한국전자인증 암호화 툴킷(UniSign Toolkit)
Android 이용 가이드

Ver 1.0.0.8

개 정 이 력

버전	변경일	변경 내용	작성자	승인자
1.0.0.0	2011.01.14	최초 작성	문종환	홍성욱
1.0.0.1	2011.02.09	PKCS#1 RSA 서명, 전자봉투 추가 예제 수정	박지혜	
1.0.0.2	2011.03.02	이동 라이선스 추가	박지혜	
1.0.0.4	2011.03.03	라이브러리와 동일한 버전으로 버전 업	박지혜	
1.0.0.5	2011.04.08	이피언스 배포버젼	박지혜	
1.0.0.6	2011.05.16	Sdcard 읽어오기, PriKey set 부분 수정	노현석	
1.0.0.7	2016.08.25	인증서 이동 서버스 시 라이선스 정보 수정	최재원	
1.0.0.8	2016.09.13	인증서 가져오기 앱내부 저장 함수 추가	최재원	

목 차

1. 구성 및 확인사항

1.1. 확인 필수 사항

- 인증서 관리 앱 개발 시 라이선스
  - 개발용 라이선스
    . 테스트를 목적으로 하는 라이선스로 3 개월 테스트 라이선스 제공
  - 제품용 라이선스
    . 안드로이드 앱 서비스를 위한 무기한 라이선스

    *** 스마트폰 APP 라이선스 정보를 한국전자인증 담당자에게 전달 :
    CertToolkitMgr.*GetLicenseInfo*();

- 인증서 이동 서비스 시 라이선스
  - 서비스 라이선스
    . 스마트폰 인증서 이동서비스를 위한 라이선스로 한국전자인증과
    협의하여 서비스 라이선스 제공
  -

    *** 스마트폰 APP 의 Package Name 를 한국전자인증 담당자에게 전달 :
    예) com.crosscert.islab

- 개발 주의 / 필요사항
  - 주의사항
    . assets 폴더를 개발 프로젝트에 추가해야 함

1.2. 제공 파일

구분	경로 파일명		설명
Library		AndroidUSTK-x.x.x.x.jar	보안툴킷 라이브러리
	armeabi	libCertTransfer.so, libUSToolkit.so	이동 라이브러리 JNI 파일 보안툴킷 JNI 파일
	armeabi-v7a	libCertTransfer.so, libUSToolkit.so	이동 라이브러리 JNI 파일 보안툴킷 JNI 파일
Document	한국전자인증 안드로이드 스마트폰 툴킷 개발 예제가이드.PDF		스마트폰 보안툴킷 개발 매뉴얼
Sample	UniSign_Sample		보안툴킷 및 인증서 이동 사용 예제

1.3. 예제프로젝트 구성

1.3.1. 프로젝트 구성 설명

폴더명	디렉토리 및 파일명	설명
.settings		Eclipse 환경설정
	org.eclipse.core.resources.prefs	Eclipse 환경설정 파일
assets		안드로이드 APP 설치시 필요한 파일
	NPKI	안드로이드 스마트폰 공인인증서 경로 - /sdcard/NPKI
	icon.png	
	KISA	한국인터넷진흥원(KISA) ROOT CA 인증서
	Crosscert	한국전자인증 CA 인증서 - 1024/2048 CA 인증서 포함
	KICA	한국전보인증 CA 인증서 - 1024/2048 CA 인증서 포함
	SignKorea	코스콤 CA 인증서 - 1024/2048 CA 인증서 포함
	TradeSign	무역정보통신 CA 인증서 - 1024/2048 CA 인증서 포함
	yessign	금융결제원 CA 인증서

