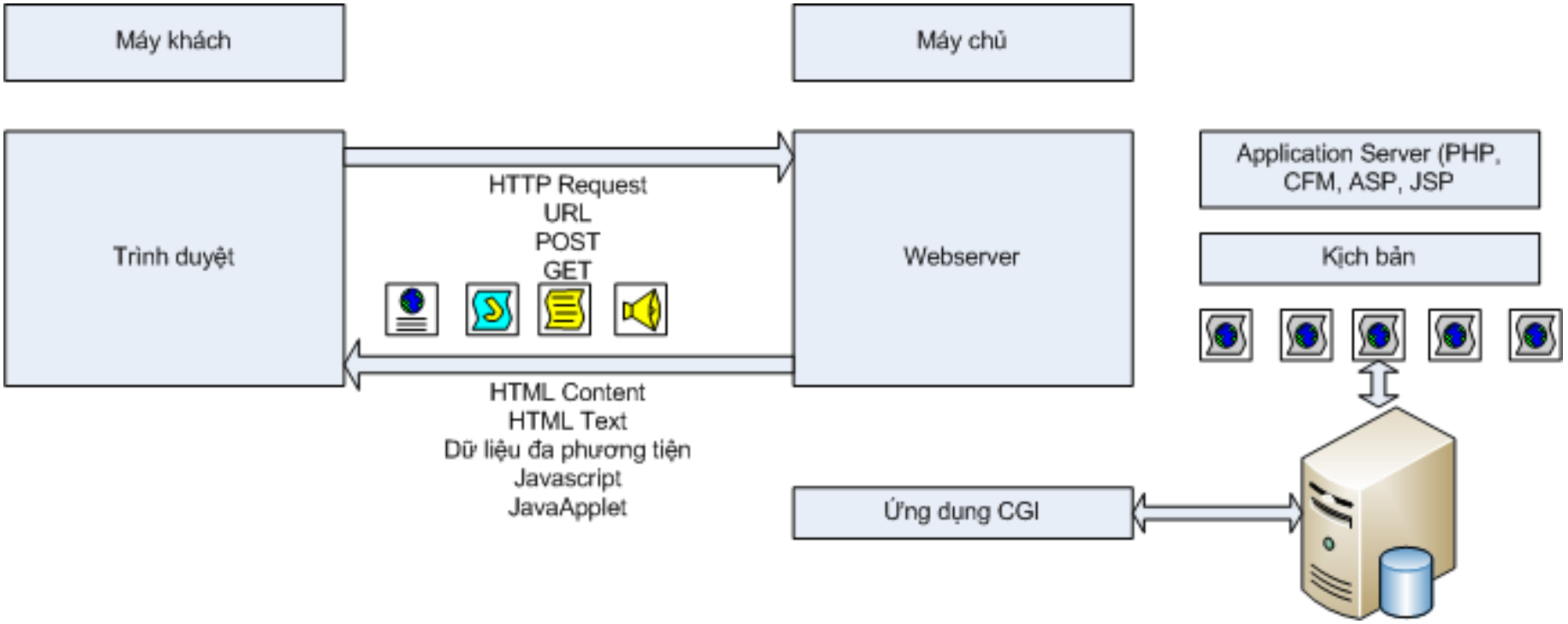
Web Server với Linux

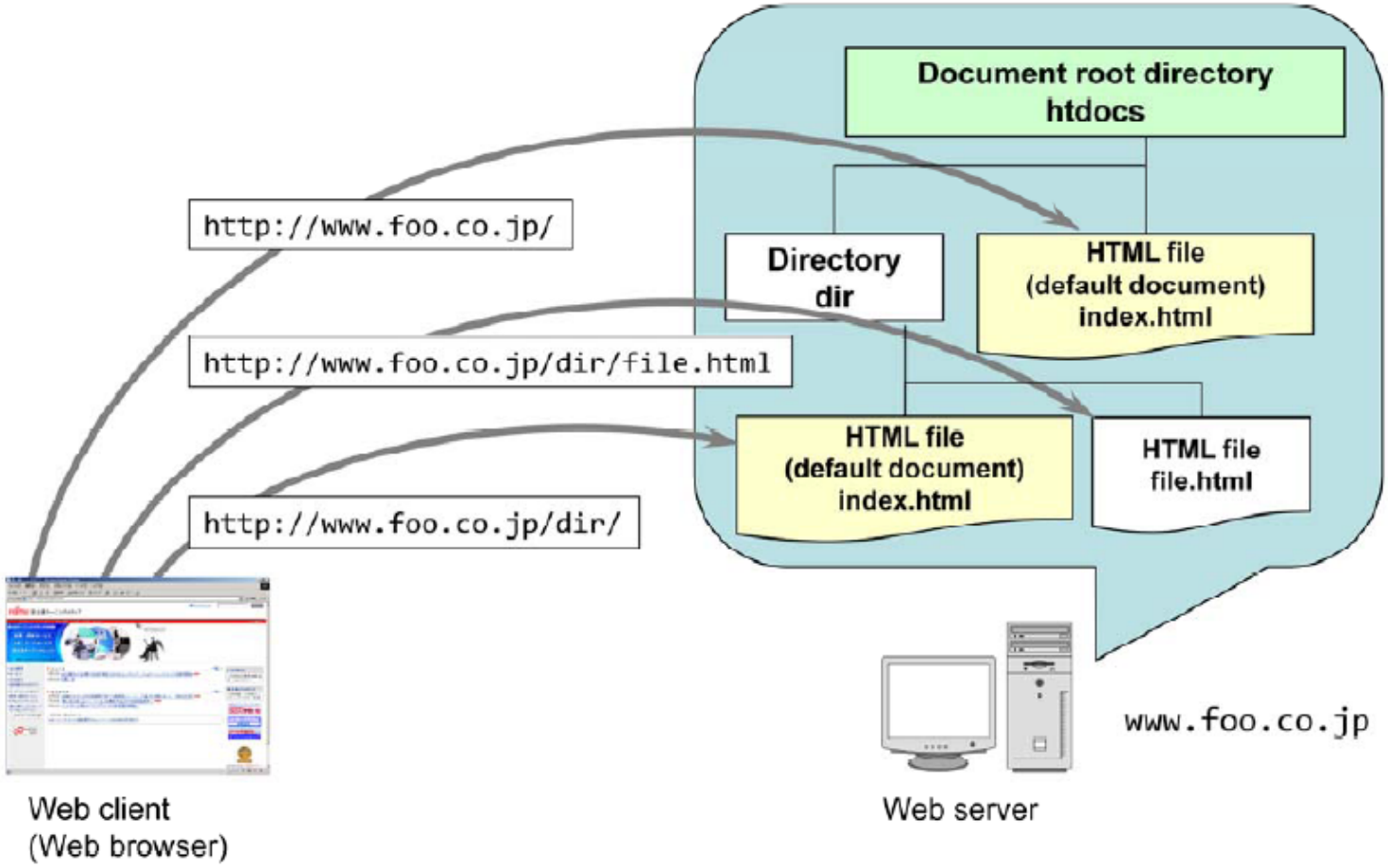
Nội dung

- Cơ chế của dịch vụ web
- Cài đặt và cấu hình webserver
- Các thao tác cơ bản của webserver
- Khái niệm SSL
- Sử dụng SSL

