한국전자인증 암호화 툴킷(UniSign Toolkit)
iPhone 이용 가이드

Ver 1.0.0.7

개 정 이 력

버전	변경일	변경 내용	작성자	승인자
1.0.0.0	2011.03.10	최초 작성	홍은미/ 김남희	홍성욱
1.0.0.1	2012.02.24	KeyChain 저장방식 적용	최근영	
1.0.0.2	2012.02.28	KeyChain 저장방식 이용시 주의사항 추가	최근영	
10.0.3	2016.08.25	인증서 이동 서비스 시 라이선스 정보 수정	최재원	
1.0.0.4	2016.09.09	Xcode 7.3.1 기준으로 샘플앱 신규 생성에 따른 매뉴얼 전체 수정	최재원	
1.0.0.5	2016.12.20	인증서 신원확인 기능 추가	최재원	
1.0.0.6	2017.02.14	인증서 이동 라이선스 삭제	최재원	
1.0.0.7	2017.06.05	비밀번호 변경 기능 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
    . 아이폰 앱 서비스를 위한 무기한 라이선스

    *** 스마트폰 APP 라이선스 정보를 한국전자인증 담당자에게 전달 : API_GetLibLicenseInfo

- 개발 주의 / 필요사항
  - 주의사항
    . 디바이스용 라이브러리를 배포하므로 빌드할 때 타겟을 다바이스로 설정해야 함

  - 주의사항
    . Certificate 폴더를 개발 프로젝트에 추가해야 함

  - 추가 프레임워크
    . Security.framework 추가

1.2. 제공 파일

구분	경로 파일명	설명
Library	USToolkitSample₩Lib₩libUSToolkitEx.a	장치용 보안툴킷 라이브러리
Header	USToolkitSample₩Lib₩OBJCCertTransfer.h USToolkitSample₩Lib₩OBJCUSToolkit.h USToolkitSample₩Lib₩USCertificate.h	UniSign Toolkit 헤더 파일

	USToolkitSample₩Lib₩USError.h USToolkitSample₩Lib₩USListMgr.h USToolkitSample₩Lib₩UST_Type.h USToolkitSample₩Lib₩USToolkitMgr.h USToolkitSample₩Lib₩USTransferMgr.h USToolkitSample₩Lib₩USUtil.h	
Document	한국전자인증 아이폰 스마트폰 툴킷 개발(KTSHOW 제외) 예제가이드.PDF	스마트폰 보안툴킷 개발 매뉴얼
Sample	USToolkitSample	아이폰 보안툴킷 및 인증서 이동 사용 예제

1.3. 예제프로젝트 구성

1.3.1. 프로젝트 구성 설명

폴더명	디렉토리 및 파일명	설명
Classes	CertList.h	인증서 리스트 헤더파일
	CertList.m	인증서 리스트 구현파일
	RowCertList.h	인증서 리스트 각각의 셀 헤더파일
	RowCertList.m	인증서 리스트 각각의 셀 구현파일
	SampleMain.h	샘플메뉴 헤더파일
	SampleMain.m	샘플메뉴 구현파일
	MoveCert.h	인증서 가져오기 헤더파일
	MoveCert.m	인증서 가져오기 구현파일
	ExportCert.h	인증서 내보내기 헤더파일
	ExportCert.m	인증서 내보내기 구현파일
	USToolkitEx_Sample.cpp	UStoolkit 기능 목록 처리 구현파일
	USToolkitSampleAppDelegate.h	App delegate 헤더 파일
	USToolkitSampleAppDelegate.mm	App delegate 구현 파일
	USToolkitSampleViewController.h	기본 view controller 헤더 파일
	USToolkitSampleViewController.mm	기본 view controller 구현 파일
	AppDelegate.h	앱 델리게이트 헤더파일
	AppDelegate .m	앱 델리게이트 구현파일

	USLoadingView.h	로딩 뷰 헤더파일
	USLoadingView.m	로딩 뷰 구현 파일
Library	Device	장치용 바이너리
	libUSToolkitEx.a	UniSign Toolkit 라이브러리
	libUSToolkit.a	툴킷을 이용한 추가 기능 라이브러리
	libCCSSL.a	인증서 이동 네트워크 라이브러리
	libCertTransfer.a	인증서 이동 라이브러리
	Simulator	에뮬용 바이너리
	libUSToolkitEx.a	UniSign Toolkit 라이브러리
	libUSToolkit.a	툴킷을 이용한 추가 기능 라이브러리
	libCCSSL.a	인증서 이동 네트워크 라이브러리
	libCertTransfer.a	인증서 이동 라이브러리
Resource	Image	
	UniSign_Background_Full.png	App background 이미지
Certificate	최상위, 상위기관 인증서	기관별로 인증서가 각 디렉토리에 저장되어 있음
Main.m		App main
Main.storyboard		UI 화면
USToolkitSample-Info.plist		속성정보
USToolkitSample.xcodeproj		프로젝트 파일