		- 1024/2048 CA 인증서 포함
libs		APP 에 포함되는 C 라이브러리 디렉토리
	armeabi	
	libCertTransfer.so	인증서 이동 라이브러리
	libUSToolkit.so	보안 툴킷 라이브러리
	AndroidUSTK_1.x.x.x.jar	보안툴킷 및 인증서 이동 Java 인터페이스
res		테스트 APP 프로그램 자원
	Drawable-hdpi	높은 DPI 디스플레이를 위한 아이콘
	drawable-ldpi	낮은 DPI 디스플레이를 위한 아이콘
	Drawable-mdpi	중간 DPI 디스플레이를 위한 아이콘
	layout	XML 형태의 사용자 인터페이스 레이아웃 리소스
	values	XML 형태의 리소스
src	com.crosscert. sample	
	intro.java	초기화면을 보여주고, 데이터를 로딩하는 클래스
	listCert.java	인증서 선택창을 띄워주는 클래스 - 인증서 리스트 획득 - 인증서 목록 선택
	mainMenu.java	메인화면을 띄워주고 아래기능을 처리하는 클래스 - 스마트폰 APP 초기화 - 인증서 비밀번호 변경 - 인증서 검증 - 본인확인 검증 - 인증서 전자서명 - 인증서 전자서명 검증
	movement	
	exportcert	인증서 PC 로 내보내기
	ExportCert1InputPasswd.java	인증서 비밀번호 입력 화면 - 인증서 비밀번호 확인
	ExportCert2ApproveNum.java	인증서 내보내기(승인번호 띄워지는 화면) - 인증서 내보내기
	importcert	PC 에서 인증서 가져오기
	ImportCert1ApprvNum.java	인증서 가져오기(승인번호 띄워지는 화면) - 인증서 가져오기
	ImportCert2InputPasswd.java	인증서 가져오기 후 비밀번호 변경 화면 - 인증서 비밀번호 변경

				ImportCert3Storage.java	인증서 가져온 후 저장하는 화면
		shared			
				BasicTemp.java	각 ACTIVITY 의 상단 바와 하단 메뉴를 가진 모든 화면의 기본이 되는 템플릿 클래스
				CertListAdapter.java	인증서리스트를 위한 custom Adapter
				MsgMgr.java	기능별 패키지에 걸쳐 데이터를 저장하고 있는 싱글톤(패키징 시 사용됨)
				PWTextWatcher.java	패스워드 문자열 입력시 유효한 입력만을 허용하도록 함

2. API 상세

2.1. License

2.1.1. Interface

구 분	내 용		비 고
클래스	CertToolkitMgr		
함수	GetLicenseInfo()		
파라미터			
설 명	정식 라이선스 발급을 위한 LicenseInfo 정보 획득 - LicenseInfo 정보 전달 시 정식 라이선스 발급이 진행됨		
Return	String	성공 : LicenseInfo 정보 실패 : null	

-

- **APP 생성시 Activity 에서 반드시 수행하여야 한다.**

2.1.2. 샘플

```
// 스마트폰 보안 라이브러리 초기화
//     *** 반드시 스마트폰 APP 시작시 설정
// Function
```

```
//                CertToolkitMgr.SetAppInfo();
// Parameters
//                this : 스마트폰 환경설정을 위한 Activity 클래스
//                "NwiT3HW40V7tDd4mC9OBRw==: 보안툴킷 라이센스
// Return Value
//
CertToolkitMgr.SetAppInfo(this, "S90AI2KVZzQiBxCISMJRLw==");
String LicenseInfo = CertToolkitMgr.GetLicenseInfo();
```

2.2. 툴킷 초기화

2.2.1. Interface

구 분	내 용		비 고
클래스	CertToolkitMgr		
함수	SetAppInfo (Context context, String license)		
파라미터	context license	Activity this 객체 라이선스(임시 or 정식)	
설 명	툴킷 사용하기 위해 라이선스를 입력하고 초기화함		
Return			

