AnySign for PC
제품 가이드

Establishing Global Trust
SOFTFORUM

Establishing Global Trust
SOFTFORUM

2. 제품 소개

2.1. 개요

AnySign for PC(이하 AnySign4PC)는 기존에 Plugin 방식으로 구동되는 XecureWeb-Multi 제품을 Non-Plugin 방식으로 구동되도록 변경하여 개발한 제품이다. 그러므로 AnySign4PC의 기능은 XWM의 모든 기능을 지원하고 있으며 XWM의 구성과 매우 유사한 형식으로 되어 있다.

그래서 제품의 설치방법도 거의 비슷하며 기존에 XWM 가 설치된 시스템에 AnySign4PC를 적용할 때 적은 작업으로도 손쉽게 적용할 수 있도록 되어 있다.

또한, AnySign4PC는 설정에 따라 XWM 의 구동도 가능하도록 스크립트부분을 개발하였기 때문에 AnySign4PC가 설치된 시스템에는 XWM 제품도 포함되어 설치된 것이다.

2.2. 주요 기능 및 특징

보안 세션 수립	- SSL Handshaking 프로토콜을 사용하여 보안 세션 수립 후 암/복호화에 사용될 세션키를 안전하여 분배 - SSL Handshaking 과정에서 전달되는 서버 인증서의 유효성 및 서버 실체 확인 - 세션 수립을 위한 상호 인증 설정에 따라 인증서 기반과 익명 방식 지원
웹 데이터 부분 암호화	- HTTPS와 같이 페이지 단위로 암호화를 설정하는 것과 달리 기밀성이 요구되는 데이터에만 부분적으로 암호화 적용
사용자 인증	- 사용자 인증에 사용될 인증서 관련 인터페이스 제공 - 사설인증서, NPKI인증서, GPKI인증서 모두 지원 - 안전한 개인키 관리(PKCS #5) - 인증서 이동성을 위한 다양한 이동 매체 지원 (USB Device, Smart Card, HSM 등) - 인증서 저장매체 강제 설정 기능 지원

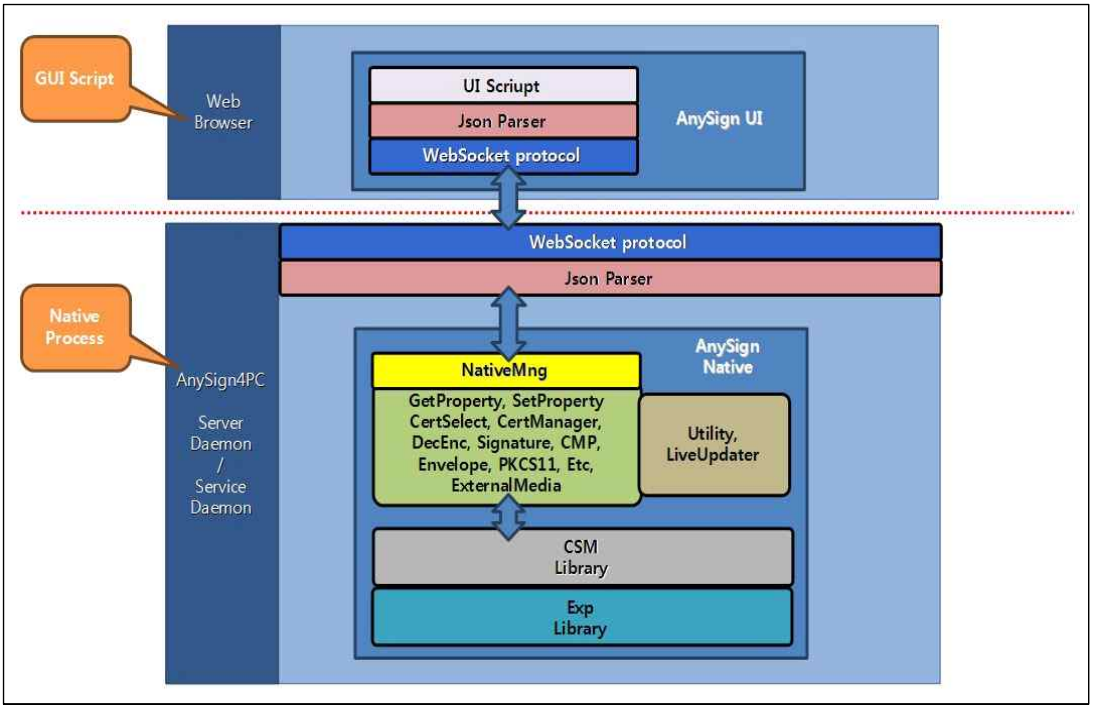
전자서명 수행 및 검증	- 식별 번호 (VID)검증 - 송/수신 데이터에 대한 부인방지 기능 제공 - 다자간 서명 지원 - 축양 전자서명 기능 지원 - 인증서에 포함된 식별번호를 이용하여 인증서 소유자의 신원확인 지원 - XecureTSA 연동으로 전자서명 시점 확인 기능 지원
표준 메시지 형식 지원	- X.509 인증서 - PKCS #7 암호 메시지 - PKCS #10 인증서 요구 메시지 (인증기관과 연동) - PKCS #12 키 이동교환 메시지
표준 알고리즘 지원	- 공개키(비대칭키) 알고리즘 : RSA 1024/2048bit, KCDSA 1024bit - 대칭키 알고리즘 : 3DES 112/168bit, RC4 128bit, RC5 128bit, SC-1 128bit, NEAT 128bit, ARIA, NES - 해쉬 알고리즘 : SHA-1 160bit, MD5 128bit, HAS 160bit, SHA-2(256/384/512bit)

2.3. AnySign for PC 구조

2.3.1. Javascript UI + AnySign4PC Daemon / Launcher Service Daemon

AnySign4PC는 크게 웹서버에 설치되는 Script 모듈부분과 사용자의 PC에 설치되어 동작되는 Native 모듈부분으로 크게 구분할 수 있다.