2. API 상세

2.1. License

2.1.1. Interface

구 분	내 용		비 고
클래스	USToolkitMgr		
함수	+ (NSString *) API_GetLibLicenseInfo:(USError **)error		
파라미터	error	오류 발생시 상세 오류 내용을 리턴 성공 : nil 실패 : USError 객체 (오류코드, 메시지는 Error 챕터 참조)	
설 명	정식 라이선스 발급을 위한 LicenseInfo 정보 획득		

	- LicenseInfo 정보 전달 시 정식 라이선스 발급이 진행됨		
Return	NSString	성공 : LicenseInfo 정보 실패 : nil	

2.1.2. 샘플

```
USError *error = nil;
NSString      *licenseInfo = [USToolkitMgr API_GetLibLicenseInfo:&error];
NSLog(@"licenseInfo : %@", licenseInfo);

[self showAlert:@"licenseinfo" message:[NSString stringWithFormat:@"실제 사용 라이선스를 발급 받기 위해서는 한국전자인증 담당자에게 다음의 정보를 보내주세요. licenseInfo : %@", licenseInfo]];
```

2.2. 툴킷 초기화

2.2.1. Interface

구 분	내 용		비 고
클래스	USToolkitMgr		
함수	+ (void) setLicense:(NSString *)license		
파라미터	License	툴킷 라이선스	
설 명	툴킷의 라이선스를 입력. 툴킷을 사용하기 위해 최초 한번 입력해야 함		
Return	-		
함수	+ (id) getInstance:(USError **)error		
파라미터	Error	오류 발생시 상세 오류 내용을 리턴 성공 : nil 실패 : USError 객체 (오류코드, 메시지는 Error 챕터 참조)	
설 명	툴킷 객체 생성		
Return	id	성공 : 툴킷 인스턴스	

		실패 : nil	

2.2.2. 샘플

```
USError *error;
    [USToolkitMgr setLicense:self.app.TOOLKIT_LICENSE];
    [USToolkitMgr getInstance:&error];
    if(error != nil)
    {
        NSLog(@"USToolkitMgr getInstance ERROR : %ld, %@", (long)[error code], [error description]);
        [self showAlert:@"actionInitToolkit" message:[NSString stringWithFormat:@"초기화 실패 : %ld, %@", (long)[error code], [error description]]];
        return;
    }
    self.toolkitMgr = [USToolkitMgr getInstance:&error];
    [self showAlert:@"actionInitToolkit" message:@"초기화 완료"];
```

2.3. 인증서 리스트

2.3.1. Interface

구 분	내 용		비 고
클래스	USListMgr		
함수	+ (NSArray *) UserCertificates		
파라미터	-		
설 명	키체인에 저장된 인증서들을 가져옴		
Return	NSArray	키체인에 저장된 인증서들의 인스턴스를 배열로 받음	

2.3.2. 샘플

```
NSMutableString *message = [NSMutableString new];
NSArray *certificates = [USListMgr UserCertificates];

if([certificates count] == 0)
{
    [self showAlert:@"actionCertList" message:@"인증서 목록이 없습니다."];
    return;
}

for (int i = 0 ; i < [certificates count]; i++)
{
    USCertificate *cert = [certificates objectAtIndex:i];
    [message appendString:[NSString stringWithFormat:@"%@%@",[cert
commonName], @"₩n"]];

    self.app.mCert = cert;
}

[self showAlert:@"actionCertList" message:message];
```

2.4. 인증서 정보

2.4.1. Interface

구 분	내 용	비 고
클래스	USCertificate	
프로퍼티	NSString *serial	
설 명	인증서 시리얼 번호	
프로퍼티	NSString *signatureAlgorithm	
설 명	인증서 서명 알고리즘	
프로퍼티	NSString *issuerDN	
설 명	인증서 발급자 식별 명칭	
프로퍼티	NSString *certValidityFrom	
설 명	유효기간 시작일	
프로퍼티	NSString *certValidityTo	
설 명	유효기간 만료일	
프로퍼티	NSString *subjectDN	
설 명	소유자 식별 명칭	
프로퍼티	NSString *keyUsage	
설 명	키 용도	
프로퍼티	NSString *commonName	
설 명	인증서 cn	