-

- **APP 생성시 Activity 에서 반드시 수행하여야 한다.**

2.2.2. 샘플

```
// 스마트폰 보안 라이브러리 초기화
//     *** 반드시 스마트폰 APP 시작시 설정
// Function
//                CertToolkitMgr.SetAppInfo();
// Parameters
//                this : 스마트폰 환경설정을 위한 Activity 클래스
```

```
//                  "NwiT3HW40V7tDd4mC9OBRw==: 보안툴킷 라이센스
// Return Value
//
CertToolkitMgr.SetAppInfo(this, "S90AI2KVZzQiBxCISMJRLw==");


// 상위기관 인증서 설정 (어플리케이션 실행 시 최초 1 회 이상 수행 필수)
File sdRoot = Environment.getExternalStorageDirectory();
AssetManager assetManager = getResources().getAssets();

String rootName = "NPKI";
String[] assetCAs = assetManager.list(rootName);
File sdNPKI = new File(sdRoot, rootName);
if(!sdNPKI.exists()) {
        sdNPKI.mkdir();
}

for(String assetCA : assetCAs) {
        File sdCA = new File(sdNPKI, assetCA);
        if(!sdCA.exists()) {
                sdCA.mkdir();
        }

        String[] assetCerts = assetManager.list(rootName+"/"+assetCA);
        for(String assetCert : assetCerts) {
                File sdCert = new File(sdCA, assetCert);
                if(assetCert.equalsIgnoreCase("user")) { continue; }

                if(!sdCert.exists()) {
                        InputStream is =
assetManager.open(rootName+"/"+assetCA+"/"+assetCert);
                        FileOutputStream fos = new FileOutputStream(sdCert);

                        int bytesRead = 0;
                        byte[] buffer = new byte[1024];
```

```
                        while((bytesRead = is.read(buffer, 0, 1024)) != -1) {
                                fos.write(buffer, 0, bytesRead);
                        }

                        fos.close();
                        is.close();
                }
        }
}
```

2.3. 인증서 리스트 초기화

2.3.1. Interface

구 분	내 용		비 고
클래스	CertListMgr		
함수	initCertList();		
파라미터	-		
설 명	NPKI 폴더에 저장되어 있는 인증서를 불러올 수 있도록 매니저 객체를 초기화함		
Return			
클래스	CertListMgr		
함수	initCertList(Activity act);		
파라미터	act	Activity	
설 명	앱내부의 NPKI 폴더에 저장되어 있는 인증서를 불러올 수 있도록 매니저 객체를 초기화함		
Return			

2.3.2. 샘플

```
/**
* NPKI 공용폴더에 있는 인증서 가져오기
*/
CertListMgr.getInstance().initCertList();

/**
* NPKI 앱내부에 있는 인증서 가져오기
*/
CertListMgr.getInstance().initCertList(listCert.this);
```

2.4. 인증서 리스트

2.4.1. Interface

구 분	내 용		비 고
클래스	CertListMgr		
함수	getUserCertList ();		
파라미터	-		
설 명	NPKI 폴더에 저장되어 있는 인증서 목록을 가져옴		
Return			

2.4.2. 샘플

```
// 인증서 획득
CertListMgr.getInstance().getUserCertList();
```

2.5. 인증서 정보

2.5.1. Interface

구 분	내 용	비 고
클래스	Cert	
Getter	getIssuerDN()	
설 명	발급자 DN 정보	
Getter	getSubjectDN()	
설 명	인증서 DN 정보	
프로퍼티	getCertPolicy()	
설 명	인증서 정책정보	
프로퍼티	getCertValidityNotAfter()	
설 명	유효기간 만료일	