Script 모듈 부분은 GUI 표현과 Non-Plugin 방식으로 동작하기 위한 HTML/Javascript/CSS 기반의 Script 세트이다.
Native 모듈은 실제로 PKI기반의 암호/서명 기능을 수행하는 Native 모듈과 Non-Plugin 방식으로 웹 브라우저와 통신하기 위한 로컬 서버 데몬 프로세스가 있으며 마지막으로 제품의 설치 및 실행을 총괄하는 서비스 데몬 프로세스로 구성되어 있다.



2.3.2. 스크립트 파일/디렉토리 구조

파일/디렉토리 구조		설명
/AnySign4PC		(기본 경로)
	/module (*1)	UI 구현부.(수정 금지)
	/locale	각 다이얼로그의 언어별 파일
	/img	이미지 파일(png, gif)
	/css	스타일 시트 파일
	/ext	AnySign을 위한 Extension 모듈 - AnySign - AnySignAjax - AnySignJSONP - AnySign4PC_min.js - SecureProto.js

		- jquery-1.11.1.js - json2.js
	AnySign4PC.js (*1)	Core Script. (수정 금지)
anySign4PCInterface.js		인터페이스 스크립트

* ext 디렉토리 구성 (수정 금지) (*1)

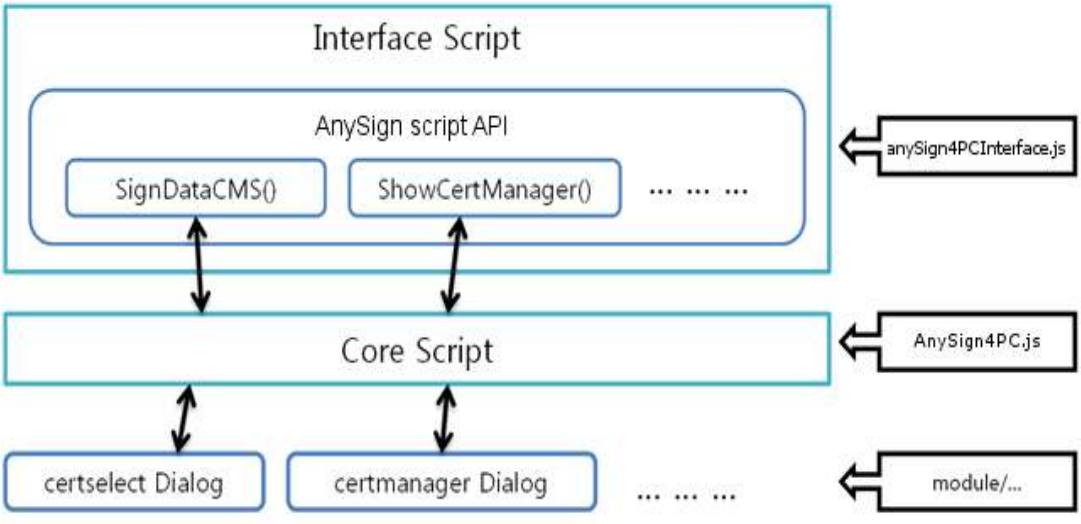
ext	설명
AnySign	HTML5 WebSocket 구현부 파일
AnySignAjax	Ajax 구현부 파일
AnySignJSONP	JSONP 구현부 파일
AnySign4PC_min.js	Secure Protocol 파일
SecureProto.js	
jquery-1.11.1.js	customized JQuery 파일
json2.js	Json 파서 파일

(*1) 수정이 허용 되지 않으며 수정하는 경우 정상 동작을 보장할 수 없음. 기타 스크립트/이미지의 경우 수정이 가능하나, 버전 업그레이드 시에 변경 내용이 적용 될 수 있도록 주의하여야 한다. 그리고 ext 디렉토리 구성표에 있는 파일들도 마찬가지이다.

2.3.3. Javascript 구성

2.3.4. Interface Script (anySign4PCInterface.js)

실제 기능 호출에 대한 인터페이스를 위한 스크립트. 기본적인 설정에 대한 내용을 포함하고 있다.



2.3.5. Core Script (AnySign4PC.js)

Native에 준하는 기능 구현부. **(임의로 수정을 금지)**

2.3.6. module (module/이하 파일들)

Dialog 단위의, 실제 UI가 담긴 모듈 파일. Javascript를 통해 HTML을 구성하여 화면에 불러온다. **(임의로 수정을 금지)**

2.3.7. AnySign4PC 위한 Extension 모듈 (ext/이하 관련 파일들)

AnySign4PC 구동과 동작의 핵심 기능을 하는 Native에 준하는 기능 구현부. **(임의로 수정을 금지)**

2.3.8. AnySign4PC 패키지 설치파일

AnySign4PC를 사용자의 PC에 설치하여 구동되도록 하기위한 제품 패키지의 설치

파일은 다음과 같다.

