

iOS용 MorningSDK v1.1.0  
-개발자 매뉴얼-

2024. 3.

Morningsoft

## 제•개정 이력

순번	제•개정 일	제•개정 내역
1	2022.05	“iOS용 개발자 매뉴얼” 발간
2	2023.05	Swift 5.0, 개발자 라이선스 지원
3	2024.01	라이선스 기능 추가
4	2024.03	전자 결제, HTTP 2.0 지원

## Contents

---

1. MorningSDK v1.1.0 소개
2. 패키지 설치 및 빌드
3. MorningSDK 사용 샘플
4. iOS용 API 목록

# 1. MorningSDK 소개

## 가. MorningSDK

모바일 앱의 해킹 기법들이 발달함에 따라 앱 마켓에서 서비스하는 앱들이 최신 기술을 활용하여 해킹되는 사례가 발생하고 있습니다. 고객을 위해 서비스를 제공하는 기관이나 기업들은 이런 최신 기술이 포함된 해킹 도구들로 부터 개인정보를 보호할 수 있는 방안이 필요합니다. MorningSDK 모바일 라이브러리는 전세계적으로 알려진 해킹 도구들인 Frida, FlyJB 등의 설치를 확인하여 민감정보의 보호가 필요한 앱의 실행을 방지하는 기능을 포함하고 있습니다. 또한 일부 해킹 소프트웨어들을 이용해서 디버깅 혹은 메모리 침범등을 시도할 경우 이를 사전에 방지하는 기능을 가지고 있습니다. 따라서 이미 알려진 해킹 도구만을 차단하는것이 아닌 새로운 해킹 도구를 차단할 수 있는 도구를 포함합니다.

또한 민감정보(사용자 아이디, 주민번호) 문자열이 소스코드에 포함되었을때 이를 난독화 하여 이미 배포한 앱으로 부터 해당 정보의 추출을 어렵게 만드는 기능을 포함하고 있습니다.

## 나. 세부 기능

포함 기능	분류
앱 무결성 및 위변조 방지	Integrity
문자열 난독화	Obfuscation
민감정보 저장	Secure Storage
암호화	Cryptography
고유 식별자 발급	AUUUID
멀웨어 탐지	Vaccine
화면 캡처 방지	DLP
mVoip 보안	VOIP
전자결제 및 문서 위변조 방지	sign

- AUUUID : Application Universally Unique Identifier
- DLP : Data Loss Prevention

## 나. 컴파일 및 실행 환경

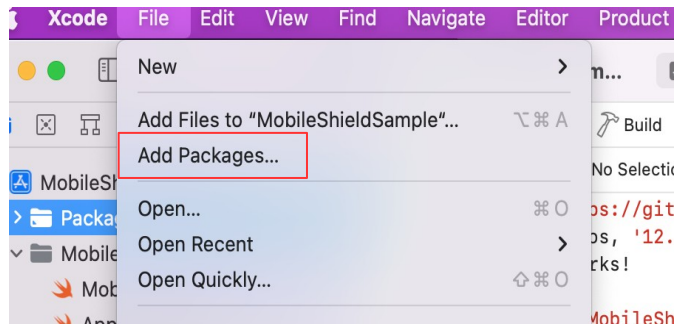
항목	내용
xCode	14.0.1 이상
Swift	5.0 이상
실행 기기	iphone 6 이상 (arm64 이상)

