.NET CryptoAPI
機能と利用法

2004.8.26

株式会社オレンジソフト

澤野弘幸

セキュリティAPIに関する技術調査
Part 3. .NET Crypto API：機能と利用法

- http://www.ipa.go.jp/security/fy15/reports/sec_api/index.html
- マイクロソフト Windowsにおけるセキュリティの基盤であるCryptoAPI の機能の概要を紹介し、以下のようなアプリケーションを例に、CryptoAPI の具体的な利用方法について解説します。
  - 暗号や電子署名をサポートしたアプリケーション
  - PKIを構成するアプリケーション
  - 暗号プロバイダ

目次

- CryptoAPIの概要
  - APIの構造、種類
  - 証明書ストア
  - おもな構造体
  - 証明書信頼リスト（CTL）
- CryptoAPIを使ったアプリケーション
  - CSP（Crypto Service Provider）について
  - ハッシュ、暗号化、復号（C/C++サンプル）
  - 証明書ストア（C/C++サンプル）
  - 証明書（C/C++サンプル）
  - 証明書の検証（C/C++サンプル）
  - PKCS#7デジタル署名と検証（C/C++サンプル）
  - PKCS#7暗号化と復号（C/C++サンプル）
  - 証明書および秘密鍵のインポートとエクスポート（C/C++サンプル）
- CryptoAPIのプロバイダの実装
  - Crypto Service Providerの実装（C/C++サンプル）
  - Revocation Providerの実装（C/C++サンプル）
- .NET Security
  - .NET Framework セキュリティツールについて
  - System.Security.Cryptography（C#サンプル）
  - System.Security.Cryptography.X509Certificates（C#サンプル）
  - CAPICOMとPInvoke（C#サンプル）

