HiSECURE函式介面 2005教育訓練

中華電信研究所
王宏騰

標題

- **網路資訊安全簡介**
- 憑證的內容介紹
- 憑證安全檢查項目
- HiSECURE的架構
- HiSECURE函式的功能介紹
- HiSECURE函式呼叫流程說明
- 專案說明
- 補充說明及Q&A

安全的層面

- 週邊防禦：封包過濾，連線狀態監控，入侵偵測
- 網路防禦：VLAN 存取控制清單，內部防火牆，稽核，入侵偵測
- 主機防禦：強化伺服器，主機入侵偵測，稽核
- 應用程式防禦：驗證連線身分，強化 IIS
- 資料與資源防禦：資料庫，檔案分享，存取控制，稽核

- 資訊安全的服務項目：
  - 隱密性(Confidentiality)→加封加密
  - 資料完整性(Integrity)
  - 來源認證(Authentication)
  - 事後存證(Non-repudiation)→從傳送資料產生一份存證資料→簽章
- 網路世界最佳的實作方法

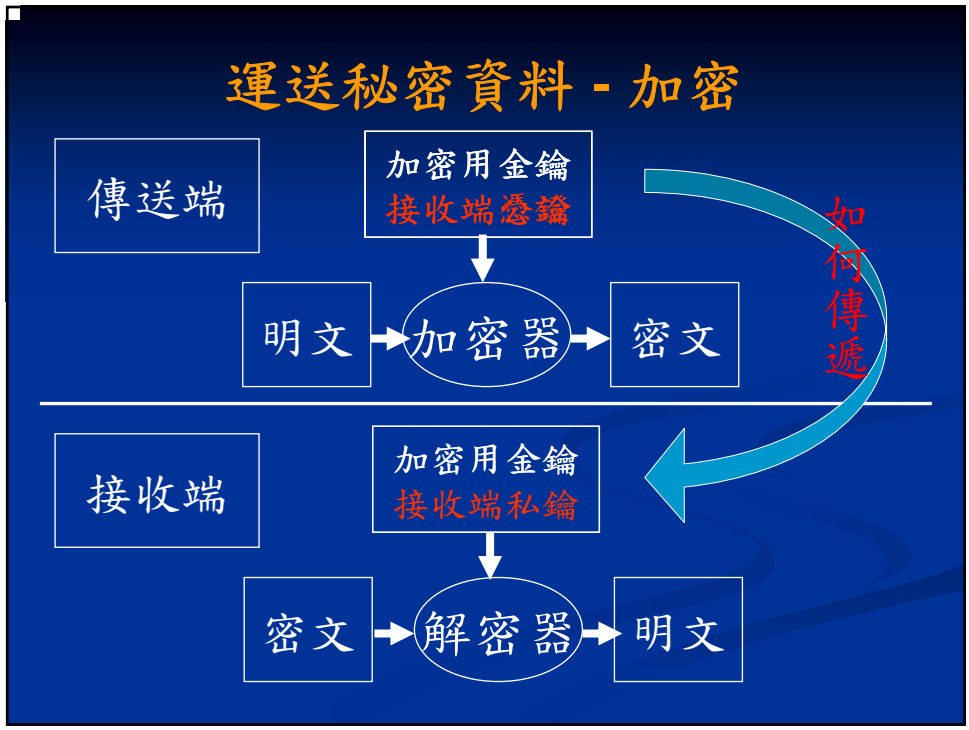
PKI

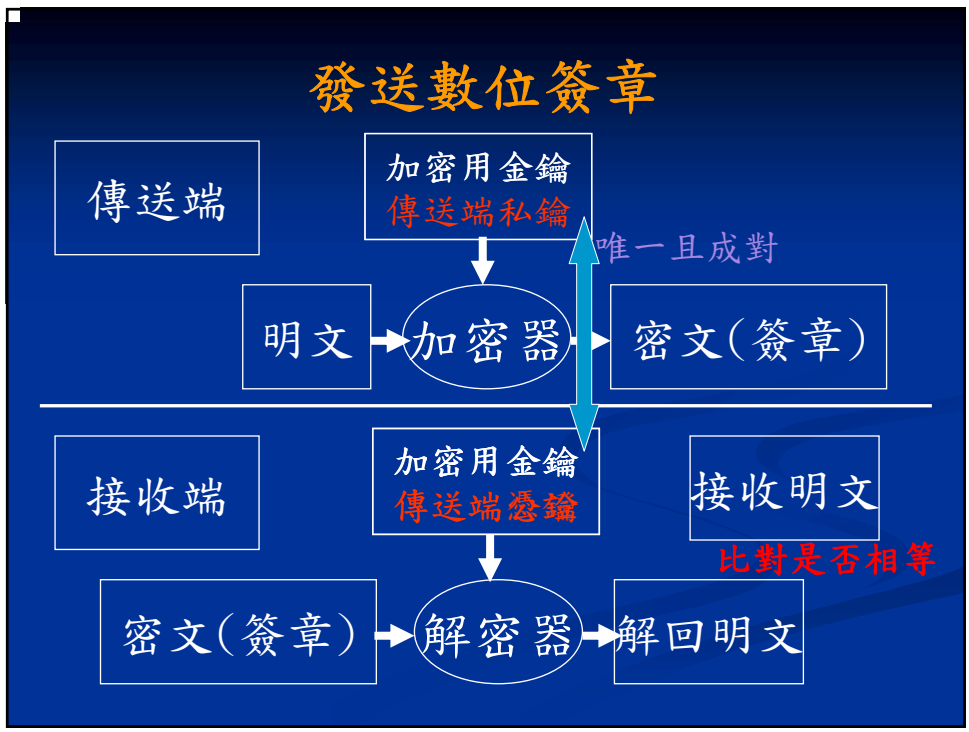
WHY PKI

- PKI的核心技術：
  - 非對稱式加解密演算法(RSA)
    - 資料的加密
      - 保障資料的隱密性
    - 數位簽章的產生
      - 保障資料的完整性，來源認證及資料存證

PKI的核心技術-RSA

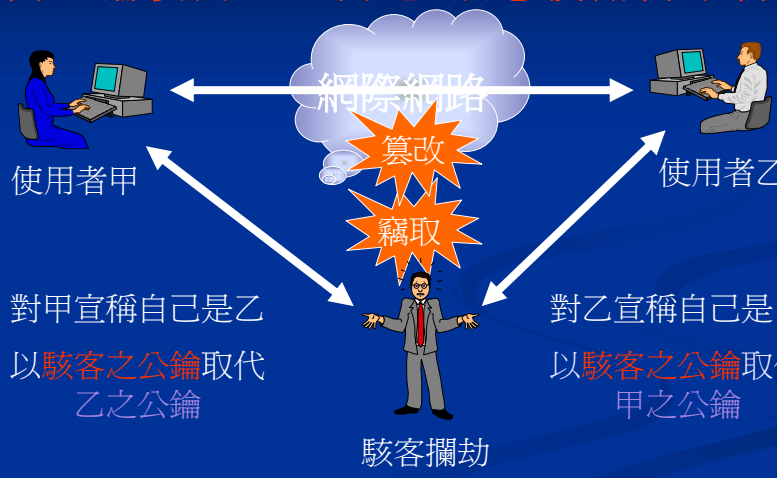
- 非對稱式演算法的特性：
  - 加解密兩端使用的是不同的金鑰
  - 這兩把金鑰是成對且唯一
  - 金鑰的產製以及演算法的運算都需要花費相當多的時間
  - 破解的難度在於金鑰的長度及亂度(目前的建議值為1024個位元)
  - 常用的作法為金鑰的主人將成對的兩把金鑰一把自己保存，另一把則公開給其他人使用(也就是公鑰私鑰名稱的由來)


