



StrikeForce Technologies Inc.
GuardedID Mobile SDK SDK
iOS Version 1.0

Corporate Headquarters

StrikeForce Technologies, Inc.
1090 King Georges Post Road #108
Edison, NJ 08837, USA
<http://www.sftnj.com>
Tel: 732 661-9641
Fax: 732 661-9647

FORWARD

The information contained herein is proprietary to StrikeForce Technologies, Inc. It may not be used, reproduced or disclosed to others except as specifically permitted in writing by StrikeForce Technologies, Inc. The recipient of this document, by its retention and use agrees to protect the information contained.

The specifications and information regarding the products in this manual are subject to change without notice. All statements, information, and recommendations in this manual are believed to be accurate but are presented without warranty of any kind, express or implied. Users must take full responsibility for their application of any products.

Table of Contents

Overview	4
Requirements	4
PhoneGap Installation	4
iOS keyboard framework integration	4

Overview

The GuardedID Mobile SDK SDK is a simple set of functions that allows an iOS application to take advantage of StrikeForce Technologies' keyboard encryption mechanism. Once installed, GuardedID Mobile SDK protects the path of sensitive typed data from keyloggers. The SDK allows the implementation of a secure keypad that overlays the native keypad and encrypts the keystrokes entered into the secure keypad. The SDK currently requires PhoneGap, across platform mobile development development software from Adobe.

Requirements

- Intel based computer running MAC OS X version 10.6 or later
- Xcode (version 4.3.2 at the time of writing)
- iOS SDK 4.3 or 5.1
- PhoneGap 1.4.1

PhoneGap Installation

Adobe® PhoneGap is a software library that bridges the gap between native mobile applications and web applications. Using PhoneGap, one can develop a web-based application using standard HTML5, JavaScript and CSS3, and then package that application in the native format expected by many different mobile devices. PhoneGap also offers hooks so that device specific functionality can be utilized by the web-based application.

Go to the <http://phonegap.com/install/> and find Phonegap 1.4.1 release. Download zip file for PhoneGap 1.4.1. Clicking on this will prompt the download of a ZIP file containing everything you need to get started.

Unzipping the file will give you numerous folders, each labeled with the targeted OS. As we are installing for iOS, open up the iOS folder and mount the dmg file. The dmg contains a pkg file. Click this and go through the installation process. After completing the install, you can go ahead and start up Xcode.

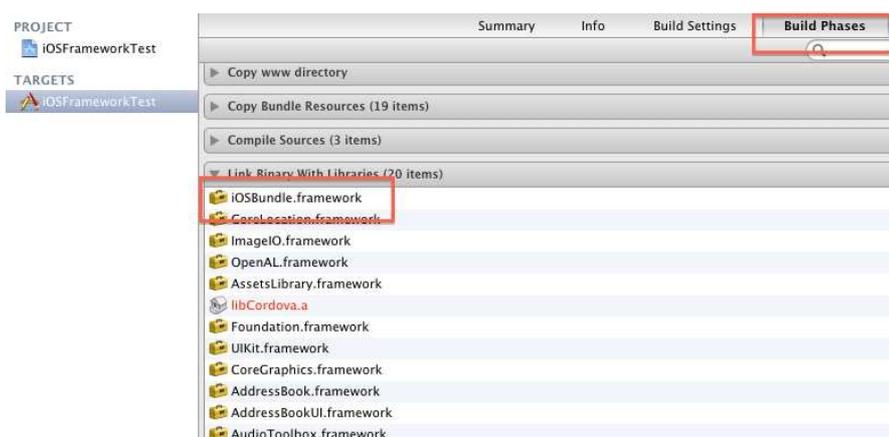
iOS keyboard framework integration

This is the framework used with Phonegap application. This will overlay native keyboard with secure keyboard in iphone phonegap application.

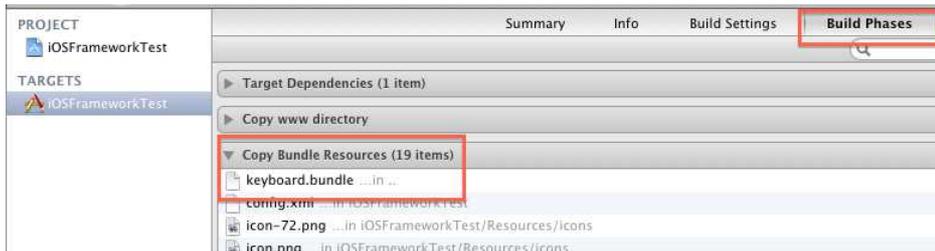


Here are the basic steps to integrate “Secure Keyboard” into your iPhone application.

- You have two files available
 - o iOSBundle.framework
 - o keyboard.bundle
- Add “iOSBundle.framework” to Link Binary with Libraries tab in application Target to Build Phases.



- Add “keyboard.bundle” to Copy Bundle Resources in application Target to Build Phases.



- Now, In AppDelegate.h file, import header “import <iOSBundle/Utility.h>”
- And create object for Utility class like “Utility *uti;”

```
#import <UIKit/UIKit.h>
#import <Cordova/CDViewController.h>
#import <iOSBundle/Utility.h>
@interface AppDelegate : NSObject <UIApplicationDelegate>{
    Utility *uti;
}
```

- In AppDelegate.m file find method “didFinishLaunchingWithOptions”. In that method after “[self.window makeKeyAndVisible];” place below code.

```
self.window.rootViewController = self.viewController;
[self.window makeKeyAndVisible];

uti = [[Utility alloc] init];
[uti myloadView];

if ([[UIDevice currentDevice] systemVersion] floatValue) >= 3.2) {
    [[NSNotificationCenter defaultCenter] addObserver:self
                                             selector:@selector(keyboardDidShow:)
                                             name:UIKeyboardDidShowNotification
                                             object:nil];

    [[NSNotificationCenter defaultCenter] addObserver:self
                                             selector:@selector(keyboardDidHide:)
                                             name:UIKeyboardDidHideNotification
                                             object:nil];
} else {
    [[NSNotificationCenter defaultCenter] addObserver:self
                                             selector:@selector(keyboardWillShow:)
                                             name:UIKeyboardWillShowNotification
                                             object:nil];

    [[NSNotificationCenter defaultCenter] addObserver:self
                                             selector:@selector(keyboardWillHide:)
                                             name:UIKeyboardWillHideNotification
                                             object:nil];
}
return YES;
```

- Add functions for “keyboardDidShow” and “keyboardWillShow” method in AppDelegate.m file.
- Add code to bind/draw custom keyboard.

```
- (void)keyboardDidShow:(NSNotification *)note {  
    NSDictionary *info = [note userInfo];  
    CGSize kbSize = [[info objectForKey:UIKeyboardFrameBeginUserInfoKey] CGRectValue].size;  
  
    uti.kbheight = kbSize.height;  
  
    uti.tempWindow = [[[UIApplication sharedApplication] windows] objectAtIndex:1];  
  
    [uti addButtonToKeyboard];  
  
}
```

- The result will be as follows:

