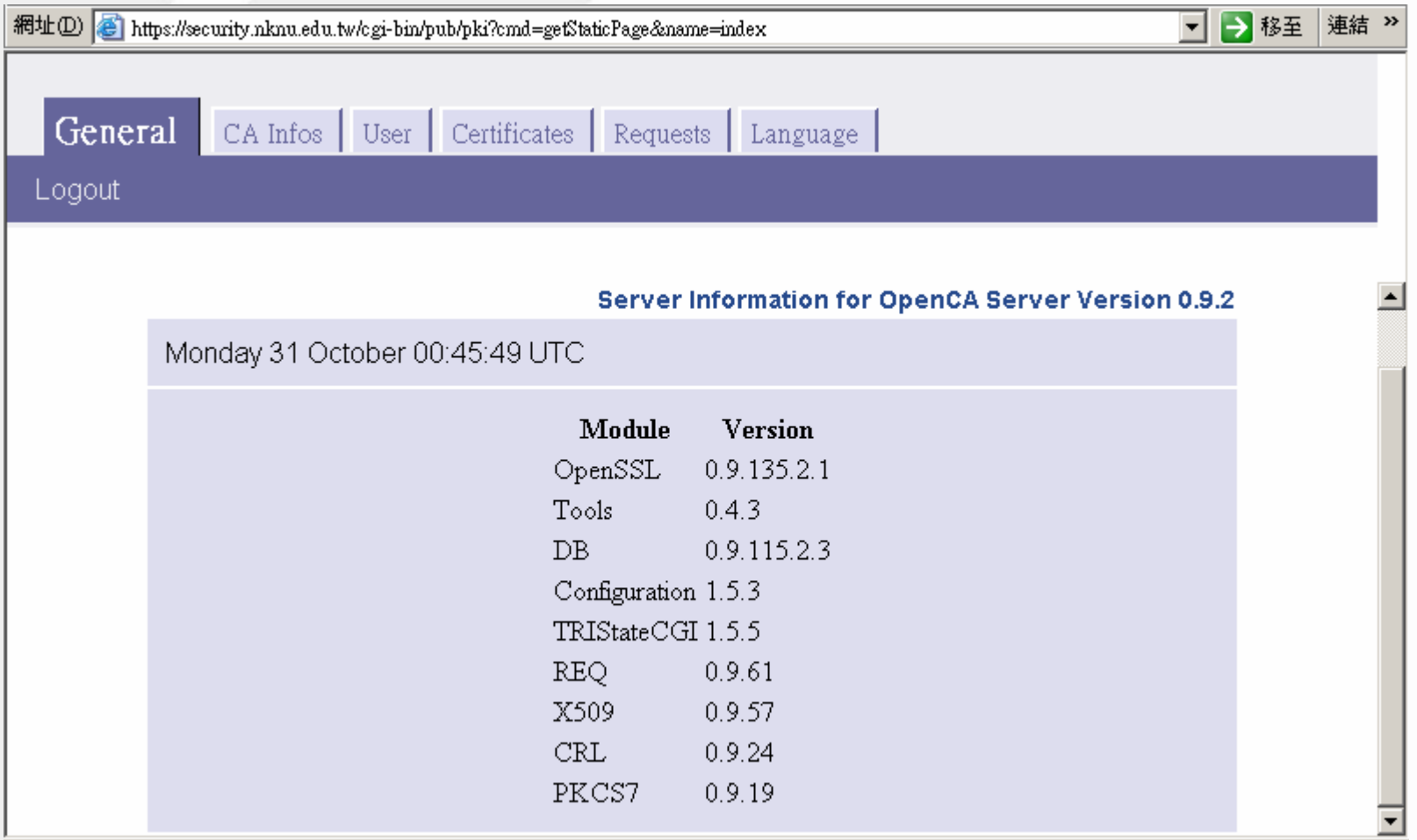
- 確認RA-Node節點管理網頁，http：//localhost/ra-node/，如上圖

輸入帳號密碼登入

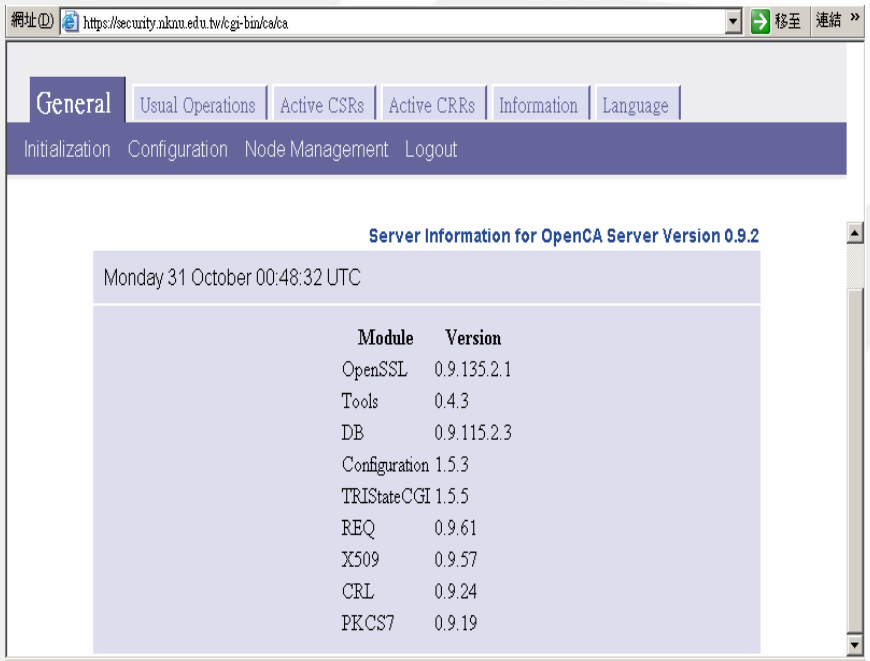


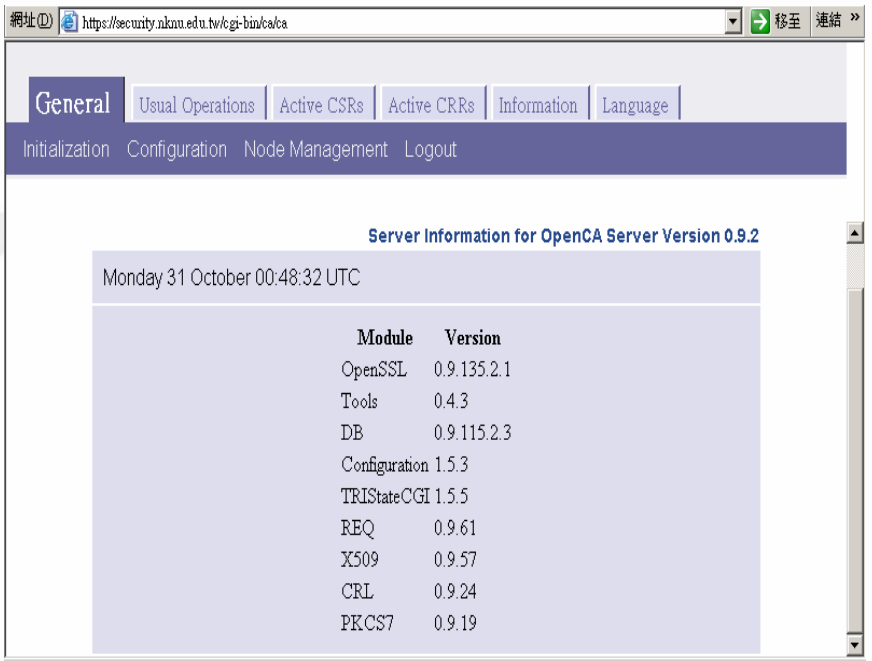
- 若您有設定<login type>爲passwd，您必須先輸入帳號密碼，預設值爲root，如上圖

