網路安全的理論與實務

楊中皇 著

第十六章 **OpenTSA**

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

伴 您 學 習 成 長 的 每 一 天

第十六章 OpenTSA

- **OpenTSA簡介**
- 安裝方法
- 使用介紹

OpenTSA發展歷史

- OpenTSA（http://www.opentsa.org/）是一套開放原始碼的時戳服務軟體計畫
- 目的在於發展既穩定又安全，符合RFC3161時戳協定標準規範
- 具備主從式架構的時戳服務中心（Time Stamp Authority, TSA）
- 主要由Zoltan Glozik負責，相關的軟體則由Eric Young及Tim Hudson所開發

OpenTSA發展歷史(續)

- 目前的OpenTSA包含以下三種功能：
- Time Stamp patch for OpenSSL: 以安裝Tarball套件方式安裝OpenSSL原始碼及時戳服務修補套件之後，OpenSSL便能夠使用時戳操作指令"ts"，此指令可執行產生時戳要求或時戳回應封包，及驗證時戳回應等相關功能
- Time Stamp client: 提供時戳客戶端以命令列的操作方式產生時戳要求，並透過HTTP或HTTPS協定連結TSA傳送時戳要求與接收TSA傳回的時戳回應，以及在接收時戳回應之後進行驗證
- Time Stamp module for Apache: 專為Apache HTTP Server設計的時戳服務模組。此模組的主要功能是用於結合Apache建立時戳服務伺服器，使OpenTSA可以透過HTTP或HTTPS協定提供時戳服務，此外它還能夠將時戳憑證儲存在MySQL或Firebird等兩種關聯式資料庫中

OpenTSA操作平台

- OpenTSA並非一套完整可獨立運行的軟體，它只是提供時戳服務的修補套件及伺服器模組
- 在使用上必須搭配OpenSSL與Apache，在OpenTSA的官方網站上有公布針對OpenSSL及Apache各版本所支援的修補套件及模組
- OpenTSA的操作平台同時取決於OpenSSL及Apache，只要某作業系統能夠以安裝Tarball套件方式安裝OpenSSL原始碼與時戳服務修補套件，以及可以安裝Apache HTTP Server與時戳服務模組，便能完整應用OpenTSA所提供的時戳服務
- 由此可知，幾乎所有的Unix-like系統都支援安裝及使用OpenTSA

安裝方法

- 介紹安裝OpenTSA以Fedora Core 4（FC4）爲安裝平台，選用的OpenSSL版本爲0.9.7e版，Apache的版本則爲1.3.33版。FC4預設安裝的OpenSSL是較新的0.9.7f版，但這並不影響OpenTSA的安裝與使用，因此也不需要先移除。
- 所使用的OpenSSL修補套件及Apache模組分別如下：
  - OpenSSL時戳服務修補套件：ts-20041109-0_9_7e-patch.gz
  - Apache時戳服務模組：mod_tsa-20041109.tgz
- 以下就分別針對這兩個部分說明相關的安裝步驟，安裝前請關閉SELinux功能，並以root權限登入系統之後，將所有需要安裝的套件檔案複製到root家目錄

安裝Perl5、libcurl（with perl binding）

- 在安裝OpenSSL時戳服務修補套件之後，會產生一支名爲"tsget"的時戳客戶端程式。由於執行該程式須要安裝Perl5及libcurl函式庫，因此事先說明這個部分的安裝
- FC4預設安裝Perl5版本爲5.8.6版，如果讀者安裝FC4時並未選取安裝Perl相關套件，請自行從FC4安裝光碟中新增安裝。而libcurl函式庫的安裝步驟如下：
  - tar xvfz WWW-Curl-2.0.tar.gz
  - cd WWW-Curl-2.0
  - perl Makefile.PL
  - make
  - make install

安裝OpenSSL及時戳服務修補套件

- 解壓縮OpenSSL原始碼到家目錄，並將ts-20041109-0_9_7e-patch.gz移到其解開後的目錄底下：
  - tar xvfz openssl-0.9.7e.tar.gz
  - mv ts-20041109-0_9_7e-patch.gz openssl-0.9.7e/
  - cd openssl-0.9.7e
- 利用patch指令更新OpenSSL原始碼以修補時戳服務的功能：
  - patch -p1 < ts-20041109-0_9_7e-patch.gz
- 執行安裝OpenSSL，要注意編譯是執行./config，並非./configure：
  - ./config
  - make
  - make test
  - make install
- 安裝完畢之後，將0.9.7e版的openssl主程式複製到/usr/bin目錄下覆蓋原先較新的版本，複製前請先備份：
  - cp /usr/bin/openssl /usr/bin/openssl-0.9.7f
  - cp /usr/local/ssl/bin/openssl /usr/bin

建立CA

- 執行OpenSSL內附的CA.sh批次檔加上-newca參數，以建立新的CA，並產生CA的私鑰與憑證，指令如下：
  - /usr/local/ssl/misc/CA.sh –newca
- 中間過程需要輸入私鑰的通行碼（Pass Phrase）以及一些使用者基本資訊

