VestSign Manual

Version 1.3.5

`

관리본개정이력표

문서명	VestSign Package

버전	날짜	내 용	작성자
1.0.0	2014.10.06	최초작성	이정훈
1.1.0	2014.12.19	GS인증 매뉴얼	이정훈
1.1.1	2015.04.23	에러코드 추가	김이구
1.2.0	2016.02.01	API /오류코드 및 기능 추가	김이구
1.3.0	2016.05.19	Manual 수정	이정훈
1.3.1	2016.06.03	Client API 수정.	정찬
1.3.2	2016.09.28	Mac LINUX 설치 안내 추가	김이구
1.3.3	2021.05.28	마이데이터 추가	이정훈
1.3.4	2021.06.28	마이데이터 용 nonce 생성 기능 추가	이정훈
1.3.5	2021.07.02	마이데이터 원문 추출 함수 추가	이정훈

목 차

표 목 차

그림 목차

1. 개요

인터넷 환경은 Microsoft Windows의 Internet Explorer와 ActiveX를 기반한 제한적인 환경에서 다양한 OS와 브라우저를 지원하는 웹 표준 환경을 지원는 추세로 변하고 있다. 또한 모바일환경의 활성화에 따라 인터넷은 다양한 서비스 영역으로 급속하게 확장되고 있다. 다양한 서비스와 플랫폼을 지원하기 위해서 웹 표준에 대한 필요성은 크게 대두되었다.

다양한 영역에서 사용되는 공인인증서 솔루션은 대부분 ActiveX와 Plug-In 기반에서 동작한다. Active-X와 Plug-In을 사용하는 공인인증 솔루션은 급변하는 사용자의 인터넷 환경을 지원하기에 한계가 있다. 각 브라우저 제조사에서도 ActiveX/Plug-In 기술의 무분별한 남용, 웹표준 준수, 보안성을 이유로 제한적으로 지원하기 시작하였으며, 향후 지원을 중단할 것으로 예고하였다..

VestSign은 웹 표준을 준수하는 모든 OS와 브라우저를 지원하는 공인인증솔루션이다. 웹 표준 기술 기반의 VestSign 솔루션은 별도의 설치프로그램 없이 전자서명, 서명검증, 인증서 관리 기능을 제공한다.

1.1. 주요 기능

무결성	전자서명 검증을 통하여 네트워크 상에서 전달되는 전자서명 데이터의 위조 또는 변조 여부 확인.
부인방지	사용자의 전자서명을 통해 거래 행위에 대해 부인방지.
사용자 인증	전자서명 검증과 인증서 유효성 검증을 통해 사용자 인증.

<표 1> 주요 기능

1.1.1. 전자서명 생성/검증

서버 어플리케이션에서 사용자의 특정 행위에 대해 전자서명이 필요 할 경우, 사용자는 VestSign의 Client API를 통하여 사용자의 인증서와 개인키를 가지고 전자서명을 수행할 수 있다. 전자서명된 문서는 인터넷을 통하여 서버측에 전송된다. 서버는 전자 서명검증을 통하여 서명문의 진위여부를 확인하고, 서명문의 위변조 여부를 확인한다.

VestSign은 전자서명문 생성시 각 국제 표준과 국내 표준 규격을[1] 준수한다.

[1] 전자서명 규격: [RFC 2315] PKCS #7, [RFC 5652] CMS(Cryptographic Message Syntax), [KCAC.TS.DSIG]
인증서 규격: X.509, [RFC 7292] PKCS #12,
해쉬 알고리즘 규격: [KCAC.TS.HASH]

1.1.2. 인증서 식별번호 검증

국내 공인인증서는 본인확인을 위한 식별번호(주민등록번호/사업자등록번호)가 포함시켜 공인인증서를 발급 하고 있다. VestSign은 사용자의 개인키에 저장된 식별번호를 추출하여 서버의 인증서로 암호화하여 전송한다. 전송된 식별번호 메시지는 서버측에서 제공되는 API를 통하여 검증을 수행하여 본인확인을 수행한다.

VestSign은 식별번호 생성/검증시 국내 표준 규격[2]을 준수한다.

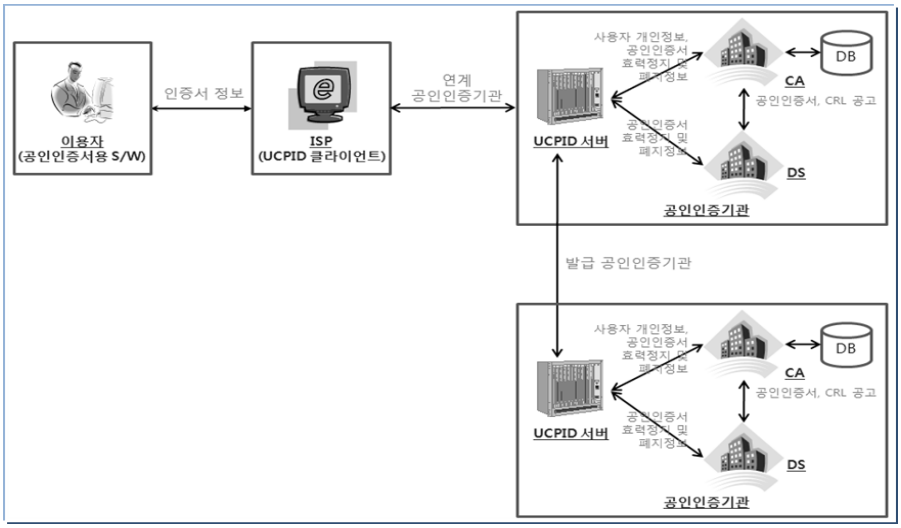
1.1.3. 인증서 유효성 검증

사용자가 생성한 전자서명문서를 검증한 후 서버는 사용자가 유효한 인증서를 이용하여 전자서명되었는지 검증이 필요하다. VestSign Server 모듈에서 제공하는 Server API를 통하여 인증서 유효성 검증을 수행한다. VestSign Server 모듈은 사용자의 인증서를 발급한 각 인증기관에 접속하여 유효성을 검증한다. 인증서 유효성 검증 방식은 아래와 같다.

- CRL : 각 인증기관의 LDAP 서버에 접속하여 인증서 폐기목록을 확인한다.
- OCSP : 각 기관에서 운영하는 OCSP Server에 접속하여 실시간 인증서 유효성을 확인한다.
- UCPID : 각 인증기관에서 운영하는 UCPID Server에 접속하여 유효성과 가입자 정보를 확인한다.

1.1.4. UCPID 검증

웹서비스 상에서 개인정보의 입력없이 공인인증서를 이용하여 본인확인을 할수 있는 서비스를 제공합니다. 해당 UCPID 서비스를 이용하면 인증서 유효성 검증 외에 공인인증 기관에서 제공하는 본인확인을 위한 개인정보를 확인 할수 있습니다.



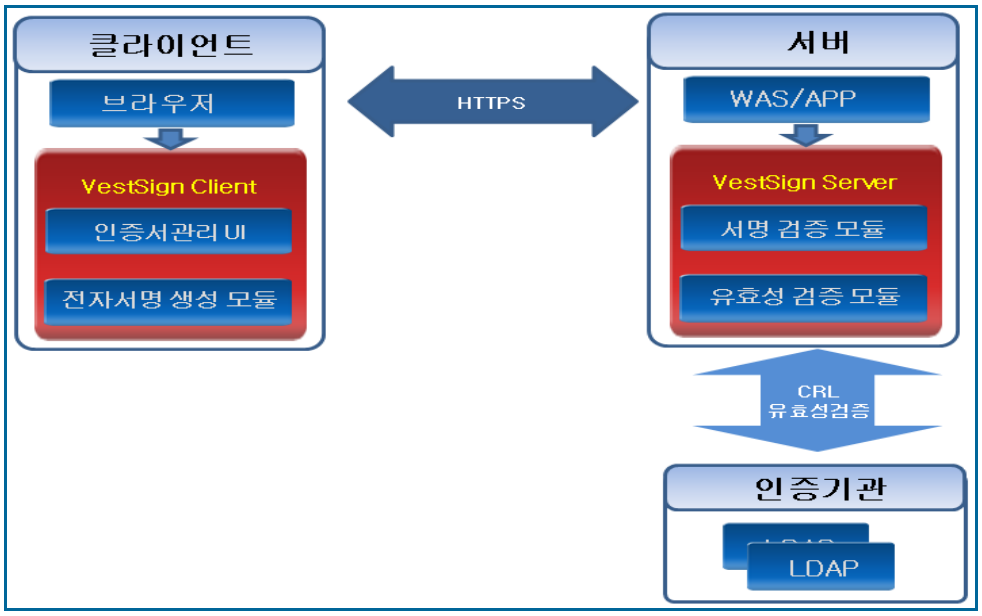
[2] 본인확인 규격: [KCAC.TS.SVID] – 식별번호를 이용한 본인확인 기술규격[v1.21]

1.1.5. 인증서 발급/폐기/갱신/관리 기능

공인인증서 발급/갱신/폐기 서비스를 받기 위한 CMP (Certificate Management Protocol) 프로토콜이 탑재되어 있어 공인인증기관과 통신하여 인증서를 발급/갱신/폐기를 수행할수 있다. 또한 Web-CMP 프로토콜을 제공한다.

1.2. 구성도

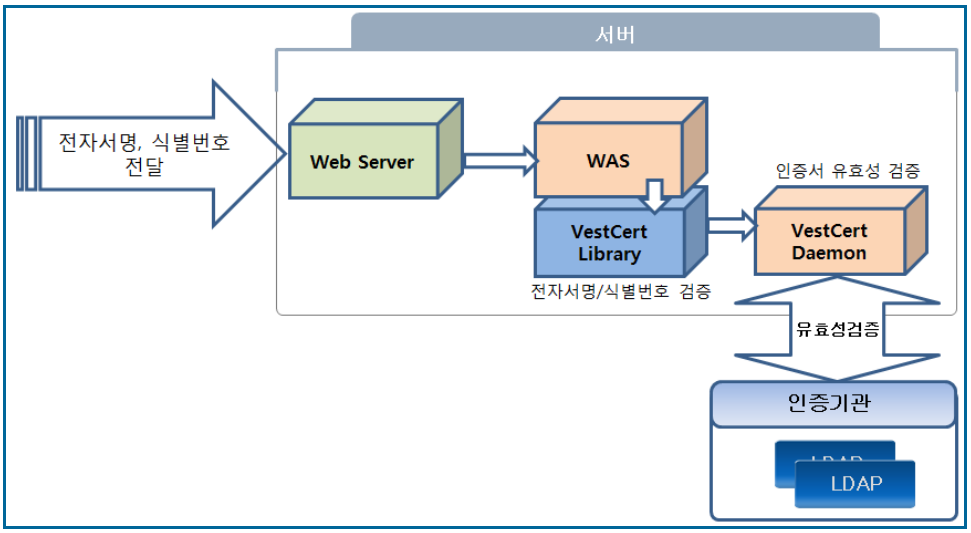
VestSign 솔루션은 클라이언트 모듈과 서버 모듈로 구성되어 있다. 클라이언트 모듈은 전자서명과 식별번호문서를 생성한다. 서버 모듈은 클라이언트에서 생성된 전자서명과 식별번호문서를 검증하는 모듈과 인증서를 유효성을 검증하는 모듈로 구성된다.



<그림 1> VestSign 구성도

1.2.1. 서버

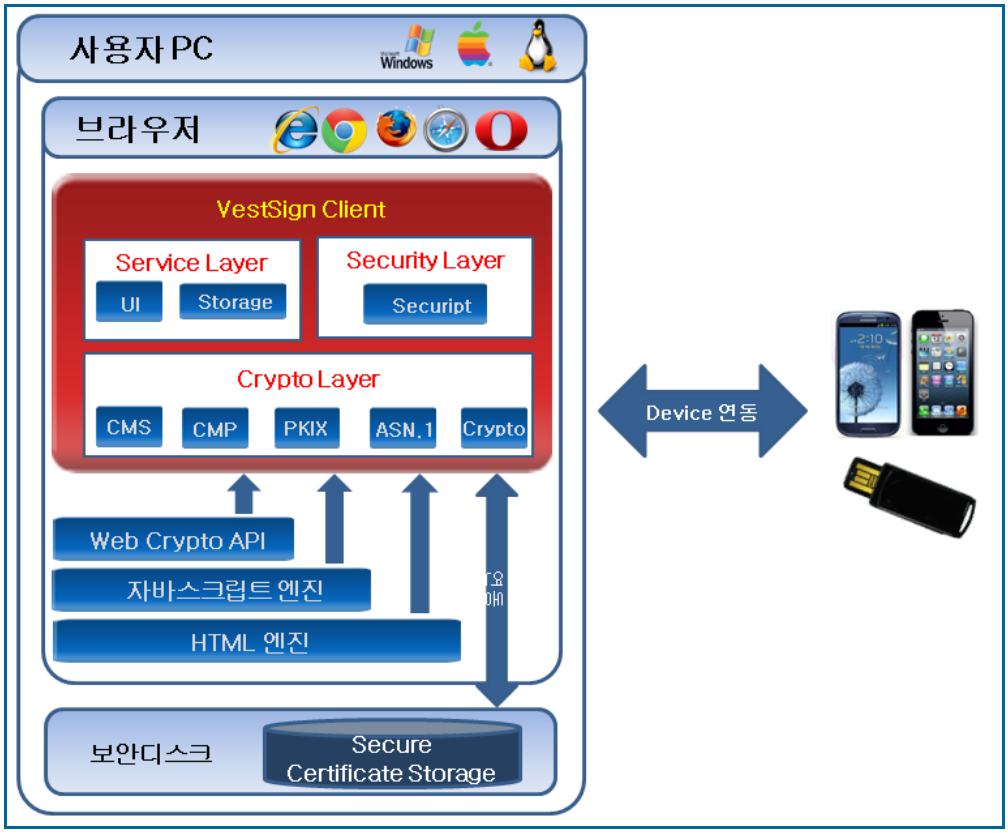
VestSign 서버모듈은 전자서명검증 모듈과 인증서 유효성검증 모듈로 구성된다.



<그림 2> 서버 구성도

1.2.2. 클라이언트

클라이언트 모듈은 암호화 라이브러리와 서비스 라이브러리, 보안 라이브러리로 구성된다.



<그림 3> 클라이언트 구성도

1.3. 지원 플랫폼

1.3.1. 서버

HardWare	Sun Sparc, IBM, HP PA-RISC, Intel Pentium
OS	Solaris 5.8 이상, Solaris x86 5.9 이상, AIX 5.1 이상 HP-UX 11.11 이상, HP-IA(itaninum) 11.23 이상 Linux Kernel 2.6 이상, Windows 2003 Server 이상
Web Server	모든 Web Server
Java Servlet Engine	jsdk 2.0/2.1/2.2/2.3 표준 규약을 지원하는 모든 Servlet Engine
JDK	Version 1.5 이상

<표 2> 서버 지원 플랫폼

1.3.2. 클라이언트

공인인증 서비스를 제공하기 위해 VestSign은 웹브라우저에서 동작하며, 각 기능별 플랫폼 지원 현황은 <표 3>과 같다.

기능	매체[3]	Windows[4]	Mac OS X[5]	Linux[6]
전자 서명 생성	하드디스크	- IE 8,9,10 이상 - Chrome, Firefox, Opera	- 10.8 이상	- Ubunu 12.04 이상 - Fedora 19 이상
	이동식디스크	- IE 8,9,10 이상 - Chrome, Firefox, Opera	- 10.8 이상	- Ubunu 12.04 이상 - Fedora 19 이상
	보안토큰	- IE 8,9,10 이상 - Chrome, Firefox, Opera	- N/A	- N/A
	인증서찾기	- IE 9, 이상 - Chrome, Firefox, Opera	- Chrome - Safari	- Chrome - Firefox
	웹저장소	- IE 9, 이상 - Chrome, Firefox, Opera	- Chrome - Safari	- Chrome - Firefox
	휴대폰인증	- IE 8,9,10 이상 - USIM, Infovine	- N/A	- N/A

<표 3> 클라이언트 기능별 지원 플랫폼

[3] 스마트 인증의 경우, 서비스 제공 업체와의 연동

[4] Microsoft Windows를 의미하며, Windows XP 이후 버전(Windows Vista, Windows 7, Windows 8 등)을 의미한다.

[5] Apple에서 개발한 Desktop OS를 의미하여, 10.8 이상을 의미한다.

[6] 현재 주요 배포판인 Ubuntu, Fedora의 주요 버전을 의미하며, Ubuntu 12.04 이상 / Fedora 19이상을 의미한다.

2. 설치 가이드

2.1. 서버

VestSign 서버모듈은 전자서명 검증 모듈과 인증서 검증모듈로 구성된다.

2.1.1. 디렉토리 구조

2.1.1.1. VestSign 데몬 디렉토리 구조

디렉토리 구조		설명	비고
VestSignValidator (Daemon)		startVS.bat stopVS.bat startvs.sh stopvs.sh	데몬 구동을 위한 스크립트
	bin	₩com₩yettiesoft₩vestsign₩*.jar	데몬구동 실행 라이브러리
	cert	₩cacert ₩srvcert ₩ucpid	인증기관 인증서 서버 인증서
	javadoc	Java doc 문서	
	lib	기반 라이브러리 /jni/Flatform]/*	기반 라이브러리 CMVP 라이브러리
	logs	CRL/OCSP 검증 로그	로그
	properties	설정파일	데몬 설정파일

<표 4> VestSign 데몬 디렉토리 구조

2.1.1.2. VestSign 라이브러리 디렉토리 구조

디렉토리 구조	설명	비고

VestSignServer			
	cert	₩cacert ₩srvcert	인증기관 인증서 서버 인증서
	javadoc	Java doc 문서	
	lib	기반 라이브러리 /jni/Flatform]/*	기반 라이브러리 CMVP 라이브러리
	logs	전자서명 검증 로그	로그
	properties	설정파일	데몬 설정파일

<표 5> VestSign 라이브러리 디렉토리 구조

2.1.2. VestSign 설치

2.1.2.1. VestSignValidator 설치

VestSign 솔루션은 VestSign_Server_[version].zip 의 형태로 배포된다.

① 패키지 파일 업로드
VestSign_Server_[version].zip 파일을 적용할 서버에 업로드 한다.

② 패키지 파일 압축 해제
$unzip VestSign_Server_[version].zip

③ Conf 파일 설정
/패키지폴더/properties/serverconfig.json 파일을 설정한다.

④ 구동 쉘 세팅
Startvs.sh 과 stopvs.sh 에서 설치 경로를 설정한다.

⑤ 구동
/패키지폴더/startvs.sh 을 실행시킨다.

2.1.2.2. VestSignServer 라이브러리 설치

VestSign 솔루션은 VestSign_Client_[version].zip 의 형태로 배포된다.

① 패키지 파일 업로드

VestSign_Client_[version].zip 파일을 적용할 서버에 업로드 한다.

② 패키지 파일 압축 해제

$unzip VestSign_ Client _[version].zip

③ Conf 파일 설정

/패키지폴더/properties/vestsign_client_config.json 파일을 설정한다.

config 파일 설정은 아래와 탕이 3가지 방법중 하나로 설정가능하다.

- 설정된 vestsign_client_config.json 파일을 WAS의 클래스패스에 복사한다.
- [user home]/VestConf 경로에 vestsign_client_config.json 를 복사한다.
- JVM 세팅에 java -DVestConfDir=/abc/VestConf 형식으로 설정한다.

④ CLASSPATH 설정

WAS CLASS_PATH에 아래의 jar 파일을 설정한다.

```
/패키지폴더/lib/com.yettiesoft.common-X.X.X.jar
/패키지폴더/lib/javarose_application-X.X.X.jar
/패키지폴더/lib/javarose_crypto-X.X.X.jar
/패키지폴더/lib/joda-time-2.2.jar
/패키지폴더/lib/ldapjdk_v4.1.jar
/패키지폴더/lib/snowchannel-1.5.0.jar
/패키지폴더/lib/vestsign_server-X.X.XX.jar
/패키지폴더/lib/sgcsp-1.0.0.0.jar
```

⑤ WAS 재기동

2.1.3. VestSign 설정

2.1.3.1. VestSign Validator 설정

VestSign Validator 설정 파일은 /패키지/properties/serverconfig.json에 위치한다.