2.5.2. 샘플

```
// 인덱스에 해당하는 인증서 정보 클래스 획득
// Function
//                  CertListMgr.getInstance().getUserCertList().get();
// Parameters
//        position : 인증서 리스트의 인덱스 번호
// Return Value
//                  curCert : 인덱스에 해당하는 인증서 정보 클래스
Cert curCert = CertListMgr.getInstance().getUserCertList().get(position);

...

// 선택된 인증서로 설정
//     - getCurCert() 함수를 통해 선택된 인증서 정보 획득
// Function
//                  CertListMgr.getInstance().setCurCert();
```

```
// Parameters
//        curCert : 인증서 정보 클래스
// Return Value
//
CertListMgr.getInstance().setCurCert(curCert);

// 선택된 인증서의 발급자 DN 추출
//
// Function
//                  CertListMgr.getInstance().getCurCert().getIssuerDN();
// Parameters
//
// Return Value
//                  IssuerDN : 선택된 인증서의 발급자 DN
String IssuerDN = CertListMgr.getInstance().getCurCert().getIssuerDN();

// 선택된 인증서의 발급자명 추출
String SubjectCN= CertUtil. parseDN(cert.getSubjectDN(), "cn");

// 선택된 인증서의 발급기관 추출
String SubjectCN= CertUtil. parseDN(cert.getSubjectDN(), "o");

// 선택된 인증서의 정책 추출
String CertPolicy = CertUtil. getCertPolicyString(cert.getCertPolicy());

// 선택된 인증서의 유효기간 만료일 추출
String IssuerDN = CertUtil. getDate(cert.getCertValidityNotAfter());
```

2.6. 인증서 저장

2.6.1. Interface

구 분	내 용	비 고

클래스	CertListMgr		
함수	writeToFile(Cert cert)		
파라미터	cert	인증서 데이터를 갖고 있는 Cert 인스턴스	
설 명	NPKI 폴더에 인증서 저장		
Return			

2.6.2. 샘플

```
Cert userCert = new Cert(Cert.CERT_TYPE_USER, signCert, signPrikey, kmCert, kmPrikey);

try {
        userCert.initCert();
        CertListMgr.getInstance().writeToFile(userCert);
} catch (Exception e) {
        e.printStackTrace();
}
```

2.7. 인증서 삭제

2.7.1. Interface

구 분	내 용		비 고
클래스	CertListMgr		
함수	deleteCert (Cert cert)		
파라미터	cert	인증서 데이터를 갖고 있는 Cert 인스턴스	
설 명	NPKI 폴더에서 인증서 삭제		
Return			

2.7.2. 샘플

```
Cert curCert = CertListMgr.getInstance().getCurCert();
CertListMgr.getInstance().deleteCert(curCert);
```

2.8. 인증서 R 값 추출

2.8.1. Interface

구 분	내 용		비 고
클래스	CertToolkitMgr		
함수	getVIDRandom(Cert cert, String passwd)		
파라미터	Cert	인증서 인스턴스	
	Password	인증서 비밀번호	
설 명	인증서 개인키 R 값 획득		
Return	Byte[]	인증서 R 값	

2.8.2. 샘플

```
byte[] vidRandom = CertToolkitMgr.getInstance().getVIDRandom(cert, passwd);
```

2.9. 전자서명(PKCS#7)

2.9.1. Interface

구 분	내 용	비 고
클래스	CertToolkitMgr	
함수	logicCMSSignedData(Cert cert, byte[] data, byte[] password)	

파라미터	cert	인증서 인스턴스	
	data	전자서명 원문	
	Password	인증서 비밀번호	
설 명	원문을 전자서명함(PKCS #7)		
Return	Byte[]	전자서명된 결과 데이터	