運送秘密資料 - 加密
傳送端
加密用金鑰
接收端憑鑰
如何傳遞
明文
加密器
密文
接收端
加密用金鑰
接收端私鑰
密文
解密器
明文


發送數位簽章
傳送端
加密用金鑰
傳送端私鑰
唯一且成對
明文
加密器
密文(簽章)
接收端
加密用金鑰
傳送端憑鑰
接收明文
比對是否相等
密文(簽章)
解密器
解回明文

為何要使用憑證

公開金鑰使用上之問題-訊息洩露而不自知



解決方式–將公鑰賦予身份(憑證)

憑證的產生

- 一開始一定要有公正認可的第三者來擔任憑證簽發者的角色
- 最主要的目的在將持有人的公鑰以及其身分相結合
- 使用到非對稱式加解密的簽章技術，憑證的簽發者須對所有的資訊予以簽章，也因如此日後也才可對此憑證加以認證

憑證的驗證

- 為了解決無法驗證的公鑰所產生之公開金鑰使用的問題，將公鑰賦予了辨識的方式成為了憑證。
- 無法驗 vs 沒有驗？
- 憑證最重要的三個檢查項目
  - 憑證內之CA簽章值是否正確
  - 檢查憑證的時間（validity）欄位是否仍在有效期限
  - 檢查憑證是否已被廢止（可藉由查詢憑證廢止清單或是執行OCSP函式）

標題

- 網路資訊安全簡介
- **憑證的內容介紹**
- 憑證安全檢查項目
- HiSECURE的架構
- HiSECURE函式的功能介紹
- HiSECURE函式呼叫流程說明
- 專案說明
- 補充說明及Q&A

憑證的功用

- 憑證的內部由於含有擁有者的相關資訊，因此可以作為身分認證使用。不過由於憑證內部個人的私密資訊較少，因此實作上多會搭配其他的驗證機制
- 憑證的內部由於含有擁有者的公開金鑰，依據非對稱式加解密演算法，他人可以使用你的憑證來將資料加密；你也可以使用別人的憑證來驗證他所送出的數位簽章

憑證結構圖

TbsCertificate資訊架構圖

An X.509 standard certificate contains the following information.

Field	Description
Version	Version number of the certificate.
Serial Number	Serial number of the certificate.
Algorithm Identifier	Signature algorithm used by the certificate signer.
Issuer Name	Name of the issuer of the certificate.
Validity:	
Not Before (Date)	Date before which the certificate is not valid.
Not After (Date)	Date after which the certificate is not valid.
Subject Name	Name of the person or entity to whom the certificate is being issued.
Subject Public Key Info:	
Algorithm	Algorithm used for the public key.
Subject Public Key	Actual public key (a bit string).
Optional Fields:	
Issuer Unique ID	If present, version must be version 2.
Subject Unique ID	If present, version must be version 2.
Extensions	Optional field. Represents additional data that an issuer can want to add to a certificate, such as e-mail address or authorization to issue certificates.
If extensions are present, version must be version 3.	

