(주)한국무역정보통신

TradeSignAPI

TSToolkit 보안 모듈 설명서

Ver. 1.3

**개정 이력**

| **버 전** | **변경일** | **기준버전** | **변경 내용** | **작성자** |
| --- | --- | --- | --- | --- |
| 1.0 | 2006.05.20 | 1,0,0,2 | . |  |
| 1.1 | 2006.08.01 | 1,0,0,3 | GetLoginData 함수 설명 추가. |  |
| 1.2 | 2007.03.20 | 1,0,0,14 | SelectCertificate 함수 기능 설명 추가 |  |
| 1.3 | 2007.07.20 | 1.0.0.14 | 에러메시지 추가 |  |
|  |  |  |  |  |

**목 차**

[**1. TSToolkit 의 기능 및 설명 1**](#_heading=h.1fob9te)

[**2. TSToolkit 설치 방법 및 주의사항 2**](#_heading=h.3znysh7)

[**2.1 TSToolkit 사용 시 주의사항** 2](#_heading=h.2et92p0)

[**2.2 TSToolkit 보안 모듈 설치** 3](#_heading=h.tyjcwt)

[***2.2.1 Client (HTML)*** *3*](#_heading=h.3dy6vkm)

[***2.2.2 Server (ASP)*** *5*](#_heading=h.1t3h5sf)

[***2.2.3 Visual Basic*** *6*](#_heading=h.4d34og8)

[***2.2.4 Visual C++*** *7*](#_heading=h.2s8eyo1)

[**2.3 TSToolkit 환경 설정 파일** 8](#_heading=h.17dp8vu)

[**3. 함수 설명 13**](#_heading=h.3rdcrjn)

[**3.1 SetConfig** 13](#_heading=h.26in1rg)

[**3.2 GetErrorMessage** 16](#_heading=h.lnxbz9)

[**3.3 AddSignInSignedDataFile** 17](#_heading=h.35nkun2)

[**3.4 Base64DecodeData** 18](#_heading=h.1ksv4uv)

[**3.5 Base64DecodeFile** 19](#_heading=h.44sinio)

[**3.6 Base64EncodeData** 20](#_heading=h.2jxsxqh)

[**3.7 Base64EncodeFile** 21](#_heading=h.z337ya)

[**3.8 CertificateValidation** 22](#_heading=h.3j2qqm3)

[**3.9 ChangeKeyFilePassword2** 23](#_heading=h.1y810tw)

[**3.10 ClearMemory** 24](#_heading=h.4i7ojhp)

[**3.11 ClearMemory2** 25](#_heading=h.2xcytpi)

[**3.12 DecryptData** 26](#_heading=h.1ci93xb)

[**3.13 DecryptEnvelopedData** 27](#_heading=h.3whwml4)

[**3.14 DecryptEnvelopedFile** 28](#_heading=h.2bn6wsx)

[**3.15 DecryptFile** 29](#_heading=h.qsh70q)

[**3.16 EncryptData** 30](#_heading=h.3as4poj)

[**3.17 EncryptFile** 31](#_heading=h.1pxezwc)

[**3.18 EnvelopData** 32](#_heading=h.49x2ik5)

[**3.19 EnvelopFile** 33](#_heading=h.2p2csry)

[**3.20 GenerateRandomNumber** 34](#_heading=h.147n2zr)

[**3.21 GenerateSymmetricDrivedKey (추후미지원)** 35](#_heading=h.3o7alnk)

[**3.22 GenerateSymmetricKey** 36](#_heading=h.23ckvvd)

[**3.23 GenSignature** 37](#_heading=h.ihv636)

[**3.24 GetCertificate** 38](#_heading=h.32hioqz)

[**3.25 GetCertificatePropertyFromID** 39](#_heading=h.1hmsyys)

[**3.26 GetDataFromLDAP** 41](#_heading=h.41mghml)

[**3.27 GetEncryptionAlgorithm** 42](#_heading=h.2grqrue)

[**3.28 GetEncryptionMode** 43](#_heading=h.vx1227)

[**3.29 GetLoginData** 44](#_heading=h.3fwokq0)

[**3.30 GetPrivateKey** 45](#_heading=h.1v1yuxt)

[**3.31 GetSignerCert** 47](#_heading=h.4f1mdlm)

[**3.32 GetSignerCount** 48](#_heading=h.2u6wntf)

[**3.33 GetSignerDN** 49](#_heading=h.19c6y18)

[**3.34 GetSigningTime** 50](#_heading=h.3tbugp1)

[**3.35 GetSymmetricIV** 51](#_heading=h.28h4qwu)

[**3.36 GetSymmetricKey** 52](#_heading=h.nmf14n)

[**3.37 HashData** 53](#_heading=h.37m2jsg)

[**3.38 HashFile** 54](#_heading=h.1mrcu09)

[**3.39 LoadCertificate** 55](#_heading=h.46r0co2)

[**3.40 LoginData** 57](#_heading=h.2lwamvv)

[**3.41 LoginDataProcess** 58](#_heading=h.111kx3o)

[**3.42 SelectCertificate** 59](#_heading=h.3l18frh)

[**3.43 SetEncryptionAlgoAndMode** 61](#_heading=h.206ipza)

[**3.44 SetPeerCertificate** 63](#_heading=h.4k668n3)

[**3.45 SetSymmetricKey** 64](#_heading=h.2zbgiuw)

[**3.46 SignData** 65](#_heading=h.1egqt2p)

[**3.47 SignFile** 66](#_heading=h.3ygebqi)

[**3.48 VerifySignature** 67](#_heading=h.2dlolyb)

[**3.49 VerifySignedData** 68](#_heading=h.sqyw64)

[**3.50 VerifySignedFile** 70](#_heading=h.3cqmetx)

[**3.51 VerifyVID** 72](#_heading=h.1rvwp1q)

[**4. 에러 메시지 73**](#_heading=h.4bvk7pj)

# **1. TSToolkit 의 기능 및 설명**

TSToolkit 는 Windows 환경에서의 Server 와 Client 에서 사용할 수 있는 보안과 암호 모듈로, 서명 및 검증, 전자봉투(Envelop) 생성 및 복호, 암복호화(Encrypt/Decrypt), Hash, Base64Encode/Decode, 인증서 검증, 신원확인 등의 기능을 지원한다. 서명 및 검증, 전자봉투생성 및 검증, 암복호화, Hash, Base64Encode/Decode 는 파일 형식을 지원한다.

서명 및 검증 기능은 서명에 사용할 인증서를 읽어 들여 서명을 하고, 서명된 값을 검증하는 기능으로 서명 조건에 따라 생성되는 서명값에는 서명시간, 상위기관인증서, CRL, 원본데이타를 설정된 Config 에 따라 삽입 여부를 결정할 수 있다. 검증 시에는 서명시간이 포함된 경우 서명된 시간을 꺼낼 수 있다.

인증서 신원확인의 경우에는 인증서의 신원을 확인하는 기능으로, 선택된 인증서의 개인키에서 랜덤값을 꺼내야 하기 때문에 반드시 인증서가 있는 곳, 즉 사용자 인증서면 사용자PC 에서, 서버인증서면 서버에서 신원확인 기능을 수행해야 한다.

# **2. TSToolkit 설치 방법 및 주의사항**

## **2.1 TSToolkit 사용 시 주의사항**

TSToolkit 의 모든 함수는 출력값을 함수의 파라미터로 받을 수 없다. 단, 연관된 프로퍼티에 출력값을 저장하고 있어서 함수 호출 후 반드시 프로퍼티로부터 값을 받아 따로 보관하여야 한다. 이 출력값은 프로퍼티내에서 다른 함수 호출전까지만 유지되기 때문에 작업시 이 점을 유의하여 프로그램을 작성하여야 한다.

기본으로 제공되는 함수들은 가장 마지막에 함수 실행 시 성공 여부 등을 리턴 하는 리턴값을 갖는다.

TSToolkit 는 Visual C++ 6.0 환경에서 개발되었으므로 .NET 에서 사용할 경우 각별한 주의가 필요하다.

모든 함수 실행 후 성공 및 실패 여부는 함수의 리턴값으로 알 수 있다, 함수 실행 시 에러가 발생한 경우에는 GetErrorMessage 함수를 통해서 오류 메시지를 꺼낼 수 있다.

## **2.2 TSToolkit 보안 모듈 설치**

### **2.2.1 Client (HTML)**

Client Side 에서 보안 모듈을 사용하기 위해서는 우선 사용자 PC 에 보안관련 모듈이 설치되어 있어야 한다. 이를 위해 사용 전에 반드시 설치부분을 거치도록 구성을 하여야 한다.

**- 설치를 위한 기본 파일 및 정보.**

| TSToolkit.cab | 실제 TSToolkit 보안 모듈 및 필요한 정보가 들어 있는 설치파일 |
| --- | --- |
| TSToolkit 의 Version | TSToolkit 의 버전은 TSToolkit 모듈 설치를 위한 기본 정보로 TSToolkitObject.js 내부에서 사용된다. |
| TSToolkitObject.js | TSToolkit 보안 모듈을 자동으로 활성화시키기 위해 Object Code 를 출력하는 자바스크립 파일. 설치 및 모든 사용 페이지에 포함되어야 한다. |
| Setup.html | 자바스크립트 파일을 포함하여 TSToolkit 보안 모듈을 설치하는 예제 파일 |

**- 설치 및 사용을 위해 모든 페이지에 삽입되는 TSToolkitObject.js 의 사용법**

설치 및 사용을 위한 Object Code 는 TSToolkitObject.js 파일에 들어 있으며, 이 파일은 설치 및 사용하는 모든 페이지에 포함 되어야 한다.

TSToolkitObject.js 파일은 TSToolkit 보안 모듈을 사용하는 모든 페이지에 아래와 같이 추가되어야 한다.

| **<script language=”javascript” src="TSToolkitObject.js"></script>** |
| --- |

**- 설치 및 사용을 위한 Object Code (TSToolkitObject.js 파일 내에 존재함)**

| **<OBJECT id="TSToolkit"**  **CLASSID="CLSID:55D9860A-AB9C-44A1-BB74-75AF7F805333"**  **codebase="./download/TSToolkit.cab#version=1,0,0,1"**  **style="LEFT: 0px; TOP: 0px" width="0" height="0" VIEWASTEXT>**  **</OBJECT>** |
| --- |

. TSToolkit 의 Version 은 Object Code 내의 codebase 에 들어가서 사용자 컴퓨터에 설치된 TSToolkit 보안모듈의 버전과 비교가 되어 codebase 내의 버전이 더 높으면 사용자 컴퓨터에 재설치가 된다. 사용자 컴퓨터에 TSToolkit 보안모듈이 설치되어 있지 않으면 바로 설치된다.

. id 는 실행되는 보안 모듈을 가리키는 ID 로, 보안모듈을 호출할 때 사용된다.

. classid 는 TSToolkit 보안모듈의 고유한 clsid 를 값으로 갖는다.

. codebase 는 설치 시에 필요한 설치파일이 어디 있는 지와 그 파일의 버전을 값으로 갖는다. 업그레이드된 보안 모듈이 발생하면 업그레이드된 버전을 적어 사용자 컴퓨터에 새버전을 설치하도록 한다.

. style 은 웹 페이지에서 보안 모듈이 표시되지 않도록 사이즈 및 위치 등을 조정한다.

. 이 Object Code 는 Setup.html 파일 내에 이미 포함 되어 있다.

. 새로운 보안 모듈을 설치하지 않고, 기존에 설치된 보안 모듈을 사용하자고 하는 경우에는 codebase 를 넣지 않으면 이미 사용자 컴퓨터에 설치 되어진 보안 모듈이 실행하게 된다. 단, 이때, 모듈이 설치되어 있지 않으면, 웹 페이지 오류를 내게 된다.

### **2.2.2 Server (ASP)**

Server Side 에서 돌아가는 웹프로그래밍을 위해서는 TSToolkit 보안 모듈을 구성요소서비스에 등록을 하여야 서버 모듈에서 사용이 가능하다.

구성 요소 서비스에 등록하는 순서는 생략한다.

