**云核客户端安全输入系统开发人员手册**

**(iOS版)**



北京云核网络技术有限公司

2017年5月23日

文档修改记录

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 版本 | 内容 | 编写人 | 编写日期 | 审核人 | 审核日期 |
| 1.0 | 新建 | 王雪松 | 2015-5-15 |  |  |
| 1.1 | 修正描述错误 | 王雪松 | 2017-4-20 |  |  |
| 1.2 | 增加集成要求、版本控制和密钥管理体系章节 | 王雪松 | 2017-5-23 |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

版权申明：

本文档的版权属于北京云核网络技术有限公司，任何人或组织未经许可，不得擅自修改、拷贝或以其它方式使用本文档中的内容。

目 录

[一、 引言 1](#_Toc483386006)

[1.1 编写目的 1](#_Toc483386007)

[1.2 背景知识及参考资料 1](#_Toc483386008)

[1.3 运行环境 1](#_Toc483386009)

[1.4 开发环境 1](#_Toc483386010)

[二、 概述 1](#_Toc483386011)

[2.1 组成 1](#_Toc483386012)

[2.2 功能特点 1](#_Toc483386013)

[2.3 技术特点 2](#_Toc483386014)

[三、 云核安全输入系统使用方法 2](#_Toc483386015)

[3.1 ISV集成云核安全输入系统要求 2](#_Toc483386016)

[3.2 在iOS项目中导入云核安全输入系统iOS版静态库 2](#_Toc483386017)

[3.2.1 前提条件 2](#_Toc483386018)

[3.2.2 导入云核安全输入系统iOS版软件 2](#_Toc483386019)

[3.2.3 初始化系统 5](#_Toc483386020)

[3.2.4 连接库 5](#_Toc483386021)

[3.2.5 创建云核安全输入框 6](#_Toc483386022)

[3.2.6 获取加密后的密码 7](#_Toc483386023)

[3.3 云核安全输入框接口说明 8](#_Toc483386024)

[3.3.1 云核安全输入框属性 8](#_Toc483386025)

[3.3.2 云核安全输入框方法 10](#_Toc483386026)

[3.3.3 密码长度变化通知 11](#_Toc483386027)

[3.3.4 银联算法说明 11](#_Toc483386028)

[3.4 密钥管理体系 12](#_Toc483386029)

[3.4.1 客户端密钥管理体系 12](#_Toc483386030)

[3.4.2 服务端密钥管理体系 12](#_Toc483386031)

[3.5 版本控制 12](#_Toc483386032)

[四、 Web与原生混合模型下安全输入系统的集成 12](#_Toc483386033)

[4.1 说明 12](#_Toc483386034)

[4.2 WebPasswordKeyboard类的初始化 12](#_Toc483386035)

[4.3 控件键盘的显示 13](#_Toc483386036)

[4.4 控制键盘的隐藏 13](#_Toc483386037)

[4.5 控制输入域内容 13](#_Toc483386038)

[4.6 提交表单 13](#_Toc483386039)

# 引言

## 编写目的

云核客户端安全输入系统是保护用户输入的敏感信息的重要手段，能对客户端操作系统进行安全增强，可大幅提升“木马”程序非法获取用户敏感信息“成本”的软件集。

云核客户端安全输入系统iOS版提供了替换iOS系统缺省非安全输入框的解决方案。本方案可以有效防止越狱iOS设备中非法钩子函数对原始输入框的截取。本文档定义了云核安全输入框的接口，并简要说明使用本安全输入框的开发步骤。

## 背景知识及参考资料

假定开发人员对下列技术有一定的理解：

|  |  |
| --- | --- |
| 技术 | 有关内容 |
| iOS开发技术 | XCode、iOS SDK等 |
| 密码技术 | 对称加密、非对称加密、摘要等 |

## 运行环境

云核客户端安全输入系统iOS版只能在iOS 6.0 （含）以上版本中使用。

## 开发环境

云核客户端安全输入系统iOS版需要苹果10.6（含）以上桌面系统，在Xcode集成环境下进行开发。

# 概述

## 组成

云核客户端安全输入系统iOS版由一个静态库给出，全部代码最终连接到客户软件中，最后的分发软件是不含此静态库的。

## 功能特点

* 防止Hook类木马程序攻击。
* 防止网络传输泄密。
* RSA PKCS1填充
* RSA 科友填充
* SM2 算法
* 其它加密算法
* 防止截屏泄密（有限）