V3

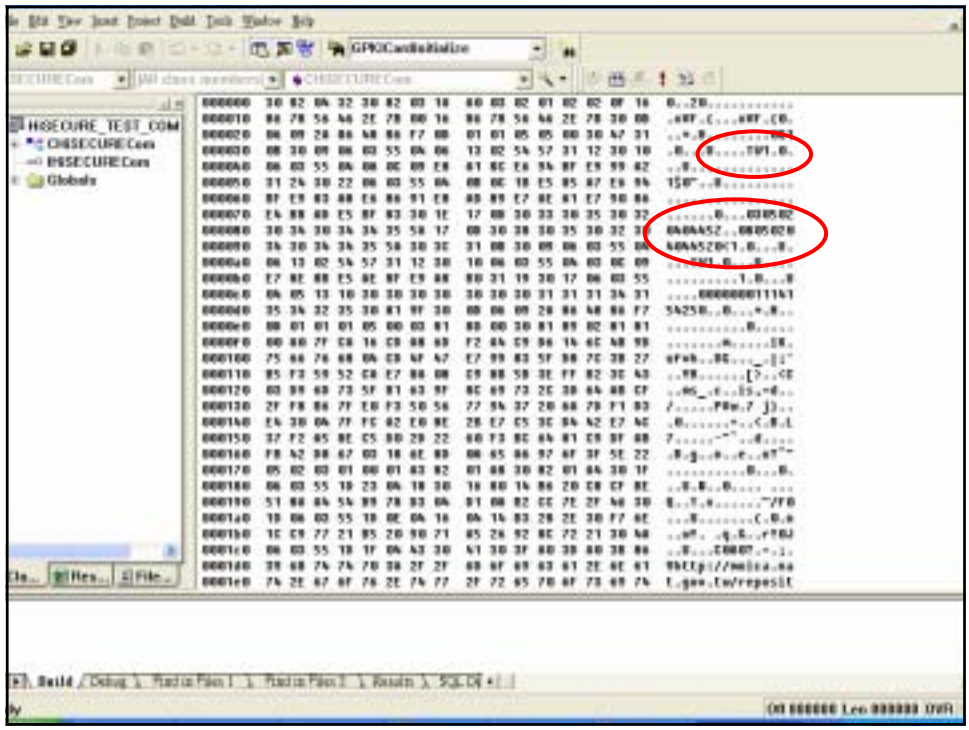
16或17個bytes

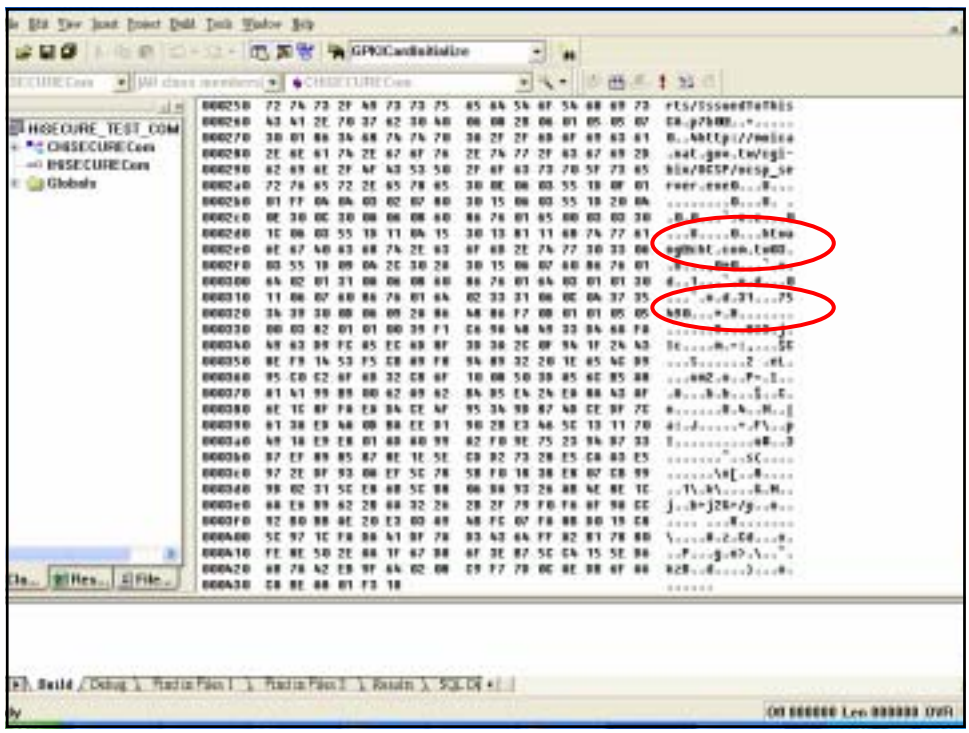
中文以UTF8編碼

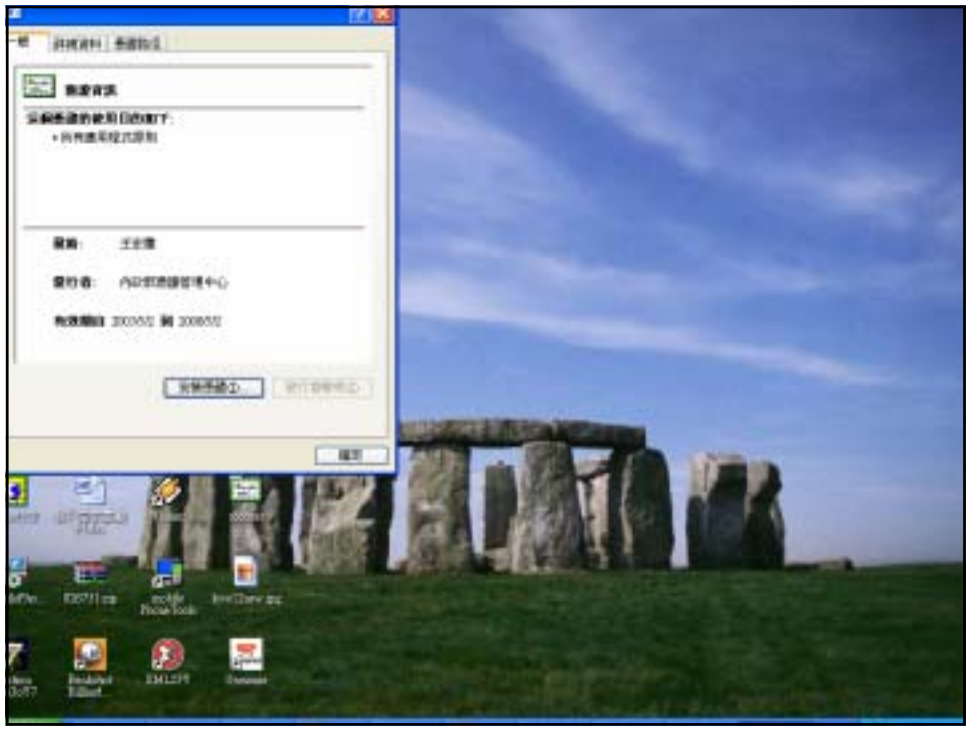
格林威治標準時間

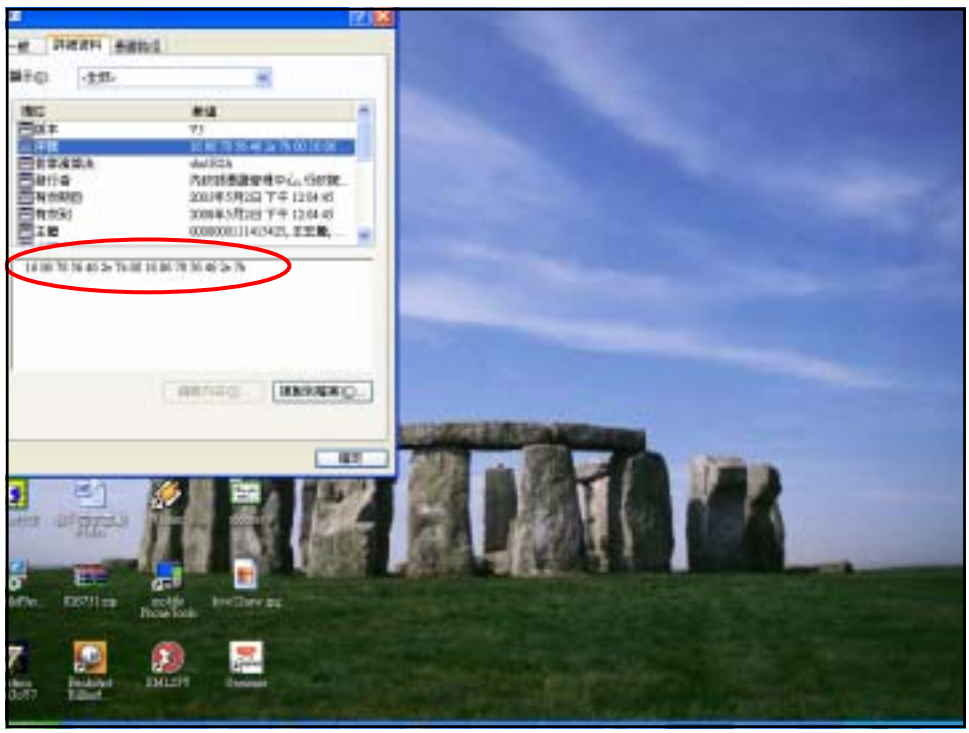
不存在

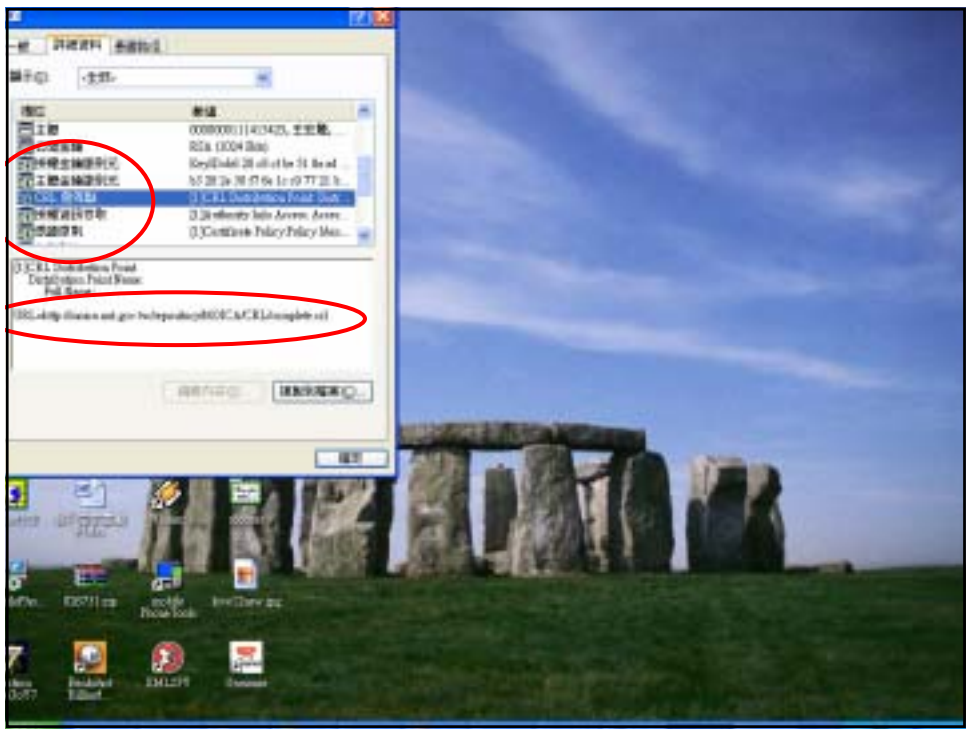
詳見GPKI憑證與憑證廢止清冊格式剖繪檔案

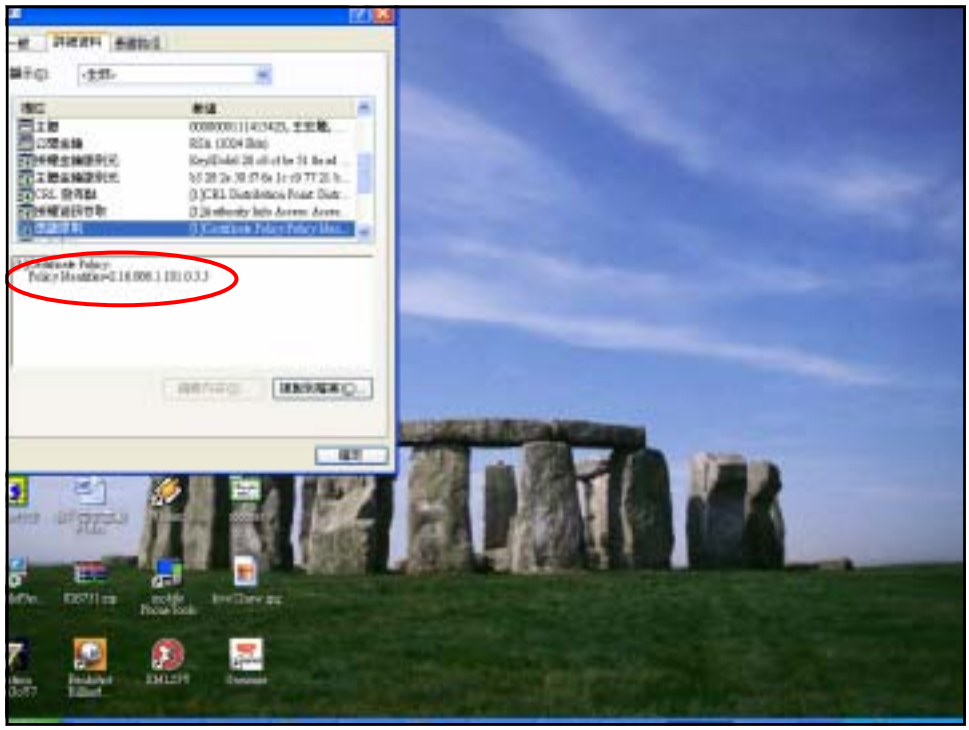


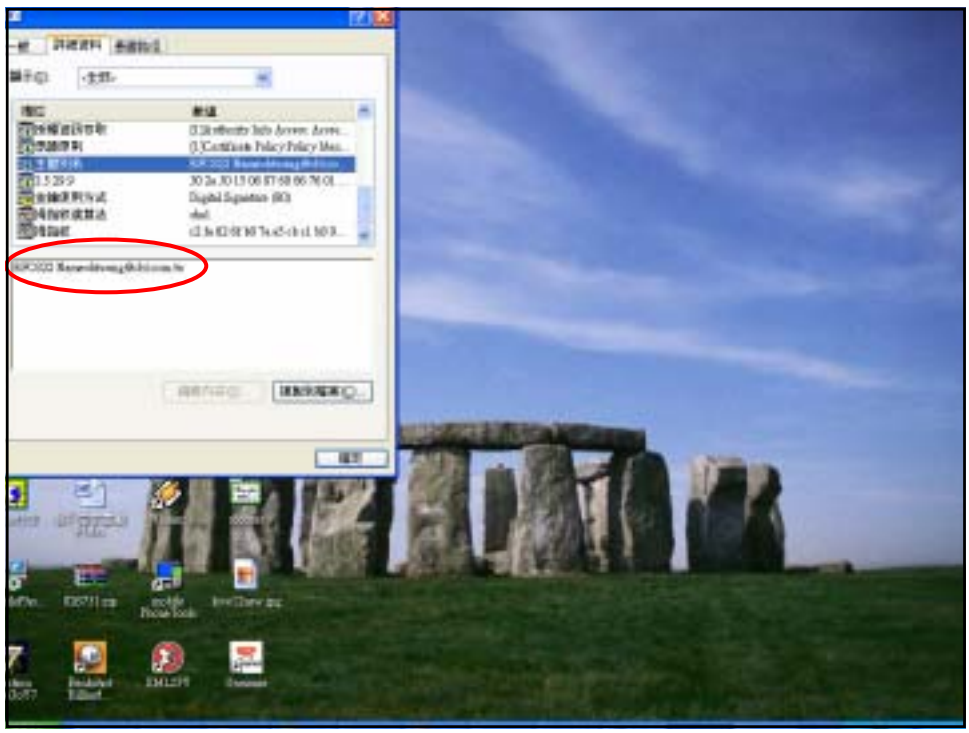


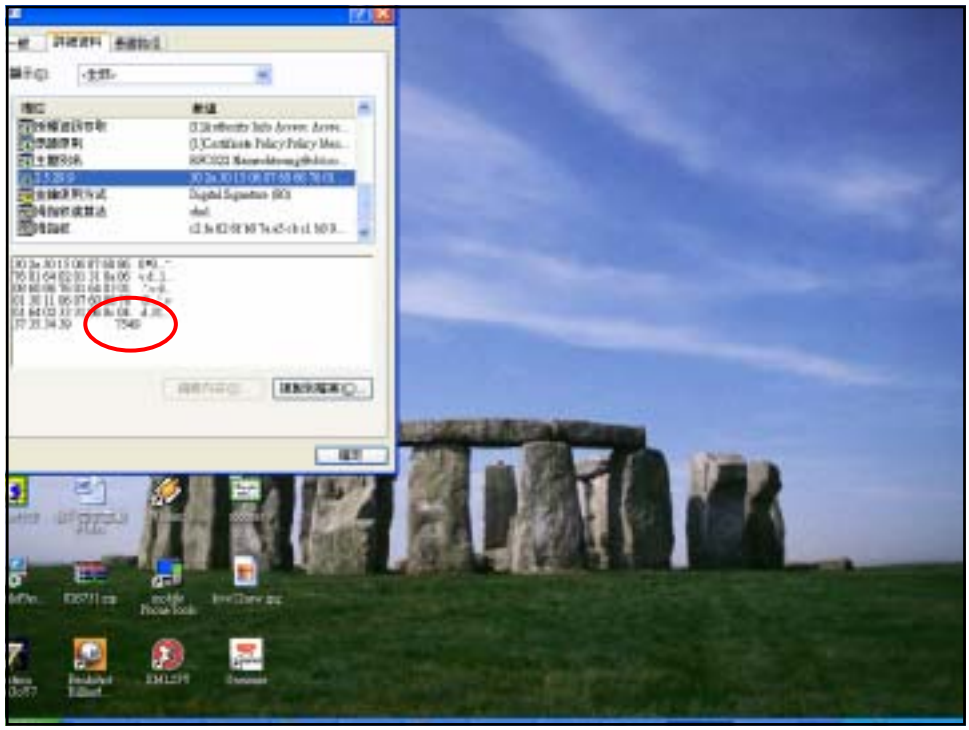












DER CODE編碼原理

TAG + LEN + DATA

TAG：用以表示接下來的資料型態，如字串、整數等等

LEN：資料的長度

DATA：實際上的真正資料

標題

- 網路資訊安全簡介
- 憑證的內容介紹
- **憑證安全檢查項目**
- HiSECURE的架構
- HiSECURE函式的功能介紹
- HiSECURE函式呼叫流程說明
- 專案說明
- 補充說明及Q&A

政府機關公開金鑰基礎建設GPKI

公鑰憑證處理安全檢查表

- 共有十八條，詳見MOICA網站
- 分為可用HiSECURE函式介面解決〔第二到第十五條〕與其他方式解決〔第一條、第十六、十七及十八條〕
- 主要分為兩個階段的檢查，一為CA端憑證的檢驗；另一為用戶端憑證的檢驗。主要的原因為GPKI為三層式的憑證架構，也因為如此第三到第八條的檢查方式同第九到第十四條

