Authentic Vision SDK for iOS

Version 8.2.0

December, 2023

The Authentic Vision Mobile SDK (AV SDK) authenticates Authentic Vision's irreproducible security labels in iOS mobile applications. It provides a simple API and ready-made scan view controller to scan AV security labels. Your application invokes the AV SDK, which displays a scanner interface. Once a scan has completed, the AV SDK returns scan results to your application for further processing.

This documentation covers integration of the Authentic Vision mobile SDK into iOS applications with Xcode.

Requirements

The use of the Authentic Vision SDK has the following basic requirements:

1. The iOS device must have a working rear camera with torch. All iPhones and some iPads qualify.
2. The target iOS version must be 12.0 or later
3. Active internet connection during the scan (up to 1 MiB of data is transferred per scan)
4. You must have an AV SDK license and API key

Downloading and Testing the SDK

The AV SDK is available as ZIP file at github.com/authenticvision/mobile-sdk-ios. The ZIP file may be extracted using macOS built-in tools.

The extracted ZIP file contains a sample project and `AuthenticVisionSDK.xcframework`. To quickly try out the SDK please open the sample project, insert your API key into `DemoTableViewController.swift`, and run the sample on an attached iOS device.

Integration with Xcode

The Authentic Vision SDK may be integrated into Xcode projects as follows:

1. Open `Documentation/AuthenticVisionSDK.doccarchive` once to import the API documentation.
2. Paste `Frameworks/AuthenticVisionSDK.xcframework` into your Xcode project's folder on disk. It is common to use a subfolder named `Frameworks` for storing vendor frameworks.
3. Open your iOS target's *General* settings.
4. Navigate to the *Frameworks, Libraries and Embedded Content* list and click the "+" button.
5. Select "Add other…", "Add files…"
6. Use the file picker to select `AuthenticVisionSDK.xcframework` from step 2.
7. Confirm that Xcode registered the framework under your project tree, and that its embedding setting is set to *Embed & Sign*.

You may now import the Authentic Vision anywhere in your Swift or Objective-C application via `import AuthenticVisionSDK`.

Scan a Label

The following assumes that you are using Apple UIKit as primary framework. Other frameworks such as Flutter, Cordova, or NativeScript may be covered at docs.authenticvision.com/sdk.

At its most basic, an application using the AV SDK must:

1. Create an `AVKScanConfig` object or change the global default scan config.
2. Create an `AVKScanViewController`, assign a delegate, and present the scan view controller.
3. Implement the `AVKScanViewControllerDelegate` protocol to receive scan results/errors.
4. Dismiss the scan view controller when an acceptable scan result or error is received.

This functionality is implemented in the UIKit Swift AV SDK sample, along with visualization of the entire SDK result, and an example for changing the AV SDK branding. The following is a minimal excerpt:

```
import AuthenticVisionSDK

AVKScanConfig.default().apiKey = "avck_fooBarBaz"

class ScanViewController: UIViewController, AVKScanViewControllerDelegate {
    func startScan() {
        do {
            let controller = try AVKScanViewController(delegate: self)
            controller.addCloseButton(withTarget: self, action: #selector(stopScan))
            present(controller, animated: true)
        } catch let error {
            displayErrorToUser(error)
        }
    }

    @objc func stopScan(_ sender: UIButton) {
        dismiss(animated: true)
    }

    func scanViewController(_ controller: AVKScanViewController,
                            unrecoverableError error: Error) {
        dismiss(animated: true) {
            displayErrorToUser(error)
        }
    }

    func scanViewController(_ controller: AVKScanViewController,
                            scanDidCompleteWith result: AVKScanResult) {
        dismiss(animated: true) {
            if result.isAuthentic {
                // - Display an "authentic product" message and product information.
                // - The URL to product information, optionally with a SIPv4 token, is
                //   available in result.campaignURL.
            } else {
                // - The label is not authentic, but not necessarily counterfeit.
                // - Display instructions to the user to scan again.
                // - Optionally evaluate result.authResult to check for counterfeits.
            }
        }
    }
}
```

Advanced Use Cases

API and Class Documentation

An Xcode docs archive is provided along with the AV SDK. Import it or view it with Xcode for a complete in-IDE reference. The entire AV SDK's API is also covered in the sample application.

Replace Colors, Logos, and Fonts

The AV SDK allows customization of the SDK to your own brand's style. You may either assign a bundle containing an asset library with named entries of the following table to the scan config's `brandingBundles`, or implement `AVKBrandingDelegate` to override branding resources. The protocol implementation must be assigned to the scan config's `brandingDelegate`.

Type	Asset Name	Description
Color	`UniversalPrimary`	Accent color for UI elements
Color	`UniversalSecondary`	Background color with sufficient contrast to `UniversalPrimary`
Color	`BackgroundScanProgress`	For non-default UX only: A color similar to `UniversalPrimary`, which is blended with a blurred camera preview
Image	`ScanLogo`	Image to be displayed at the top center of the scan screen
Image	`ScanCloseButton`	Image used by `addCloseButtonWithTarget:action:` of `AVKScanViewController`
Data	`ScanDoneSound`	Sound for successful scans when audio feedback is enabled

A standard font family for the AV SDK may be set via e.g. `AVKScanConfig.default().fontFamily = "American Typewriter"`. Custom font selection logic may be implemented via `AVKBrandingDelegate`.

Localization

The Authentic Vision SDK has been translated to the following languages: Arabic, Chinese (Simplified), Chinese (Traditional), English, German, French, Italian, Japanese, and Spanish. Please reach out to your contact at Authentic Vision if additional localization is required.

The app environment's language is automatically adopted by the AV SDK. It may be overridden regardless of iOS settings via the `locale` property of `AVKScanConfig`.

Server-Sided Authentication

Documentation for server-sided validation of scan results is provided at docs.authenticvision.com/sdk. The AV SDK provides an *attestation token* through `AVKScanResult` for that purpose, either as separate property or as part of the campaign URL. Some attestation modes require configuration via `AVKScanConfig`.

Labels require special configuration for high-security use cases. Please discuss your specific use case with your contact at Authentic Vision if you wish to use AV labels as a key for a service.

Version Information

AV SDK version information may be retrieved as follows. Accessing version information allows you to enhance your own audit trail, or simply display versions of all components in your application's about section.

```
let version = AVKVersionInfo.sdkVersion
let commit = AVKVersionInfo.sdkGitHash
print("Authentic Vision SDK \(version) (\(commit))")
```

Legal

The AV SDK is made available to you by Authentic Vision GmbH, Ludwig-Bieringer-Platz 1, 5071 Salzburg, Austria. It is proprietary software and licensed individually.

The AV SDK additionally contains third-party open source software and content. A detailed list of open source software is available in `foss.md`, provided along with this documentation file.