確定公用網頁是否成功


網址(D) https://security.nknu.edu.tw/cgi-bin/pub/pki?cmd=getStaticPage&name=index
移至
連結
General
CA Infos
User
Certificates
Requests
Language
Logout
Server Information for OpenCA Server Version 0.9.2
Monday 31 October 00:45:49 UTC
Module Version
OpenSSL 0.9.135.2.1
Tools 0.4.3
DB 0.9.115.2.3
Configuration 1.5.3
TRIStateCGI 1.5.5
REQ 0.9.61
X509 0.9.57
CRL 0.9.24
PKCS7 0.9.19

CA管理網頁及CA-Node管理網頁



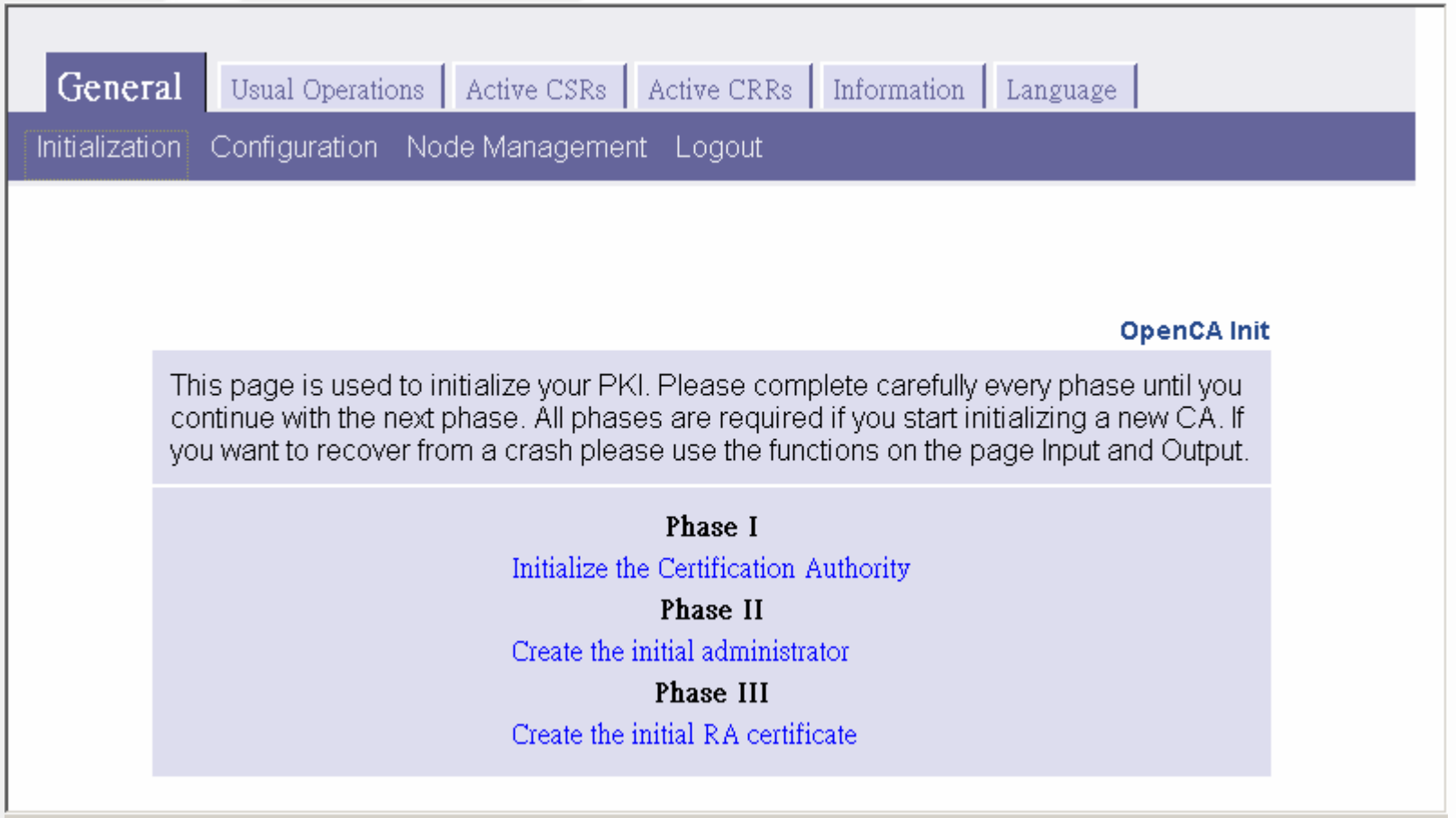


- 如果都可以看到上述網頁，表示RA、RA-Node與Pub設定沒有問題。同樣的，我們可用相同的方法測試CA、CA-Node，如果能正常運作，便會出現上左圖的CA管理網頁及上右圖的CA-Node管理網頁

憑證管理中心設定

- 當我們架設好CA及RA之後，第一次啓動CA，先要做初始化動作。
- 以下介紹初始化動作步驟：
- 先登入到CA伺服器管理網頁
  - http：//localhost/ca/，選擇一般(General)-->初始化(Initialization)

OpenCA初始化


General
Usual Operations
Active CSRs
Active CRRs
Information
Language
Initialization
Configuration
Node Management
Logout
OpenCA Init
This page is used to initialize your PKI. Please complete carefully every phase until you continue with the next phase. All phases are required if you start initializing a new CA. If you want to recover from a crash please use the functions on the page Input and Output.
Phase I
Initialize the Certification Authority
Phase II
Create the initial administrator
Phase III
Create the initial RA certificate