2.4.2. 샘플

```
    NSMutableString *info = [NSMutableString new];
    [info appendString:[NSString stringWithFormat:@"%@ : %@₩n",@"subjectDN",
[self.app.mCert subjectDN]]];
```

```
    [info appendString:[NSString stringWithFormat:@"%@ : %@₩n",@"validityPeriod",
[self.app.mCert validityPeriod]]];
    [info appendString:[NSString stringWithFormat:@"%@ : %@₩n",@"keyUsage",
[self.app.mCert keyUsage]]];

    UIAlertController * alert=    [UIAlertController
                                   alertControllerWithTitle:@"알림"
                                   message:info
                                   preferredStyle:UIAlertControllerStyleAlert];

    UIAlertAction* ok = [UIAlertAction
                         actionWithTitle:@"ok"
                         style:UIAlertActionStyleDefault
                         handler:^(UIAlertAction * action)
                         {

                         }];

    [alert addAction:ok];

    [self presentViewController:alert animated:YES completion:nil];

    NSLog(@"Cert DN : %@₩n", [cert subjectDN]);
    NSLog(@"Expiry Date : %@₩n", [cert validityPeriod]);
    NSLog(@"Cert Key Usage : %@₩n", [cert keyUsage]);
```

2.5. 인증서 저장

2.5.1. Interface

구 분	내 용	비 고
클래스	USListMgr	

함수	+ (BOOL) add:(USCertificate *)cert subjectDN:(NSString *)dn		
파라미터	cert	인증서 데이터를 갖고 있는 USCertificate 인스턴스	
	dn	인증서 식별 정보	
설 명	키체인에 인증서 저장		
Return	BOOL	키체인 인증서 저장 결과	

2.5.2. 샘플

```
USCertificate     *cert = [[USCertificate alloc] initWithSignCert:signCert
signPrikey:signPrikey kmCert:kmCert kmPrikey:kmPrikey];
[cert setToolkit:self.toolkitMgr];

[USListMgr add:cert subjectDN:[cert subjectDN]];
NSArray *arr = [USListMgr AllCertificates];
NSLog(@"AllCertificates count : %ld",[arr count]);
```

2.6. 인증서 삭제

2.6.1. Interface

구 분	내 용		비 고
클래스	USListMgr		
함수	+ (BOOL) remove:(USCertificate *)cert subjectDN:(NSString *)dn		
파라미터	cert	인증서 데이터를 갖고 있는 USCertificate 인스턴스	
	dn	인증서 식별 정보	
설 명	키체인에서 인증서 삭제		
Return	BOOL	키체인 특정 인증서 삭제 결과	
함수	+ (BOOL) clear		

파라미터	-		
설 명	키체인에서 모든 인증서 삭제		
Return	BOOL	키체인 모든 인증서 삭제 결과	

2.6.2. 샘플

```
BOOL ret = [USListMgr remove:self.app.mCert subjectDN:[self.app.mCert subjectDN]];

NSString *message = [NSString new];
if(ret)
{
    message = @"인증서 삭제 성공";
}
else
{
    message = @"인증서 삭제 실패";
}
UIAlertController * alert=   [UIAlertController
                              alertControllerWithTitle:@"알림"
                              message:message
                              preferredStyle:UIAlertControllerStyleAlert];

UIAlertAction* ok = [UIAlertAction
                     actionWithTitle:@"ok"
                     style:UIAlertActionStyleDefault
                     handler:^(UIAlertAction * action)
                     {
                     }];
[alert addAction:ok];
[self presentViewController:alert animated:YES completion:nil];
```

2.7. 인증서 R 값 추출

2.7.1. Interface

구 분	내 용		비 고
클래스	USToolkitMgr		
함수	- (NSData *) CERT_GetVIDRandomWithPrikey:(NSString *)password encPrikey:(NSData *)encPrikey error:(USError **)error		
파라미터	password	인증서 비밀번호	
	encPrikey	인증서 개인키 데이터	
	Error	오류 발생시 상세 오류 내용을 리턴 성공 : nil 실패 : USError 객체 (오류코드, 메시지는 Error 챕터 참조)	
설 명	인증서 R 값 추출		
Return	NSData	인증서 R 값	