2.9.2. 샘플

```
// 선택된 인증서 정보 획득
Cert cert = CertListMgr.getInstance().getCurCert();

// 서명할 데이터
byte[] inputData="This is example input data".getBytes();
// 서명 결과 데이터 : Base64 문자열로 인코딩 된다.
String inputbase64 = "";

// 인증서 비밀번호
String passwd = "88888888";

try{
        // 인증서 전자서명
        // Function
        //              CertToolkitMgr.getInstance().logicCMSSignedData(...);
        // Parameters
        //      cert : 선택된 인증서
        //      inputdata : 서명할 문자열
        //      passwd.getBytes() : 인증서 패스워드
        // Return Value
        //              resultData : 전자서명 메시지
byte[] resultData=CertToolkitMgr.getInstance().logicCMSSignedData(cert, inputData,
passwd.getBytes());
```

```
        // 바이너리 데이터 Base 64 인코딩
        // Function
        //              CertToolkitMgr.getInstance().utilBase64Encode(...);
        // Parameters
        //      resultData : Base64 로 인코딩할 바이너리 데이터
        // Return Value
        //              inputbase64 : Base64 로 인코딩된 문자열, 실패이면 null
        inputbase64 =CertToolkitMgr.getInstance().utilBase64Encode(resultData);

        if(inputbase64!=null)
        Toast.makeText(getApplication(), new String(inputbase64),
Toast.LENGTH_SHORT).show();
        verifyDataBase64 = inputbase64;
}catch (USToolkitException e) {
        Toast.makeText(getApplication(), e.getMessage(), Toast.LENGTH_SHORT).show();
}
```

2.10. 전자봉투

2.10.1. Interface

구 분	내 용		비 고
클래스	CertToolkitMgr		
함수	logicCMSEnvleopedData(int algorithm, Cert cert, byte[] data)		
파라미터	algorithm	암호화 알고리즘	
	Cert	인증서 데이터	
	data	암호화할 원문 데이터	
설 명	전자봉투 포맷으로 데이터 암호화		
Return	Byte[]	전자봉투로 암호화된 결과 데이터	

2.10.2. 샘플

```
// 암호화할 데이터 원문
byte[] inputData="This is example input data".getBytes();

// 암호화 결과 데이터 변수 : Base64 문자열로 인코딩 된다.
String inputbase64 = "";

try{
// 인증서 정보 클래스 생성
        // Function
        //          new Cert(...);
        // Parameters
        //          "" : 인증서 경로
        //          Cert.CERT_TYPE_USER : 사용자 인증서 플래그
        // Return Value
        //          cert : 인증서 정보 클래스

        Cert cert = new Cert("", Cert.CERT_TYPE_USER);

        // 인증서 정보클래스에 암호화용 인증서 설정
        // Function
        //          cert.setBKMCertB64(...);
        // Parameters
        //        B64KMCert : Base64 인코딩된 사용자 인증서 문자열
        // Return Value
        //
        cert.setBKMCertB64(B64KMCert);

        // 인증서 정보 추출 초기화
        // Function
        //                  cert.initCert(...);
        // Parameters
        //
```

```
        // Return Value
        //              cert.initCert();
        // EnvelopeData 생성
        // Function
//              CertToolkitMgr.getInstance().logicCMSEnvleopedData (...);
        // Parameters
        //              encAlg : 문자열을 암호화하기위해
사용할 대칭키 암호화 알고리즘
        //              Cert : 선택된 인증서(대칭키를 암호화함)
        //              inputdata : 암호화할 문자열    //
        // Return Value
        //              resultData : envelopedData 메시지
        byte[] resultData=CertToolkitMgr.getInstance().logicCMSEnvleopedData(
androidustk.USC_ALG_SYMMENC_SEED_CBC,
cert,
inputData);

        // 바이너리 데이터 Base 64 인코딩
        // Function
        //              CertToolkitMgr.getInstance().utilBase64Encode(...);
        // Parameters
        //              resultData : Base64 로 인코딩할 바이너리 데이터
        // Return Value
        //              inputbase64 : Base64 로 인코딩된 문자열, 실패이면 null
        inputbase64 = CertToolkitMgr.getInstance().utilBase64Encode(
resultData);

if(inputbase64!=null)
        Toast.makeText(getApplication(), new String(inputbase64),
Toast.LENGTH_SHORT).show();
        envelopedDataBase64 = inputbase64;
}catch (NullPointerException e) {
        Toast.makeText(getApplication(), e.getMessage(), Toast.LENGTH_SHORT).show();
}catch (USToolkitException e) {
        Toast.makeText(getApplication(), e.getMessage(), Toast.LENGTH_SHORT).show();
```

```
}
```