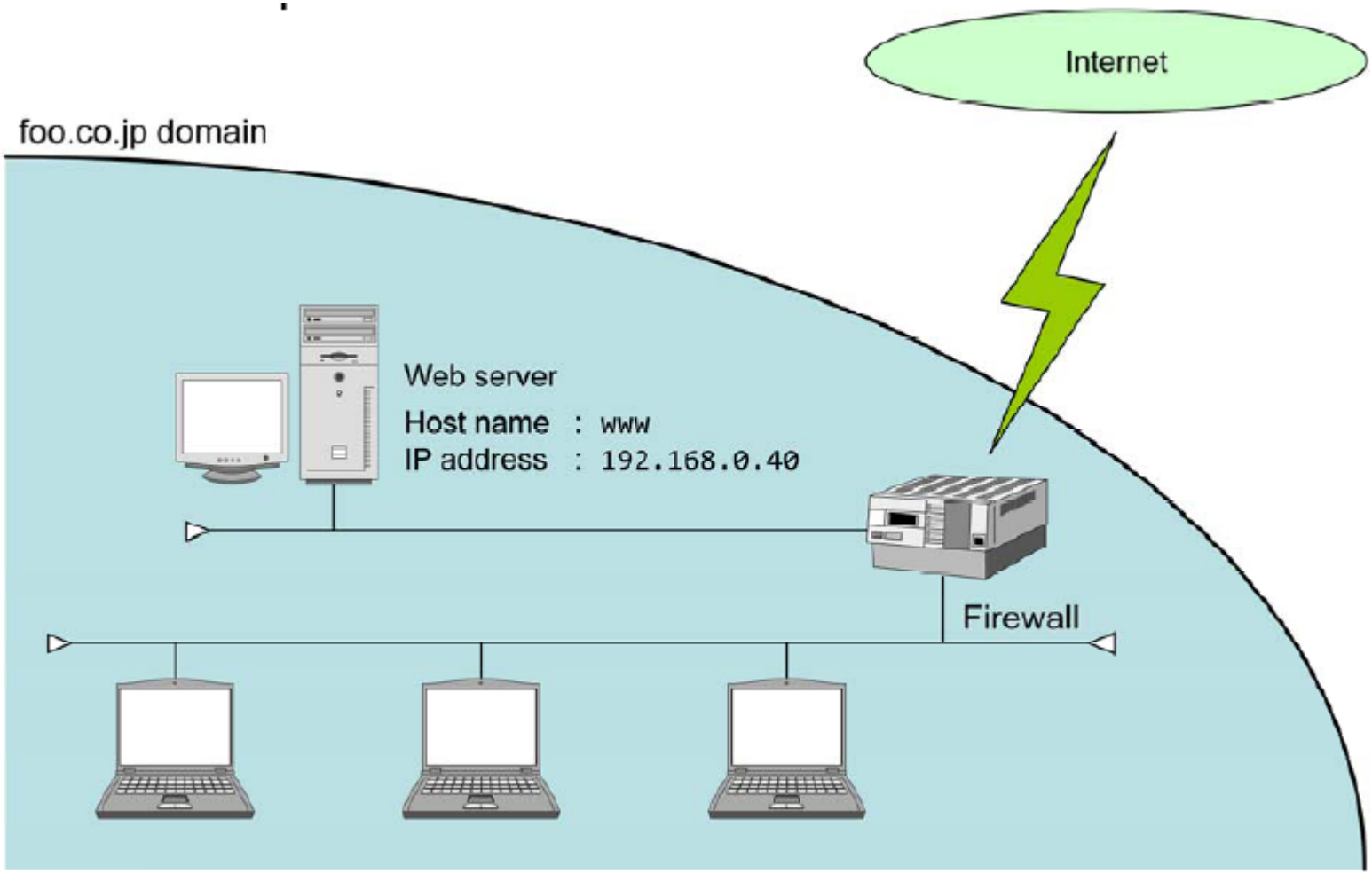
Cơ chế của dịch vụ web



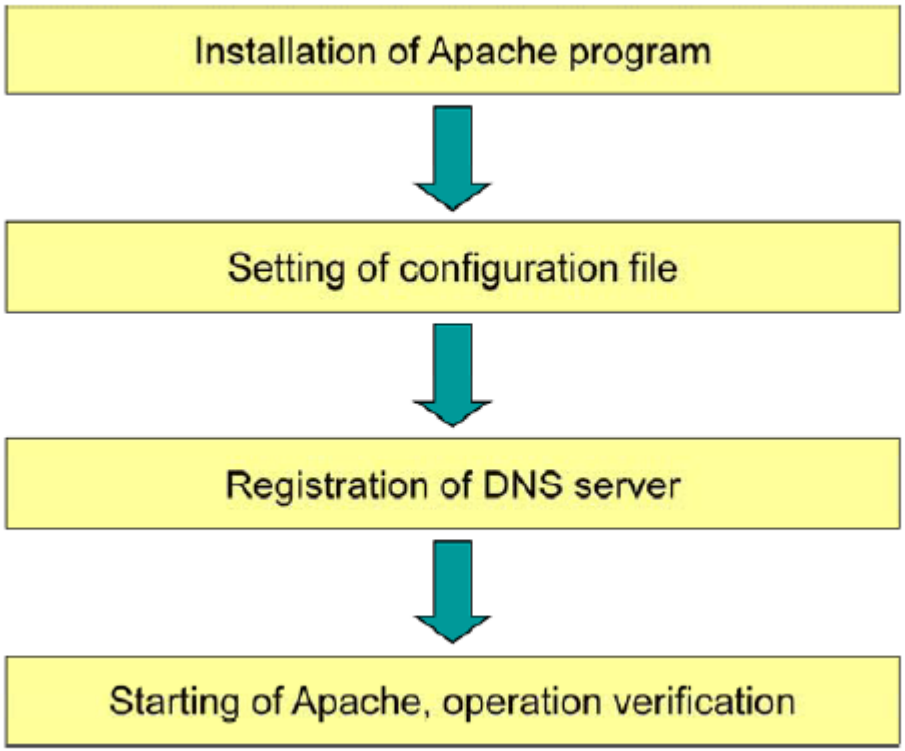
URL và địa chỉ vật lý



Triển khai webserver



Qui trình cài đặt



Cài đặt Apache

1. Obtaining of Apache

```
Official site of apache
The Apache Software Foundation        http://www.apache.org/
```

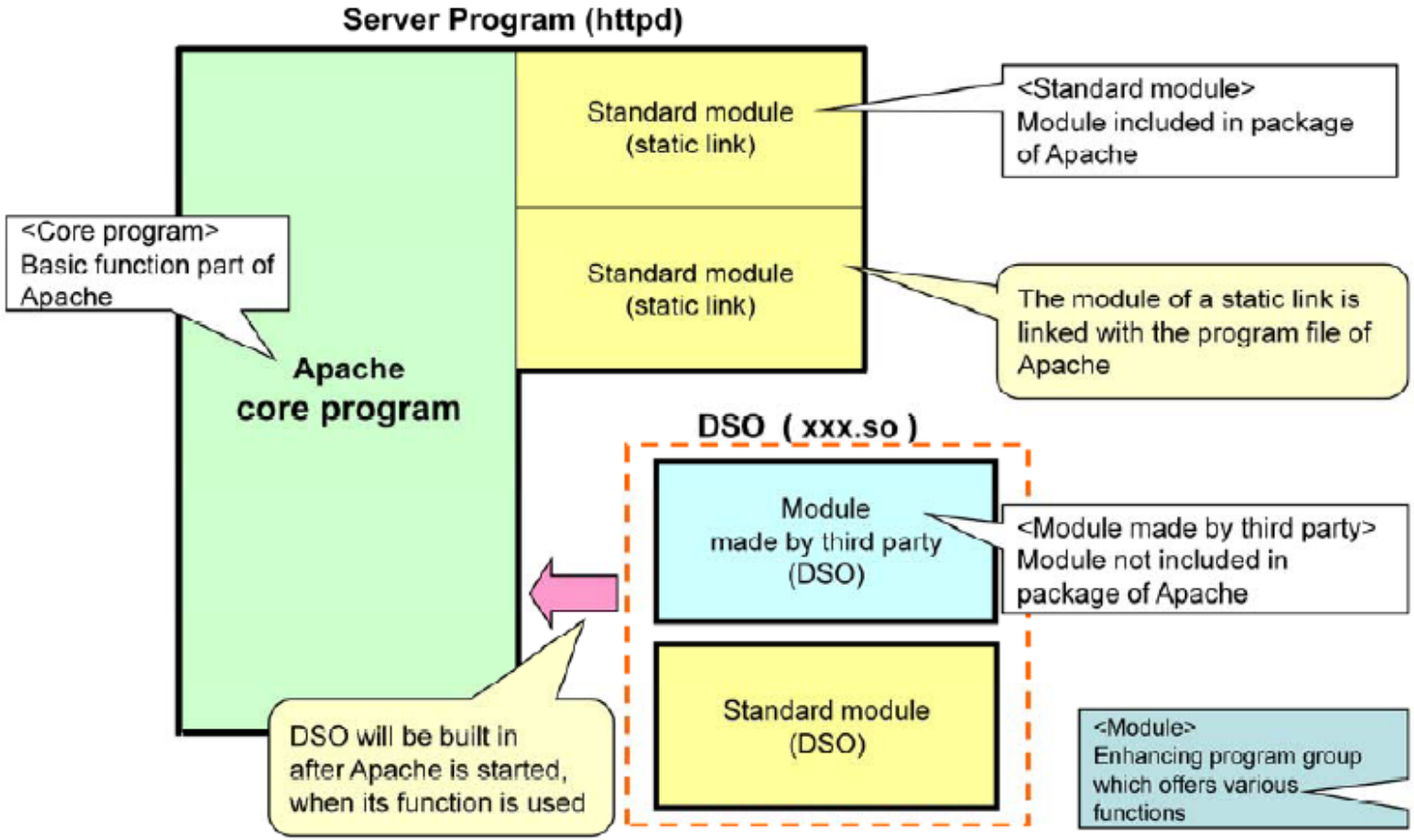
2. Decompression and development

```
# gzip -cd httpd-2.0.50.tar.gz | tar xvf -
          :
# ls -F httpd-2.0.50/
ABOUT_APACHE  InstallBin.dsp  NWGNUmakefile     apachenw.mcp.zip  docs/         os/
Apache.dsp    LAYOUT          README            build/            emacs-style   server/
Apache.dsw    LICENSE         README.platforms  buildconf*        httpd.spec    srclib/
BuildBin.dsp  Makefile.in     VERSIONING        config.layout     include/      support/
CHANGES       Makefile.win    acconfig.h        configure*        libhttpd.dsp  test/
INSTALL       NOTICE          acinclude.m4      configure.in      modules/
#
```

3. Compilation and installation

```
# cd httpd-2.0.50/
# ./configure --enable-ssl
          :
# make
          :
# make install
          :
#
```

Thành phần của Apache



Các mô đun mặc định

✓ **mod_access**	mod_disk_cache	✓ **mod_negotiation**
✓ **mod_actions**	mod_echo	mod_nw_ssl
✓ **mod_alias**	✓ **mod_env**	mod_proxy
✓ **mod_asis**	mod_example	mod_proxy_connect
✓ **mod_auth**	mod_expires	mod_proxy_ftp
mod_auth_anon	mod_ext_filter	mod_proxy_http
mod_auth_dbm	mod_file_cache	mod_rewrite
mod_auth_digest	mod_headers	✓ **mod_setenvif**
mod_auth_ldap	✓ **mod_imap**	✓ **mod_so**
✓ **mod_autoindex**	✓ **mod_include**	mod_speling
mod_cache	mod_info	mod_ssl
mod_cern_meta	✓ **mod_isapi**	✓ **mod_status**
✓ **mod_cgi**	mod_ldap	mod_suexec
mod_cgid	✓ **mod_log_config**	mod_unique_id
mod_charset_lite	mod_log_forensic	✓ **mod_userdir**
mod_dav	✓ **mod_logio**	mod_usertrack
mod_dav_fs	mod_mem_cache	mod_vhost_alias
mod_deflate	✓ **mod_mime**	
✓ **mod_dir**	mod_mime_magic	※ The module with ✓ sign is built in by default.

Cấu hình các mô đun (tĩnh)

- Module built-in

```
# cd httpd-2.0.50/
# ./configure --enable-ssl --disable-userdir
          :
# make
          :
# make install
          :
#
```

Module 'mod_ssl' is added, and module 'mod_userdir' is deleted

- Verification of module built-in

```
# /usr/local/apache2/bin/httpd -l
Compiled in modules:
  core.c                    mod_status.c
  mod_access.c              mod_autoindex.c
  mod_auth.c                mod_asis.c
  mod_include.c             mod_cgi.c
  mod_log_config.c mod_negotiation.c
  mod_env.c                 mod_dir.c
  mod_setenvif.c            mod_imap.c
  mod_ssl.c                 mod_actions.c
  prefork.c                 mod_alias.c
  http_core.c               mod_so.c
  mod_mime.c
#
```

It displays static linked modules list

We can see that the module 'mod_ssl' is built-in

We can see that module 'mod_userdir', which is usually built in by default, is not built-in

※ We have changed lines for module list to display for convenience of space. Originally, the execution result of 'httpd -l' is displayed continuously

Cấu hình các mô đun (động)

- Module built-in

```
# cd httpd-2.0.50/
# ./configure --enable-echo=shared
          :
# make
          :
# make install
          :
#
```

Module 'mod_echo' is added as DSO.

- Verification of module built-in

```
# cd /usr/local/apache2/modules/
# ls
httpd.exp  mod_echo.so
#
```