## 技术特点

云核安全输入框（PasswordTextField）是基于UIKit中UITextField开发的。输入内容存放在安全输入系统的核心中，通过UITextField原有方法并不能获取到原文，有效的防止了内容泄漏。

云核安全输入框输入的内容是加密后存放在安全输入系统的核心中的，有效防止了内存扫描程序。

云核安全输入框支持自定义属性，有键盘类型、按键乱序、最大/最小长度等。

# 云核安全输入系统使用方法

## ISV集成云核安全输入系统要求

ISV在集成云核安全输入系统时，需要施加加壳等安全加固措施，防止APP被反编译和调试，增加APP被破解难度。

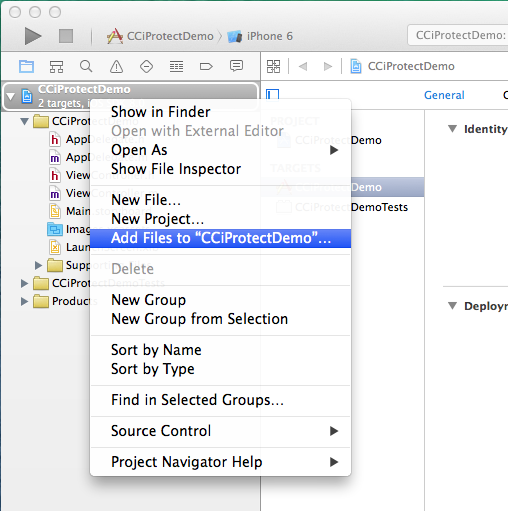
## 在iOS项目中导入云核安全输入系统iOS版静态库

### 前提条件

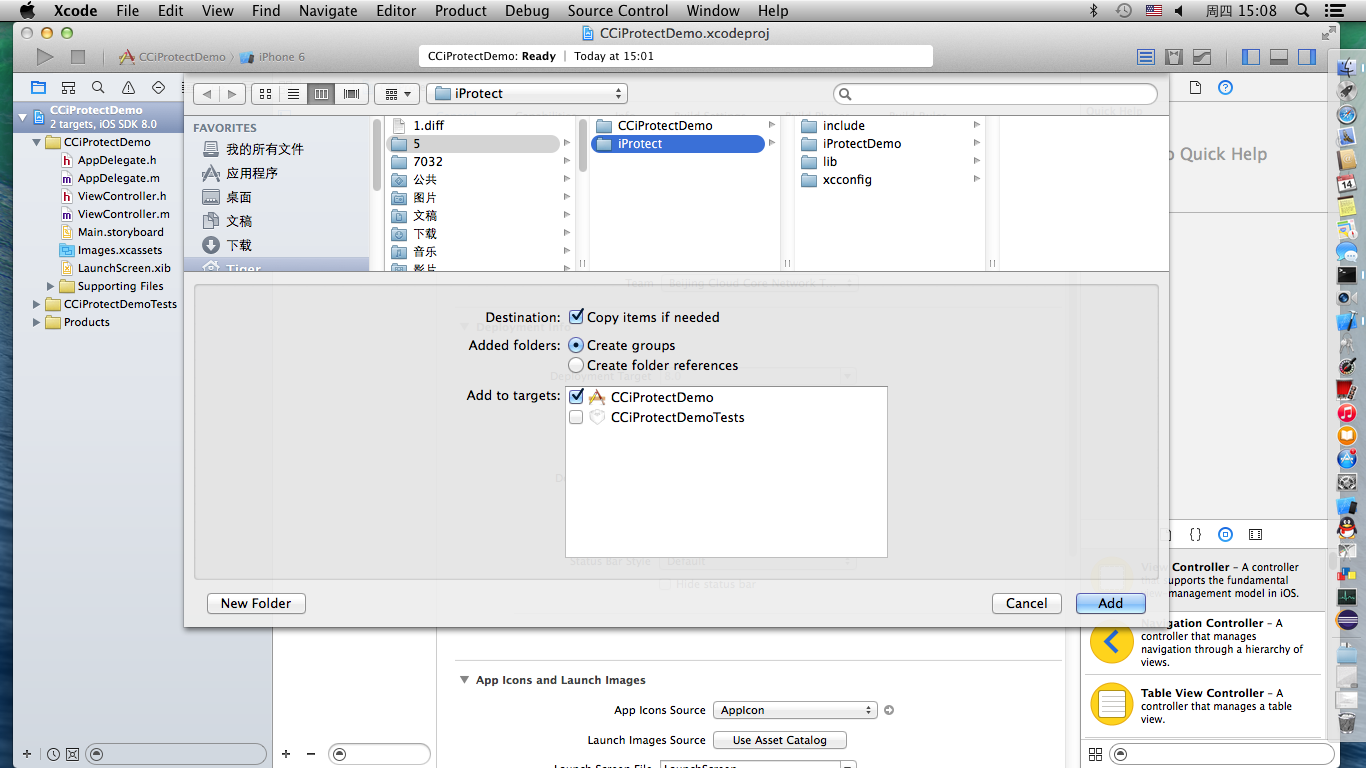
您已经通过Xcode集成开发环境创建了一个iOS软件项目，这里假设Xcode版本为6.0，项目名字CCiProtectDemo。