OpenSSL的CA建立過程

```
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:
-----
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.
-----
Country Name (2 letter code) [AU]:TW
State or Province Name (full name) [Some-State]:Taiwan
Locality Name (eg, city) []:Kaohsiung
Organization Name (eg, company) [Internet Widgits Pty Ltd]:NKNU
Organizational Unit Name (eg, section) []:ICE
Common Name (eg, YOUR name) []:Chung-Huang Yang
Email Address []:chyang@nknucc.nknu.edu.tw
```

修改OpenSSL的設定檔openssl.cnf

- openssl.cnf位於/usr/local/ssl目錄下，以vi編輯器開啓，找到下列屬性並修改如下：
  - [CA_default]
  - dir =/usr/local/ssl/misc/demoCA
  - [req]
  - default_bits =2048
  - default_keyfile =/usr/local/ssl/misc/demoCA/private/cakey.pem
  - [tsa_config1]
  - dir =/usr/local/ssl/misc/demoCA

建立TSA Server的root私鑰及憑證

- 指令如下：
  - openssl req -x509 -newkey rsa -out /usr/local/ssl/misc/demoCA/cacert.pem -outform pem
- 需要輸入私鑰通行碼及使用者資訊

產生憑證申請

- 爲TSA Server私鑰向CA申請憑證，指令如下：
  - openssl req -new -key /usr/local/ssl/misc/demoCA/private/tsakey.pem -out /usr/local/ssl/misc/demoCA/tsareq.pem

爲TSA Server私鑰產生憑證申請的過程

```
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.
-----
Country Name (2 letter code) [AU]:TW
State or Province Name (full name) [Some-State]:Taiwan
Locality Name (eg, city) []:Kaohsiung
Organization Name (eg, company) [Internet Widgits Pty Ltd]:NKNU
Organizational Unit Name (eg, section) []:ICE
Common Name (eg, YOUR name) []:Chung-Huang Yang
Email Address []:chyang@nknucc.nknu.edu.tw
Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:
```

簽發TSA Server的憑證

- 簽發憑證的指令如下：
  - openssl ca -in /usr/local/ssl/misc/demoCA/tsareq.pem
- 執行後，會先在畫面列出憑證申請的相關資訊，確認無誤後即可簽發TSA Server憑證
- 簽發之後，會將憑證的所有資訊，包括公開金鑰，全部列示在畫面上

TSA Server憑證簽發確認訊息(部分)

```
Using configuration from /usr/local/ssl/openssl.cnf
Enter pass phrase for
/usr/local/ssl/misc/demoCA/private/cakey.pem:
Check that the request matches the signature
Signature ok
Certificate Details:
        Serial Number: 1 (0x1)
        Validity
            Not Before: Aug 14 01:12:15 2005 GMT
            Not After : Aug 14 01:12:15 2006 GMT
        Subject:
            countryName               = TW
            stateOrProvinceName       = Taiwan
            organizationName          = NKNU
            organizationalUnitName    = ICE
            commonName                = Chung-Huang Yang
            emailAddress              = chyang@nknucc.nknu.edu.tw
        X509v3 extensions:
            X509v3 Basic Constraints:
                CA:FALSE
            Netscape Comment:
                OpenSSL Generated Certificate
            X509v3 Subject Key Identifier:
               5D:24:49:3C:44:16:9D:97:0A:9B:E1:FC:AC:54:84:78:44:A5:DE:04
```

TSA Server憑證存檔

- 憑證存檔之後，接著將其複製到demoCA的目錄下：
  - cp /usr/local/ssl/misc/demoCA/newcerts/01.pem /usr/local/ssl/misc/demoCA/tsacert.pem

安裝Apache時戳服務模組

- 解壓縮所有要用到的套件
- 在root家目錄下解壓縮以下套件：
  - cd ~
  - tar xvfz apache_1.3.33.tar.gz
  - tar xvfz mod_ssl-2.8.22-1.3.33.tar.gz
  - tar xvfz mod_tsa-20041109.tgz

安裝Apache及SSL模組

- 指令依序如下：
  - cd mod_ssl-2.8.22-1.3.33
  - ./configure --with-apache=../apache_1.3.33 --with-eapi-only
  - cd ../apache_1.3.33
  - ./configure --enable-module=so --enable-rule=EAPI
  - make
  - make install
- Apache安裝完畢之後，請將/usr/local/apache/bin/目錄下的apxs執行檔複製到/usr/local/bin目錄：
  - cp /usr/local/apache/bin/apxs /usr/local/bin

安裝mod_tsa

- 安裝指令如下：
  - make OPENSSL=/usr/local/ssl
  - make install
- 如果要結合使用MySQL或Firebird資料庫，務必將TS_MYSQL或TS_FBSQL的值設爲1