ASP 페이지에서 사용하기 위해서는 우선 모듈을 구성요소서비스에 등록을 한다. 그 후 사용하고자 하는 페이지에서 기본적인 변수를 설정하는 TSToolkitConfig.asp 파일을 포함시키고, 구성 요소 서비스에 등록된 모듈의 Instance 를 생성하여, 생성된 Instance 를 이용하여 작업을 한다.

| **/\* 기본적인 변수가 설정된 ASP 파일 \*/**  **<!--#include file = "TSToolkitConfig.asp" -->**  **<%**  **Dim TSToolkit**  **/\* TSToolkit 보안 모듈의 Instance 를 생성한다. \*/**  **set TSToolkit = Server.CreateObject("TSToolkit.TSToolkit.1")**  **…**  **%>** |
| --- |

### **2.2.3 Visual Basic**

Visual Basic 에서 TSToolkit 보안 모듈을 사용하기 위해서는 우선 TSToolkit 보안 모듈을 레지스트리에 등록하고, 등록된 라이브러리를 Visual Basic 프로젝트에 참조 시킨다.

| **[프로젝트(P)] [참조(N)…] 메뉴를 선택하여 “TSToolkit” 를 선택한다.** |
| --- |

Visual Basic 소스 안에서는 우선 라이브러리를 선언하고 선언된 라이브러리의 Instance 를 생성한 후, 생성된 Instance 를 이용하여 메소드를 사용한다. 사용한 후에는 Instance 를 제거하여 메모리를 정리한다.

| **‘ TSToolkit 보안 모듈을 선언한다.**  **Public eTSToolkit As TSToolkit**  **‘ 선언된 TSToolkit 보안 모듈에 Instance 를 생성하여 설정한다.**  **Set eTSToolkit = New TSToolkit**  **‘ 생성된 Instance 를 이용해 필요한 기능을 사용한다.**  **nRet = eTSToolkit.Base64Decode(strDecode, strEncodedData)**  **...**  **‘ 사용이 완료되면 Instance 를 정리 한다.**  **Set eTSToolkit = Nothing** |
| --- |

### **2.2.4 Visual C++**

Visual C++ 에서는 tlb 파일을 Import 시키고, Import 된 TSToolkit 보안 모듈의 포인터를 선언하여 사용하도록 한다.

라이브러리를 StdAfx.h 파일에 소스코드 수준에서 링크를 설정한다.

| **#import “TSToolkit.tlb” no\_namespace, named\_guides** |
| --- |

제공하는 기능을 사용하기 위해서 라이브러리의 포인터를 선언한다.

| **ITSToolkit \*eTSToolkit;**  **::CoCreateInstance(\_\_uuidof(TSToolkit), NULL, CLSCTX\_ALL, \_\_uuidof(ITSToolkit), (LPVOID\*)&eTSToolkit);** |
| --- |

포인터를 이용하여 필요한 기능을 사용한다.

| **nRet = eTSToolkit->Base64Decode(&pvBase64Decode);** |
| --- |

TSToolkit 모듈을 사용하기 위한 프로그램 초기화시 AfxOleInit() 함수를 사용한다.

다른 초기화 함수를 이용하여 초기화하기도 한다.

## **2.3 TSToolkit 환경 설정 파일**

TSToolkit 의 함수들을 살펴보면 SetConfig 함수처럼, 정해진 값, 또는 정해진 형식에 맞 는 값을 설정해 주어야 하는 경우가 있다. 이 때, 사용할 값들을 정리해둔 파일들이 있다.

이 파일은 한 사이트 또는 한 프로세스에는 동일한 환경을 가져간다는 생각으로 여러 군데 함수 호출 시마다 각각 지정 해야 할 값들을 환경 설정 파일 내에 변수로 설정하고, 그 변수를 함수 호출 시 사용하여, 변수의 값이 바뀌더라도, 소스의 변화는 최소화 할 수 있게 구성한 것이다.

환경 설정 파일을 쓴다는 것이 부담스럽다고 한다면 함수 호출시마다 해당하는 상수 값 등을 설정하여서 사용하면 된다.

여기서는 Web 환경에서 사용할 수 있는 JavaScript 파일로 설명한다.

| //\*\* 환경 설정시 주의사항 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*//  // . 인증기관 LDAP 정보 모음 은 LDAP 에서 인증서를 가져올 때 반드시 사용된다.  //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*//  //\*\* 기본정보 설정 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*//  // == 인증기관 관련 정보 모음 ================================= //  var CA\_LDAP\_INFO = "KISA:dirsys.rootca.or.kr:389|KICA:ldap.signgate.com:389|SignKorea:dir.signkorea.com:389|Yessign:203.233.91.35:389|CrossCert:dir.crosscert.com:389|TradeSign:ldap.tradesign.net:389|NCASign:ds.nca.or.kr:389|";  // == 인증서 정책 관련 ================================ //  // -- 법인 상호연동용 OID 모음  var FIRST\_COMP\_CERT\_POLICIES = "1 2 410 200012 1 1 3:법인범용|1 2 410 200004 5 1 1 7:법인범용|1 2 410 200005 1 1 5:법인범용|1 2 410 200004 5 2 1 1:법인범용|1 2 410 200004 5 4 1 2:법인범용|1 2 410 200004 5 3 1 1:법인범용|1 2 410 200004 5 3 1 2:법인범용|";  // -- 개인 상호연동용 OID 모음  var FIRST\_INDI\_CERT\_POLICIES = "1 2 410 200012 1 1 1:개인범용|1 2 410 200004 5 1 1 5:개인범용|1 2 410 200005 1 1 1:개인범용|1 2 410 200004 5 2 1 2:개인범용|1 2 410 200004 5 4 1 1:개인범용|1 2 410 200004 5 3 1 9:개인범용|";  // -- 모든 인증서 허용  var ALL\_CERT\_POLICIES = "";  // ===================================================== //  // == 인증서 저장매체 관련 ============================= //  var HARD\_DISK = 0;  var REMOVABLE\_DISK = 1;  var IC\_CARD = 2;  var PKCS11 = 3;  // ===================================================== //  // == 인증서 Type 관련 ================================= //  var CERT\_TYPE\_SIGN = 1;  var CERT\_TYPE\_KM = 2;  var DATA\_TYPE\_PEM = 0;  var DATA\_TYPE\_BASE64 = 1;  // ===================================================== //  // == HASH 알고리즘 ==================================== //  var HASH\_ID\_MD5 = 1;  var HASH\_ID\_RIPEMD160 = 2;  var HASH\_ID\_SHA1 = 3; // 기본적으로 사용함.  var HASH\_ID\_HAS160 = 4;  // ===================================================== //  // == 대칭키 알고리즘 & 모드 ========================== //  var SYMMETRIC\_ID\_DES = 1;  var SYMMETRIC\_ID\_3DES = 2; // 기본적으로 사용함.  var SYMMETRIC\_ID\_SEED = 3;  var SYMMETRIC\_MODE\_ECB = 1;  var SYMMETRIC\_MODE\_CBC = 2; // 기본적으로 사용함.  var SYMMETRIC\_MODE\_CFB = 3;  var SYMMETRIC\_MODE\_OFB = 4;  // ===================================================== //  // == 인증서 정보 관련 설정값 ========================== //  var CERT\_ATTR\_VERSION = 1;  var CERT\_ATTR\_SERIAL\_NUBMER = 2;  var CERT\_ATTR\_SIGNATURE\_ALGO\_ID = 3;  var CERT\_ATTR\_ISSUER\_DN = 4;  var CERT\_ATTR\_SUBJECT\_DN = 5;  var CERT\_ATTR\_SUBJECT\_PUBLICKEY\_ALGO\_ID = 6;  var CERT\_ATTR\_VALID\_FROM = 7;  var CERT\_ATTR\_VALID\_TO = 8;  var CERT\_ATTR\_PUBLIC\_KEY = 9;  var CERT\_ATTR\_SIGNATURE = 10;  var CERT\_ATTR\_KEY\_USAGE = 11;  var CERT\_ATTR\_AUTORITY\_KEY\_ID = 12;  var CERT\_ATTR\_SUBJECT\_KEY\_ID = 13;  var CERT\_ATTR\_EXT\_KEY\_USAGE = 14;  var CERT\_ATTR\_SUBJECT\_ALT\_NAME = 15;  var CERT\_ATTR\_BASIC\_CONSTRAINT = 16;  var CERT\_ATTR\_POLICY = 17;  var CERT\_ATTR\_CRLDP = 18;  var CERT\_ATTR\_AIA = 19;  var CERT\_ATTR\_VALID = 20;  // ===================================================== //  // == 인증서 Type 관련 ================================= //  var DATA\_TYPE\_CACERT = 1;  var DATA\_TYPE\_SIGN\_CERT = 2;  var DATA\_TYPE\_KM\_CERT = 3;  var DATA\_TYPE\_CRL = 4;  var DATA\_TYPE\_ARL = 5;  // ===================================================== //   | **\*\* 여기까지는 함수에서 사용하는 대표적인 변수값들을 정의 해 놓은 것이다.**  **차후 추가는 있을 수 있어도 바뀔 일은 거의 없다.** | | --- |   //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*//  //\*\* 환경 설정 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*//  // 인증서 선택시 기본 매체.  **var STORAGE\_TYPE = HARD\_DISK;**  // 사용하고자 하는 인증서 정책 모음.  **var POLICIES = FIRST\_COMP\_CERT\_POLICIES;**  // 서명시 필요한 Config 조절.  // 서명 생성시 인증서 포함 여부, 0 : 서명자 인증서만 포함.(기본), 1 : 서명자 & CA 인증서 포함.  **var INC\_CERT\_SIGN = 0;**  // 서명 생성시 CRL 인증서 포함 여부, 0 : 미포함 (기본), 1 : 포함,  **var INC\_CRL\_SIGN = 0;**  // 서명 생성시 서명시간 포함 여부, 0 : 미포함, 1 : 포함(기본)  **var INC\_SIGN\_TIME\_SIGN = 1;**  // 서명 생성시 원본데이타 포함 여부 , 0 : 미포함, 1 : 포함(기본)  **var INC\_CONTENT\_SIGN = 1;**  // 인증서 검증에 필요한 Config 조절  // 사용자 인증서 검증 조건, 0 : CRL 체크 안함. 1 : 현재시간기준으로 유효한 CRL 사용(기본), 2 : 현재 시간기준으로 유효한 CRL 못 구할 시 이전 CRL 사용.  **var USING\_CRL\_CHECK = 1;**  // CA 인증서 검증 조건, 0 : ARL 체크 안함. 1 : 현재시간기준으로 유효한 ARL 사용(기본), 2 : 현재 시간기준으로 유효한 CRL 못 구할 시 이전 ARL 사용.  **var USING\_ARL\_CHECK = 0;**  var CTL\_INFO = "";   | **\*\* 여기까지는 위에 정의된 변수들을 이용하거나 새로운 값을 설정하여서 실제적으로 사이트 운영 환경을 결정하는 부분이다. 사이트 운영 환경은 사이트 운영 정책에 따라 어느 때든지 변경할 수 있다.**  **\*\* 인증서 정책의 경우, 위에 정의된 변수에 추가적으로 허용하고자 하는 인증서 정책을 덧붙일 수 있다.**  **예) var POLICIES = FIRST\_COMP\_CERT\_POLICIES + “1 2 3 4 5 6:테스트용|”;**  **\*\* 현재 보이는 변수 값이 가장 기본적인 환경설정 기준이다.** | | --- |     // Envelop 테스트시 사용하는 상대방 인증서  var pemSignCert, pemSignKey, pemKMCert, pemKMKey;  pemSignCert = "-----BEGIN CERTIFICATE-----MIIE8TCCA9mgAwIBAgI…………………중간생략……………………InCKA3JB4ZJFr4/HKMo-----END CERTIFICATE-----";  pemSignKey = "-----BEGIN ENCRYPTED PRIVATE KEY-----MIIC/jBIBgkqhkiG9w0B……중간생략…………fSsgU56XDXYt8P/sm3Ew=-----END ENCRYPTED PRIVATE KEY-----";  pemKMCert = "-----BEGIN CERTIFICATE-----MIIElzCCA3+gAwIBAgI………………………중간생략……………………1RIScHB8PNzh3CjbNVR-----END CERTIFICATE-----";  pemKMKey = "-----BEGIN ENCRYPTED PRIVATE KEY-----MIICzjBIBgkqhkiG9w0B……중간생략…………VaRE9iJfQw9w+LZ6m070=-----END ENCRYPTED PRIVATE KEY-----";   | **\*\* 위의 변수들은 Envelop 예제를 웹에서 테스트할 필요한 상대방 인증서와 개인키로 테스트 용으로 적용한 것이다. 실적용시에는 상대방의 인증서만 설정하고, 개인키는 따로 읽어서 처리하도록 해야 한다.** | | --- |   //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*// |
| --- | --- | --- | --- |

# **3. 함수 설명**

## **3.1 SetConfig**

STDMETHOD(SetConfig)( /\*[in]\*/BSTR btSiteName, /\*[in]\*/BSTR btLDAPInfo, /\*[in]\*/BSTR btCTLInfo, /\*[in]\*/BSTR btInitPolicies, /\*[in]\*/short nIncludeCertPath, /\*[in]\*/short nIncludeSigningTime, /\*[in]\*/short nIncludeCRL, /\*[in]\*/short nIncludeContent, /\*[in]\*/short nCRLCheckOption, /\*[in]\*/short nARLCheckOption, /\*[out, retval]\*/short \*pVal);

**기능**

모듈 실행에 필요한 기본적인 Config 를 설정한다. 한번 설정된 값은 메모리를 정리하는 함수를 호출하기 전, 또는 모듈이 메모리에서 내려가기 전까지 유지 된다.