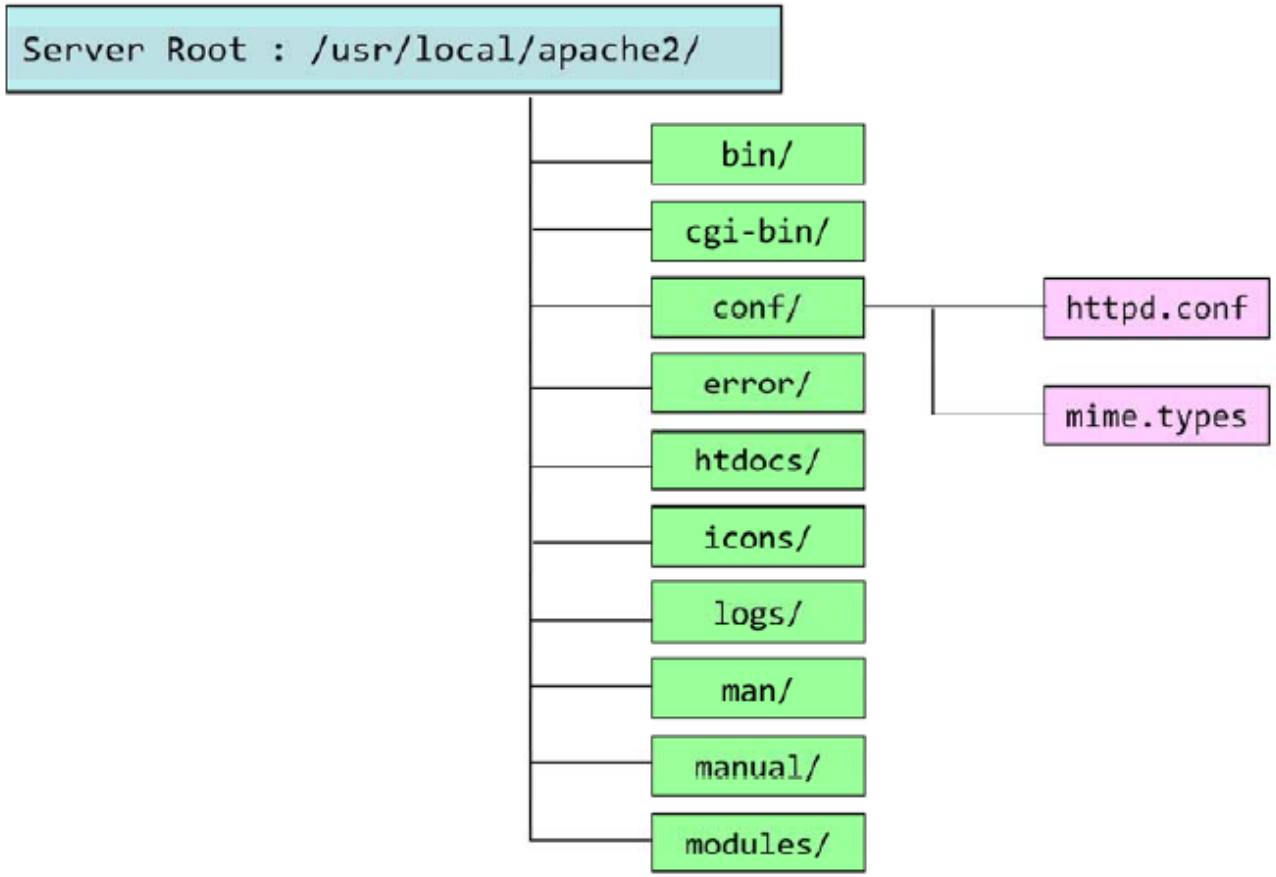
It verifies that module 'mod_echo' is installed under '/usr/local/apach2/modules'.

- Verification of module built-in (verification of '/usr/local/apache2/conf/httpd.conf' file)

```
#
# Dynamic Shared Object (DSO) Support
#
          :
LoadModule echo_module modules/mod_echo.so
```

Verify description to build-in module 'mod_echo' has been added.

Thư mục Apache



Các tệp cấu hình

- /etc/apache2.conf
- Có thể gắn thêm nhiều tệp khác

Các môi trường cấu hình

- <Directory>
- <DirectoryMatch>
- <Files>
- <FilesMatch>
- <IfDefine>
- <IfModule>
- <IfVersion>
- <Location>
- <LocationMatch>
- <Proxy>
- <ProxyMatch>
- <VirtualHost>

Các lệnh cấu hình

- Chỉ dùng một dòng, không có thẻ mở và đóng
- Có thể nằm ở ngoài hoặc trong một môi trường
- Chỉ có tác dụng trong môi trường

Cấu hình cơ bản (httpd.conf)

```
ServerRoot      "/usr/local/apache2"
Listen  80
User    nobody
Group   nobody
ServerAdmin     webmaster@foo.co.jp
ServerName      www.foo.co.jp:80
DocumentRoot    "/usr/local/apache2/htdocs"
DirectoryIndex  index.html index.html.var
```

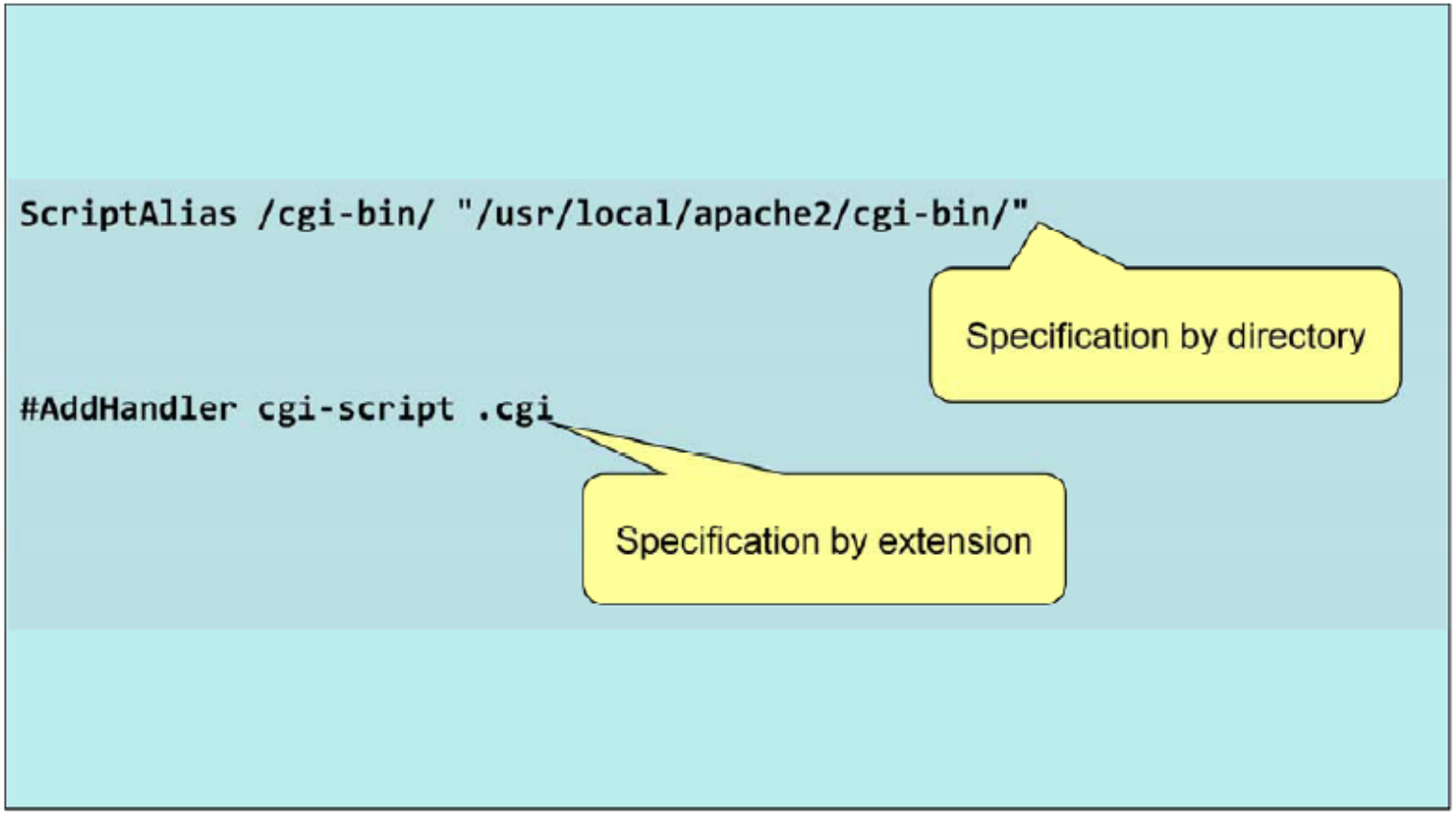
Default value : #-1

Default value : you@example.com

Default value : www.example.com:80

CGI

Editing of '/usr/local/apache2/conf/httpd.conf' file



Khai báo một thư mục

- <directory></directory>
- .htaccess
- Allowoverride [None, Directive]
  - Sử dụng Directive trong .htaccess
- Allow, Deny
  - Allow from all
  - Allow from 192.168.192.0/18 hut.edu.vn
- Order Allow,Deny

Options

- All
- ExecCGI
- FollowSymLinks
- Includes Server-side
- IncludesNOEXEC
- Indexes
- MultiViews
- SymLinksIfOwnerMatch

.htaccess

- Khai báo các thuộc tính của thư mục
- Được apache đọc và kích hoạt trực tiếp

- <Directory>
- <DirectoryMatch>
- <Files>
- <FilesMatch>
- <IfDefine>
- <IfModule>
- <IfVersion>
- <Location>
- <LocationMatch>
- <Proxy>
- <ProxyMatch>
- <VirtualHost>

Cấu hình DNS

1. Editting the zone data file

Edit the forward lookup zone data file

```
        :
www.foo.co.jp.              IN     A        192.168.0.40
        :
```

Edit the reverse lookup zone data file

```
        :
40.0.168.192.in-addr.arpa.          IN     PTR      www.foo.co.jp.
        :
```

2. Restart of DNS server

```
# kill `cat /var/run/named.pid`
# /usr/local/sbin/named
```

3. Verification of registered information

```
# host www.foo.co.jp
www.foo.co.jp has address 192.168.0.40
#
```

Khởi động webserver

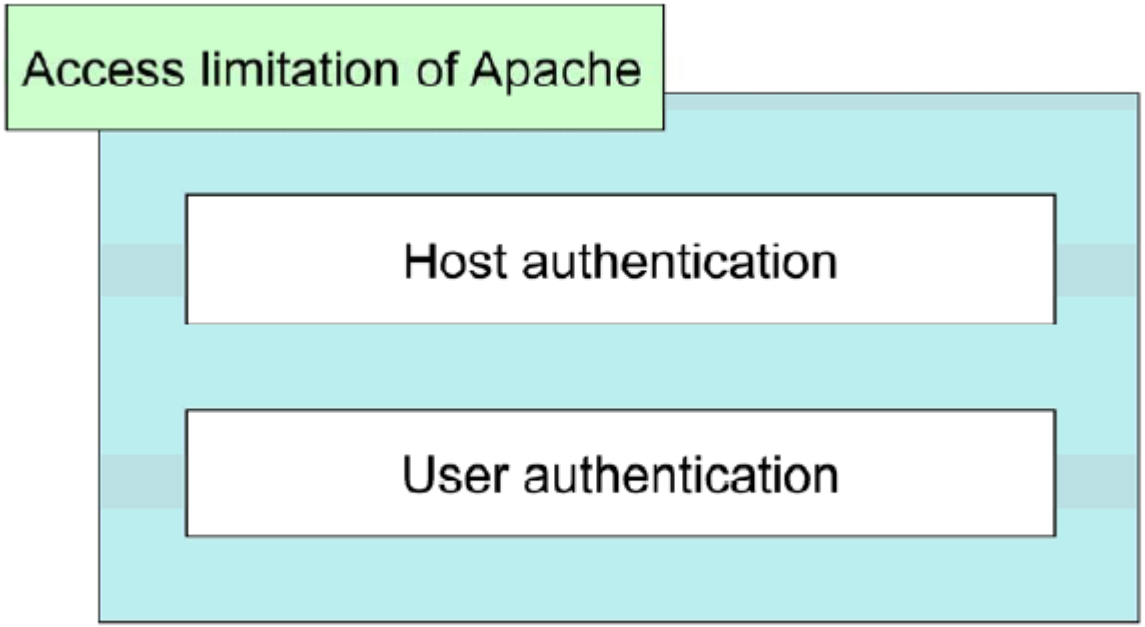
- Web server startup by ‘ apachectl’ command

```
# /usr/local/apache2/bin/apachectl start
#
```