2.11. 전자서명(PKCS#1)

2.11.1. Interface

구 분	내 용		비 고
클래스	CertToolkitMgr		
함수	logicSignature(cert, passwd, data);		
파라미터	Cert	인증서 인스턴스	
	Password	인증서 비밀번호	
	data	전자서명 원문	
설 명	Signature 값 획득		
Return	Byte[]	Signature 값	

2.11.2. 샘플

```
byte[] signature = CertToolkitMgr.getInstance().logicSignature(cert, passwd, data);
```

2.12. RSA 암호화

2.12.1. Interface

구 분	내 용		비 고
클래스	CertToolkitMgr		
함수	cryptRSA(justoolkit.USC_ALG_ASYMM_RSA1024, pubkey, data);		
파라미터	algorithm	알고리즘	
	Pubkey	공개키	

	data	평문	
설 명	암호문 값 획득		
Return	Byte[]	암호문	

2.12.2. 샘플

```
byte[]encKey = CertToolkitMgr.getInstance().cryptRSA(justoolkit.USC_ALG_ASYMM_RSA1024, pubkey, rnd);
```

2.13. SEED 암호화

2.13.1. Interface

구 분	내 용		비 고
클래스	CertToolkitMgr		
함수	cryptSeed(data, key, iv);		
파라미터	Data	평문	
	Key	암호화 키	
	iv	암호화 IV	
설 명	암호문 값 획득		
Return	Byte[]	암호문	

2.13.2. 샘플

```
byte[]data = CertToolkitMgr.getInstance().cryptSeed(dataOrigin, sKeyBytes, defIV);
```

2.14. 인증서 공개키 추출

2.14.1. Interface

구 분	내 용		비 고
클래스	CertToolkitMgr		
함수	certGetPublicKey(cert)		
파라미터	Cert	인증서 바이너리	
설 명	인증서 공개키 값 획득		
Return	Byte[]	공개키	

2.14.2. 샘플

```
byte[] signature = CertToolkitMgr.getInstance().logicSignature(cert, passwd, data);
```

2.15. 인증서 가져오기

2.15.1. Interface : transV2Init

구 분	내 용		비 고
클래스	CertTransferMgr		
함수	transV2Init(String crsKey, boolean state)		
파라미터	crsKey	이동 라이선스	
	state	이동 라이브러리 사용 주체 (라이브러리 타입 : false)	
설 명	인증서 이동 라이브러리 초기화		
Return	boolean	성공 : true 실패 : false	

```
try {
	CertTransferMgr.SetAppInfo(this, "EDA5DA97");
	transfer = CertTransferMgr.getInstance();
} catch (InitializeException e) {
	e.printStackTrace();
}

try {
	transfer.transV2Init("EDA5DA97", false);
} catch (InitializeException e1) {
	e1.printStackTrace();
}
```

2.15.2. Interface : transV2SendReceiverInfo

구 분	내 용		비 고
클래스	CertTransferMgr		
함수	transV2SendReceiverInfo (String deviceUInfo, String deviceName, String authnum, int authnumType)		
파라미터	deviceUInfo	기기 고유정보	
	deviceName	기기 이름	
	authnum	인증서 이동 승인번호	
	authnumType	인증서 이동 승인번호 유형	
설 명	승인번호 입력하여 인증서 가져오기 준비 완료를 알림		
Return	String	인증서 이동 승인번호	

```
boolean ret =
transfer.transV2SendReceiverInfo(Secure.getString(getContentResolver(),
Secure.ANDROID_ID), deviceName, authnum, 0x10);
```