2.7.2. 샘플

```
    USError *error = nil;
    USToolkitMgr      *toolkit = [USToolkitMgr getInstance:&error];

    NSData      *vidR = [toolkit CERT_GetVIDRandomWithPrikey:@"88888888"
encPrikey:[self.app.mCert dataForType:kUSSignPrikey] error:&error];
    NSString      *vidRString = [NSString binToHexString:vidR error:&error];
    UIAlertController * alert=     [UIAlertController
                                    alertControllerWithTitle:@"알림"
                                    message:vidRString
                                    preferredStyle:UIAlertControllerStyleAlert];

    UIAlertAction* ok = [UIAlertAction
                         actionWithTitle:@"ok"
                         style:UIAlertActionStyleDefault
```

```
                                handler:^(UIAlertAction * action)
                                {
                                }];
    [alert addAction:ok];
    [self presentViewController:alert animated:YES completion:nil];
```

2.8. 인증서 비밀번호 변경

2.8.1. Interface

구 분	내 용		비 고
클래스	USToolkitMgr		
함수	(USCertificate *) logicChangeCert:(USCertificate *)cert currentPassword:(NSString *)password newPassword:(NSString *)newPassword error:(USError **)error ;		
파라미터	password	인증서 비밀번호	
	newPassword	새로운 인증서 비밀번호	
	Error	오류 발생시 상세 오류 내용을 리턴 성공 : nil 실패 : USError 객체 (오류코드, 메시지는 Error 챕터 참조)	
설 명	인증서 비밀번호 변경		
Return	NSData	비밀번호 변경된 인증서	

2.8.2. 샘플

```
-(IBAction)actionChangePassword:(id)sender{

    NSString* oldPwd = @"88888888";
    NSString* newPwd = @"00000000";
```

```
    USError *error = nil;
    USToolkitMgr      *toolkit = [USToolkitMgr getInstance:&error];
    USCertificate     *cert = [toolkit logicChangeCert:self.app.mCert currentPassword:oldPwd
newPassword:newPwd error:&error];

    NSString *msg;
    if(nil == error)
    {
        [USListMgr add:cert subjectDN:cert.subjectDN];
        msg = @"비밀번호가 변경되었습니다.";
    }
    else
    {
        msg = [NSString stringWithFormat:@"비밀번호 변경에 실패했습니다.₩n(%ld)",
(long)[error code]];
    }
    [self showAlert:@"비밀번호 변경" message:msg];
}
```

2.9. 전자서명(PKCS#7)

2.9.1. Interface

구 분	내 용		비 고
클래스	USToolkitMgr		
함수	- (NSData *) logicSignedData:(USCertificate *)cert data:(NSData *)data password:(NSString *)password error:(USError **)error;		
파라미터	cert	서명용 인증서	

	Data	전자서명할 데이터 원문	
	Password	인증서 비밀번호	
	Error	오류 발생시 상세 오류 내용을 리턴 성공 : nil 실패 : USError 객체 (오류코드, 메시지는 Error 챕터 참조)	
설 명	PKCS#7 전자서명		
Return	NSData	전자서명된 결과 데이터	

2.9.2. 샘플

```
NSData   *data = nil;
NSString *test = @"test1234";
data = [test dataUsingEncoding:NSUTF8StringEncoding];
NSString *password = @"88888888";
NSData *ret = [toolkit logicSignedData:self.app.mCert data:data password:password
error:&error];
mSignedData = [NSString base64Encode:ret error:&error];
NSLog(@"signedData : %@", mSignedData);
if(nil == toolkit) {
    return;
}
if(nil == certListViewController.selectedCert) {
    return;
}
NSData   *data = nil;
NSString *test = @"test1234";
data = [test dataUsingEncoding:NSUTF8StringEncoding];
NSString *password = @"88888888";
NSData      *ret    =    [toolkit    logicCMSSignedData:certListViewController.selectedCert
data:data password:password error:&error];
NSString *retString = [NSString base64Encode:ret error:&error];
signedData = retString;
```

```
    NSLog(@"signedData : %@", retString);
```

2.10. 전자서명(PKCS#7) 검증

2.10.1. Interface

구 분	내 용		비 고
클래스	USToolkitMgr		
함수	- (NSData *) CMS_VerifySignedData:(NSData *)signedData error:(USError **)error;		
파라미터	signedData	전자서명 검증할 서명문	
	Error	오류 발생시 상세 오류 내용을 리턴 성공 : nil 실패 : USError 객체 (오류코드, 메시지는 Error 챕터 참조)	
설 명	PKCS#7 전자서명 검증		
Return	NSData	원문 데이터	