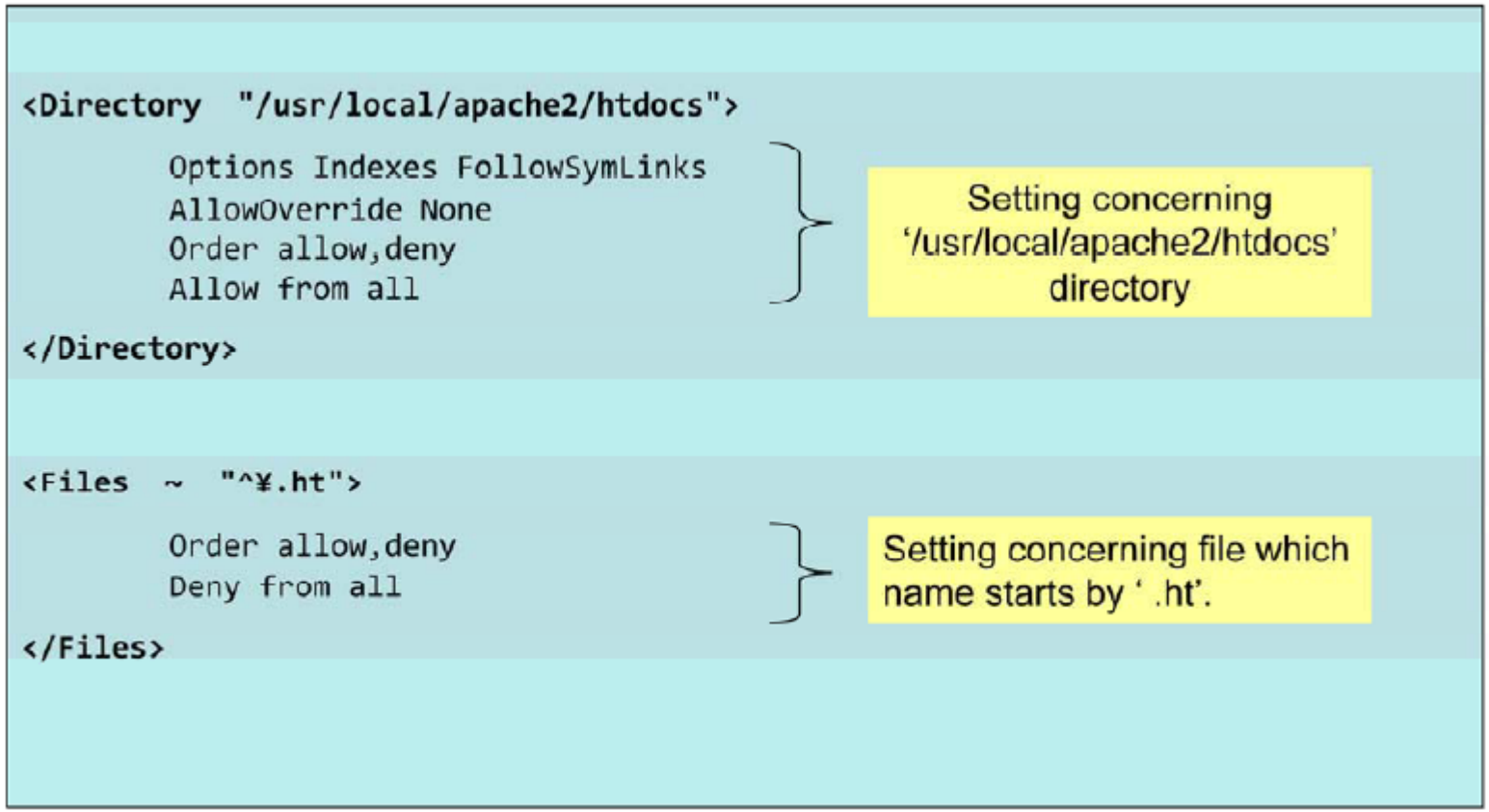
- Verification of Web server’s startup

```
# ps -ef | grep httpd
root     15633     1 16 17:26 ?        00:00:01 /usr/local/apache2/bin/httpd -k start
nobody   15634 15633  0 17:26 ?        00:00:00 /usr/local/apache2/bin/httpd -k start
nobody   15635 15633  0 17:26 ?        00:00:00 /usr/local/apache2/bin/httpd -k start
nobody   15636 15633  0 17:26 ?        00:00:00 /usr/local/apache2/bin/httpd -k start
nobody   15637 15633  0 17:26 ?        00:00:00 /usr/local/apache2/bin/httpd -k start
nobody   15638 15633  0 17:26 ?        00:00:00 /usr/local/apache2/bin/httpd -k start
#
```

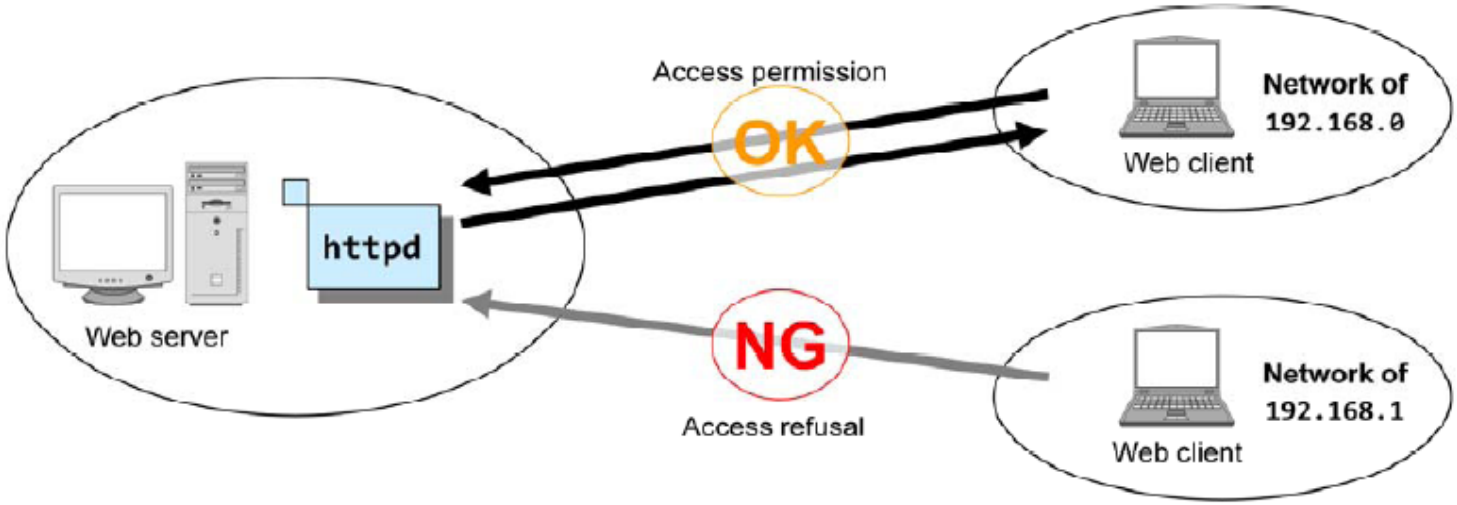
Bảo mật webserver



Quản lý truy cập tài nguyên



Kiểm soát máy tính



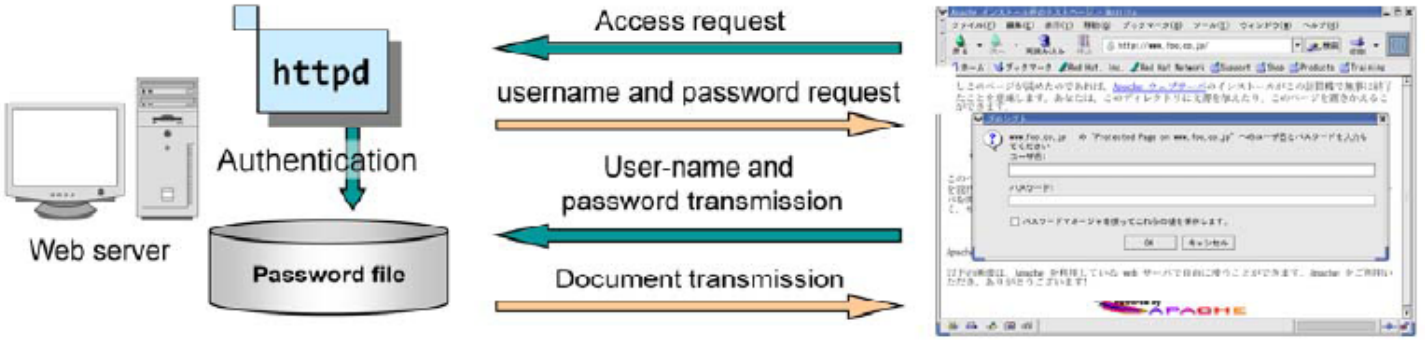
Editing '/usr/local/apache2/conf/httpd.conf' file

```
<Directory  "/usr/local/apache2/htdocs">

    Order  deny,allow
    Allow  from  192.168.0.0/24
    Deny   from  all

</Directory>
```

Kiểm soát tài khoản



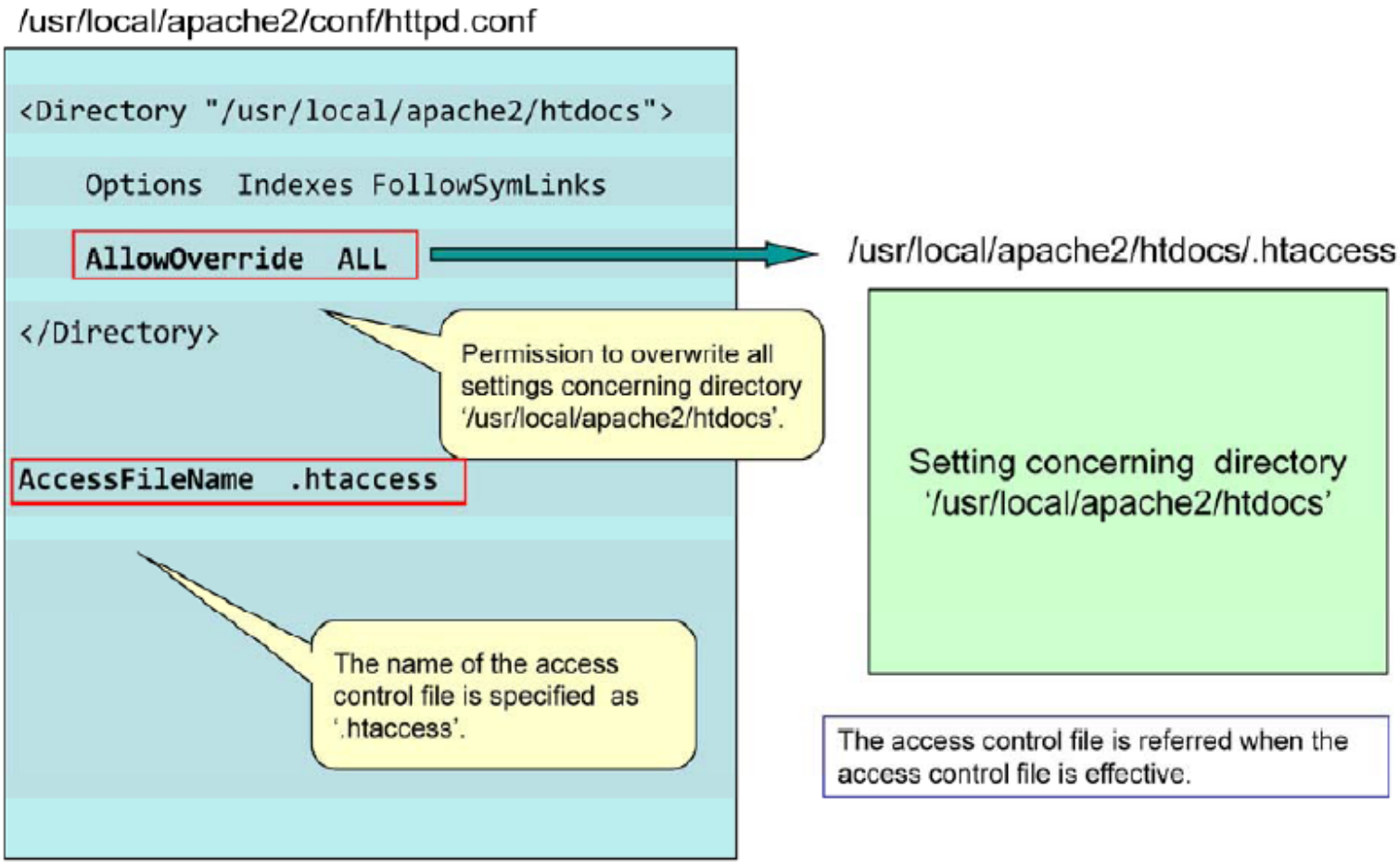
Editing '/usr/local/apache2/conf/httpd.conf' file

```
<Directory "/usr/local/apache2/htdocs">
    AuthType      Basic
    AuthName      "Protected Page on www.foo.co.jp"
    AuthUserFile  /usr/local/apache2/conf/.htpasswd
    require  user          taro hanako
</Directory>
```