필드		예시	설명
module-name		certcheck_enginer	모듈 명
module-version		v1.2.0	모듈 버전
charset		utf-8	Config 파일의 문자셋
secure-mode		false	VestSign Server 모듈과 통신 암호화 통신 여부
server	ocsp-server-protocol	http	인증기관의 OCSP 서버와 통신하는 프로토콜
	ocsp-server-ip	203.233.91.231	인증기관 OCSP 서버 IP
	ocsp-server-port	4612	인증기관 OCSP 서버 Port
	ocsp-timeout	5000	ocsp time out
	client-signature	true	인증기관 통신시 서버서명 여부
	verify-server-signature	false	인증기관 OCSP 서버 서명 검증 여부
log	log-dir	./logs/	로그 dir
	log-prefix	vestsign_verify_	로그파일 prefix
	log-suffix	.log	로그파일 확장자
	log-level	5	로그레벨 0 : audit, 1 : fatal, 2 : error, 3 : warning, 4 : info, 5 : debug
audit	audit-dir	./logs/audit/	감사로그 dir
	audit-crl-prefix	vestsign_crl_	crl 검증 감사로그 prefix
	audit-ocsp-prefix	vestsign_ocsp_	ocsp 검증 감사로그 prefix
	audit-ucpid-prefix	vestsign_ucpid_	ucpid 검증 감사로그 prefix
	audit-mydata-prefix	vestsign_mydata_	mydata 검증 감사로그 prefix

	audit-suffix	.log	감사로그 확장자
accept-cert	cert-dn	yessign	허용할 인증기관의 목록
daemon	daemon-port	9000	데몬의 Listen Port
	daemon-thread	100	최대 쓰레드 개수
	tcp-dump	true	TCP 통신 로그 덤프 여부
certificate	cacert	./cert/cacert	인증기관의 인증서 위치
	servercert	./cert/srvcert/signCert.der	서버 인증서
	serverkey	./cert/srvcert/signPri.key	서버 개인키
	serverpwd	43a54c4d69779f259e1d39688aff2a3b	서버 인증서 패스워드
ucpid	ca-name	yessign	ucpid 사용 기관 명
	version	0	버전
	ip	203.233.91.235	ucpid 서비스 기관 ip
	port	4719	ucpid 서비스 기관 port
	timeout	1000	소켓 timeout (ms)
	cp-code	Y00000000004	cp code
	cert-dir	./cert/ucpid/yessign	인증서 위치
	sign-cert-pwd	93a2b92024f9909e23360e4e7300cd52	서명용 인증서 패스워드
	km-cert-pwd	93a2b92024f9909e23360e4e7300cd52	KM용 인증서 패스워드
mydata	validity-signing-time	-1	Mydata 검증 유효시간
	status-check	0	인증서 유효성 검증 방법
	mydata-accept-cert	"1.2.410.200005.1.1.1"	허용 oid

<표 6> VestSign Validator 데몬 설정

2.1.3.2. VestSign Server 라이브러리 설정

VestSign Server 라이브러리 설정 파일은 /패키지/properties/vestsign_client_config.json에 위치한다.

필드		예시	설명
module-name		Vestsign_client	모듈 명
module-version		v1.2.0	모듈 버전
secure-mode		false	VestSign Client 모듈과 통신 암호화 통신 여부
tcp-dump		true	TCP 통신 로그 덤프 여부
engines	server-ip	127.0.0.1	VestSign Validator 데몬의 IP
	server-port	9000	VestSign Validator 데몬의 Port
log	log-dir	./logs/	로그 Dir
	log-prefix	vs_client_	로그파일 prefix
	log-suffix	.log	로그파일 확장자
	log-level	5	로그레벨 0 : audit, 1 : fatal, 2 : error, 3 : warning, 4 : info, 5 : debug
audit	audit-dir	./logs/audit/	감사로그 Dir
	audit-prefix	vs_client_audit_	감사로그 prefix
	audit-suffix	.log	감사로그 suffix
certificate	servercert	./cert/srvcert/signCert.der	서버 인증서 위치
	serverkey	./cert/srvcert/signPri.key	서버 개인키 위치
	serverpwd	43a54c4d69779f259e1d39688aff2a3b	서버 인증서 패스워드

<표 7> VestSign Server 라이브러리 설정

2.2. 클라이언트

VestSign 클라이언트 모듈은 전자서명모듈로 구성된다.

2.2.1. 디렉토리 구조

디렉토리 구조		파일 리스트	비고
VestSign v1.0		vestsign.js	
	doc	VestSign_Manaual_v1.1.1_201504.23.doc	
	languages	confirmLang.js manageLang.js storageLang.js	
	library	information dragiframe.js iecompatibility.js json3.min.js jsrose.js jsrose_crypto.js keySafer.js moment.min.js vestsign.core.js vestsign.error.js vestsign.util.js VestCertSetup.msi VestSign.swf	
	sample	Test sample html	
	script	confirm.js manage.js storageSelected.js	
	styles	css image js	

	views	confirm.html library.html manage.html storageSelected.html	

<표 8> VestSign 클라이언트 디렉토리 구조

2.2.1.1. 디렉토리 세부 사항

가) Root 경로

VestSign모듈의 최상단 경로로 VestSign 이용하기 위한 vestsign.js를 제공한다.

나) doc

VestSign의 주요 기능 및 구조를 소개하는 매뉴얼 문서를 제공한다.

다) language

VestSign에서 사용하는 모든 언어들에 대한 다국어(한글, 영어, 일어)를 정의한다.

라) library

VestSign에서 사용하는 주요 기능들을 모듈과 연결해주는 스크립트를 제공한다.

마) sample

VestSign에서 사용하는 sample html 및 jsp 등의 test 소스를 제공한다.

바) script

VestSign에서 사용하는 화면에 대한 이벤트 처리를 담당하는 소스를 제공한다.

사) styles

VestSign에 사용하는 화면을 구성하는 이미지 및 css 정보에 대한 소스를 제공한다.

아) views

VestSign에서 화면을 구성하는 html 소스를 제공한다.

3. 개발 가이드

VestSign라이브러리는 전자서명, 식별번호검증, 인증서검증의 기능을 제공하는 라이브러리이다.

3.1. 서버

3.1.1. 적용 방법

WAS에 적용되어 있는 VestSign Server 라이브러리는 전자서명검증, 식별번호검증, 인증서 유효성 검증 등을 위한 가각의 API를 제공한다. 사용자가 VestSign Client 모듈을 이용하여 만든 전자서명문과 식별번호 문서를 서버로 전달받아 해당 API를 이용하여 검증을 수행한다.

3.1.2. API Reference

3.1.2.1. 전자서명 검증

■ Class 정보

Class Name	SignVerifier.class
Package	com.yettiesoft.vestsign.external.SignVerifier

■ 생성자

SignVerifier	
prototype	public SignVerifier(byte[] bSignedData, int statusCheck)
설명	전사서명 검증 생성자
Parameters	bSignedData – byte[] 형의 Base64 encoding 되어 있는 서명문 statusCheck - 사용자 인증서 유효성 검증 방법 지정 CommonConst.CERT_STATUS_CRL: CRL 검증(0) CommonConst.CERT_STATUS_OCSP : OCSP 검증(1) CommonConst.CERT_STATUS_NONE : 검증안함(2)
Return	N/A

SignVerifier

prototype	public SignVerifier(byte[] bSignedData, int statusCheck, int encodingRule)
설명	전사서명 검증 생성자
Parameters	bSignedData – byte[] 형의 Base64 또는 Hex encoding 되어 있는 서명문 statusCheck - 사용자 인증서 유효성 검증 방법 지정 CommonConst.CERT_STATUS_CRL: CRL 검증(0) CommonConst.CERT_STATUS_OCSP : OCSP 검증(1) CommonConst.CERT_STATUS_NONE : 검증안함(2) encodingRule - bSignedData 의 인코딩 룰 지정 CommonConst.ENCODE_BASE64 : base64 인코딩(0) CommonConst.ENCODE_HEX : hex 인코딩(1)
Return	N/A

SignVerifier	
prototype	public SignVerifier(java.lang.String sSignedData, int statusCheck)
설명	전사서명 검증 생성자
Parameters	sSignedData – String 형의 Base64 encoding되어 있는 서명문 statusCheck - 사용자 인증서 유효성 검증 방법 지정 CommonConst.CERT_STATUS_CRL: CRL 검증(0) CommonConst.CERT_STATUS_OCSP : OCSP 검증(1) CommonConst.CERT_STATUS_NONE : 검증안함(2)
Return	N/A

SignVerifier	
prototype	public SignVerifier(java.lang.String sSignedData, int statusCheck, int encodingRule)
설명	전사서명 검증 생성자
Parameters	sSignedData – String 형의 Base64 또는 Hex encoding되어 있는 서명문 statusCheck - 사용자 인증서 유효성 검증 방법 지정 CommonConst.CERT_STATUS_CRL: CRL 검증(0) CommonConst.CERT_STATUS_OCSP : OCSP 검증(1)

	CommonConst.CERT_STATUS_NONE : 검증안함(2) encodingRule - bSignedData 의 인코딩 룰 지정 CommonConst.ENCODE_BASE64 : base64 인코딩(0) CommonConst.ENCODE_HEX : hex 인코딩(1)
Return	N/A

SignVerifier	
prototype	public SignVerifier(byte[] bSignedData)
설명	전사서명 검증 생성자. default로 CRL 검증한다.
Parameters	bSignedData – byte[] 형의 Base64 encoding되어 있는 서명문
Return	N/A

SignVerifier	
prototype	public SignVerifier(java.lang.String sSignedData)
설명	전사서명 검증 생성자. default로 CRL 검증한다.
Parameters	sSignedData – String 형의 Base64 encoding되어 있는 서명문
Return	N/A

SignVerifier	
prototype	public SignVerifier(byte[] message, String certPem, byte[] signature, int statusCheck)
설명	전사서명 검증 생성자. 원문, 인증서, PKCS1 서명값을 각각 입력받아 검증한다.
Parameters	message – 서명 원문 certPem – PEM 형식의 인증서 signature – PKCS1 서명값 statusCheck - 사용자 인증서 유효성 검증 방법 지정 CommonConst.CERT_STATUS_CRL: CRL 검증(0) CommonConst.CERT_STATUS_OCSP : OCSP 검증(1) CommonConst.CERT_STATUS_NONE : 검증안함(2)

Return	N/A

■ 함수

verify	
prototype	public void verify()
설명	전자 서명 검증 함수. 함수 실행 후 getLastErrorCode() 와 getLastErrorMsg() 를 통해서 정상 여부를 판별한다. 0 : 서명 검증 정상 및 인증거 검증 유효 -10 : 인증서 만료 -20 : 인증서 폐기 -30 : 인증서 상태 확인 불가 그외 오류
Parameters	N/A
Return	N/A

getSignerCertificate	
prototype	public CertificateInfo getSignerCertificate()
설명	서명자 인증서 정보 class 추출
Parameters	N/A
Return	CertificateInfo 전자서명한 인증서

getSignedMessage	
prototype	public byte[] getSignedMessage()
설명	서명 원문 추출하여 binary 형태로 반환한다.
Parameters	N/A
Return	전자서명 원문

getSignedMessageText	
prototype	public String getSignedMessageText()
설명	서명 원문 추출하여 String 형태로 반환한다.

Parameters	N/A
Return	전자서명 원문

getSignature	
prototype	public byte[] getSignature()
설명	PKCS1 서명값을 byte[]로 반환한다.
Parameters	N/A
Return	pkcs1 서명값

setStatusCheck	
prototype	public void setStatusCheck(int statusCheck)
설명	인증서 유효성 검증 방법을 설정한다.
Parameters	statusCheck - 사용자 인증서 유효성 검증 방법 지정 CommonConst.CERT_STATUS_CRL: CRL 검증(0) CommonConst.CERT_STATUS_OCSP : OCSP 검증(1) CommonConst.CERT_STATUS_NONE : 검증안함(2)
Return	N/A

setContentInfoOfSignedData	
prototype	public void setContentInfoOfSignedData(byte[] contentInfoOfSignedData)
설명	전자 서명문(CMS)을 설정한다.
Parameters	contentInfoOfSignedData – CMS 서명문
Return	N/A

isKCDSA	
prototype	public boolean isKCDSA()
설명	서명 검증 후 KCDSA 알고리즘을 사용한는 인증서를 이용한 서명인지 판별한다.
Parameters	N/A
Return	true : kcdsa 알고리즘 사용 인증서

getLastErrorCode	
prototype	publid int getLastErrorCode();
설명	마지막 error code를 반환하는 함수
Parameters	N/A
Return	마지막 error code(int 형)

getLastErrorMsg	
prototype	publid java.lang.String getLastErrorMsg();
설명	마지막 error 메시지를 반환하는 함수
Parameters	N/A
Return	마지막 error 메시지(String 형)

3.1.2.2. 식별번호 검증

■ Class 정보

Class Name	VidVerifier.class
Package	com.yettiesoft.vestsign.external.VidVerifier

■ 생성자

VidVerifier	
prototype	public VidVerifier();
설명	식별번호 검증 생성자
Parameters	N/A
Return	N/A

■ 함수

verifyVirtualID	
prototype	public boolean verifyVirtualID(java.lang.String vid, java.lang.String ssn,

	CertificateInfo cert);
설명	식별번호 검증 함수
Parameters	vid – Base64 encoding 되어있는 VID 메시지 ssn – 인증서 발급 시 사용된 주민등록번호 또는 사업자번호 cert – VID 검증에 사용되는 사용자 인증서 (전자서명문으로부터 자동 추출된다.)
Return	검증 결과 (true: 성공, false: 실패)

verifyVirtualID	
prototype	public boolean verifyVirtualID(java.lang.String vid, java.lang.String ssn, CertificateInfo cert, int encodingRule);
설명	식별번호 검증 함수
Parameters	vid – Hex 또는 Base64 encoding 되어있는 VID 메시지 ssn – 인증서 발급 시 사용된 주민등록번호 또는 사업자번호 cert – VID 검증에 사용되는 사용자 인증서 (전자서명문으로부터 자동 추출된다.) encodingRule - vid 메시지의 encoding 룰 지정 CommonConst.ENCODE_BASE64 : base64 인코딩(0) CommonConst.ENCODE_HEX : hex 인코딩(1)
Return	검증 결과 (true: 성공, false: 실패)

writeServerCertScript	
prototype	public String writeServerCertScript(String var)
설명	VID 메시지 암호화를 위한 서버 인증서 출력한다. JavaScript var(변수) 형식으로 출력한다.
Parameters	var – 서버인증서 출력 시, JavaScript 변수
Return	생성된 JavaScript 문자
writeServerCertPem	
prototype	public String writeServerCertPem()
설명	VID 메시지 암호화를 위한 서버 인증서를 PEM 형식으로 출력한다.

Parameters	N/A
Return	PEM 형식의 인증서

getLastErrorCode	
prototype	publid int getLastErrorCode();
설명	마지막 error code를 반환하는 함수
Parameters	
Return	마지막 error code(int 형)

getLastErrorMsg	
prototype	publid java.lang.String getLastErrorMsg();
설명	마지막 error 메시지를 반환하는 함수
Parameters	
Return	마지막 error 메시지(String 형)

3.1.2.3. 파일 전자서명 검증

■ Class 정보

Class Name	**FileSignVerifier.class**
Package	**com.yettiesoft.vestsign.external.FileSignVerifier**

■ 생성자

FileSignVerifier	
prototype	public FileSignVerifier(String inputFilePath, String outputFilePath , int statusCheck)
설명	파일 전자서명 검증 생성자
Parameters	inputFilePath – 전자서명된 입력 파일 Path(파일명 포함) outputFilePath – 전자서명 검증후 원문 출력 파일 Path(파일명 포함)

	statusCheck – 사용자 인증서 유효성 검증 방법 지정 CommonConst.CERT_STATUS_CRL: CRL 검증(0) CommonConst.CERT_STATUS_OCSP : OCSP 검증(1) CommonConst.CERT_STATUS_NONE : 검증안함(2)
Return	N/A

FileSignVerifier	
prototype	public FileSignVerifier(String inputFilePath, String outputFilePath, String plainFilePath, int statusCheck)
설명	파일 전자서명 검증 생성자
Parameters	inputFilePath – 원문이 없는 전자서명 입력 파일 Path(파일명 포함) outputFilePath – 전자서명 검증후 원문 출력 파일 Path(파일명 포함) plainFilePath – 원문 입력 파일 Path(파일명 포함) statusCheck – 사용자 인증서 유효성 검증 방법 지정 CommonConst.CERT_STATUS_CRL: CRL 검증(0) CommonConst.CERT_STATUS_OCSP : OCSP 검증(1) CommonConst.CERT_STATUS_NONE : 검증안함(2)
Return	N/A

■ 함수

verify	
prototype	public void verify()
설명	전자 서명 검증 함수. 함수 실행 후 getLastErrorCode() 와 getLastErrorMsg() 를 통해서 정상 여부를 판별한다. 0 : 서명 검증 정상 및 인증거 검증 유효 -10 : 인증서 만료 -20 : 인증서 폐기 -30 : 인증서 상태 확인 불가 그외 오류
Parameters	N/A
Return	N/A

getLastErrorCode	
prototype	publid int getLastErrorCode();
설명	마지막 error code를 반환하는 함수
Parameters	N/A
Return	마지막 error code(int 형)

getLastErrorMsg	
prototype	publid java.lang.String getLastErrorMsg();
설명	마지막 error 메시지를 반환하는 함수
Parameters	N/A
Return	마지막 error 메시지(String 형)

3.1.2.4. 인증서 검증

■ Class 정보

Class Name	CertVerifier.class
Package	com.yettiesoft.vestsign.external.CertVerifier

■ 생성자

CertVerifier	
prototype	public CertVerifier()
설명	인증서 상태 검증 생성자
Parameters	N/A
Return	N/A

■ 함수

verify	
prototype	public boolean verify(String certPem, int statusCheck)

설명	인증서 상태 검증 함수. PEM 형식의 인증서와 statusCheck 옵션을 입력받아 CRL 또는 OCSP 검증을 수행한다. 함수 실행 후 getLastErrorCode() 와 getLastErrorMsg()를 통해서 정상 여부를 판별한다. 0 : 인증거 검증 유효 -10 : 인증서 만료 -20 : 인증서 폐기 -30 : 인증서 상태 확인 불가 그외 오류
Parameters	certPem – PEM 형식의 인증서 statusCheck – 사용자 인증서 유효성 검증 방법 지정 CommonConst.CERT_STATUS_CRL: CRL 검증(0) CommonConst.CERT_STATUS_OCSP : OCSP 검증(1)
Return	정상 여부

verify	
prototype	public boolean verify(String encodedCert, int statusCheck, int encodingRule)
설명	인증서 상태 검증 함수. Base64 또는 Hex 인코딩된 인증서와 statusCheck 옵션을 입력받아 CRL 또는 OCSP 검증을 수행한다. 함수 실행 후 getLastErrorCode() 와 getLastErrorMsg()를 통해서 정상 여부를 판별한다. 0 : 인증거 검증 유효 -10 : 인증서 만료 -20 : 인증서 폐기 -30 : 인증서 상태 확인 불가 그외 오류
Parameters	encodedCert – Base64 또는 Hex 인코딩된 인증서 statusCheck – 사용자 인증서 유효성 검증 방법 지정 CommonConst.CERT_STATUS_CRL: CRL 검증(0) CommonConst.CERT_STATUS_OCSP : OCSP 검증(1) encodingRule - encodedCert의 encoding 룰 지정 CommonConst.ENCODE_BASE64 : base64 인코딩(0) CommonConst.ENCODE_HEX : hex 인코딩(1)
Return	정상 여부

verify	
prototype	public boolean verify(byte[] cert, int statusCheck)
설명	인증서 상태 검증 함수. Byte Array 인증서와 statusCheck 옵션을 입력받아 CRL 또는 OCSP 검증을 수행한다. 함수 실행 후 getLastErrorCode() 와 getLastErrorMsg()를 통해서 정상 여부를 판별한다. 0 : 인증거 검증 유효 -10 : 인증서 만료 -20 : 인증서 폐기 -30 : 인증서 상태 확인 불가 그외 오류
Parameters	cert – 인증서 statusCheck – 사용자 인증서 유효성 검증 방법 지정 CommonConst.CERT_STATUS_CRL: CRL 검증(0) CommonConst.CERT_STATUS_OCSP : OCSP 검증(1)
Return	정상 여부

getLastErrorCode	
prototype	publid int getLastErrorCode();
설명	마지막 error code를 반환하는 함수
Parameters	N/A
Return	마지막 error code(int 형)

getLastErrorMsg	
prototype	publid java.lang.String getLastErrorMsg();
설명	마지막 error 메시지를 반환하는 함수
Parameters	N/A
Return	마지막 error 메시지(String 형)