- 윈도우 계열 OS : AnySign_Installer.exe

- 리눅스 계열 OS : anysign4pc_linux_i386.deb(rpm), anysign4pc_linux_x86_64.deb(rpm)

- Mac 계열 OS : anysign4pc_mac_universal.pkg

2.4. 지원 Platform

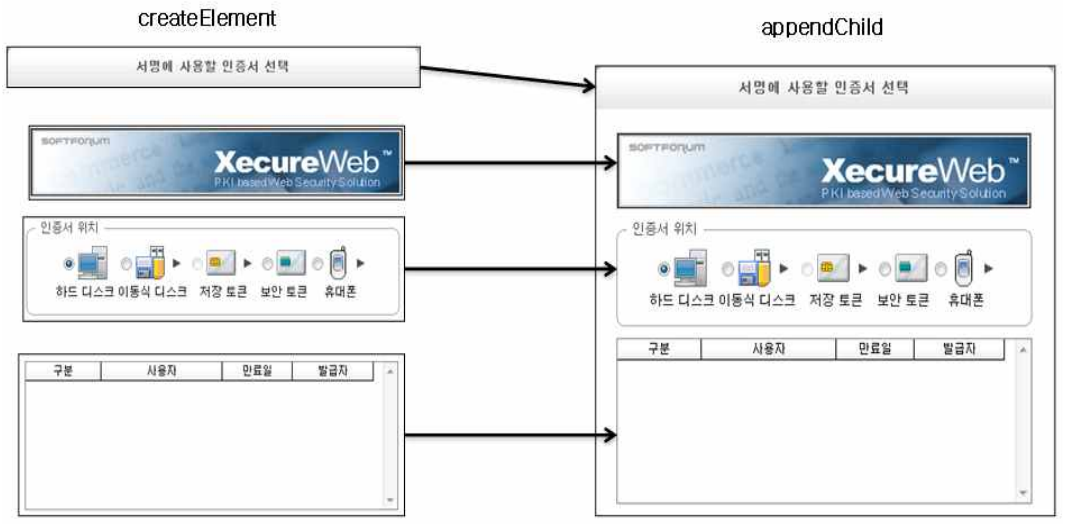
OS Version			Browser Type																			
			IE7		IE8		IE9		IE10		IE11		Edge		Chrome 24.0 ↑		Safari 5.0 ↑		Firefox 27.0 ↑		Opera 20.0 ↑	
			32	64	32	64	32	64	32	64	32	64	32	64	32	64	32	64	32	64	32	64
Windows	xp	32bit	O	-	O	-	-	-	-	-	-	-	-	-	O	-	O	-	O	-	O	-
		64bit	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	vista	32bit	O	-	O	-	O	-	-	-	-	-	-	-	O	-	O	-	O	-	O	-
		64bit	O	-	O	O	O	O	-	-	-	-	-	-	O	-	O	-	O	-	O	-
	7	32bit	-	-	O	-	O	-	O	-	O	-	-	-	O	-	O	-	O	-	O	-
		64bit	-	-	O	O	O	O	O	O	O	O	-	-	O	-	O	-	O	-	O	-
	8	32bit	-	-	-	-	-	-	O	-	-	-	-	-	O	-	O	-	O	-	O	-
		64bit	-	-	-	-	-	-	O	O	-	-	-	-	O	-	O	-	O	-	O	-
	8.1	32bit	-	-	-	-	-	-	-	-	O	-	-	-	O	-	O	-	O	-	O	-
		64bit	-	-	-	-	-	-	-	-	O	O	-	-	O	-	O	-	O	-	O	-
	10	32bit	-	-	-	-	-	-	-	-	O	-	O	-	O	-	O	-	O	-	O	-
		64bit	-	-	-	-	-	-	-	-	O	O	O	O	O	-	O	-	O	-	O	-
Mac	10.6	64bit	-	-	-	-	-	-	-	-	-	-	-	-	-	O	-	O	O	O	-	-
	10.10	64bit	-	-	-	-	-	-	-	-	-	-	-	-	-	O	-	O	O	O	-	O
Linux	ubuntu 14.04	32bit	-	-	-	-	-	-	-	-	-	-	-	-	O	-	-	-	O	-	O	-
		64bit	-	-	-	-	-	-	-	-	-	-	-	-	-	O	-	-	-	O	-	O
	Fedora 21	32bit	-	-	-	-	-	-	-	-	-	-	-	-	O	-	-	-	O	-	-	-
		64bit	-	-	-	-	-	-	-	-	-	-	-	-	-	O	-	-	-	O	-	-