設定httpd.conf

- 安裝mod_tsa模組之後，接著必須修改Apache的設定檔httpd.conf，以促使Apache啓動時載入。
- 開啓httpd.conf檔編修，指令如下：
  - vi /usr/local/apache/conf/httpd.conf
- 增修以下設定內容：
  - # 新增請求服務的埠號8080
  - Listen 140.127.40.46:8080
  - # 設定埠號8080供TSA Server使用
  - Port 8080
  - # 設定HTTP Server的名稱
  - ServerName pki.nknu.edu.tw
  - # 設定包含tsa.conf組態檔
  - Include /usr/local/apache/tsa.conf
  - # 設定TSA Server的虛擬主機位址
  - NameVirtualHost 140.127.40.46:8080/tsa
  - <VirtualHost 140.127.40.46:8080/tsa>
  - </VirtualHost>

設定tsa.conf

- 由於在httpd.conf中有設定要包含tsa.conf組態檔，因此請將mod_tsa目錄下的tsa.conf複製到/usr/local/apache/目錄，並修改相關的設定如下：
  - # 設定TSA序號檔
    TSASerialFile conf/tsaserial
  - # 設定TSA憑證檔
  - TSACertificate /usr/local/ssl/misc/demoCA/tsacert.pem
  - # 設定憑證存放路徑
  - TSACertificateChain /usr/local/ssl/misc/demoCA/certs
  - # 設定TSA私鑰檔
  - TSAKey /usr/local/ssl/misc/demoCA/private/tsakey.pem

啓動Apache執行TSA服務

- 指令如下：
  - /usr/local/apache/bin/apachectl start
- 啓動之後，可以檢視Apache的日誌檔內容，確認TSA服務是否成功啓動或有其他錯誤訊息：
  - more /usr/local/apache/logs/error_log
- 正確啓動的相關訊息如下頁圖示。如果完全不能啓動Apache，而且還顯示無法載入mod_tsa模組的錯誤訊息，則請務必再次確認SELinux功能是否已經關閉，因爲啓動SELinux將會使得Apache無法載入mod_tsa模組執行OpenTSA Server

Apache啓動訊息（與TSA服務有關的部分）


```
[Sun Aug 14 10:39:48 2005] [notice] mod_tsa:re-initialization started
[Sun Aug 14 10:39:48 2005] [notice] mod_tsa:serial file is re-used:
    /usr/local/apache/conf/tsaserial
[Sun Aug 14 10:39:48 2005] [notice] mod_tsa:crypto device is set to: builtin
[Sun Aug 14 10:39:48 2005] [notice] mod_tsa:X.509 signer certificate is loaded:
/usr/local/ssl/misc/demoCA/tsacert.pem
[Sun Aug 14 10:39:48 2005] [notice] mod_tsa:X.509 signer certificate chain is loaded:
/usr/local/ssl/misc/demoCA/certs
[Sun Aug 14 10:39:48 2005] [notice] mod_tsa:signer key is initialised:
/usr/local/ssl/misc/demoCA/private/tsakey.pem
[Sun Aug 14 10:39:48 2005] [notice] mod_tsa:default policy is set to: 1.1.2
[Sun Aug 14 10:39:48 2005] [notice] mod_tsa:acceptable policy set includes: 1.1.3
[Sun Aug 14 10:39:48 2005] [notice] mod_tsa:acceptable policy set includes: 1.1.4
[Sun Aug 14 10:39:48 2005] [notice] mod_tsa:acceptable message digest set includes: sha1
[Sun Aug 14 10:39:48 2005] [notice] mod_tsa:acceptable message digest set includes: md5
[Sun Aug 14 10:39:48 2005] [notice] mod_tsa:accuracy is set to: 60 secs, 0 millisecs, 0
    microsecs
[Sun Aug 14 10:39:48 2005] [notice] mod_tsa:clock precision is set to: 0
[Sun Aug 14 10:39:48 2005] [notice] mod_tsa:ordering is switched On
[Sun Aug 14 10:39:48 2005] [notice] mod_tsa:TSA name inclusion is switched On
[Sun Aug 14 10:39:48 2005] [notice] mod_tsa:ESS certificate id chain inclusion is switched On
[Sun Aug 14 10:39:48 2005] [notice] mod_tsa:database driver is set to: None
[Sun Aug 14 10:39:48 2005] [notice] mod_tsa:module has started successfully
[Sun Aug 14 10:39:48 2005] [notice] Apache/1.3.33 (Unix) configured -- resuming normal
    operations
[Sun Aug 14 10:39:48 2005] [notice] Accept mutex:sysvsem (Default: sysvsem)
```

