Strong Customer Authentication for Apple Pay on Mac mini with M2 Pro and Magic keyboard with Touch ID, running macOS Sonoma 14.4

Security Target

Version 4.0
October 1, 2024

Apple
One Apple Park Way
Cupertino, CA 95014

Table of Contents

1. Introduction

1.1. Purpose

The purpose of this document is to define the Target of Evaluation (TOE) for meeting the requirements of Directive (EU) 2015/2366 of the European Parliament and of the Council of 25 November 2015 on payment services in the internal market (PSD2) and the Commission Delegated Regulation (EU) 2018/389 of 27 November 2017, focusing on the Strong Customer Authentication and Dynamic Linking for Apple Pay.

1.2. Abbreviations

Abbreviation	Meaning
AP	Application Processor
API	Application Programming Interface
AR	Authorization Random
CL	Contactless
CRS	Contactless Registry Service
CVV	Card Verification Value
HSM	Hardware Security Module
I/O	Input / Output
MAC	Message Authentication Code
MK	Magic Keyboard
NFC	Near Field Communication
OS	Operating System
PNO	Payment Network Operator
PSD2	Directive (EU) 2015/2366 of the European Parliament and of the Council of 25 November 2015
SCA	Strong Customer Authentication
SIP	System Integrity Protection
SKS	Secure Key Store
SSE	An application in the Secure Enclave managing the pairing between the Secure Enclave and the Secure Element
TSM	Trusted Service Manager
TSP	Token Service Provider
UID	Unique Identifier

2. ST Introduction

2.1. Security Target Reference

This Security Target is identified with the following information:

Security Target identifiers	
Title	Strong Customer Authentication for Apple Pay on Mac mini with M2 Pro and Magic keyboard with Touch ID running macOS Sonoma 14.4, Security Target
Version	4.0
Date	October 1, 2024
Developer	Apple Inc.

2.2. Target of Evaluation Reference

TOE identifier
Strong Customer Authentication for Apple Pay on Mac mini with M2 Pro and Magic keyboard with Touch ID running macOS Sonoma 14.4

The Target of Evaluation (TOE) platform is a Mac mini with M2 Pro and Magic keyboard with Touch ID, running macOS Sonoma 14.4, with the following platform identifiers:

Platform identifiers	
Device	Mac mini with M2 Pro 2023 Magic keyboard with Touch ID for Mac models with Apple silicon Magic keyboard with Touch ID and Numeric Keypad for Mac models with Apple silicon Note: both Magic Keyboard models have part numbers starting with “MK2”
Operating System	macOS Sonoma 14.4 (Build 23E214)
Developer	Apple Inc.

Note: in the rest of the document, the term Magic Keyboard will be used standing for any of the Magic Keyboard device identified in the above table.

The TOE consists of a range of hardware and software components as listed below, which are all developed by Apple.

TOE Component	Version	Description
Apple Wallet App (Wallet)	App part of macOS Sonoma 14.4	Authentication policy on data and services In-app transaction data management

TOE Component	Version	Description
Application Processor[1] (AP)	M2 Pro	Authentication policy on data and services Transaction data management
Boot Loader	macOS Sonoma 14.4	Allows the device to start and boot the operating system
Secure Enclave	sepOS (part of macOS Sonoma 14.4)	Authentication Setup: • Enrollment of the authentication material, • User authentication verification, Authentication Prover: • Password verification, • Biometrics matching, • Authentication policy on data
• **SSE**		Manages the pairing between the Secure Enclave and the Secure Element
• **SKS**		Hardware Cryptographic module
• **BioApp**		Provides functionality for processing biometric data and generating biometric templates
macOS Platform	Device operating system platform (macOS Sonoma 14.4) executing on Application Processor (AP) with the following Apple Pay services that are included in the TOE:	
• **Security Framework**	macOS Sonoma 14.4	Provides functionality to protect information, establish trust, and control access to software
• **Logind**	macOS Sonoma 14.4	Provides functionality for managing user logins and sessions
• **NFCd**	macOS Sonoma 14.4	Provides communication layer between the TOE and the Secure Element
• **Safari**	Version 17.4 (19618.1.15.11.12)	Browser
• **System Settings**	macOS Sonoma 14.4	Allows the user to indicate their preferred system settings for the device, operating system and applications
• **Launchpad**	macOS Sonoma 14.4	Provides the functionality for the macOS user interface
Magic Keyboard	Keyboard controller Firmware version: 0x420	Device used by the user to operate the TOE, offering keyboard feature as well as TouchID sensor
• **Bluetooth controller**	Firmware version: 0x206	Provides wireless connection between the Mac mini and the Magic Keyboard with TouchID
• **Biometric Sensor (Touch ID)**	Touch ID sensor within the Magic Keyboard as listed in the platform identifiers table	Sensor for fingerprint capture
• **Crypto Block**	Firmware version: 0x5190	Support security features of the Magic Keyboard protecting the sensitive data

[1] Only the parts of the AP related to the TSF are included within the TOE scope. The GPU of the AP is not relevant to the TSF and is therefore not part of the TOE.

The Secure Element of the device is separately certified according to the Common Criteria and is therefore out of scope of this evaluation.

Note: *In the evaluated configuration the cryptographic modules are supplied by Apple as part of macOS and sepOS. Readers may draw some assurance from the conformance to FIPS 140-3 certified by the Cryptographic Module Validation Program for corecrypto for each major release (Apple corecrypto User Space Module for ARM, Apple corecrypto Kernel Space Module for ARM and the Apple Secure Key Store Cryptographic Module).*

Additionally, the browser, Safari, is evaluated for each major macOS release using the collaborative PP (cPP), Protection Profile for Application Software Version 1.3.

The TOE guidance document is listed in the following table.

Apple Pay Guidance	Reference	Version
Strong Customer Authentication for Apple Pay on Mac mini with M2 Pro and Magic keyboard with Touch ID, running macOS Sonoma 14.4: Guidance	[AGD]	4.0

2.3. TOE Overview

2.3.1. TOE type

The TOE is a combination of Hardware and Software components that implement Strong Customer Authentication and Dynamic Linking for Apple Pay on the Mac mini with M2 Pro and Magic keyboard with Touch ID, running macOS Sonoma 14.4.