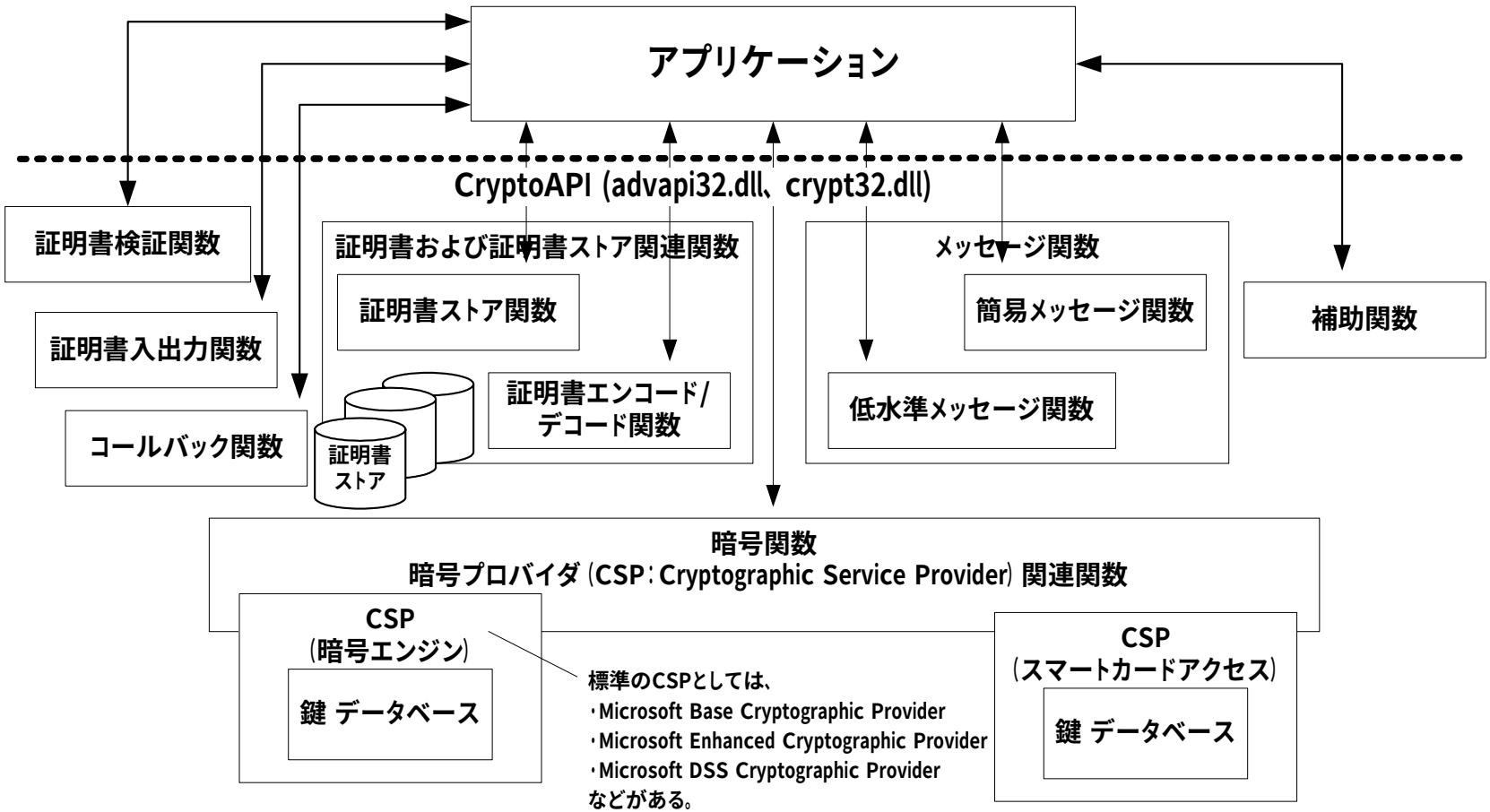
Win32

.NET Framework

CryptoAPIの概要

CryptoAPIの概要

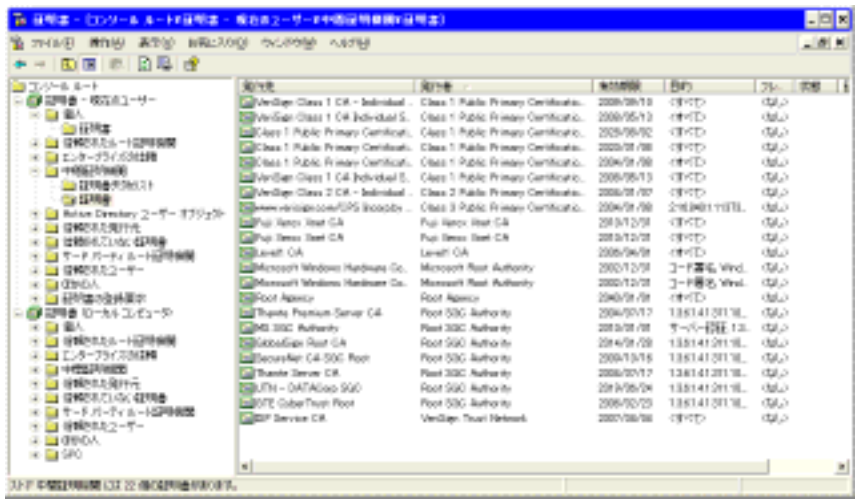
APIの構造、種類



CryptoAPIの概要

証明書ストア

- 公開鍵証明書や失効リストを保持する。
- 保持する形態は多様
  - レジストリ
  - 証明書ストアをシリアライズしたファイル
  - CMS(PKCS#7)
  - LDAPサーバー
- システムで持つ標準の証明書ストア
  - 「個人」”MY”：秘密鍵とペアになった証明書
  - 「ほかの人」”ADDRESSBOOK”：他のEE証明書
  - 「中間証明機関」”CA”：中間CA証明書
  - 「信頼されたルート証明期間」”ROOT”：トラストポイントとなるルートCA証明書



mmc.exeで証明書スナップインを追加して表示している画面

CryptoAPIの概要

構造体

- **構造体の種類**
  - General Cryptography Structures
  - Common Certificate Structures
  - X.509 Certificate Extension Structures
  - Message Structures
  - OID Support Structures
  - Certificate Chain Structures
  - CSP Structures
  - WinTrust Structures

Common Certificate Structuresの一部

CERT_INFO	解析された証明書の各フィールドの情報を持つ
CERT_CONTEXT	証明書コンテキスト。証明書のバイナリデータ、CERT_INFO、証明書ストアのハンドルを持つ
CRL_INFO	解析されたCRLの各フィールドの情報を持つ
CRL_CONTEXT	CRLコンテキスト。CRLのバイナリデータ、CRL_INFO、証明書ストアのハンドルを持つ
CTL_INFO	解析されたCTLの各フィールドの情報を持つ
CTL_CONTEXT	CTLコンテキスト。CTLのバイナリデータ,CTL_INFO、証明書ストアのハンドルなどを持つ

CryptoAPIの概要

証明書信頼リスト -CTL-

- **ルート証明書の配布に使う。**
  - CMS signedDataにMS独自のcontentを格納
- Out of Band なルート証明書の配布手法の一つの解決策。

