StrongKey Android Client Library 1.0

Preview 1 - Release Notes

1. Introduction

The StrongKey Android Client Library (SACL) is an open-source, native Android library providing support for the FIDO2 protocol. It provides the following features:

- It is supported on Android 9 (API 28) “Pie” or greater;
- It supports the Java programming language, and does <u>not</u> require the use of JavaScript or the WebView component to deliver FIDO capability. Note that it does <u>not</u> support the use of external *Security Keys* – only platform keys;
- It uses the *AndroidKeystore* - taking advantage of the *Trusted Execution Environment (TEE)* or a *Secure Element (SE)*, whichever is present - for key-generation, storage and usage. It is always used as a *user verifying platform authenticator (UVPA)*. Devices without the TEE or SE cannot install apps using the SACL;
- It supports *registration, authentication* and *transaction authorization* using “dynamic linking” - a core requirement of the European Union’s Payment Services Directive 2 regulation for *Strong Customer Authentication (SCA)*;
- It supports Android's *BiometricPrompt* API for verifying users before enabling use of the FIDO key;
- It has out-of-the-box integration with the open-source FIDO® Certified StrongKey FIDO Server (SKFS) – just add your mobile app to the flow;
- It includes a sample e-Commerce web-application – the *Sample FIDO App for e-Commerce (SFAECO)* – to demonstrate 4 basic functions:
    - User enrollment
    - FIDO registration
    - FIDO authentication and
    - User confirmation of business transactions with the user’s registered FIDO key
- The server side components of the SFAECO app are available as a Java Enterprise Edition (JEE) application to support the mobile sample app. This JEE application makes webservice requests of the SKFS;
- It includes a sample browser based web-application – Back Office Application (BOA) - to work in concert with the SFAECO app to perform sample back-office business functions. But, the primary purpose is to demonstrate the use of FIDO for strong authentication and to review business transactions performed by app users, as well as see data collected by the app when performing *transaction confirmation (TXC)*;
- It includes a second sample browser based web-application - FIDO Key Management - to demonstrate the newly announced *single sign-on (SSO)* capability of the SKFS with *JSON Web Token (JWT)* using x509 based *JSON Web Signatures (JWS)*;
- These 3 web applications have been installed on a demo server on the internet (https://psd2demo.strongkey.com) against which the mobile app makes REST webservice calls;

The SACL has been tested with an Essential PH-1, Google Pixel 3a and Google Pixel 4a phones – the first two running Android 9 (Pie) with API 28, and the Pixel 4a with Android R (API 30). The device must have a fingerprint enrolled to support the use of the SACL. While it is likely to work on most Android devices with biometric capability, your mileage may vary.

NOTE: If it works with a phone you have, we would love to hear from you with the phone's brand, model and whether it is using the TEE or an SE; thank you.

This is a preview release where some things may be a little rough at the edges. However, it provides insight into the capability the completed product will have, and enables early adopters to design and code native Android business application(s) to work with the SACL. StrongKey's goal is to deliver a production quality release by the end of June 2021.

Feedback on the SACL may be sent to getsecure@strongkey.com. Or, post them on the forum of the repository (SourceForge and Github) where you downloaded the distribution. Thank you for using the SACL.

2. Notes and Known Issues

#	Issue
SACL-0001	In this Preview Release 1 (PR1), the SACL is designed to allow multiple credentials to be registered from the SFAECO app, to the same Relying Party ID (RPID). Normally, this would not be allowed, since an Authenticator designed for use by a single user, must only have one credential registered to an RPID active at any time. However, this can be very cumbersome when designing your business app and testing it with one mobile device. As such, PR1 does not enforce uniqueness of FIDO keys to a specific RPID within the device. When StrongKey finalizes a Production release later this year, the enforcement will become default.
SACL-0002	On the Pixel 3a and 4a, WiFi services will not automatically turn on if Location Services are turned off, resulting in error messages when attempting any function requiring access to the SFAECO services. Please check for these messages in Logcat if they don't show up in a Toast message. **Workaround**: If your phone is configured to work with your WiFi router, but fails to connect it could be for a variety of reasons: Location Services is off; Data Saver is on; Advanced Network settings permit the phone to turn off network services when the device goes to sleep, etc. Use the browser on your phone to visit any website. This triggers network services to start. Once started, the SFAECO app will work fine. However, if the device goes to sleep even for a few seconds, it is likely to turn networks services off. You may have to repeat the process to get back on the WiFi network – or turn Location Services on to keep the service on.
SACL-0003	When a user enrolls, and if the device goes to sleep, the SACL will sometimes not persist the *counter* value for the Authenticator. This will