**파라미터**

btSiteName (intput)

각 사이트 명으로 현재는 사용하지 않는다. 공백문자열로 설정한다.

btLDAPInfo (intput)

허용되는 인증기관의 LDAP 정보 리스트. 각 공인인증기관에 대해서는 “ | “ 로, 인증기관정보는 인증기관명, URL 및 포트를 “ : “ 로 구분한다. 마지막은 반드시 “ | “ 으로 끝나도록 한다.

btCTLInfo (intput)

인증서 검증에 사용되는 CTL 정보. 현재는 공백문자열로 설정한다.

btInitPolicies (intput)

허용되는 인증서 정책 및 정책명 리스트. 각 허용정책의 OID들은 “ | “ 로 구분되고,인증서 정책과 정책명의 구분은 “ : “ 로 구분된다. 마지막은 반드시 “ | “ 으로 끝나도록 한다.

nStorageType (intput)

기본으로 인증서를 읽는 저장매체를 지정한다.

nIncludeCertPath (intput)

서명데이타 생성시 상위 인증기관 인증서 포함여부. 기본값으로 포함하지 않도록 한다.

nIncludeSigningTime (intput)

서명데이타 생성시 서명시간 포함 여부. 기본값으로 포함하도록 한다.

nIncludeCRL (intput)

서명데이타 생성시 CRL 포함 여부. 기본값으로 포함하지 않도록 한다.

nIncludeCRL (intput)

서명데이타 생성시 원문 데이터 포함 여부. 기본값으로는 포함하도록 한다.

nCRLCheckOption (intput)

인증서 검증시 CRL 검증 포함 여부.

nARLCheckOption (intput)

인증서 검증시 ARL 검증 포함 여부.

pVal (output)

함수 실행시의 에러코드. 에러가 없으면 0 을 리턴 한다. 1 이상은 에러.

**추가 설명**

- 각 인증기관별 LDAP 정보.

| 인증기관명 | LDAP DNS | 포트 |
| --- | --- | --- |
| KISA | dirsys.rootca.or.kr | 389 |
| KICA | ldap.signgate.com | 389 |
| SignKorea | dir.signkorea.com | 389 |
| Yessign | ds.yessign.or.kr | 389 |
| CrossCert | dir.crosscert.com | 389 |
| TradeSign | ldap.tradesign.net | 389 |
| NCASign | ds.nca.or.kr | 389 |

\*KISA 의 경우, 2007년 3월 이후에는 Open 되지 않는다.

- 저장매체관련 설정값.

| 설정값 | 저장매체 |
| --- | --- |
| 0 | HARD\_DISK |
| 1 | REMOVABLE\_DISK |
| 2 | IC\_CARD |
| 3 | PKCS11 |

- 서명데이타 생성시 상위 인증기관 인증서 포함여부

| 설정값 | 상위 인증기관 인증서 포함여부 |
| --- | --- |
| 0 | 포함 안 함 (기본) |
| 1 | 포함함 |

\* 상위 인증기관 인증서를 포함하면 서명데이타의 사이즈가 기본 2~4 K 정도 더 늘어난다.

\* 서명자의 인증서는 기본으로 포함되게끔 되어 있다.

- 서명데이타 생성시 서명시간 포함 여부

| 설정값 | 서명시간 포함 여부 |
| --- | --- |
| 0 | 포함 안함 |
| 1 | 포함 함 (기본) |

- 서명데이타 생성시 CRL 포함 여부

| 설정값 | CRL 포함 여부 |
| --- | --- |
| 0 | 포함 안함 (기본) |
| 1 | 포함 함 |

\* CRL 을 포함할 경우, CRL 크기에 따라 서명데이타의 사이즈는 기본 서명데이타 사이즈 에 CRL 사이즈 , 그리고, 추가 몇십 Byte 가 추가된다.

- 인증서 검증시 CRL 검증 포함 여부

| 설정값 | CRL 검증 포함 여부 |
| --- | --- |
| 0 | 검증 안함 |
| 1 | 현재 유효한 CRL로 검증함 (기본) |
| 2 | 유효하지 않은 CRL로 검증함 |

- 인증서 검증시 ARL 검증 포함 여부

| 설정값 | ARL 검증 포함 여부 |
| --- | --- |
| 0 | **검증 안함** (기본) |
| 1 | 현재 유효한 CRL로 검증함 |
| 2 | 유효하지 않은 CRL로 검증함 |

## **3.2 GetErrorMessage**

STDMETHOD(GetErrorMessage)(/\*[out, retval]\*/BSTR \*pVal);

**기능**

본 함수 호출 바로 전에 실행된 함수에서 발생된 에러의 에러메시지를 리턴 한다.

**파라미터**

pVal (output)

본 함수 호출 바로 전에 실행된 함수에서 발생된 에러의 에러메시지

**추가 설명**

오류가 없을 경우에는 “” 이 리턴 된다.

오류가 있을 시에는 오류번호를 포함하는 아래와 같은 형식의 오류메시지가 나온다.

---------------------

오류번호 : \*\*\*\*

오류메시지 : \*\*\*\*\*\*

상세메시지 : \*\*\*\*\*\*

---------------------

## **3.3 AddSignInSignedDataFile**

STDMETHOD(AddSignInSignedDataFile)(/\*[in]\*/BSTR btSignedDataFile, /\*[in]\*/BSTR btAddedSignedDataFile, /\*[out, retval]\*/short \*pVal);

**기능**

본 함수 호출 바로 전에 실행된 함수에서 발생된 에러의 에러메시지를 리턴한다.

**파라미터**

btSignedDataFile (input)

서명을 추가하고자 하는 서명데이타 파일의 경로.

btAddedSignedDataFile (input)

서명이 추가된 서명데이타가 저장될 경로.

pVal (output)

함수 실행시의 에러코드. 에러가 없으면 0 을 리턴한다. 1 이상은 에러.

**추가 설명**

함수 호출전 추가하고자 하는 서명자의 인증서가 먼저 메모리에 로딩되어야 한다.

**연관된 함수**

SignFile

SelectCertificate

LoadCertificate

## **3.4 Base64DecodeData**

STDMETHOD(Base64DecodeData)(/\*[in]\*/BSTR btEncodedData, /\*[out, retval]\*/short \*pVal);

**기능**

Base64 Encoding 된 문자열을 Base64 Decoding 한다.

**파라미터**

vEncodedData (intput)

Base64 Encoding 된 데이터.

pVal (output)

함수 실행시의 에러코드. 에러가 없으면 0 을 리턴한다. 1 이상은 에러.

**연관된 프로퍼티(리턴값)**

OutData

함수 실행 후 Base64 Decoding 된 값을 가지고 있다.

**연관된 함수**

Base64EncodeData

**추가 설명**

## **3.5 Base64DecodeFile**

STDMETHOD(Base64DecodeFile)(/\*[in]\*/BSTR btEncodedDataFile, /\*[in]\*/BSTR btDataFile, /\*[out, retval]\*/short \*pVal);

**기능**

Base64 Encoding 된 문자열을 가진 파일을 읽어서 Base64 Decoding 한 후 지정된 경로에 저장한다.

**파라미터**

btEncodedDataFile (intput)

Base64 Encoding 된 데이터를 읽어 들일 절대 경로.

btDataFile (intput)

Base64 Decoding 된 데이터를 저장할 절대경로.

pVal (output)

함수 실행시의 에러코드. 에러가 없으면 0 을 리턴 한다. 1 이상은 에러.

**연관된 함수**

Base64EncodeFile

**추가 설명**

## **3.6 Base64EncodeData**

STDMETHOD(Base64EncodeData)(/\*[in]\*/BSTR btData, /\*[out, retval]\*/short \*pVal);

**기능**

일반 데이터를 Base64 Encoding 한다.

**파라미터**

btData (intput)

Base64 Encoding 할 일반 데이터.

pVal (output)

함수 실행시의 에러코드. 에러가 없으면 0 을 리턴 한다. 1 이상은 에러.

**연관된 프로퍼티(리턴값)**

OutData

함수 실행 후 Base64 Encoding 된 값을 가지고 있다.

**연관된 함수**

Base64DecodeData

**추가 설명**

## **3.7 Base64EncodeFile**

STDMETHOD(Base64EncodeFile) (/\*[in]\*/BSTR btDataFile, /\*[in]\*/BSTR btEncodedDataFile, /\*[out, retval]\*/short \*pVal);

**기능**

일반 데이터를 가진 파일을 읽어서 Base64 Encoding 한 후 지정된 경로에 저장한다.

**파라미터**

btDataFile (intput)

일반 데이터를 읽어 들일 절대 경로.

btEncodedDataFile (intput)

Base64 Encoding 된 데이터를 저장할 절대경로.

pVal (output)

함수 실행시의 에러코드. 에러가 없으면 0 을 리턴 한다. 1 이상은 에러.

**연관된 프로퍼티 (리턴값)**

**연관된 함수**

Base64DecodeFile

**추가 설명**

## **3.8 CertificateValidation**

STDMETHOD(CertificateValidation)(/\*[in]\*/BSTR btCert, /\*[out, retval]\*/short \*pVal);

**기능**

읽어들인 인증서의 유효성 검증을 수행한다.

**파라미터**

btCert (output)

읽어 들인 인증서. PEM 형식의 인증서

pVal (output)

함수 실행시의 에러코드. 에러가 없으면 0 을 리턴 한다. 1 이상은 에러.

**연관된 프로퍼티(리턴값)**

OutData

**연관된 함수**

SetConfig

**추가 설명**

SetConfig 함수에서 설정된 CRL, ARL 검증 옵션에 영향을 받는다.

CRL, ARL 모두 지정된 위치에 우선 저장되며, 유효한 CRL, ARL 만을 사용하여 검증할 때는 먼저 저장된 CRL, ARL을 살펴보고, 유효하면 바로 사용하고, 유효하지 않을 때에는 LDAP 에서 유효한 CRL, ARL 을 가져와 검증을 수행한다.

에러코드가 0 이면 유효한 인증서이다. 에러코드가 0 이 아닌 경우에는 유효하지 않은 인증서에 해당하는 에러코드(ERR\_REVOKED\_CERT)인지 확인이 필요하다. 폐지 또는 효력정지로 판단된 경우, OutData 프로퍼티에 폐지 또는 효력정지된 시간이 설정된다. 폐지와 효력정지를 따로 구분하여 값을 리턴하지는 않는다.

## **3.9 ChangeKeyFilePassword2**

STDMETHOD(CertificateValidation)(/\*[in]\*/BSTR btCert, /\*[out, retval]\*/short \*pVal);

**기능**

읽어들인 인증서의 유효성 검증을 수행한다.

**파라미터**

btCert (output)

읽어 들인 인증서. PEM 형식의 인증서

pVal (output)

함수 실행시의 에러코드. 에러가 없으면 0 을 리턴 한다. 1 이상은 에러.

**연관된 프로퍼티(리턴값)**

OutData

**연관된 함수**

SetConfig

**추가 설명**

SetConfig 함수에서 설정된 CRL, ARL 검증 옵션에 영향을 받는다.

CRL, ARL 모두 지정된 위치에 우선 저장되며, 유효한 CRL, ARL 만을 사용하여 검증할 때는 먼저 저장된 CRL, ARL을 살펴보고, 유효하면 바로 사용하고, 유효하지 않을 때에는 LDAP 에서 유효한 CRL, ARL 을 가져와 검증을 수행한다.

에러코드가 0 이면 유효한 인증서이다. 에러코드가 0 이 아닌 경우에는 유효하지 않은 인증서에 해당하는 에러코드(ERR\_REVOKED\_CERT)인지 확인이 필요하다. 폐지 또는 효력정지로 판단된 경우, OutData 프로퍼티에 폐지 또는 효력정지된 시간이 설정된다. 폐지와 효력정지를 따로 구분하여 값을 리턴하지는 않는다.

## **3.10 ClearMemory**

STDMETHOD(ClearMemory)(/\*[out, retval]\*/short \*pVal);

**기능**

서명데이타 검증시 설정된 서명데이타 정보들(서명자인증서, 서명자인증서DN, 서명시간)을 메모리에서 삭제한다.

**파라미터**

pVal (output)

함수 실행시의 에러코드. 에러가 없으면 0 을 리턴 한다. 1 이상은 에러.

**연관된 프로퍼티(리턴값)**

**연관된 함수**

VerifySignedData

VerifySignedFile

LoginDataProcess

**추가 설명**

VerifySignedData 또는 VerifySignedFile 함수 호출시 설정된 서명데이타 정보들을 메모리에서 삭제한다. 함수 호출 이후에 반드시 호출되어야 한다.

## **3.11 ClearMemory2**

STDMETHOD(ClearMemory2)(/\*[in]\*/short nDataType, /\*[out, retval]\*/short \*pVal);

**기능**

입력되는 데이터타입에 따른 서명데이타 검증시 설정된 서명데이타 정보들(서명자인증서, 서명자인증서DN, 서명시간)을 메모리에서 삭제한다.