フィールド	内容
SignedData	
encapContentInfo	
contentType	1.3.6.1.4.1.311.10.1
content	
使用目的（複数）	サーバー認証 クライアント認証 など
識別名	CTLの識別名（オプション）
開始有効期日	
終了有効期日	（オプション）
ハッシュ関数のID	拇印アルゴリズム(SHA1)
信頼する証明書の拇印（複数）	
certificates	信頼する証明書（オプション）
signerInfos	CTL自体の規定に含まれない（オプション）

CryptoAPIを使ったアプリケーション

CSP（Crypto Service Provider）について

- 暗号や、ハッシュ、デジタル署名などののアルゴリズム を実装している。
- ソフトウエアとして暗号アルゴリズムを提供するものや、セキュリティトークンなどのハードウエアを抽象化するためのものがある。
- Provider TypeとProvider Nameで識別

標準のProviderType

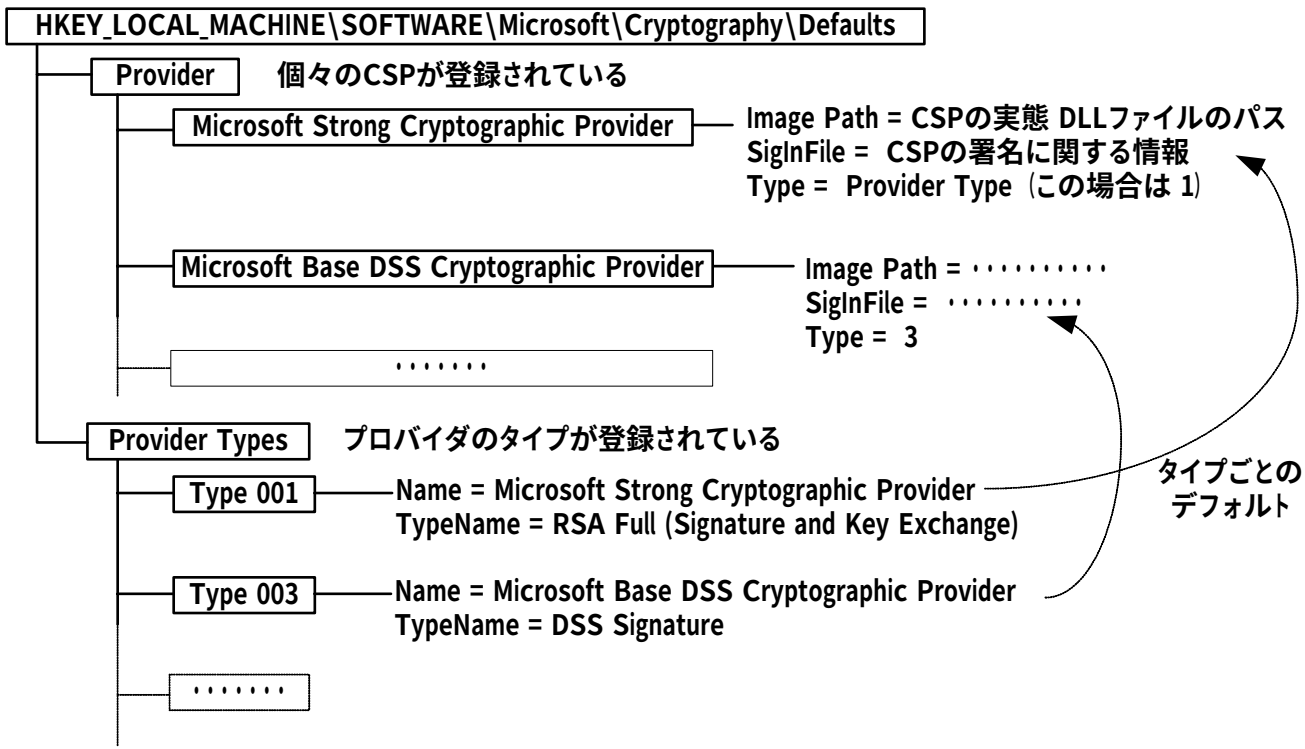
Provider Type	定義値	鍵交換	署名	共通鍵暗号	ハッシュ
PROV_RSA_FULL	1	RSA	RSA	RC2,RC4	MD5,SHA
PROV_RSA_SIG	2	—	RSA	—	MD5,SHA
PROV_DSS	3	—	DSS	—	MD5,SHA
PROV_FORTEZZA	4	KEA	DSS	Skipjack	SHA
PROV_MS_EXCHANGE	5	RSA	RSA	CAST	MD5
PROV_SSL	6	RSA	RSA	多種	多種
PROV_RSA_SCHANNEL	12	RSA	RSA	RC4、DES、トリプルDES	MD5,SHA
PROV_DSS_DH	13	DH	DSS	CYLINK MEK	MD5,SHA
PROV_RSA_AES	24	RSA	RSA	RC2,RC4,AES	MD5,SHA