Password registration by 'htpasswd' command

```
# /usr/local/apache2/bin/htpasswd -c /usr/local/apache2/conf/.htpasswd taro
New password:
Re-type new password:
Adding password for user taro
#
```

Liên hệ giữa tệp cấu hình và tệp kiểm soát tài khoản



Log

Access log

```
192.168.0.40 - - [12/Aug/2004:20:05:42 +0900] "GET / HTTP/1.1" 401 512
192.168.0.40 - taro [12/Aug/2004:20:07:24 +0900] "GET / HTTP/1.1" 200 1630
192.168.0.40 - taro [12/Aug/2004:20:07:25 +0900] "GET /apache_pb.gif HTTP/1.1"
        :
```

Error log

```
[Fri Aug 13 10:29:30 2004] [error] [client 192.168.0.40] File does not exist:
/usr/local/apache2/htdocs/test.html
[Fri Aug 13 10:29:43 2004] [error] [client 192.168.0.40] (13)Permission denied: exec of
'/usr/local/apache2/cgi-bin/printenv' failed
        :
```

Agent log

```
Mozilla/5.0 (X11; U; Linux i686; ja-JP; rv:1.4) Gecko/20030922
Mozilla/5.0 (X11; U; Linux i686; ja-JP; rv:1.4) Gecko/20030922
Mozilla/5.0 (X11; U; Linux i686; ja-JP; rv:1.4) Gecko/20030922
        :
```

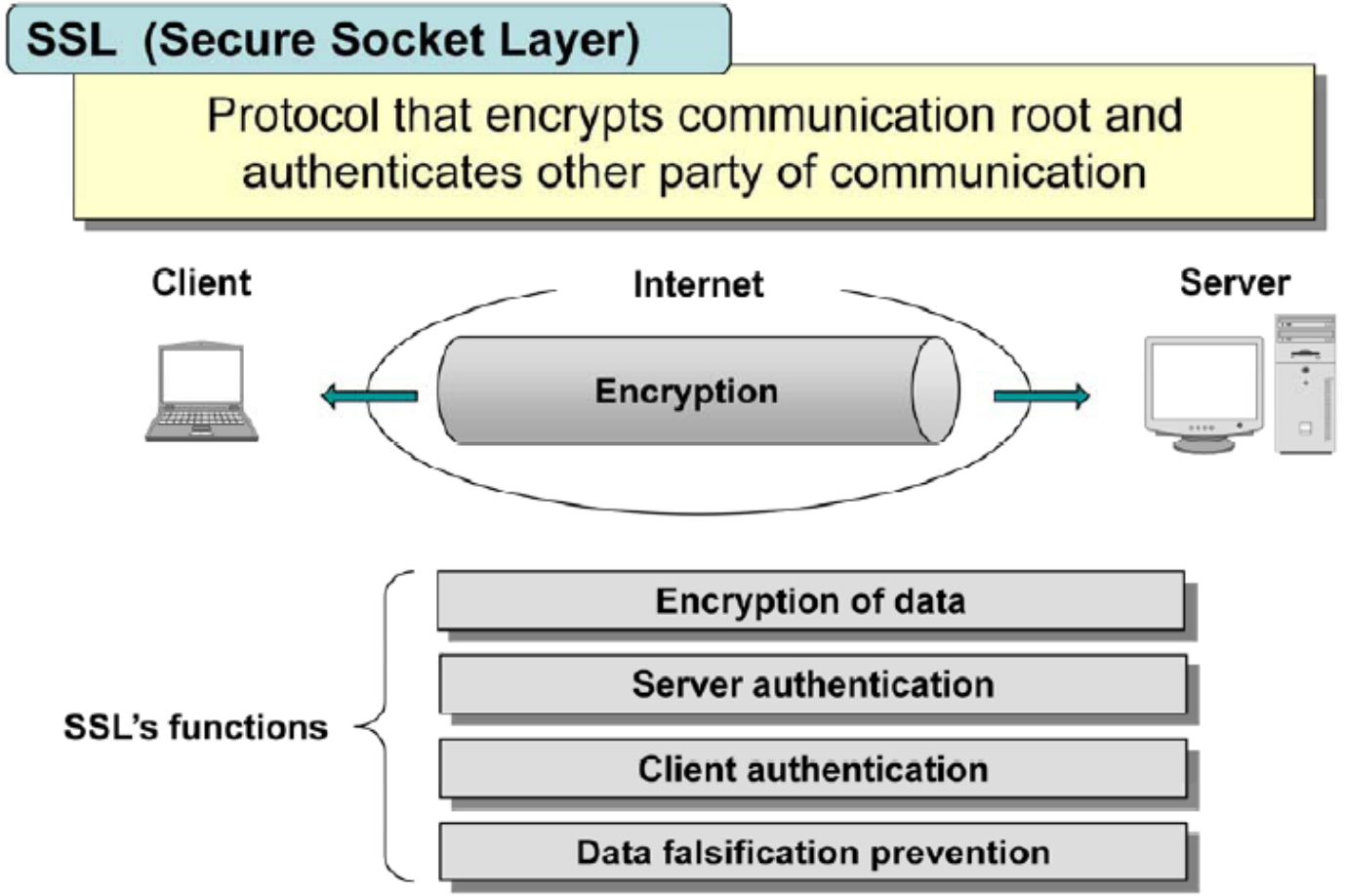
Referrer log

```
- -> /index.html.var
http://www.foo.co.jp/ -> /apache_pb.gif
- -> /cgi-bin/printenv
        :
```

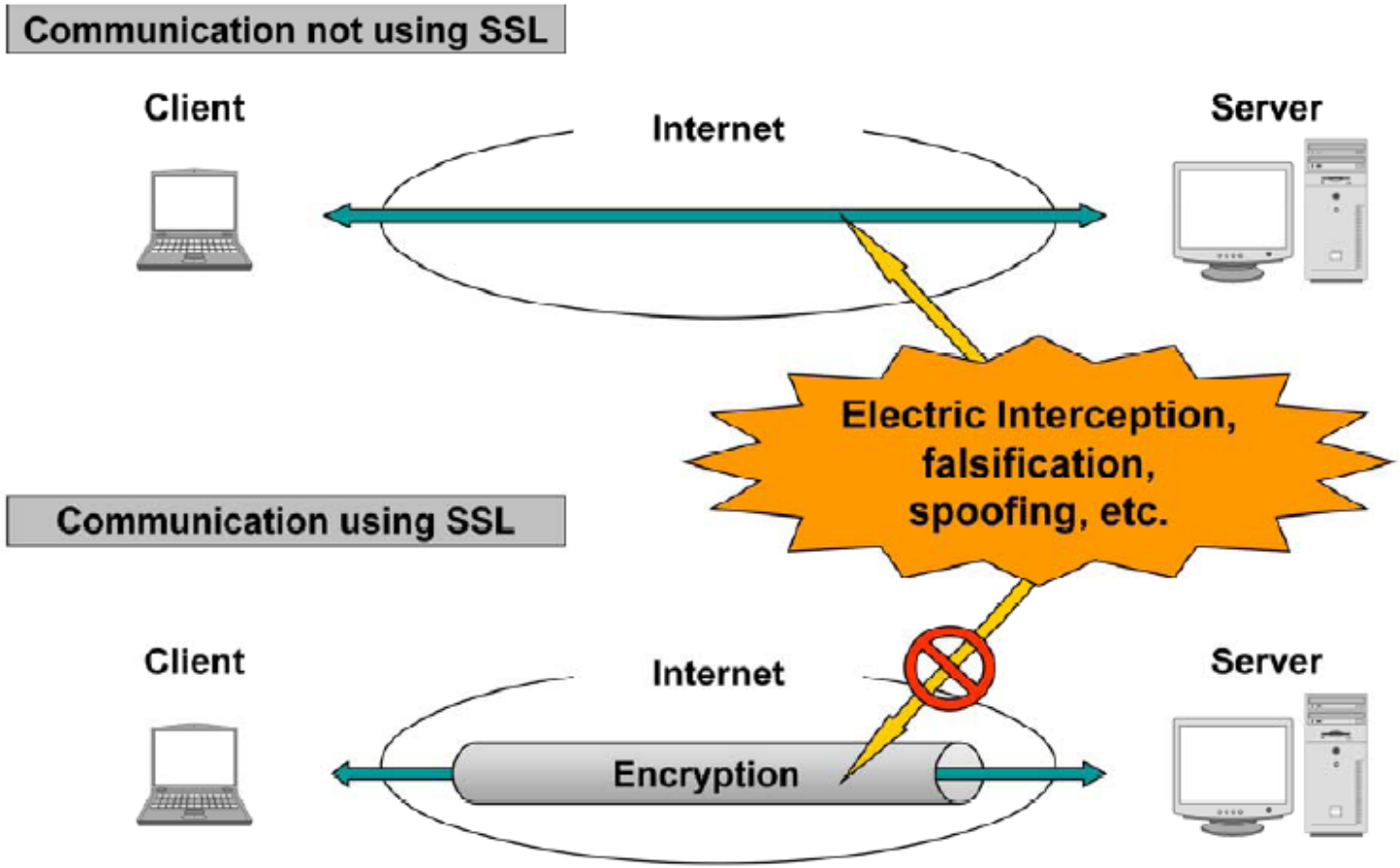
Cấu hình log

```
HostnameLookups Off
        :
ErrorLog logs/error_log
        :
LogLevel warn
        :
LogFormat "%h %l %u %t ¥"%r¥" %>s %b ¥"%{Referer}i¥" ¥"%{User-Agent}i¥""
combined
LogFormat "%h %l %u %t ¥"%r¥" %>s %b" common
LogFormat "%{Referer}i -> %U" referer
LogFormat "%{User-agent}i" agent
        :
CustomLog logs/access_log common
        :
#CustomLog logs/referer_log referer
#CustomLog logs/agent_log agent
```

