Introduction to Software Authentication

Release R3

Developer

Contents

1. Introduction 3

1.1. Overview 3

1.2. Authentication Options 3

1.3. Requirements, recommendations, and permissions 4

1.4. Terminology 4

2. Requirements 5

2.1. Software Authentication Server Requirements 5

2.2 Accessory Requirements 5

3. How Software Authentication Works 6

3.1. Token Requests 6

3.2. Provisioning 7

3.2.1 Factory Provisioning 7

3.2.2 In-Field Provisioning 7

3.2.2.1 HomeKit In-Field Provisioning Requirements 7

3.2.2.2 AirPlay audio In-Field Provisioning Requirements 8

3.3. Registration 8

3.4. Activation 8

4. Software Authentication Management 9

5. Revision History 10

1. Introduction

1.1. Overview

Authentication provides a mechanism for Apple devices to trust the identities and feature sets of third-party accessories. Authentication helps ensure that accessories have completed certification requirements for the specific technologies that they support. MFi-licensed technologies support hardware- or software-based authentication. Accessories must implement only one support authentication mechanism.

The following MFi-licensed technologies support token-based software authentication:

- AirPlay audio
- Find My network
- HomeKit

1.2. Authentication Options

Hardware-based authentication incorporates the Apple authentication co-processor. This component must be procured from an MFi Authorized Distributor, and must be integrated into the accessory. However, it's a one-time implementation and does not require ongoing maintenance. Technical details about this co-processor are provided in the *Accessory Interface Specification*, available in the MFi Portal Content Center.

Token-based software authentication involves setting up a server to request software tokens from Apple and provisioning the tokens onto accessories. Software token authentication requires server setup and development of a mechanism to add the tokens onto an accessory.

1.3. Requirements, recommendations, and permissions

This document contains statements that are incorporated by reference into legal agreements between Apple and its Licensees. The use of the words *must, must not, required, shall, shall not, should, should not, recommended, not recommended, may, optional,* and *deprecated* in a statement have the following meanings:

- *Must, shall,* or *required* means the statement is an absolute requirement.
- *Must not*, *shall not*, or *prohibited* means the statement is an absolute prohibition.
- *Should* or *recommended* means the full implications must be understood before choosing a different course.
- *Should not* or *not recommended* means the full implications must be understood before choosing this course.
- *May* or *optional* means the statement is truly optional, and its presence or absence cannot be assumed.
- *deprecated* means the statement is provided for historical purposes only and is equivalent to 'shall not'.

The absence of requirements, recommendations, or permissions for a specific accessory design in this document must not be interpreted as implied approval of that design. Licensees should ask Apple for feedback on accessory designs that are not explicitly mentioned in this document.

1.4. Terminology

Throughout this document, these terms have specific meanings:

- The term *Apple device* refers to an iPhone, iPad, iPod, or Mac (running iOS, iPadOS, or macOS).
- The term *accessory* refers to any third-party product intended to interface with a device.
- The term *Service Provider* refers to the MFi Licensee who will download software authentication tokens using their server.
- The term *Product Plan ID (PPID)* refers to the unique identifier which is generated by the MFi Portal for each approved Product Plan.
- The term *Token* refers to software authentication entities that are downloaded by the MFi Licensee and included in Licensed Products.

2. Requirements

Both the software authentication server and the accessory that supports software authentication must comply with the requirements outlined in relevant MFi-licensed technology specifications.

2.1. Software Authentication Server Requirements

The server must be hosted either on a secure cloud or on a secure physical server, not a personal computer. Server setup requires submitting a Certificate Signing Request (CSR) to Apple as outlined in the server specification. Apple will generate a server certificate, which must be integrated in the server. This is the mechanism for Apple to securely validate that the server is authorized to request software authentication tokens.

Once the server setup is complete, the Service Provider can request software authentication tokens via the appropriate server APIs for product plans that they are linked to.

Detailed server requirements are outlined in the *Software Token Authentication Server Specification.*

2.2 Accessory Requirements

The accessory requirements may vary depending on the technology type. Please review the applicable specification located in the Content Center of the MFi Portal.

- **HomeKit accessory:** *HomeKit Accessory Protocol Specification*
- **AirPlay audio accessory:** *AirPlay Audio SDK Integration guide*
- **Find My network accessory:** *Find My Network Accessory Specification*

3. How Software Authentication Works

3.1. Token Requests

Software authentication tokens can only be successfully requested if the following conditions are met:

- The approved Product Plan supports software authentication
- The Product Plan lists the company requesting the software authentication token as a Service Provider
- The amount of tokens requested were already allocated by Apple and available to download for the specific Product Plan
- The Service Provider has completed the software authentication server setup

The licensee can request tokens that have already been allocated via the available software authentication server API. During this process, the licensee's software authentication server contacts the Apple server with a token request for a specific PPID. In response to this request, the Apple server provides tokens that can be provisioned onto accessories associated with that PPID.

Once the Product Plan is approved and is in the "Testing" phase, Apple allocates a default amount of 1,000 tokens for testing purposes. After the Product Plan completes certification and is in the "Production" phase, Apple allocates an additional 1,000,000 tokens by default.