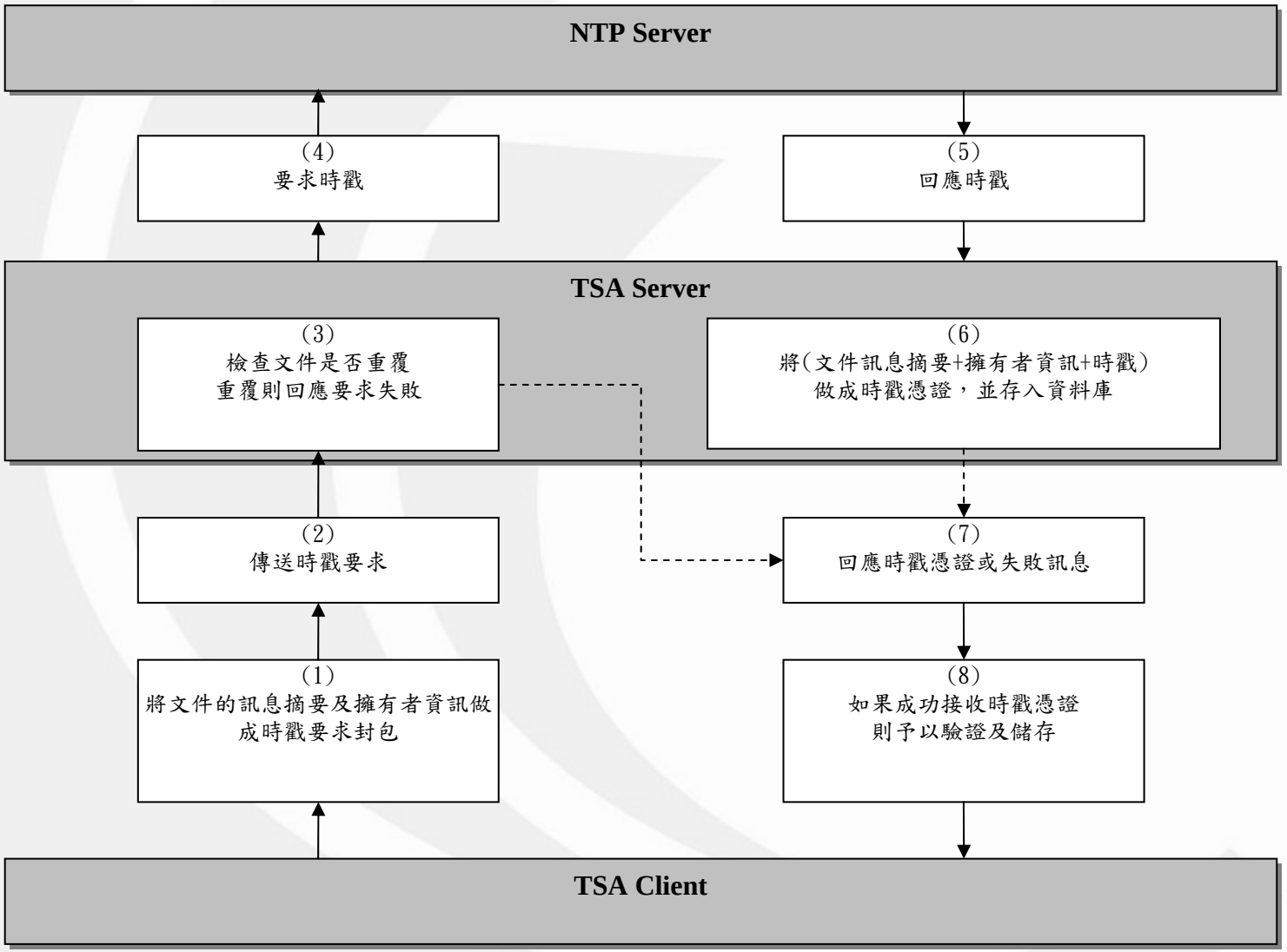
時戳服務應用簡介

- 時戳服務是在通訊雙方之間扮演著「公正第三方」的角色
- 主要作用是提供「某份文件在某一特定時間點之前已經存在」的證明，稱之爲時戳憑證
- 通常應用於「不可否認性」的安全服務
- 將之應用於網際網路方面，只要是具有時限性的服務，均得以時戳憑證作爲其有效性的佐證
- 例如：電子金融交易、電子票券、電子智慧財產權及專利權、電子法律文件或存證服務...等

時戳服務應用簡介(續)