Khái niệm SSL



Sự cần thiết của SSL



Các loại mã hóa

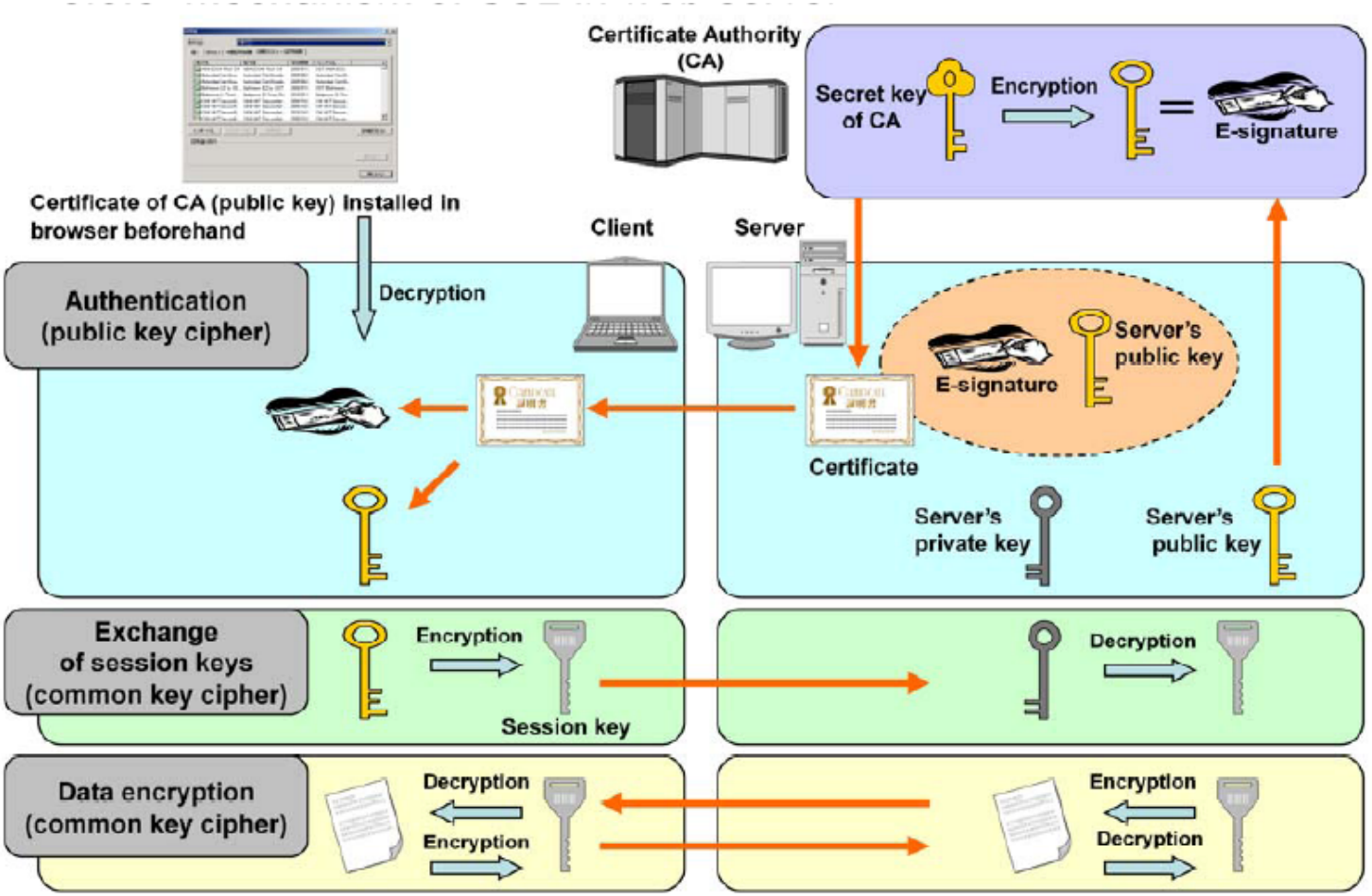
Common key cipher

- Use the same key for the encryption and the decryption.
- The mechanism of the cipher communication being simple, the encryption/decryption load is light.
- **The problem is: how to receive and to pass the key ?**

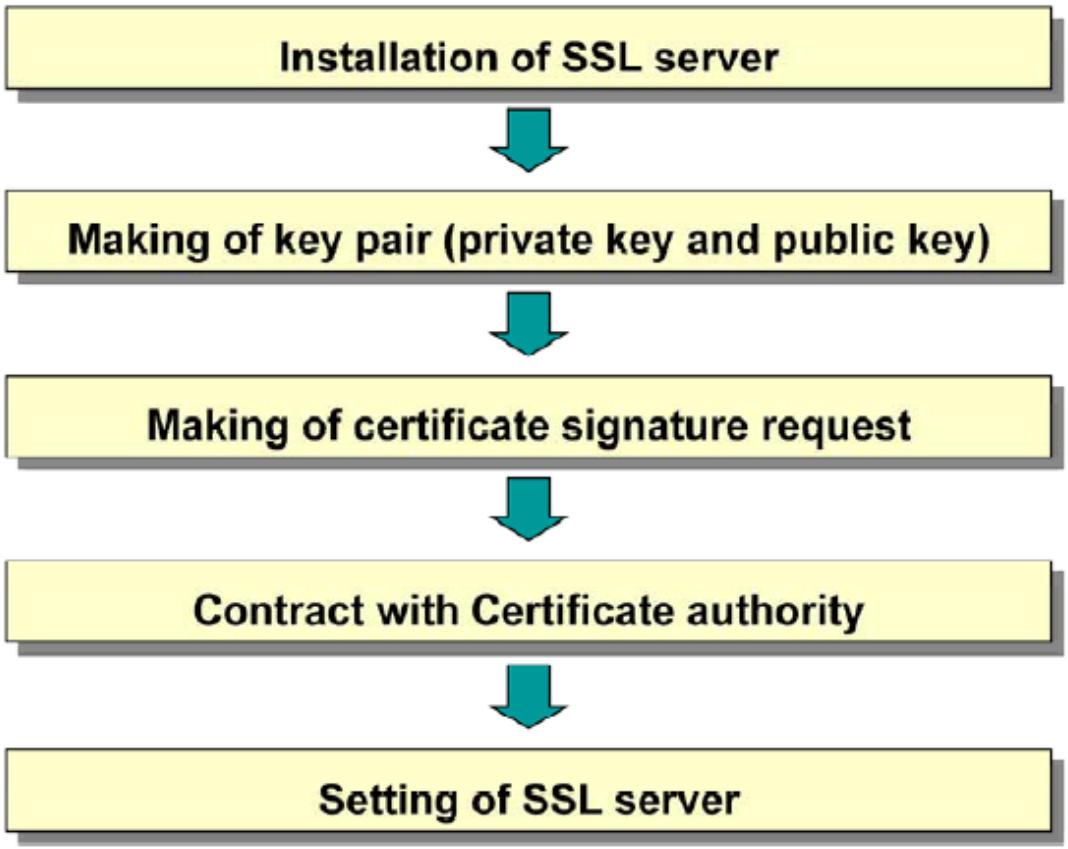
Public key cipher

- Use a different key to encrypt and to decode.
- Make **a pair of public key and private key**, and **open the public key** to the public.
- Only the private key, that forms a pair, can decipher the information encrypted by the public key.
- Only the public key, that forms a pair, can decipher the information encrypted by the private key.
- The mechanism of the cipher communication being complex, the load is heavy.

Cơ chế sử dụng SSL



Cài đặt SSL


Installation of SSL server
Making of key pair (private key and public key)
Making of certificate signature request
Contract with Certificate authority
Setting of SSL server

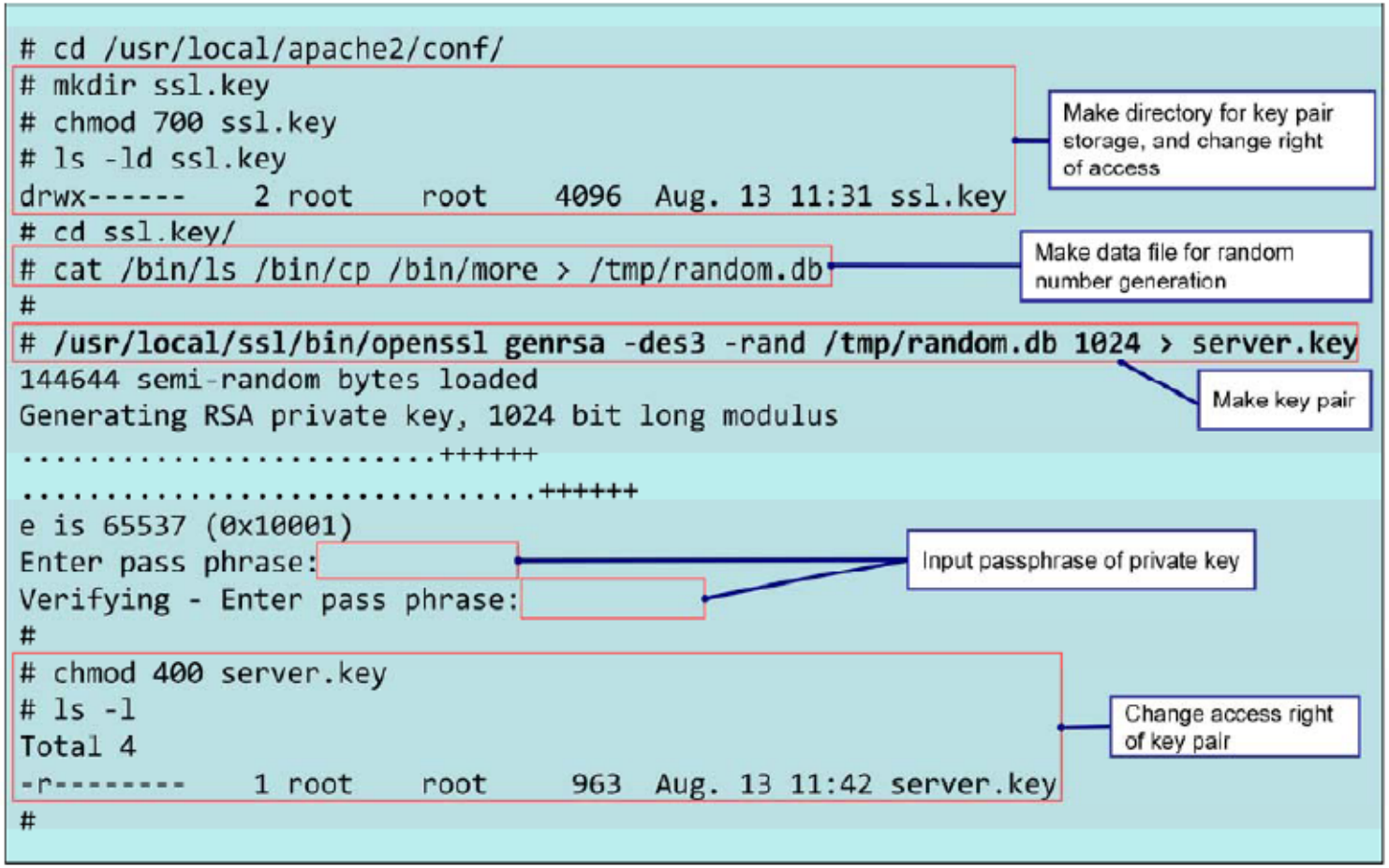
Cài đặt SSL

```
# gzip -cd openssl-0.9.7d.tar.gz | tar xvf -
# cd openssl-0.9.7d/
# ./config
        :
# make
        :
# make install
        :
#
```