3. UI Guide

3.1. Dialog

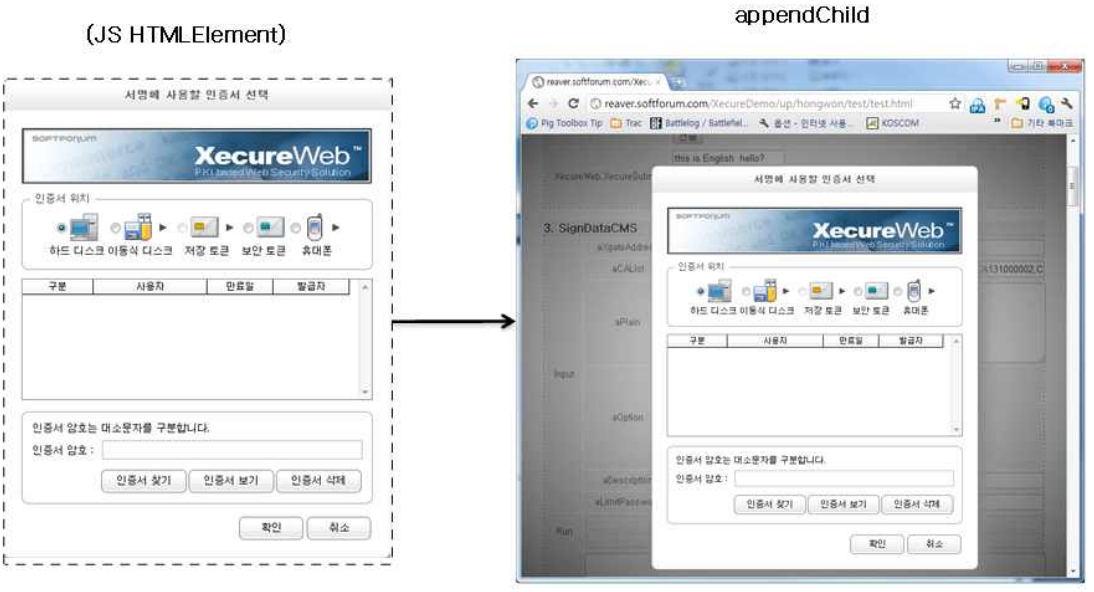
3.1.1. Create Dialog

HTML Element의 create/append를 반복하여 HTML DOM tree 구성



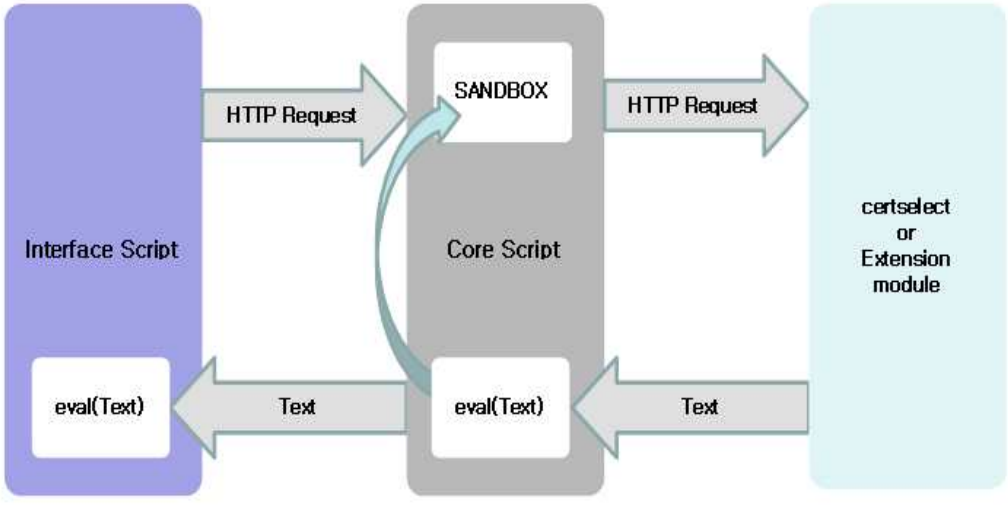
3.1.2. Append Dialog to WebPage

최종적으로 웹페이지에 append 되면서 화면에 보임



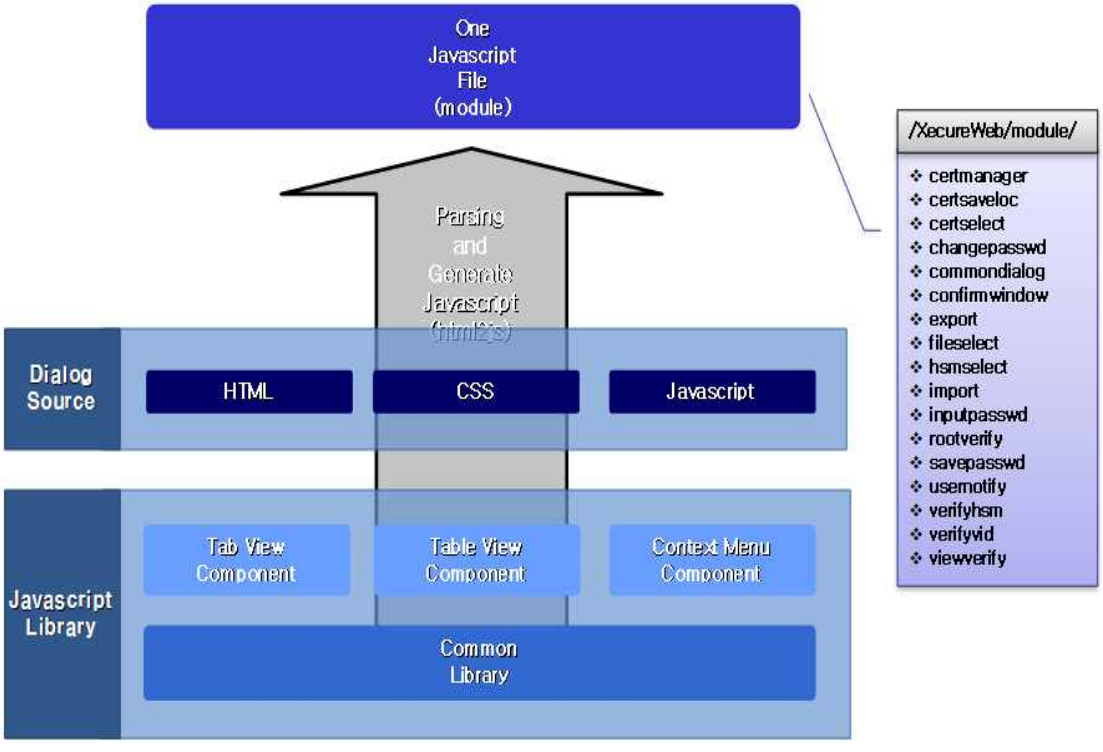
3.1.3. Dynamic Script Loading

Interface Script와 Core Script를 결합해 ajax를 통해 스크립트를 동적으로 로드



Dynamic Script Loading을 함으로써 불필요한 로딩을 줄이고 스크립트 간 충돌을 최소화 하였지만 이로 인해 디버깅이 난해해지고 오버헤드(속도 및 메모리) 문제가 발생한다.

3.1.4. Dialog Module Build



3.1.5. Dialog Module

3.1.5.1. 모듈 구성

module	설명
certmanager	인증서 관리UI
certsaveloc	인증서 저장매체 선택 UI
certselect	인증서 선택 UI
certselectwide	인증서 선택 wide UI (우리은행에만 현재 적용)
changepasswd	인증서 비밀번호 변경 UI
confirmwindow	전자서명 정보 확인 UI
export	인증서 내보내기 UI
fileselect	파일 선택 UI
hsmselect	보안토큰 서비스 제공자 선택 UI
iccard	IC 카드 PIN 번호 입력 UI
iccardlist	IC 카드 리더기 선택 UI
impport	인증서 가져오기 UI
inputpasswd	인증서 암호 입력 UI
rootverify	최상위인증기관 인증서 검증 UI
savepasswd	인증서 저장 패스워드 입력 UI
selectonrbg	인증서 가져오기/내보내기 선택 UI
usernotify	사용자 알림/발급자 정보 표시 UI (인증서 정보 다이얼로그의 사용자 알림 버튼)

verifyhsm	보안토큰 PIN 번호 입력 UI
verifyvid	소유자 검증 시 신원확인정보(주민등록번호 or 사업자등록번호) 입력 UI
guidewindow	처리 중 안내 UI
viewverify	인증서 보기 UI

4.1.4. 스크립트 Setting

- Clinet side Script

AnySign4PC를 사용하는 페이지는 기본 설정과 기능 호출을 위해서 인터페이스 스크립트 파일 (anySign4PCInterface.js)을 확인하여 설정 한다.

```
<script src="../anySign4PCInterface.js"></script>
```

그리고 Internet Explorer 의 문서처리 모드에 따른 스크립트의 오동작 문제로 인해서 브라우저의 버전정보를 잘못 가져오는 경우를 방지하기 위해 아래와 같은 DOCTYPE 설정이 필요하다.