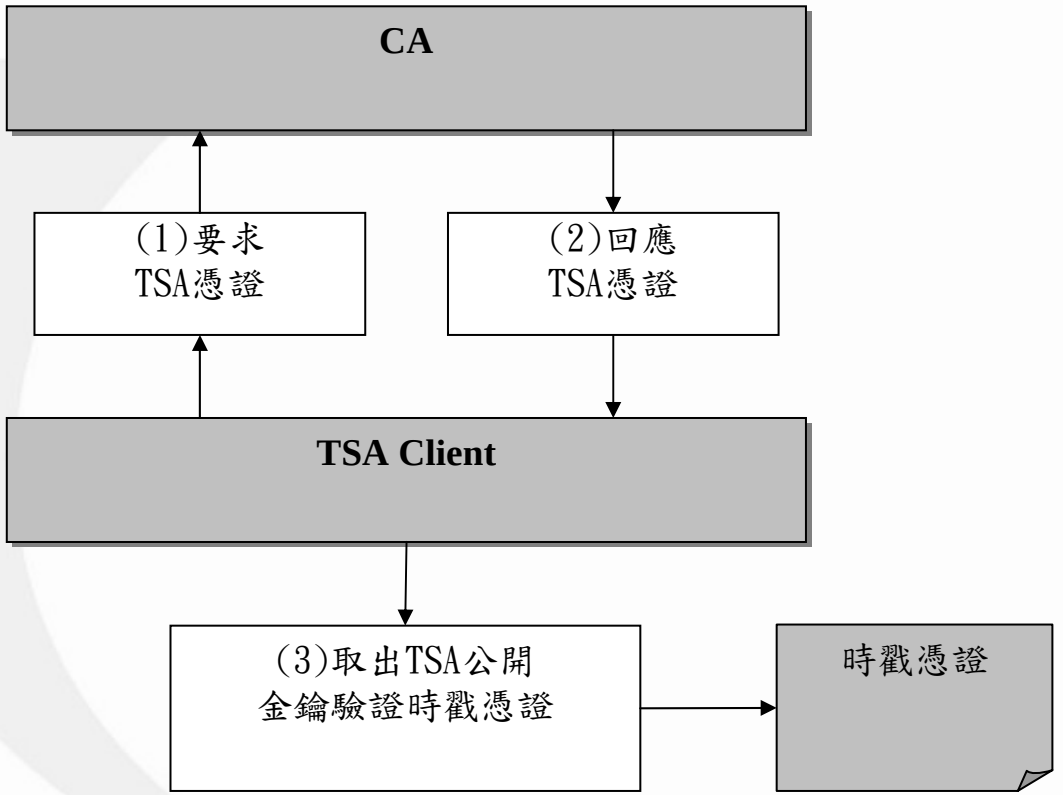
- 當使用者認爲需要爲文件申請時戳憑證時，首先需將文件以雜湊函數演算法（如MD5、SHA-1）計算出訊息摘要，再連同文件擁有者的相關資訊，傳送給時戳服務中心（TSA Server）要求申請簽發時戳憑證
- TSA Server在收到時戳要求後，即向網路時間服務（Network Time Protocol, NTP）伺服器取得具有公信力的時間作爲時戳，之後將擁有者資訊及其文件，連同時戳以TSA Server本身的私鑰做成包含時戳憑證的電子簽章，並予以回應要求時戳的使用者。
- 由於時戳服務是要證明某份文件在某一特定時間點之前已經存在，因此如果重覆要求的話，TSA Server就會回應要求失敗的訊息給使用者。

時戳憑證要求及簽發的流程



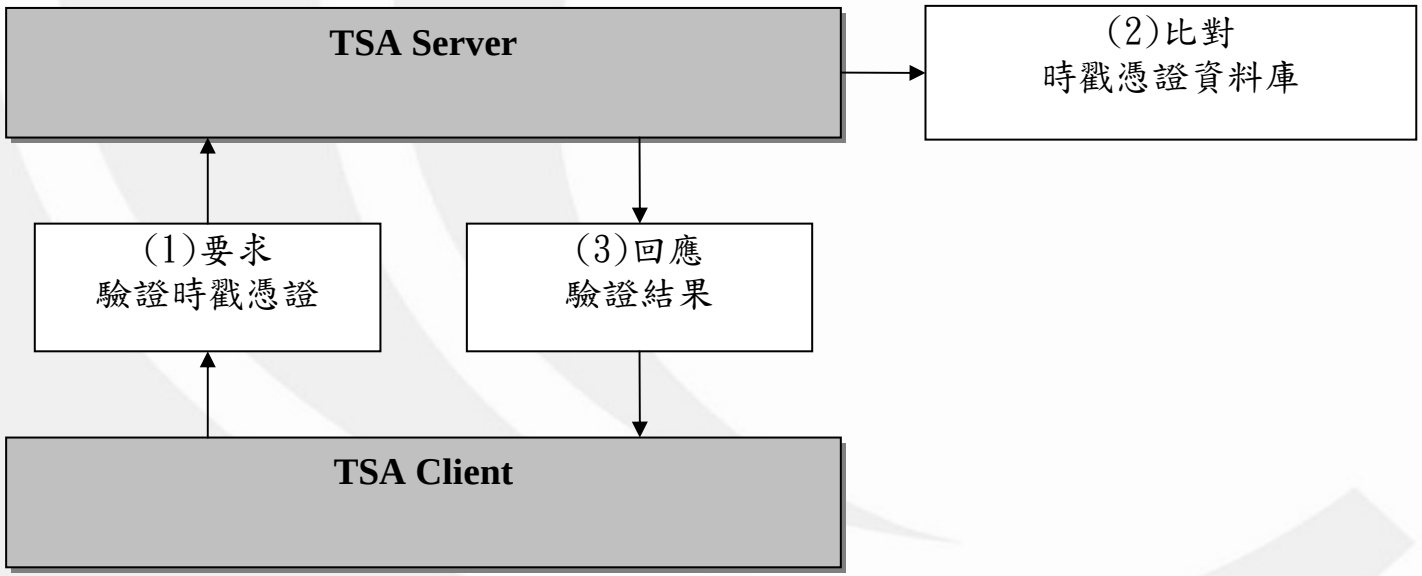
以TSA公開金鑰驗證時戳憑證

- 使用者在接收到TSA Server回應的時戳憑證之後，通常需要驗證，驗證主要有兩種方式：
- 一是透過CA取得TSA Server憑證，以憑證中的公開金鑰驗證時戳憑證是否確爲使用者要求時戳服務的TSA Server所簽發，如右圖