第一階段初始化項目

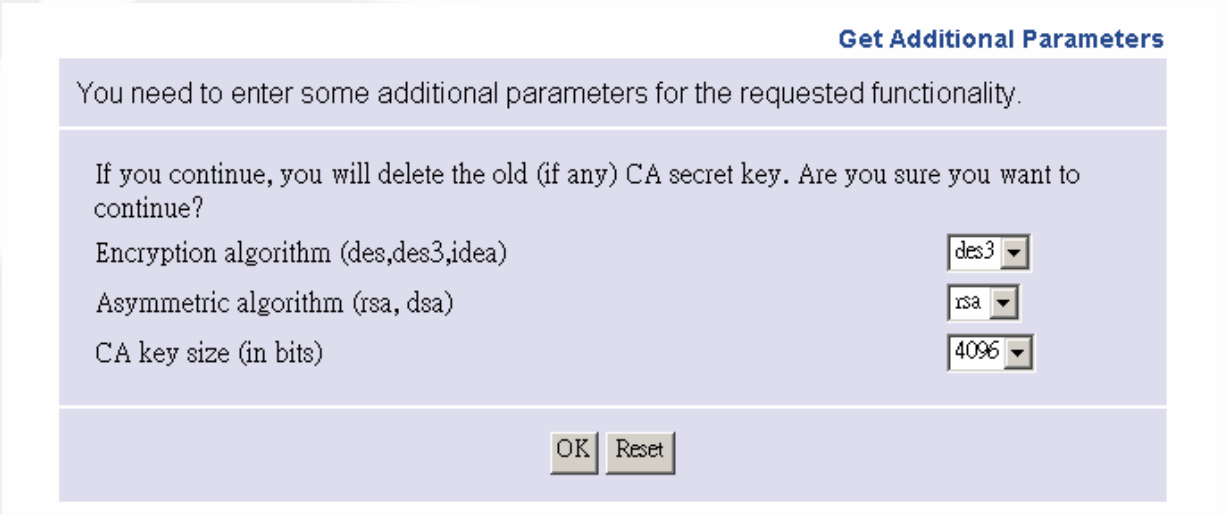
- 進入第一階段(Phrase I)，初始化資料庫並產生憑證管理中心的私密金鑰與憑證(Phase 1--Initialize the Certification Authority)

顯示SQL資料庫初始化訊息RA

```
select * from ca_certificate;
drop table ca_certificate;
create table ca_certificate (ca_cert_key VARCHAR (255) NOT NULL PRIMARY KEY,
format TEXT, data TEXT, dn TEXT, cn TEXT, email TEXT, status TEXT, public_key
TEXT, notafter BIGINT) TYPE=BDB;
select * from crl;
drop table crl;
create table crl (crl_key VARCHAR (255) NOT NULL PRIMARY KEY, status TEXT,
format TEXT, data TEXT, last_update TEXT, next_update TEXT) TYPE=BDB;
select * from crr;
drop table crr;
create table crr (crr_key BIGINT NOT NULL PRIMARY KEY, cert_key BIGINT,
submit_date TEXT, format TEXT, data TEXT, dn TEXT, cn TEXT, email TEXT, ra TEXT,
rao TEXT, status TEXT, reason TEXT, loa TEXT) TYPE=BDB;
select * from request;
drop table request;
create table request (req_key BIGINT NOT NULL PRIMARY KEY, format TEXT, data
TEXT, dn TEXT, cn TEXT, email TEXT, ra TEXT, rao TEXT, status TEXT, role TEXT,
public_key TEXT, scep_tid TEXT, loa TEXT) TYPE=BDB;
```

- 首先初始化資料庫，本系統採用的資料庫是MySQL，先點選顯示SQL資料庫初始化訊息(Show SQL statements for database initialization)，再點選初始化資料庫(Initialize Database)選項，完成初始化資料庫動作

選擇金鑰產生演算法



- 選擇初始化(initialize) -->初始化第一階段(initialize phase 1) -->產生新的私密金鑰(generate new secret key)
- 加密演算法可選擇DES、DES3與IDEA，非對稱演算法可選擇 RSA或DSA，金鑰長度自由選擇，長度愈長，保密效果愈好，最後會要求輸入通行碼保護，如上圖

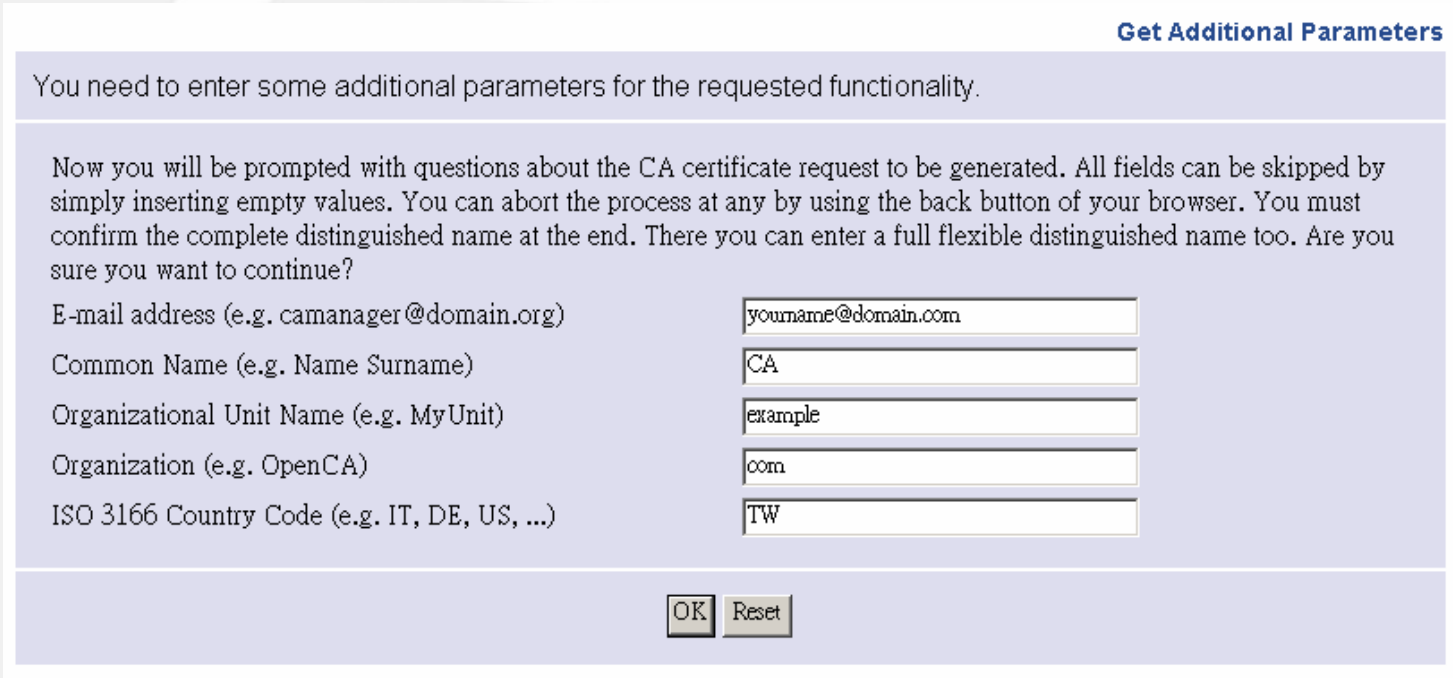
產生CA私密金鑰

CA Secret Key Generation

Following you can find the result of the generation process.