If the Licensee has exhausted the allocated amount of tokens, they can request additional Tokens using the process below:

Navigate to the "Software Authentication Management" page in the "Resources" tab of the MFi Portal

- On this page, find the appropriate PPID and click on "Request entity"
- On the pop-up page, enter the additional amount of Tokens in the "Requested quantity" and add reason for the request in the "Comments" section
- Once completed, click on the "Request" button

The request will be sent to the MFi Licensee's account manager for approval. Upon approval, the new amount of Tokens will be allocated in the MFi Portal, and an email confirmation will be sent to MFi contact who requested the additional Tokens.

3.2. Provisioning

Once the tokens are allocated, they must be provisioned for each individual accessory. Provisioning the tokens onto an accessory can be done through either factory provisioning or in-field provisioning.

3.2.1 Factory Provisioning

Accessories that contain HomeKit, AirPlay audio, or Find My network out of the factory must provision the token in the factory at the time of manufacturing.

3.2.2 In-Field Provisioning

For accessories that have previously been manufactured or sold, in-field provisioning may be used to upgrade them to become HomeKit or AirPlay audio enabled through a firmware update. For IP-based accessories, the accessory can request a token either directly from the server or through the accessory's app.

3.2.2.1 HomeKit In-Field Provisioning Requirements

Accessories that use in-field provisioning and do not contain the HomeKit setup code must be accompanied by an iOS application to enable provisioning. The app must include the performAccessorySetupUsingRequest API to retrieve a setup payload.

A signed entitlement agreement with the Apple Developer Program is required to use the API. After your Product Plan is approved, you will receive an email requesting the information required for the entitlement agreement. This information includes your company's Developer Program Team ID, physical address, and contact details for an authorized signatory. Once the agreement has been signed and returned, the entitlement will be available in your company's Apple Developer Program account, which is accessible on the Apple Developer website.

Note: The API is only available for iOS 15.4+ and iPadOS 15.4+. Provide appropriate error messaging if the user attempts to initiate setup on an older version of iOS.

Third-Party App HomeKit Provisioning User Experience

Initiate HomeKit Setup

1. The user downloads and opens the accessory's app from the App Store. The iOS device must be joined to the network and connected to the Internet.
2. The app ensures the accessory's firmware supports HomeKit and updates the firmware if necessary.
3. The app requests permission from the user to access the home configuration database. The app must provide a method (such as a button) to allow the user to initiate HomeKit setup.

Retrieve Setup Payload

1. The setup payload is either retrieved from your server and written to the accessory by the app, or retrieved directly from the accessory and passed through using the HomeKit Setup Payload API.
2. The user is presented with a UI overlay provided by HomeKit during which the accessory is added to the user's home and configured.

Complete HomeKit Setup

1. The UI overlay allows the user to add the accessory to his or her home, assign it to a room, and add it as a favorite device.
2. The UI overlay is dismissed and the user returns to the accessory's app UI.
3. The app indicates to the user that the accessory has completed HomeKit setup.

3.2.2.2 AirPlay audio In-Field Provisioning Requirements

Accessories that contain AirPlay audio technology and use software authentication must support a mechanism to provision the accessory in the field. This is generally done through an application

3.3. Registration

Once a token has been provisioned onto an accessory, the Licensee must register use of the token with Apple. Additional details on how to register a token are available in the *Software Token Authentication Server Specification*.

3.4. Activation

Activation of a provisioned token happens at the time of accessory setup which is initiated by the user. A token can be used only once to activate an accessory. To reactivate an accessory (e.g., due to factory reset), a new token will be issued by the iOS device that must be provisioned on the accessory and used for the next activation.

4. Software Authentication Management

The Product Plan Owner and Service Provider can view the details about software authentication tokens that Apple has allocated for a particular product plan they are linked to in the “Software Authentication Management” page of the “Resources” section of the MFi Portal.

The following information is available to the Product Plan Owner and Service Provider for each PPID:

- **Product Plan ID:** Unique ID assigned to the accessory product plan in the MFi Portal
- **Accessory Name:** Name of the accessory as listed in the product plan
- **Status:** Status of the product plan
- **Allocated:** Amount of tokens that are allocated by Apple for the specific product plan
- **Available:** Amount of tokens that the service provider can download
- **Vended (Total):** Total amount of tokens that the service provider has downloaded
- **Vended (Not Registered):** Amount of tokens that the service provider has downloaded but not yet registered via the server API
- **Vended (Registered):** Amount of tokens that the service provider has downloaded and registered via the server API
- **Destroyed:** Amount of tokens that the service provider has downloaded and destroyed via the server API
- **Revoked:** Amount of tokens that the service provider has downloaded and Apple has revoked
- **Activated:** Amount of tokens that the service provider has downloaded and activated via the server API
- **First Activated On:** Date and time the first token was activated for the product plan
- **Request entity:** Link for the MFi Licensee to request additional tokens from Apple

5. Revision History

This chapter describes the changes to *Introduction to Software Authentication R3* from the previous revision.

- Updated section 3.1 with new process to request for additional auth entities

Apple Inc.
One Apple Park Way
Cupertino, CA 95014
408-996-1010