2.10.2. 샘플

```
USError *error = nil;
    USToolkitMgr *toolkit = [USToolkitMgr getInstance:&error];

    if(nil == toolkit) {
        NSLog(@"toolkit is nil");
        return;
    }

    if(nil == mSignedData) {
        NSLog(@"cert is nil");
        return;
    }
```

```
NSData   *data = [NSData base64Decode:mSignedData error:&error];
NSData   *ret = [toolkit CMS_VerifySignedData:data error:&error];
NSString      *retString = [[NSString alloc] initWithData:ret
encoding:NSUTF8StringEncoding];

NSLog(@"verify p7 : %@", retString);
```

2.11. 전자서명(PKCS#1)

2.11.1. Interface

구 분	내 용		비 고
클래스	USToolkitMgr		
함수	-(NSData *) logicSignature: password: data: algorithm: error:		
파라미터	Cert	인증서 인스턴스	
	Password	인증서 사용자 비밀번호	
	Data	전자서명할 원문 데이터	
	Algorithm	서명 알고리즘	
	Error	오류 발생시 상세 오류 내용을 리턴 성공 : nil 실패 : USError 객체 (오류코드, 메시지는 Error 챕터 참조)	
설 명	PKCS#1 전자서명		
Return	NSData	전자셔명 결과 데이터	

2.11.2. 샘플

```
USError *error = nil;
USToolkitMgr *toolkit = [USToolkitMgr getInstance:&error];
```

```
    if(toolkit == nil) {
        NSLog(@"toolkit is nil");
        return;
    }

    if(self.app.mCert == nil) {
        NSLog(@"cert is nil");
        return;
    }

    NSData   *data = nil;
    NSString      *test = @"test1234";
    data = [test dataUsingEncoding:NSUTF8StringEncoding];
    NSString      *password = @"88888888";

    // USC_ALG_SIGN_SHA1WithRSA_PKCS      : 1112
    // USC_ALG_SIGN_SHA256WithRSA_PSS     : 1123
    NSInteger     alg = USC_ALG_SIGN_SHA1WithRSA_PKCS;
    NSData   *ret = [toolkit logicSignature:self.app.mCert password:password data:data
algorithm:alg error:&error];
    NSString *retString = [NSString base64Encode:ret error:&error];
    NSLog(@"signature : %@", retString);
```

2.12. 전자봉투

2.12.1. Interface

구 분	내 용	비 고
클래스	USToolkitMgr	
함수	- (NSData *) CMS_EnvelopedData:(NSInteger)encAlg kmCert:(NSData *)kmCert inputData:(NSData *)inputData error:(USError **)error	

파라미터	encAlg	암호화 알고리즘	
	kmCert	인증서 데이터	
	inputData	암호화할 원문 데이터	
	Error	오류 발생시 상세 오류 내용을 리턴 성공 : nil 실패 : USError 객체 (오류코드, 메시지는 Error 챕터 참조)	
설 명	PKCS#7 EnvelopedData 기반의 데이터 암호화		
Return	NSData	전자봉투로 암호화된 결과 데이터	