실행 운영체제	iOS 12.4 이상
---------	-------------

## 2. 패키지 설치 및 빌드

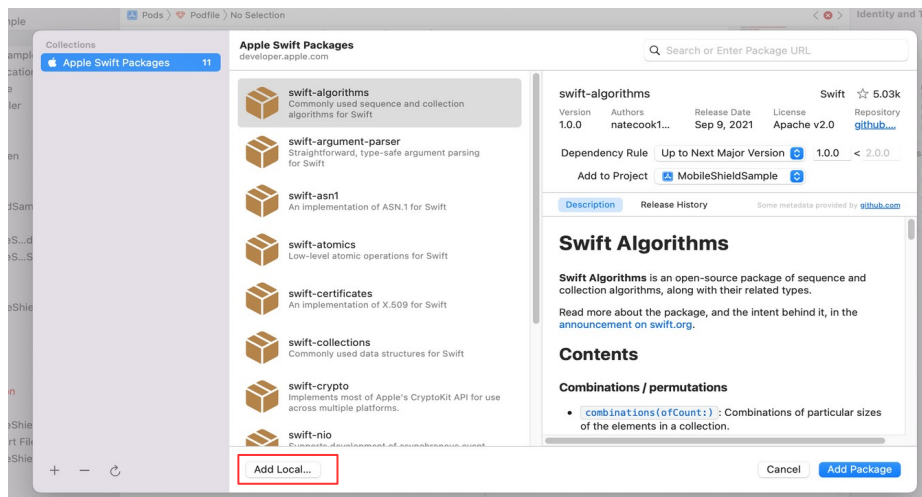
가. 프로젝트에 패키지 추가

- xCode의 상단 메뉴에서 “Add Packages” 를 선택합니다.



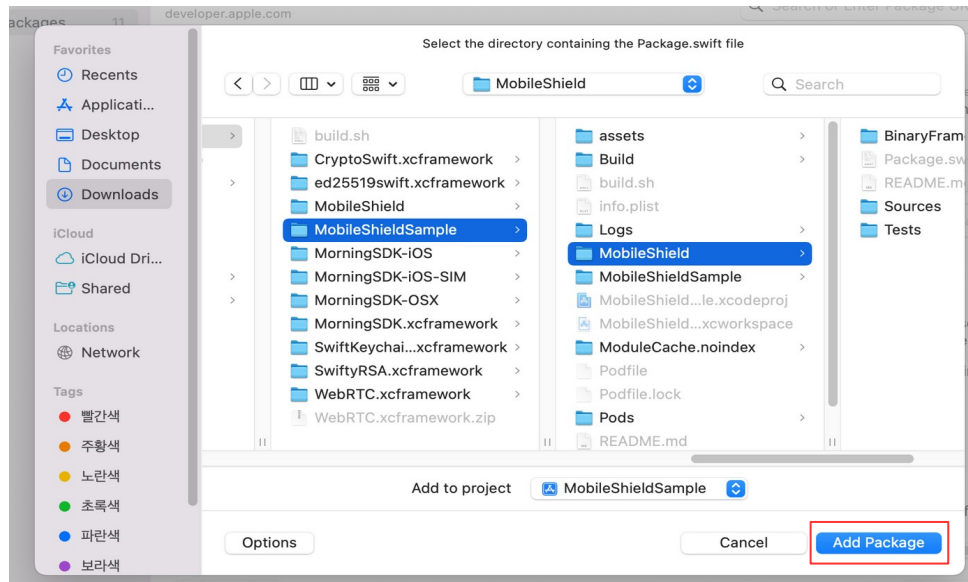
나. 로컬 패키지 추가

- 팝업된 대화 상자에서 “Add Local”을 선택합니다.