- Compilation and installation of Apache

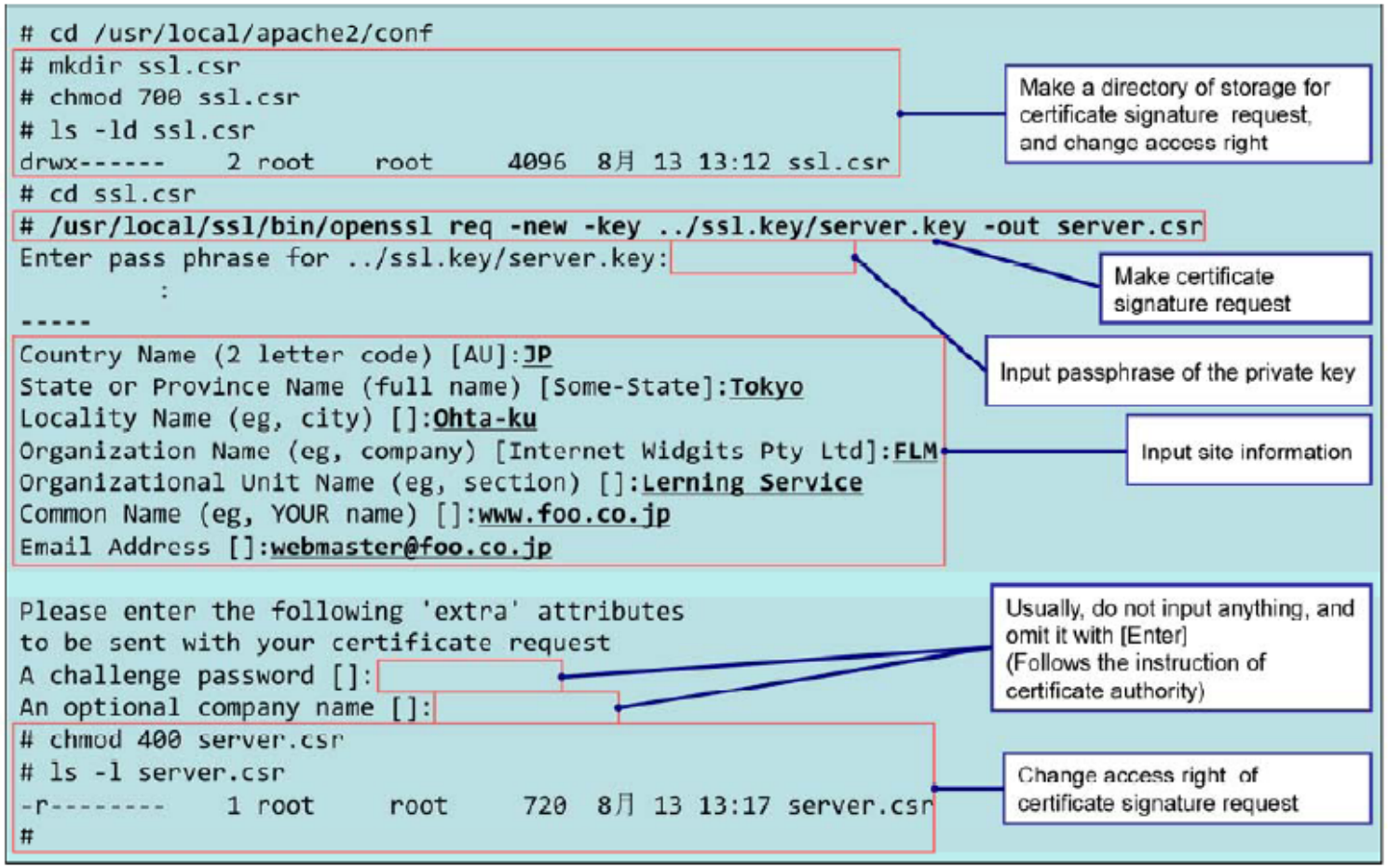
```
# PATH=/usr/local/ssl:/usr/local/ssl/bin:$PATH ;export PATH
#
# cd httpd-2.0.50/
# ./configure --enable-ssl
        :
# make
        :
# make install
        :
#
```

Quản lý cặp khóa




```
# cd /usr/local/apache2/conf/
# mkdir ssl.key
# chmod 700 ssl.key
# ls -ld ssl.key
drwx------    2 root    root    4096  Aug. 13 11:31 ssl.key
# cd ssl.key/
# cat /bin/ls /bin/cp /bin/more > /tmp/random.db
#
# /usr/local/ssl/bin/openssl genrsa -des3 -rand /tmp/random.db 1024 > server.key
144644 semi-random bytes loaded
Generating RSA private key, 1024 bit long modulus
.........................++++++
.............................++++++
e is 65537 (0x10001)
Enter pass phrase:
Verifying - Enter pass phrase:
#
# chmod 400 server.key
# ls -l
Total 4
-r--------    1 root    root     963  Aug. 13 11:42 server.key
#
```

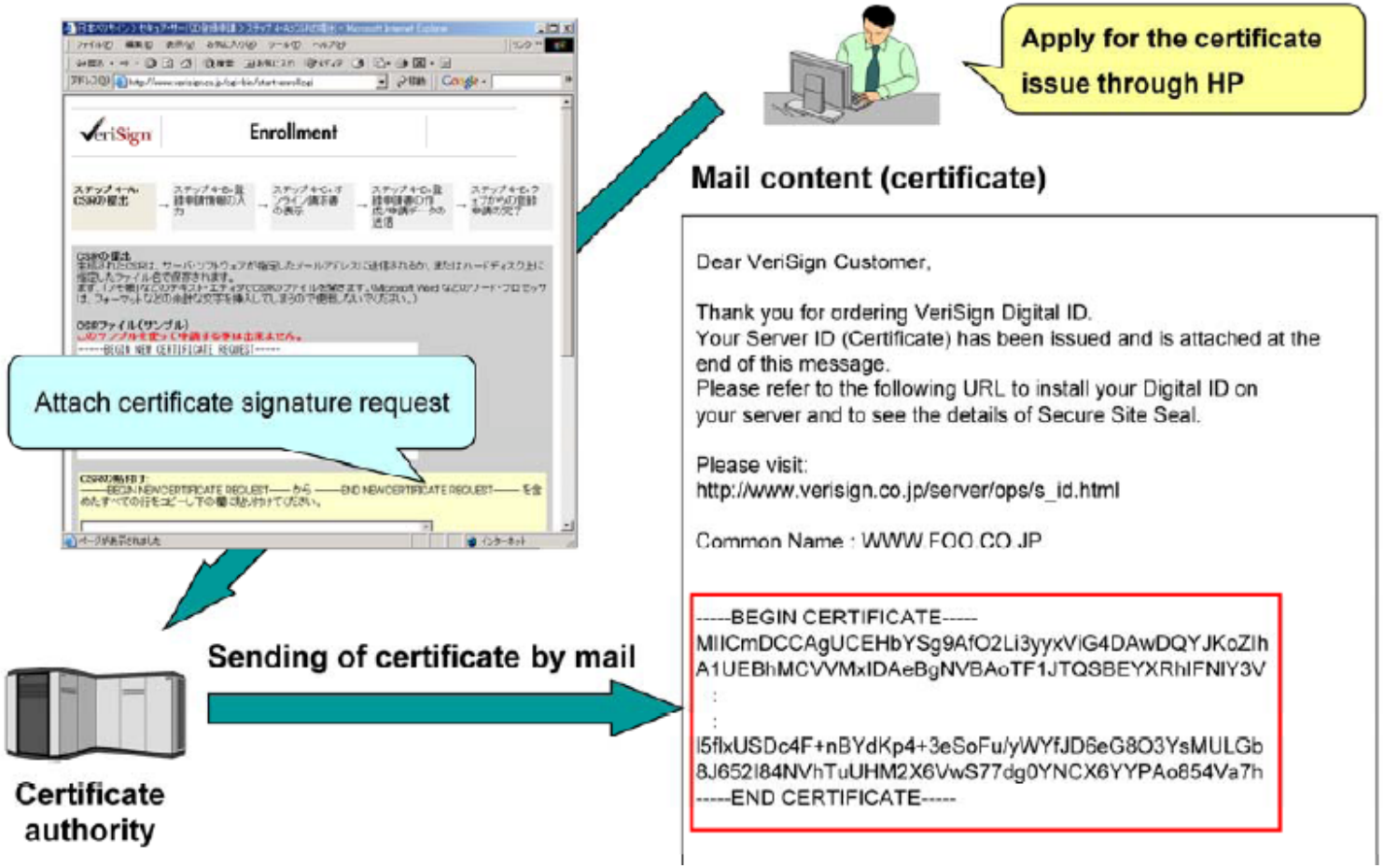
Ký bằng chứng chỉ




```
# cd /usr/local/apache2/conf
# mkdir ssl.csr
# chmod 700 ssl.csr
# ls -ld ssl.csr
drwx------    2 root    root     4096  8月 13 13:12 ssl.csr
# cd ssl.csr
# /usr/local/ssl/bin/openssl req -new -key ../ssl.key/server.key -out server.csr
Enter pass phrase for ../ssl.key/server.key:
        :
-----
Country Name (2 letter code) [AU]:JP
State or Province Name (full name) [Some-State]:Tokyo
Locality Name (eg, city) []:Ohta-ku
Organization Name (eg, company) [Internet Widgits Pty Ltd]:FLM
Organizational Unit Name (eg, section) []:Lerning Service
Common Name (eg, YOUR name) []:www.foo.co.jp
Email Address []:webmaster@foo.co.jp

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:
# chmod 400 server.csr
# ls -l server.csr
-r--------    1 root     root      720  8月 13 13:17 server.csr
#
```

Xác thực


Apply for the certificate issue through HP
Enrollment
Attach certificate signature request
Sending of certificate by mail
Certificate authority
Mail content (certificate)
Dear VeriSign Customer,
Thank you for ordering VeriSign Digital ID.
Your Server ID (Certificate) has been issued and is attached at the end of this message.
Please refer to the following URL to install your Digital ID on your server and to see the details of Secure Site Seal.
Please visit:
http://www.verisign.co.jp/server/ops/s_id.html
Common Name : WWW.FOO.CO.JP
-----BEGIN CERTIFICATE-----
MIICmDCCAgUCEHbYSg9AfO2Li3yyxViG4DAwDQYJKoZIh
A1UEBhMCVVMxIDAeBgNVBAoTF1JTQSBEYXRhIFNIY3V
:
:
l5flxUSDc4F+nBYdKp4+3eSoFu/yWYfJD6eG8O3YsMULGb
8J652I84NVhTuUHM2X6VwS77dg0YNCX6YYPAo854Va7h
-----END CERTIFICATE-----