向TSA Server要求驗證時戳憑證

- 另一種方式是直接將時戳憑證再傳回給TSA Server要求進行驗證，TSA Server接收到使用者的驗證要求後，即比對其資料庫，並將驗證結果回傳，如下圖。使用者以此進一步確認時戳憑證是否為該TSA Server所簽發



ts指令的使用

- ts指令是於安裝OpenSSL時戳服務修補套件成功後，附加於OpenSSL的指令。它是OpenTSA主要的工具，同時具備Client/Server兩端所需的功能
- 可用於結合公開金鑰基礎建設（Public Key Infrastructure, PKI）系統建構時戳服務。ts指令包含以下三種參數：
  - **-query**：在TSA Client端產生包含文件訊息摘要及擁有者資訊等內容的時戳要求封包，並將時戳要求傳送給TSA Server
  - **-reply**：TSA Server接收到時戳要求，並確認該文件尚未申請時戳之後，產生含有時戳憑證內容的時戳回應封包，將之回傳給TSA Client
  - **-verify**：TSA Client收到時戳回應或時戳憑證後即予以驗證

-query參數的選項

選項	說明
-rand file:file...	設定亂數產生器要用的亂數資料檔案，可以指定多個檔案以":"分隔。
-config configfile	設定要用的OpenSSL組態檔，如果未設定的話，將使用OPENSSL_CONF環境變數所設定的檔案。
-data file_to_hash	設定要產生時戳要求的文件檔案或資料，如果沒有指定，則會要求以標準輸入裝置（如鍵盤）輸入資料。
-digest digest_bytes	直接設定一組訊息摘要產生時戳要求，設定的方式必須是以16進位碼格式表示，中間可以用":"分隔，如1A:F6:01:... 或1AF601…。
-sha1	選擇要對文件計算訊息摘要所用的雜湊函數演算法，預設是SHA-1。其他設定還有-md2、-md4、-md5及-ripemd160…等。
-policy object_id	以OID設定希望TSA Server產生時戳回應封包或時戳憑證所依據的策略。
-no_nonce	設定傳送時戳要求時不要使用臨時亂數。不過建議別選用此選項，使用臨時亂數有助於防制重送攻擊。
-cert	設定TSA Server回應時能包含它所用以簽署的憑證。
-in request.tsq	指定要顯示可讀內容的時戳要求檔。
-out request.tsq	設定執行時戳要求後，儲存的檔案路徑及名稱。
-text	將DER格式的時戳要求檔設定輸出為常人可讀的純文字格式。

-query的使用範例

- 將testfile以SHA-1雜湊函數演算法計算其檔案的訊息摘要，並產生時戳要求檔儲存成testfile.tsq。同時設定-cert選項，以便將來收到時戳回應時會包含TSA Server簽署用的憑證資訊。指令如下：
  - openssl ts -query -data testfile -sha1 -cert -out testfile.tsq
- 顯示時戳要求檔testfile.tsq內容，包含版本、檔案使用的雜湊函數演算法、檔案的訊息摘要、產生時戳回應策略代碼、臨時亂數、是否要求憑證…等資訊。指令如下：
  - openssl ts -query -in testfile.tsq –text

顯示時戳要求檔的資訊

```
Version: 1
Hash Algorithm: sha1
Message data:
    0000 - fc ff 57 d1 12 69 2c 8f-1c 3d 33 07 14 35 8c 59
..W..i,..=3..5.Y
    0010 - f1 c2 68 cc                                      ..h.
Policy OID: unspecified
Nonce: 0x545518748AD59BB6
Certificate required: yes
Extensions:
```

-reply參數的選項

選項	說明
-config configfile	設定要用的OpenSSL組態檔，如果未設定的話，將使用OPENSSL_CONF環境變數所設定的檔案。
-section tsa_section	設定產生時戳回應時，所要使用的相關組態檔設定區段名稱。
-queryfile request.tsq	指定據以產生時戳回應的時戳要求檔。
-passin password_src	設定TSA私鑰的密碼。
-signer tsa_cert.pem	指定簽署時戳憑證所使用的TSA憑證檔。
-inkey private.pem	指定簽署時戳憑證所使用的TSA私鑰檔。
-chain certs_file.pem	指定憑證鏈結檔。
-policy object_id	以OID設定產生時戳憑證所依據的策略。如果TSA Client傳來的時戳要求封包中已有設定，以TSA Client的為主。
-in response.tsr	指定要顯示可讀內容的時戳回應檔（或時戳憑證）。
-token_in	搭配-in選項使用，指定僅顯示時戳回應檔內的時戳憑證內容，而不顯示回應狀態資訊。
-out response.tsr	設定產生時戳回應後，儲存的檔案路徑及名稱。
-token_out	指定輸出的檔案僅包含時戳憑證，不含回應狀態資訊。
-text	將DER格式的時戳回應檔設定輸出為常人可讀的純文字格式。