3.1.2.5. UCPID 검증

- Class 정보

Class Name	UcpidVerifier.class
Package	com.yettiesoft.vestsign.external.UcpidVerifier

■ 생성자

UcpidVerifier	
prototype	public UcpidVerifier()
설명	UCPID 검증 생성자
Parameters	N/A
Return	N/A

■ 함수

verify	
prototype	public UCPIDResult verify(String signedPersonInfoReq, int encodingRule)
설명	UCPID 검증 함수. 함수 실행 후 getLastErrorCode() 와 getLastErrorMsg()를 통해서 정상 여부를 판별한다. UCPIDResult Class를 통해 정보를 추출한다.
Parameters	signedPersonInfoReq – 사용자 요청 정보에 대한 서명값 encodingRule - signedPersonInfoReq 메시지의 encoding 룰 지정 CommonConst.ENCODE_BASE64 : base64 인코딩(0) CommonConst.ENCODE_HEX : hex 인코딩(1)
Return	UCPIDResult : UCPID 검증 결과 Class

verify	
prototype	public UCPIDResult verify(String signedPersonInfoReq, int encodingRule, String cpCode)
설명	UCPID 검증 함수. 함수 실행 후 getLastErrorCode() 와 getLastErrorMsg()를 통해서 정상 여부를 판별한다. UCPIDResult Class를 통해 정보를 추출한다.
Parameters	signedPersonInfoReq – 사용자 요청 정보에 대한 서명값 encodingRule - signedPersonInfoReq 메시지의 encoding 룰 지정 CommonConst.ENCODE_BASE64 : base64 인코딩(0) CommonConst.ENCODE_HEX : hex 인코딩(1)

	cpCode – CP 코드, null 이면 conf에 설정된 값을 사용하고 입력되면 입력된 값을 사용한다.
Return	UCPIDResult : UCPID 검증 결과 Class

verify	
prototype	public UCPIDResult verify(String signedPersonInfoReq, String cpRequestNumber, int encodingRule)
설명	UCPID 검증 함수. 함수 실행 후 getLastErrorCode() 와 getLastErrorMsg()를 통해서 정상 여부를 판별한다. UCPIDResult Class를 통해 정보를 추출한다.
Parameters	signedPersonInfoReq – 사용자 요청 정보에 대한 서명값 cpRequestNumber – CP 요청 번호, null 이면 랜덤으로 생성됨. encodingRule - signedPersonInfoReq 메시지의 encoding 룰 지정 CommonConst.ENCODE_BASE64 : base64 인코딩(0) CommonConst.ENCODE_HEX : hex 인코딩(1)
Return	UCPIDResult : UCPID 검증 결과 Class

verify	
prototype	public UCPIDResult verify(String signedPersonInfoReq, String cpRequestNumber, int encodingRule, String cpCode)
설명	UCPID 검증 함수. 함수 실행 후 getLastErrorCode() 와 getLastErrorMsg()를 통해서 정상 여부를 판별한다. UCPIDResult Class를 통해 정보를 추출한다.
Parameters	signedPersonInfoReq – 사용자 요청 정보에 대한 서명값 cpRequestNumber – CP 요청 번호, null 이면 랜덤으로 생성됨. encodingRule - signedPersonInfoReq 메시지의 encoding 룰 지정 CommonConst.ENCODE_BASE64 : base64 인코딩(0) CommonConst.ENCODE_HEX : hex 인코딩(1) cpCode – CP 코드, null 이면 conf에 설정된 값을 사용하고 입력되면 입력된 값을 사용한다.
Return	UCPIDResult : UCPID 검증 결과 Class

getLastErrorCode

prototype	publid int getLastErrorCode();
설명	마지막 error code를 반환하는 함수
Parameters	N/A
Return	마지막 error code(int 형)

getLastErrorMsg	
prototype	publid java.lang.String getLastErrorMsg();
설명	마지막 error 메시지를 반환하는 함수
Parameters	N/A
Return	마지막 error 메시지(String 형)

3.1.2.6. 전자봉투 생성 및 복호화

■ Class 정보

Class Name	**Enveloper.class**
Package	**com.yettiesoft.vestsign.external.Enveloper**

■ 생성자

Enveloper	
prototype	public Enveloper()
설명	전자 봉투 생성자
Parameters	N/A
Return	N/A

■ 함수

envelope	
prototype	public String envelope(String message, String certPem, int encodingRule)

설명	전자 봉투 생성 함수. 함수 실행 후 getLastErrorCode() 와 getLastErrorMsg() 를 통해서 정상 여부를 판별한다.
Parameters	message – 입력 평문 String certPem – PEM 형식 인증서 encodingRule - 출력 String Encoding 룰 지정 CommonConst.ENCODE_BASE64 : base64 인코딩(0) CommonConst.ENCODE_HEX : hex 인코딩(1)
Return	Base64 또는 Hex 인코딩된 전자봉투 값.

envelope	
prototype	public byte[] envelope(byte[] message, byte[] cert)
설명	전자 봉투 생성 함수. 함수 실행 후 getLastErrorCode() 와 getLastErrorMsg() 를 통해서 정상 여부를 판별한다.
Parameters	message – byte[] 입력 평문 cert – byte[] 인증서
Return	byte[] 형 전자봉투 값.

deEnvelope	
prototype	public String deEnvelope (String envelopeMessage, byte[] privateKey, String password, int encodingRule)
설명	전자 봉투 복호화 함수. 함수 실행 후 getLastErrorCode() 와 getLastErrorMsg() 를 통해서 정상 여부를 판별한다.
Parameters	envelopeMessage – encoding 된 전자봉투 값. 인코딩은 encodingRule 로 설정 privateKey – 인증서 개인키 password – 인증서 패스워드 encodingRule - envelopeMessage Encoding 룰 지정 CommonConst.ENCODE_BASE64 : base64 인코딩(0) CommonConst.ENCODE_HEX : hex 인코딩(1)
Return	복호화된 원문 메시지

deEnvelope	
prototype	public byte[] deEnvelope (byte[] envelopeMessage, byte[] privateKey, String password)
설명	전자 봉투 복호화 함수. 함수 실행 후 getLastErrorCode() 와 getLastErrorMsg() 를 통해서 정상 여부를 판별한다.
Parameters	envelopeMessage – byte[] 전자봉투 값 privateKey – 인증서 개인키 password – 인증서 패스워드
Return	복호화된 원문 메시지

envelopeFile	
prototype	public int envelopeFile(String inputFilePath, String outputFilePath, byte[] cert)
설명	파일 전자 봉투 생성 함수. 평문 파일 Path를 입력받아 전자봉투 생성해서 출력될 파일 Path로 내보낸다. 함수 실행 후 getLastErrorCode() 와 getLastErrorMsg()를 통해서 정상 여부를 판별한다.
Parameters	inputFilePath – 평문 입력 파일 Path(파일명 포함) outputFilePath – 전자봉투 출력 파일 Path(파일명 포함) cert – 인증서
Return	0 : 정상, 그외 오류

envelopeFile	
prototype	public int envelopeFile(String inputFilePath, String outputFilePath, String certPem)
설명	파일 전자 봉투 생성 함수. 평문 파일 Path를 입력받아 전자봉투 생성해서 출력될 파일 Path로 내보낸다. 함수 실행 후 getLastErrorCode() 와 getLastErrorMsg()를 통해서 정상 여부를 판별한다.
Parameters	inputFilePath – 평문 입력 파일 Path(파일명 포함) outputFilePath – 전자봉투 출력 파일 Path(파일명 포함) certPem – PEM 형식의 인증서

Return	0 : 정상, 그외 오류

deEnvelopeFile	
prototype	public int deEnvelopeFile (String inputFilePath, String outputFilePath, byte[] privateKey, String password)
설명	파일 전자 봉투 복호화 함수. 전자봉투 파일 Path를 입력받아 복호화한 값을 출력될 파일 Path로 내보낸다. 함수 실행 후 getLastErrorCode() 와 getLastErrorMsg() 를 통해서 정상 여부를 판별한다.
Parameters	inputFilePath – 전자봉투 입력 파일 Path(파일명 포함) outputFilePath – 복호화 출력 파일 Path(파일명 포함) privateKey – 인증서 개인키 password – 인증서 패스워드
Return	0 : 정상, 그외 오류

getLastErrorCode	
prototype	publid int getLastErrorCode();
설명	마지막 error code를 반환하는 함수
Parameters	N/A
Return	마지막 error code(int 형)

getLastErrorMsg	
prototype	publid java.lang.String getLastErrorMsg();
설명	마지막 error 메시지를 반환하는 함수
Parameters	N/A
Return	마지막 error 메시지(String 형)

3.1.2.7. 대칭키 암/복호화

- Class 정보

Class Name	Cipher.class
Package	com.yettiesoft.vestsign.external.Cipher

■ 생성자

Cipher	
prototype	public Cipher()
설명	Cipher 생성자. 알고리즘 SEED, CBC 모드, PKCS7 padding 이 default로 설정 된다.
Parameters	N/A
Return	N/A

Cipher	
prototype	public Cipher(int algorithm)
설명	Cipher 생성자. 알고리즘은 입력을 받고, CBC 모드, PKCS7 padding 이 default로 설정 된다.
Parameters	algorithm – 대칭키 알고리즘 ex)Cipher.ALGORITHM_SEED
Return	N/A

Cipher	
prototype	public Cipher(int algorithm, int mode, int padding)
설명	Cipher 생성자. 알고리즘, 모드, 패딩 값을 입력 받는다. 사용 가능 알고리즘에는 SEED, ARIA, AES, LEA 가 있다. * SEED : Cipher.ALGORITHM_SEED * ARIA : Cipher.ALGORITHM_ARIA * AES : Cipher.ALGORITHM_AES * LEA : Cipher.ALGORITHM_LEA 모드는 ECB, CBC, CFB, OFB, CTR 5가지 모드를 사용할 수 있다. * ECB : Cipher.MODE_ECB * CBC : Cipher.MODE_CBC * CFB : Cipher.MODE_CFB * OFB : Cipher.MODE_OFB * CTR : Cipher.MODE_CTR 패딩 설정으로는 zero 패딩과, PKCS7 패딩이 있다. ECB모드와 CBC 모드에서

	는 패딩을 반드시 설정해야 된다. * NONE : Cipher.PADDING_NONE * ZERO : Cipher.PADDING_ZERO * PKCS7 : Cipher.PADDING_PKCS7
Parameters	algorithm – 대칭키 알고리즘 ex)Cipher.ALGORITHM_SEED mode – 모드 설정 ex)Cipher.MODE_ECB padding – 패딩 설정 ex)Cipher.PADDING_PKCS7
Return	N/A

■ 함수

encrypt	
prototype	public byte[] encrypt(byte[] plain, byte[] key, byte[] iv) throws VSException
설명	암호화 함수. ECB 모드에서는 iv 가 필요 없지만 나머지 모드에서는 iv 값을 설정해야 된다. iv 는 block size 크기와 같아야 된다. block size는 모두 16byte 이다.
Parameters	plain – 평문 key – 마스터 키 iv – iv 값
Return	암호화 값

encrypt	
prototype	public String encrypt(String strPlain, String strPw) throws VSException
설명	암호화 함수. String 평문을 받아서 암호값을 base64 인코딩된 String을 리턴한다. strPw 값을 이용해서 내부적으로 master key와 iv를 생성해서 사용한다.
Parameters	strPlain – String 형의 평문 strPw – String 형의 패스워드
Return	base64 인코딩된 암호화 값

encryptBase64	
prototype	public String encryptBase64(String base64Plain, String base64Key, String base64Iv) throws VSException
설명	Base64 형태의 입출력을 수행하는 암호화 함수. 모든 입력값과 출력값을 base64 인코딩된 String을 사용한다.
Parameters	base64Plain – base64 인코딩된 평문 base64Key – base64 인코딩된 마스터 키 base64Iv – base64 인코딩된 IV
Return	base64 인코딩된 암호화 값

encryptFile	
prototype	public void encryptFile(String inputFilePath, String outputFilePath, byte[] key, byte[] iv) throws VSException
설명	파일 암호화 함수. 평문 파일 Path를 입력받아 암호화 해서 출력될 파일 Path로 내보낸다. ECB 모드에서는 iv 가 필요 없지만 나머지 모드에서는 iv 값을 설정해야 된다. iv 는 block size 크기와 같아야 된다. block size는 모두 16byte 이다.
Parameters	inputFilePath – 평문 입력 파일 Path(파일명 포함) outputFilePath – 암호화 출력 파일 Path(파일명 포함) key –마스터 키 iv – iv 값
Return	N/A

encryptFile	
prototype	public void encryptFile(String inputFilePath, String outputFilePath, String strPw) throws VSException
설명	파일 암호화 함수. 평문 파일 Path를 입력받아 암호화 해서 출력될 파일 Path로 내보낸다. strPw 값을 이용해서 내부적으로 master key와 iv를 생성해서 사용한다.

Parameters	inputFilePath – 평문 입력 파일 Path(파일명 포함) outputFilePath – 암호화 출력 파일 Path(파일명 포함) strPw – String 형의 패스워드
Return	N/A

decrypt	
prototype	public byte[] decrypt(byte[] encData, byte[] key, byte[] iv) throws VSException
설명	복호화 함수. ECB 모드에서는 iv 가 필요 없지만 나머지 모드에서는 iv 값을 설정해야 된다. iv 는 block size 크기와 같아야 된다. block size는 모두 16byte 이다.
Parameters	encData – 암호문 key – 마스터 키 iv – iv 값
Return	복호화 값

decrypt	
prototype	public String decrypt(String base64EncData, String strPw) throws VSException
설명	복호화 함수. String base64 암호문을 받아서 복호화된 String을 리턴한다. strPw 값을 이용해서 내부적으로 master key와 iv를 생성해서 사용한다.
Parameters	base64EncData – base64 인코딩된 암호문 strPw – String 형의 패스워드 iv – iv 값
Return	String 형 복호화 값

decryptBase64	
prototype	public String decryptBase64(String base64EncData, String base64Key, String base64Iv) throws VSException
설명	Base64 형태의 입출력을 수행하는 복호화 함수. 모든 입력값과 출력값을 base64 인코딩된 String을 사용한다.

Parameters	base64EncData – base64 인코딩된 암호문 base64Key – base64 인코딩된 마스터 키 base64Iv – base64 인코딩된 IV
Return	base64 인코딩된 복호화 값

decryptFile	
prototype	public void decryptFile(String inputFilePath, String outputFilePath, byte[] key, byte[] iv) throws VSException
설명	파일 복호화 함수. 암호화된 파일 Path를 입력받아 복호화 해서 출력될 파일 Path로 내보낸다. ECB 모드에서는 iv 가 필요 없지만 나머지 모드에서는 iv 값을 설정해야 된다. iv 는 block size 크기와 같아야 된다. block size는 모두 16byte 이다.
Parameters	inputFilePath – 암호화된 입력 파일 Path(파일명 포함) outputFilePath – 복호화 출력 파일 Path(파일명 포함) key – 마스터 키 iv – iv 값
Return	N/A

decryptFile	
prototype	public void decryptFile(String inputFilePath, String outputFilePath, String strPw) throws VSException
설명	파일 복호화 함수. 암호화된 파일 Path를 입력받아 복호화 해서 출력될 파일 Path로 내보낸다. strPw 값을 이용해서 내부적으로 master key와 iv를 생성해서 사용한다.
Parameters	inputFilePath – 암호화된 입력 파일 Path(파일명 포함) outputFilePath – 복호화 출력 파일 Path(파일명 포함) strPw – String 형의 패스워드
Return	N/A

setBitBlockSize	
prototype	public void setBitBlockSize(int bitBlockSize)

설명	CFB Mode 일 경우 설정 한다. default 값은 0 이다.
Parameters	bitBlockSize – 8 이상 128이하 값으로 8의 배수로 입력한다. 이 값으로 blockSize가 결정 된다. (blockSize = bitBlockSize / 8)
Return	N/A

3.1.2.8. 인증서 정보 추출

■ Class 정보

Class Name	**CertificateInfo.class**
Package	**com.yettiesoft.vestsign.external.CertificateInfo**

■ 생성자

CertificateInfo	
prototype	public CertificateInfo (byte[] bCert)
설명	인증서 정보를 추출하는 CertificateInfo 클래스의 생성자
Parameters	bCert – byte[] 형 인증서
Return	N/A

CertificateInfo	
prototype	public CertificateInfo (String certPem)
설명	인증서 정보를 추출하는 CertificateInfo 클래스의 생성자
Parameters	certPem – PEM 형식 인증서
Return	N/A

CertificateInfo	
prototype	public CertificateInfo (SGCertificate cert)
설명	인증서 정보를 추출하는 CertificateInfo 클래스의 생성자
Parameters	cert – com.yettiesoft.javarose.standard.cert.SGCertificate 형의 인증서
Return	N/A

■ 함수

getVersion	
prototype	public int getVersion()
설명	인증서 버전을 반환한다.
Parameters	N/A
Return	인증서 버전

getSerial	
prototype	public int getSerial ()
설명	인증서 일련번호를 반환한다.
Parameters	N/A
Return	int 형 인증서 시리얼

getSerialToHex	
prototype	public String getSerialToHex()
설명	인증서 일련번호를 반환한다.
Parameters	N/A
Return	hex 로 변환된 인증서 시리얼

getSignatureAlgorithm	
prototype	public String getSignatureAlgorithm()
설명	인증서 서명 알고리즘을 반환한다.
Parameters	N/A
Return	서명 알고리즘

getSignatureAlgorithmIdentifier	
prototype	public String getSignatureAlgorithm()
설명	인증서 서명 알고리즘을 반환한다.