### 导入云核安全输入系统iOS版软件

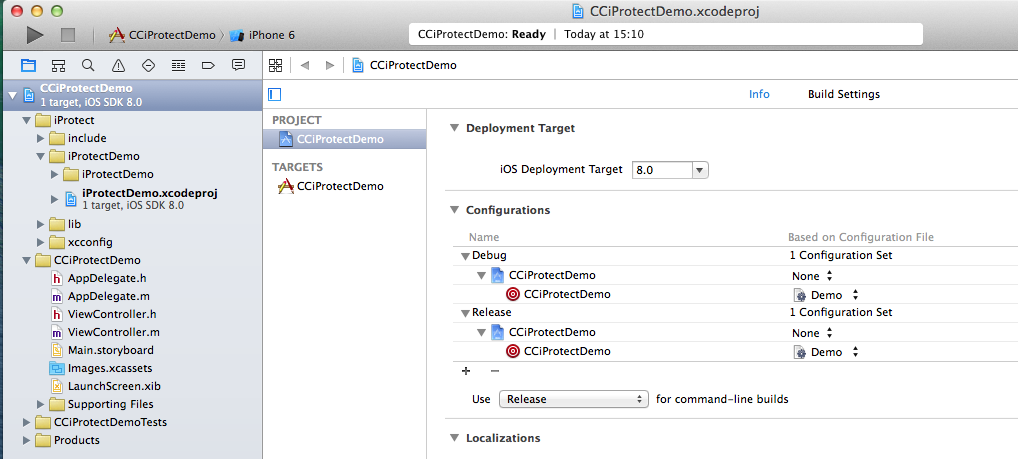
* 在Xcode的Project Navigator窗口中选中项目，单击鼠标右键，选择Add Files to “CCiProtectDemo”…菜单项，如下图所示：