2.3.2. TOE usage and major security features

The TOE includes the components implementing Strong Customer Authentication (SCA) and Dynamic Linking for Apple Pay. The TOE is the Mac mini with M2 Pro and Magic keyboard with Touch ID, running macOS Sonoma 14.4 operating system.

The operating system manages the device hardware, provides Apple Pay functionalities, and provides the technologies required to enforce Strong Customer Authentication and Dynamic Linking for Apple Pay e-commerce transactions. Dynamic Linking for a transaction is the link between the authentication code generated upon successful SCA, with both the transaction's original specific amount and the identity of the payee.

The operating system provides a consistent set of capabilities allowing the supervision of enrolled devices. This includes the preparation of devices for deployment, the subsequent management of the devices, and the termination of management.

The TOE platform protects itself by having its own code and data protected from unauthorized access (using hardware provided memory protection features), by securing user and TOE Security Functionality (TSF) data, and by ensuring the integrity and authenticity of TSF updates and downloaded applications, and by locking the TOE upon user request or after a defined time of user inactivity.

The TOE provides protection of data at rest, and access control mechanisms for use by applications. Access control for data and services, including Apple Pay, rely on the enforcement of user authentication.

To use Apple Pay, a user must have a passcode set on the device and, optionally, biometrics (Touch ID). User authentication to authorize an Apple Pay transaction on an enrolled device is provided by a user-defined password and the user enrolled biometrics. The minimum length of the password, password rules, and the maximum number of consecutive failed authentication attempts is statically set by Apple for each macOS release. Biometrics are enrolled and managed by the User. Up to 5 fingerprints can be enrolled using the biometric sensor of the Magic Keyboard on a single TOE by the user. Up to 5 Magic Keyboards can be paired with a single Mac device.

For greater convenience when using multiple Apple devices, some devices can automatically unlock other Apple devices in certain situations. For this security target, the considered usage is: *A Mac can be unlocked by an Apple Watch*. This can be enabled by the user with the "Auto Unlock with Apple Watch" setting in the System Settings.

When Auto Unlock with Apple Watch is enabled, a Mac device running macOS Sierra 10.12 or later can be unlocked using a paired Apple Watch Series 3 or later. The user can also use their Apple Watch to approve other requests to enter their administrator password. This usage does not include the user authentication for Apple Pay transactions. The paired Apple Watch that is used to unlock the Mac or approves other requests is not part of the TOE. The Secure Enclave component of the TOE implements a secure channel with the Watch.

The Secure Enclave is responsible for ensuring user authorization (the combination of User authentication and user intent) before a payment is authorized from the device.

2.3.3. Non-TOE hardware/software/firmware

The TOE environment includes the Secure Element (Hardware, Operating System, CRS applet and payment applets in the Secure Element) and the NFC Controller (NFCC).

The Secure Element is included in the device but is outside the scope of the TOE boundary as it is separately evaluated to Common Criteria. The Secure Element is contained in the same package as the NFCC. Payment applets in the Secure Element manage the payment process for Apple Pay transactions. The NFCC is not connected to any antenna and is thus not usable for NFC transactions. It is used only for communications between the TOE and the Secure Element.

The TOE relies on its environment to facilitate Apple Pay transactions; the transaction data is always processed by the Secure Element. The Secure Element only allows payment data to be sent from the device after it receives authorization from the Secure Enclave. For each transaction, the payment applets hosted on the Secure Element generate a payment cryptogram. This cryptogram and the Device Account Number form a transaction-specific dynamic security code, which is sent from the device to the card issuer (or its tokenization service provider such as a payment network) to use to verify each transaction.

The subsystems of the TOE are listed in Section 2.4 of this Security Target. All other components and subsystems of macOS (including user space and kernel software), hardware and subsystems included in the device (including the networking subsystem) and the paired Apple Watch are all considered to be part of the TOE environment. The networking subsystem provides connectivity to the Apple servers responsible for managing Apple Pay transactions and to the Apple Watch.

The "Unlock with Apple Watch" feature relies on a secure pairing process between the TOE and the Apple Watch, and a secure unlock process thereafter.

For Mac mini with M2 Pro, the user can connect up to three external displays in the following configurations:

- Connect one external display with up to 8K resolution at 60 Hz using the HDMI port.

- Connect one display with up to 4K resolution at 60 Hz using the HDMI port and up to two external displays with up to 6K resolution at 60 Hz using the Thunderbolt ports.
- Connect one external display with up to 4K resolution at 240 Hz using the HDMI port.
- Connect one display with up to 4K resolutions at 144 Hz using the HDMI port and one external display with up to 6K resolution at 60 Hz using the Thunderbolt ports.

https://support.apple.com/guide/mac-mini/apd8e4fbbb97/mac

2.4. TOE Description

2.4.1.TOE Architecture

The TOE platform includes the components implementing Strong Customer Authentication (SCA) and dynamic linking for Apple Pay.

User authentication is managed by the Secure Enclave. The Secure Enclave is a dedicated secure subsystem integrated into Apple systems on chip (SoCs). The Secure Enclave is isolated from the main processor to provide an extra layer of security and is designed to keep sensitive user data secure even if the Application Processor kernel were to be compromised.

macOS allows the Apple Pay services and other security functions of the TOE to operate.

The Secure Element (outside of the TOE) is the secure component that holds the Apple Pay secrets and processes the Apple Pay transactions. The Magic keyboard with Touch ID (and the Magic keyboard with Touch ID and Numeric Keypad) provides a Touch ID sensor in an external keyboard that can be used with any Mac with Apple silicon. The Magic keyboard with Touch ID performs the role of the biometric sensor; it doesn't store biometric templates, perform biometric matching or enforce security policies. The Touch ID sensor in the Magic keyboard with Touch ID must be securely paired to the Secure Enclave on the Mac before it can be used, and then the Secure Enclave performs the enrollment and matching operations and enforces security policies in the same way it would for a built-in Touch ID sensor.

A Magic keyboard with Touch ID can be securely paired with only one Mac at a time, but a Mac can maintain secure pairings with several Magic Keyboards with Touch ID (up to 5). Pairing can be performed by the user.

The guidance documentation of the TOE is listed in the *Apple Pay Guidance* table of section 2.2. The guidance documentation of the TOE is available online at Apple Pay Security Certifications.