公鑰憑證處理安全檢查表

- 安全取得Root CA(即GRCA)憑證(1)
- 系統需設定信賴的憑證保證等級(2)
- 驗證憑證內簽章(3,9)
- 檢驗憑證內之金鑰使用用途是否正確(4,10)
- 檢驗有效期限(5,11)
- 使用CRL來驗證憑證廢止狀態(6,12)
- 檢驗CRL是否為正確且最新的(7,8,13,14)

公鑰憑證處理安全檢查表

- 對傳送之訊息加簽電子簽章以驗證用戶身份(15)
- 系統的設計應以Challenge-Response或是Nonce機制來防範重送(replay)攻擊(16)
- 對用戶私密資料應以強渡128bits以上的安全通道來保護(17)
- 系統應定期校時，以確保系統時間之正確性(18)

- **PKI架構下我們可能的工作**
  - 使用私鑰的部分
    - 解開秘密資料
    - 產生數位簽章
  - 使用公鑰的部分
    - 產生秘密資料
    - 驗證數位簽章
  - 公鑰憑證的驗證與資訊取得
- **HiSECURE相對提供的支援**
  - 資料加解密 – 對稱與非對稱
  - 數位簽章與驗證 – SHA1 with RSA
  - 憑證資訊的取得
  - 憑證廢止驗證 – CRL 及 OCSP

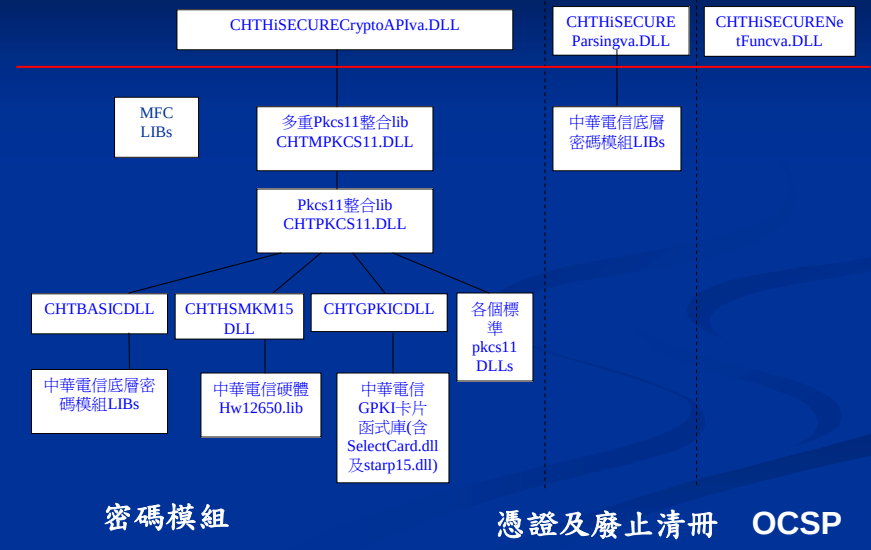
標題

- 網路資訊安全簡介
- 憑證的內容介紹
- 憑證安全檢查項目
- **HiSECURE的架構**
- HiSECURE函式的功能介紹
- HiSECURE函式呼叫流程說明
- 專案說明
- 補充說明及Q&A

HiSECURE架構

- 函式介面分為三大類
  - 密碼模組部分