Parameters	N/A
Return	SGAlgorithmIdentifier 서명 알고리즘

getSignature	
prototype	public byte[] getSignature()
설명	인증서 서명값을 반환한다.
Parameters	N/A
Return	서명값

getIssuer	
prototype	public String getIssuer()
설명	인증서 발급자 DN을 반환한다.
Parameters	N/A
Return	발급자 DN

getIssuer	
prototype	public String getIssuer(String entryName)
설명	인증서 발급자 DN 값 중 entryName에 해당하는 필드를 반환한다. entryName 은 cn, o, ou, l, e, c 값을 입력할 수 있다.
Parameters	entryName – cn, o, ou, l, e, c 값
Return	entryName에 해당하는 값

getIssuer	
prototype	public String getIssuer(String entryName, int index)
설명	인증서 발급자 DN 값 중 entryName에 해당하는 필드를 반환한다. entryName 은 cn, o, ou, l, e, c 값을 입력할 수 있다. 두개 이상 존재할 경우 index에 해당하는 값을 반환한다.
Parameters	entryName – cn, o, ou, l, e, c 값 index – 0 이상 값

Return	entryName에 해당하는 값

getStartDate	
prototype	public Date getStartDate()
설명	인증서 유효기간 시작 시간을 반환한다.
Parameters	N/A
Return	유효기간 시작 시간

getStartDate	
prototype	public Date getStartDate(String format)
설명	인증서 유효기간 시작 시간을 반환한다. 입력된 날짜 format 형식으로 변환해서 출력한다.
Parameters	format – 출력할 날짜의 포맷 형식
Return	유효기간 시작 시간

getEndDate	
prototype	public Date getEndDate()
설명	인증서 유효기간 종료 시간을 반환한다.
Parameters	N/A
Return	유효기간 종료 시간

getEndDate	
prototype	public String getEndDate(String format)
설명	인증서 유효기간 종료 시간을 반환한다. 입력된 날짜 format 형식으로 변환해서 출력한다.
Parameters	format – 출력할 날짜의 포맷 형식
Return	유효기간 종료 시간

getSubject	
prototype	public String getSubject()

설명	인증서 사용자 DN 값을 반환한다.
Parameters	N/A
Return	사용자 DN 값

getSubject	
prototype	public String getSubject(String entryName)
설명	인증서 사용자 DN 값을 반환한다. DN 값 중 entryName에 해당하는 필드를 반환한다. entryName 은 cn, o, ou, l, e, c 값을 입력할 수 있다.
Parameters	entryName – cn, o, ou, l, e, c
Return	entryName에 해당하는 값

getSubject	
prototype	public String getSubject(String entryName, int index)
설명	인증서 사용자 DN 값을 반환한다. DN 값 중 entryName에 해당하는 필드를 반환한다. entryName 은 cn, o, ou, l, e, c 값을 입력할 수 있다. 두개 이상 존재할 경우 index에 해당하는 값을 반환한다.
Parameters	entryName – cn, o, ou, l, e, c index – 0 이상값
Return	entryName에 해당하는 값

getPublicKey	
prototype	public byte[] getPublicKey()
설명	인증서 공개키를 반환한다.
Parameters	N/A
Return	공개키

getPublicKeyAlgorithm	
prototype	public String getPublicKeyAlgorithm()
설명	인증서 공개키 알고리즘을 반환한다.

Parameters	N/A
Return	공개키 알고리즘

getAuthorityKeyIdentifier	
prototype	public byte[] getAuthorityKeyIdentifier()
설명	인증서 Authority Key Identifier를 반환한다.
Parameters	N/A
Return	인증서 Authority Key Identifier

getSubjectKeyIdentifier	
prototype	public byte[] getSubjectKeyIdentifier()
설명	인증서 Subject Key Identifier를 반환한다.
Parameters	N/A
Return	인증서 Subject Key Identifier

getKeyUsage	
prototype	public byte[] getKeyUsage()
설명	인증서 확장 사용 용도를 반환한다.
Parameters	N/A
Return	확장 사용 용도

getPolicyIdentifier	
prototype	public String getPolicyIdentifier()
설명	인증서 정책 필드값을 반환한다.
Parameters	N/A
Return	정책 필드

getUserNotice	
prototype	public String getUserNotice()

설명	인증서 User Notice 값을 반환한다.
Parameters	N/A
Return	인증서 User Notice 값

getUsage	
prototype	public String getUsage ()
설명	인증서 용도을 반환한다.
Parameters	N/A
Return	인증서 용도

isCorporationUsage	
prototype	public boolean isCorporationUsage()
설명	법인 범용 인증서 유무를 확인한다.
Parameters	N/A
Return	법인 범용 인증서 유무(true: 법인 범용, false: 개인 범용 외 기타)

getSubjectAlternativeName	
prototype	public String getSubjectAlternativeName ()
설명	인증서 보유자의 AltName 필드를 반환한다.
Parameters	N/A
Return	AltName

getCRLDistPoint	
prototype	public String getCRLDistPoint ()
설명	인증서 CRL 분기점을 반환한다.
Parameters	N/A
Return	CRL 분기점

getOCSPUrl

prototype	public String getOCSPUrl ()
설명	인증서 OCSP URL을 반환한다.
Parameters	N/A
Return	OCSP URL

getCertificate	
prototype	public SGCertificate getCertificate()
설명	인증서를 SGCertificate Class로 반환한다.
Parameters	N/A
Return	SGCertificate Class

getCertificateToPEM	
prototype	public String getCertificateToPEM ()
설명	인증서를 PEM 형태로 반환한다.
Parameters	N/A
Return	PEM 형태의 인증서

3.1.2.9. HASH 생성

■ Class 정보

Class Name	**VSHash.class**
Package	**com.yettiesoft.vestsign.util.VSHash**

■ 함수

hash	
prototype	public static byte[] hash(byte[] input, int algorithm) throws VSException
설명	Hash 값을 생성해서 반환한다.
Parameters	input – 입력값

	algorithm – hash 알고리즘 VSHash.ALGORITHM_MD2 VSHash.ALGORITHM_MD4 VSHash.ALGORITHM_MD5 VSHash.ALGORITHM_SHA1 VSHash.ALGORITHM_SHA224 VSHash.ALGORITHM_SHA256 VSHash.ALGORITHM_SHA384 VSHash.ALGORITHM_SHA512
Return	hash 값

hash	
prototype	public static String hash(byte[] input, int algorithm, int outEncoding)
설명	Hash 값을 생성해서 반환한다.
Parameters	input – 입력값 algorithm – hash 알고리즘 VSHash.ALGORITHM_MD2 VSHash.ALGORITHM_MD4 VSHash.ALGORITHM_MD5 VSHash.ALGORITHM_SHA1 VSHash.ALGORITHM_SHA224 VSHash.ALGORITHM_SHA256 VSHash.ALGORITHM_SHA384 VSHash.ALGORITHM_SHA512 outEncoding – 출력 String의 인코딩 룰 VSHash. ENCODE_BASE64 VSHash. ENCODE_HEX
Return	인코딩된 hash String

hash	
prototype	public static String hash(String input, int algorithm, int outEncoding)

설명	Hash 값을 생성해서 반환한다.
Parameters	input – 입력값 algorithm – hash 알고리즘 VSHash.ALGORITHM_MD2 VSHash.ALGORITHM_MD4 VSHash.ALGORITHM_MD5 VSHash.ALGORITHM_SHA1 VSHash.ALGORITHM_SHA224 VSHash.ALGORITHM_SHA256 VSHash.ALGORITHM_SHA384 VSHash.ALGORITHM_SHA512 outEncoding – 출력 String의 인코딩 룰 VSHash. ENCODE_BASE64 VSHash. ENCODE_HEX
Return	인코딩된 hash String

3.1.2.10. BASE64 인코딩 생성

■ Class 정보

Class Name	**VSBase64.class**
Package	**com.yettiesoft.vestsign.util.VSBase64**

■ 함수

encodeBytes	
prototype	public static byte[] encodeBytes(byte[] data)
설명	입력값을 Base64 인코딩한 값을 byte[] 로 반한한다.
Parameters	data – 입력값
Return	Base64 인코딩한 값

encode	
prototype	public static String encode (byte[] data)
설명	입력값을 Base64 인코딩한 값을 String으로 반한한다.

Parameters	data – 입력값
Return	Base64 인코딩한 값

encode	
prototype	public static String encode (String data)
설명	입력값을 Base64 인코딩한 값을 String으로 반한한다.
Parameters	data – 입력값
Return	Base64 인코딩한 값

decodeBytes	
prototype	public static byte[] decodeBytes(byte[] data)
설명	입력값을 Base64 디코딩한 값을 byte[]로 반한한다.
Parameters	data – Base64 인코딩 값
Return	Base64 디코딩한 값

decode	
prototype	public static byte[] decode(String data)
설명	입력값을 Base64 디코딩한 값을 byte[]로 반한한다.
Parameters	data – Base64 인코딩 값
Return	Base64 디코딩한 값

3.1.2.11. HEX 인코딩 생성

■ Class 정보

Class Name	**VSHex.class**
Package	**com.yettiesoft.vestsign.util.VSHex**

■ 함수

encodeBytes

prototype	public static byte[] encodeBytes(byte[] data)
설명	입력값을 Hex 인코딩한 값을 byte[] 로 반한한다.
Parameters	data – 입력값
Return	Hex 인코딩한 값

encode	
prototype	public static String encode (byte[] data)
설명	입력값을 Hex 인코딩한 값을 String으로 반한한다.
Parameters	data – 입력값
Return	Hex 인코딩한 값

encode	
prototype	public static String encode (String data)
설명	입력값을 Hex 인코딩한 값을 String으로 반한한다.
Parameters	data – 입력값
Return	Hex 인코딩한 값

decodeBytes	
prototype	public static byte[] decodeBytes(byte[] data)
설명	입력값을 Hex 디코딩한 값을 byte[]로 반한한다.
Parameters	data – Hex 인코딩 값
Return	Hex 디코딩한 값

decode	
prototype	public static byte[] decode(String data)
설명	입력값을 Hex 디코딩한 값을 byte[]로 반한한다.
Parameters	data – Hex 인코딩 값
Return	Hex 디코딩한 값

3.1.2.12. MYDATA 검증

■ Class 정보

Class Name	MyData.class
Package	com.yettiesoft.vestsign.external.MyDataVerifier

■ 생성자

MyDataVerifier	
prototype	public MyDataVerifier()
설명	MYDATA 검증 생성자
Parameters	N/A
Return	N/A

■ 함수

verify	
prototype	public UCPIDResult verify(String jsonString, String ucpidNonce, String consentNonse, String transactionID)
설명	MYDATA 검증 함수. 함수 실행 후 getLastErrorCode() 와 getLastErrorMsg() 를 통해서 정상 여부를 판별한다. UCPIDResult Class를 통해 정보를 추출한다.
Parameters	jsonString – 사용자 요청 정보 및 consent에 대한 서명값 ucpidNonce – ucpidNonce 데이터 consentNonce – consentNonce 데이터 transactionID – transactionID 데이터
Return	UCPIDResult : UCPID 검증 결과 Class

genNonce	
prototype	public String genNonce()
설명	MYDATA 용 난수를 생성하는 함수이다.
Parameters	

Return	String : url safe base64 encoding된 128bit 난수값

getPersonInfoReq	
prototype	public static byte[] getPersonInfoReq(String jsonString);
설명	마이데이터 서명문에서 personInfoReq의 서명원문을 추출하는 함수
Parameters	마이데이터 서명문
Return	byte[] : 마이데이터 서명문에서 personInfoReq의 서명원문

getConsent	
prototype	public static byte[] getConsent(String jsonString);
설명	마이데이터 서명문에서 consent의 서명원문을 추출하는 함수
Parameters	마이데이터 서명문
Return	byte[] : 마이데이터 서명문에서 consent의 서명원문

getLastErrorMsg	
prototype	public java.lang.String getLastErrorMsg();
설명	마지막 error 메시지를 반환하는 함수
Parameters	N/A
Return	마지막 error 메시지(String 형)

getLastErrorMsg	
prototype	public java.lang.String getLastErrorMsg();
설명	마지막 error 메시지를 반환하는 함수
Parameters	N/A
Return	마지막 error 메시지(String 형)

3.2. 클라이언트

3.2.1. 적용 방법

vestsign.js를 열어 12라인의 setting list를 수정한다.

변수명	하위 변수	설명
Title		html 화면에대한 title을 설정한다.
baseUrl		root경로인 vestsign.js의 위치를 설정한다.
ablePwd		비밀번호창의 활성화 여부를 설정한다.
installFilePath		다운로드 링크를 설정한다.
keystrokeEncryption		사용할 키보드보안 업체를 설정한다.
keySaferPath		키보드보안 스크립트의 경로를 설정한다.
keySaferConfig		키보드보안에 관련된 세부내용을 설정한다.
storage	hardDisk	하드디스크 기능을 사용하기위해 설정한다.
	useDisk	이동식디스크 기능을 사용하기위해 설정한다.
	secureToken	보안토큰 기능을 사용하기 위해 설정한다.
	saveToken	저장토큰 기능을 사용하기 위해 설정한다.
	certificateFile	인증서 찾기 기능을 사용하기 위해 설정한다.
	secureDisk	안전디스크 기능을 사용하기 위해 설정한다.
	smartPhone	휴대폰인증 기능을 사용하기 위해 설정한다.
version	vestCert	버전체크에 사용될 exe 버전을 설정한다.
	vestCertSecurityService	버전체크에 사용될 해당 dll 버전을 설정한다.
infovine	wParam	해당 업체의 휴대폰인증 기능을 사용하기 위해 업체가 직접 설정한다.
	lParam	해당 업체의 휴대폰인증 기능을 사용하기 위해 업체가 직접 설정한다.
	version	해당 업체의 휴대폰인증 기능을 사용하기 위해 업체가 직접 설정한다.
	url	해당 업체의 휴대폰인증 기능을 사용하기 위해 업체가 직접 설정한다.
dreamsecurity	tokenorder	해당 업체의 휴대폰인증 기능을 사용하기 위해 업체가 직접 설정한다.
	sitecode	해당 업체의 휴대폰인증 기능을 사용하기 위해 업체가 직접 설정한다.
	modcode	해당 업체의 휴대폰인증 기능을 사용하기 위해 업체가 직접 설정한다.

	siteURL	해당 업체의 휴대폰인증 기능을 사용하기 위해 업체가 직접 설정한다.
	serviceIP	해당 업체의 휴대폰인증 기능을 사용하기 위해 업체가 직접 설정한다.
	servicePort	해당 업체의 휴대폰인증 기능을 사용하기 위해 업체가 직접 설정한다.

<표 9> VestSign 옵션

적용하고자 하는 페이지(*.html, *.jsp)에서 vestsign.js와 json3.min.js를 import한다.

ex)

```
<script type="text/javascript" src="../vestsign.js"></script>
<script type="text/javascript" src="../library/json3.min.js"></script>
```

추가 후, API Reference를 참조하여 사용한다..

(sign.html, sign.jsp 이동 시 sign.html, sign.jsp 파일 내에 있는 상대경로를 수정해야 한다.)

3.2.2. API Reference

3.2.2.1. INIT

VestSign의 설정값을 추가 설정하는 함수이다. 해당 함수를 통해 각 기능별 함수를 호출하기전에 설정값을 추가 설정 및 변경할 수 있다.

Init	
Prototype	yettie.init(config);
설명	설정값 추가 함수
Parameters	config – config에 추가할 object ex) yettie.init({title: "title"});
Return	

3.2.2.2. SIGN

VestSign의 전자서명 기능을 수행하는 함수이다. Option에 따라 다양한 서명문을 생성할 수 있다. (PKCS#7, PKCS#1 등 지원)

또한 해당 함수를 통해 인증서 목록을 보여주며 인증서 선택을 통해 전자서명 기능을 수행한다.

sign	
prototype	yettie.sign(plain, option, callback, errorcallback)
설명	전자서명을 수행한다.
Parameters	plain – 전자서명할 서명 원문 option – 서명에 필요한 option 값(object 형태) encoding:"base64" 내부적으로 사용할 encoding charset: "utf-8" 전자서명 원문 charset signType: "2" 전자서명 형태(default 2 – PKCS#7 sign) callback – 서명 완료 시 결과를 받는 callback 함수 errorcallback – 서명 처리 시 error를 처리할 callback 함수
Return	

3.2.2.3. FILESIGN

VestSign의 파일 전자서명 기능을 수행하는 함수이다. Option에 따라 다양한 서명문을 생성할 수 있다. (PKCS#7, PKCS#1 등 지원)

또한 해당 함수를 통해 인증서 목록을 보여주며 인증서 선택을 통해 파일 전자서명 기능을 수행한다.

Sign	
prototype	yettie.fileSign(outfile, plain, option, callback, errorcallback)
설명	파일 전자서명을 수행한다.
Parameters	outfile – 저장될 파일명 plain – 원문 파일 경로 option – 서명에 필요한 option 값(object 형태) encoding:"base64" 내부적으로 사용할 encoding charset: "utf-8" 전자서명 원문 charset signType: "2" 전자서명 형태(default 2 – PKCS#7 sign) callback – 서명 완료 시 결과를 받는 callback 함수 errorcallback – 서명 처리 시 error를 처리할 callback 함수
Return	

3.2.2.4. VERIFYSIGNATURE

VestSign의 전자서명문을 검증하는 함수이다.

verifySignature	
prototype	yettie.verifySignature(signedMsg, callback, errorcallback)
설명	전자서명문 검증을 수행한다.
Parameters	signedMsg – 검증할 전자서명문 callback – 검증 완료 시 결과를 받는 callback 함수 errorcallback – 검증 기능 수행 시 error를 처리할 callback 함수
Return	

3.2.2.5. FILEVERIFYSIGNATURE

VestSign의 파일 전자서명문을 검증하는 함수이다.

fileVerifySignature	
prototype	yettie.fileVerifySignature(outfile, signedMsg, params, option, callback, errorcallback)
설명	파일 전자서명문 검증을 수행한다.
Parameters	signedMsg – 검증할 전자서명문 callback – 검증 완료 시 결과를 받는 callback 함수 errorcallback – 검증 기능 수행 시 error를 처리할 callback 함수
Return	

3.2.2.6. ENVELOPE

VestSign의 전자봉투 암호화를 수행하는 함수이다.

다음 함수 수행 시 전자봉투 암호화에 이용할 인증서를 선택한 후 전자봉투 암호화를 수행한다.

envelope	
prototype	yettie.envelope(plain, option, callback, errorcallback)
설명	인증서를 이용한 전자봉투 암호화를 수행한다.
Parameters	plain – 전자봉투에 이용할 원문 option – 전자봉투 옵션 callback – 전자봉투 완료 시 결과를 받는 callback 함수 errorcallback – 전자봉투 기능 수행 시 error를 처리할 callback 함수
Return	

3.2.2.7. DEENVELOPE

VestSign의 전자봉투 복호화를 수행하는 함수이다.

다음 함수 수행 시 전자봉투 복호화에 이용할 인증서를 선택한 후 전자봉투 복호화를 수행한다.

deenvelope	
prototype	yettie.deenvelope(plain, outputfile, option, callback, errorcallback)
설명	인증서를 이용한 전자봉투 복호화를 수행한다.
Parameters	plain – 전자봉투 암호화된 암호문 outputfile – 전자봉투 복호화 시 저장할 파일 경로 option – 전자봉투 옵션 callback – 전자봉투 복호화 완료 시 결과를 받는 callback 함수 errorcallback – 전자봉투 복호화 기능 수행 시 error를 처리할 callback 함수
Return	

3.2.2.8. FILEENVELOPE

VestSign의 파일 전자봉투 암호화를 수행하는 함수이다.

다음 함수 수행 시 파일 전자봉투 암호화에 이용할 인증서를 선택한 후 파일 전자봉투 암호화를 수행한다.

fileEnvelope	
prototype	yettie.fileEnvelope(outfile, certificate, plain, option, callback, errorcallback)
설명	인증서를 이용한 파일 전자봉투 암호화를 수행한다.
Parameters	plain – 전자봉투에 이용할 원문 option – 전자봉투 옵션 callback – 전자봉투 완료 시 결과를 받는 callback 함수 errorcallback – 전자봉투 기능 수행 시 error를 처리할 callback 함수
Return	

3.2.2.9. FILEDEENVELOPE

VestSign의 파일 전자봉투 복호화를 수행하는 함수이다.

다음 함수 수행 시 파일 전자봉투 복호화에 이용할 인증서를 선택한 후 파일 전자봉투 복호화를 수행한다.

fileDeenvelope	
prototype	yettie.fileDeenvelope(outfile, certificate, plain, option, callback, errorcallback)
설명	인증서를 이용한 파일 전자봉투 복호화를 수행한다.
Parameters	outputfile – 파일 전자봉투 복호화 시 저장될 파일 경로 certificate – pem 구조 형식 plain – 파일 전자봉투로 암호화된 암호문 파일의 경로 option – reserved 변수(파일 전자봉투에 대한 설정은 해당 API 내에서 설정됨) callback – 파일 전자봉투 복호화 완료 시 결과를 받는 callback 함수 errorcallback – 파일 전자봉투 복호화 기능 수행 시 error를 처리할 callback 함수
Return	

3.2.2.10. ISSUE

VestSign의 인증서 발급 기능을 수행한다.