The identifiers of the TOE components are given in the *TOE Component* table of section 2.2.

The distribution channels for Users to obtain the devices include:

- The "Apple Store" which is either physical store or online store at https://www.apple.com
- Apple retailer
- Resellers
- Other specific channels for Government and business

The component parts of the TOE are shown in Figure 1:

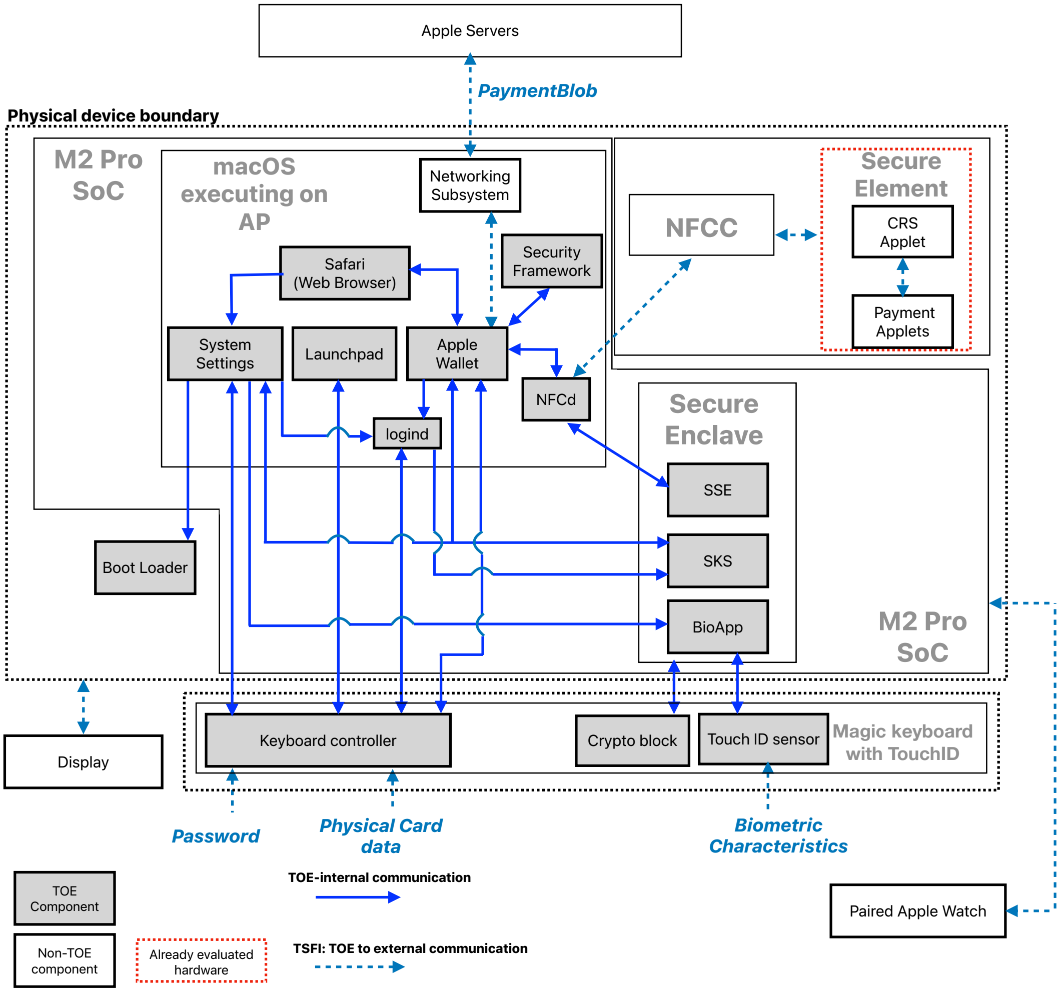

Figure 1: TOE Components and subsystems

2.4.2. Subsystems of the TOE

This section further breaks down the TOE components, providing more detail about the subsystems of the TOE.

The subsystems of the TOE consist of:

- Secure Enclave: The software components of the TOE residing in the Secure Enclave. This subsystem includes several applications executing on the Secure Enclave operating system.
  - BioApp is an application which provides functionality for processing biometric data and generating biometric templates.
  - The SKS (Secure Key Store) is a hardware cryptographic module. The module is embedded inside the Secure Enclave and packaged within the Application Processor.
  - The SSE (Secure Enclave - Secure Element) manages the pairing between the Secure Enclave and the Secure Element, allowing the Secure Element to process only genuine and authorized

Apple Pay transactions. The SSE application maintains sensitive pairing material, allowing Secure Element and Secure Enclave to perform a mutual authentication before exchanging data.

- Apple Wallet: The Wallet app subsystem is an application executing as part of macOS that handles the enrollment of payment applications and governs the payment operation.
- macOS components executing on Application Processor (AP):
  - logind: The component of macOS that provides the user interface to handle user password authentication
  - Launchpad: The component of macOS that handles I/O with the console, and thereby provides the functionality for the macOS user interface
  - NFCd: This daemon facilitates the communication between Apple Wallet and the Secure Element
  - Safari browser: Web browser included with the OS. Provides a web interface to conduct payment transactions
  - Security Framework: This is an API[2] provided by the OS to provide cryptographic support that can be used to protect information, establish trust, and control access to software
  - System Settings application: This application allows the user to modify various system settings
- Host device component:
  - Boot-loader: This subsystem consists of code that is executed during the boot sequence of the device. The boot-loader is responsible for ensuring that the device boots using software with assured integrity and authenticity
  - Bluetooth controller: Allows the Mac mini to connect to the Magic Keyboard
- Magic keyboard with Touch ID:
  - Keyboard controller: Allows the Magic Keyboard to connect to the Mac mini
  - Crypto Block: This is a hardware cryptographic module performing the cryptographic operations securing the data exchanged between the Magic Keyboard and the Mac mini.
  - Touch ID sensor: This is the hardware component and associated drivers that allow fingerprint data to be captured and passed to BioApp for enrollment and matching.

All other Mac, macOS, and keyboard components are outside of the TOE, including the Secure Element together with the NFCC hardware, and the XNU macOS kernel. The macOS subsystem components are individual applications.