Cấu hình server

Setting of certificate acquired from certificate authority

```
# mkdir /usr/local/apache2/conf/ssl.crt
#
# mv server.pem /usr/local/apache2/conf/ssl.crt/
```

Make directory for certificate authority storage

Store certificate acquired from certificate authority

Edting of '/usr/local/apache2/conf/ssl.conf' file

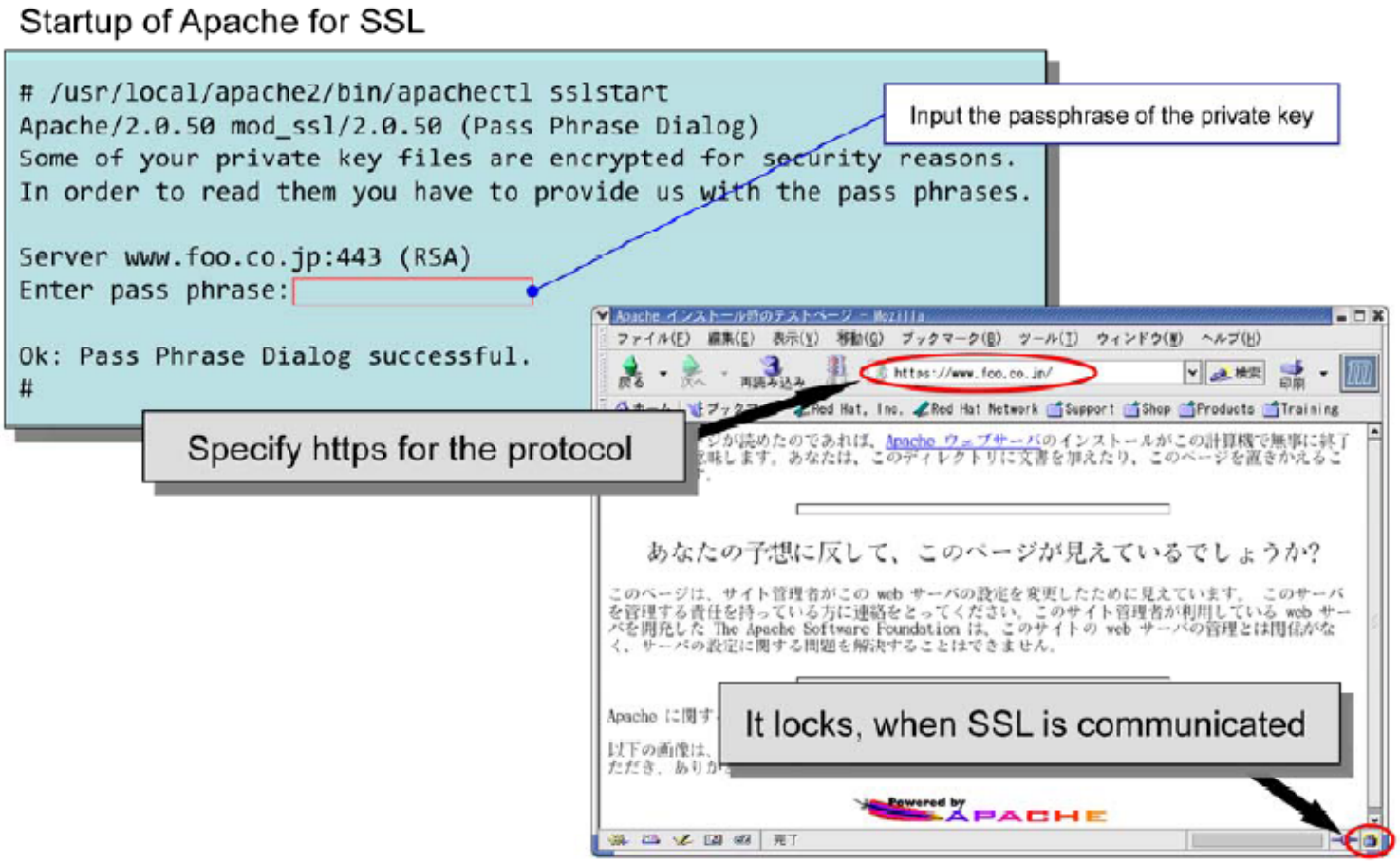
```
<IfDefine SSL>
        :
<VirtualHost _default_:443>
        :
ServerName www.foo.co.jp:443
ServerAdmin webmaster@foo.co.jp
        :
#   SSL Engine Switch:
SSLEngine on
        :
#   Server Certificate:
SSLCertificateFile /usr/local/apache2/conf/ssl.crt/server.pem
        :
#   Server Private Key:
SSLCertificateKeyFile /usr/local/apache2/conf/ssl.key/server.key
        :
</VirtualHost>
</IfDefine>
```

Use SSL (default)

Specify certificate's storage place

Specify storage place of key pair

Kiểm tra cấu hình



Giấu mật khẩu

Signature for private key

```
# cd /usr/local/apache2/conf/ssl.key
# /usr/local/ssl/bin/openssl rsa -in ./server.key -out ./server_signed.key
Enter pass phrase for ./server.key:
writing RSA key
#
```

Sign for private key

Input passphrase of private key

Editing '/usr/local/apache2/conf/ssl.conf' file

```
        :
#   Server Private Key:
SSLCertificateKeyFile /usr/local/apache2/conf/ssl.key/server_signed.key
        :
```

Set the signed private key to the server

Start of Apache for SSL

```
# /usr/local/apache2/bin/apachectl sslstart
#
```

Passphrase input is not requested

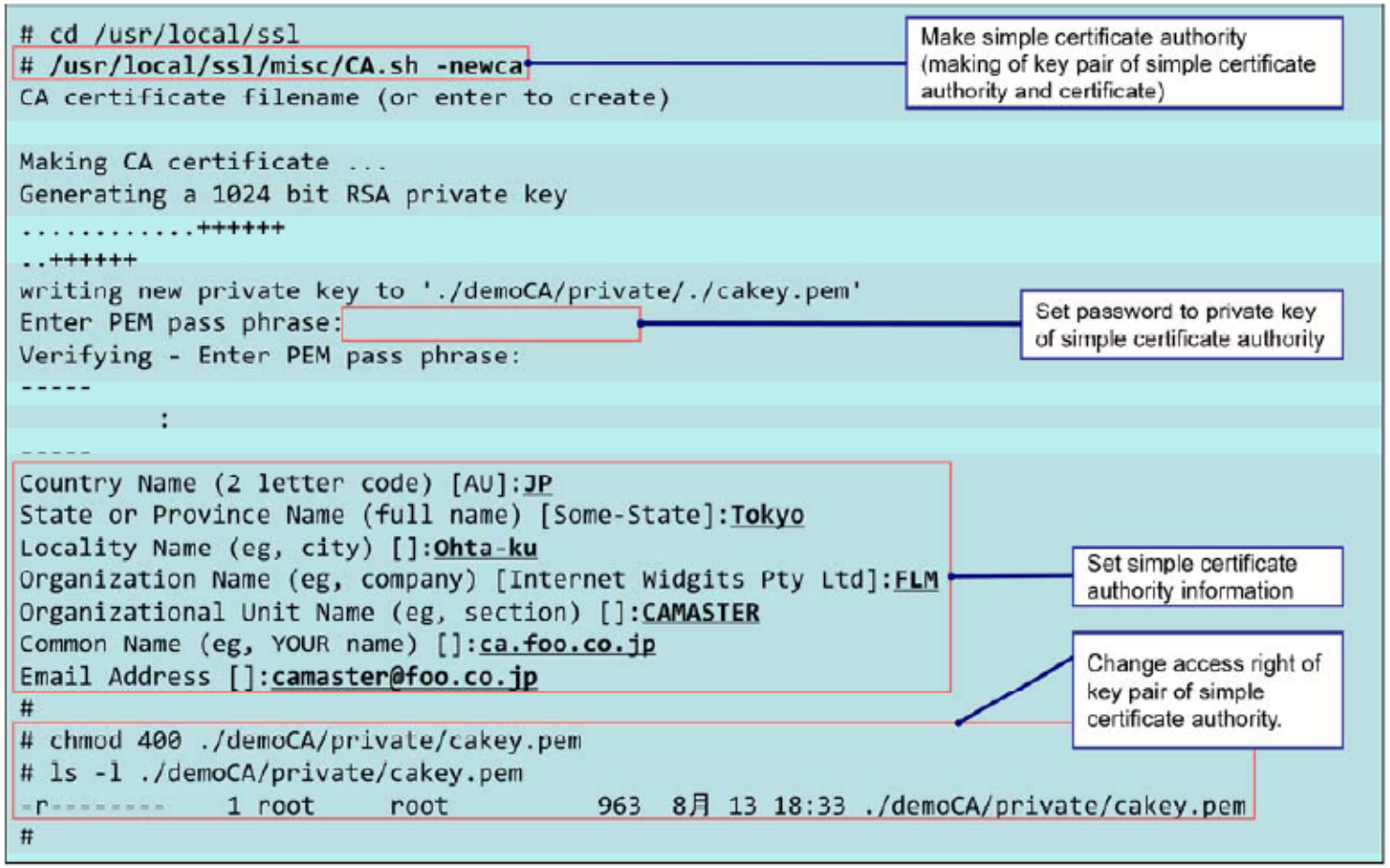
Cấu hình xác thực client

```
        :
#   Certificate Authority (CA):
#   Set the CA certificate verification path where to find CA
#   certificates for client authentication or alternatively one
#   huge file containing all of them (file must be PEM encoded)
#   Note: Inside SSLCACertificatePath you need hash symlinks
#         to point to the certificate files. Use the provided
#         Makefile to update the hash symlinks after changes.
SSLCACertificatePath /usr/local/apache2/conf/ssl.crt
SSLCACertificateFile /usr/local/apache2/conf/ssl.crt/cacert.pem

        :

#   Client Authentication (Type):
#   Client certificate verification type and depth.  Types are
#   none, optional, require and optional_no_ca.  Depth is a
#   number which specifies how deeply to verify the certificate
#   issuer chain before deciding the certificate is not valid.
SSLVerifyClient require
SSLVerifyDepth  10
        :
```

Chứng chỉ đơn giản




```
# cd /usr/local/ssl
# /usr/local/ssl/misc/CA.sh -newca
CA certificate filename (or enter to create)

Making CA certificate ...
Generating a 1024 bit RSA private key
............++++++
..++++++
writing new private key to './demoCA/private/./cakey.pem'
Enter PEM pass phrase:
Verifying - Enter PEM pass phrase:
-----
        :
-----
Country Name (2 letter code) [AU]:JP
State or Province Name (full name) [Some-State]:Tokyo
Locality Name (eg, city) []:Ohta-ku
Organization Name (eg, company) [Internet Widgits Pty Ltd]:FLM
Organizational Unit Name (eg, section) []:CAMASTER
Common Name (eg, YOUR name) []:ca.foo.co.jp
Email Address []:camaster@foo.co.jp
#
# chmod 400 ./demoCA/private/cakey.pem
# ls -l ./demoCA/private/cakey.pem
-r--------    1 root     root          963  8月 13 18:33 ./demoCA/private/cakey.pem
#
```