**파라미터**

nDataType (output)

메모리에 올라온 데이터 의 종류를 설정한다.

pVal (output)

함수 실행시의 에러코드. 에러가 없으면 0 을 리턴 한다. 1 이상은 에러.

**연관된 프로퍼티(리턴값)**

**연관된 함수**

LoadCertificate

SelectCertificate

VerifySignedData

VerifySignedFile

LoginDataProcess

…

**추가 설명**

VerifySignedData 또는 VerifySignedFile 등의 함수 호출시 설정된 정보들을 메모리에서 삭제한다. 호출되는 함수에 따라 DataType 은 달라진다. 0 인 경우는 설정된 모든 메모리가 Clear 된다. 1 인 경우는 서명검증 절차를 진행후에 설정된 메모리를 Clear하고, 2 인 경우는 로딩된 인증서 정보가 모두 Clear 된다.

## **3.12 DecryptData**

STDMETHOD(DecryptData)(/\*[in]\*/BSTR btEncryptedData, /\*[in]\*/BSTR btEncryptionKey, /\*[out, retval]\*/short \*pVal);

**기능**

EncryptedData 를 Decrypt 한다. 함수 실행 이전에 SetEncryptionAlgoAndMode 함수를 호출하여 Encrypt/Decrypt 에 필요한 알고리즘과 모드를 설정할 수 있다. 설정하지 않는 경우에는 기본으로 설정된 알고리즘과 모드가 사용된다.

**파라미터**

btEncryptedData (intput)

Decrypt 하기 위한 EncryptedData

btEncryptionKey (intput)

Encrypt / Decrypt 에 사용되는 대칭키.

pVal (output)

함수 실행시의 에러코드. 에러가 없으면 0 을 리턴 한다. 1 이상은 에러.

**연관된 프로퍼티(리턴값)**

OutData

함수 실행 후 Decrypt 된 값을 가지고 있다.

**연관된 함수**

SetEncryptionAlgoAndMode

EncryptData

**추가 설명**

대칭키는 Base64 Encoding 된 형태의 값으로 설정해 주어야 한다.

## **3.13 DecryptEnvelopedData**

STDMETHOD(DecryptEnvelopedData) (/\*[in]\*/BSTR btEnvelopedData, /\*[out, retval]\*/short \*pVal);

**기능**

EnvelopedData 를 복호화 하여 데이타를 꺼낸다.

**파라미터**

btEnvelopedData (intput)

복호화 하고자 하는 EnvelopedData.

pVal (output)

함수 실행시의 에러코드. 에러가 없으면 0 을 리턴 한다. 1 이상은 에러.

**연관된 프로퍼티(리턴값)**

OutData

**연관된 함수**

EnvelopData

SelectCertificate

LoadCertificate

**추가 설명**

EnvelopedData 를 복호화 할 때는 EnvelopedData 생성시 사용된 인증서의 인증서와 개인키가 메모리에 로드되어 있어야 한다. 내부적으로 인증서 셋트를 메모리 하였을 경우에는 키분배용(암호용)인증서와 서명용 인증서 중 맞는 것을 택일하여 복호화한다.

## **3.14 DecryptEnvelopedFile**

STDMETHOD(DecryptEnvelopedFile)(/\*[in]\*/BSTR btEnvelopedDataFile, /\*[in]\*/BSTR btDataFile, /\*[out, retval]\*/short \*pVal);

**기능**

EnvelopedData 를 복호화 하여 데이타를 꺼낸다.

**파라미터**

btEnvelopedDataFile (intput)

복호화 하고자 하는 EnvelopedData 파일의 절대 경로.

btDataFile (intput)

복호화 후 나오는 데이터를 저장할 절대 경로.

pVal (output)

함수 실행시의 에러코드. 에러가 없으면 0 을 리턴 한다. 1 이상은 에러.

**연관된 프로퍼티(리턴값)**

**연관된 함수**

EnvelopFile

SelectCertificate

LoadCertificate

**추가 설명**

EnvelopedData 를 복호화 할 때는 EnvelopedData 생성시 사용된 인증서의 인증서와 개인키가 메모리에 로드되어 있어야 한다. 내부적으로 인증서 셋트를 메모리 하였을 경우에는 키분배용(암호용)인증서와 서명용 인증서 중 맞는 것을 택일하여 복호화한다.

## **3.15 DecryptFile**

STDMETHOD(DecryptFile)(/\*[in]\*/BSTR btEncryptedDataFile, /\*[in]\*/BSTR btDataFile, /\*[in]\*/BSTR btEncryptionKey, /\*[out, retval]\*/short \*pVal);

**기능**

대칭키로 Encrypt 된 데이터 파일을 읽어서 Decrypt 한 내용을 지정된 경로에 저장한다.

**파라미터**

btEncryptedDataFile (intput)

Encrypt 된 데이터 파일의 절대 경로.

btDataFile (intput)

Decrypt 한 내용을 저장할 절대 경로.

btEncryptionKey (intput)

Encrypt / Decrypt 에 사용되는 대칭키.

pVal (output)

함수 실행시의 에러코드. 에러가 없으면 0 을 리턴 한다. 1 이상은 에러.

**연관된 프로퍼티(리턴값)**

**연관된 함수**

SetEncryptionAlgoAndMode

EncryptFile

**추가 설명**

대칭키는 Base64 Encoding 된 형태의 값으로 설정해 주어야 한다.

## **3.16 EncryptData**

STDMETHOD(EncryptData)(/\*[in]\*/BSTR btData, /\*[in]\*/BSTR btEncryptionKey, /\*[out, retval]\*/short \*pVal);

**기능**

입력 받은 일반 데이터를 주어진 대칭키로 Encrypt 한다.

**파라미터**

btData (intput)

Encrypt 할 데이터.

btEncryptionKey (intput)

Encrypt / Decrypt 에 사용되는 대칭키.

pVal (output)

함수 실행시의 에러코드. 에러가 없으면 0 을 리턴 한다. 1 이상은 에러.

**연관된 프로퍼티(리턴값)**

OutData

함수 실행 후 Encrypt 된 데이터를 가지고 있다.

**연관된 함수**

SetEncryptionAlgoAndMode

DecryptData

**추가 설명**

대칭키는 Base64 Encoding 된 형태의 값으로 설정해 주어야 한다.

## **3.17 EncryptFile**

STDMETHOD(EncryptFile)(/\*[in]\*/BSTR btDataFile, /\*[in]\*/BSTR btEncryptedDataFile, /\*[in]\*/BSTR btEncryptionKey, /\*[out, retval]\*/short \*pVal);

**기능**

일반 데이터가 들어있는 파일을 읽어서 Encrypt 한 후 지정된 경로에 저장한다.

**파라미터**

btDataFile (intput)

Encrypt 하고자 하는 데이터 파일의 절대경로.

btEncryptedDataFile (intput)

Encrypt 된 데이터를 저장할 파일의 절대경로.

btEncryptionKey (intput)

Encrypt / Decrypt 에 사용되는 대칭키.

pVal (output)

함수 실행시의 에러코드. 에러가 없으면 0 을 리턴 한다. 1 이상은 에러.

**연관된 프로퍼티(리턴값)**

**연관된 함수**

SetEncryptionAlgoAndMode

DecryptFIle

**추가 설명**

대칭키는 Base64 Encoding 된 형태의 값으로 설정해 주어야 한다.

## **3.18 EnvelopData**

STDMETHOD(EnvelopData)(/\*[in]\*/BSTR btData, /\*[out, retval]\*/short \*pVal);

**기능**

상대방 인증서로 데이터를 Envelop 하여 EnvelopedData 를 생성한다.

**파라미터**

btData (intput)

Envelop 할 일반 데이터.

pVal (output)

함수 실행시의 에러코드. 에러가 없으면 0 을 리턴 한다. 1 이상은 에러.

**연관된 프로퍼티(리턴값)**

OutData

함수 실행 후 EnvelopedData 를 가지고 있다.

**연관된 함수**

DecryptEnvelopedData

SetPeerCertificate

**추가 설명**

EnvelopedData 를 생성하기 위해서는 비대칭암호화에 사용할 인증서를 설정해 주어야 한다.

## **3.19 EnvelopFile**

STDMETHOD(EnvelopFile)(/\*[in]\*/BSTR btDataFile, /\*[in]\*/BSTR btEnvelopedDataFile, /\*[out, retval]\*/short \*pVal);

**기능**

데이터 파일을 읽어서 EnvelopedData 를 생성하여 지정된 경로에 저장한다.

**파라미터**

btDataFile (input)

Envelop 하고자 하는 데이터 파일의 절대 경로.

btEnvelopedDataFile (input)

EnvelopedData 를 저장하는 절대 경로.

pVal (output)

함수 실행시의 에러코드. 에러가 없으면 0 을 리턴 한다. 1 이상은 에러.

**연관된 프로퍼티(리턴값)**

**연관된 함수**

DecryptEnvelopedFile

SetPeerCertificate

**추가 설명**

EnvelopedData 를 생성하기 위해서는 비대칭암호화에 사용할 인증서를 설정해 주어야 한다.

## **3.20 GenerateRandomNumber**

STDMETHOD(GenerateRandomNumber)(/\*[in]\*/short nRandomNumberLen, /\*[out, retval]\*/short \*pVal);

**기능**

지정된 길이를 갖는 Random값을 생성한다.

**파라미터**

nRandomNumberLen (input)

Random 값의 길이를 지정한다.

pVal (output)

함수 실행시의 에러코드. 에러가 없으면 0 을 리턴 한다. 1 이상은 에러.

**연관된 프로퍼티(리턴값)**

OutData

함수 실행 후 생성된 Random 값을 가지고 있다.

**연관된 함수**

**추가 설명**

## **3.21 GenerateSymmetricDrivedKey (추후미지원)**

STDMETHOD(GenerateSymmetricDerivedKey)(/\*[in]\*/BSTR btPwd, /\*[out, retval]\*/short \*pVal);

**기능**

특정문자열이 입력되면 그에 해당하는 동일한 키를 생성한다.

**파라미터**

btPwd (intput)

키를 만드는데 사용되는 특정문자열.

pVal (output)

함수 실행시의 에러코드. 에러가 없으면 0 을 리턴 한다. 1 이상은 에러.

**연관된 프로퍼티(리턴값)**

OutData

**연관된 함수**

SetEncryptionAlgoAndMode

**추가 설명**

입력된 특정문자열로 이미 설정된 대칭키 알고리즘에 맞는 길이의 키를 생성한다. 이 함수의 경우 특정 사이트의 추후 지원을 위한 함수 이므로, 지정되지 않은 사이트의 경우, 사용하지 않도록 한다.

## **3.22 GenerateSymmetricKey**

STDMETHOD(GenerateSymmetricKey)(/\*[in]\*/BSTR btSeedStr, /\*[out, retval]\*/short \*pVal);

**기능**

대칭키를 생성한다.

**파라미터**

btSeedStr (intput)

키를 만드는데 사용되는 특정문자열.

pVal (output)

함수 실행시의 에러코드. 에러가 없으면 0 을 리턴 한다. 1 이상은 에러.

**연관된 프로퍼티(리턴값)**

OutData

**연관된 함수**

SetEncryptionAlgoAndMode

**추가 설명**

특정 문자열이 널일 경우에는 랜덤한 대칭키를 생성한다. 특정 문자열이 입력되는 경우에만 입력된 특정문자열로 이미 설정된 대칭키 알고리즘에 맞는 길이의 키를 생성한다.

현재 특정문자열 입력으로 키를 생성하는 기능은 미지원 중이다.

## **3.23 GenSignature**

STDMETHOD(GenSignature)(/\*[in]\*/short nDataType, /\*[in]\*/BSTR btData, /\*[out, retval]\*/short \*pVal);

**기능**

입력되는 데이터 타입에 따라 데이터 또는, 데이터파일의 경로를 받아서 PKCS#7 의 서명데이터 형태가 아닌 단지 서명값만을 생성한다.

**파라미터**

nDataType (intput)

btData 에 어떤 값이 들어오는지 나타낸다.

btData (intput)

nDataType 에 따라 데이터, 또는 데이터 파일의 절대 경로.

pVal (output)

함수 실행시의 에러코드. 에러가 없으면 0 을 리턴 한다. 1 이상은 에러.

**연관된 프로퍼티(리턴값)**

OutData

**연관된 함수**

VerifySignature

**추가 설명**

nDataType 이 0 인 경우에는 btData에 서명값을 생성할 데이터가 들어온다. 1 인 경우에는 데이터 파일의 절대 경로가 들어온다.

데이터 타입에 상관없이 생성된 서명값은 OutData 프로퍼티에 설정된다.