  - 憑證及憑證廢止清單解析部分
  - 網路相關部分
- 密碼模組部分→以PKCS11為基礎，處理資料加密、簽章的產生及驗證等功能
- 憑證及憑證廢止清單解析部分→取得憑證內部資訊及查詢憑證廢止狀態
- 網路相關部分目前僅提供OCSP函式→提供即時的憑證狀態查詢功能

HiSECURE函式介面整體架構圖



HiSECURE包含(1)

- 標頭檔

```
Chthisecurecryptoapiva.h
Chthisecureparsingva.h
Chthisecurenetfuncva.h
Chthisecurestructva.h
errortable.h
pks11.h pkcs11f.h pkcs11t.h pkcs11x.h
GPKICFunction.h
```

HiSECURE包含(2)

- 動態連結函式庫
  - Chthisecurecryptoapiva.dll
  - Chthisecureparsingva.dll
  - Chthisecurenetfuncva.dll
  - chtmpkcs11.dll chtpkcs11.dll
  - chtbasicdll.dll chtgpkicdll.dll
  - chthsmkm15dll.dll
  - GPKICardFunction.dll
  - bfiveucs.dll ucsbfive.dll

HiSECURE包含(3)

- 靜態連結函式庫
  - Chthisecurecryptoapiva.lib
  - Chthisecureparsingva.lib
  - Chthisecurenetfuncva.lib
  - GPKICardFunction.lib

HiSECURE包含(4)