- DOCTYPE이 선언되지 않을 경우 문서모드를 다르게 인식해 오동작 할 수 있음
(참고: http://www.w3schools.com/tags/tag_doctype.asp)

```
<!DOCTYPE html>
```

- Server side Script

AnySign4PC를 사용하는 페이지는 다음과 같은 Server side Script 의 설정을 통해

SessionID를 설정하는 작업이 필요하다.
AnySign4PC 는 기능을 수행하는 로컬 서버데몬에서 사용자가 접속한 사이트의 웹 SessionID를 기반으로 세션정보가 각각 생성되어 관리되기 때문에 이러한 설정을 통해 SessionID 값을 전달해야 한다.

웹서버에서 제공하는 sessionID (e.g. JSESSIONID)를 직접 사용하는 것이 보안적으로 문제가 될 수 있다. 이러한 경우에는 아래의 예시와 같이 해당 sessionID 값을 한번 변조하여 사용함으로서 민감한 정보가 노출되는 것을 방지할 수 있다.

```
<script language='javascript'>
<%
    VidVerifier vid = new VidVerifier(new XecureConfig());
    out.println(vid.ServerCertWriteScript());

    String HashedSessionID = "";

    // 웹세션ID 해쉬
    String id = session.getId();
    HashedSessionID = cipher.getHash("SHA256",id);

    out.println("AnySign.mAnySignSID = '" + HashedSessionID + "';");
%>
</script>
```

또한 AnySign4PC 의 SessionID를 통한 기능 세션 구분을 하지 않을 경우에는 아래와 같은 서버 스크립트 적용 없이 mAnySignSID 값을 고정된 값으로 설정하여 적용하면 된다.

```
<script language='javascript'>
<%
    VidVerifier vid = new VidVerifier(new XecureConfig());
    out.println(vid.ServerCertWriteScript());

    String HashedSessionID = "";

    // 고정 세션 ID
    HashedSessionID = "anysignTestSID19810531";

    out.println("AnySign.mAnySignSID = '" + HashedSessionID + "';");
%>
</script>
```

4.1.5. 설치 파일 배포

설치될 파일명이 변경 되거나 추가된 경우 anySign4PCInterface.js 를 수정해야 한다. anySign4PCInterface.js 파일안의 AnySignInitialize() 함수의 내부 값을 수정한다.