Microsoft Base Cryptographic Provider
Microsoft Strong Cryptographic Provider
Microsoft Enhanced Cryptographic Provider

PROV_RSA_FULLである
標準で提供されるプロバイダの
Provider Name 3つ

CryptoAPIを使ったアプリケーション

レジストリのCSP



CryptoAPIを使ったアプリケーション

ハッシュ、暗号化、復号

利用するAPI

関数	説明
CryptEnumProviders	CSPを列挙する。
CryptAcquireContext	CSPのインスタンスを生成し、ハンドルを獲得する。
CryptDeriveKey	暗号鍵を得る。
CryptEncrypt	暗号化する。
CryptDecrypt	復号する。
CryptCreateHash	ハッシュのインスタンスを生成し、ハンドルを獲得する
CryptGetHashParam	ハッシュ値を得る。
CryptHashData	ハッシュの入力を与える。

CryptoAPIを使ったアプリケーション

ハッシュ(C/C++サンプル)

```
HCRYPTPROV hCryptProv;
HCRYPTKEY hKey;
BYTE *pbBuffer;       // ハッシュする入力データ
DWORD dwBufLen;    // ハッシュする入力データの長さ
BYTE *pbHash         // ハッシュ値の格納バッファ
BYTE *dwHashLen     // ハッシュ値の長さ
// CSPのハンドルの獲得
::CryptAcquireContext(&hCryptProv, NULL,
    MS_ENHANCED_PROV,    // CSPの名前
    PROV_RSA_FULL,         // CSPのタイプ
    0);
// SHA1ハッシュ関数のハンドルを獲得
::CryptCreateHash(hCryptProv, CALG_SHA1, 0, 0, &hHash);
::CryptHashData(hHash, pbBuffer, dwBufLen, 0))
::CryptGetHashParam(hHash, HP_HASHVAL, NULL, &dwHashLen, 0);
pbHash = (BYTE*)malloc(dwHashLen))
::CryptGetHashParam(hHash, HP_HASHVAL, pbHash, &dwHashLen, 0);
printf("The hash is:  ");
for(i = 0 ; i < dwHashLen ; i++) {
    printf("%2.2x ",pbHash[i]);   // ハッシュ値の出力
}
printf("¥n");
```

CSPを使ったハッシュの例

CryptoAPIを使ったアプリケーション

暗号化 (C/C++サンプル)

```
HCRYPTPROV hCryptProv;
HCRYPTKEY hKey;
HCRYPTKEY hXchgKey;
HCRYPTHASH hHash;
char *szPassword = “password”;
BYTE *pbBuffer;      // 暗号化対象データと暗号化データのバッファ
DWORD dwBufferLen; // 暗号化対象データの長さ
DWORD dwCount;    // 暗号化データのバッファの長さ
/*暗号化データのバッファ ブロック暗号ブロック長と入力データから判断した出力バッファのサイズを決める。*/
// CSPのハンドルの獲得
::CryptAcquireContext(&hCryptProv, NULL,
MS_ENHANCED_PROV,   // CSPの名前
PROV_RSA_FULL,        // CSPのタイプ
0);
// MD5ハッシュ関数のハンドルを獲得
::CryptCreateHash(hCryptProv, CALG_SHA1, 0, 0, &hHash);
// パスワードをハッシュ
::CryptHashData(hHash,  (BYTE *)szPassword, strlen(szPassword), 0);
//鍵ハンドルの獲得
::CryptDeriveKey(
hCryptProv,
CALG_3DES, // アルゴリズム
hHash, //ハッシュ関数のハンドル
KEYLENGTH_3DES | CRYPT_EXPORTABLE | CRYPT_NO_SALT, //鍵長
&hKey); //鍵ハンドル
::CryptEncrypt(hKey,
                        0,
                        TRUE, //データはこれで最後
                        0,
                        pbBuffer, // [in/out]暗号化対象データと暗号化データのバッファ
                        &dwCount, //暗号化対象データの長さ
                        dwBufferLen);  // pbBufferの長さ
::CryptDestroyKey(hKey); //鍵ハンドルの破棄
::CryptDestroyHash(hHash); //ハッシュハンドルの破棄
::CryptReleaseContext(hCryptProv, 0);// CSPハンドルの破棄
```