- 手冊或指引
  - 中華電信HiSECURE函式介面 5.1版.doc
  - 中華電信HiSECURE GPKIC卡片補充函式.doc
  - 中華電信HiSECURE GPKI IC卡密碼模組CHTGPKICDLL操作指引.doc
  - 中華電信HiSECURE 軟體密碼模組CHTBASICDLL操作指引.doc
  - 中華電信HiSECURE 硬體密碼模組CHTHSMKM15DLL操作指引.doc

HiSECURE5.0 → 5.1

- 修正5.0的部份程式BUG
- 對於卡片密碼模組的更新，於程式碼不需更動的情況下，補上新的函式庫即可支援新的卡片
- 對於憑證廢止清冊作業提供新的函式
- 錯誤代碼的整併

標題

- 網路資訊安全簡介
- 憑證的內容介紹
- 憑證安全檢查項目
- HiSECURE的架構
- **HiSECURE函式的功能介紹**
- HiSECURE函式呼叫流程說明
- 專案說明
- 補充說明及Q&A

函式的功能介紹-分類總覽

1. 啟用與關閉相關函式
2. 物件相關函式
3. 雜湊函式
4. 非對稱式加解密相關函式
5. 對稱式加解密相關函式
6. 憑證解析函式
7. 憑證廢止狀況查詢函式
8. OCSP相關函式

啟用與關閉相關函式

- ***InitModule***：啟動密碼模組函式。
- ***CloseModule***：釋放Module所佔用的資源。
- ***InitSession***：Session為新版本安全保密函式庫運作的基本單位，此函式的功能為設定一個Session以提供後續的函式使用。
- ***CloseSession***：釋放Session所佔用的資源。

中華電信軟體密碼模組	GPKI IC卡密碼模組	中華電信保密硬體模組	提供標準PKCS11模組

物件相關函式

- ***GenerateGCAv3PrivateKeyStruct***：產生GCAv3形式的私密金鑰結構。
- ***GetKeyObjectHandle***：產生密碼模組運算時候所需要的金鑰物件。
- ***DeleteKeyObject***：釋放金鑰物件所佔用的資源。

雜湊函式

- ***HashFunction***：根據輸入的雜湊函式演算法對資料產生雜湊值。

  雜湊函式：

  資料 → 雜湊值 （單向運算）

  使用時機：

  搭配非對稱式演算法使用

非對稱式加解密相關函式

- ***MakeSignature***：根據輸入的演算法對資料產生數位簽章函式。
- ***VerifySignature***：根據輸入的演算法來檢驗資料的簽章值是否正確。
- ***PublicKeyEncryption***：根據輸入之演算法執行公鑰加密函式。
- ***PrivateKeyDecryption*** ：根據輸入之演算法執行私鑰解密函式。

對稱式加解密相關函式

- ***SymEncryptionAlg***：根據輸入的演算法執行對稱式加解密演算法加密函式。
- ***SymDecryptionAlg***：根據輸入的演算法執行對稱式加解密演算法解密函式。

憑證解析函式

- ***DecodeCertificate***：將憑證資料轉為基本憑證結構。
- ***CertBasicStructDestructor***：將基本憑證結構內部資料清除。
- ***VerifyCertSignature***：檢驗憑證內之簽章值是否為其CA所簽署。
- ***GetSerialNumber***：得出位元組型態的憑證序號。
- ***GetCertIssuerDN***：得出憑證憑證簽發者(CA)的Distinguished Name (DN)。
- ***GetCertSubjectDN***：得出憑證主體(Subject)的Distinguished Name (DN)。

憑證解析函式(續)

- ***GetLDAPTypeCertIssuerDN***：得出憑證憑證簽發者(CA)的Distinguished Name (DN)。為LDAP型態。
- ***GetLDAPTypeCertSubjectDN***：得出憑證主體(Subject)的Distinguished Name (DN)。為LDAP型態。
- ***GetRDNFromDN***：從DN字串中取出個別的資訊。
- ***GetCertValidity***：得到憑證的啟用及廢止時間〔以Struct TM的方式輸出〕。
- ***GetCertInfo***：取得憑證的一些基本資訊，以憑證資訊結構輸出。

憑證解析函式(續)

- ***GetExtension***：根據輸入之憑證延伸欄位OID取出個別的憑證延伸欄位資料。
- ***GetSubjectDirectoryAttributes***：從SubjectDirectoryAttributes憑證延伸欄位取出內含資料。
- ***ExtractOIDFromDerCode***：從以DERCODE編碼之OID資料取出OID資料。

憑證廢止狀況查詢函式

- ***DecodeCRL***：將憑證廢止清冊資料轉換為基本CRL結構。
- ***CRLBasicStructDestructor***：將基本憑證廢止清單結構內部資料清除。
- ***VerifyCrlSignature***：使用簽署CRL的憑證來檢驗CRL簽章的正確性。
- ***SearchCRL***：從憑證廢止清單〔包含Complete CRL以及Delta CRL〕取得憑證狀態或廢止相關資訊。

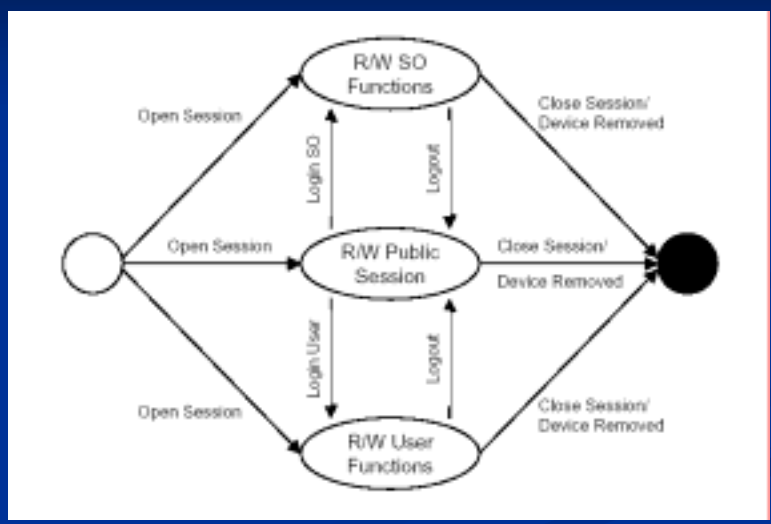
OCSP相關函式

- ***BuildTobesignedOCSPRequest***：產生未含簽章的ToBeSigned OCSPRequest資料。
- ***QueryOCSfromSignedOCSPRequest***：將呼叫BuildTobesignedOCSPRequest 函式所產生之ToBeSigned OCSPRequest以及隨後產生之簽章合併成OCSP Request以執行線上憑證狀態查詢的功能。

標題

- 網路資訊安全簡介
- 憑證的內容介紹
- 憑證安全檢查項目
- HiSECURE的架構
- HiSECURE函式的功能介紹
- **HiSECURE函式呼叫流程說明**
- 專案說明
- 補充說明及Q&A

密碼模組函式介面設計(作法)



函式呼叫流程說明(1)

- 密碼模組部分

  起始密碼模組(InitModule)

  -> 起始運作環境(InitSession)