기관별 인증서를 발급할 시에 참조번호, 인가코드 발급이 선행되어야 한다.

issue	
prototype	yettie.issue(option, callback, errorcallback)
설명	인증서 발급 기능을 수행한다.
Parameters	option – 인증서 발급 시 필요한 정보를 입력할 option option.ca (발급할 인증서의 인증기관 이름) ("yessign", "signkorea", "crosscert", "signgate, "kica") option.refNum (인증서 발급 시 필요한 참조번호) option.authCode (인증서 발급 시 필요한 인가코드) callback – 인증서 발급 완료 시 결과를 받는 callback 함수 errorcallback – 인증서 발급 기능 수행 시 error를 처리할 callback 함수
Return	

3.2.2.11. UPDATE

VestSign의 인증서 갱신 기능을 수행한다.

update	
prototype	yettie.update(option, callback, errorcallback)
설명	인증서 갱신 기능을 수행한다.
Parameters	option – 인증서 갱신 시 필요한 정보를 입력할 option callback – 인증서 갱신 완료 시 결과를 받는 callback 함수 errorcallback – 인증서 갱신 기능 수행 시 error를 처리할 callback 함수
Return	인증서 갱신은 인증서 발급과는 달리 인증서 선택 후 이루어지기 때문에 별도의 option에 ca 이름을 넣을 필요가 없다. 또한 갱신은 참조번호와 인가코드를 입력하지 않는다.

3.2.2.12. REVOKE

VestSign의 인증서 폐기 기능을 수행한다.

revoke	
prototype	yettie.revoke(option, callback, errorcallback)
설명	인증서 폐기 기능을 수행한다.
Parameters	option – 인증서 폐기 시 필요한 정보를 입력할 option callback – 인증서 폐기 완료 시 결과를 받는 callback 함수 errorcallback – 인증서 폐기 기능 수행 시 error를 처리할 callback 함수
Return	인증서 폐기는 인증서 발급과 달리 인증서 선택 후 이루어지기 때문에 별도의 option에 ca 이름을 넣을 필요가 없다. 또한 폐기는 참조번호와 인가코드를 입력하지 않는다.

3.2.2.13. RECOVER

VestSign의 인증서 재발급 기능을 수행한다.

recover	
prototype	yettie.recover(option, callback, errorcallback)
설명	인증서 재발급 기능을 수행한다.
Parameters	option – 인증서 재발급 시 필요한 정보를 입력할 option option.ca (재발급 할 인증서의 인증기관 이름) ("yessign", "signkorea", "crosscert", "signgate, "kica") option.refNum (인증서 재발급 시 필요한 참조번호) option.authCode (인증서 재발급 시 필요한 인가코드) callback – 인증서 재발급 완료 시 결과를 받는 callback 함수 errorcallback – 인증서 재발급 기능 수행 시 error를 처리할 callback 함수
Return	

3.2.2.14. EXPORTP12

VestSign의 인증서 내보내기 기능을 수행한다.

하드디스크/이동식디스크의 NPKI에 저장되어 있는 개인 인증서를 PKCS#12 형태의 인증서로 내보내기한다.

exportP12	
prototype	yettie.exportP12(callback, errorcallback)
설명	인증서 내보내기 기능을 수행한다.
Parameters	callback – 인증서 내보내기 완료 시 결과를 받는 callback 함수 errorcallback – 인증서 내보내기 기능 수행 시 error를 처리할 callback 함수
Return	

3.2.2.15. IMPORTP12

VestSign의 인증서 가져오기 기능을 수행한다.

PKCS#12 형태의 인증서를 하드디스크의 NPKI에 가져오기한다.

importP12	
prototype	yettie.importP12(callback, errorcallback)
설명	인증서 가져오기 기능을 수행한다.
Parameters	callback – 인증서 가져오기 완료 시 결과를 받는 callback 함수 errorcallback – 인증서 가져오기 기능 수행 시 error를 처리할 callback 함수
Return	

3.2.2.16. CHANGEPIN

VestSign의 인증서 비밀번호를 변경 기능을 수행한다.

changePin	
prototype	yettie.changePin(callback, errorcallback)
설명	인증서 비밀번호 변경 기능을 수행한다.
Parameters	callback – 인증서 비밀번호 변경 완료 시 결과를 받는 callback 함수 errorcallback – 인증서 비밀번호 변경 기능 수행 시 error를 처리할 callback 함수
Return	

3.2.2.17. CHECKPIN

VestSign의 인증서 비밀번호 확인 기능을 수행한다.

checkPin	
prototype	yettie.checkPin(callback, errorcallback)
설명	인증서 비밀번호 확인 기능을 수행한다.
Parameters	callback – 인증서 비밀번호 확인 완료 시 결과를 받는 callback 함수 errorcallback – 인증서 비밀번호 확인 기능 수행 시 error를 처리할 callback 함수
Return	

3.2.2.18. changeStorage

VestSign의 인증서 저장매체 변경 기능을 수행한다.

하드디스크의 NPKI에 있는 인증서를 이동식디스크의 NPKI에 저장을 한다던지 그 반대의 기능을 수행한다.

changeStorage	
prototype	yettie.changeStorage(callback, errorcallback)
설명	인증서 저장매체 변경 기능을 수행한다.
Parameters	callback – 인증서 저장매체 변경 완료 시 결과를 받는 callback 함수 errorcallback – 인증서 저장매체 변경 기능 수행 시 error를 처리할 callback 함수
Return	

3.2.2.19. signFormUCPID

VestSign의 본인확인서비스 기능을 수행한다.

signFormUCPID	
prototype	yettie.signFormUCPID(userAgreement, realName, birthdate, gender, nationalInfo, option, callback, errorcallback)
설명	본인확인서비스 기능을 수행한다.
Parameters	userAgreement – 개인정보제공 및 활용동의 약관 realName – ISP에서 필요로 하는 개인정보 항목 중 사용자 이름 동의 항목 birthdate – ISP에서 필요로 하는 개인정보 항목 중 사용자 생년월일 동의 항목 gender – ISP에서 필요로 하는 개인정보 항목 중 사용자 성별 동의 항목 nationalInfo – ISP에서 필요로 하는 개인정보 항목 중 사용자 국적 동의 항목 option – 서명 함수와 동일한 옵션 callback – 본인확인서비스 기능 완료 시 결과를 받는 callback 함수 errorcallback – 본인확인서비스 기능 수행 시 error를 처리할 callback 함수
Return	본인확인서비스 기능 중 option은 전자서명 처리에서와 동일한 option을 사용한다.

3.2.2.20. GETHASH

VestSign의 hash 기능을 수행한다.

getHash	
prototype	yettie.getHash(plainType, plain, mode, callback, errorcallback)
설명	hash 기능을 수행한다.
Parameters	plainType – plain의 형태 (0: 원문, 1: 원문의 파일 경로) plain – hash 기능을 수행할 plainType에 맞는 원문 형태 mode - hash algorithm (0: sha1, 1: sha256) callback – hash 기능 완료 시 결과를 받는 callback 함수 errorcallback – hash 기능 수행 시 error를 처리할 callback 함수
Return	

3.2.2.21. GETPCINFO

VestSign의 PC정보 확인 기능을 수행한다.

getHash	
prototype	yettie.getPCInfo(option, callback, errorcallback)
설명	PC정보 확인 기능을 수행한다.
Parameters	option – reserved 변수 callback – PC정보 확인 기능 완료 시 결과를 받는 callback 함수 errorcallback – PC정보 확인 기능 수행 시 error를 처리할 callback 함수
Return	

3.2.2.22. ENCRYPT

VestSign의 대칭키 암호화 기능을 수행한다.

encrypt	
prototype	yettie.encrypt(plain, args, keys, callback, errorcallback)
설명	암호화 기능을 수행한다.
Parameters	plain – 암호화 기능을 수행할 원문 args – 암호화 기능을 수행할 때 필요한 알고리즘, 운영모드, 패딩 정보 args.algorithm - 대칭키 알고리즘 (0: seed, 1: aes, 2: aria, 3: TDES) args.mode – 운영모드 (0: CBC, 1: ECB) args.padding – padding (0: pkcs, 1: no, 2: zero) keys – 암호화에 사용할 key callback – 암호화 기능 완료 시 결과를 받는 callback 함수 errorcallback – 암호화 기능 수행 시 error를 처리할 callback 함수
Return	대칭키 암호화 시 운영모드와 padding과의 상관관계로 인해 지원하지 않는 패딩이 있어 error가 발생할 수 있다. Ex) ECB mode 일 경우 no padding은 error 가 발생한다.

3.2.2.23. DECRYPT

VestSign의 대칭키 복호화 기능을 수행한다.

decrypt	
prototype	yettie.decrypt(plain, args, keys, callback, errorcallback)
설명	복호화 기능을 수행한다.
Parameters	plain – 복호화 기능을 수행할 암호문 args – 복호화 기능을 수행할 때 필요한 알고리즘, 운영모드, 패딩 정보 args.algorithm - 대칭키 알고리즘 (0: seed, 1: aes, 2: aria, 3: TDES) args.mode – 운영모드 (0: CBC, 1: ECB) args.padding – padding (0: pkcs, 1: no, 2: zero) keys – 복호화에 사용할 key callback – 복호화 기능 완료 시 결과를 받는 callback 함수 errorcallback – 복호화 기능 수행 시 error를 처리할 callback 함수
Return	대칭키 복호화 시 운영모드와 padding과의 상관관계로 인해 지원하지 않는 패딩이 있어 error가 발생할 수 있다. Ex) ECB mode 일 경우 no padding은 error 가 발생한다.

3.2.2.24. SCRIPTFILEENCRYPT

VestSign의 파일 대칭키 암호화 기능을 수행한다.

scriptFileEncrypt	
prototype	yettie.scriptFileEncrypt(fileObj, outputFileName, keyValue, option, callback, errorcallback)
설명	파일 암호화 기능을 수행한다.
Parameters	fileObj – File object outputFileName – 암호화 후 저장할 파일 이름 keyValue – 암호화에 사용할 key option – reserved 변수 callback – 파일 암호화 기능 완료 시 결과를 받는 callback 함수 errorcallback – 파일 암호화 기능 수행시 error를 처리할 callback 함수
Return	기존 암복호화는 다르게 지정된 알고리즘을 이용한다. (SEED-CBC)

3.2.2.25. SCRIPTFILEDECRYPT

VestSign의 파일 대칭키 복호화 기능을 수행한다.

scriptFileDecrypt	
prototype	yettie.scriptFileDecrypt(fileObj, outputFileName, keyValue, option, callback, errorcallback)
설명	파일 복호화 기능을 수행한다.
Parameters	fileObj – File object outputFileName – 복호화 후 저장할 파일 이름 keyValue – 복호화에 사용할 key option – reserved 변수 callback – 복호화 기능 완료 시 결과를 받는 callback 함수 errorcallback – 파일 복호화 기능 수행시 error를 처리할 callback 함수
Return	기존 암복호화는 다르게 지정된 알고리즘을 이용한다. (SEED-CBC)

3.2.2.26. TRAYON

VestSign의 tray를 활성화한다.

trayOn	
prototype	yettie.trayOn (callback)
설명	VestSign의 tray를 활성화한다.
Parameters	callback – tray 활성화 후 결과를 받는 callback 함수
Return	

3.2.2.27. TRAYOFF

VestSign의 tray를 비활성한다.

setLanguage	
prototype	yettie.trayOff (callback)
설명	VestSign의 tray를 비활성화한다.
Parameters	callback – tray 비활성화 후 결과를 받는 callback 함수
Return	

3.2.2.28. GETVESTCERTVERSION

VestSign의 버전을 확인한다.

getVestCertVersion	
prototype	yettie.getVestCertVersion (option, callback, errorcallback)
설명	VestSign의 버전을 확인한다.
Parameters	option - reserved 변수 callback – 버전 확인 완료시 결과를 받는 callback 함수 errorcallback – 버전 확인 기능 수행 시 error를 처리할 callback 함수
Return	

3.2.2.29. GETFILEPATH

VestSign의 선택한 파일 경로를 가져온다.

setLanguage	
prototype	yettie.getFilePath (option, callback, errorcallback)
설명	VestSign의 선택한 파일 경로를 가져온다.
Parameters	option - reserved 변수 callback – 파일을 선택 후 경로를 받는 callback 함수 errorcallback – 파일 경로를 가져오는 기능 수행 시 error를 처리할 callback 함수
Return	

3.3. 적용예제

3.3.1. 전자서명 생성

```
<script type="text/javascript" src="../app/vestsign.js"></script>
<script type="text/javascript" src="../app/library/json3.min.js"></script>

<SCRIPT language=javascript>

if(!(typeof(console) === 'object' && typeof(console.log) === 'object')){
    console = {};
    console.log = function() {};
}

  var option = {
       encoding: 'hex',
       charset: 'utf-8',          // utf-8, euc-kr 선택
       signtype: '2' // p7서명
  };

  var callback = function (result) {
    document.getElementById('signedMsg').placeholder = result.signature;
    document.getElementById('signedMsg').value = result.signature;
  };

  var errorcallback = function (error) {
    // error = { code, getReason() };
    if(error.code == -9999) console.log(error.code);   // 취소 버튼 이벤트 error.code: -9999
    else alert(error.msg);
  };

  function sign() {
    var plain = document.getElementById('signMsg').value;
                if (plain == null || plain == "")
                {
                        var plain = document.getElementById('signMsg').placeholder;
                }
    yettie.sign(plain, option, callback, errorcallback);
  };


  function signVerifyServer() {

                if(sForm.signedMsg.value == ""){
                        alert("전자서명문 메시지가 올바르지 않습니다.");
                        return ;
                }
                document.sForm.method = "POST";
```

```
            document.sForm.action = "bmt1_1_result.jsp";
            document.sForm.submit();

  };

</SCRIPT>

...
<button type="button" class="pure-button pure-input-1 pure-button-primary" onClick=sign()>전자서명 생성</button>
...
```

3.3.2. 전자서명 검증

```
<%@ page import="java.util.Vector" %>
<%@ page import="com.yettiesoft.javarose.standard.cert.SGCertificate" %>
<%@ page import="com.yettiesoft.vestsign.external.*" %>
<%@ page import="com.yettiesoft.vestsign.util.*" %>
<%@ page import="java.io.*" %>

<%
        String sm = request.getParameter("signedMsg");
        SignVerifier sv = new SignVerifier(sm, cert_verify, 1);
        sv.verify();
        int nVerifierResult = sv.getLastErrorCode();

                CertificateInfo   cert = sv.getSignerCertificate();
%>
..
전자서명 원문
<%=sv.getSignedMessageText()%>
사용자 인증서 정책
<%=cert.getPolicyIdentifier()%>
사용자 인증서 DN
<%=cert.getSubject()%>
사용자 인증서 serial
<%=cert.getSerial()%>
에러코드
<%=sv.getLastErrorCode()%>
검증결과
<%=sv.getLastErrorMsg()%>
```

3.3.3. MyData 검증

```
String strSignedMyData = “”; // mydata 서명값
String mobileUcpidNonce = "1cb04cd8163ea9e07898e1d2d961c12e";
String mobileConsentNonce = "djVJqSSmujAS...";
String transactionID = "abcd";

MyDataVerifier verifier = new MyDataVerifier();
UCPIDResult result = verifier.verify(strSignedMyData, mobileUcpidNonce,
                    mobileConsentNonce, transactionID);

if(verifier.getLastErrorCode() != 0){
        System.out.println("error code:"+verifier.getLastErrorCode());
        System.out.println("error Msg:"+verifier.getLastErrorMsg());
}
else{
        System.out.println("version            : " + result.getVersion());
        System.out.println("ucpidNonce         : " + result.getUcpidNonce());
        System.out.println("cpRequestNumber    : "+ result.getCpRequestNumber());
        System.out.println("certDn             : " + result.getCertDn());
        System.out.println("cpCode             : " + result.getCpCode());
        if (result.getVersion() == 1) {
                        System.out.println("ci             : " + result.getCi());
                        System.out.println("di             : " + result.getDi());
                        System.out.println("realName       : " + result.getRealName());
                        if (result.getGender() == UCPIDResult.GENDER_WOMAN){
                                System.out.println("gender             : " + "여성");
                        }
                        else if (result.getGender() == UCPIDResult.GENDER_MAN){
                                System.out.println("gender             : " + "남성");
                        }
                        else{
                                System.out.println("gender             : " + "정보없음");
                        }
                        if (result.getNationalInfo() == UCPIDResult.NATIONAL_LOCAL){
                                System.out.println("NationalInfo       : " + "내국인");
                        }
                        else if (result.getNationalInfo() ==
UCPIDResult.NATIONAL_FOREIGNER){
                                System.out.println("NationalInfo       : " + "외국인");
                        }
                        else{
                                System.out.println("NationalInfo       : " + "정보없음");
                        }
                        System.out.print("birthDate            : "+result.getBirthDate());
                } else if (result.getVersion() == 2) {
                        System.out.println("di                 : " + result.getDi());
                        System.out.println("realName           : " + result.getRealName());
                        if (result.getGender() == UCPIDResult.GENDER_WOMAN){
                                System.out.println("gender             : " + "여성");
                        }
                        else if (result.getGender() == UCPIDResult.GENDER_MAN){
                                System.out.println("gender             : " + "남성");
                        }
```

```
                        else{
                                System.out.println("gender              : " + "정보없음");
                        }

                        if (result.getNationalInfo() == UCPIDResult.NATIONAL_LOCAL){
                                System.out.println("NationalInfo        : " + "내국인");
                        }
                        else if (result.getNationalInfo() ==
UCPIDResult.NATIONAL_FOREIGNER){
                                System.out.println("NationalInfo        : " + "외국인");
                        }
                        else{
                                System.out.println("NationalInfo        : " + "정보없음");
                        }

                        System.out.println("birthDate           : "+result.getBirthDate());
                        System.out.println("ciUpdate            : "+result.getCiUpdate());
                        System.out.println("ci                  : "+result.getCi());
                        System.out.println("ci2                 : "+result.getCi2());
                }
```

4. 운영 가이드

4.1. 서버인증서 관리

VestSign 서버 모듈은 OCSP 서비스를 제공 하기 위하여 통신에 사용될 서버인증서를 인증기관으로부터 발급 받는다. 서버인증서는 1년의 유효기간을 가지고 있다. 1년마다 갱신하여야 한다.
갱신될 인증서는 아래의 경로에 위치한다.

구분	설명	위치
VestSign 데몬	서버 인증서	/패키지/cert/srvcert/signCert.der
	서버 개인키	/패키지/cert/srvcert/signPri.key
VestSign 라이브러리	서버 인증서	/패키지/cert/srvcert/signCert.der
	서버 개인키	/패키지/cert/srvcert/signPri.key

<표 10> VestSign 서버인증서 관리

4.2. 구동/종료

모듈	구동	종료
VestCert	C:₩Program Files (x86)₩VestCert ₩VestCert.exe	작업관리자 이용
VestSignServer	Jar 라이브러리로 구동 없음	N/A
VestSignValidator	/[패키지설치 디렉토리]/startvs.sh	/[패키지설치 디렉토리]/stopvs.sh

4.3. 오류코드

4.3.1. VestSign Library 오류코드

오류코드	CODE_NAME	설명
0	SUCCESS	[전자서명/유효성] 검증 성공
-10	CERT_EXPIRED	[전자서명/유효성] 인증서가 만료되었습니다.
-20	CERT_REVOKED	[전자서명/유효성] 인증서가 폐기되었습니다.