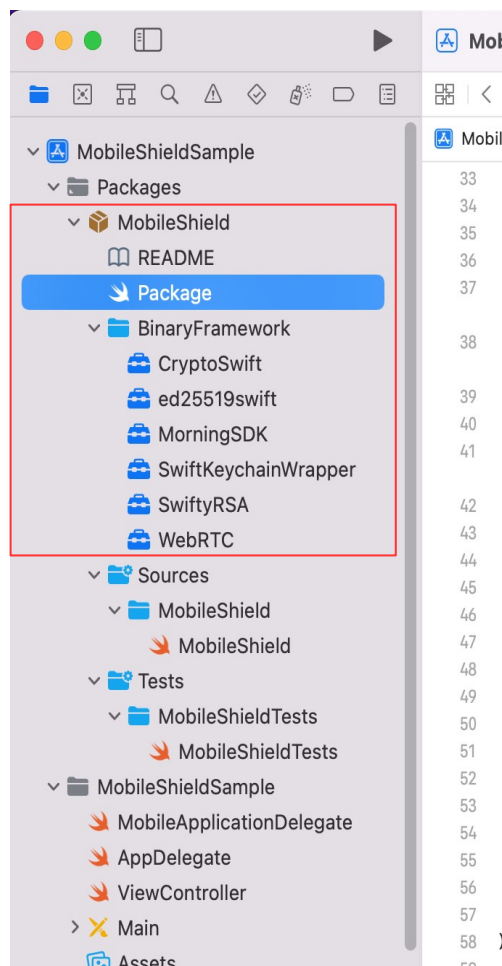
#### 다. 프로젝트에 패키지 추가

- 패키지가 포함된 “MorningSDK” 폴더를 선택하고 “Add Package” 버튼을 클릭합니다.
- MorningSDK/Library/iOS/Swift/MorningSDK.zip 경로의 파일을 압축을 풀어 패키지를 추가해 줍니다.