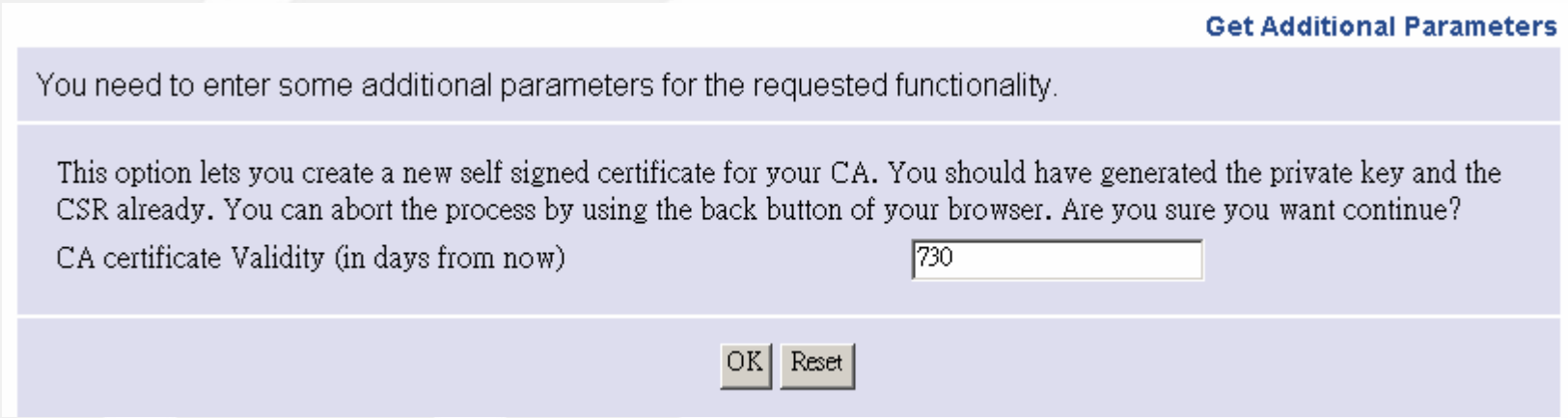
```
-----BEGIN RSA PRIVATE KEY-----
Proc-Type: 4,ENCRYPTED
DEK-Info: DES-EDE3-CBC,BB84F75178107A8D

qaLBTri9Mt4Yv9NK2DF6g9Gvc27KYv0L776jFZPDNPEjws/0w/homz9c63ODqBv7
aVc8b7dDb3Fen5Zb1S7CBxrEJq+pSAVQr3GpucY6afHgwzXV1Zr1qYqb7mmi6Ziw
2KwHtA16cyAduBUfT2hUngxeZphPLOq4/AKackGbPqhzjtQVkfoNCT2pdyDWnt7J
lV2r23i5jo2OmrLidMj6UlJmmiJgmtP6kHS1qH/gyetKT3etCk6v/SQiXGgABpwG
au3iobtRGtuW9pWws300IwWmo0QmJU70S6X6DZLXtxdzd7dLevT4/EA1gmv9bBl0
UQY6Dd5vw0gMNFy86gDdCs3UyWQHLJ1Z3TBvBKEKkmITs1xD3L6GTxE6LgqkEP7e
HDVRjgondQ/YUneei8J3DRWklgq4oENe9AOn5DTiBq3pDoPcI6tpfvwazI6u8S74
7OzlD5piQ499Nh7VPPIU6CuAISHy88WBvblyaS7UESgO7zlOUSbo5vp04s400qmh
ZEI09Z3aNvl4tgaz3f8J7tOgxiSpwOrgB8+uFmo4U5ZnWvocM2WsanlBkCCF8TsU
2qFjDbXGqWIhbRujam2eU3fSw/+mKNTxjmbjY28oYYyO2mUU1tb+6bsMF8f39QDE
58dD4mnQ2aMZhKzxQhgaYdpuyT9Lpvsoazq9EAcORb2lZMa9v/1BzVUJCN3jCgAT
2L4V3E2s7R+AVBMAQesR8M/f7mT3bjfFHIT40q42JKsoVsj5Jf9iOYcEw9EK4OFc
8YJ6hTEfdCvjpCTML349PyCJAcBzmpCFEb5BAtDQcyBR1LZNq35L6Qirzj0Ce/BV
GkaUue1fzvh7cQkTNhfydipV72xwrg01BuwiZogqHgex/9ilwr4mIOxJTPp0dBgb
Mh1RwqfpSyyd7VMVqS8aaVIfE6dlooroql/IcTjpML2qN7yqCkImbK/ly2Cy5Nhe
```

產生CA憑證要求



- 選擇初始化(initialize) --> 初始化第一階段(initialize phase 1) -->產生新的憑證要求(generate new cert request)
- 在適當欄位輸入欲建立憑證管理中心的憑證資料

設定CA憑證有效期限

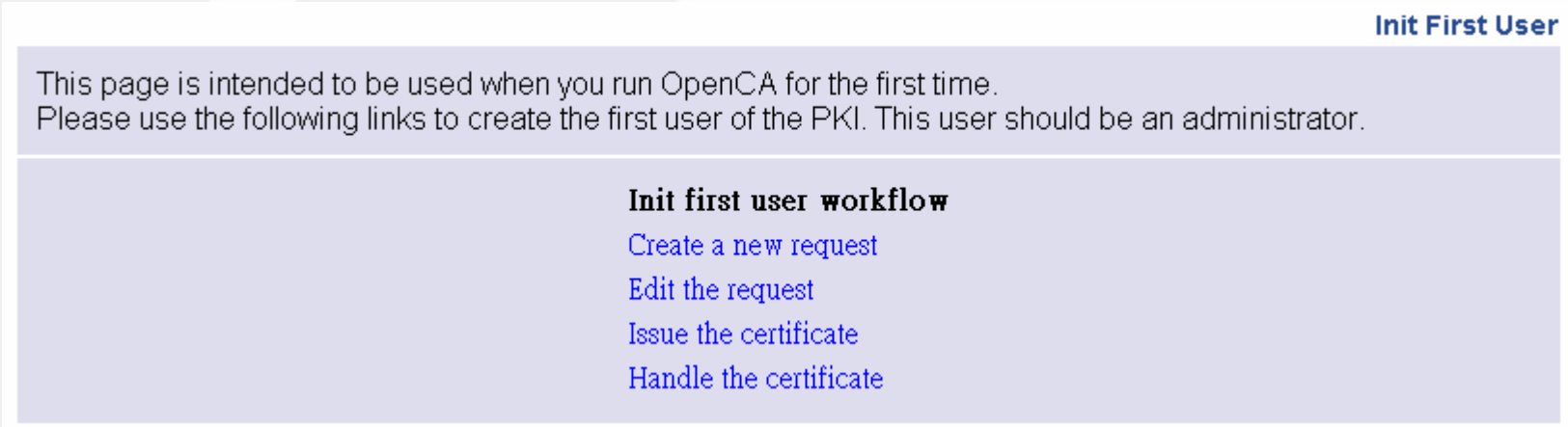


- 選擇初始化(initialize) --> 初始化第一階段(initialize phase 1)-->自行簽署CA憑證(Self Signed CA Certificate)
- 有效期限預設值730天