-30	UNKWON_ERROR	[전자서명/유효성] 기타 오류입니다.
-101	CONFIG_FILE_READ_ERROR	Config file read error.
-102	CONFIG_JSON_PARSING_ERROR	Config file json parsing error.
- 103	CONFIG_DECRYPT_PASSWORD_ERROR	Fail decrypt password.
- 151	ASN_INVALID_MESSAGE_SIZE_ERROR	Invalid message size.
- 152	ASN_ENCODED_ERROR	ASN Encoded error.
- 201	CHANNEL_NOT_EXIST_CHANNEL_ERROR	The connected channel does not exist.
- 202	CHANNEL_WRITE_ERROR	Channel write error.
- 203	CHANNEL_READ_ERROR	Channel read error.
- 211	CHANNEL_PASSED_BY_VALIDATOR_ERROR	Passed by validator error.
- 251	SV_CERT_HEX_OR_BASE64_DECODE_FAIL	Hex or Base64 Decode fail.
- 252	SV_CERT_SIGNED_MESSAGE_MISMATCH_ERROR	Signed message mismatch.
- 253	SV_CERT_INVALID_SIGNED_DATA_TYPE_ERROR	SignedData validation check error.
- 254	SV_CERT_INVALID_CONTENT_INFO_TYPE_ERROR	ContentInfo validation check error.
- 255	SV_CERT_VALID_CHECK_ERROR	Certification validation check error.
- 256	SV_CERT_NOT_SUPPORTED_CERTIFICATE_STATUS_OPTION	Not supported certificate status option.
- 257	SV_CERT_IO_ERROR	Certificate io error.
- 258	SV_CERT_UNSUPPORTED_ALGORITHM	Unsupported Algorithm.
- 259	SV_FILE_READ_ERROR	file read error.
- 260	SV_FILE_WRITE_ERROR	file write error.
- 301	VV_PRIVATEKEY_LOADING_ERROR	private key file loading fail.
- 302	VV_CERTIFICATE_LOADING_ERROR	Certificate loading fail.
- 303	VV_VID_MESSGAE_DECRYPT_SESSION_KEY	Decrypt session error.
- 304	VV_VID_MESSGAE_DEENVELOPE_ERROR	vid message deenvelope error.
- 305	VV_VID_MESSGAE_DECODE_RANDOM_ERROR	vid message decode random error.
- 306	VV_VID_MESSGAE_GET_HASH_ALGORITHM	Get Hash Algorithm error.
-307	VV_VID_MESSGAE_HASH_CONTENTS_ENCODE_ERROR	Hash Contents encode error.

-308	VV_VID_MESSGAE_HASH_ERROR	Hash error.
-309	VV_VID_MESSGAE_NOT_SUPPORTED_ALGORITHM	Not supported algorithm.
-310	VV_VID_MESSGAE_MISMATCH_ERROR	vid message mismatch.
-351	CV_CERT_HEX_OR_BASE64_DECODE_FAIL	Hex or Base64 Decode fail.
-352	CV_CERT_PEM_DECODE_FAIL	Certificate Pem Decode fail.
-401	ENV_ENVELOPE_ERROR	message envelope error.
-402	ENV_DEENVELOPE_ERROR	env message deEnvelope error.
-403	ENV_NOT_ENV_DATA_ERROR	ContentType is not envelopedData.
-404	ENV_INPUT_MSG_NULL	Input message is null.
-405	ENV_HEX_OR_BASE64_DECODE_FAIL	Hex or Base64 Decode fail.
-406	ENV_FILE_READ_ERROR	file read error.
-407	ENV_FILE_WRITE_ERROR	file write error.
-421	CIP_ENCRYPT_ERROR	cipher encrypt error.
-422	CIP_DECRYPT_ERROR	cipher decrypt error.
-423	CIP_FILE_READ_ERROR	file read error.
-424	CIP_FILE_WRITE_ERROR	file write error.
-901	ETC_HASH_ALGORITHM_ERROR	unsupported hash algorithm.
-902	ETC_HASH_ERROR	hash error.
-999	UNKNOWN_EXCEPTION	Unknown Exception.

<표 11> VestSign 서버 오류 코드

4.3.2. VestSignValidator 오류코드

오류코드	CODE_NAME	설명
-1001	CONFIG_FILE_READ_ERROR	Config file read error.
-1002	CONFIG_JSON_PARSING_ERROR	Config file json parsing error.
-1011	CONFIG_CERTIFICATE_LOADING_ERROR	Certificate loading fail.
-1012	CONFIG_DECRYPT_PASSWORD_ERROR	Fail decrypt password.
-1013	CONFIG_PRIVATEKEY_LOADING_ERROR	private key file loading fail.

-1051	ASN_INVALID_MESSAGE_SIZE_ERROR	Invalid message size.
-1052	ASN_ENCODED_ERROR	ASN Encoded error.
-1101	CHANNEL_FAIL_CREATE_SECURECHANNEL_ERROR	Fail create securechannel.
-1102	CHANNEL_SECURE_MESSAGE_INPUT_IS_NULL_ERROR	Secure Message input is null.
-1103	CHANNEL_INVALID_MESSAGE_TYPE_ERROR	Invalid message type.
-1151	SERVICE_OCSP_ERROR	OCSP Service Error.
-1152	SERVICE_CRL_ERROR	CRL Service Error.
-1153	SERVICE_UCPID_ERROR	UCPID Service Error.
-1164	SERVICE_CERTIFICATE_EXTRACT_CERTIFICATE_ERROR	Extract Certificate error
-1165	SERVICE_CERTIFICATE_VALID_CHECK_ERROR	Certification validation check error
-1166	SERVICE_CERTIFICATE_PARSING_ERROR	Certificate parsing error
-1167	SERVICE_PKCS7_MESSAGE_PARSING_ERROR	PKCS7(CMS) Message Parsing error.
-1168	SERVICE_SIGNED_MESSAGE_VALIDATION_ERROR	Signed message hash algorithm error.
-1169	SERVICE_ISSUER_CERT_NOT_FOUND_ERROR	Issuer Certificate not found error.
-1999	UNKNOWN_EXCEPTION	Unknown Exception.

<표 12> VestSign Validator 데몬 오류 코드

4.3.3. 클라이언트 오류코드

구분	정의	에러 코드	에러 설명
공통	ServiceError_INVALID_INPUT	10000	입력 값이 잘못되었습니다.
토큰	ServiceError_TOKEN_NOT_INITIALIZED	11000	보안디스크가 초기화 되지 않았습니다.
	ServiceError_TOKEN_NOT_FOUND	11001	보안디스크가 존재하지 않습니다.
	ServiceError_TOKEN_BAD	11002	보안디스크의 상태가 비정상입니다. 초기화 하세요.
	ServiceError_TOKEN_UBIKEY_NOT_INSTALLED	11003	Ubikey 가 설치되지 않았습니다. 프로그램을 설치 해주세요.
	ServiceError_TOKEN_UBIKEY_NOT_LATE	11004	Ubikey 가 최신 버전이 아닙니다.

	ST_VERSION		프로그램을 업데이트 해주세요.
	ServiceError_TOKEN_UBIKEY_INVALID_OPTIONS	11005	Ubikey 옵션이 유효하지 않습니다.
SSL	ServiceError_SSLCONFIG_SERVICE_SSL_INIT_FAILED	11200	SSL 서비스 초기화에 실패하였습니다.
서비스 공통	ServiceError_SERVICE_REJECTED	11500	올바른 MangoWire 메시지가 아니므로, 서비스가 거절되었습니다.
	ServiceError_SESSIONID_NOT_EXIST	11501	세션이 만료되었거나 잘못되었습니다. 다시 접속하세요.
	ServiceError_SESSION_IS_USING	11502	다른 곳에서 세션이 사용 중입니다. 다시 접속하세요.
	ServiceError_OPERATION_NOT_SUPPORTED	11503	지원되지 않는 기능입니다.
	ServiceError_OPERATION_NOT_EXPECTED	11504	현재 이 기능을 수행할 수 없습니다.
	ServiceError_INVALID_INPUT_TOKENID	11505	토큰 식별자가 잘못되었습니다.
	ServiceError_MEMORY_ALLOCATION_FAILED	11506	메모리 할당에 실패했습니다.
	ServiceError_NO_SSL_CERTIFICATE	11507	등록된 SSL 인증서가 존재하지 않습니다.
서비스	ServiceError_CERTIFICATE_NOT_FOUND	11508	인증서가 존재하지 않습니다.
인증서 삭제	ServiceError_DELETE_CERTIFICATE_FAILED	11509	인증서 삭제에 실패했습니다.(기타 에러)
	ServiceError_DELETE_CERTIFICATE_INVALID_CERTINDENTIFIER	11510	입력값이 잘못 되었습니다.
	ServiceError_DELETE_PROGRAM_FILES_PATH_DELETE_WARNING	11511	Program files 에 저장된 인증서는 삭제할 수 없습니다.
	ServiceError_DELETE_PASSWORD_INCORRECT	11512	인증서 삭제에 실패했습니다(비밀번호를 확인하세요).
encrypt vid random	ServiceError_ENCRYPT_VIDRANDOM_INVALID_CERTINDENTIFIER	11513	입력값이 잘못 되었습니다.
	ServiceError_ENCRYPT_VIDRANDOM_FAILED	11514	EncryptVIDRandom failed.
	ServiceError_ENCRYPT_VIDRANDOM_TOKEN_NOT_INITIALIZE	11515	보안디스크가 초기화 되지 않았습니다
키쌍 생성	ServiceError_GENERATE_KEYPAIR_INVALID_ARGUMENT	11516	입력값이 잘못 되었습니다.
	ServiceError_GENERATE_KEYPAIR_FAILED	11517	Gen key fail

	ServiceError_GENERATE_KEYPAIR_TOKEN_NOT_INITIALIZE	11518	보안디스크가 초기화 되지 않았습니다.
서명 생성	ServiceError_GENERATE_SIGNATURE_NOT_EXPECTED_KEYIDENTIFIER	12017	not expected keyIdentifier
	ServiceError_GENERATE_SIGNATURE_FAILED	12018	전자서명에 실패하였습니다.
	ServiceError_GENERATE_SIGNATURE_TOKEN_NOT_INITAILIZE	12019	보안디스크가 초기화 되지 않았습니다.
	ServiceError_GENERATE_SIGNATURE_FAILED_PIN_INCORRECT	12020	전자서명에 실패하였습니다(비밀번호를 확인하세요).
	ServiceError_GENERATE_SIGNATURE_FAILED_PIN_LOCKED	12021	전자서명에 실패하였습니다(장치가 잠겼습니다.
	ServiceError_GENERATE_SIGNATURE_FAILED_SGPKCS8_PRIVATEKEYINFO_DECODE_FAILED	12022	전자서명에 실패하였습니다(비밀번호를 확인하세요).
	ServiceError_GENERATE_SIGNATURE_ENCRYPT_VIDRANDOM_FAILED	12023	전자서명에 실패하였습니다(식별번호 생성에 실패
	ServiceError_GENERATE_SIGNATURE_INVALID_ARGUMENT	12024	입력값이 잘못되었습니다.
	ServiceError_GENERATE_SIGNATURE_CANCELED	12025	인증서 가져오기를 실패하였습니다.
	ServiceError_GENERATE_SIGNATURE_KSTOKEN_PIN_INCORRECT	12026	통합보안토큰의 패스워드가 잘못되었습니다.
	ServiceError_GENERATE_SIGNATURE_KOSCOM_SIGN_MUST_HAVE_CERTIFICATE	12027	Koscom 서명은 인증서가 있어야합니다.
GetCertificateList	ServiceError_GET_CERTIFICATE_LIST_FAILED	12028	Function Failed
	ServiceError_GET_CERTIFICATE_LIST_TOKEN_NOT_INITIALIZE	12030	보안디스크가 초기화 되지 않았습니다.
GetCertificate	ServiceError_GET_CERTIFICATE_INVALID_ARGUMENT	12100	입력값이 잘못되었습니다.
	ServiceError_GET_CERTIFICATE_FAILED	12101	인증서 가져오기를 실패하였습니다.
	ServiceError_GET_CERTIFICATE_NOT_FOUND	12102	인증서를 찾을 수 없습니다.
	ServiceError_GET_CERTIFICATE_TOKEN_NOT_INITIALIZE	12103	보안디스크가 초기화 되지 않았습니다.
setMatchedContext	ServiceError_SETMATCHED_CONTEXT_INPUT_CANCELED	12200	비밀번호 입력을 취소했습니다.

	ServiceError_SETMATCHED_CONTEXT_INVALID_CUSTOM_SID	12201	잘못된 session ID 가 입력되었습니다."
	ServiceError_SETMATCHED_CONTEXT_CUSTOM_SID_IS_NULL	12202	session ID 가 NULL 로 입력되었습니다.
	ServiceError_SETMATCHED_CONTEXT_CREATE_SESSION_UNIT_FAILED	12203	session 생성에 실패했습니다.
GetCACertificate	ServiceError_GET_CA_CERTIFICATE_INVALID_ARGUMENT	12300	입력값이 잘못되었습니다.
PushCertificate	ServiceError_PUSH_CERTIFICATE_INVALID_ARGUMENT	12400	입력값이 잘못 되었습니다.
	ServiceError_PUSH_CERTIFICATE_NOT_EXPECTED_KEYIDENTIFIER	12401	not expected keyIdentifier
	ServiceError_PUSH_CERTIFICATE_FAILED	12402	PushCertificate failed.
	ServiceError_PUSH_CERTIFICATE_TOKEN_NOT_INITIALIZE	12403	보안디스크가 초기화 되지 않았습니다.
인증서 검증	ServiceError_VERIFY_CERTIFICATE_FAILED	12500	인증서 비밀번호 확인에 실패했습니다.(기타 에러)
	ServiceError_VERIFY_CERTIFICATE_INVALID_CERTINDENTIFIER	12501	입력값이 잘못 되었습니다.
	ServiceError_VERIFY_PASSWORD_INCORRECT	12502	패스워드가 틀립니다.
서명문 토큰 생성	ServiceError_GENERATE_SIGNATURE_TOKEN_PIN_INCORRECT	12600	전자서명에 실패하였습니다(비밀번호를 확인하세요)
	ServiceError_GENERATE_SIGNATURE_TOKEN_CERT_PIN_INCORRECT	12601	전자서명에 실패하였습니다(비밀번호를 확인하세요)
	ServiceError_GENERATE_SIGNATURE_TOKEN_PIN_LOCKED	12602	전자서명에 실패하였습니다(장치가 잠겼습니다)
	ServiceError_GENERATE_SIGNATURE_TOKEN_CERT_PIN_LOCKED	12603	전자서명에 실패하였습니다(장치가 잠겼습니다)
키쌍 생성	ServiceError_GENERATE_KEYPAIR_CANCELLED	12700	키쌍 생성이 사용자에 의해 취소되었습니다.
	ServiceError_GENERATE_KEYPAIR_PIN_INCORRECT	12701	비밀번호가 잘못되어 키쌍 생성에 실패하였습니다.
	ServiceError_GENERATE_KEYPAIR_PIN_LOCKED	12702	비밀번호가 잘못되어 장치가 잠겨, 키쌍 생성에 실패하였습니다.
CMP 공통	ServiceError_CMP_MEMORY_ALLOCATION_FAILED	12800	메모리 할당에 실패했습니다.

	ServiceError_CMP_SERVER_CONNECT_FAILED	12801	서버에 접속할 수 없습니다.
인증서 발급	ServiceError_CMP_ISSUE_INVALID_ARGUMENT	12900	인증서 발급에 대한 입력값이 잘못 되었습니다.
	ServiceError_CMP_ISSUE_NOT_SUPPORTED_CA	12901	지원되지 않는 인증 기관 코드가 입력되었습니다.
	ServiceError_CMP_ISSUE_INPUTPIN_CANCELED	12902	비밀번호 입력을 취소했습니다.
	ServiceError_CMP_ISSUE_PKCS5_ENCRYPT_FAILED	12903	PKCS#5 암호화에 실패했습니다.
	ServiceError_CMP_ISSUE_MAKE_ENCRYPTED_PRIVATEKEY_INFO_FAILED	12904	PKCS#8 메시지 구성에 실패했습니다.
	ServiceError_CMP_ISSUE_SAVE_CERTIFICATE_FAILED	12905	인증서 저장에 실패했습니다.
	ServiceError_CMP_ISSUE_IMPORT_INIT_FAILED	12906	인증서 가져오기 초기화에 실패했습니다.
	ServiceError_CMP_ISSUE_IMPORT_SIGN_CERTIFICATE_IMPORT_FAILED	12907	서명용 인증서 가져오기에 실패했습니다.
	ServiceError_CMP_ISSUE_IMPORT_KM_CERTIFICATE_IMPORT_FAILED	12908	암호용 인증서 가져오기에 실패했습니다.
	ServiceError_CMP_ISSUE_IMPORT_CA_PUB_IMPORT_FAILED	12909	인증기관 인증서 가져오기에 실패하였습니다.
	ServiceError_CMP_ISSUE_IMPORT_FINAL_FAILED	12910	인증서 가져오기에 실패하였습니다.
	ServiceError_CMP_ISSUE_NOT_SUPPORTED_BILL	12911	과금 발급은 현재 지원되지 않습니다.
인증서 갱신	ServiceError_CMP_UPDATE_INVALID_ARGUMENT	13000	인증서 갱신에 대한 입력값이 잘못 되었습니다.
	ServiceError_CMP_UPDATE_NOT_SUPPORTED_CA	13001	지원되지 않는 인증 기관 코드가 입력되었습니다.
	ServiceError_CMP_UPDATE_INPUTPIN_CANCELED	13002	비밀번호 입력을 취소했습니다.
	ServiceError_CMP_UPDATE_EXPORT_CERTIFICATE_AND_KEY_FAILED	13003	갱신할 인증서를 가져오는데 실패했습니다.
	ServiceError_CMP_UPDATE_ADD_OLD_CERTIFICATE_FAILED	13004	갱신할 인증서를 추가하는데 실패했습니다.
	ServiceError_CMP_UPDATE_ADD_OLD_KEY_FAILED	13005	갱신할 키파일을 추가하는데 실패했습니다.
	ServiceError_CMP_UPDATE_PKCS5_ENC	13006	PKCS#5 암호화에 실패했습니다.

	RYPT_FAILED		
	ServiceError_CMP_UPDATE_MAKE_ENCRYPTED_PRIVATEKEY_INFO_FAILED	13007	PKCS#8 메시지 구성에 실패했습니다.
	ServiceError_CMP_UPDATE_SAVE_CERTIFICATE_FAILED	13008	인증서 저장에 실패했습니다.
	ServiceError_CMP_UPDATE_IMPORT_INIT_FAILED	13009	인증서 가져오기 초기화에 실패했습니다.
	ServiceError_CMP_UPDATE_IMPORT_SIGN_CERTIFICATE_IMPORT_FAILED	13010	서명용 인증서 가져오기에 실패했습니다.
	ServiceError_CMP_UPDATE_IMPORT_KM_CERTIFICATE_IMPORT_FAILED	13011	암호용 인증서 가져오기에 실패했습니다.
	ServiceError_CMP_UPDATE_IMPORT_CA_PUB_IMPORT_FAILED	13012	인증기관 인증서 가져오기에 실패하였습니다.
	ServiceError_CMP_UPDATE_IMPORT_FINAL_FAILED	13013	인증서 가져오기에 실패하였습니다.
	ServiceError_CMP_UPDATE_NOT_SUPPORTED_BILL	13014	과금 갱신은 현재 지원되지 않습니다.
인증서 폐기	ServiceError_CMP_REVOKE_INVALID_ARGUMENT	13100	인증서 폐기에 대한 입력값이 잘못 되었습니다.
	ServiceError_CMP_REVOKE_NOT_SUPPORTED_CA	13101	지원되지 않는 인증 기관 코드가 입력되었습니다.
	ServiceError_CMP_REVOKE_INPUTPIN_CANCELED	13102	비밀번호 입력을 취소했습니다.
	ServiceError_CMP_REVOKE_EXPORT_CERTIFICATE_AND_KEY_FAILED	13103	폐기할 인증서를 가져오는데 실패했습니다.
	ServiceError_CMP_REVOKE_ADD_OLD_CERTIFICATE_FAILED	13104	폐기할 인증서를 추가하는데 실패했습니다.
	ServiceError_CMP_REVOKE_ADD_OLD_KEY_FAILED	13105	폐기할 키파일을 추가하는데 실패했습니다.
PC 정보 수집	ServiceError_GET_PCIDENTITY_FAILED_MEMORY_ALLOCATION_FAILED	13200	메모리 할당에 실패했습니다.
	ServiceError_GET_PCIDENTITY_FAILED_INVALID_WINDOWS	13201	단말 식별 값을 가져오지 못했습니다(Windows 외 타 OS 는 추후 지원합니다).
	ServiceError_GET_PCIDENTITY_FAILED	13202	단말 식별 값을 가져오지 못했습니다(기타 에러).
비밀번호 변경(인증서 관리)	ServiceError_CHANGE_PIN_FAILED_INVALID_CERTINDENTIFIER	13300	입력값이 잘못 되었습니다.
	ServiceError_CHANGE_PIN_FAILED_INPU	13301	비밀번호 입력을 취소했습니다.