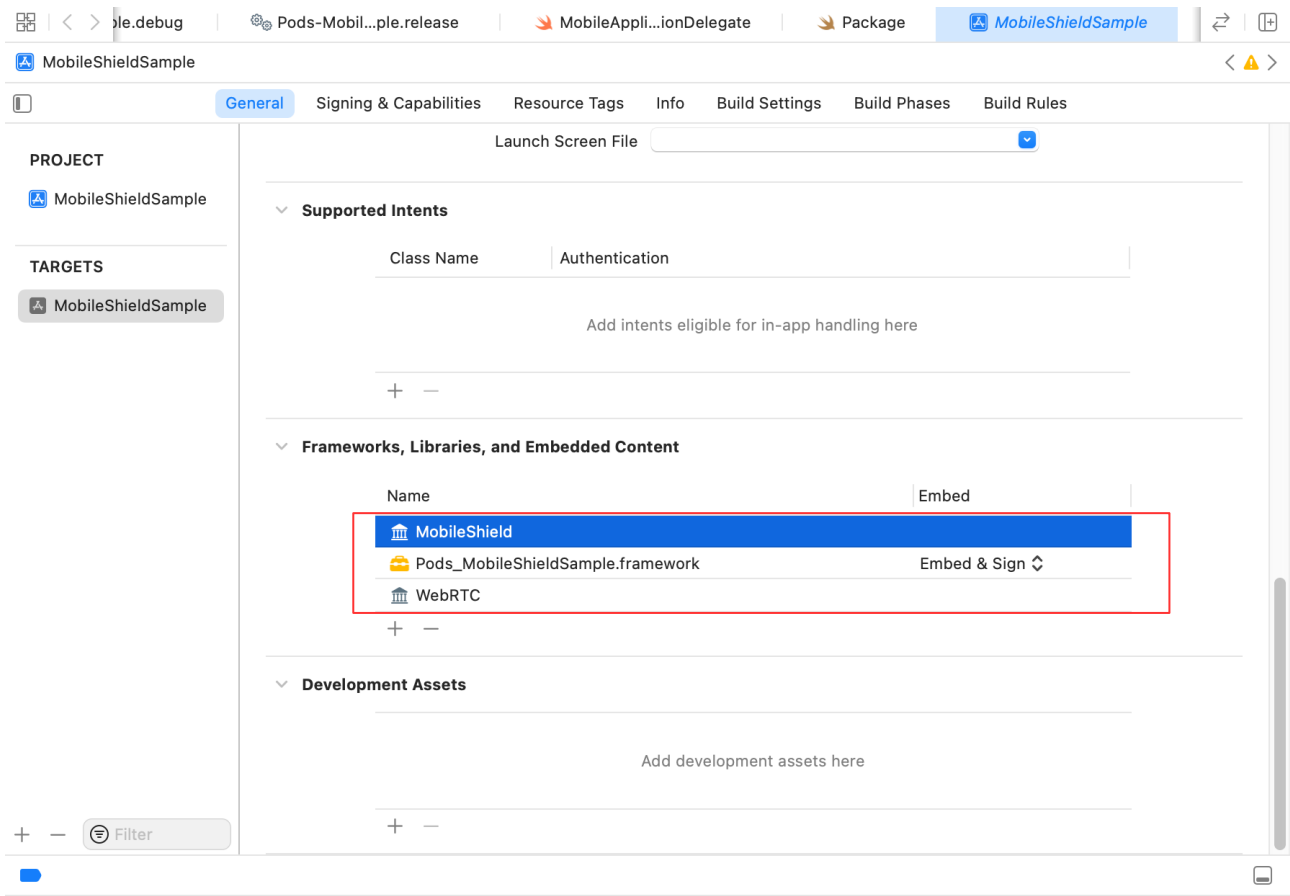
#### 라. 워크스페이스 패키지 추가 확인

- xCode의 워크스페이스 창에서 추가된 MorningSDK 패키지를 확인할 수 있습니다.



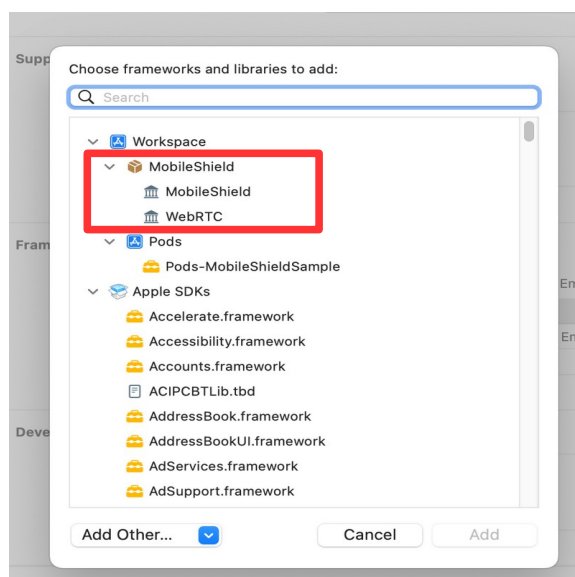
마. 프로젝트에서 사용하기 위한 Framework 추가

- General → Frameworks, Libraries, and Embedded Content에 사용 Framework 추가합니다.



바. 프레임워크를 리스트에서 선택

- 패키지에서 MorningSDK Framework를 추가한다.



사. 소스코드에서 해당 Framework 사용



- 추가된 패키지는 프로젝트에서 “import MorningSDK” 문장을 통해서 사용할 수 있습니다.



#### 아. 라이브러리 사용시 필수 설정 값들 확인

- 소스코드에서 다음과 같이 필수 설정값들을 설정해야 사용할 수 있습니다.

```
override func viewDidLoad() {
    MorningSDK.setLicense(licenseData : “Your-License-Data”, configData :
    “voip.yourdomain.com\ncom.yourdomain.voip\nnyourdomain.com\nYour-Developer-License”)
    ...
}
```

## 자. 프로젝트 빌드를 위한 설정 확인

- 프로젝트 기본 설정을 swift 5를 선택합니다.

Build Settings

Basic Customized All Combined Levels SWIFT\_VERSION

Swift Compiler - Language

Setting MobileShield

Swift Language Version Swift 5

## - 프로젝트 설정 확인

Build Phases

Filter

Target Dependencies (0 items)

Run Build Tool Plug-ins (0 items)

[CP] Check Pods Manifest.lock

Compile Sources (3 items)

Link Binary With Libraries (3 items)

Name	Status
Pods_MobileShieldSample.framework	Required
WebRTC	Required
MobileShield	Required

Drag to reorder linked binaries

Copy Bundle Resources (3 items)

Embed Frameworks (1 item)

### 3. MorningSDK 사용 샘플

#### 가. 라이브러리 초기화 샘플

##### 라이브러리 초기화 샘플

```
MorningSDK.setLicense(mContext, "XXXX", "A\\nB\\nC\\nD")
MorningSDK.setTamperProtectEnable(true)
MorningSDK.setMagiskHideProtectEnable(true)
MorningSDK.setDebuggerProtectEnable(true)
MorningSDK.setSignatureVerifyEnable(true)
MorningSDK.setFridaProtectEnable(true)
MorningSDK.setRootingCheckEnable(true)
MorningSDK.setScreenProtectEnable(true)
MorningSDK.setMarketVerifyEnable(true)
MorningSDK.setDebuggableProtectEnable(true)
```