CA鏈結重建成功

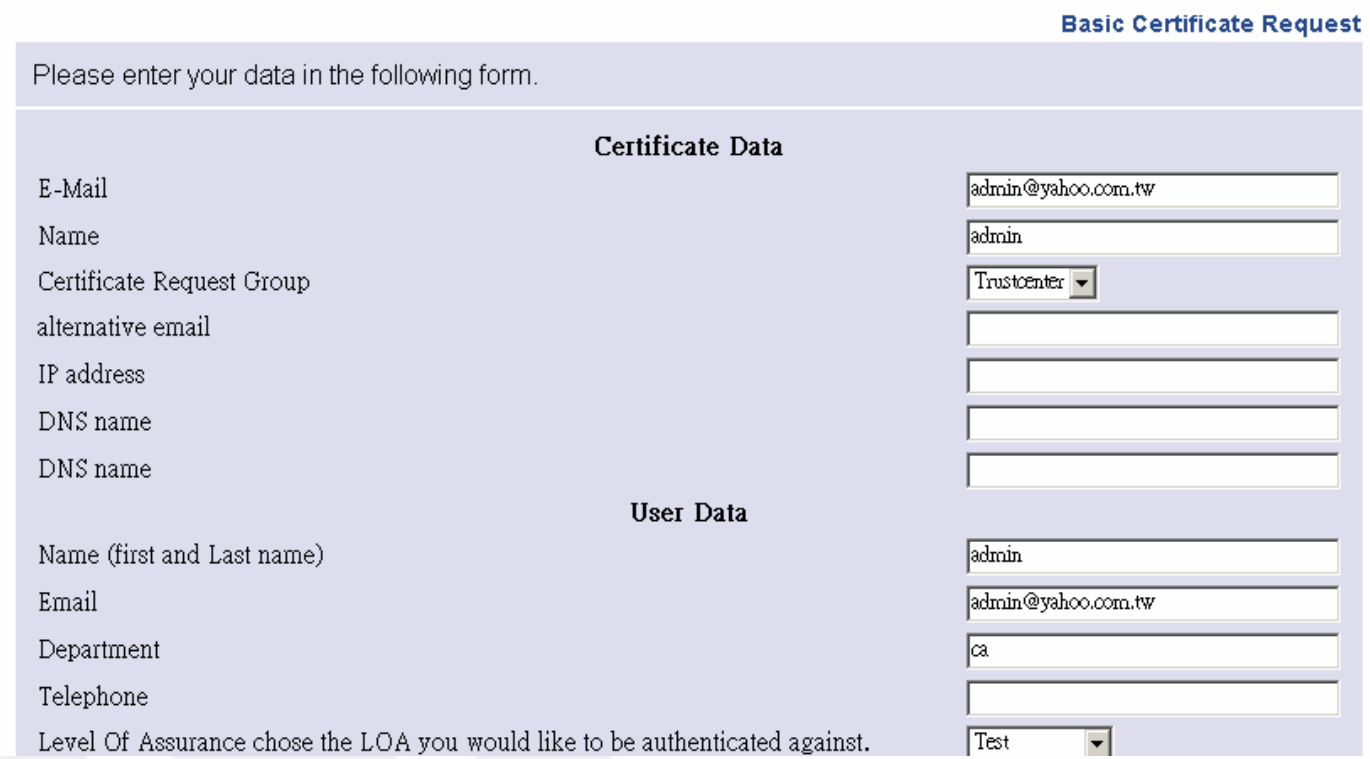
- 最後步驟，選擇初始化-->初始化第一階段-->重建CA鏈結(Rebuild CA Chain)

產生CA管理者的憑證要求



- 接下來進入第二階段(Phase II)—產生CA管理者憑證

CA管理者憑證申請單



Basic Certificate Request

Please enter your data in the following form.

Certificate Data

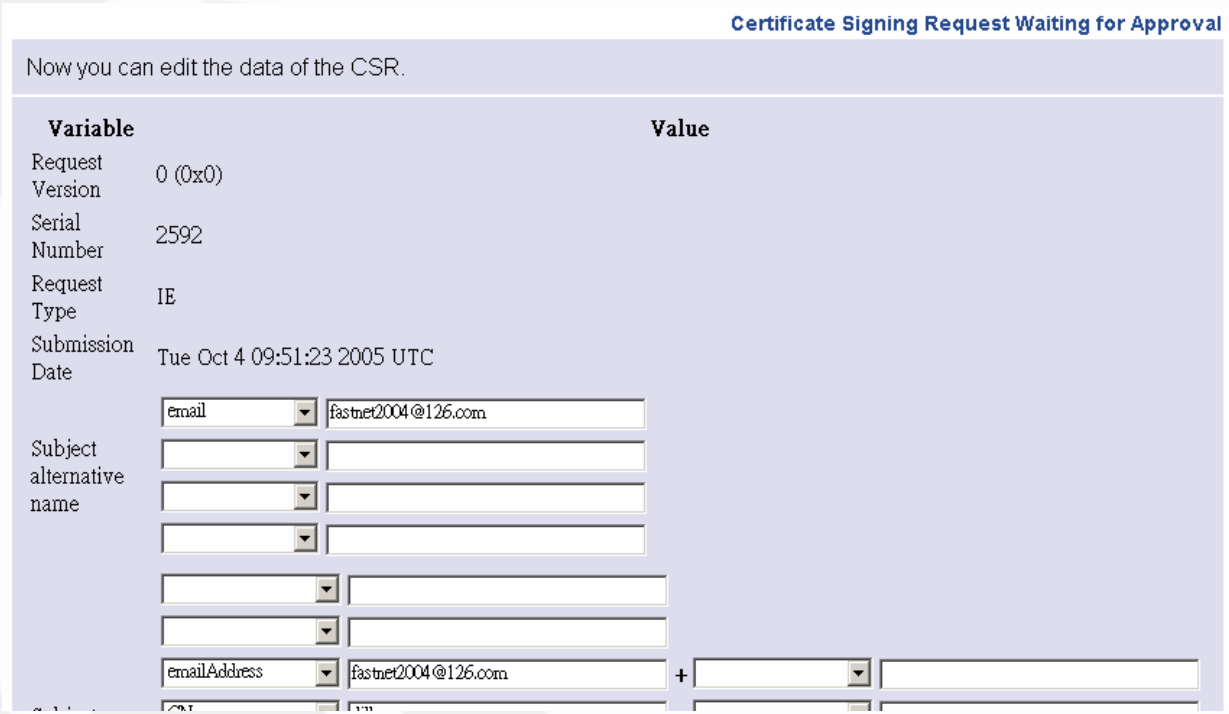
E-Mail	admin@yahoo.com.tw
Name	admin
Certificate Request Group	Trustcenter
alternative email	
IP address	
DNS name	
DNS name	

User Data

Name (first and Last name)	admin
Email	admin@yahoo.com.tw
Department	ca
Telephone	
Level Of Assurance chose the LOA you would like to be authenticated against.	Test

- 選擇初始化(initialize) -->初始化第二階段(initialize phase 2)--> 產生新的憑證要求(Create a new request)
- 這個是CA管理者的憑證要求，在欄位填入適當的值

編輯CA管理者憑證申請單



- 選擇初始化-->初始化第二階段-->編輯憑證要求(edit request)：審核申請單填寫無誤後，上傳至憑證管理中心等待簽發
- 選擇初始化--> 初始化第二階段-->處理憑證要求(handle request)：簽發憑證之前輸入CA私密金鑰保護通行碼，完成簽發憑證手續