[2] Refer to https://developer.apple.com/documentation/security/

2.4.3. TOE Lifecycle

The TOE lifecycle phases are as follows:

Lifecycle Phases			
Design	HW/FW design	SW design	Secure Element Applet design
Fabrication	HW fabrication	SW implementation	Applet development
Integration	Mac mini and Magic keyboard with Touch ID integration *Assembly, Trust provisioning, FW integration, SW and applet loading*		
Device Issuance	Device delivery to User		
Initialization	User account creation, device pairing Account setup: iCloud, Apple ID		
Enrollment/ Provisioning	User Authentication setup *Password setup, biometrics enrollment*		Apple Pay provisioning
Usage	Device Usage *Device unlock, User Authentication, macOS use, (optional) macOS/keyboard firmware update*		Apple Pay transaction
Termination	Physical destruction	macOS recovery mode	Apple Pay termination

The Design, Fabrication and Integration phases are entirely within the control of Apple Inc.

The Device Issuance phase is the physical delivery of the device to the User. This can be directly, in the case of an individual, or through a third party or an entity that is responsible for providing the device to the User.

Apple's model of the Initialization phase requires that the device is claimed by the User by associating it with the User's iCloud account and Apple ID. Before that point, Strong User Authentication and associated TSFs are not relevant, and Apple Pay is not accessible. Apple Pay services require that a valid iCloud account and user authentication credentials are setup (password and optionally biometrics). During the Initialization phase, the Mac mini and the Magic Keyboard are paired, including the secure pairing between the Touch ID sensor in the Magic Keyboard and the Secure Enclave on the Mac.

The Usage phase describes the period when the Apple Pay service is activated and used by the associated User.

The Termination phase describes deactivation of the Apple Pay service for the User and may involve physical destruction of the device as well as a complete reset or erase and recovery of macOS (macOS recovery mode) or Apple Pay service termination by the User.

When the device is received, the model of the device should be checked to verify that the model number is one of those listed in Section 2.2. This can be accomplished using any of the following methods.

- Physically checking:
  - Mac mini: the serial number is printed on the underside of the Mac and on the original packaging.
  - Magic keyboard with Touch ID: The serial number of your Magic Keyboard is on the bottom surface of the device, along with other markings.

  The User can then enter that serial number on the Check Coverage page to find the model.
- Once authenticated to the device the information is available to device users in the "About this Mac" overview, accessed from the menu, or within the System Information app.

2.4.4. TOE security features

The logical security features of the TOE are summarized as follows:

- User authentication and management
- Secure channel between the Mac mini and the Magic Keyboard
- Secure channel between the Secure Enclave and the Secure Element
- Secure channel between the Secure Enclave and the Apple Watch
- Card Data management
- Apple Pay payment transaction processing and management
- Operating System update
- iCloud logout and disk erasure

2.5. Description of the Apple Pay Service

This section contains a generic description of the Apple Pay service in general, which includes the TOE as well as the TOE environment and non-TOE hardware, software, and services.

2.5.1. Card provisioning

When a user adds a credit, debit, or prepaid card (including store cards) to Apple Wallet, the device encrypts the card information and securely sends it, along with other information about the user's account and device, through Apple Pay servers, to the card issuer or the card issuer's authorized service provider (usually the payment network). Using this information, the card issuer (or its service provider) will determine whether to approve adding the card to Apple Wallet.

As part of the card provisioning process, Apple Pay uses three server-side calls to send and receive communication with the card issuer or payment network:

- Required Fields
- Check Card
- Link and Provision

The card issuer or payment network uses these calls to enable the card issuer to verify, approve, and add cards to Apple Wallet. These client-server sessions are protected for confidentiality and integrity using TLS 1.2.

The full card numbers are never stored on the device or on Apple servers. Instead, a unique Device Account Number is created, encrypted, and then stored in the Secure Element. This unique Device Account Number is encrypted in such a way that Apple cannot access it. The Device Account Number is unique and different from most credit or debit card numbers; the card issuer or payment network can prevent its use on a magnetic stripe card, over the phone, or on websites. The Device Account Number located in the Secure Element is isolated from the TOE, is never stored on Apple servers, and never backed up to iCloud.

With macOS 14 or later, when a user provisions an eligible payment card, the user can push provision the card to other Apple Pay-capable devices on the same iCloud account using the Multi-device provisioning feature. Nothing is copied from the original device; the other devices provision using the same flow they would use during device setup.

2.5.1.1.Adding credit or debit cards manually

To add a card manually, the name, card number, expiration date, and card verification value (CVV) are used to facilitate the provisioning process. From within System Settings, Apple Wallet, or the Apple Watch app, users can enter that information by typing it. After all the fields are filled in, the Check Card process verifies the fields other than the CVV. They are then encrypted and sent to the Apple Pay server.

If a terms and conditions ID is returned with the Check Card process, Apple downloads and displays the terms and conditions of the card issuer to the user. If the user accepts the issuer’s terms and conditions, Apple sends the ID of the terms that were accepted as well as the CVV to the Link and Provision process.

2.5.1.2.Adding credit or debit cards from an iTunes Store account

For a credit or debit card on file with iTunes, the user may be required to reenter their Apple ID password. The card number is retrieved from iTunes and the Check Card process is initiated. If the card is eligible for Apple Pay, the Apple Wallet application downloads and displays terms and conditions of the card issuer, then sends along the terms and conditions ID and the card security code to the Link and Provision process. Additional verification may occur for iTunes account cards on file.

2.5.1.3.Adding credit or debit cards from a card issuer’s website

Some card issuers provide the ability to initiate the card provisioning process for Apple Wallet directly from their websites. In this case, the user initiates the task by selecting a card to provision on the card issuer’s website. The user is then redirected to a self-contained Apple sign-in experience (contained within Apple’s domain) and is asked to sign in with their Apple ID. Upon successfully signing in, the user then chooses one or more devices to provision the card to and is required to confirm the provisioning result on each respective target device.

2.5.1.4.Additional verification

A card issuer can decide whether a credit or debit card requires additional verification. Depending on what is offered by the card issuer, the user may be able to choose between different options for additional verification, such as a text message, email, a customer service call, or a method in an approved third-party app to complete the verification. For text messages or email, the user is presented an option to select from contact information the issuer already holds on file. A code is sent, which must be entered into System Settings, Apple Wallet, or the Apple Watch app. For customer service or verification using an app, the issuer performs their own communication process.