	T_CANCELED		
	ServiceError_CHANGE_PIN_FAILED_INVALID_CERT_TYPE	13302	비밀번호 변경에 실패했습니다(인증서 형식에 문제가 발생했습니다).
	ServiceError_CHANGE_PIN_FAILED_PIN_INCORRECT	13303	비밀번호 변경에 실패했습니다(비밀번호를 확인하세요).
	ServiceError_CHANGE_PIN_FAILED_FILE_WRITE_ERROR	13304	비밀번호 변경에 실패했습니다(인증서를 저장할 때 문제가 발생했습니다).
	ServiceError_CHANGE_PIN_FAILED	13305	비밀번호 변경에 실패하였습니다(기타 에러).
인증서 내보내기(인증서 관리)	ServiceError_EXPORT_CERTIFICATE_FAILED_INPUT_CANCELED	13400	비밀번호 입력을 취소했습니다.
	ServiceError_EXPORT_CERTIFICATE_FAILED_SELECT_CANCELED	13401	인증서 내보내기를 취소했습니다.
	ServiceError_EXPORT_CERTIFICATE_FAILED_INVALID_CERT_TYPE	13402	인증서 내보내기에 실패했습니다(인증서 형식에 문제가 발생했습니다).
	ServiceError_EXPORT_CERTIFICATE_FAILED_SEARCH_CERTIFICATE_FAILED	13403	인증서 내보내기에 실패했습니다(인증서를 찾지 못했습니다).
	ServiceError_EXPORT_CERTIFICATE_FAILED_PIN_INCORRECT	13404	인증서 내보내기에 실패했습니다(비밀번호를 확인하세요).
	ServiceError_EXPORT_CERTIFICATE_FAILED_ADD_CERTIFICATELIST_FAILED	13405	인증서 내보내기에 실패했습니다(add certificate fail).
	ServiceError_EXPORT_CERTIFICATE_FAILED_ENCODE_PFX_FAILED	13406	인증서 내보내기에 실패했습니다(encode pfx fail).
	ServiceError_EXPORT_CERTIFICATE_FAILED	13407	인증서 내보내기에 실패했습니다(기타 에러).
인증서 가져오기(인증서 관리)	ServiceError_IMPORT_CERTIFICATE_FAILED_SELECT_CANCELED	13500	인증서 가져오기에 실패했습니다(인증서 선택을 취소했습니다).
	ServiceError_IMPORT_CERTIFICATE_FAILED_INPUT_CANCELED	13501	인증서 가져오기에 실패했습니다(비밀번호 입력을 취소했습니다).
	ServiceError_IMPORT_CERTIFICATE_FAILED_INVALID_PFX	13502	인증서 가져오기에 실패했습니다(PFX 형식의 인증서가 아닙니다).
	ServiceError_IMPORT_CERTIFICATE_FAILED_INVALID_PFX_PASSWORD	13503	인증서 가져오기에 실패했습니다(비밀번호를 확인하세요).
	ServiceError_IMPORT_CERTIFICATE_FAIL	13504	인증서 가져오기에 실패했습니다(기타

	ED		에러).
PIN 검증	ServiceError_VERIFY_PIN_FAILED_INVALID_CERTINDENTIFIER	13600	유효하지 않은 CERTINDENTIFIER 입니다.
	ServiceError_VERIFY_PIN_FAILED_INPUT_CANCELED	13601	비밀번로 입력이 취소되었습니다.
	ServiceError_VERIFY_PIN_FAILED	13602	비밀번호 확인에 실패했습니다(비밀번호를 확인하세요).
저장매체 변경	ServiceError_CHANGE_STORAGE_FAILED_INVALID_CERTINDENTIFIER	13700	입력값이 잘못 되었습니다.
	ServiceError_CHANGE_STORAGE_FAILED_INVALID_TOKENINDENTIFIER	13701	입력된 매체는 사용할 수 없는 매체입니다.
	ServiceError_CHANGE_STORAGE_FAILED_INPUT_CANCELED	13702	인증서 저장매체 변경에 실패했습니다(비밀번호 입력을 취소했습니다).
	ServiceError_CHANGE_STORAGE_FAILED_CERTIFICATE_AND_KEY_FAILED	13703	인증서 저장매체 변경에 실패했습니다.
	ServiceError_CHANGE_STORAGE_FAILED_PIN_INCORRECT	13704	인증서 저장매체 변경에 실패했습니다(비밀번호를 확인하세요).
	ServiceError_CHANGE_STORAGE_SAME_TOKEN	13705	변경할 인증서 저장매체가 같습니다.
	ServiceError_CHANGE_STORAGE_FAILED	13706	인증서 저장매체 변경에 실패했습니다(기타 에러)
인증서검증	ServiceError_VALIDATE_CERTIFICATE_INVALID_CERTINDENTIFIER	13800	입력값이 잘못 되었습니다.
	ServiceError_VALIDATE_CERTIFICATE_INVALID_CERTIFICATE	13801	인증서 형식이 잘못되었습니다.
	ServiceError_VALIDATE_CERTIFICATE_CRL_FAILED	13802	인증서 검증에 실패했습니다.
	ServiceError_VALIDATE_CERTIFICATE_FAILED	13803	인증서 유효성 검증에 실패했습니다(기타 에러).
세션 관리	ServiceError_SESSION_MANAGER_SESSION_ID_IS_NULL	13900	session id 가 없어 session 저장에 실패했습니다.
트레이 아이콘 관리	ServiceError_OPERATE_TRAY_INVALID_TRAY_VENDOR	14000	잘못된 tray 목록입니다.
	ServiceError_OPERATE_TRAY_INVALID_TRAY_OPERATE	14001	잘못된 tray operate 동작입니다.
서명 검증	ServiceError_VERIFY_SIGNATURE_INVALID_ARGUMENT	14100	입력값이 잘못되었습니다.
	ServiceError_VERIFY_SIGNATURE_PLAIN_	14101	원문이 필요한 전자서명입니다.

	IS_NULL		
	ServiceError_VERIFY_SIGNATURE_UNSUPPORT_SIGNTYPE	14102	아직 지원되지 않는 전자서명입니다.
	ServiceError_VERIFY_SIGNATURE_INVALID_X509_TYPE	14103	X509 인증서형태가 아닙니다.
	ServiceError_VERIFY_SIGNATURE_INVALID_PUBLIC_KEY_TYPE	14104	public key 형태가 아닙니다.
	ServiceError_VERIFY_SIGNATURE_VERIFY_FAILED	14105	서명검증에 실패했습니다.
VID 검증	ServiceError_VERIFY_VID_INVALID_CERTID	14200	입력값이 잘못되었습니다.
	ServiceError_VERIFY_VID_INVALID_KEYID	14201	입력값이 잘못되었습니다.
	ServiceError_VERIFY_VID_INVALID_IDN	14202	입력값이 잘못되었습니다.
	ServiceError_VERIFY_VID_TOKEN_NOT_INITIALIZE	14203	보안디스크가 초기화 되지 않았습니다.
	ServiceError_VERIFY_VID_NOT_FOUND	14204	입력값이 잘못되었습니다.
	ServiceError_VERIFY_VID_NOT_INVALID_X509_TYPE	14205	X509 인증서형태가 아닙니다.
	ServiceError_VERIFY_VID_GET_RANDOM_FAILED	14206	random 을 가져오는데 실패했습니다.
	ServiceError_VERIFY_VID_VERIFY_FAILED	14207	VID 검증에 실패했습니다.
Hash	ServiceError_GET_HASH_FAILED_INVALID_INPUT	14300	유효하지 않은 입력입니다.
	ServiceError_GET_HASH_FAILED_INVALID_ALGORITHM	14301	유효하지 않은 알고리즘입니다.
	ServiceError_GET_HASH_FAILED_UNSUPPORTED_DIGEST_ALGORITHM	14302	지원하지 않은 hash 알고리즘입니다.
	ServiceError_GET_HASH_FAILED	14303	HASH 실패하였습니다.
암호화	ServiceError_ENCRYPT_FAILED_INVALID_INPUT	14400	유효하지 않은 입력입니다.
	ServiceError_ENCRYPT_FAILED_KEY_IS_NULL	14401	암호화 키가 null 입니다.
	ServiceError_ENCRYPT_FAILED_IV_IS_NULL	14402	암호화 IV 가 유효하지 않습니다.
	ServiceError_ENCRYPT_FAILED_UNSUPPORTED_KEY_LEN	14403	지원하지 않는 키 길이 입니다.
	ServiceError_ENCRYPT_FAILED_UNSUPPORTED_ALGORITHM	14404	지원하지 않은 알고리즘입니다.

	ServiceError_ENCRYPT_FAILED_UNSUPPORTED_MODE	14405	지원하지 않는 암호화 모드입니다.
	ServiceError_ENCRYPT_FAILED	14406	암호화에 실패하였습니다.
복호화	ServiceError_DECRYPT_FAILED_INVALID_INPUT	14500	유효하지 않은 입력입니다.
	ServiceError_DECRYPT_FAILED_KEY_IS_NULL	14501	복호화 키가 null 입니다.
	ServiceError_DECRYPT_FAILED_IV_IS_NULL	14502	복호화 IV 가 유효하지 않습니다.
	ServiceError_DECRYPT_FAILED_UNSUPPORTED_KEY_LEN	14503	지원하지 않는 키 길이 입니다.
	ServiceError_DECRYPT_FAILED_UNSUPPORTED_ALGORITHM	14504	지원하지 않은 알고리즘입니다.
	ServiceError_DECRYPT_FAILED_UNSUPPORTED_MODE	14505	지원하지 않는 복호화 모드입니다.
	ServiceError_DECRYPT_FAILED	14506	복호화에 실패하였습니다.
envelope	ServiceError_ENVELOPE_FAILED_INVALID_INPUT	14600	전자봉투 입력 값이 유효하지 않습니다.
deenvelope	ServiceError_DEENVELOPE_FAILED_INVALID_INPUT	14700	전자봉투 복호화의 입력값이 유효하지 않습니다.
키보드보안	ServiceError_KEYBOARDPROTECTION_INVALID_ARGUMENT	20000	입력값이 잘못되었습니다.
	ServiceError_KEYBOARDPROTECTION_CREATE_FAILED	20001	키보드보안 연동 실패했습니다.
	ServiceError_KEYBOARDPROTECTION_INIT_FAILED	20002	키보드보안 초기화 실패했습니다.
	ServiceError_KEYBOARDPROTECTION_GETPIN_FAILED	20003	입력값 획득에 실패 했습니다.(키보드보안 연동 오류)
	ServiceError_KEYBOARDPROTECTION_GETPUBLICKEY_FAILED	20004	키보드보안 공개키 획득에 실패했습니다.
mobile USIM	ServiceError_MOBILE_USIM_NOT_PRESENT	21000	유심칩이 존재하지 않습니다.
	ServiceError_TOKEN_MOBILE_USIM_INVALID_OPTIONS	21001	전자봉투를 위한 입력값이 유효하지 않습니다.
기타	ServiceError_UNKNOWN	21002	아직 등록되지 않은 오류코드

<표 13> VestSign 클라이언트 오류 코드

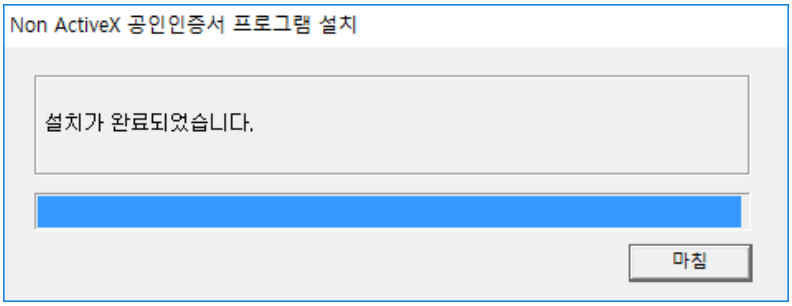
5. 이용 가이드

5.1. VestCert 설치 방법

다음은 하드디스크나 이동식디스크, 보안매체 등을 이용한 전자 서명문 생성 시 필요한 VestCert에 대한 실행 방법을 나타낸다. 먼저, 전자서명을 위한 매체로 보안디스크, 하드디스크, 이동식 디스크 또는 보안토큰을 선택한 경우 해당 동작을 수행할 수 있는 프로그램이 실행되고 있지 않다면 <그림 4>와 같이 토큰 관리자 실행 안내 메시지 창이 표시된다. 설치파일 다운로드 후 <그림 5>와 같이 설치를 진행한 후, 사용자의 동의가 이루어 진 경우 <그림 6>과 같이 해당 프로그램이 실행된다. <그림 7>이 후 사용자는 타 매체와 유사한 형태로 공인인증서를 사용할 수 있게 된다.

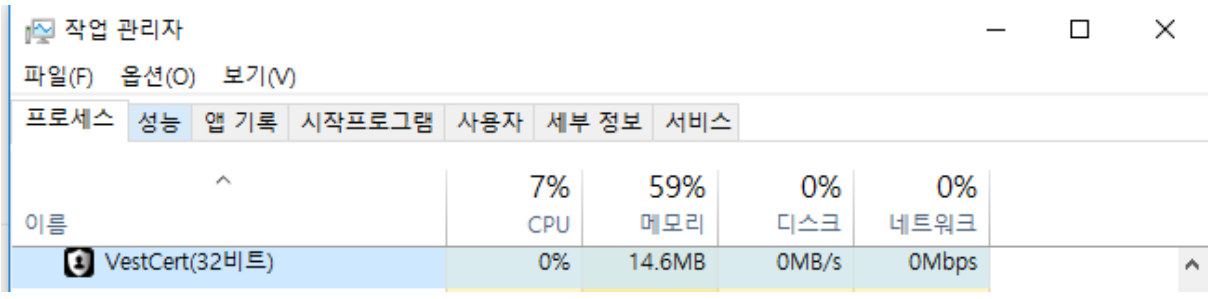
[Windows]

<그림 4> VestCert 구동되지 않은 경우



<그림 5> VestCert 설치 화면

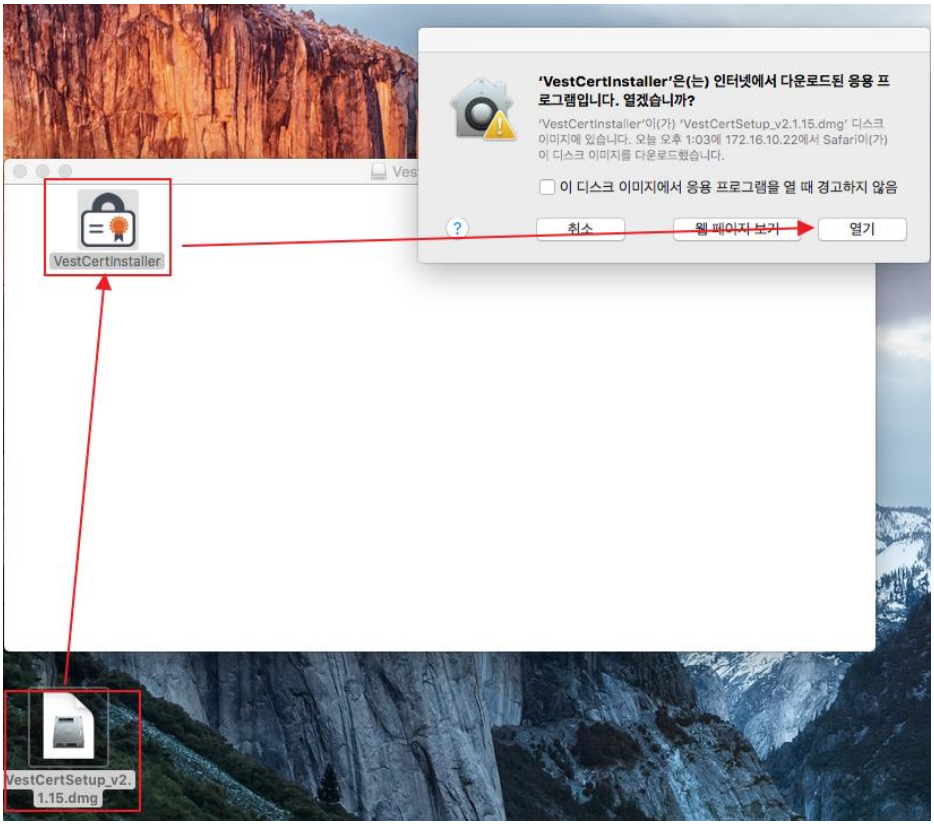
<그림 6> VestCert 설치 후 인증서 로딩 화면



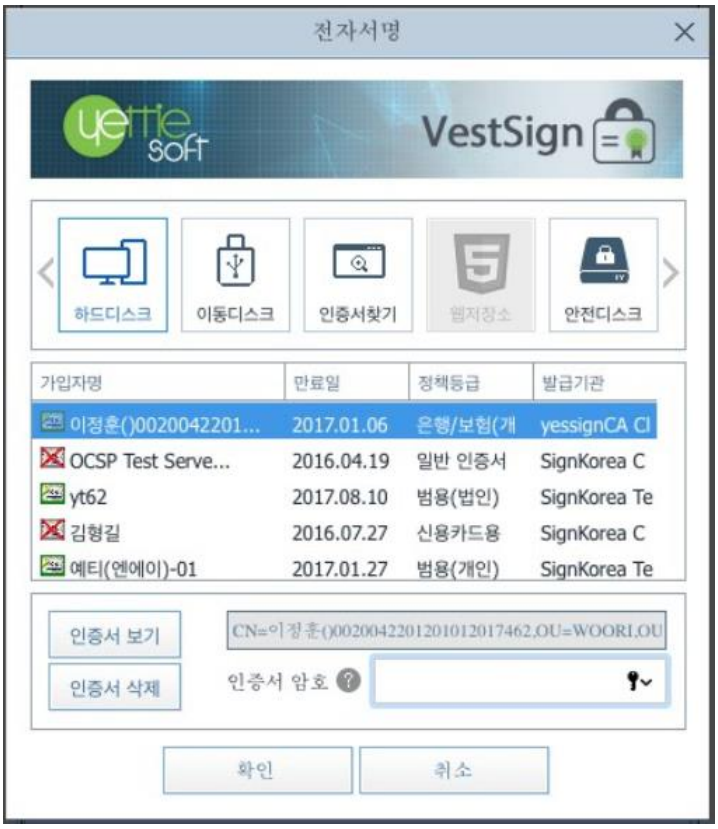
<그림 7> VestCert 구동 확인

[MAC]