2.12.2. 샘플

```
USError *error = nil;
USToolkitMgr *toolkit = [USToolkitMgr getInstance:&error];

if(toolkit == nil) {
    NSLog(@"toolkit is nil");
    return;
}

NSData   *data = nil;

NSString      *test = @"test1234";
data = [test dataUsingEncoding:NSUTF8StringEncoding];

 NSData* kmCert = [NSData
base64Decode:@"MIIFoDCCBIigAwIBAgICVrMwDQYJKoZIhvcNAQELBQAwUzELMAkGA1UEBh
MCS1IxEjAQBgNVBAoMCUNyb3NzQ2VydDEVMBMGA1UECwwMQWNjcmVkaXRlZENBMRkwF
wYDVQQDDBBDcm9zc0NlcnRUZXN0Q0EyMB4XDTEyMDcyNDA1MDkwMFoXDTEzMDcyNDE0
NTk1OVowTDELMAkGA1UEBhMCS1IxEjAQBgNVBAoMCUNyb3NzQ2VydDEVMBMGA1UECww
MQWNjcmVkaXRlZENBMRIwEAYDVQQDDAlUZXN0IFBBTjEwggEiMA0GCSqGSIb3DQEBAQUA
A4IBDwAwggEKAoIBAQDKz8vk85JOg5umDgGr/rh1sA/I5G4DlcoboecCA8WM+Jo/eoBPQo6Xx
RFKQAOZpD1C3Sz0uKFTWJm8WCoCIDrhU8La9q98qwmtunw2g+zoFm6aayUVd3raSp8FDokQ
```

```
CSQGFPz+j8pbPS2kbdu3StVTjJ+q4WJEOX8fyidBTk0x3RO2tDvk5D0+XSJoixTlvaK8RcweQ1EJxj
6Dfb6kHIBpHdRJAXRm9xYnBiMdpHRBDQuoRd0MXadcDwoeqWpo60wiGxaCp6CRn1qLEHINS
0e0DX1maBHLtU5IZ2X4119TNPkoVyIqWTCyUG3Vn1Bo5LBMhSSHPeUQ2u01FyKg+t1bAgMB
AAGjggKDMIICfzCBkwYDVR0jBIGLMIGIgBQS095hufkygYNWNXfsuJrVX+HaLqFtpGswaTELMAk
GA1UEBhMCS1IxDTALBgNVBAoMBEtJU0ExLjAsBgNVBAsMJUtvcmVhIENlcnRpZmljYXRpb24gQ
XV0aG9yaXR5IENlbnRyYWwxGzAZBgNVBAMMEktpc2EgVGVzdCBSb290Q0EgNYIBBDAdBgNV
HQ4EFgQUYZ7asItYRbRsh2vRJYIOgcmP67UwDgYDVR0PAQH/BAQDAgUgMH8GA1UdIAEB/w
R1MHMwcQYKKoMajJpEBQQBATBjMC0GCCsGAQUFBwIBFiFodHRwOi8vZ2NhLmNyb3NzY2Vy
dC5jb20vY3BzLmh0bWwwMgYIKwYBBQUHAgIwJh4kx3QAIMd4yZ3BHLKUACDRTMKk0rgAIMd
4yZ3BHMeFssiy5AAuMGgGA1UdEQRhMF+gXQYJKoMajJpECgEBoFAwTgwJVGVzdCBQQU4xM
EEwPwYKKoMajJpECgEBATAxMAsGCWCGSAFlAwQCAaAiBCA0WxlTeulybnajQamWy843aKDAF
36JkTCzteWwYldXZzCBgAYDVR0fBHkwdzB1oHOgcYZvbGRhcDovL3Rlc3RkaXIuY3Jvc3NjZXJ0L
mNvbTozODkvY249czFkcDZwNSxvdT1jcmxkcCxvdT1BY2NyZWRpdGVkQ0Esbz1Dcm9zc0Nlcn
QsYz1LUj9jZXJ0aWZpY2F0ZVJldm9jYXRpb25MaXN0MEoGCCsGAQUFBwEBBD4wPDA6BggrBg
EFBQcwAYYuaHR0cDovL3Rlc3RvY3NwLmNyb3NzY2VydC5jb206MTQyMDMvT0NTUFNlcnZlcjA
NBgkqhkiG9w0BAQsFAAOCAQEAJHAdOXI4YqW1i1cRNmrY7ttpCp7MIPTSH4vYReikN1vE30EC
vcAXbS0hx3j+j81xMNs3o85uaFZmgkMRJZqgP6zJYr1Ni0E8qnqAvIpisq/mt/0ZCzHgGTdVxSDm
gPrhO7l3v3brKXfM5tata2/gYHLoVjMB6r5GHn6AMIxYlADGPfP/M8qq1a8TZUniJG8exMBfC24a
1m9O6u8iGzIxsF2nLrBZRH5opO2IdhONdcQK+P5zX4W9er79twNTjDrhKvVjHndu7IGrClGaApT
D+TyRYlO8uc4TOxCGPNnSGgxyazROBP7Z7yojymHAKLWqWQvR3/ldCLk8u4cMV8QNXQ=="
error:&error];

    NSInteger     alg = USC_ALG_SYMMENC_SEED_CBC;
    NSData    *ret = [toolkit CMS_EnvelopedData:alg kmCert:kmCert inputData:data
error:&error];
    NSString *retString = [NSString base64Encode:ret error:&error];
    NSLog(@"EnvelopedData : %@", retString);
```

2.13. RSA 암호화

2.13.1. Interface

구 분	내 용		비 고
클래스	USToolkitMgr		
함수	- (NSData *) cryptRSA: pubkey: data: error:;		
파라미터	encAlg	암호화 알고리즘	
	pubkey	공개키 데이터	
	data	암호화할 원문 데이터	
	Error	오류 발생시 상세 오류 내용을 리턴 성공 : nil 실패 : USError 객체 (오류코드, 메시지는 Error 챕터 참조)	
설 명	키체인에서 인증서 삭제		
Return	NSData	암호화된 결과 데이터	