2.5.2.Payment authorization

For devices having a Secure Enclave, a payment can be made only after it receives authorization from the Secure Enclave. This involves confirming that the user has authenticated with Touch ID or the device password and has double-pressed the Touch ID sensor. Touch ID, if available, is the default method, but the password can be used at any time. A password is automatically offered after three unsuccessful attempts to match a fingerprint; after five unsuccessful attempts, the password is required. A password is also required when Touch ID is not configured or not enabled for Apple Pay.

2.5.2.1.Using a shared pairing key

Communication between the Secure Enclave and the Secure Element takes place over a serial interface, with the Secure Element connected to the Near Field Communication (NFC) controller, which in turn is connected to the Application Processor. Though not directly connected, the Secure Enclave and the Secure Element can communicate securely using a shared pairing key. The encryption and authentication of the communication are based on AES, with cryptographic nonces used by both sides to protect against replay attacks.

2.5.2.2.Authorizing a secure transaction

When the user authorizes a transaction, which includes a physical gesture communicated directly to the Secure Enclave, the Secure Enclave sends signed data about the type of authentication and details about the type of transaction (contactless or e-commerce) to the Secure Element, tied to an Authorization Random (AR) value. The AR value is generated in the Secure Enclave when a user first provisions a credit card and persists while Apple Pay is enabled, protected by the Secure Enclave's encryption and anti-rollback mechanism. It is securely delivered to the Secure Element by leveraging the pairing key. On receipt of a new AR value, the Secure Element marks any previously added cards as deleted.

2.5.2.3.Using a payment cryptogram for dynamic security

Payment transactions originating from the payment applets include a payment cryptogram along with a Device Account Number. This cryptogram, a one-time code, is computed using a transaction counter and a key. The transaction counter is incremented for each new transaction. The key is provisioned in the payment applet during personalization and is known by the payment network or the card issuer or both. Depending on the payment scheme, other data may also be used in the calculation, including:

- An Apple Pay server nonce
- User verification results, such as Cardholder Verification Method (CVM) information

These security codes are provided to the payment network and to the card issuer, which allows the issuer to verify each transaction. The length of these security codes may vary based on the type of transaction.

2.5.3.Paying with cards using Apple Pay

2.5.3.1.Paying with cards within apps

Apple Pay can also be used to make payments on macOS apps. When users pay within apps using Apple Pay, Apple receives the encrypted transaction information to route to the developer or merchant. Before that information is sent to the developer or merchant, Apple re-encrypts it with a developer-specific key. Apple Pay retains anonymous transaction information, such as approximate purchase amount. This information can’t be tied to the user and never includes what the user is buying.

When an app initiates an Apple Pay payment transaction, the Apple Pay servers receive the encrypted transaction from the device prior to the merchant receiving it. The Apple Pay servers then re-encrypt the transaction with a merchant-specific key before relaying it to the merchant.

When an app requests a payment, it calls an API to determine whether the device supports Apple Pay and whether the user has credit or debit cards that can make payments on a payment network accepted by the merchant. The app requests any pieces of information it needs to process and fulfill the transaction, such as the billing and shipping address, and contact information. The app then asks macOS to present the Apple Pay sheet, which requests information for the app as well as other necessary information, such as the card to use.

At this time, the app is presented with city, state, and zip code information to calculate the final shipping cost. The full set of requested information isn’t provided to the app until the user authorizes the payment with Touch ID or the device password. After the payment is authorized, the information presented in the Apple Pay sheet is transferred to the merchant.

2.5.3.2.Paying with cards within App Clips

An App Clip is a small part of an app that allows a user to do a task quickly (such as renting a bike or paying for parking) without downloading the full app. If an App Clip supports payments, the user can use Sign in with Apple, then make a payment using Apple Pay. When a user makes a payment from within an App Clip, all security and privacy measures are the same as when a user pays within an app.

2.5.3.3.App payment authorization

When the user authorizes the payment, a call is made to the Apple Pay servers to obtain a cryptographic nonce, which is similar to the value returned by the NFC terminal used for in-store transactions. The nonce, along with other transaction data, is passed to the Secure Element to compute a payment credential that is encrypted with an Apple key. The encrypted payment credential is returned to the Apple Pay servers, which decrypt the credential, verify the nonce in the credential against the nonce originally sent by the Apple Pay servers, and re-encrypt the payment credential with the merchant key associated with the Merchant ID. The payment is then returned to the device, which hands it back to the app through the API. The app then passes it along to the merchant system for processing. The merchant can then decrypt the payment credential with its private key for processing. This, together with the signature from Apple’s servers, allows the merchant to verify that the transaction was intended for this particular merchant, and ensures dynamic linking of the transaction with its amount and the payee.

The APIs require an entitlement that specifies the supported Merchant IDs. An app can also include additional data (such as an order number or customer identity) to send to the Secure Element to be signed, ensuring the transaction can’t be diverted to a different customer. This is accomplished by the app developer, who can specify applicationData on the PKPaymentRequest. A hash of this data is included in the encrypted payment data. The merchant is then responsible for verifying that their applicationData hash matches what’s included in the payment data.

In order to process payments, the merchant takes the following steps:

- Send the payment information to their server, along with the other information needed to process the order
- Verify the hashes and signature of the payment data
- Decrypt the encrypted payment data, and confirm the validity of the transactionId, currencyCode, transactionAmount, and applicationData fields
- Submit the payment data to the payment processing network and the order to their order-tracking system

2.5.3.4. Paying with cards at websites

Apple Pay can be used to make payments at websites on Mac computers with Touch ID. Apple Pay transactions started on a Mac can also be completed on another Apple Pay–enabled device using the same iCloud account.

Apple Pay on the web requires that all participating websites register with Apple. After the domain is registered, domain name validation is performed only after Apple issues a TLS client certificate. Websites supporting Apple Pay are required to serve their content over HTTPS. For each payment transaction, websites need to obtain a secure and unique merchant session with an Apple server using the Apple-issued TLS client certificate. Merchant session data is signed by Apple. After a merchant session signature is verified, a website may query whether the user has an Apple Pay–capable device and whether they have a credit, debit or prepaid card activated on the device. No other details are shared.

If the user doesn’t want to share this information, they can disable Apple Pay queries in Safari privacy settings on Mac devices.