  -> 取得金鑰控制指標(GetKeyObjectHandle)

  -> 呼叫密碼相關函式

  -> 釋放金鑰資源(DeleteKeyObject)

  -> 結束運作環境(CloseSession)

  -> 結束密碼模組(CloseModule)

呼叫流程說明(2)

- 憑證解析部分

  取得憑證資料

  -> 轉為憑證基本結構

  -> 取出憑證資訊結構(此時已可取出一些基本資料，但不包含延伸欄位內部資料)

  -> 取出個別憑證延伸欄位

  -> 取出所要資料(DERCODE)

呼叫流程說明(3)

- 憑證狀態查詢部分

  取得憑證廢止清單

  -> 轉為憑證廢止清單基本結構

  -> 呼叫憑證狀態查詢函式

  -> 取得憑證狀態

呼叫流程說明(4)

- OCSP部分

  產生OCSP查詢結構

  -> 產生OCSP查詢封包資料

  -> 自行對OCSP查詢封包資料簽章(非必要條件，需視服務提供者的規定)

  -> 執行OCSP查詢動作

標題

- 網路資訊安全簡介
- 憑證的內容介紹
- 憑證安全檢查項目
- HiSECURE的架構
- HiSECURE函式的功能介紹
- HiSECURE函式呼叫流程說明
- **專案說明**
- 補充說明及Q&A

專案說明
網址
http://moica.nat.gov.tw/html/web-data/API.htm

範例的特點

- WWW動態網頁
- 自卡片讀取憑證資料
- 憑證的解析
- 非對稱式的加解密測試
- 數位簽章的產生與驗證

自卡片內取出憑證

```
unsigned char       *Cert;
int                 ret, iCertLen = 0;

ret = GetCertificateFromGPKICard(iCertID, Cert,
&iCertLen, NULL);
if(ret == Buffer_Too_Small)
{
        Cert = new unsigned char[iCertLen];
        ret = GetCertificateFromGPKICard(iCertID, Cert,
        &iCertLen, NULL);
}
```

憑證解析

```
CertInfoStruct        SCertInfo;
CertBasicStruct       cbs;

ret = DecodeCertificate(Cert, iCertLen, cbs);
ret = GetCertInfo(cbs, SCertInfo);
……
unsigned int          sanoid[4] = {2,5,29,17};
bool                  bcritical;
unsigned char         ucsda[128];
int                   isdal = 128;

ret = GetExtension(SCertInfo.ucExtension,
SCertInfo.iExtensionLength, sanoid, 4, &bcritical,
ucsda, &isdal);
```

使用公鑰加密

```
unsigned long         m, s, k;
unsigned char         eCert[2048];
int                   ieCertLen = 2048;
ret = InitModule("chtbasicdll.dll", NULL, &m);
ret = InitSession(m, CKF_SERIAL_SESSION, NULL, 0, &s);
ret = GetKeyObjectHandle(m, s, 1, NULL, 0, (void*) &eCert,
ieCertLen, &k);

unsigned char         pPlain[128], pCipherData[256];
int                   iPlainLength = 128, iCipherDataLength = 256;
ret = PublicKeyEncryption(m, s, CKM_RSA_PKCS, pPlain,
iPlainLength, k, pCipherData,&iCipherDataLength);
ret = DeleteKeyObject(m, s, k);
ret = CloseSession(m, s);
ret = CloseModule(m);
```

使用私鑰解密

```
unsigned long           m, s, k;
char                    sPassWD[8]      = “12345678”;
int                     iPassWDLength= 10;
ret = InitModule("chtgpkicdll.dll", NULL, &m);
ret = InitSession(m, CKF_SERIAL_SESSION,
sPassWD, iPassWDLength, &s);
ret = GetKeyObjectHandle(m, s, 0, NULL, 0, (void*)"2", 1, &k);

unsigned char           pCipher[128], pPlainData[128];
int                     iCipherLength = 128, iPlainDataLength = 128;
ret = PrivateKeyDecryption(m, s, CKM_RSA_PKCS, pCipher,
iCipherLength, k, pPlainData, &iPlainDataLength);
ret = DeleteKeyObject(m, s, k);
ret = CloseSession(m, s);
ret = CloseModule(m);
```

使用私鑰產生數位簽章

```
unsigned long           m, s, k;
char                    sPassWD[8]      = “12345678”;
int                     iPassWDLength= 10;
ret = InitModule("chtgpkicdll.dll", NULL, &m);
ret = InitSession(m, CKF_SERIAL_SESSION,
sPassWD, iPassWDLength, &s);
ret = GetKeyObjectHandle(m, s, 0, NULL, 0, (void*)"2", 1, &k);

unsigned char           pMessage[128], pSignature[256];
int                     iMessageLength = 128, iSignatureLength = 256;
ret = MakeSignature(m, s, CKM_SHA1_RSA_PKCS, pMessage,
iMessageLength, k, pSignature, &iSignatureLength);
ret = DeleteKeyObject(m, s, k);
ret = CloseSession(m, s);
ret = CloseModule(m);
```

使用公鑰驗證數位簽章

```
unsigned long           m, s, k;
unsigned char           vCert[2048;
int                     ivCertLen = 2048;
ret = InitModule("chtbasicdll.dll", NULL, &m);
ret = InitSession(m, CKF_SERIAL_SESSION, NULL, 0, &s);
ret = GetKeyObjectHandle(m, s, 1, NULL, 0, (void*) &vCert,
ivCertLen, &k);

unsigned char           pMessage[128], pSignature[128];
int                     iMessageLength = 128, iSigLength = 128;
ret = VerifySignature(m, s, CKM_SHA1_RSA_PKCS, pMessage,
iMessageLength, k, pSignature, iSigLength);
ret = DeleteKeyObject(m, s, k);
ret = CloseSession(m, s);
ret = CloseModule(m);
```

標題

- 網路資訊安全簡介
- 憑證的內容介紹
- 憑證安全檢查項目
- HiSECURE的架構
- HiSECURE函式的功能介紹
- HiSECURE函式呼叫流程說明
- 專案說明
- **補充說明及Q &A**

補充說明

- 雙金鑰對的使用
- 憑證廢止清冊的作法
- 中文字碼的問題
- 時戳服務
- 標準版與專業版的差異
- 微軟CryptoAPI與HiSECURE
- 版本種類以及取得的方式

雙金鑰對-IC卡使用注意事項

- GPKI是採用雙金鑰對(Dual-Key)的政策，所以每張IC卡會有兩對金鑰對，一對用於簽章，一對用於加解密，CA會分別為這兩對金鑰對簽發一張憑證，所以IC卡裡會存有兩張憑證，對應到這兩對金鑰對，所以要看您的AP需要的簽章或加解密，在不同時機使用不同的金鑰對與憑證，不要用錯了，否則可能會有安全及法律上的問題。