トリプルDESで暗号化の例

CryptoAPIを使ったアプリケーション

復号 (C/C++サンプル)

```
HCRYPTPROV hCryptProv;
HCRYPTKEY hKey;
HCRYPTKEY hXchgKey;
HCRYPTHASH hHash;
char *szPassword = “password”;
BYTE *pbBuffer;       // 複号対象データと複号結果データのバッファ
DWORD dwCount;     //複号対象データの長さ



(鍵ハンドルであるhKeyの獲得までの処理は暗号化と同様)

::CryptDecrypt(hKey, 0, TRUE, 0, NULL, &dwCount);
pbBuffer = (BYTE *)malloc(pbBuffer);
::CryptDecrypt(hKey, 0, TRUE, 0, pbBuffer, &dwCount);


free(pbBuffer);
```

暗号文の復号の例

CryptoAPIを使ったアプリケーション

証明書ストア (C/C++サンプル)

```
HCERTSTORE   hCertStore;       /*証明書ストアハンドル*/
PCCERT_CONTEXT  pCertContext   /*証明書コンテキスト*/
hCertStore = ::CertOpenSystemStore(NULL, “ADDRESSBOOK”);
while(pCertContext = ::CertEnumCertificatesInStore(hCertStore, pCertContext)) {
    char szNameString[256];
    ::CertGetNameString(pCertContext,
                        CERT_NAME_RDN_TYPE,
                        0,          /* CERT_NAME_ISSUER_FLAG だと発行元*/
                        NULL, szNameString, 256);
    printf(“subject : %s¥n”, szNameString);
}
::CertFreeCertificateContext(pCertContext);
::CertCloseStore(hCertStore, CERT_CLOSE_STORE_CHECK_FLAG);
```

証明書ストアから証明書コンテキストを獲得する例

CryptoAPIを使ったアプリケーション

証明書 (C/C++サンプル)

```
char szCertFile[] = “mycert.cer”
PCCERT_CONTEXT pCertContext;     /*証明書コンテキスト*/
CFile file(szCertFile, CFile::modeRead);
BYTE *pszBuf = new BYTE[file.GetLength()];
file.Read(pszBuf, file.GetLength());
pCertContext = ::CertCreateCertificateContext(
                      X509_ASN_ENCODING | PKCS_7_ASN_ENCODING,
                      pszBuf,
                      file.GetLength());
file.Close();
delete(pszBuf);
if (pCertContext == NULL) {
    DWORD dwErr = ::GetLastError();
}
```

X.509証明書ファイルから証明書コンテキストを作成する例

CryptoAPIを使ったアプリケーション

証明書の検証（C/C++サンプル）

- **証明書パスエンジン を作成し**（CertCreateCertificateChainEngine）、**検証を行う。**
- **パスの検証と獲得は**CertGetCertificateChain
- **証明書の失効検証は**CertVerifyRevocation

Demo.

PKCS#7signedData
デジタル署名と検証 (C/C++サンプル)

関数	説明
CryptSignMessage	デジタル署名をする。
CryptVerifyDetachedMessageSignature	デジタル署名（クリア署名）の検証をする。
CryptVerifyMessageSignature	デジタル署名（オペイク署名）の検証をする。

Demo.

PKCS#7envelopedData
暗号化と復号 (C/C++サンプル)

関数	説明
CryptEncryptMessage	暗号化をする。。
CryptDecryptMessage	復号をする
CryptSignAndEncryptMessage	デジタル署名と暗号化をする。
CryptDecryptAndVerifyMessageSignature	デジタル署名の検証と復号をする。

Demo.

証明書および秘密鍵のインポートとエクスポート(C/C++サンプル)

PKCS #12を扱う関数

関数	説明
PFXExportCertStoreEx	引数に指定された証明書ストアにある証明書と関連する秘密鍵を、引数に指定されたパスワードで保護してPKCS #12フォーマットで取り出す。
PFXImportCertStore	引数に指定されたPKCS #12フォーマットのデータとパスワードからから関連する秘密鍵を取り出し、内部の証明については、戻り値の証明書ストアハンドルで参照可能とする。
PFXIsPFXBlob	PKCS #12フォーマットのデータかチェックする。
PFXVerifyPassword	PKCS #12のパスワードであるかチェックする。

Demo.