After a merchant session is validated, all privacy and security measures are the same as when a user pays within an app.

If the user is transmitting payment-related information from a Mac to an iPhone or Apple Watch, Apple Pay Handoff uses the end-to-end encrypted Apple Identity Service (IDS) protocol to transmit payment-related information between the user’s Mac and the authorizing device. The IDS client on the Mac uses the user’s device keys to perform encryption so no other device can decrypt this information, and the keys are not available to Apple. Device discovery for Apple Pay Handoff contains the type and unique identifier of the user’s credit cards along with some metadata. The device-specific account number of the user’s card is not shared and it continues to remain stored securely on the user’s iPhone or Apple Watch. Apple also securely transfers the user’s recently used contact, shipping and billing addresses over iCloud Keychain.

After the user authorizes a payment using Touch ID, Face ID, a passcode or double-clicking the side button on Apple Watch, a payment token uniquely encrypted to each website’s merchant certificate is securely transmitted from the user’s iPhone or Apple Watch to their Mac and then delivered to the merchant’s website.

Only devices in proximity to each other may request and complete payment. Proximity is determined through Bluetooth Low Energy (BLE) advertisements.

The Apple Pay Handoff feature is not in the scope of this Security Target. After the Handoff process, the transaction validation is ensured by the user’s iPhone or Apple Watch. The transaction validation is covered by the Security Target of the iPhone or the Apple Watch.

2.5.3.5. Automatic payments and Merchant Tokens

In macOS 13 or later, apps and websites that offer Apple Pay can take advantage of Apple Pay merchant tokens that enable secure payments consistently across a user’s Apple devices. The updated Apple Pay payment sheet in macOS 13 also optimizes preauthorized payment experiences. New transaction types in the Apple Pay API allow app and website developers to fine-tune the payment sheet experience for subscriptions, recurring bills, instalment payments, and automatic reloads of card balances.

Merchant tokens are not device-specific, and therefore allow for continuity of recurring payments if the user removes a payment card from the device.

2.5.3.6.Payments to multiple merchants

In macOS 13 or later, Apple Pay includes the ability to specify purchase amounts for multiple merchants within a single Apple Pay payment sheet. This allows the flexibility to let customers make a bundled purchase, such as a travel package with flight, rental car, and hotel, then send payments to individual merchants.

2.5.4.Rendering cards unusable with Apple Pay

Credit, debit, and prepaid cards added to the Secure Element can only be used if the Secure Element is presented with authorization using the same pairing key and Authorization Random (AR) value from when the card was added. On receipt of a new AR value, the Secure Element marks any previously added cards as terminated. This allows the operating system to instruct the Secure Enclave to render cards unusable by marking its copy of the AR as invalid under the following scenarios:

- The password is disabled
- The user signs out of iCloud
- The user selects Erase All Content and Settings
- The device is restored from Recovery Mode

2.5.4.1.Suspending, removing, and erasing cards

Users can suspend Apple Pay on the device by placing their devices in Lost Mode using Find My. Users also have the ability to remove and erase their cards from Apple Pay using Find My, iCloud.com, or directly on their devices using Apple Wallet. The ability to make payments using cards on the device is suspended or removed from Apple Pay by the card issuer or respective payment network, even if the device is offline and not connected to a cellular or Wi-Fi network. Users can also call their card issuer to suspend or remove cards from Apple Pay.

When a user erases the entire device – using Erase All Content and Settings, using Find My, or restoring their device – Mac instructs the Secure Element to mark all cards as terminated. This has the effect of immediately changing the cards to an unusable state until the Apple Pay servers can be contacted to fully erase the cards from the Secure Element. Independently, the Secure Enclave marks the Authorization Random as invalid so that further payment authorizations for previously enrolled cards are not possible. When the device is online, it attempts to contact the Apple Pay servers to help ensure that all cards in the Secure Element are erased.

2.5.5.Unlock with Apple Watch

A Mac device running macOS Sierra 10.12 or later can be unlocked using a paired Apple Watch Series 3 or later. The user can also use their Apple Watch to approve other requests to enter their administrator password. This works anywhere the user needs to type their Mac password, such as when viewing passwords in Safari preferences, unlocking a locked note, approving an app installation, or unlocking settings in System Settings. The paired Apple Watch does not authenticate a user for use with Apple Pay.

2.5.6.Apple Card

2.5.6.1.Apple Card application

On supported models of iPhone and Mac, a user can securely apply for an Apple Card.

In iOS 12.4 or later, macOS 10.14.6 or later, and watchOS 5.3 or later, Apple Card can be used with Apple Pay to make payments in stores, in apps and on the web. Apple Card is currently only available for qualifying applicants in the United States.

To apply for Apple Card, the user must be signed into their iCloud account on an Apple Pay–compatible iOS or iPadOS device and have two-factor authentication set up on the iCloud account. When the application is approved, Apple Card is available in the Apple Wallet app or within System Settings > Wallet & Apple Pay across any of the eligible devices the user has signed in with their Apple ID.

When a user applies for Apple Card, user identity information is securely verified by Apple's identity provider partners and then shared with Goldman Sachs Bank USA for the purposes of identity and credit evaluation.

Information such as the social security number or ID document image provided during the application is securely transmitted to Apple's identity provider partners and/or Goldman Sachs Bank USA encrypted with their respective keys. Apple cannot decrypt this data.

The income information provided during the application, and the bank account information used for bill payments, are securely transmitted to Goldman Sachs Bank USA encrypted with their key. The bank account information is saved in Keychain. Apple cannot decrypt this data.

When adding Apple Card to the Apple Wallet app, the same information as when a user adds a credit or debit card may be shared with the Apple partner bank, Goldman Sachs Bank USA, and with Apple Payments Inc. This information is used only for troubleshooting, fraud prevention, and regulatory purposes.

In iOS 14.6 or later, iPadOS 14.6 or later, and watchOS 7.5 or later, the organizer of an iCloud family with an Apple Card can share their card with their iCloud Family members over the age of 13. User authentication is required to confirm the invitation. Apple Wallet uses a key in the Secure Enclave to compute a signature that binds the owner and the invitee. That signature is validated on Apple servers.

Optionally, the organizer can set a transaction limit to the participants. Participant cards can also be locked to pause their spending at any time through the Apple Wallet app. When a co-owner or participant over the age of 18 accepts the invitation and applies, they go through the same application process in the Apple Wallet app as defined above.