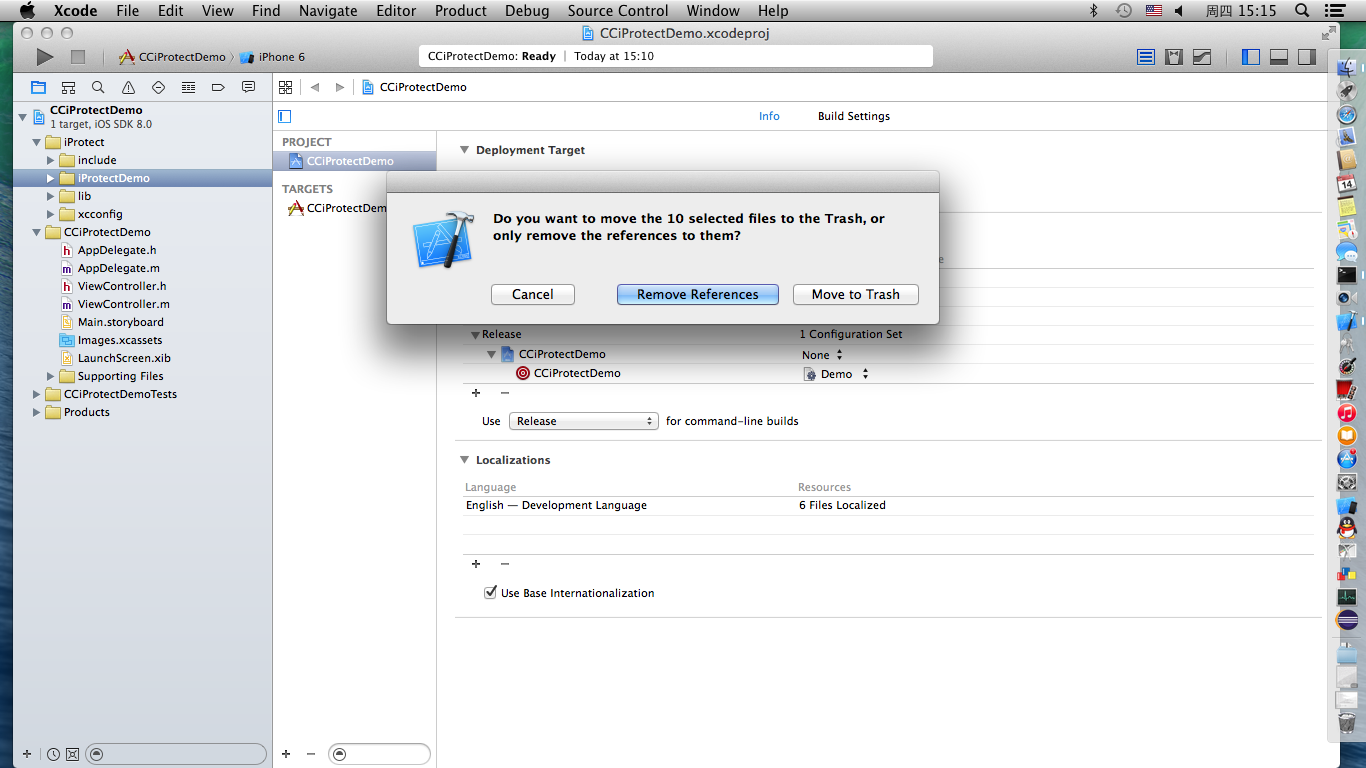
* 在弹出的对话框中选中iProtect目录（里面有头文件、静态库、XCConfig文件和iProtectDemo程序），确认钩选“Copy items if needed”复选按钮，钩选“Create groups”单选按钮，点击“Add”按钮完成iProtect软件的导入。



* 配置Configuration文件（XCConfig），在Xcode的Project Navigator窗口中选中CCiProtectDemo项目，右侧窗口选中CCiProtectDemo项目，将Configurations列表展开，所有使用了本软件的目标（Target）均需将XCConfig配置文件选中，如下图所示：



* 删除导入的iProtectDemo目录，如下图所示：



### 初始化系统

在项目的应用代理类中初始化系统：

#import "iProtect.h"

......

- (BOOL)application:(UIApplication\*)application didFinishLaunchingWithOptions:(NSDictionary\*)launchOptions

{

......

iProtectInitialize();

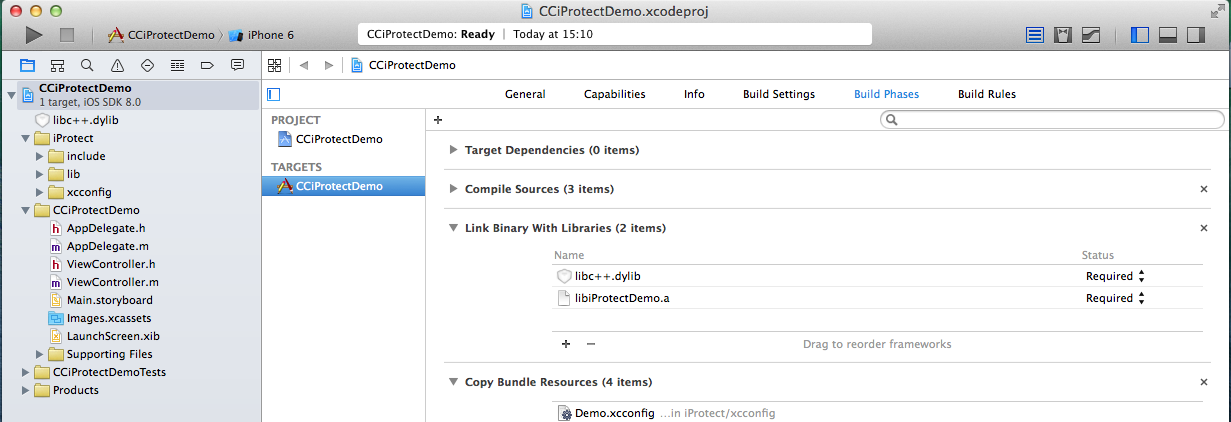
......