## **3.24 GetCertificate**

STDMETHOD(GetCertificate)(/\*[in]\*/short nCertType, /\*[in]\*/short nCertOutPutType, /\*[out, retval]\*/short \*pVal);

**기능**

메모리에 로드되어 있는 인증서 셋트에서 선택한 용도에 맞는 인증서를 지정된 형식으로 리턴 한다.

**파라미터**

nCertType (input)

꺼내고자하는 인증서의 용도.

nCertOutPutType (input)

인증서를 꺼낼 때의 인증서 형태.

pVal (output)

함수 실행시의 에러코드. 에러가 없으면 0 을 리턴 한다. 1 이상은 에러.

**연관된 프로퍼티(리턴값)**

OutData

**연관된 함수**

SelectCertificate

LoadCertificate

**추가 설명**

- 꺼내고자 하는 인증서의 용도

| 설정값 | 인증서 용도 |
| --- | --- |
| 1 | 서명용 인증서 |
| 2 | 키분배용 인증서 |

- 인증서를 꺼낼 때의 인증서 형태

| 설정값 | 인증서 용도 |
| --- | --- |
| 0 | PEM 형식 |
| 1 | Base64 Encoding |

## **3.25 GetCertificatePropertyFromID**

STDMETHOD(GetCertificatePropertyFromID)(/\*[in]\*/BSTR btCert, /\*[in]\*/short nPropertyID, /\*[out, retval]\*/short \*pVal);

**기능**

이미 읽어 들인 인증서 세트에서 지정된 타입의 인증서를 PEM 형식으로 리턴 한다.

**파라미터**

btCert (input)

분석하고자 하는 인증서

nPropertyID (input)

인증서 내에서 꺼내고자 하는 속성에 지정된 번호.

pVal (output)

함수 실행시의 에러코드. 에러가 없으면 0 을 리턴 한다. 1 이상은 에러.

**연관된 프로퍼티(리턴값)**

OutData

**연관된 함수**

SelectCertificate

**추가 설명**

분석하고자 하는 인증서는 PEM 형태만을 지원한다.

인증서 일련번호의 경우, OutData 프로퍼티를 통해서는 16진수형태의 문자열이 리턴되고,

OutDataNum 프로퍼티로 10진수가 리턴된다.

- 인증서 속성에 지정된 번호

| ID | Property 명 |
| --- | --- |
| 1 | 인증서 버전 (현재 3 이 정상임) |
| 2 | 일련번호 |
| 3 | 서명 알고리즘 |
| 4 | 발급자 DN |
| 5 | 서명자 DN |
| 6 | 서명자 공개키 알고리즘 |
| 7 | 유효기간 시작 (YYYY-MM-DD HH:MM:SS) |
| 8 | 유효기간 끝(YYYY-MM-DD HH:MM:SS) |
| 9 | 공개키 |
| 10 | 서명값(Signature) |
| 11 | 키 사용용도(Key Usage) |
| 12 | 기관 키 식별자 |
| 13 | 주체 키 식별자 |
| 14 | 확장 키 사용 용도 |
| 15 | 주체 대체이름 |
| 16 | Basic Constraint |
| 17 | 인증서 정책 |
| 18 | CRL DP |
| 19 | 기관정보 액세스 (Authority Info Address) |
| 20 | 유효기간 전체  (YYYY-MM-DD HH:MM:SS ~ YYYY-MM-DD HH:MM-SS) |

## **3.26 GetDataFromLDAP**

STDMETHOD(GetDataFromLDAP)(/\*[in]\*/BSTR btURL, /\*[in]\*/short nDataType, /\*[out, retval]\*/short \*pVal);

**기능**

입력된 DN 값을 분석하여 연결할 LDAP 을 찾고, 입력된 데이터 타입에 맞는 데이터를 LDAP 에서 읽어서 리턴한다.

**파라미터**

btURL (input)

가져오고자 하는 데이타의 DN

nDataType (input)

가져오고자 하는 데이터의 타입.

pVal (output)

함수 실행시의 에러코드. 에러가 없으면 0 을 리턴 한다. 1 이상은 에러.

**연관된 프로퍼티(리턴값)**

OutData

**연관된 함수**

SetConfig

**추가 설명**

SetConfig 함수에서 설정되는 LDAPInfo 에 LDAP 정보를 가지고 있어야 한다. btURL 파라미터에는 DN 값을 설정한다.

- 데이터 타입 지정 번호

| 데이터 타입 | 리턴값 |
| --- | --- |
| 1 | btURL 에 해당하는 첫번째 인증서 |
| 2 | btURL 에 해당하는 서명용 인증서 |
| 3 | btURL 에 해당하는 암호용 인증서 |
| 4 | btURL 에 해당하는 CRL |
| 5 | btURL 에 해당하는 ARL |

## **3.27 GetEncryptionAlgorithm**

STDMETHOD(GetEncryptionAlgorithm)(/\*[out, retval]\*/short \*pVal);

**기능**

API 에 설정된 대칭키 알고리즘 을 리턴한다.

**파라미터**

pVal (output)

함수 실행시의 에러코드. 에러가 없으면 0 을 리턴 한다. 1 이상은 에러.

**연관된 프로퍼티(리턴값)**

OutDataNum

**연관된 함수**

SetEncryptionAlgoAndMode

**추가 설명**

SetEncryptionAlgoAndMode 함수로 알고리즘을 설정하지 않을 경우에는 기본으로 설정된 Triple-DES 를 리턴한다.

- 대칭키 알고리즘

| 리턴값 | 알고리즘 |
| --- | --- |
| 1 | DES |
| 2 | **Triple-DES (기본)** |
| 3 | SEED |

## **3.28 GetEncryptionMode**

STDMETHOD(GetEncryptionMode)(/\*[out, retval]\*/short \*pVal);

**기능**

API 에 설정된 대칭키 알고리즘 의 Mode 를 리턴한다.

**파라미터**

pVal (output)

함수 실행시의 에러코드. 에러가 없으면 0 을 리턴 한다. 1 이상은 에러.

**연관된 프로퍼티(리턴값)**

OutDataNum

**연관된 함수**

SetEncryptionAlgoAndMode

**추가 설명**

SetEncryptionAlgoAndMode 함수로 모드를 설정하지 않을 경우에는 기본으로 설정된 CFB 모드를 리턴한다.

- 대칭키 알고리즘 모드

| 리턴값 | 알고리즘 모드 |
| --- | --- |
| 1 | ECB Mode |
| 2 | **CBC Mode (기본)** |
| 3 | CFB Mode |
| 4 | OFB Mode |

## **3.29 GetLoginData**

STDMETHOD(GetLoginData)(/\*[in]\*/short nLoginDataType, /\*[out, retval]\*/short \*pVal);

**기능**

로그인 데이터를 분석한 후 메모리에 로드 되어 있는 로그인 데이터를 꺼낸다.

**파라미터**

nLoginDataType (input)

꺼내고자 하는 로그인 데이터의 번호.

pVal (output)

함수 실행시의 에러코드. 에러가 없으면 0 을 리턴 한다. 1 이상은 에러.

**연관된 프로퍼티(리턴값)**

OutData

**연관된 함수**

LoginDataProcess

**추가 설명**

LoginDataProcess 함수 호출 후 분석된 로그인 데이터에서 특정 데이터를 꺼낸다.

- 꺼내고자 하는 로그인 데이터의 번호

| 설정값 | 로그인 데이터의 번호 |
| --- | --- |
| 1 | SessionID |
| 2 | 신원확인 식별자 |
| 3 | 사용자 정보 |

## **3.30 GetPrivateKey**

STDMETHOD(GetPrivateKey)(/\*[in]\*/short nKeyType, /\*[in]\*/short nKeyOutPutType, /\*[in]\*/BSTR btPwd, /\*[out, retval]\*/short \*pVal);

**기능**

메모리에 로드되어 있는 인증서 셋트에서 선택한 용도에 맞는 개인키를 지정된 형식으로 리턴 한다. 암호가 입력되는 경우에는 입력된 암호로 암호화되어서 리턴된다.

**파라미터**

nKeyType (input)

꺼내고자하는 개인키의 용도.

nKeyOutPutType (input)

개인키를 꺼낼 때의 개인키 형태.

btPwd (input)

개인키를 꺼낼 때 이 파라미터에 값이 있으면 입력된 값으로 암호화된 개인키를 리턴한다. “” 일 경우에는 기존의 암호로 암호화되서 리턴된다.

pVal (output)

함수 실행시의 에러코드. 에러가 없으면 0 을 리턴 한다. 1 이상은 에러.

**연관된 프로퍼티(리턴값)**

OutData

**연관된 함수**

SelectCertificate

LoadCertificate

**추가 설명**

- 꺼내고자 하는 개인키 의 용도

| 설정값 | 개인키 용도 |
| --- | --- |
| 1 | 서명용 개인키 |
| 2 | 키분배용 개인키 |

- 개인키를 꺼낼 때의 개인키 형태

| 설정값 | 인증서 용도 |
| --- | --- |
| 0 | PEM 형식 |
| 1 | Base64 Encoding |

## **3.31 GetSignerCert**

STDMETHOD(GetSignerCert)(/\*[in]\*/short nSignerIndex, /\*[out, retval]\*/short \*pVal);

**기능**

입력된 서명순서에 해당하는 서명자의 인증서를 PEM 형식으로 리턴한다.

**파라미터**

nSignerIndex (input)

가져오고자 하는 서명순서.

pVal (output)

함수 실행시의 에러코드. 에러가 없으면 0 을 리턴 한다. 1 이상은 에러.

**연관된 프로퍼티(리턴값)**

OutData

**연관된 함수**

VerifySignedData

VerifySignedFIle

GetSignerCount

GetSignerDN

GetSigningTime

**추가 설명**

nSignerIndex 는 1부터 시작된다. 총 서명수는 GetSignerCount 로 얻을 수 있다.

## **3.32 GetSignerCount**

STDMETHOD(GetSignerCount)(/\*[out, retval]\*/short \*pVal);

**기능**

서명검증 후 가장 최근에 이루어진 서명데이타의 서명 개수를 리턴한다.

**파라미터**

pVal (output)

함수 실행시의 에러코드. 에러가 없으면 0 을 리턴 한다. 1 이상은 에러.

**연관된 프로퍼티(리턴값)**

OutDataNum

**연관된 함수**

VerifySignedData

VerifySignedFIle

GetSignerCert

GetSignerDN

GetSigningTime

**추가 설명**

## **3.33 GetSignerDN**

STDMETHOD(GetSignerDN)(/\*[in]\*/short nSignerIndex, /\*[out, retval]\*/short \*pVal);

**기능**

입력된 서명순서에 해당하는 서명자의 인증서의 DN을 리턴한다.

**파라미터**

nSignerIndex (input)

가져오고자 하는 서명순서.

pVal (output)

함수 실행시의 에러코드. 에러가 없으면 0 을 리턴 한다. 1 이상은 에러.

**연관된 프로퍼티(리턴값)**

OutData

**연관된 함수**

VerifySignedData

VerifySignedFIle

GetSignerCount

GetSignerCert

GetSigningTime

**추가 설명**

nSignerIndex 는 1부터 시작된다. 총 서명수는 GetSignerCount 로 얻을 수 있다.

## **3.34 GetSigningTime**

STDMETHOD(GetSigningTime)(/\*[in]\*/short nSignerIndex, /\*[out, retval]\*/short \*pVal);

**기능**

입력된 서명순서에 해당하는 서명자의 인증서의 DN을 리턴한다.

**파라미터**

nSignerIndex (input)

가져오고자 하는 서명순서.

pVal (output)

함수 실행시의 에러코드. 에러가 없으면 0 을 리턴 한다. 1 이상은 에러.

**연관된 프로퍼티(리턴값)**

OutData

**연관된 함수**

VerifySignedData

VerifySignedFIle

GetSignerCount

GetSignerCert

GetSigningTime

**추가 설명**

nSignerIndex 는 1부터 시작된다. 총 서명수는 GetSignerCount 로 얻을 수 있다. 해당하는 서명정보에 서명시간이 설정되어 있지 않은 경우에는 ‘ 0 ‘ 을 리턴한다. 서명시간은 LocalTime 을 기준으로 리턴한다.

## **3.35 GetSymmetricIV**

STDMETHOD(GetSymmetricIV)(/\*[out, retval]\*/short \*pVal);

**기능**

생성 또는 EnvelopedData 에서 꺼낸 대칭키의 IV 값을 리턴한다.

**파라미터**

pVal (output)

함수 실행시의 에러코드. 에러가 없으면 0 을 리턴 한다. 1 이상은 에러.

**연관된 프로퍼티(리턴값)**

OutData

**연관된 함수**

GetSymmetricKey

SetSymmetricKeyAndIV

GenerateSymmetricKey

DecryptEnvelopedData

DecryptEnvelopedFile

**추가 설명**

## **3.36 GetSymmetricKey**

STDMETHOD(GetSymmetricKey)(/\*[out, retval]\*/short \*pVal);

**기능**

생성 또는 EnvelopedData 에서 꺼낸 대칭키의 Key 값을 리턴한다.