2.15.3. Interface : TRANS_ImportCert:

구 분	내 용		비 고
클래스	CertTransferMgr		
함수	transV2ImportCert()		
파라미터			
설 명	인증서 가져오기 수행		
Return	boolean	성공 :0 실패 : error	

```
if(transfer.transV2ImportCert()) {
        byte[] signCert = transfer.transGetSignCert();
        byte[] signPrikey = transfer.transGetSignPriKey();
        byte[] kmCert = transfer.transGetKmCert();
        byte[] kmPrikey = transfer.transGetKmPriKey();
}
```

2.15.4. Interface : TRANS_Finalize

구 분	내 용		비 고
클래스	CertTransferMgr		
함수	transV2Finalize()		
파라미터			
설 명	인증서 가져오기 라이브러리 해제		
Return			

```
transfer.transV2Finalize();
```

2.16. 인증서 내보내기

2.16.1. Interface : transV2Init

구 분	내 용		비 고
클래스	CertTransferMgr		
함수	transV2Init(String crsKey, boolean state)		
파라미터	crsKey	이동 라이선스	
	state	이동 라이브러리 사용 주체 (라이브러리 타입 : false)	
설 명	인증서 이동 라이브러리 초기화		
Return	boolean	성공 : true 실패 : false	

```
try {
	CertTransferMgr.SetAppInfo(this, "EDA5DA97");
	transfer = CertTransferMgr.getInstance();
} catch (InitializeException e) {
	e.printStackTrace();
}

try {
	transfer.transV2Init("EDA5DA97", false);
} catch (InitializeException e1) {
	e1.printStackTrace();
}
```

2.16.2. Interface : transV2GenerateCertNum

구 분	내 용	비 고
클래스	CertTransferMgr	
함수	transV2GenerateCertNum(String szSerialNum, byte password[], byte certBuf[], byte priKeyBuf[])	

파라미터	szSerialNum	기기 고유정보	
	password	인증서 사용자 비밀번호	
	certBuf	인증서 바이너리 데이터	
	priKeyBuf	개인키 바이너리 데이터	
설 명	인증서 이동 승인번호 생성		
Return	String	성공 : 승인번호 13 자리 실패 : null	

```
String authnum =
transfer.transV2GenerateCertNum(Secure.getString(getContentResolver(),
Secure.ANDROID_ID), password.getBytes(), signCert, signPrikey);
```

2.16.3. Interface : transV2IsReceiverConnected:

구 분	내 용		비 고
클래스	CertTransferMgr		
함수	transV2IsReceiverConnected()		
파라미터			
설 명	인증서 가져오기 측 접속 확인		
Return	boolean	접속 : true 미접속 또는 실패 : false	

```
if(transfer.transV2IsReceiverConnected()){
	transfer.transV2ExportCert(justoolkit.USC_ALG_SYMM_SEED, signCert,
signPrikey, kmCert, kmPrikey);
}else{
	Log.i("CertMove", "연결 실패");
	return false;
}
```

2.16.4. Interface : transV2ExportCert

구 분	내 용		비 고
클래스	CertTransferMgr		
함수	transV2ExportCert(int nEncAlg, byte certBuf[], byte priKeyBuf[], byte kmCertBuf[], byte kmPriKeyBuf[])		
파라미터	nEncAlg	PFX 생성 알고리즘	
	certBuf	서명용 인증서	
	priKeyBuf	서명용 개인키	
	kmCertBuf	암호용 인증서	
	kmPriKeyBuf	암호용 개인키	
설 명	인증서 내보내기		
Return	boolean	성공 : true 실패 : false	

```
transfer.transV2ExportCert(justoolkit.USC_ALG_SYMM_SEED, signCert,
signPrikey, kmCert, kmPrikey);
```

2.16.5. Interface : TRANS_Finalize

구 분	내 용		비 고
클래스	CertTransferMgr		
함수	transV2Finalize()		
파라미터			
설 명	인증서 가져오기 라이브러리 해제		
Return			

```
transfer.transV2Finalize();
```