OpenCA安裝最後階段

- 最後第三階段(Phase 3)--產生RA 管理者憑證，步驟同第二階段，產生CA管理者憑證。如此一來便大功告成。
- 接下來要進行的是資料交換步驟，先將CA與RA管理者憑證下載到RA伺服器。
  - 登入到CA節點管理介面，http：//localhost/ca-node/。選擇管理介面(Administration)-->資料交換(Dataexchange)，選擇登錄全部資料到下一級中心(Enroll data to a lower level of the hierarchy)-->all。
  - 切換到RA節點管理中心，http：//localhost/ra-node/。選擇管理介面(Administration)-->資料交換(Dataexchange)，選擇由上一級下載全部資料(Download data from a higher level of the hierarchy-->All)

OpenCA的使用

- OpenCA是屬於階層式的架構，因此主要身份有三種：
  - (1)一般使用者，透過公用介面來進行憑證的申請
  - (2)RA管理者，利用RA管理的介面來審查申請者的資料是否有誤，一般在這個部份都會進行真實身份的審查
  - (3)CA管理者，最上層的管理者，利用CA管理介面來進行憑證的核准或再一次的審查

使用者憑證申請步驟

- 使用者先登入公用介面，http：//localhost/pub/，選擇使用者(User)-->提出憑證要求(Request a Certificate)-->瀏覽器自動偵測憑證要求格式(Request a certificate with automatic browser detection)
- 讓電腦自動判斷瀏覽器種類，便可進行憑證申請的要求

填寫憑證申請單

Basic Certificate Request

Please enter your data in the following form.

Certificate Data

E-Mail	
Name	
Certificate Request Group	Internet
alternative email	
IP address	
DNS name	
DNS name	

User Data

Name (first and Last name)	
Email	
Department	
Telephone	
Level Of Assurance chose the LOA you would like to be authenticated against.	Test

- 填寫憑證要求申請單基本資料
- 填寫完畢後會傳送到RA等候批准

RA管理者審核資料

- RA管理者登入到RA管理介面，http://localhost/ra/
- 選擇有效憑證要求(Active CSRs)-->新的(New)-->搜尋(search)
- 搜尋後會列出最近的憑證申請列表

有效憑證申請列表

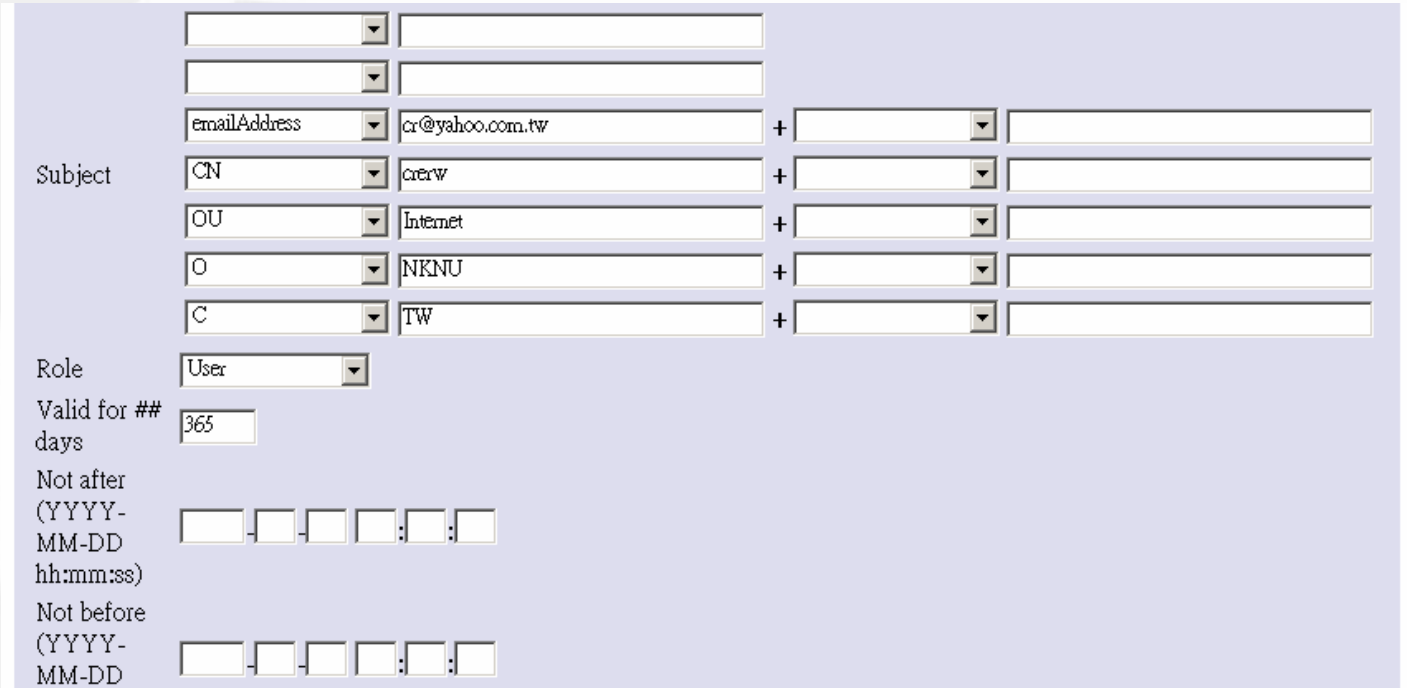
New Certificate Signing Requests

Monday 31 October 01:58:30 UTC

Serial	Submit Name	Submitted On	Requested Role	Requested LOA
1312	emailAddress=cr@yahoo.com.tw,CN=crerw,OU=Internet,O=NKNU,C=TW	Fri Sep 23 16:46:46 2005 UTC	User	Test
1568	emailAddress=cr12@yahoo.com.tw,CN=crerw,OU=Internet,O=NKNU,C=TW	Fri Sep 23 17:02:00 2005 UTC	User	Test
2592	emailAddress=fastnet2004@126.com,CN=dill,OU=Internet,O=NKNU,C=TW	Tue Oct 4 09:51:23 2005 UTC	User	Test