**파라미터**

pVal (output)

함수 실행시의 에러코드. 에러가 없으면 0 을 리턴 한다. 1 이상은 에러.

**연관된 프로퍼티(리턴값)**

OutData

**연관된 함수**

GetSymmetricKey

SetSymmetricKeyAndIV

GenerateSymmetricKey

DecryptEnvelopedData

DecryptEnvelopedFile

**추가 설명**

## **3.37 HashData**

STDMETHOD(HashData)(/\*[in]\*/short nHashAlgoritm, /\*[in]\*/BSTR btHash, /\*[out, retval]\*/short \*pVal);

**기능**

일반 데이터를 읽어서 지정된 Hash 알고리즘으로 Hash 값을 생성한다.

**파라미터**

nHashAlgorithmID (input)

Hash 알고리즘.

btHash (input)

Hash 값을 생성하고자 하는 데이터.

pVal (output)

함수 실행시의 에러코드. 에러가 없으면 0 을 리턴 한다. 1 이상은 에러.

**연관된 프로퍼티(리턴값)**

OutData

**연관된 함수**

**추가 설명**

- 사용 가능한 Hash 알고리즘

| 설정값 | Hash 알고리즘 |
| --- | --- |
| 1 | MD5 |
| 2 | RIPEMD160 |
| 3 | **SHA1 (기본)** |
| 4 | HAS160 |

## **3.38 HashFile**

STDMETHOD(HashFile)(/\*[in]\*/short nHashAlgoritm, /\*[in]\*/BSTR btDataFile, /\*[out, retval]\*/short \*pVal);

**기능**

일반 데이터 파일을 읽어서 지정된 Hash 알고리즘으로 Hash 값을 생성한다.

**파라미터**

nHashAlgorithmID (input)

Hash 알고리즘.

btDataFile (input)

Hash 값을 생성하고자 하는 데이터 파일의 절대 경로

pVal (output)

함수 실행시의 에러코드. 에러가 없으면 0 을 리턴 한다. 1 이상은 에러.

**연관된 프로퍼티(리턴값)**

OutData

**연관된 함수**

**추가 설명**

- 사용 가능한 Hash 알고리즘

| 설정값 | Hash 알고리즘 |
| --- | --- |
| 1 | MD5 |
| 2 | RIPEMD160 |
| 3 | **SHA1 (기본)** |
| 4 | HAS160 |

## **3.39 LoadCertificate**

STDMETHOD(LoadCertificate)(/\*[in]\*/int nType, /\*[in]\*/BSTR btSignCert, /\*[in]\*/BSTR btSignKey, /\*[in]\*/BSTR btKMCert, /\*[in]\*/BSTR btKMKey, /\*[in]\*/BSTR btPassword, /\*[out, retval]\*/short \*pVal);

**기능**

PEM 형식의 문자열들을 읽어서 인증서를 로딩하거나, 각 파일의 절대 경로를 받아서 인증서를 로딩한다.

**파라미터**

nType (input)

btSignCert 부터 btKMKey 까지 PEM형식의 인증서 인지, 파일 경로인지 나타낸다.

btSignCert (input)

PEM형식의 SignCert, 또는 SignCert 파일의 절대 경로

btSignKey (input)

PEM형식의 SignKey, 또는 SignKey 파일의 절대 경로

btKMCert (input)

PEM형식의 KMCert, 또는 SignCert 파일의 절대 경로

btKMKey (input)

PEM형식의 KMKey, 또는 KMKey 파일의 절대 경로

btPassword (input)

설정되는 인증서의 암호

pVal (output)

함수 실행시의 에러코드. 에러가 없으면 0 을 리턴 한다. 1 이상은 에러.

**연관된 프로퍼티(리턴값)**

**연관된 함수**

SelectCertificate

GetCertificate

GetPrivateKey

**추가 설명**

nType 이 0 인 경우에는 PEM 형식의 인증서를 설정해야 한다. nType 이 1인 경우에는 인증서 파일들의 절대 경로를 입력한다. 모든 개인키들은 인증서 암호로 암호화된 상태여야 한다.

이 함수로 로딩된 인증서와 개인키는 GetCertificate, GetPrivateKey 함수를 이용하여 꺼낸 수 있다.

## **3.40 LoginData**

STDMETHOD(LoginData)(/\*[in]\*/BSTR btSessionID, /\*[in]\*/BSTR btSSN, /\*[in]\*/BSTR btUserInfo, /\*[out, retval]\*/short \*pVal);

**기능**

로그인에 필요한 정보들을 이미 설정된 인증서로 서명 및 암호화(Envelop)한다.

**파라미터**

btSessionID (input)

서버에서 받아온 임의의 값.

btSSN (input)

이미 설정된 인증서의 신원확인 식별자.

btUserInfo (input)

서버로 전달하고자 하는 내용.

pVal (output)

함수 실행시의 에러코드. 에러가 없으면 0 을 리턴 한다. 1 이상은 에러.

**연관된 프로퍼티(리턴값)**

OutData

**연관된 함수**

LoginDataProcess

SelectCertificate

LoadeCertificate

SetPeerCertificate

**추가 설명**

서버에서 인증서 신원확인에 사용되는 btSSN 을 알지 못할 경우에는 0 으로 설정하면 서버에서 신원확인이 이루어지지 않는다.

함수 실행에 성공하면 OutData 프로퍼티에 로그인데이타가 설정된다.

## **3.41 LoginDataProcess**

STDMETHOD(LoginDataProcess)(/\*[in]\*/BSTR btLoginData, /\*[out, retval]\*/short \*pVal);

**기능**

로그인 데이터를 읽어서 분석한다. 로그인 데이터에 신원확인 식별자 값이 설정되어 있는 경우에는 로그인 데이터를 보낸 서명자의 인증서와 함께 신원확인을 수행한다.

**파라미터**

btLoginData (input)

분석하고자 하는 로그인 데이터.

pVal (output)

함수 실행시의 에러코드. 에러가 없으면 0 을 리턴 한다. 1 이상은 에러.

**연관된 프로퍼티(리턴값)**

**연관된 함수**

LoginData

SelectCertificate

LoadCertificate

GetLoginData

ClearMemory

**추가 설명**

인증서 신원확인에 사용되는 신원확인 식별자 값이 “ 0 “ 으로 설정되었으면 서버에서 신원확인이 이루어지지 않는다. 함수 호출 이후 VerifyVID 함수를 호출하여 진행 할 수 있다. 단, 메모리를 정리하는 ClearMemory 함수 호출 전에 이루어져야 한다.

로그인 데이터에 들어있는 정보들은 GetLoginData 함수를 통해서 하나씩 얻을 수 있다. LoginDataProcess 함수 호출 후 필요한 데이터들을 모두 꺼내었으면, ClearMemory 함수를 호출하여 내부메모리를 정리한다.

## **3.42 SelectCertificate**

STDMETHOD(SelectCertificate)(/\*[in]\*/short nBaseStorageType, /\*[in]\*/short nSelectType, /\*[in]\*/BSTR btUserInfo, /\*[out, retval]\*/short \*pVal);

**기능**

UI 를 이용하여 로딩하고자 하는 인증서를 선택하고, 인증서 암호를 입력 받아서 선택한 인증서를 메모리에 로딩한다.

**파라미터**

nBaseStorageType (input)

인증서 선택시 기본으로 읽어들일 저장매체

nSelectType (input)

인증서를 읽는 방식 선택.

btUserInfo (input)

UI 와 같이 읽어 들일 사용자 인증서의 DN 또는 UI 없이 읽어 들일 인증서 암호.

pVal (output)

함수 실행시의 에러코드. 에러가 없으면 0 을 리턴 한다. 1 이상은 에러.

**연관된 프로퍼티(리턴값)**

**연관된 함수**

SetInitConfig

LoadCertificate

GetCertificate

GetPrivateKey

**추가 설명**

- 인증서를 읽는 방식이 0 이면

기본적인 UI 를 사용하여 인증서 선택 및 암호를 입력 받는다. 단, btUserInfo 에 인증서 DN 입력시 해당하는 인증서 파일의 여부를 찾아서 UI 에 보여주고, 인증서 암호를 입력받는다.

- 인증서를 읽는 방식이 1 이면.

btUserInfo 에 입력된 인증서 암호로 TradeSign 에서 발급한 인증서중에서 동일한 암호를 사용하는 인증서를 찾아서 메모리에 로딩한다. 이 때 매치되는 인증서가 하나인 경우에는 UI 입력 없이 인증서가 메모리에 로드된다. 매치되는 인증서가 여려개일 경우에는 UI 에 목록을 출력하여 사용할 인증서를 선택하도록 한다.

- 인증서를 읽는 방식이 2 이면.

btUserInfo 에 “인증서DN**|**인증서암호” 형식으로 구성되어 입력된 인증서DN과 인증서 암호로 하드디스크에 위치한 인증서중에서 동일한 DN과 암호를 사용하는 인증서를 찾아서 메모리에 로딩한다.

- 저장매체관련 설정값.

| 설정값 | 저장매체 |
| --- | --- |
| 0 | HARD\_DISK |
| 1 | REMOVABLE\_DISK |
| 2 | IC\_CARD |
| 3 | PKCS11 |

## **3.43 SetEncryptionAlgoAndMode**

STDMETHOD(SetEncryptionAlgoAndMode)(/\*[in]\*/short nEncryptionAlgoritm, /\*[in]\*/short nEncryptionMode, /\*[out, retval]\*/short \*pVal);

**기능**

Encrypt / Decrypt , Envelop 처리시에 사용되는 대칭키 알고리즘을 설정한다.

**파라미터**

nBaseStorageType (input)

인증서 선택시 기본으로 읽어들일 저장매체

nSelectType (input)

인증서를 읽는 방식 선택.

btUserInfo (input)

UI 와 같이 읽어 들일 사용자 인증서의 DN 또는 UI 없이 읽어 들일 인증서 암호.

pVal (output)

함수 실행시의 에러코드. 에러가 없으면 0 을 리턴 한다. 1 이상은 에러.

**연관된 프로퍼티(리턴값)**

**연관된 함수**

GetEncryptionAlgorithm

GetEncryptionMode

EncryptData

DecryptData

EncryptFile

DecryptFile

EnvelopData

EnvelopFile

DecryptEnvelopedData

DecryptEnvelopedFile

**추가 설명**

이 함수로 사용하고자 하는 대칭키 알고리즘과 모드를 명시적으로 설정하지 않을 경우에는 기본값인 Triple-DES, CBC 모드로 처리된다.

- 대칭키 알고리즘 - 대칭키 알고리즘 모드

| 리턴값 | 알고리즘 |
| --- | --- |
| 1 | DES |
| 2 | **Triple-DES (기본)** |
| 3 | SEED |
| 리턴값 | 알고리즘 모드 |
| 1 | ECB Mode |
| 2 | **CBC Mode (기본)** |
| 3 | CFB Mode |
| 4 | OFB Mode |

## **3.44 SetPeerCertificate**

STDMETHOD(SetPeerCertificate)(/\*[in]\*/BSTR btCert, /\*[out, retval]\*/short \*pVal);

**기능**

Envelop 하기 전에 Envelop 에 사용할 상대방 인증서를 설정한다.

**파라미터**

btCert (input)

Envelop 에 사용할 상대방 인증서. PEM 또는 Base64 형태의 문자열로 처리한다.

pVal (output)

함수 실행시의 에러코드. 에러가 없으면 0 을 리턴 한다. 1 이상은 에러.

**연관된 프로퍼티(리턴값)**

**연관된 함수**

EnvelopData

EnvelopFile

**추가 설명**

btCert 는 PEM 형식 또는 Base64 형태의 문자열이 입력되어야 한다.

## **3.45 SetSymmetricKey**

STDMETHOD(SetSymmetricKeyAndIV)(/\*[in]\*/BSTR btSymmetricKey, /\*[in]\*/BSTR btSymmetricIV, /\*[out, retval]\*/short \*pVal);

**기능**

외부에서 API 내부로 대칭키의 Key와 IV 값을 설정한다.

**파라미터**

btSymmetricKey (input)

Envelop 또는 Encrypt 에 사용할 대칭키의 Key를 설정한다.

btSymmetricIV (input)

Envelop 또는 Encrypt 에 사용할 대칭키의 IV를 설정한다.

pVal (output)

함수 실행시의 에러코드. 에러가 없으면 0 을 리턴 한다. 1 이상은 에러.

**연관된 프로퍼티(리턴값)**

**연관된 함수**

GetSymmetricKey

GetSymmetricIV

DecryptData

DecryptFile

EncryptData

EncryptFile

EnvelopData

EnvelopFile

**추가 설명**

## **3.46 SignData**

STDMETHOD(SignData)(/\*[in]\*/BSTR btData, /\*[out, retval]\*/short \*pVal);

**기능**

이미 설정된f명값을 생성한다.