#### 나. 키체인을 이용한 데이터 저장 샘플

##### 키체인을 이용한 데이터 저장 샘플

```
// 앱에서 데이터를 저장하기 위해서는 반드시 암호화 해야 함
// MorningSDK는 iOS KeyChain을 이용해서 저장하므로 사용자의 암호화 키를 사용하지 않음
// 사용자가 기기를 잠금 상태에 따라 Keychain의 잠금 상태도 동일하다. (기기 잠금해제 = 키 체인 잠금해제)
var mUserPassword : String = "password##!!"
print("User Password : \(mUserPassword) ")
MorningSDK.writeString(key : "UserPassword", value : mUserPassword)
print("User Password : \(MorningSDK.readString(key : "UserPassword")) ")
var mUserNumber : Int = "12341234"
print("User Number : \(mUserNumber) ")
MorningSDK.writeInteger(key : "UserNumber", value : mUserNumber)
print("User Number : \(MorningSDK.readInteger(key : "UserNumber")) ")
var mUserLogin : Bool = true
MorningSDK.writeBool(key : "UserLogin", value : mUserLogin)
print("User Login : \(MorningSDK.readBool(key : "UserLogin")) ")
```

#### 다. 민감 정보 문자열 난독화 샘플

##### 민감 정보 문자열 난독화 샘플

```
// 민감정보인 대칭키 암호화 패스워드를 사용하는 부분
// 기존 소스코드
var mAESPassword : String = "password##!!"
print("AES Password : \(mAESPassword) ")

// IDS_AES_PASSWORD는 프로그램 빌드전 암호화 되어 추가됨
// 중요 정보가 앱의 소스코드에 문자열로 포함되지 않아 안전함
// 수정된 소스코드
print("AES Password : \(MorningSDK.getObfuscation(IDS_AES_PASSWORD)) ")
```

## 4. iOS용 API 목록

### 가. 앱 무결성 및 위변조 방지

함수 원형	fun securityCheck()
설명	MorningSDKController의 멤버 함수로 주요 UIViewController에서 MorningSDKController를 상속받아 사용할 경우 자동으로 호출됨
매개변수	없음
결과값	무결성 검증 실패 및 멀웨어가 탐지 되었을 경우 앱을 종료함
비고	

함수 원형	func isJailbrokenEnabled() -> Bool
설명	Jailbreak 방지기능의 설정 여부
파라미터	없음
결과값	true : 설정됨 false : 설정 안됨
비고	

함수 원형	func isRunInEmulatorEnabled() -> Bool
설명	에뮬레이터에서 실행하는 것을 방지하는 기능의 설정 여부
파라미터	없음
결과값	true : 설정됨 false : 설정 안됨
비고	

함수 원형	func isDebuggedEnabled() -> Bool
설명	디버깅을 실행하는 것을 방지하는 기능의 설정 여부
파라미터	없음
결과값	true : 설정됨 false : 설정 안됨
비고	

함수 원형	func isTamperedEnabled() -> Bool
설명	파일 템퍼를 방지하는 기능의 설정 여부
파라미터	없음
결과값	true : 설정됨 false : 설정 안됨
비고	

함수 원형	func isReverseEngineeredEnabled() -> Bool
설명	역컴파일 방지하는 기능의 설정 여부
파라미터	없음

결과값	true : 설정 됨 false : 설정 안됨
비고	

함수 원형	func isRuntimeHookedEnabled() -> Bool
설명	실시간 후킹을 방지하는 기능의 설정 여부
파라미터	없음
결과값	true : 설정 됨 false : 설정 안됨
비고	

함수 원형	func isProxiedEnabled() -> Bool
설명	프록시를 방지하는 기능의 설정 여부
파라미터	없음
결과값	true : 설정 됨 false : 설정 안됨
비고	