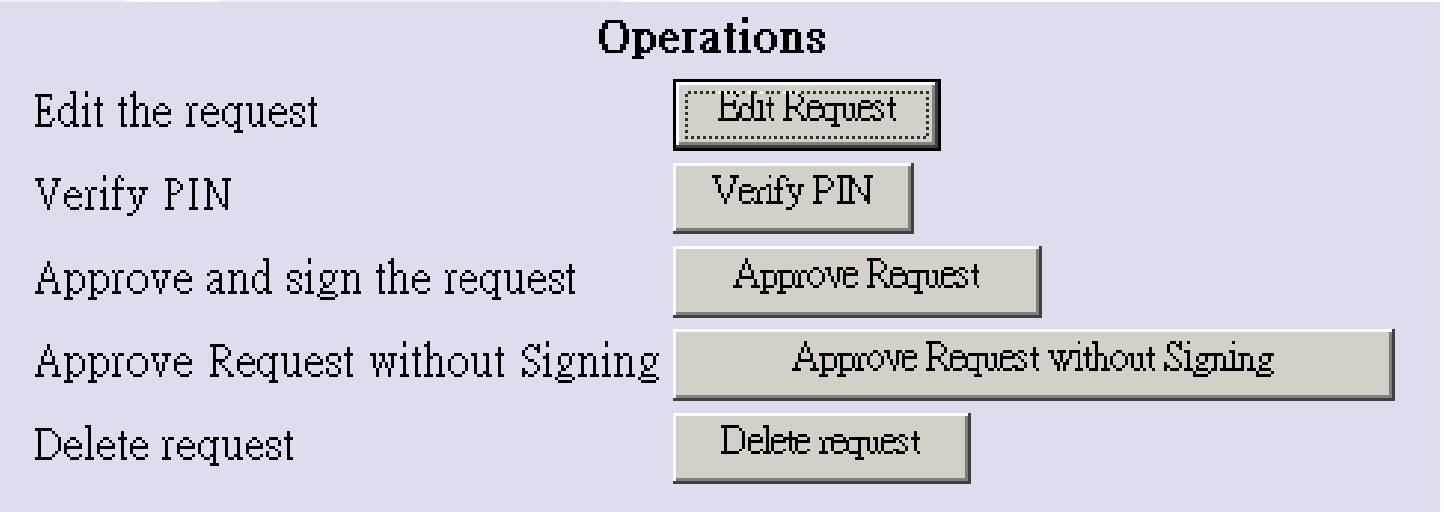
No Extra References

註明有效期限



- 點選序號進入使用者資料檢查，檢查無誤後，按下Edit Request來註明其有效期限(可設定爲365天)

RA管理審核功能



- 最後點選批准要求但未簽署按鈕(Approve Request without signing) ，如上圖

CA管理者發佈憑證

- CA管理者登入CA管理介面，http://localhost/ca/，選擇有效憑證要求(Active CSRs)-->已審核的(Approved)檢查已審查通過，尚待發佈的憑證資料
- 若CA同意憑證請求，按發佈憑證(issue the certificate)選項。若認爲該申請資料有問題，可按刪除申請(Delete Request)

CA有效憑證申請列表

Approved Certificate Signing Requests

Monday 31 October 02:20:34 UTC

Operator Serial		Submit Name	Approved On	Requested Role	Requested LOA
n/a	1312	emailAddress=cr@yahoo.com.tw, CN=crerw, OU=Internet, O=NKNU, C=TW	n/a	User	Test

No Extra References

CA簽發憑證

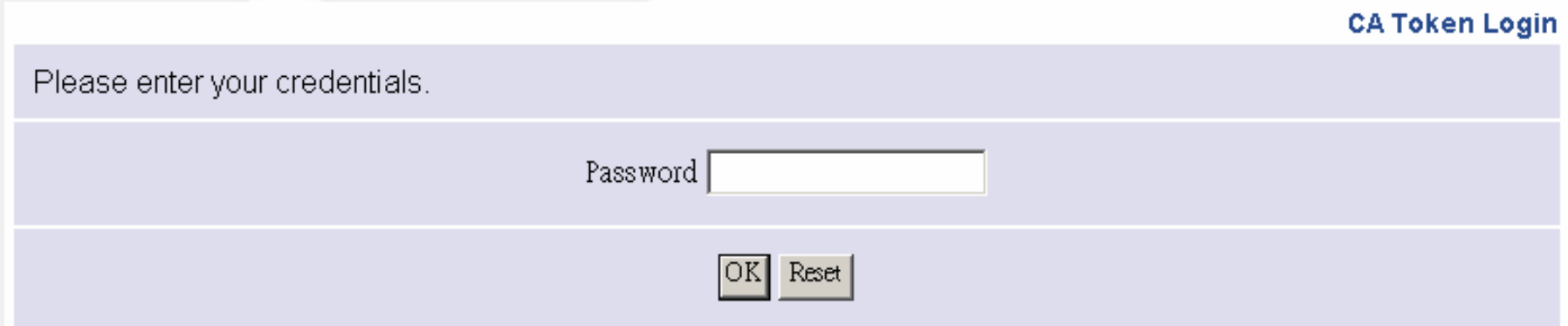
Modulus (key size)	1024
Public Key Algorithm	rsaEncryption
Public Key	Modulus (1024 bit): 00:9d:cc:d1:61:0f:66:f9:f7:1c:8b:8e:da:90:d9: cd:07:ab:3f:f7:64:cd:02:fc:78:e0:0d:f9:1b:fd: 39:6c:78:4e:e4:25:cc:ff:fd:7e:ab:4a:e9:e9:b4: a5:c6:54:01:1e:60:85:e1:7b:49:0b:90:5d:68:4e: dd:24:08:2d:8e:4c:63:ad:d8:e6:a6:3e:cd:ac:21: 07:ab:6a:ac:33:36:5c:84:b7:9a:b8:4e:db:57:74: 71:3c:07:a5:1e:c5:49:0e:55:cc:3a:bf:d2:45:1f: 9b:70:81:c6:f1:cc:fe:e9:93:49:21:e0:54:41:65: e5:86:93:cb:4d:a0:39:fc:75 Exponent: 65537 (0x10001)
Signature Algorithm	sha1WithRSAEncryption
Name (first and Last name)	crouse
Email	cr@yahoo.com.tw
Department	n/a
Telephone	n/a

Operations

Issue certificate [Issue certificate]

Delete request [Delete request]