**파라미터**

btData (input)

서명하고자 하는 데이터.

pVal (output)

함수 실행시의 에러코드. 에러가 없으면 0 을 리턴 한다. 1 이상은 에러.

**연관된 프로퍼티(리턴값)**

OutData

**연관된 함수**

SetConfig

SelectCertificate

LoadCertificate

SignFile

VerifySignedData

**추가 설명**

서명값 생성에 필요한 조건들은 SetConfig 함수의 nIncludeCertPath, nIncludeSigningTime, nIncludeCRL, nIncludeCRL 파라미터들에서 결정한다.

## **3.47 SignFile**

STDMETHOD(SignFile)(/\*[in]\*/BSTR btDataFile, /\*[in]\*/BSTR btSignedDataFile, /\*[out, retval]\*/short \*pVal);

**기능**

이미 설정된 인증서로 입력된 데이터 파일을 읽어서 PKCS#7 형식의 서명값을 생성하여 입력된 파일 경로에 저장한다.

**파라미터**

btDataFile (input)

서명하고자 하는 데이터 파일의 절대 경로.

btSignedDataFile (input)

서명값을 저장할 파일의 절대 경로.

pVal (output)

함수 실행시의 에러코드. 에러가 없으면 0 을 리턴 한다. 1 이상은 에러.

**연관된 프로퍼티(리턴값)**

**연관된 함수**

SetConfig

SelectCertificate

LoadCertificate

SignData

VerifySignedFile

**추가 설명**

서명값 생성에 필요한 조건들은 SetConfig 함수의 nIncludeCertPath, nIncludeSigningTime, nIncludeCRL, nIncludeCRL 파라미터들에서 결정한다.

## **3.48 VerifySignature**

STDMETHOD(VerifySignature)(/\*[in]\*/short nDataType, /\*[in]\*/BSTR btData, /\*[in]\*/BSTR btSignature, /\*[in]\*/BSTR btCert, /\*[out, retval]\*/short \*pVal);

**기능**

원문데이터, 서명값, 서명자 인증서 등을 읽어서 서명값을 검증한다.

**파라미터**

nDataType (intput)

btData 에 어떤 값이 들어오는지 나타낸다.

btData (intput)

nDataType 에 따라 서명값을 검증하기 위한 원문 데이터, 또는 서명값을 검증하기 위한 원문 데이터 파일의 절대 경로.

btSignedData (input)

검증하고자 하는 서명값

btCert (input)

PEM 형식의 서명자의 인증서

pVal (output)

함수 실행시의 에러코드. 에러가 없으면 0 을 리턴 한다. 1 이상은 에러.

**연관된 프로퍼티(리턴값)**

OutData

**연관된 함수**

GenSignature

**추가 설명**

nDataType 이 0 인 경우에는 btData에 서명값을 검증하기 위한 원문 데이터가 들어온다. 1 인 경우에는 서명값을 검증하기 위한 원문 데이터 파일의 절대 경로가 들어온다.

## **3.49 VerifySignedData**

STDMETHOD(VerifySignedData)(/\*[in]\*/BSTR btSignedData, /\*[in]\*/BSTR btContentData, /\*[out, retval]\*/short \*pVal);

**기능**

PKCS#7 형식의 서명데이터를 읽어서 검증한 후 원문데이터를 리턴한다.

**파라미터**

btSignedData (input)

PKCS#7 형태의 서명데이터.

btContentData (input)

원문 데이터가 포함되지 않은 서명데이터를 검증할 때 사용할 데이터.

pVal (output)

함수 실행시의 에러코드. 에러가 없으면 0 을 리턴 한다. 1 이상은 에러.

**연관된 프로퍼티(리턴값)**

OutData

**연관된 함수**

SetConfig

SelectCertificate / LoadCertificate

SignFile

VerifySignedData

GetSignerCount

GetSignerCert

GetSignerDN

GetSigningTime

ClearMemory

CertificationValidation

**추가 설명**

원문 데이터를 포함한 서명데이터 검증 후 나온 원문데이터는 OutData 프로퍼티에 저장된다. 원문 데이터를 포함하지 않는 서명데이터 검증 시에는 원문 데이터가 OutData 프로퍼티에 저장되지 않는다.

서명데이터 검증 후 서명관련 정보를 얻을 때는, GetSignerCount, GetSignerCert, GetSignerDN, GetSigningTime 함수를 이용하고, 정보를 다 얻은 후에는 ClearMemory 함수를 호출하여 메모리에 로딩되어 있는 서명관련 정보들을 삭제한다.

이 함수 호출후에는 서명자의 인증서를 얻어서 인증서 검증을 따로 진행한다.

## **3.50 VerifySignedFile**

STDMETHOD(VerifySignedFile)(/\*[in]\*/BSTR btSignedDataFile, /\*[in]\*/BSTR btDataFile, /\*[in]\*/BSTR btContentFile, /\*[out, retval]\*/short \*pVal);

**기능**

PKCS#7 형식의 서명데이터 파일을 읽어서 검증한 후 원문데이터를 입력된 경로에 저장한다.

**파라미터**

btSignedDataFile (input)

PKCS#7 형태의 서명데이터 파일의 절대 경로.

btDataFile (input)

원문 데이터가 포함된 서명데이터를 검증 후 나온 원문데이터를 저장할 절대 경로.

btContentFile (input)

원문 데이터가 포함되지 않은 서명데이터를 검증할 때 사용할 데이터 파일의 절대 경로.

pVal (output)

함수 실행시의 에러코드. 에러가 없으면 0 을 리턴 한다. 1 이상은 에러.

**연관된 프로퍼티(리턴값)**

**연관된 함수**

SetConfig

SelectCertificate / LoadCertificate

SignFile

VerifySignedData

GetSignerCount

GetSignerCert

GetSignerDN

GetSigningTime

ClearMemory

CertificationValidation

**추가 설명**

원문 데이터를 포함한 서명데이터 검증 후 나온 원문데이터는 OutData 프로퍼티에 저장된다. 원문 데이터를 포함하지 않는 서명데이터 검증 시에는 원문데이터가 OutData 프로퍼티에 저장되지 않는다.

서명데이터 검증 후 서명관련 정보를 얻을 때는, GetSignerCount, GetSignerCert, GetSignerDN, GetSigningTime 함수를 이용하고, 정보를 다 얻은 후에는 ClearMemory 함수를 호출하여 메모리에 로딩되어 있는 서명관련 정보들을 삭제한다.

이 함수 호출 후에는 서명자의 인증서를 얻어서 인증서 검증을 따로 진행한다.

## **3.51 VerifyVID**

STDMETHOD(VerifyVID)(/\*[in]\*/BSTR btSSN, /\*[out, retval]\*/short \*pVal);

**기능**

이미 로딩되어 있는 인증서 셋트나 로그인데이타 처리시 얻은 랜덤값과 인증서로 입력된 신원확인 식별자 값과 신원확인을 진행한다.

**파라미터**

btSSN (input)

검증하고자 하는 인증서의 신원확인 식별자.

pVal (output)

함수 실행시의 에러코드. 에러가 없으면 0 을 리턴 한다. 1 이상은 에러.

**연관된 프로퍼티(리턴값)**

OutData

**연관된 함수**

SelectCertificate / LoadCertificate

LoginDataProcess

**추가 설명**

신원확인 식별자와 인증서 간의 신원확인이 일치하면 OutData 프로퍼티에 “true” 값이 설정된다. 신원확인이 일치하지 않을 경우에는 “false” 값이 설정된다.

# **4. 에러 메시지**

툴킷에서 발생할 수 있는 에러메시지를 정리하였습니다.