- **AnySign4PC 경우**

PrintObjectTag()가 호출 되면서 anySign4PCInterface.js에 설정 된 aAnySignVersion 값과, 설치되어 있는 모듈과의 버전을 비교해 기존 설치 된 버전이 더 낮은 경우에는 AnySign4PC 의 라이브업데이트 기능을 통해 aAnySignVersion 에 명시된 버전의 AnySign4PC 클라이언트 모듈을 설치하게 된다. 이때 라이브업데이트로 설치되는 패키지 파일은 AnySign4PC 공식 CDN 경로에 명시한 버전값의 URL 경로에 미리 설치파일이 업로드 되어 있어야 한다.

그리고 클라이언트 모듈이 설치되어있지 않은 경우, mPlatformList에 설정 된

aInstallPage로 페이지를 Redirect하여 AnySign4PC의 수동 설치를 시도한다.
(플랫폼별로 다른 설치 페이지 작성 가능)

```
var aVersion = "1.0.5.6";
var aAnySignVersion = "1.1.0.0";
  ...
this.mPlatformList =
[
    {
        aName               : "linux",
        aSearchWord         : "Linux",
        aInstallPath        :
        [
                "../install/anysign4pc_linux_i386.deb",
                "../install/anysign4pc_linux_i386.rpm",
                "../install/anysign4pc_linux_x86_64.deb",
                "../install/anysign4pc_linux_x86_64.rpm"
        ],
        aInstallPage        : "./installAnySign.html"
    },
    {
        aName                       : "mac universal",
        aSearchWord                 : "Mac",
        aAnySignInstallPath         : "../install/anysign4pc_mac_universal.pkg",
        aInstallPage                : "./installAnySign.html"
    },
    {
        aName                       : "windows 32bit",
        aSearchWord                 : "Win32",
        aCABInstallPath             : "../install/xwcup_install_windows_x86.cab",
        aInstallPath                : "../install/xwcup_install_windows_x86.exe",
        aAnySignInstallPath         : "../install/AnySign_Installer.exe",
        aInstallPage                : "./installAnySign.html"
    },
    {
        aName                       : "windows 64bit",
        aSearchWord                 : "Win64",
        aCABInstallPath             : "../install/xwcup_install_windows_x64.cab",
        aInstallPath                : "../install/xwcup_install_windows_x64.exe",
        aAnySignInstallPath         : "../install/AnySign_Installer.exe",
        aInstallPage                : "./installAnySign.html"
    },
  ...
```

aInstallPath 및 aInstallPage는 샘플 제공 편의상 상대 경로로 설정 되어 있으나, 개발/운영상에는 절대 경로 주소 사용을 권장한다.

ex) "/XecureDemo/xwuni/install/xecureweb-unified-plugin_1.0.3.0_i386.deb" 혹은
aBasePath + "/install/xecureweb-unified-plugin_1.0.3.0_i386.deb" 등.

설정이 완료 되면 지정된 path 에 설치 파일을 복사 하여 가져다 둔다.

- 패키지 설치파일 요청 및 설치 절차

패키지 설치파일을 사이트적용에 준비하는 단계는 아래와 같은 순서로 진행하면 된다.

① 연구소에 사이트에 적용하기 위한 패키지 설치파일 개발을 요청한다.
② 연구소에서 요청한 버전의 개발 및 패키지를 완료하고 해당 버전의 CDN 경로에 등록한다.
③ CDN 등록이 완료되면 등록된 CDN URL 경로를 전달 받는다.
④ 사이트에 적용한 Script 의 설정파일에서 제품 버전정보 변수값을 요청한 설치파일의 버전정보값으로 수정한다.
⑤ 최종적으로 검토 차원에서 Script 에 고정된 CDN 경로 변수값과 ③에서 전달받은 CDN URL 경로가 일치하는지 확인한다.

이렇게 CDN 만을 통해 설치파일을 배포하는 이유는 라이브업데이트 기능에서 일괄적으로 동일한 패키지 설치파일을 사용하여 업데이트를 하기 때문이다.