A physical card can be ordered from Apple Card in Apple Wallet. After the user receives the physical card, it is activated using the NFC tag that's in the bifold envelope of the physical card. The tag is unique per card and cannot be used to activate another user's card. Alternatively, the card can be manually activated in Apple Wallet settings. Additionally, the user can also choose to lock or unlock the physical card at any time from Apple Wallet.

2.6. TOE Use Cases

The TOE covers the following use cases:

Use Case	Description
UC.Device_Usage	Device usage The User can manage the device's authentication credentials, including enrolling new biometric templates, updating biometric templates, deleting biometric templates and changing the password.

Use Case	Description
UC.OS_Update	Device Software Update The User can perform an update of the software in the macOS device or the Magic Keyboard firmware to a new version. This use case requires that the user verifies the device's password. This use case ensures preservation of the User settings on the device: • No change to the User's authentication credentials (password or any biometrics) • No change to the User's data within the Secure Element unless specified by the data's issuer
UC.Apple_Pay_Install_Init	Apple Pay installation and initialization The User can provision a new card Apple Wallet
UC.Apple_Pay_Usage	Apple Pay usage The User can perform Apple Pay transactions.
UC.End_Of_Service	Termination by User The User can end the Apple Pay mode of operation by performing a card removal in Apple Wallet. The User can also end all current Apple Pay services by un-registering their iCloud account. Termination by card issuer The card issuer can perform a de-registration of an Apple Pay card that was provisioned on the User's device, following a card revocation or a user account termination.
UC.End_Of_Life	Termination of device The User can clear a device from all their settings and data by performing a disk full erase or device reset. The User could also end the life of their device by physically destroying it.

3. Evaluation Assurance

3.1. Common Criteria Reference

This Security Target is based on the following Common Criteria ™ (CC) publications:

Common Criteria	CC Version	Revision	Date
Part 1: Introduction and general model	CC:2022	R1	November 2022
Part 2: Security functional requirements	CC:2022	R1	November 2022
Part 3: Security assurance requirements	CC:2022	R1	November 2022

3.2. CC Conformance claim

This Security Target is **conformant** to CC Part 2 and CC Part 3.

3.3. Protection Profile Conformance claim

This Security Target does not claim any conformance to an existing Protection Profile.

3.4. Assurance Level

The evaluation assurance level (EAL) for this work is EAL 2 augmented with ADV_FSP.3 and ALC_FLR.3:

Assurance Class	
ADV: Development	ADV_ARC.1 Security architecture description
	ADV_FSP.3 Functional specification with complete summary
	ADV_TDS.1 Basic design
AGD: Guidance documents	AGD_OPE.1 Operational user guidance
	AGD_PRE.1 Preparative procedures
ALC: Life-cycle support	ALC_CMC.2 Use of a CM system
	ALC_CMS.2 Parts of the TOE CM coverage
	ALC_DEL.1 Delivery procedures
	ALC_FLR.3 Systematic flaw remediation
ASE: Security Target evaluation	ASE_CCL.1 Conformance claims
	ASE_ECD.1 Extended components definition
	ASE_INT.1 ST introduction
	ASE_OBJ.2 Security objectives
	ASE_REQ.2 Derived security requirements
	ASE_SPD.1 Security problem definition
	ASE_TSS.1 TOE summary specification
ATE: Tests	ATE_COV.1 Evidence of coverage
	ATE_FUN.1 Functional testing
	ATE_IND.2 Independent testing – sample

Assurance Class	
AVA: Vulnerability assessment	AVA_VAN.2 Vulnerability analysis

4. Security Problem Definition

4.1. Assets

The assets of the TOE are:

Asset	Sensitivity*	Type	Definition
D.User_Password	I, C	User	Password value setup by the User, used to wake up the device after power loss, unlock the device, and to authorize payments (Apple Pay transactions).
D.User_Bio	I, C	User	Biometric data for the enrolled User's biometrics and used to unlock the device and to authorize payments (Apple Pay transactions).
D.Card_Data	I, C	User, TSF	Apple Pay card data, including credit/debit card number, User's name, expiration date, CVV, and transaction data (e.g. transaction history). Note: The card data is used as TSF data when the TOE sends it encrypted to the card issuer, via Apple servers, in order to generate the Device Account Number stored in the Secure Element.
D.User_Intent	I	User	State of the device resulting from a physical interaction of the User with the TOE, characterizing the intent of the User to perform an Apple Pay transaction.
D.Payment_Data	I	User	Apple Pay payment data being authorized by the user. For Apple Pay, critical elements are the amount (including the currency), the emitter, and the recipient (S.Merchant) which constitute the core of the Dynamic Linking.
D.OS	I	TSF	OS/MK firmware version currently installed on the device. This asset is not the OS/MK firmware code itself or only the version number of that code. This asset is the set of elements that compose an OS/MK firmware version as prepared for that device, including elements verifiable by that device during boot (for instance the sepOS), and by the User through an identified version number.
D.User_Configuration	I	TSF	User configuration is a representation of the user personal configuration including Apple Pay, and Touch ID settings.
D.SEP_Configuration	I	TSF	Secure Enclave configuration is a representation of the state of the Secure Enclave in the device. It comprises (but is not limited to) the Secure Enclave OS version and the state of the user authentication functions (password, enrolled biometrics, authenticated credentials, retry counters, and authentication-based access control settings).

Asset	Sensitivity*	Type	Definition
D.SEP_SE	I, C	TSF	Secret data that allows secure communication between the Secure Enclave and the Secure Element during processing of an Apple Pay transaction.
D.SEP_Bio	I, C	TSF	Secret data that allows secure communication between the Secure Enclave and the biometric sensor during processing of biometric authentication.
D.Keyboard_Secret	I, C	TSF	Secret data that allows secure communication between the Mac mini and each paired Magic Keyboard.
D.SEP_Watch	I, C	TSF	Secret data that allows secure communication between the Secure Enclave on the TOE and the Secure Enclave on the Paired Watch, in the case the Paired Watch is used to unlock the TOE. The pairing secret allows the TOE and the Paired Watch to exchange sensitive data like the unlock secret.
D.Unlock_Secret	I,C	TSF	Secret data that is shared by the Secure Enclave on the TOE with the Paired Watch Secure Enclave to unlock the TOE. It is released by the Paired Watch to facilitate the TOE unlock. This asset is created by the TOE when the Unlock with Apple Watch feature is enabled by the User and does not exist if there is no Paired Watch or if this feature is disabled.