| 에러  CODE | 에러메시지 | 에러 발생 예 | 대처방법(확인사항) |
| --- | --- | --- | --- |
| 100 | 인증서 선택이 취소되었습니다. | UI 에서 취소 버튼 누른 경우 | 자바 스크립트 에서 에러코드 "100" 인 경우 에러를 사용자 에게 보여주지 않도록 함 |
| 101 | 알맞지 않은 인증서 속성입니다. 현재 지원하지 않는 저장매체 입니다. 인증서 정보가 올바르지 않습니다. 지원하지 않는 알고리즘 입니다. 알맞지 않은 모드 입니다. 허용되 않는 데이터 타입입니다. 전체 서명수보다 입력하신 서명정보 Index가 더 큽니다. 알맞지 않은 키 종류입니다. 알맞지 않은 인증서 정책 모음 입니다. 알맞지 않은 인증서 종류입니다. 설정된 인증서가 올바르지 않습니다. 현재 지원하지 않는 저장매체 입니다. 알 수 없는 데이타 타입입니다. 로그인 메시지 형식이 맞지 않습니다. 지원되지 않는 HASH 알고리즘 입니다. 사용되지 않는 환경설정 입니다. | 상황에 맞지 않는 입력값이 있을 경우. (예: 범위를 벗어난 입력값) | Config에서 설정한 값이 정확한지 재확인 필요 |
| 102 | 암호화할 데이터가 없습니다. 암호화에 사용할 대칭키가 없습니다. 복호화할 데이터가 없습니다. 복호화에 사용할 대칭키가 없습니다. 암호화할 파일의 위치가 없습니다. 암호화된 내용을 저장할 위치가 없습니다. 복호화된 내용을 저장할 위치가 없습니다. 복호화에 사용할 대칭키가 없습니다. Base64Encoding할 데이터가 없습니다. Base64Encoding할 파일의 위치가 없습니다. Base64Decoding된 내용을 저장할 위치가 없습니다. Base64Decoding할 파일의 위치가 없습니다. Hash할 데이터가 설정되어 있지 않습니다. Hash할 파일의 위치가 없습니다. 서명할 데이터가 없습니다. 검증할 서명된 데이터가 없습니다. 서명할 파일의 위치가 없습니다. .. 이외 여러가지 메시지가 있을 수 있음 | 필요한 입력값이 없을 경우 | 함수 사용하기 전 필요한 값 및 설정값 재확인 API 사용 예제를 보고 순서대로 빠진 함수가 없는지 확인 |
| 103 | Base64Encoding 에 실패하였습니다. | Base64 Encoding 에 실패한 경우 | ??? |
| 104 | Base64Decoding 에 실패하였습니다. | Base64 Decoding 에 실패한 경우. | EncodedData 입력시 좌우에 공백문자 들어가지 않도록 재확인 |
| 105 | 암호화할 파일 읽기에 실패하였습니다. 복호화할 파일 읽기에 실패하였습니다. Base64Encoding할 파일 읽기에 실패하였습니다. Base64Decoding할 파일 읽기에 실패하였습니다. 파일 읽기에 실패하였습니다. 서명된 파일 읽기에 실패하였습니다. 데이터 파일 읽기에 실패하였습니다. EnvelopedData 파일 읽기에 실패하였습니다. CA 인증서 모음 파일 읽기에 실패하였습니다. 서명용 인증서 파일이 없거나 파일 읽기에 실패하였습니다. 서명용 개인키 파일이 없거나 파일 읽기에 실패하였습니다. 암호용 인증서 파일 읽기에 실패하였습니다. 암호용 개인키 파일 읽기에 실패하였습니다. 서명용 개인키 파일이 없거나 파일 읽기에 실패하였습니다. 암호용 개인키 파일이 없거나 파일 읽기에 실패하였습니다. 데이터 파일 읽기에 실패하였습니다. | 특정 파일을 읽기에 실패한 경우 (파일이 없거나, 파일의 내용이 없는 경우) | 환경함수에 설정된 파일이 있는지 재확인 파일이 있더라도 size가 0byte가 안되도록 재확인 |
| 106 | 파일 저장에 실패하였습니다. Base64Encoding된 내용 저장에 실패하였습니다. Base64Decoding된 내용 저장에 실패하였습니다. 서명값 저장에 실패하였습니다. 원문 저장에 실패하였습니다. EnvelopedData를 저장에 실패하였습니다. 데이터 파일 저장에 실패하였습니다. 서명값 저장에 실패하였습니다. 개인키 저장에 실패하였습니다. | 특정 위치에 파일을 저장하는데 실패한 경우 (쓰기 금지, 공간부족, 부정확한 위치등 으로 발생함) | 저장될 곳의 권한설정 확인 저장할 수 있는 여유공간이 있는지 확인 환경설정화일에 설정한 디렉토리가 있는지 재확인 |
| 107 | 복호화된 개인키에 이상이 있습니다. | 읽어들인 데이터가 모자라거나 알맞지 않을 경우 (복호화된 개인키의 길이부족, 인증서 파일의 내용 부족 등 ) | 인증서의 파일 크기 및 위치 재확인 암호문의 크기 및 좌우 여백 확인 |
| 108 | 설정된 인증서와 개인키가 없습니다. 서명용 인증서나 개인키가 설정되어 있지 않습니다. | 사용 해야 할 서명용 인증서와 개인키가 설정되어 있지 않은 경우 | 설정화일에 설정된 서명용 인증서 및 개인키의 설정 재확인 |
| 109 | 설정된 인증서와 개인키가 없습니다. | 사용 해야 할 서명용 및 암호용 인증서와 개인키가 모두 설정되어 있지 않은 경우 | 설정화일에 설정된 인증서 및 개인키의 설정 및 파일 재확인 |
| 110 | 인증서 분석에 실패하였습니다. 서명자 인증서 분석에 실패하였습니다. 설정된 서명용 인증서 분석에 실패하였습니다. | 인증서 분석에 실패한 경우 | 인증서 파일이 제대로된 파일인지 확인 |
| 111 | 없음 | 개인키 복호화에 실패한 경우 (인증서 암호 분실, 대소문자 미구분, 한영키 미구분 등) | 개인키 복호화 실패 (인증서 암호분실, 대소문자 미구분, 한영키 미구분등) |
| 112 | 서명용 인증서와 암호용 인증서의 암호가 다릅니다. 두 인증서 암호를 맞춰주시기 바랍니다. | 서명용 개인키와 암호용 개인키의 암호가 틀린경우 (두 개인키의 암호를 동일하게 맞추어 주세요) | 서명용 개인키와 암호용 개인키의 암호가 틀린경우 (두 개인키의 암호를 동일하게 맞추어 주세요) |
| 113 | 인증서 정책 꺼내기에 실패하였습니다. | 인증서 정책 가져오기 실패 | 인증서의 사용용도를 확인하세요 |
| 114 | 인증서에 정책이 없습니다. | 인증서에 정책항목이 없습니다.  (올바른 인증서인지 확인 바랍니다.) | 올바른 인증서인지 확인 |
| 115 | 랜덤값이 없습니다. | 개인키에 있는 랜덤값이 존재하지 않을 경우 ( 개인키가 올바른지 확인 바랍니다.) | 개인키가 올바른파일인지 확인 |
| 116 | 개인키 분석에 실패하였습니다. | 개인키 분석에 실패한 경우 | 개인키 파일의 위치 및 환경설정파일 재확인 |
| 117 | 없음 | 없음 |  |
| 118 | 없음 | 없음 |  |
| 119 | 없음 | 없음 |  |
| 120 | 없음 | 없음 |  |
| 121 | LDAP 에서 데이터를 꺼내오는데 실패했습니다. | LDAP에서 데이터 가져오기 실패 | 인터넷이 되는지 확인 LDAP 주소가 맞는지 확인 |
| 122 | LDAP 조회에 실패하였습니다. | LDAP에서 파일(인증서,CRL,ARL등) 검색에 실패 | 인터넷이 되는지 확인 LDAP주소 및 검색 주소가 맞는지 확인 |
| 123 | 서버연결에 실패하였습니다.! | LDAP 연결에 실패 | 인터넷이 되는지 확인 LDAP 주소가 맞느지 확인 |
| 124 | UUTF8String::KSCToUTF8 ! | 한글처리를 위해 UTF8로 변환 실패 |  |
| 125 | DN에 인증기관 관련 O 값이 없습니다. | DN에서 인증기관 관련 정보("O=")가 없습니다. | 인증서의 DN 정보 확인 인증서 위치 및 환경설정 확인 |
| 126 | CA 정보가 제대로 설정되어 있지 않습니다. | ???? |  |
| 127 | CA의 LDAP 정보가 없습니다. | 인증서에 CA의 LDAP관련 정보가 없습니다. | 인증서의 LDAP 관련정보확인 인증서 위치 및 환경설정 확인 |
| 128 | CaPubs 생성에 필요한 CA 인증서 갯수가 부족합니다. CaPubs 생성에 실패하였습니다. CaPubs 정리에 실패하였습니다. CaPubs 꺼내기에 실패하였습니다. | CaPubs 파일에 담겨진 정보가 부족합니다.(???) | CaPubs 파일의 size 및 위치 재확인 |
| 129 | CaPubs 생성에 필요한 CA 인증서 갯수가 부족합니다. | CaPubs 생성에 필요한 CA 인증서 개수가 부족합니다. | CaPubs 파일 size 및 위치 재확인 |
| 130 | CaPubs 정리에 실패하였습니다. | CaPubs 안의 인증서 소팅 실패 | CaPubs 파일 size 및 위치 재확인 |
| 131 | CA 인증서 모음 파일분석에 실패하였습니다. | CaPubs 인증서 구조 파싱 실패 | CaPubs 파일 size 및 위치 재확인 |
| 132 | CaPubs 에 최상위 인증기관 인증서를 추가하는데 실패하였습니다. | CaPubs 에 최상위 인증기관 인증서 추가 실패 | CaPubs 파일 size 및 위치 재확인 |
| 133 | CaPubs 에 인증기관 인증서를 추가하는데 실패하였습니다. | CaPubs 에 인증기관 인증서 추가 실패 | CaPubs 파일 size 및 위치 재확인 |
| 134 | CA 인증서 모음에 인증서가 부족합니다. | CaPubs에 인증서가 부족함 | CaPubs 파일 size 및 위치 재확인 |
| 135 | CaPubs 분석에 실패하였습니다. | CaPubs 인증서 구조 파싱 실패 | CaPubs 파일 size 및 위치 재확인 |
| 136 | 인증서 경로 검증에 실패하였습니다. | 인증서 경로검증 실패 |  |
| 137 | 인증서 유효성 검증에 실패하였습니다. | 인증서 유효성 검증 실패 | 인증서 기간이 만료되었는지 확인 |
| 138 | 없음 | 없음 | 인증서 깨졌을 수 있음 -> 발급기관에 문의 |
| 139 | 허용된 인증서 정책이 아닙니다. | 허용되지 않은 인증서 정책 | 인증서의 DN 정보 확인 인증서 위치 및 환경설정 확인 |
| 140 | 로그인 사용자와 신원확인 정보가 일치하지 않습니다. | 신원확인 정보 불일치 | 주민번호 및 입력값 재확인 |
| 141 | 유효하지 않은 인증서입니다. | 인증서 유효기간 에러 | 인증서 기간이 만료되었는지 확인 |
| 142 | PKCS#7 구조 만들기에 실패하였습니다. | PKCS#7 구조 생성 에러 | 인증서 및 개인키 위치 및 환경설정화일 확인 |
| 143 | PKCS#7 구조 만들기에 실패하였습니다. | PKCS#7 구조 생성 에러 | 인증서 및 개인키 위치 및 환경설정화일 확인 |
| 144 | 알 수 없는 ContentType 입니다. | 알수없는 ContentType | 인증서 및 개인키 위치 및 환경설정화일 확인 |
| 145 |  | 없음 | 인증서 및 개인키 위치 및 환경설정화일 확인 |
| 146 | PKCS#7 구조 만들기에 실패하였습니다. | PKCS#7 구조 생성 에러 | 인증서 및 개인키 위치 및 환경설정화일 확인 |
| 147 | PKCS#7에서 Content 종류 꺼내기에 실패하였습니다. | PKCS#7 Content 종류 에러 | 인증서 및 개인키 위치 및 환경설정화일 확인 |
| 148 | PKCS#7 의 Content 가 EnvelopedData 가 아닙니다. | PKCS#7의 Content 가 EnvelopedData가 아님 | 인증서 및 개인키 위치 및 환경설정화일 확인 |
| 149 | PKCS#7에서 EnvelopedData 꺼내기에 실패하였습니다. | PKCS#7의 EnvelopedData 꺼내기 실패 | 인증서 및 개인키 위치 및 환경설정화일 확인 |
| 150 | 서명값 분석에 필요한 데이터를 설정하는데 실패하였습니다. Envelop할 데이타를 설정하는데 실패하였습니다. | Envelop할 데이터 설정 실패 | 환경설정 확인 |
| 151 | 만료된 인증서는 서명에 사용될 수 없습니다. | 만료된 인증서로 서명할수 없습니다. | 인증서 유효기간 확인 |
| 152 | 서명자의 인증서를 설정하는데 실패하였습니다. | ?? | 인증서 및 개인키 위치 및 환경설정화일 확인 |
| 153 | 서명자의 CRL을 설정하는데 실패하였습니다. | 서명자의 CRL 설정 실패 | 인터넷 연결 확인 및 환경화일에서 CRL cash 디렉토리가 있는지 확인 |
| 154 | 서명자의 상위 인증서를 설정하는데 실패하였습니다. | 서명자의 상위 인증서 설정 실패 | 환경화일에서 capubs 파일 재확인 |
| 155 | 서명값 생성에 실패하였습니다. | 서명값 생성 실패 | 서명값 생성에 실패했으므로 환경 및 입력정보 재확인 |
| 156 | 서명값을 분석하는데 실패하였습니다. | 서명값 분석 실패 | 입력된 서명값 확인 |
| 157 | 서명한 사람의 인증서가 하나도 없습니다. 서명한 사람의 인증서와 서명 정보의 갯수가 맞지 않습니다. | 서명자의 인증서가 없거나 서명자의 인증서와 서명자의 정보가 맞지 않음 | 서명자의 인증서가 없거나 서명정보의 개수(인증서 수)가 맞지 않으므로 서명값 재확인 |
| 158 | 서명정보가 하나도 없습니다. 허용된 서명 갯수보다 서명정보가 더 많습니다. | 서명정보가 없음. 허용된 서명 개수보다 서명정보가 더 많음 | 서명정보가 없음. 허용된 서명갯수보다 서명정보가 더 많으므로 서명값 재확인 |
| 159 | 서명정보에서 서명자 정보 꺼내기에 실패하였습니다. | 서명정보에서 서명자 정보 꺼내기 실패 | 입력된 서명값 재확인 |
| 160 |  | 없음 |  |
| 161 | 설정된 인증서가 없습니다. 수신자 인증서를 찾을 수 없습니다. | 수신자 인증서가 없음 | 서명값안의 수신자 인증서 정보가 없습니다. 서명값 재확인 |
| 162 | 수신자의 인증서를 설정하는데 실패하였습니다. | 수신자 인증서 설정 에러 | 수신자 인증서 설정에 실패. 입력 및 환경정보 재확인 |
| 163 | EnvelopedData를 만드는데 실패하였습니다. | EnvelopedData 생성 실패 | 전자봉투 생성실패 |
| 164 | EnvelopedData를 분석하는데 실패하였습니다. | EnvelopedData 분석 실패 | 전자봉투값 분석 실패. 전자봉투값 재확인 |
| 165 | 본인의 인증서와 개인키정보를 설정하는데 실패하였습니다. | 키관리용 인증서 설정 실패 | KMCert 인증서 설정 실패. 환경정보 재확인 |
| 166 | 본인의 인증서와 개인키정보를 설정하는데 실패하였습니다. | 본인 인증서와 개인키정보 설정 실패 | 인증서 및 개인키 설정 실패. 환경정보 재확인 |
| 167 | Content를 가지고 오는데 실패하였습니다. | Content 가져오기 실패 | 입력값 재확인 |
| 168 | 대칭키와 IV 를 꺼내는데 실패하엿습니다. | 대칭키와 IV값 가져오기 실패 | 복호화에 필요한 정보 가져오기에 실패. 설정값 재확인 |
| 169 | 랜덤값 생성에 실패하였습니다. | 랜덤값 생성 실패 | 랜덤값 생성에 실패 |
| 170 | 대칭키를 생성하는데 실패하였습니다. SecretKey 생성에 실패하였습니다. | 대칭키 생성 실패 | 대칭키 생성 실패. |
| 171 | SessionKey와 IV 가 존재하지 않습니다. 대칭키와 IV 가 존재하지 않습니다. | SessionKey와 IV가 존재하지 않음 | 세션키와 초기값이 없음. |
| 172 | 대칭키로 복호화하는데 실패하였습니다. | 대칭키 복호화 실패 | 복호화 실패. 대칭키 재확인 |
| 173 | 대칭키로 암호화하는데 실패하였습니다. | 대칭키 암호화 실패 | 암호화 실패. 대칭키 재확인 |
| 174 | Hash 값 생성에 실패하였습니다. | Hash값 생성 실패 | Hash 값 생성 실패 |
| 175 | Login 메시지 만들기에 실패하였습니다. | Login 메시지 만들기 실패 | 로그인 메시지 생성 실패. 입력값 재확인 |
| 176 | 로그인 메시지 분석에 실패하였습니다. | 로그인 메시지 분석 실패 | 로그인 메시지 분석 실패. 입력값 재확인 |
| 177 | 인터넷으로 인증기관 인증서 받기에 실패하였습니다. | 인터넷으로 인증기관 인증서 받기 실패 | 인터넷으로 인증기관 인증서 받기 실패. 인터넷 연결여부 재확인 |
| 178 | 사용자 인증서 무결성 검증에 실패했습니다. CA 인증서 무결성 검증에 실패했습니다. | 사용자 인증서 무결성 검증 실패 | 사용자 인증서 무결성 검증 실패. 사용자의 인증서 재확인 |