- 제품 버전이 1.1.0.0 인 경우 CDN 경로 (빨간 숫자 부분만 변경됨)

```
http://download.softforum.com/Published/AnySign/v1.1.0.0/AnySign_Installr.exe

https://download.softforum.com/Published/AnySign/v1.1.0.0/AnySign_Installr.exe
```

4.1.6. AnySign4PC 의 실행 및 사용

AnySign4PC는 다음과 같은 자바스크립트 함수를 호출하여 사용 가능하다.

```
<script>
        PrintObjectTag ();
</script>
```

PrintObjectTag 호출 이후 AnySign 인스턴스가 생성 되며, 다음과 같이 해당 자바스크립트 인스턴스를 통해 AnySign4PC의 각 기능을 호출하여 사용할 수 있다.

```
AnySign.SignDataCMS(…, …, …, …);
```

실제로 스크립트 전체적으로 AnySign4PC와 XWM 의 실행 구동을 결정하는 부분은 PrintObjectTag 함수 내부에 있다.

해당 함수 내부에서 mAnySignEnable 의 설정값이 true 로 되어있으면 실제로는 AnySign.StartAnySign 함수가 실행되고 해당 함수는 AnySign4PC를 실행시키며 모든 인터페이스 스크립트에 AnySign4PC 가 연동되게 되어 있다.

mAnySignEnable 이 false 로 설정되어 있으면 PrintObjectTag 함수 내부에서는 다시 AnySign.PrintObjectTag 함수가 실행되면서 예전의 스크립트와 동일하게 XWM가 구동되고 모든 인터페이스 스크립트에 연동된다.

4.1.8. Javascript UI와 비동기 처리 및 Callback

기존의 XWM 에서는 UI가 들어간 부분에 대해서만 비동기식으로 동작하도록 구성되어 있었으며 UI가 없는 API 의 경우는 동기식으로 동작되도록 Javascript 가 작성되었었다.

```
1) 동기식 API :  var a = b();

2) 비동기식 API :  b ( aUserCallback, aErrCallback );
```

*** AnySign4PC 에서는 기존 동기식 API로 동작되는 것을 모두 비동기식 API 로 수정하였다.**

하지만 AnySign4PC 의 경우는 모든 API 의 호출 및 결과값 반환 처리를 web socket과 Ajax를 사용해 처리하고 있으므로 결과값 반환을 비동기 이벤트로 발생하는 web socket과 Ajax의 응답메시지로 처리해야 한다.

그러므로 AnySign4PC 는 web socket 또는 Ajax에서 이벤트가 발생하면 모두 Callback 함수를 사용하여 처리하고 있으며 API 호출에 대한 응답도 Callback 함수를 사용하여 처리하고 있다.

mAnySignEnable 변수값에 따라 AnySign4PC 와 XWM 가 모두 동작 가능한 Javascript 이므로 XWM 가 동작되는 Javascript 부분에는 아직 동기식으로 처리가 되는 부분이 남아있을 수 있다. 하지만 대부분의 Javascript 는 AnySign4PC 의 구동을 위해 Callback 함수로 동작되도록 하여 비동기식으로 전환되어 있다.

- aErrCallback

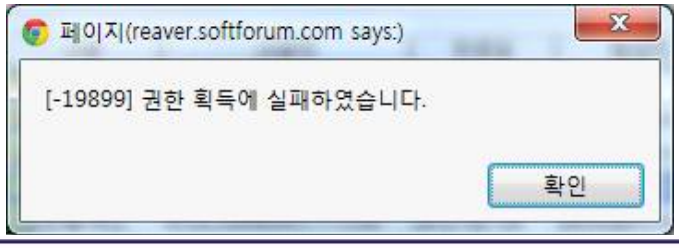
1) 호출한 API가 실패를 반환하기 위해 호출 할 Closure Function. 부연설명이 없는 한, 다음과 같은 방식으로 에러 정보를 반환한다.

```
function some_err_function (result) {
    alert (result.code); // 에러 코드
    alert (result.msg); // 에러 메시지
}
```

aErrCallback으로 입력 한 Function의 첫 번째 Parameter로 {code: [에러코드], msg: [에러 메시지]} 형태의 object를 입력하여 호출.

2) 에러 콜백을 입력하지 않는 경우, 공통으로 미리 정의 된 공통 에러 콜백 함수를

통해 "[에러코드] 에러 메시지" 형태의 Javascript Alert창을 출력한다.



	동기식	비동기식	동기식 → 비동기식
호출	result = someAPI	someAPI (…, …, callback, errCallback);	result = someAPI (…, …);

실패	if (result != 0)
	{
	alert("failed : " + lastErrMsg());
	}
		function errCallback (error)
		{
		alert("failed : " + error.msg);
		}
			if (result != 0)
			{
			errCallback (lastErrMsg());
			}

성공	else
	{
	alert("result : " + result);
	}
		var callback = function (result)
		{
		alert("result : " + result);
		}
			else
			{
			callback(result);
			}

입력 값은 String이 아닌 function statement 혹은 function literal임에 주의.