2.13.2. 샘플

```
    USError *error = nil;
    USToolkitMgr *toolkit = [USToolkitMgr getInstance:&error];
    NSData   *data = [@"data test" dataUsingEncoding:NSUTF8StringEncoding];
    NSString      *pubkeyString = [self.app.mCert publicKey];
    NSData   *pubkey = [NSData hexStringToBin:pubkeyString error:&error];
    NSData   *ret = [toolkit cryptRSA:USC_ALG_ASYMM_RSA1024 pubkey:pubkey data:data
error:&error];

    NSLog(@"ret : %@", ret);
```

2.14. SEED 암호화

2.14.1. Interface

구 분	내 용		비 고
클래스	USToolkitMgr		
함수	- (NSData *) cryptSeed: key: iv: error:;		
파라미터	data	암호화할 원문	
	Key	암호화 키	
	IV	암호화 IV	
	Error	오류 발생시 상세 오류 내용을 리턴 성공 : nil 실패 : USError 객체 (오류코드, 메시지는 Error 챕터 참조)	
설 명	키체인에서 인증서 삭제		
Return	NSData	암호화된 결과 데이터	

2.14.2. 샘플

```
USError *error = nil;
USToolkitMgr *toolkit = [USToolkitMgr getInstance:&error];

NSData   *data = [@"data test" dataUsingEncoding:NSUTF8StringEncoding];
NSData   *key =   [toolkit CRYPT_GenerateRandom:16 error:&error];
NSData   *iv = [toolkit CRYPT_GenerateRandom:16 error:&error];

NSData   *result = [toolkit cryptSeed:data key:key iv:iv error:&error];
```

2.15. 인증서 신원확인

2.15.1. Interface

구 분	내 용		비 고
클래스	USToolkitMgr		
함수	- (BOOL) CERT_VerifyVID:(NSData *)cert vidRandom:(NSData *)vidRandom socialNumber:(NSString *)socialNumber error:(USError **)error;		
파라미터	cert	인증서	
	vidRandom	vidRandom 값	
	socialNumber	인증서 소유자 주민등록번호	
	Error	오류 발생시 상세 오류 내용을 리턴 성공 : nil 실패 : USError 객체 (오류코드, 메시지는 Error 챕터 참조)	
설 명	인증서 신원확인		
Return	NSData	신원확인 결과	

2.15.2. 샘플

```
USError *error = nil;
USToolkitMgr *toolkit = [USToolkitMgr getInstance:&error];
if(toolkit == nil)
{
    [self showAlert:@"본인확인" message:@"툴킷이 초기화 되지 않았습니다."];
    return;
}
if(self.app.mCert == nil)
{
    [self showAlert:@"본인확인" message:@"선택된 인증서가 없습니다."];
    return;
}
```

```
    NSString *pw = @"avirexu12@";// 테스트 인증서 비밀번호
    NSData   *vidR = [toolkit CERT_GetVIDRandomWithPrikey:pw encPrikey:[self.app.mCert
dataForType:kUSSignPrikey] error:&error];
    // 인증서 사용자 주민등록번호 또는 사업자 번호
    NSString *sID = @"3481820005096";// 테스트 인증서 주민번호
    BOOL      ret = [toolkit CERT_VerifyVID:[self.app.mCert dataForType:kUSSignCert]
vidRandom:vidR socialNumber:sID error:&error];
    if(ret && nil == error)
    {
        [self showAlert:@"본인확인" message:@"본인 확인 검증에 성공하였습니다."];
    }
    else
    {
        [self showAlert:@"본인확인" message:[NSString stringWithFormat:@"본인 확인
검증에 실패하였습니다.₩n(%ld)", (long)[error code]]];
    }
```

2.16. 공인인증서 이동(승인번호 생성)

2.16.1. API

구분	내 용		비 고
클래스	USTransfer		
함수	- (NSString *) TRANS_GenerateCertNum:(USTransDeviceOSType)type direction:(USTransDirection)direction serial:(NSString *)serial error:(USError **)error;		
파라미터	type direction serial error	디바이스 OS 구분 가져오기, 내보내기 구분 시리얼 번호 에러 객체	
설명	인증서 이동 중계서버로부터 인증서 이동에 필요한 승인번호 획득		

Return	NSString Error	승인번호 13 자리 ✓ 성공 : [resultMessage result] == YES ✓ 실패 : [resultMessage result] == NO ✓ 실패시 오류코드 :[resultMessage errorCode] 4.TroubleShooting 참조 ✓ 실패시 메시지 : [resultMessage errorMessage] 4.TroubleShooting 참조	