}

......

### 连接库

在Xcode的Project Navigator窗口中选中项目，在右侧TARGETS中选中目标（比如下图中CCiProtectDemo），选中Build Phases标签页，打开“Link Binary With Libraries”列表，点击“加号”按钮，在弹出的对话框中选中libc++.dylib项，点击“Add”按钮。在“Link Binary With Libraries”表中，确认有libiProtectXXXX.a和libc++.dylib两项。如下图所示：

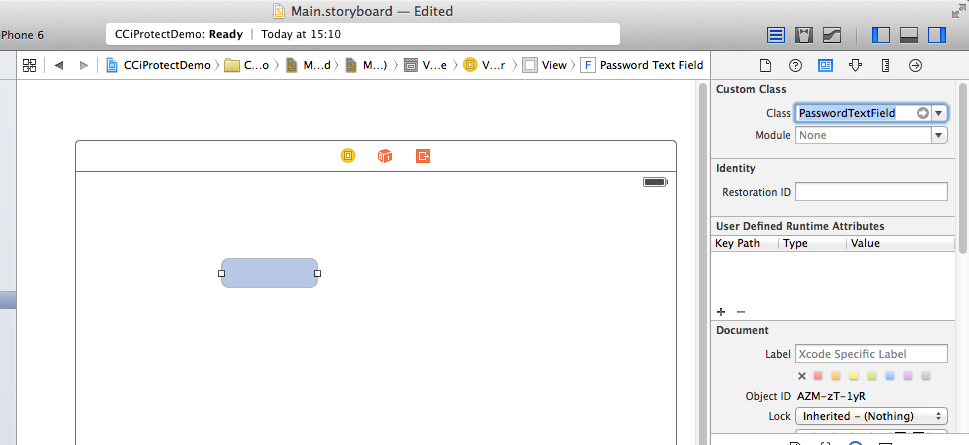


注意：上述XXXX为客户代码，具体请参考收到的软件包lib目录中静态库文件名。

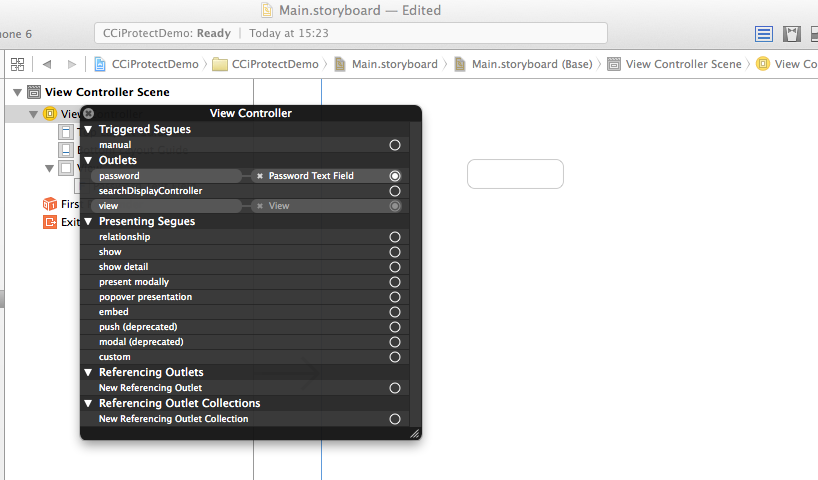
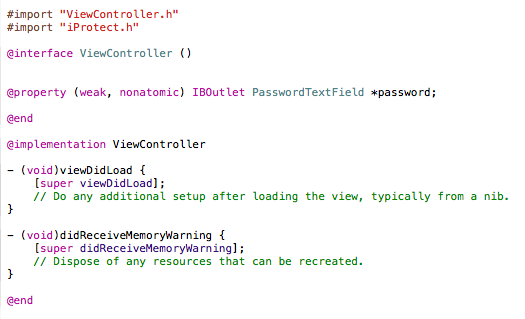
### 创建云核安全输入框

iProtect的核心对象PasswordTextField安全输入框的使用方法同iOS SDK中UITextField控件使用方法基本一致，可以通过代码创建，也可以通过Interface Builder创建。这里采用Interface Builder为例：