7. JavaScript API

7.1. 전자서명 API

7.1.1. 일반 전자서명

7.1.1.1. SignDataCMS

함수명	SignDataCMS (aXgateAddress, aCAList, aPlain, aOption, aDescription, aLimitPassword, aUserCallback, aErrCallback)	
파라미터	aXgateAddress	1) Xgate 주소 정보 2) 형식 : " xgate ip (or hostname) : ssl direct port : ssl proxy port “ ex 1) window.location.hostname + ":443:8080“ ex 2) "210.124.178.59:443:8080"
	aCAList	1) 전자서명시 사용할 수 있는 인증서를 발행한 발행기관의 CN List 2) 각 CN 사이의 구분자는 , 이다. 3) 만약 금결원에서 발행한 공인 인증서와 소프트포럼 인증기관에서 발행한 인증서만으로 서명을 할 수 있게 하려면 "yessign CA,소프트포럼 인증기관"과 같이 입력한다. 4) 각 인증기관별로 특정한 policy OID를 가지는 인증서만을 filtering할 때에는 인증기관의 cn값 다음에 원하는 policy OID를 “:”으로 구분하여 추가한다. ex) “yessignCA:1.2.410.200005.1.1.1,SignKorea CA”
	aPlain	1) 서명할 데이터 (크기제한 없음)
	aOption	1) 서명 원문 확인창 생성 여부와 기타 옵션 설정 0x00000000 (0) : 기본 서명 0x00000001 (1) : 서명 원문 확인 0x00000100 (256) : 리턴 값을 Base64 인코딩 (기본: Hexa 인코딩) 0x00100000 (1048576) : 인증서 위치 캐쉬 2) 위의 옵션들은 조합하여 사용 가능하다.
	~~aDescription~~	1) aOptoin 이 1, 3 인 경우 서명 원문 확인 창에 보일 설명문. NULL 혹은 "" 이면 default msg 로 세팅된다. default msg : 다음의 문서에 전자서명 합니다. 2) 현재 지원 하지 않는 기능
	aLimitPassword	1) 전자서명시 서명에 사용할 인증서를 선택한 후 해당 인증서의 잘못된 암호를 몇 번까지 허용할지를 지정한다.

		2) 이 parameter 에 해당하는 값이 없을 때 default 값은 3 이다
	aUserCallback	1) 서명문을 처리할 유저 콜백(함수)
	aErrCallback	1) 서명문을 처리할 에러 콜백(함수)
반환값	성공이면 서명문 리턴, 실패면 “” 리턴	
설명	1) 주어진 데이터를 ca_name 에 주어진 발행기관에서 발급된 인증서들 중 하나로 서명한다. 2) 서명 알고리즘 : 주어진 인증서의 서명 알고리즘	
사용 예		

```
<script>
Function aUserCallback (aResult)
{
alert( “서명문은 = “ + aResult);
}

Function aErrCallback (aResult)
{
Alert(“”+aResult.code);
Alert(“에러 메시지”+aResult.msg);
}

var aXgateAddress = window.location.hostname + “:443:8080”;
var aCAList = “yessignCA:1.2.410.200005.1.1.1,SignKorea CA”;
var aPlain = “이 내용이 전자 서명됩니다”;
var aOption = 0;
var aDescription = “”;
var aLimitPassword = 3;
}
</script>

<input type="button" class="inputbutton" style="height:auto;"
value="XecureWeb.SignDataCMS"
onClick="XecureWeb.SignDataCMS (
aXgateAddress,
aCAList,
aPlain,
aOption,
aDescription,
aLimitPassword,
aUserCallback,
aErrCallback);/>
```

7.3.1.4. XecureLink

함수명	XecureLink (link)	
파라미터	link	1) 접속하려는 페이지의 URL
반환값	성공이면 암호문 리턴, 실패하면 "" 리턴	
설명	보안세션을 연결하거나 갱신하고 보안정보로 데이터를 암호화한다.	
사용 예	전송	

7.3.1.5. XecureSubmit

함수명	XecureSubmit (form)	
파라미터	form	html <form> 을 말하며 일반적으로 'this'를 사용 하여 전송
반환값	성공이면 암호문 리턴, 실패하면 "" 리턴	
설명	보안세션을 연결하거나 갱신하고 보안정보로 데이터를 암호화한다	
사용 예	<form id="form_XecureSubmit" method="post" action="./enc_server_response.jsp?abc=def&pop=hi" method="post" onSubmit='return XecureSubmit(this, aSessionKey);'>	

7.3.1.6. XecureWeb.BlockEnc

함수명	XecureWeb.BlockEnc(aPath, aPlain, aMethod)

파라미터	aPath	1) 접속하려는 페이지의 URL
	aPlain	1) 암호화할 평문
	aMethod	1) GET/POST/RESET 방식 결정
반환값	성공이면 암호문 리턴, 실패하면 "" 리턴	
설명	보안 세션을 연결하거나 갱신하고 연결된 보안정보로 데이터를 암호화한다	
사용 예	<script> aPath = "/XecureDemo/up/trunk/qa_test/enc_server_response.jsp"; aPlain = "abc=def&pop=hi"; aMethod = "GET"; </script> <input type="button" value="XecureWeb.BlockEnc" onClick="XecureWeb.BlockEnc(aPath,aPlain,aMethod);/>	

7.3.1.7. XecureWeb.BlockDec

함수명	XecureWeb.BlockDec(aCipher)	
파라미터	aCipher	암호문 데이터
반환값	성공이면 암호문 리턴, 실패하면 "" 리턴	
설명	암호문을 복호화 한다	

사용 예	<script>
	aCipher = “pyeY22B5B/C1k49Ngs+nZ7IB5EWnacUOgH3oqncjpx82Di53hfZxddJkNZ….“; // 암호문
	</script>
	
	document.write(XecureWeb.BlockDec(aCipher));	