2.16.2. Sample

```
self.transfer = [[USTransferMgr alloc] init:kUSTransV1_0
                                      appID:nil
                                     appKey:nil
                                       type:kUSTransOwn
                                      error:&error];


    if(error != nil && 0 != [error code])
    {
        NSLog(@"[error code] : %ld",(long)[error code]);
        NSString *initResult = [NSString stringWithFormat:@"툴킷 초기화에
실패했습니다.₩n(%ld)", (long)[error code]];

        [self showAlert:@"오류" message:initResult];
        return;
    }
    NSString *result = nil;
    switch (self.moveType) {
        case 0:
            self.lbTitle.text = @"공인인증서 가져오기";
            result = [self.transfer TRANS_GenerateCertNum:kUSTransIOS
                                                direction:kUSTransImport
                                                   serial:[USUtil serial]
```

```
                                                    error:&error];
            break;
        case 1:
            self.lbTitle.text = @"공인인증서 내보내기";
            result = [self.transfer TRANS_GenerateCertNum:kUSTransIOS
                                                direction:kUSTransExport
                                                   serial:[USUtil serial]
                                                    error:&error];
            break;
        default:
            break;
    }
    if(nil == result || 13 != [result length])
    {
        result = [NSString stringWithFormat:@"(%ld)", (long)[error code]];
        NSLog(@"승인번호 생성 실패 :%@", result);
        [self showAlert:@"오류" message:result];
        return;
    }

    NSString *generateApprovalNumber = result;

    NSString *firstString   = [generateApprovalNumber
substringWithRange:NSMakeRange(0,4)];
    NSString *secondString = [generateApprovalNumber
substringWithRange:NSMakeRange(4,4)];
    NSString *thirdString = [generateApprovalNumber
substringWithRange:NSMakeRange(8,5)];

    if(generateApprovalNumber != nil && [generateApprovalNumber length] == 13)
    {
        firstString   = [generateApprovalNumber substringWithRange:NSMakeRange(0,4)];
        secondString = [generateApprovalNumber substringWithRange:NSMakeRange(4,4)];
        thirdString = [generateApprovalNumber substringWithRange:NSMakeRange(8,5)];
```

```
        NSLog(@"firstString : %@",firstString);
        NSLog(@"secondString : %@",secondString);
        NSLog(@"thirdString : %@",thirdString);

        self.tfFirstApprovalNumber.text = firstString;
        self.tfSecondApprovalNumber.text = secondString;
        self.tfThirdApprovalNumber.text = thirdString;

        self.waitTimerEnabled = YES;

        SEL selOp = @selector(isPCconnected);
        self.threadOperation = [[NSThread alloc] initWithTarget:self selector:selOp
object:nil];
        [self.threadOperation start];
    }
```

2.17. 공인인증서 이동(중계서버 접속 및 결과 수신)

2.17.1. API

구분	내 용		비 고
클래스	USTransfer		
함수	-(void) isPCconnected;		
파라미터			
설명	인증서 이동시 중계서버와 연결		
Return	void	-(void)onResult:(ResultMessage *)result;에서 인증서 가져오기 결과를 수신하고 정상일 경우 인증서 비밀번호 입력 페이지로 이동. ✓ 성공 : [resultMessage result] == YES	

		✓ 실패 : [resultMessage result] == NO ✓ 실패시 오류코드 :[resultMessage errorCode] 4.TroubleShooting 참조 ✓ 실패시 메시지 : [resultMessage errorMessage] 4.TroubleShooting 참조	

2.17.2. Sample

```
-(void) isPCconnected{
    NSLog(@"       □isPCconnected");
    NSAutoreleasePool* pool = [[NSAutoreleasePool alloc] init];
    USError* error = nil;
    if ([self.transfer TRANS_IsPCConnected:&error]) {
        [self.waitTimer invalidate];

        [self performSelectorOnMainThread:@selector(showHUD)
                                  withObject:nil
                               waitUntilDone:YES];

        SEL selector;
        switch (self.moveType) {
            case 0:
                selector = @selector(import);
                break;
            case 1:
                selector = @selector(export:);
                break;
            default:
                selector = nil;
                break;
        }
        [NSThread detachNewThreadSelector:selector toTarget:self withObject:nil];
```

```
    } else {
        NSLog(@"PC Connection fail");
    }

    [pool release];
}
```