* 打开一个StoryBoard（或者XIB）文件，在界面上增加一个UITextField对象，把其类型改为PasswordTextField。如下图所示：



* 在此xib的所有者类中创建一个IBOutlet，类型为PasswordTextField，在xib中连接此IBOutlet到创建的PasswordTextField对象。如下图所示：



### 获取加密后的密码

在xib文件加载成功后，设置PasswordTextField属性。

调用PasswordTextField输入框的getValue:方法取得加密后的密码。

注：此处可以参考软件包中的iProtectDemo项目，用Xcode打开。

## 云核安全输入框接口说明

### 云核安全输入框属性

|  |  |
| --- | --- |
| 属性 | 说明 |
| NSString \*maskChar | 输入框中显示的字符掩码，默认为@"●" ，一般密码框为@"\*"、@"#"等字符掩码。 |
| uint32\_t encryptType | 加密类型。此属性通常通过预定义宏设置，这些宏来源于XCConfig文件，如未设定正确的XCConfig文件，则编译可能会失败。  注：   1. 某些客户存在不止一种加密算法，则会存在多个宏，根据业务需要选择此值。 2. 此属性必须设置正确，否则会出现多种可能错误，比如getValue取回空值（nil），或者服务端解密失败等等。 3. 针对具体宏的意义请参考软件包中XCConfig文件中注释。 |
| NSString \*accepts | 可以接受的字符（正则表达式），默认@"[:print:]+"。此属性用来供verify方法进行内容检测。 |
| uint32\_t contentType | 可以接受的内容类型，取值为V\_CONTENTTYPE\_ANY或者V\_CONTENTTYPE\_NUM、V\_CONTENTTYPE\_LETTER、V\_CONTENTTYPE\_PUNCT这三个变量的或。此属性用来供verify方法进行内容检测。 |
| BOOL passwordMode | 内容是否是密码，此选项和UITextField类的属性secureTextEntry区别是，此选项用于verify方法进行内容检测 |
| NSString \*dictionaryWords | 密码字典，用“,”分割的字符串。例如@"123456,abcdef,654321"列出了三个字典密码，供verify方法进行内容检测 |
| unsigned int maxLength | 允许最大长度，默认为12。 |
| unsigned int minLength | 允许最小长度，默认为6。 |
| BOOL kbdRandom | 设置SA安全键盘输入按键顺序是否随机，YES为随机，NO为不随机。默认YES。 |
| KeyboardType kbdType | 键盘类型：  \* 下面四种键盘类型为第一套键盘，支持横、竖屏，支持iPhone和iPad  KeyboardTypeLowerCaseLetter：小写字母键盘  KeyboardTypeCapitalLetter：大写字母键盘  KeyboardTypeNumber：数字键盘  KeyboardTypeSymbol：符号键盘  KeyboardTypePinNumber：用来输入纯数字的iPhone键盘，仅支持iPhone竖屏  \* 下面五种键盘类型为第二套键盘，仅支持iPhone竖屏  KeyboardType2LowerCaseLetter：小写字母键盘  KeyboardType2CapitalLetter：大写字母键盘  KeyboardType2NumberSymboll：数字＋16个字符  KeyboardType2NumberSymbol2：数字＋另外16个字符  KeyboardType2Number ：纯数字 |
| NSString \*applicationPlatformModulus | 应用平台模数，非必须设置。根据encryptType值来决定是否需要设置。 |
| NSString \*encryptionPlatformModulus | 加密平台模数，非必须设置。根据encryptType值来决定是否需要设置。 |
| NSString \*applicationPlatformPublicKey | 应用平台公钥，非必须设置。根据encryptType值来决定是否需要设置。 |
| NSString \*encryptionPlatformPublicKey | 加密平台公钥，非必须设置。根据encryptType值来决定是否需要设置。 |
| NSString \*applicationPlatformPublicKeyX  和  NSString \*applicationPlatformPublicKeyY | 应用平台SM2公钥坐标，非必须设置。根据encryptType值来决定是否需要设置。 |
| NSString \*encryptionPlatformPublicKeyX  和  NSString \*encryptionPlatformPublicKeyY | 加密平台SM2公钥坐标，非必须设置。根据encryptType值来决定是否需要设置。 |
| TouchKeyType touchKeyType | iPhone或者iPod touch上，软件盘点击时弹起的放大键效果行为，取值如下：  TouchKeyTypeAuto:越狱环境不弹键、非越狱弹键  TouchKeyTypeAlways：总是弹键  TouchKeyTypeNone：永不弹键 |
| WorkMode mode | 设置工作模式，通常有三种模式：  Product、Test和Development |
| BOOL keyboardHiddenOnOKKeyClicked | 设置点击完成后是否收起键盘，默认YES |
| FinishBlock finishBlock | 键盘输入完成后的回调block |
| BOOL highlightEffect | 仅适用于KeyboardTypePinNumber类型，设置键盘是否有高亮状态，默认YES |
| KeyboardSwitchMode switchMode | 仅适用于第二套类型键盘，设置键盘类型之间的切换模式  KeyboardSwitchModeLetterSymbol：字母与符号  KeyboardSwitchModeLetterNumber：字母与数字  KeyboardSwitchModeLetterSymbolNumber：字母、符号与数字 |
| BOOL showTipBar | 仅适用于第二套类型键盘，设置是否显示键盘上方的提示条，默认NO |
| 其他属性和UITextField属性一致 | |