	result in error messages or crashes because the SKFS’ default security policy is to require that Authenticator counter values always be incremented. Logcat messages will show exceptions with an HTTP 500 error. **Workaround**: Should this occur, Force Stop the app in your App Settings. Make sure that your network is working (by using a browser to visit some website), and try it again. You should notice that the *counter* value should be greater than “1” in the *Registered Key* page.
SACL-0004	If a FIDO registration or authentication operation fails in the app, this is most likely due to network timeouts; but a secondary reason is that it is more than 5 minutes since the user unlocked the device. The SACL has a default value of 5 minutes for enabling use of the *AndroidKeystore* after the user has successfully unlocked the mobile device using their PIN, Pattern or fingerprint. This is designed to simulate the Regulatory Technical Specification (RTS) of the EU PSD2 regulation. As a result, no biometric prompt will show up during FIDO key-generation or authentication with the FIDO key. However, a biometric prompt will always show up when the user is authorizing a payment transaction, displaying the “dynamic link” with appropriate payment information. The app was deliberately designed to show different modes in which a FIDO operation can work depending on how the app chooses to use the SACL – either by relying upon the default timeout for the use of AndroidKeystore on an unlocked device, or explicitly prompting for a user’s fingerprint when confirming a transaction. **Workaround**: Lock the phone by pressing the Power button, unlock it with PIN, Patter or fingerprint and restart the app to perform the necessary operation. It will succeed this time.
SACL-0005	The sample app was written to use Android’s Biometric API 1.0.1. However, when you load the app and SACL, Android Studio will prompt you to update your Biometric library to 1.1.0. Do not do that, as the code will not compile. **Workaround**: Keep the Biometric API library at 1.0.1 for now; StrongKey will update its code to use the 1.1.0 library and provide an update in the near future.

SACL-0006	When attempting to build the app, Android Studio will complain: “"Unable to find a matching configuration of project :sacl:”. This is because you’ve most likely extracted the distribution into a folder that does not match the settings of the “sacl” module in the **settings.gradle** file.
	Workaround: Open the **settings.gradle** file for the SampleSACLFidoEcoApp module, and change the value of the **projectDir** variable to reflect where the “sacl” module can be found on your computer. For example, if the location originally showed:
	`“/usr/local/workspace/android/StrongKeyAndroidClientLibrary/sacl”`
	and you now have the SACL in
	`“/Users/jdoe/android/StrongKeyAndroidClientLibrary/sacl”`

	make the modifications and rebuild the app.
SACL-0007	The AAGUID (Authenticator Attestation Globally Unique Identifier) used by the Preview release of SACL is: "CAFEBABECAFEBEEF0123456789ABCDEF". When it goes into Generally Available (GA) status as a production quality release, the AAGUID will become: "5341434C323032304b4F52D999D03ECB"
SACL-0008	Testing multiple apps that use the SACL on the same mobile device has not been tested currently. So, the behavior of this use-case is, currently, unknown. However, we anticipate testing this capability in the next update and providing guidance.

The SACL can be used by app developers for all types of apps in finance, healthcare, education, government, gaming, enterprise apps, etc. While mobile devices have made it significantly easier for users to authenticate to the apps' sites (by using biometrics), behind the curtain, they unfortunately still use the ancient password based authentication scheme that are susceptible to attack and are responsible for more than 95% of all data breaches.

With the SACL and the SKFS, Android apps can now forever leave passwords behind. We hope you find this opportunity exciting to protect your users, as well as your own sites by eliminating the awful passwords that plague our lives.

Drop us a note at getsecure@strongkey.com to tell us what you think.

Have fun!