-reply的使用範例

- 指定使用的OpenSSL組態檔，依據該檔的設定以時戳要求檔testfile.tsq向TSA Server要求回應，並將TSA Server回傳的時戳回應封包儲成testfile.tsr檔。指令如下：
  - openssl ts -reply -config /usr/local/ssl/openssl.cnf -queryfile testfile.tsq -out testfile.tsr
- 顯示時戳回應檔testfile.tsr的內容，包含回應的狀態訊息、版本、回應的策略代碼、檔案使用的雜湊函數演算法及訊息摘要、時戳、臨時亂數、目錄資訊…等資訊。指令如下：
  - openssl ts -reply -in testfile.tsr –text

顯示時戳回應檔的資訊

```
Status info:
Status: Granted.
Status description: unspecified
Failure info: unspecified
TST info:
Version: 1
Policy OID: 1.2.3.4.1
Hash Algorithm: sha1
Message data:
    0000 - fc ff 57 d1 12 69 2c 8f-1c 3d 33 07 14 35 8c 59
..W..i,..=3..5.Y
    0010 - f1 c2 68 cc                                      ..h.
Serial number: 0x02
Time stamp: Aug 17 01:31:31 2005 GMT
Accuracy: 0x01 seconds, 0x01F4 millis, 0x64 micros
Ordering: yes
Nonce: 0x545518748AD59BB6
TSA: DirName:/C=TW/ST=Taiwan/O=NKNU/OU=ICE/CN=Chung-Huang
Yang/emailAddress=chyang@nknucc.nknu.edu.tw
Extensions:
```

-reply的使用範例(續)

- 將時戳要求檔testfile.tsq以TSA Server本身的憑證（公開金鑰）及私鑰做成包含時戳憑證的電子簽章，並將其儲存成時戳回應檔testfile.tsr。指令如下：
  - openssl ts -reply -config /usr/local/ssl/openssl.cnf -queryfile testfile.tsq -inkey tsakey.pem -signer tsacert.pem -out testfile.tsr
- 請注意TSA的憑證檔tsacert.pem及私鑰檔tsakey.pem的路徑，如果依照先前於16.2.1節介紹安裝時所設的路徑，則此兩檔的完整路徑如下：
  - /usr/local/ssl/misc/demoCA/private/tsakey.pem
  - /usr/local/ssl/misc/demoCA/tsacert.pem
- 產生時戳回應時僅建立不含回應狀態資訊的時戳憑證，並將其內容儲成DER格式的時戳憑證檔testfile_token.der，指令如下：
  - openssl ts -reply -config /usr/local/ssl/openssl.cnf -queryfile testfile.tsq -out testfile_token.der -token_out

-verify參數的選項

選項	說明
-data file_to_hash	以檔案驗證時戳回應封包或時戳憑證。此選項不可以和-digest及-queryfile選項同時使用。
-digest digest_bytes	以檔案的訊息摘要驗證時戳回應封包或時戳憑證。此選項不可以和-data及-queryfile選項同時使用。
-queryfile request.tsq	以原來的時戳要求檔驗證時戳回應封包或時戳憑證。此選項不可以和-data及-digest選項同時使用。
-in response.tsr	要驗證的時戳回應檔，這個選項必須設定。
-token_in	設定只要針對時戳憑證內容驗證。
-CApath trusted_cert_path	設定Client端已信任之CA憑證的存取路徑。
-CAfile trusted_certs.pem	設定一個含有一組已信任之自我簽署CA憑證的檔案。
-untrusted cert_file.pem	設定一個含有一組尚未信任之CA憑證的檔案。這個檔案必須包含TSA簽署的憑證及所有中間CA的憑證，但如果時戳回應封包已經含括這些內容則不在此限。

-verify的使用範例

- 指令如下：
  - openssl ts -verify -data testfile -in testfile.tsr -CAfile cacert.pem -untrusted tsacert.pem
- 上述指令是以CA憑證及TSA憑證驗證時戳回應檔testfile.tsr是否確實由所要求的TSA Server發出，請注意兩個憑證檔cacert.pem及tsacert.pem的設定。如果是於本機進行測試的話，完整的路徑應該如下：
  - /usr/local/ssl/misc/demoCA/cacert.pem
  - /usr/local/ssl/misc/demoCA/tsacert.pem

tsget的使用

- tsget指令是獨立執行的程式，執行的時候，指令前面不需要先加上openssl。在安裝完OpenSSL的時戳修補套件之後，tsget會產生在OpenSSL程式安裝目錄下的misc目錄（/usr/local/ssl/misc）。
- 有別於ts指令，tsget僅具有client端的單純功能，而且它只能傳送時戳要求及接收時戳回應。它既不能產生時戳要求封包，也無法驗證時戳回應封包或時戳憑證的內容
- 然而tsget仍有使用的需要，在前面曾經介紹於Apache Server中安裝並載入mod_tsa模組，以便使得Apache可以啓動TSA服務。
- 因此tsget就可以藉由HTTP或HTTPS協定向提供TSA服務的Apache Server傳送時戳要求封包，而且可以一次傳送兩個以上的時戳要求，同時它亦能接收來自Apache Server所回傳的時戳回應封包