### 云核安全输入框方法

|  |  |
| --- | --- |
| 方法声明 | 说明 |
| (NSString\*)getValue:(NSString\*)timestamp; | 获得加密后的输入内容，timestamp是时间戳，应该通过网络通信从服务器端取得。某些加密类型timestamp传nil。 |
| (short)verify; | 依据accepts属性、contentType属性、passwordMode属性和dictionaryWords属性验证用户输入信息是否合法。  返回值：  0：合法  -1：内容为空  -2：输入小于最小长度  -3：输入字符不可接受  -4：输入内容是简单密码  -5：输入内容是字典密码  -6：输入内容类型不符合  -10：未知错误  -10：初始化内容验证器错误 |
| -(void)clear; | 清除输入内容 |
| -(short)getLength; | 获得输入内容长度 |
| -(long)getVersion; | 获得版本 |
| -(char)getComplexDegree; | 获得输入内容强度 |
| -(NSString\*)lastError; | 取得最后一次错误描述 |

### 密码长度变化通知

1. 包含头文件：#import "iProtect.h"
2. 注册密码长度变化通知处理函数，假设处理密码更新的方法是:

- (void)doPasswordChanging:(NSNotification\*)noti;

则可以通过以下语句注册密码长度变化通知:

[[NSNotificationCenter defaultCenter] addObserver:self selector:@selector(doPasswordChanging:) name:PasswordChangedNotification object:nil];

1. 取得密码旧长度、当前长度和密码长度发生变化的密码输入框实例

通过NSNotification对象的userInfo字典可以取得密码的旧长度、当前长度和密码长度发生变化的密码输入框实例，其key值如下：

|  |  |  |
| --- | --- | --- |
| 项 | 项的Key值 | 项的类型 |
| 旧密码长度 | kPasswordOldLength | 整型 |
| 当前密码长度 | kPasswordCurrentLength | 整型 |
| 发生变化的密码框实例 | kPasswordTextField | PasswordTextField |

### 银联算法说明

银联在线和银联商务两种加密类型需要使用银联ANSI X9.8转换算法，特提供如下特殊接口，分两种用法：

1. 使用PasswordTextField来输入密码，此时，调用接口：

-(NSString\*)getValue:(NSString\*)timestamp withPan:(NSString\*)pan;

* 参数：timestamp是时间戳，通过网络通信从服务器端取得。
* pan为主帐号，如果传nil,则为不带主帐号的X9.8转化，否则为带X9.8转化。
* 返回值：密文