CryptoAPIのプロバイダの実装

Crypto Service Providerの実装(C/C++サンプル)

- **開発手順**
  - CSPDK (Cryptographic Service Provider Developer's Toolkit) **の入手**
  - advapi32.dll**の置換**
    - **日本版**OS**では、**WindowsNT4.0**のみ**
  - **開発**
    - **アルゴリズムの実装**
    - **テスト用署名の作成**
    - **レジストリへの登録**
    - **デバック**
  - **正式版な署名を入手**

Demo.

Revocation Providerの実装 (C/C++サンプル)

- **標準の証明書検証機構では実装されていない処理を追加可能**
  - OCSP
  - **ポリシー初期パラメータ**
  - JPKIやLGPKIの証明書のような、URI指定されていないCRLの取得

Demo.

.NET Security

セキュリティツール

- **証明書作成ツール**Makecert.exe
- **証明書マネージャ ツール** Certmgr.exe
- **証明書検査ツール**Chktrust.exe
- **コードアクセス セキュリティポリシーツール**Caspol.exe
- **ファイル署名ツール**Signcode.exe
- **アクセス許可表示ツール**Permview.exe
- PEVerify**ツール**PEverify.exe
- Secutil**ツール**Secutil.exe
- **レジストリ設定ツール**Setreg.exe
- **ソフトウェア発行元証明書テスト ツール**Cert2spc.exe
- **厳密名ツール**Sn.exe

.NET Framework におけるセキュリティの要素

- 検証
- コードアクセスセキュリティ
- 権限
- エビデンス
- セキュリティポリシー
- ロールベースセキュリティ
- 暗号
  - 以降にサンプルで解説します…

System.Security.Cryptography

- **暗号サービスの提供**
  - ○○○CryptoServiceProvider
  - **ハッシュ**：MD5、SHA1、SHA256、SHA512
  - **共通鍵**：DES、RC2、Rijndael、TripleDES
  - **公開鍵**：DSA、RSA (PKCS#1 V1.5、OAEP)
  - **鍵つきハッシュ**：HMACSHA1、MACTripleDES
- **署名サンプルプログラム** (C#)
  - HashAlgorithm
  - RSACryptoServiceProvider
- **暗号化サンプルプログラム** (C#)
  - DESCryptoServiceProvider
  - CryptoStream

Demo.

System.Security.Cryptography.X509Certificates

- X509Certificate
  - **ほとんど**Get**メソッドのみ**
- X509CertificateCollection
- X509CertificateCollection.X509CertificateEnumerator
- **証明書参照サンプルプログラム** (C#)

Demo.

.NET Security

CAPICOM

- CrtpyoAPIを使うCOM
- マイクロソフトからインストーラをダウンロード
- CABファイルを解凍し、capicom.infでcapicom.dllをインストール
- 登録
  - > regsvr32 capicom.dl

CAPICOMが提供するクラスなど

Certificate Store Objects	証明書、証明書ストアなどを扱うクラスたち
Digital Signature Objects	PKCS #7 signedDataを扱うクラスたち
Enveloped Data Objects	PKCS #7 envelopedDataを扱うクラスたち
Data Encryption Objects	PKCS #7 EncryptedDataを扱うクラスたち
Auxiliary Objects	AlgorithmやOIDなどのクラスや、Settings、Utilities などのクラスも。
Interoperability Interfaces	ICertContext、ICertStore、IChainContext
Enumeration Types	アルゴリズムや、証明書ステータス、エラーなど各種の値の定義

.NET Security

CAPICOMサンプルプログラム (C#)

- **証明書ストア参照サンプルプログラム**
  - CAPICOM.StoreClass
  - CAPICOM.Certificate
  - CAPICOM.Certificates
- **証明書検証サンプルプログラム**
  - CAPICOM.Store
  - CAPICOM.Certificate
  - CAPICOM.Chain
  - CertFindChainInStore
- XML**署名サンプルプログラム**
  - CAPICOM.Store
  - System.Security.Cryptography.X509Certificates
  - System.Security.Cryptography.Xml
  - PublicKeyBlobFromCertificateRawData
  - ImportPublicKeyBlob
- CSP**を使った処理や、証明書パス検証、**PKCS#7**の処理などは**CryptoAPI**を呼び出す。**(PInvoke**する**)**。**

Demo.

おわり

http://www.orangesoft.co.jp
http://www.sgoma.org
sawano@orangesoft.co.jp