함수 원형	func isBreakpointAtEnabled() -> Bool
설명	중단점 설정을 방지하는 기능의 설정 여부
파라미터	없음
결과값	true : 설정 됨 false : 설정 안됨
비고	

함수 원형	func setJailbrokenEnabled(enable : Bool?)
설명	Jailbreak 방지기능을 설정
파라미터	enable : 설정값
결과값	없음
비고	

함수 원형	func setRunInEmulatorEnabled(enable : Bool?)
설명	에뮬레이터에서 실행하는 것을 방지하는 기능을 설정
파라미터	enable : 설정값
결과값	없음
비고	

함수 원형	func setDebuggedEnabled(enable : Bool?)
설명	디버깅을 실행하는 것을 방지하는 기능을 설정
파라미터	enable : 설정값
결과값	없음

비고	
----	--

함수 원형	func setTamperedEnabled(enable : Bool?)
설명	파일 템퍼(Tamper)를 방지하는 기능을 설정
파라미터	enable : 설정값
결과값	없음
비고	

함수 원형	func setReverseEngineeredEnabled(enable : Bool?)
설명	역컴파일 방지하는 기능을 설정
파라미터	enable : 설정값
결과값	없음
비고	

함수 원형	func setRuntimeHookedEnabled(enable : Bool?)
설명	실시간 후킹을 방지하는 기능을 설정
파라미터	enable : 설정값
결과값	없음
비고	

함수 원형	func setProxiedEnabled(enable : Bool?)
설명	프록시를 방지하는 기능을 설정
파라미터	enable : 설정값
결과값	없음
비고	

함수 원형	func setBreakpointAtEnabled(enable : Bool?)
설명	중단점 설정을 방지하는 기능을 설정
파라미터	enable : 설정값
결과값	없음
비고	

## 나. 문자열 난독화

함수 원형	func getObfuscation(symbol: String) -> String?
설명	난독화 문자열을 복원함
파라미터	symbol : 문자열을 구분할 수 있는 키값
결과값	복원된 평문
비고	

함수 원형	func setObfuscation(obfuscationData: String) -> void
설명	난독화 데이터 설정
파라미터	obfuscationData : 별도의 프로그램으로 생성한 난독화 데이터
결과값	없음
비고	

함수 원형	func isStringObfuscationEnabled() -> Bool
설명	문자열 난독화 기능의 설정 여부
파라미터	없음
결과값	true : 설정 됨 false : 설정 안됨
비고	

함수 원형	func setStringObfuscationEnabled(enable : Bool?)
설명	문자열 난독화 기능을 설정
파라미터	enable : 설정값
결과값	없음
비고	

#### 다. 민감정보 저장

함수 원형	func writeString(key: String?, value: String?)
설명	(키, 텍스트) 형식의 데이터를 keychain을 이용해서 안전하게 저장함
파라미터	key: 데이터와 연관된 키값 value : 텍스트 데이터
결과값	없음
비고	

함수 원형	func readString(key: String) -> String?
설명	Key에 해당하는 텍스트 형식의 데이터를 얻어옴
파라미터	key: 데이터와 연관된 키값
결과값	텍스트 데이터(민감정보)
비고	

함수 원형	func writeInteger(key: String?, value: Int?)
설명	(키, 정수) 형식의 데이터를 keychain을 이용해서 안전하게 저장함
파라미터	key: 데이터와 연관된 키값 value : 정수 데이터
결과값	없음
비고	

함수 원형	func readInteger(key: String) -> Int
설명	Key에 해당하는 정수 형식의 데이터를 얻어옴
파라미터	key: 데이터와 연관된 키값
결과값	정수 데이터
비고	

함수 원형	func writeBool(key: String?, value: Bool?)
설명	(키, 이진) 형식의 데이터를 keychain을 이용해서 안전하게 저장함
파라미터	key: 데이터와 연관된 키값 value : 이진 데이터
결과값	없음
비고	

함수 원형	func readBool(key: String) -> Bool
설명	Key에 해당하는 이진 형식의 데이터를 얻어옴
파라미터	key: 데이터와 연관된 키값
결과값	이진 데이터
비고	