tsget指令的選項

選項	說明
-h server_url	要傳送時戳要求的目標TSA Server，如：http://info.szikszi.hu:8080/tsa或 https://info.szikszi.hu:8443/tsa。
-e extension	設定接收的時戳回應檔之儲存副檔名，預設是.tsr。
-o output	指定接收的時戳回應檔要儲存的檔名，只有在一次傳送單一時戳要求時有效。如果傳送多個時戳要求，將會以原始檔名加上-e選項設定的副檔名儲存。
-v	顯示目前處理的時戳要求名稱。
-d	切換使用libcurl函式庫連線時的顯示資訊為囉唆模式，這意味會呈現較詳細且有助偵錯的訊息。
-k private_key.pem	欲連線HTTPS Server時，Client端需要指定本身使用的私鑰檔。
-p key_password	欲連線HTTPS Server時，設定Client端使用的私鑰檔之存取通行碼。
-c client_cert.pem	欲連線HTTPS Server時，Client端需要指定本身使用的憑證檔。
-C CA_certs.pem	指定已信任之CA憑證檔，欲連線HTTPS Server時，-C或-P選項必須選一設定。
-P CA_path	指定已信任之CA憑證檔的儲存路徑，欲連線HTTPS Server時，-C或-P選項必須選一設定。
-rand file:file...	設定亂數產生器要用的亂數資料檔案，可以指定多個檔案以":"分隔。
-g EGD_socket	設定用以取得亂數資料的EGD socket名稱。
[request]...	指定時戳要求檔，可同時列出多個。如果沒有指定任何檔案，則需要於鍵盤輸入資料，以做成時戳要求封包。

tsget的使用範例

- 向URL位址爲http://140.127.40.46:8080/tsa的TSA Server要求時戳回應，指令如下：
  - tsget -h http://140.127.40.46:8080/tsa testfile.tsq
- 由於tsget無法建立時戳要求檔（如testfile.tsq），因此在使用前仍必須以openssl的ts指令產生時戳要求。
- 向TSA Server送出兩個時戳要求檔，分別是testfile1.tsq及testfile2.tsq，以要求時戳回應。
- 執行時顯示目前處理的時戳要求檔名稱，及詳細的連線與處理過程資訊，並將TSA Server回傳的時戳回應封包儲存爲副檔名爲.reply的檔案。指令如下：
  - tsget -h http://140.127.40.46:8080/tsa -v -d -e .reply testfile1.tsq testfile2.tsq
- 執行後會產生testfile1.reply及testfile2.reply兩個時戳回應檔

以tsget傳送兩個時戳要求給TSA Server


```
testfile1.tsq: sending request* About to connect() to 140.127.40.46 port
8080
*   Trying 140.127.40.46... * connected
* Connected to 140.127.40.46 (140.127.40.46) port 8080
> POST /tsa HTTP/1.1
User-Agent: OpenTSA tsget.pl/1.1.14.1
Host: 140.127.40.46:8080
Pragma: no-cache
Content-Type: application/timestamp-query
Accept: application/timestamp-reply
Content-Length: 52
Expect: 100-continue
< HTTP/1.1 100 Continue
< HTTP/1.1 200 OK
< Date: Wed, 17 Aug 2005 02:53:14 GMT
< Server: Apache/1.3.33 (Unix)
< Content-Length: 1873
< Content-Type: application/timestamp-reply
* Connection #0 to host 140.127.40.46 left intact
, reply received, ./testfile1.reply written.
testfile2.tsq: sending request* Re-using existing connection! (#0) with
host 140.127.40.46
* Connected to 140.127.40.46 (140.127.40.46) port 8080
> POST /tsa HTTP/1.1
User-Agent: OpenTSA tsget.pl/1.1.14.1
Host: 140.127.40.46:8080
Pragma: no-cache
Content-Type: application/timestamp-query
Accept: application/timestamp-reply
Content-Length: 52
Expect: 100-continue
< HTTP/1.1 100 Continue
< HTTP/1.1 200 OK
< Date: Wed, 17 Aug 2005 02:53:14 GMT
< Server: Apache/1.3.33 (Unix)
< Content-Length: 1873
< Content-Type: application/timestamp-reply
* Connection #0 to host 140.127.40.46 left intact
, reply received, ./testfile2.reply written.
* Closing connection #0
```

參考資料

- John Viega, Matt Messier, Pravir Chandra, *Network Security with OpenSSL*, O'Reilly, June 2002
- OpenTSA, "OpenTSA patch installation", http://www.opentsa.org/ts/ts-install-20041109.html
- OpenTSA, "mod_tsa installation", http://www.opentsa.org/mod_tsa/mod_tsa-install-20041109.html
- OpenTSA, "ts manual", http://www.opentsa.org/ts/ts-20041109.html
- OpenTSA, "tsget manual", http://www.opentsa.org/ts/tsget-20041109.html
- 中華電信研究所, "電子時戳服務介紹", http://210.241.69.194/download/timeStamp_intro_920915.pdf
- 葉志青、楊中皇、褚芳達, "結合IC卡強化時戳服務之設計與實現", http://crypto.nknu.edu.tw/publications/2004IMAD.pdf, 2004年資訊管理應用與發展研討會, 2004年6月