輸入CA私密金鑰保護通行碼

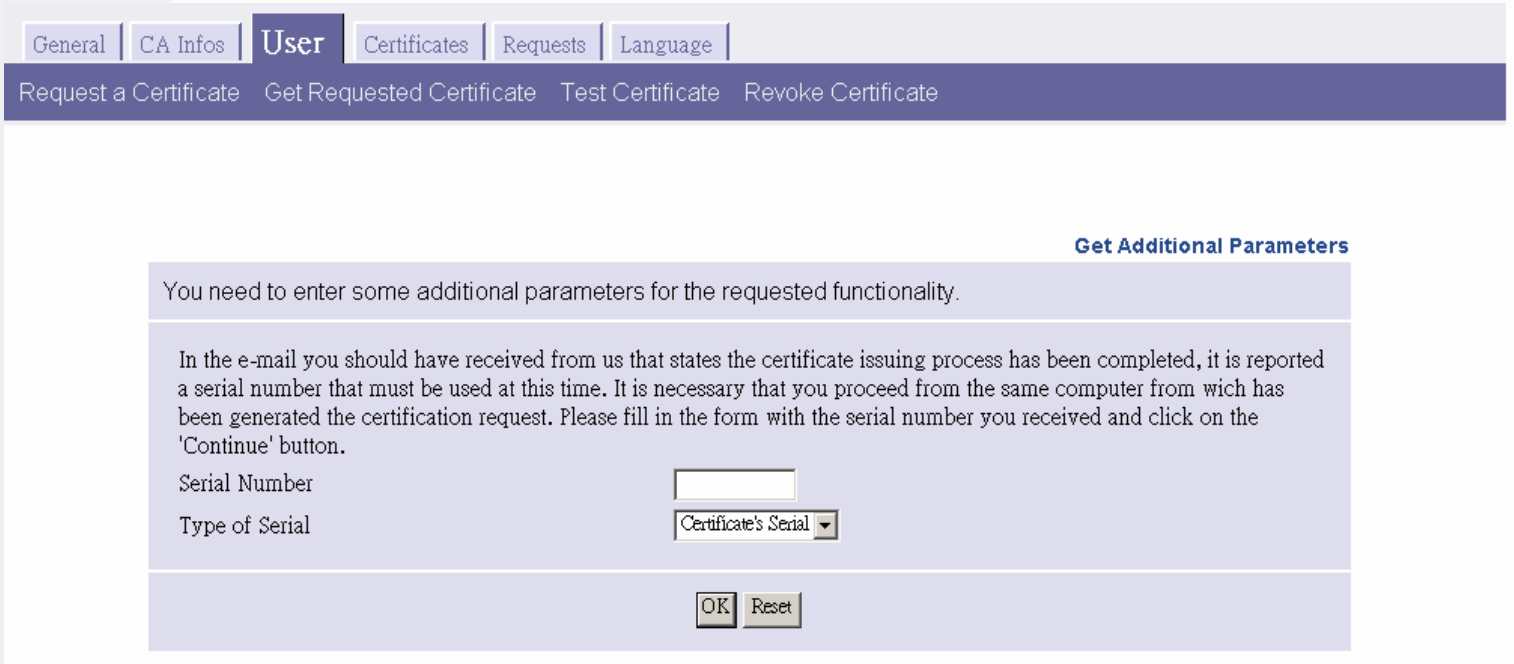


- 發佈之前要完成簽署動作
- 先輸入CA私密金鑰保護通行碼，進行簽發憑證動作

注意事項

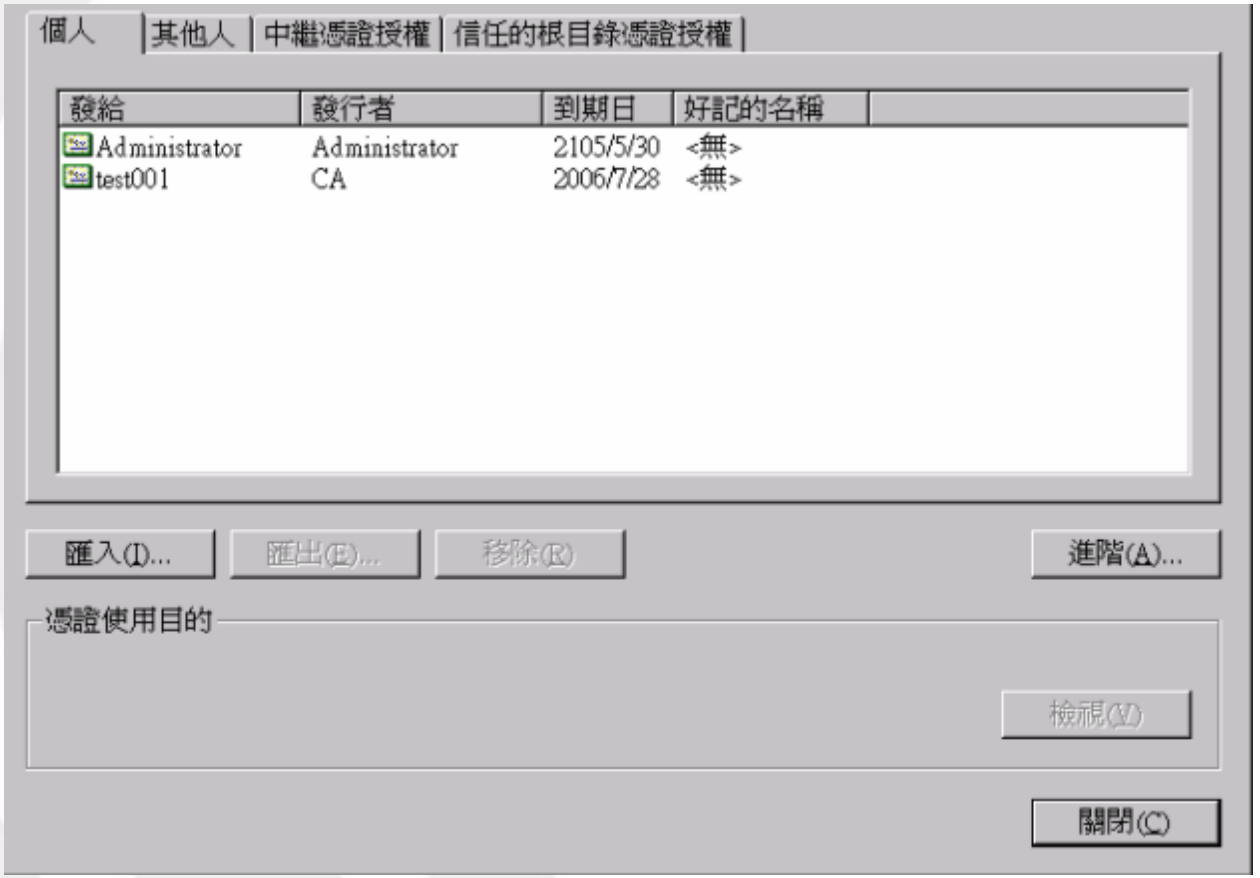
- 若CA、RA兩者不使用同一個資料庫，則CA需輸出已簽署過的憑證到資料庫
  - 連線到ca-node管理介面，http：//localhost/ca-node，選擇管理介面(Administration)-->資料交換(dataexchange)-->登錄資料到下一級(Enroll data to a lower level of the hierarchy)-->Certificates
  - RA再從資料庫載入已簽署憑證到RA，http：//localhost/ra-node。選擇管理介面(Administration)-->資料交換(dataexchange)-->下載上一級資料(Download data from a higher level of the hierarchy)-->Certificates
- 如此一來CA與RA兩個資料庫的資料才能同步

使用者下載已發佈的憑證



- 使用者到公用介面(http：//localhost/pub)下載申請核准通過且完成簽署的憑證
- 選擇使用者(User)-->獲得要求憑證(Get Requested Certificate)，輸入當初申請的憑證序號(Serial number)，下載過程會自動安裝到瀏覽器上

檢視已安裝憑證



- 檢查瀏覽器是否已安裝下載的憑證，選擇IE-->工具-->網際網路選項-->內容-->憑證

檢視憑證內容

- 檢視下載憑證內容並附有一個對應的私密金鑰

檢視憑證詳細資料

- 檢視憑證詳細資料，憑證內容符合X.509憑證格式

參考資料

- OpenCA，http://www.openca.org
- Apache，http://www.apache.org
- Mod_ssl，http://www.mod-ssl.org
- OpenSSL，http://www.openssl.org
- Perl，http://www.perl.com/
- MySQL，http://www.mysql.com
- OpenLDAP，http://www.openldap.org