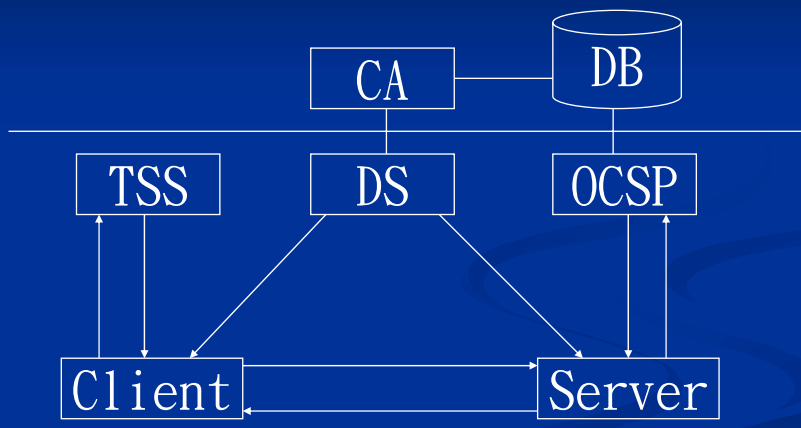
憑證廢止清冊的作法

- GPKI的憑證廢止清冊分為完整的廢止清冊(complete)與差異的廢止清冊(delta)
- 建議的使用方式為每個月固定一日下載完整的廢止清冊，中間的每一日則下載差異的廢止清冊
- 搭配廢止清冊函式GetCRLRecord將廢止的相關訊息一筆筆的放置於資料庫中，之後憑證廢止的查詢則是於資料庫中進行

中文字碼的問題

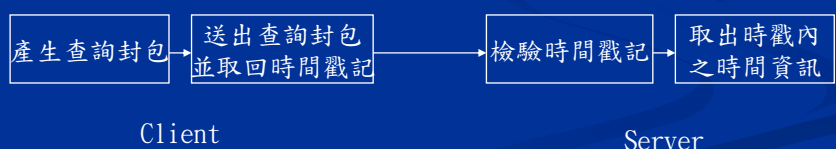
- HiSECURE的功能
  - 真實反應憑證內部所存放的資料
  - 中文字碼的部分在GPKI 憑證與憑證廢止清冊格式剖繪的規定下以UTF8編碼方式存放
  - UTF8轉BIG5或是CNS11643(戶役政字集)
  - 中華電信相關系統的作法—文鼎公司

時戳服務



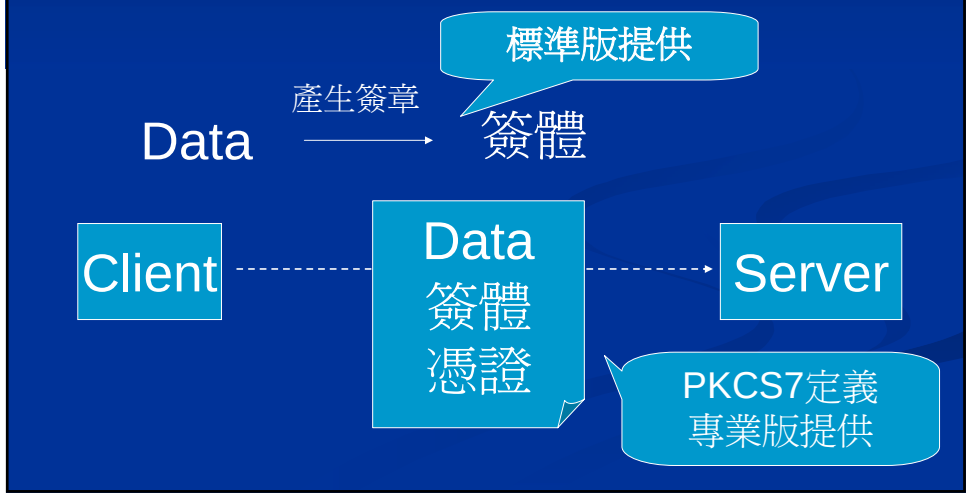
時戳服務函式介面

- makeSignedTSReq　產生時戳服務查詢封包
- requestTST　送出查詢封包並取回時間戳記
- verifyTST　查詢時間戳記是否正確
- getGenTimefromTST　取得Generalized-Time

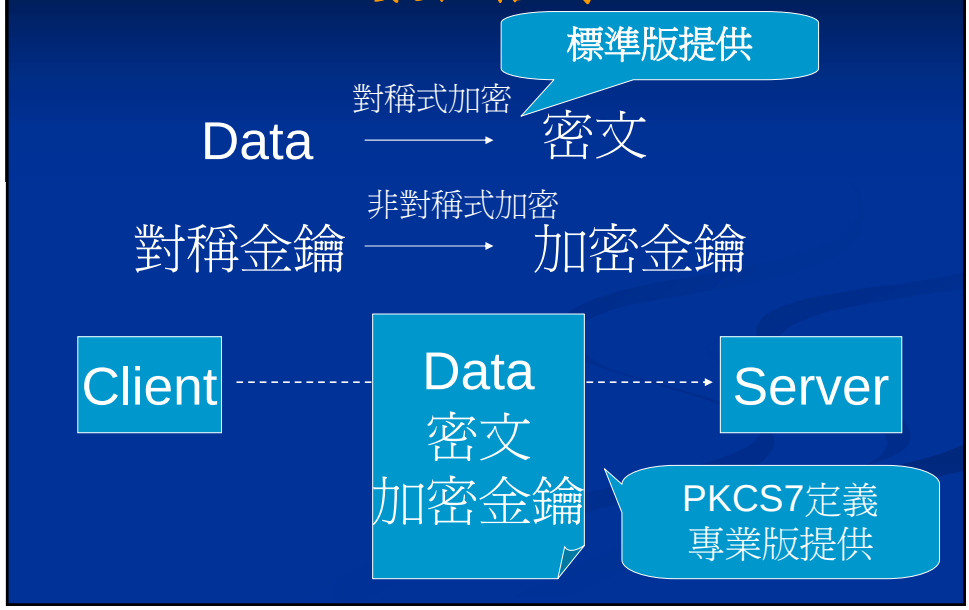


標準版與專業版

- 數位信封與數位簽章的提供



數位信封



版本種類以及取得的方式

- C語言提供三個平台(windows, Linux及Unix)，如申請標準版api的話，請上各CA的網站填申請表上傳後，並列印申請書 + 發文給各CA的業主(網站上都有詳細說明)。未來有更版的話，將發郵件通知給用戶。請按原帳號密碼上網下載 ；洽購專業版的話，請洽中華電信數據分公司 23445012 蔡鴻璟先生
- Java語言須等到今年5,6月之後才提供
- 測試版本(含開發套件、兩台讀卡機加上六張測試卡)聯絡窗口：中華電信研究所 8F0專案計畫 謝政勳 先生 (03) 424-4311
  相關網頁：gtestca.nat.gov.tw

結 語

- 藉由中華電信研究所所提供的 API，可以發展一個公開金鑰憑證管理體系，除了可以剖析憑證以及憑證廢止清單所包含的資訊，也提供了一種建構於對稱與非對稱加解密演算法的點對點安全保密傳輸方式。
- 因此，由中華電信研究所所提供的 API對開發一個建構在公開金鑰基礎建設(PKI)基本架構之上的應用系統已是相當足夠。


Q&A