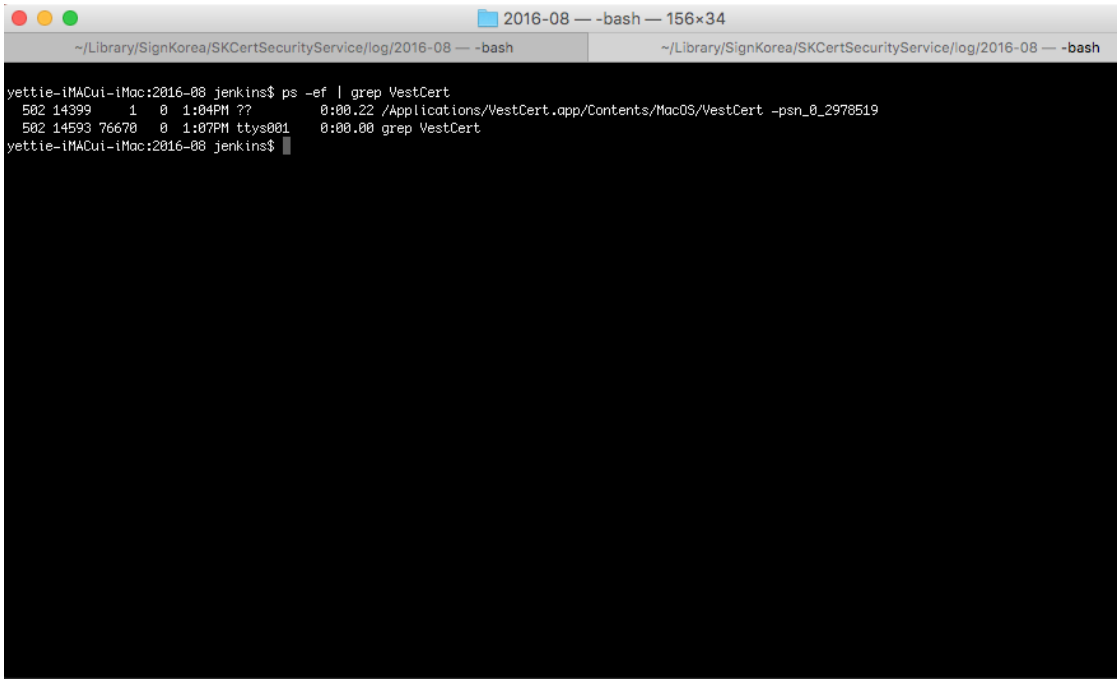
<그림 8> [MAC] VestCert 구동되지 않은 경우 다운로드



<그림 9> [MAC] 다운로드 및 설치



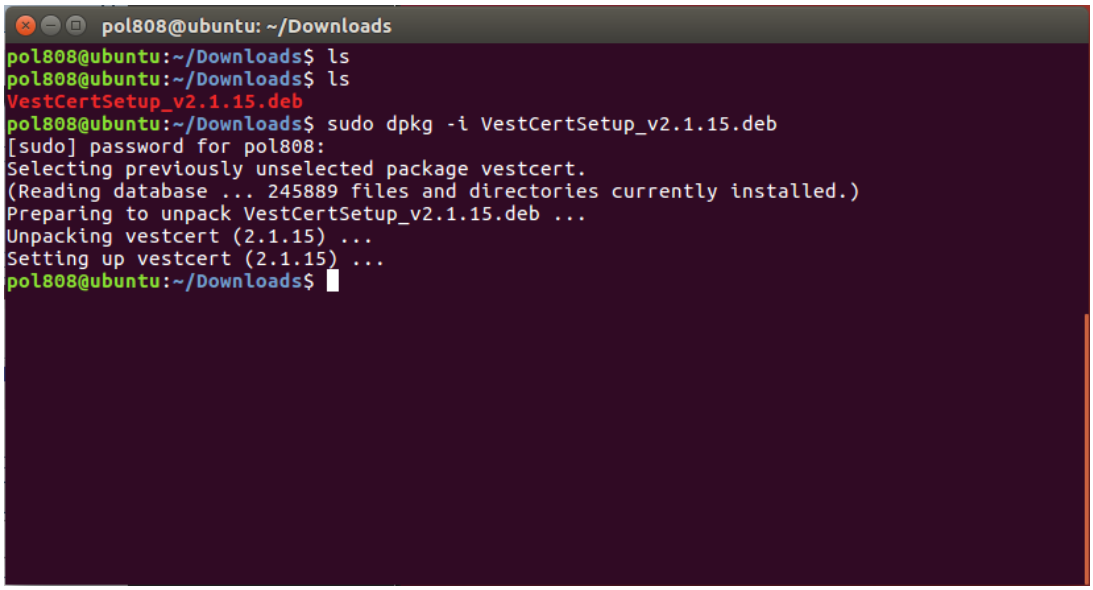
<그림 10> [MAC] VestCert 설치 후 인증서 로딩 화면



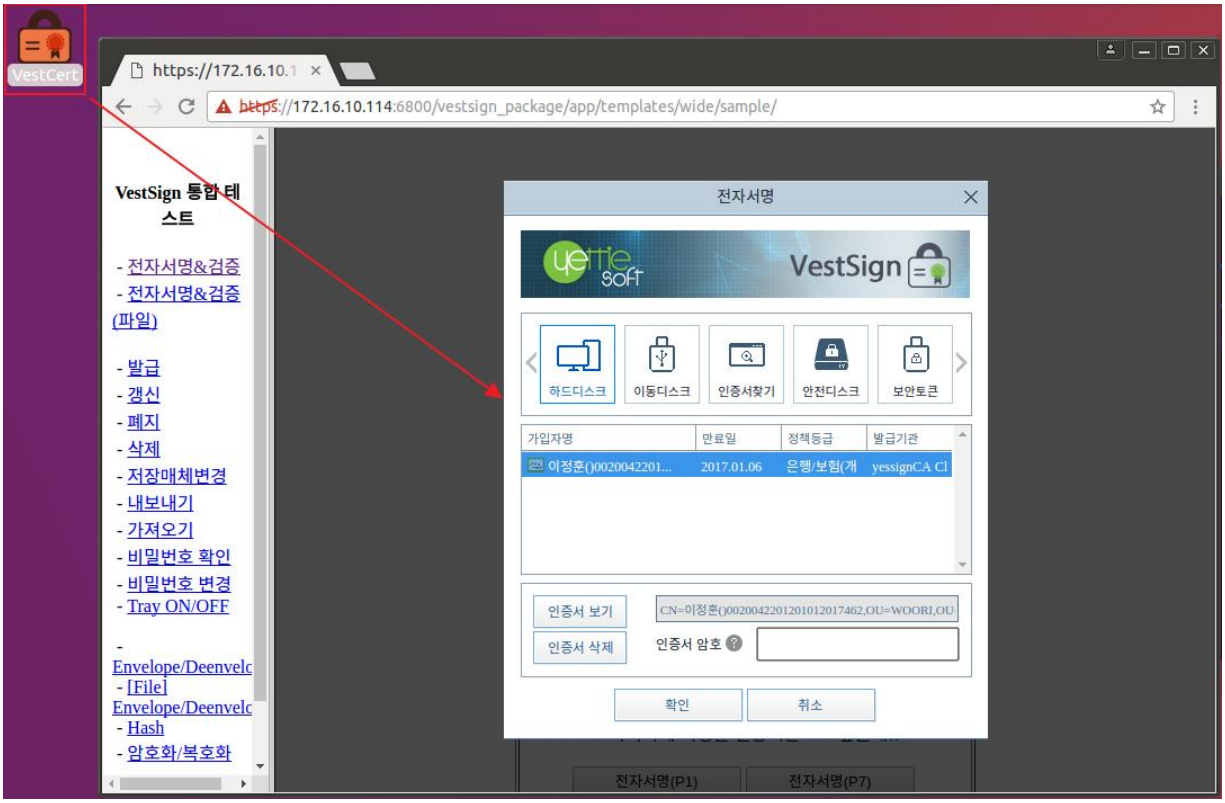
<그림 11> [MAC] VestCert 구동 확인

[LINUX]

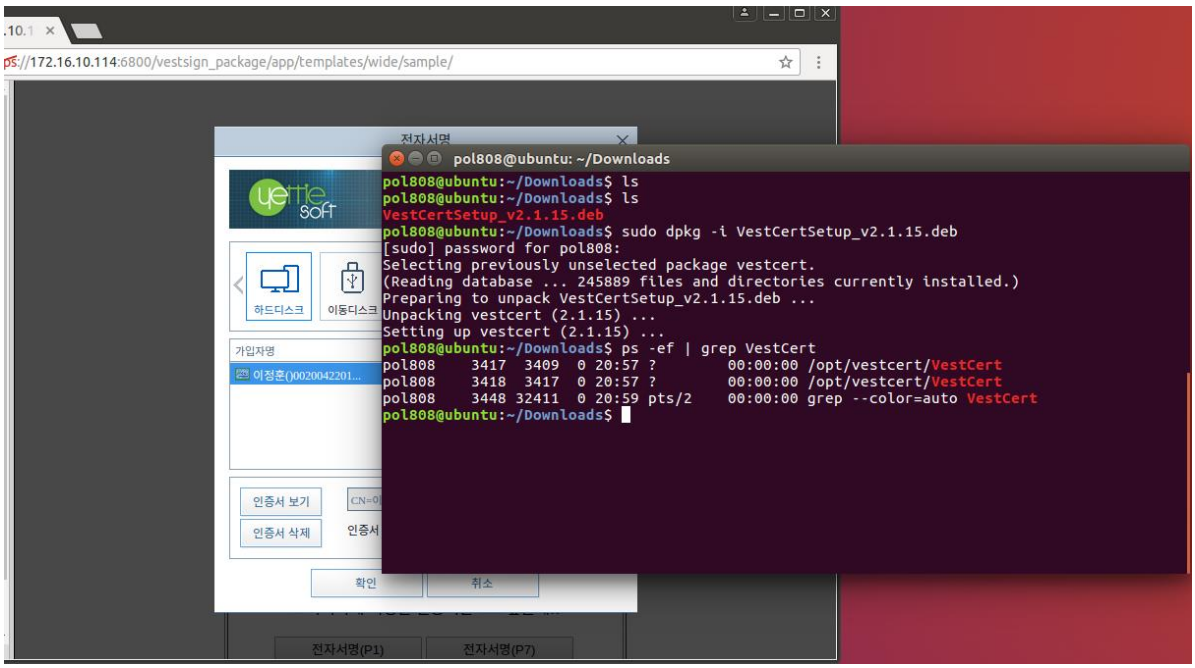
<그림 12> [linux] VestCert 구동되지 않은 경우 다운로드



<그림 13> [LINUX] 다운로드 및 설치



<그림 14> [LINUX] VestCert 설치 후 인증서 로딩 화면



<그림 15> [LINUX] VestCert 구동 확인

5.2. VestSign 설치방법

VestSign은 순수 자바스크립트와 HTML만으로 구성된 솔루션이다. 따라서 해당 프로그램을 웹 서버의 HTML/JSP Document ROOT 경로에 업로드하고 웹 페이지에서 해당 모듈을 호출하면 된다..

5.3. 전자 서명문 생성

VestSign은 전자 서명문 생성을 위해 국내외 규격을 준수한다.

5.3.1. 하드 디스크

하드디스크에 저장된 인증서로 전자 서명문을 생성하는 방법이다.

<그림 5>는 전자 서명 시 하드디스크에 저장된 공인인증서를 선택하는 이미지이다. <그림 6>은 비밀번호를 입력하여 전자 서명문을 생성하는 이미지이다. <그림 7>은 해당 전자 서명문의 결과를 기존에 사용 중인 솔루션을 통해 검증한 결과이며, VestSign이 생성한 데이터로 전자서명과 개인식별 번호가 정상적으로 검증됨을 확인 할 수 있다.

<그림 16> 하드디스크에 저장된 인증서 선택 화면

<그림 17> 인증서 비밀번호 입력 화면

전자서명문 서버검증

항목	RESULT
전자서명문	3082077206092a864886f70d010702a08207633082075f020101310f300d0609 6086480165030402010500301d06092a864886f70d010701a010040e7369676 e617475726520746573 74a082059f3082059b30820483a00302010202041954 e6d2300d06092a864886f70d01010b05003052310b3009060355040613026b7
전자서명 원문	signature test
사용자 인증서 정책	1.2.410.200005.1.1.4
사용자 인증서 DN	cn=김이구() 00880472008041718800117 4,ou=SHB,ou=personal4IB,o=yessign,c=kr
사용자 인증서 serial	424994514
에러코드	0
검증결과	certificate is valid

<그림 18> 전자서명 결과 화면

5.3.2. 이동식 디스크

이동식 디스크에 저장된 인증서로 전자 서명문을 생성하는 방법이다.

<그림 8>은 전자 서명 시 이동식 디스크에 저장된 공인인증서를 선택하는 화면이다. 이 후 전자서명 flow는 하드 디스크와 동일하다.

<그림 19> 이동식 디스크에 저장된 인증서 선택 화면

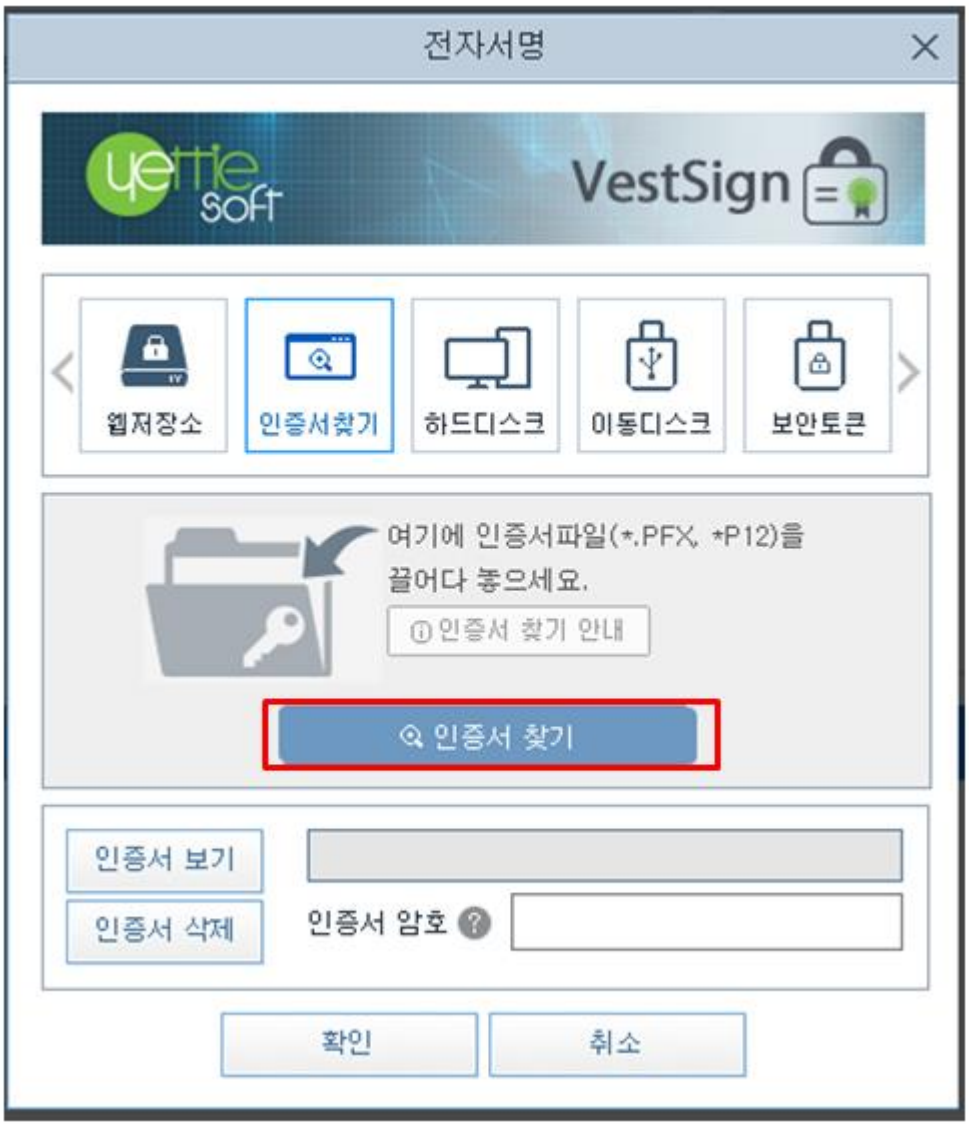
5.3.3. 보안매체

보안매체는 보안토큰등과 같은 보안매체등을 말한다. 보안토큰는 전자 서명 생성 키등 비밀 정보를 안전하게 저장 및 보관할 수 있고 기기 내부에 프로세스 및 암호 연산 장치가 있어 전자 서명 키 생성, 전자서명 생성 및 검증 등이 가능한 하드웨어 장치이다. 이 후 전자서명 flow는 하드 디스크와 동일하다.

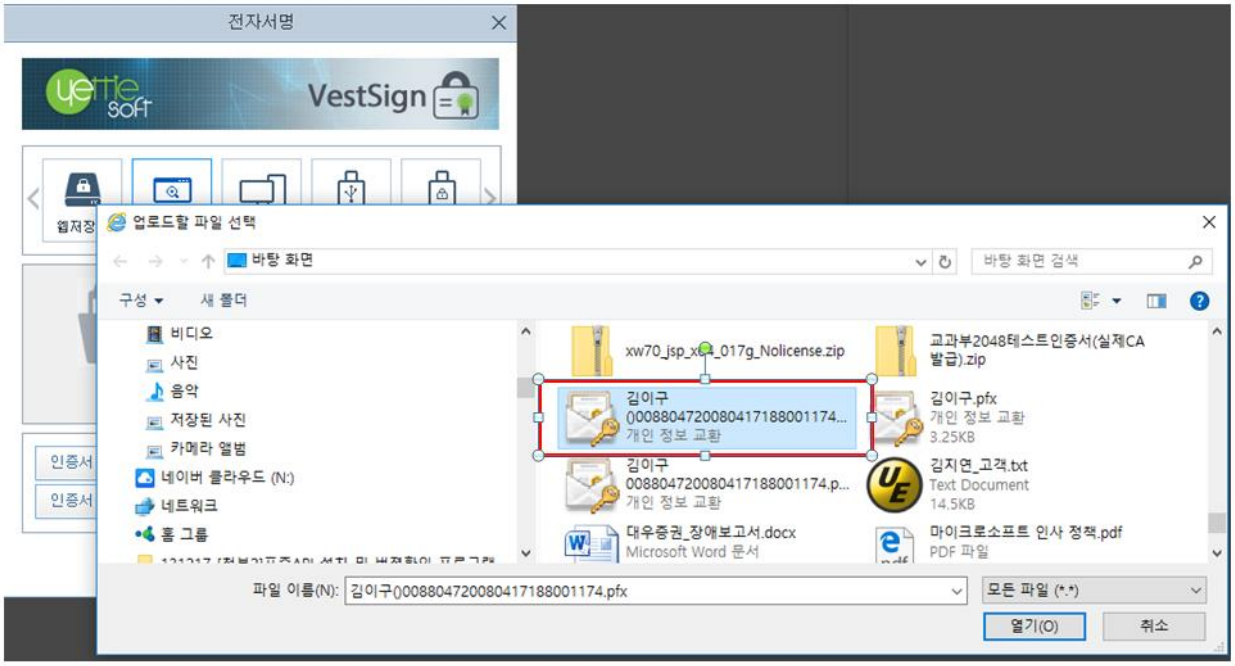
<그림 20> 보안토큰에 저장된 인증서

5.3.4. 인증서 찾기

pfx 파일이나 p12 파일을 이용한 전자서명 방식이다.



<그림 21> 인증서 찾기



<그림 22> 인증서 찾기(인증서 선택)

<그림 23> 비밀번호 입력

<그림 24> 인증서 찾기(인증서 확인)

전자서명문 서버검증

항목	RESULT
전자서명문	3082077206092a864886f70d010702a08207633082075f020101310f300d0609 6086480165030402010500301d06092a864886f70d010701a010040e7369676 e617475726520746573374a082059f3082059b30820483a00302010202041954 e6d2300d06092a864886f70d01010b05003052310b3009060355040613026b7
전자서명 원문	signature test
사용자 인증서 정책	1.2.410.200005.1.1.4
사용자 인증서 DN	cn=김이구() 0088047200804171880011174,ou=SHB,ou=personal4IB,o=yessign,c=kr
사용자 인증서 serial	424994514
에러코드	0
검증결과	certificate is valid

<그림 25> 전자서명 결과 화면

5.3.5. 휴대폰 인증

휴대폰을 이용한 전자서명 서비스와 연동하여 전자서명을 하는 기능이다.
휴대폰 저장매체를 서비스하는 서비스공급자와 협의후 제공 된다.

<그림 26> 휴대폰 인증 서비스 화면