1. 使用科蓝公司VX框架开发，此时，直接使用PasswordKeyboardDelegate类：

假设PasswodKeyboardDelegate类的实例是pkDelegate，则设置timestamp属性如下：

pkDelegate.timestamp = ts；

* ts：时间戳，通过网络通信从服务器端取得。

再调用其接口获得秘文：

-(NSString\*)getPasswordWithPan:(NSString\*)pan;

* 参数：pan为主帐号，如果传nil,则为不带主帐号的X9.8转化，否则为带X9.8转化。
* 返回值：密文

## 密钥管理体系

### 客户端密钥管理体系

安全输入系统产生密文时只进行公钥操作（加密运算），对公钥无需进行保护。

安全输入系统在内部保存敏感信息时采用了自定义算法，其密钥长度达到128位，且分散保存，具有足够的安全性。

### 服务端密钥管理体系

服务端解密程序，支持更新密钥（私钥）操作，具体密钥管理体系由ISV或者APP最终客户自行设计。

## 版本控制

安全输入系统版本控制采用了成熟的SVN版本控制系统，软件被分为核心算法和客户化两部分。不同工程师分配给了不同权限，核心算法由最少的具有密码产品开发经验的工程师进行开发，客户化部分完成界面定制，由普通iOS工程师进行维护。

PasswordTextField类的getVersion方法返回当前版本（核心算法版本），我司将通过安全的通信方式将此版本对应的特性（或者变更记录）交给客户。

# Web与原生混合模型下安全输入系统的集成

## 说明

因不同ISV开发苹果应用时所采用的混合模型应用框架不一致，故在这种模型下安全输入系统的集成工作也是不固定的。但是总的步骤是一样的，本手册会说明此集成步骤，集成前需要注意以下几点：

1. 此集成工作，要求ISV完成一些开发工作。

Web界面上一个输入域对应一个WebPasswordKeyboard类的实例，此对应关系由ISV自行维护。

1. WebPasswordKeyboard类的属性及方法基本同3.2节。

## WebPasswordKeyboard类的初始化

创建此类的实例后，可以按照3.2节设置此类的属性。最关键的两个属性是webView和jsCallbackDelegate，见demo中MainViewController类中webViewDidFinishLoad:方法的实现。

## 控件键盘的显示

捕获div的click事件，在此事件中调用WebPasswordKeyboard类的showKeyboard方法。在demo中，是在index.js文件中app.bindEvents方法中绑定click事件的。对应的native端的代码是iProtecty类中方法：

- (void)showKeyboard:(CDVInvokedUrlCommand\*)command；

键盘显示以后，安全输入系统会调用JSCallbackProtocol中方法：

- (void)keyboardHeight:(float)height

withWebPasswordKeyboard:(WebPasswordKeyboard \*)keyboard；

在此方法中，安全输入系统通知ISV，当前键盘的高度，ISV可根据此值适当移动webView的位置，以防输入框被遮挡。

## 控制键盘的隐藏

点击键盘上的完成按钮，隐藏键盘。

键盘隐藏以后，安全输入系统会调用JSCallbackProtocol中方法：

- (void)keyboardHeight:(float)height

withWebPasswordKeyboard:(WebPasswordKeyboard \*)keyboard；

在此方法中，安全输入系统通知ISV，当前键盘的高度为0，ISV可根据此值将移动过的webView恢复原来的位置。

## 控制输入域内容

ISV框架中必须实现JSCallbackProtocol，并将此实现的实例传递给WebPasswordKeyboard类的属性jsCallbackDelegate（见4.2.1）。

此协议中方法决定了如何修改Web输入域中内容，见demo中MainViewController类中以下方法的实现：

- (void)passwordLength:(NSUInteger)length

withWebPasswordKeyboard:(WebPasswordKeyboard\*)keyboard;

## 提交表单

捕获提交按钮的click事件，在此事件中取时间戳，取到时间戳后，就可以拿到密文，将此密文放回表单中的隐藏输入域，就可以正式提交表单了。

Demo中，此事件的捕获见index.js中app.bindEvents方法。

代码流程是index.js中onSubmit ---->iProtect类的getValue方法---->MainViewController类的getTimestamp方法---->MainViewController类的onTimestamp方法---->回到index.js文件中onRealSubmit，此时已可以正式提交表单了。