## 라. 암호화

함수 원형	func initRSA2048String()
설명	텍스트(String)의 RSA 암호화를 위한 public/private 키 생성(2048 bits)
파라미터	없음
결과값	없음
비고	

함수 원형	func encryptRSA2048String(plainText: String) -> String
설명	텍스트(String)의 RSA 암호화를 수행하여 암호문을 생성함
파라미터	plainText : 평문 텍스트
결과값	Base64형식의 암호문 텍스트
비고	암호화를 위한 공용키는 keychain에 저장된 최근 키를 사용함 만약 키가 생성되지 않았을 경우 암호화 전에 자동 생성됨

함수 원형	func decryptRSA2048String(cipherText: String) -> String
설명	텍스트(String)의 RSA 복호화를 수행하여 평문을 생성함
파라미터	cipherText : Base64형식의 암호문 텍스트
결과값	평문 텍스트
비고	복호화를 위한 공용키는 keychain에 저장된 최근 키를 사용함

함수 원형	func initAES256String()
-------	-------------------------



설명	AES(256bits)로 평문 텍스트(String)을 암호화 하기 위한 키 생성 함수
파라미터	없음
결과값	없음
비고	이전 키는 초기화되어 더이상 사용하지 못함

함수 원형	func encryptAES256String(plainText: String) -> String
설명	AES(256bits)로 평문 텍스트(String)를 암호화 하여 암호문을 생성함
파라미터	plainText : 평문 텍스트(String)
결과값	없음
비고	암호화를 위한 키는 자동 생성되어 keychain에 안전하게 보관됨

함수 원형	func decryptAES256String(cipherText: String) -> String
설명	AES(256bits)로 암호문 텍스트(String)를 복호화 하여 평문을 얻음
파라미터	cipherText : Base64로 인코딩된 암호문
결과값	평문 텍스트(String)
비고	복호화를 위한 키는 자동 생성되어 keychain에 안전하게 보관됨

함수 원형	func computeSHA256String(plainText: String) -> String
설명	SHA256으로 해시 문자열 생성
매개변수	plainText : 평문 텍스트
결과값	해시 텍스트(String)
비고	

#### 마. 고유 식별자 발급

함수 원형	func getAUUID()-> String
설명	Application UUID(고유 식별값)를 생성함, 모바일 단말의 식별자로 사용할 수 있음
파라미터	없음
결과값	AUUUID
비고	앱의 재설치 및 앱 데이터 삭제시 AUUUID는 새로 생성됨

#### 바. 멀웨어 탐지

함수 원형	func runScan()
설명	멀웨어 스캔 시작
파라미터	없음
결과값	없음
비고	

#### 사. 화면 캡처 방지

함수 원형	func isScreenProtectEnabled() -> Bool
설명	화면 캡처 방지기능의 설정 여부
파라미터	없음
결과값	true : 설정됨 false : 설정 안됨
비고	

함수 원형	func setScreenProtectEnabled(enable : Bool?)
설명	화면 캡처 방지기능을 설정
파라미터	enable : 설정값
결과값	없음
비고	

## 아. 자가 테스트

함수 원형	func testMorningSDK()
설명	자가 테스트
매개변수	없음
결과값	없음
비고	Xcode Console로 테스트 결과를 확인할 수 있음

## 자. 라이선스 설정

함수 원형	func setLicense(licenseData: String, configData: String) -> void
설명	라이선스 데이터 설정
파라미터	LicenseData : 발급받은 라이선스데이터 configData : 사용을 위한 설정
결과값	없음
비고	

## 차. 로깅 설정

함수 원형	func setLogLevel(logLevel : Int) -> void
설명	로깅 레벨 설정
파라미터	logLevel : 로깅 레벨을 설정함
결과값	없음
비고	LogUtil.LEVEL_NONE LogUtil.LEVEL_ERROR LogUtil.LEVEL_WARN LogUtil.LEVEL_INFO LogUtil.LEVEL_DEBUG