*: I = Integrity, C = Confidentiality

4.2. Subjects

The subjects of this Security Target are:

Subject	Definition
S.User	User of Apple Pay on the device, able to: - Authenticate on the device (biometrics or password) - Manage authentication credentials (biometrics or password) - Manage device configuration (OS version, iCloud account) - Provision/enroll cards - Authorize Apple Pay transactions by providing consent for the transaction to proceed (biometrics/password and user intent) - Cancel the Apple Pay service
S.Apple_Servers	Apple servers in charge of: - Management of S.User iCloud account - Management of S.User provisioning/enrollment in Apple Pay - Management of OS releases, including the Wallet application - Device's interface for processing Apple Pay transactions (contact S.Issuer)
S.Issuer	The card issuer (or its service provider) is the third party in charge of: - Management of S.User data for Apple Pay services - Processing Apple Pay transactions
S.Merchant	The merchant is the third-party accepting payment through an Apple Pay transaction.

Subject	Definition
S.SE	Certified Secure Element of the device including the TOE.
S.Apple_Watch	Apple Watch of S.User paired with the TOE (if applicable).

4.3. Assumptions

The assumptions for this Security Target are the following:

Assumption	Definition
A.DEVICE_AUTH	The User of the device ensures that all the Apple Pay activities that are performed on the device have been authorized by the user. All authentication credentials (biometrics or password) for the device that are enabled for use with Apple Pay are owned and protected by the User of the device.
A.PERSO	The Apple Pay card issuers guarantee the correctness of the card data input in the device during provisioning/enrollment: Data shall uniquely identify a financial payment means and be linked to the account owned by the identified user.
A.WATCH_USER	If the User optionally pairs an Apple Watch to their Mac device, they own it and ensure the confidentiality of its authentication credentials.
A.NO_EVIL_ROOT_USER	All Users of the device with root privileges are non-hostile.

4.4. Threat Agents

The threat agents of this TOE are the following:

Threat Agent	Definition
S.Attacker	A threat agent trying to interact with the Apple Pay system fraudulently, trying to modify the configuration and data of genuine Users' devices or forging data on their own device.

4.5. Threats

The threats to assets of this TOE are the following:

Threat	Name	Definition	Assets
T.CORRUPT	Corrupted Transaction or Transfer	An attacker attempts to corrupt an Apple Pay transaction. To gain from the attack, the attacker could be the emitter of the transaction or transfer, and attempt to reduce what it intends to pay (debit amount from attacker's account) from what it was supposed to pay (credit amount request or transfer amount agreed between attacker and victim). The attacker could also attempt to corrupt a payment when it is the recipient, and increase what it supposed to receive (credit transaction amount) from what it was supposed to receive (debit amount agreed). The attacker could also attempt to modify the recipient of a credit transaction by changing the payee in an Apple Pay transaction.	D.Payment_Data D.User_Configuration D.OS

Threat	Name	Definition	Assets
T.PHYSICAL	Physical	The loss or theft of the device may give rise to loss of confidentiality of User data including credentials and TSF data. These physical access threats may involve attacks which attempt to access the device through external hardware ports or wireless connection of the Magic Keyboard, through its user interface, and also through direct and possibly destructive access to its storage media. The goal of such attacks is to access Apple Pay from a lost or stolen device which is not expected to return to its User. *Note: Defending against device re-use after physical compromise is out of scope.*	D.User_Password D.User_Bio D.Card_Data D.Unlock_Secret* D.Keyboard_Secret D.SEP_Watch*
T.RECOVER	Card Recovery	An attacker attempts to recover Apple Pay card data from an erased or blocked device and use it to perform a financial transaction or transfer. The attacker will target potential breaches in the process of Apple Pay cancellation, Apple Pay card revocation or macOS recovery mode.	D.User_Configuration D.Card_Data D.OS
T.REPLAY	Replay	An attacker attempts to replay an Apple Pay transaction.	D.Payment_Data
T.SILENT	Silent Transaction	An attacker attempts to modify the behavior of the device, in order to perform silent Apple Pay transactions for some benefit. The attacker would have to perform the attack without knowledge of the device's rightful owner who will be the victim of the attack.	D.User_Intent D.User_Configuration D.SEP_Configuration D.SEP_SE D.OS D.Keyboard_Secret
T.SKIMMING	Authentication Bypass	An attacker attempts to perform a payment with Apple Pay, bypassing the required authentication step (biometrics data verification or password verification).	D.User_Intent D.Payment_Data D.SEP_Configuration D.User_Configuration D.OS D.Keyboard_Secret
T.USURP	Card Ownership Usurpation	An attacker could attempt to authenticate on a device, with a goal of using any provisioned cards on that device. The attacker could focus on the card data during Apple Pay provisioning.	D.User_Bio D.User_Password D.User_Configuration D.Keyboard_Secret D.SEP_Bio D.Card_Data D.OS D.Unlock_Secret* D.SEP_Watch

* Partial threat: By unlocking the device, the attacker could have a higher chance to access Apple Pay data.

**Partial threat: the device can be unlocked using the paired Watch, but it does not allow the User to use provisioned cards.

The following table shows the Assets & Threats mapping:

Asset – Property **-** **I = Integrity** **C = Confidentiality**		**T.CORRUPT**	**T.PHYSICAL**	**T.RECOVER**	**T.REPLAY**	**T.SKIMMING**	**T.SILENT**	**T.USURP**
D.Unlock_Secret	I,C		X					X
D.User_Bio	I,C		X					X
D.User_Password	I,C		X					X
D.User_Intent	I					X	X	
D.Payment_Data	I	X			X	X		
D.Card_Data	I,C		X	X				X
D.OS	I	X		X		X	X	X
D.User_Configuration	I	X		X		X	X	X
D.SEP_Configuration	I					X	X	
D.SEP_Watch	I,C		X					X
D.SEP_SE	I,C						X	
D.SEP_Bio	I,C							X
D.Keyboard_Secret	I,C		X			X	X	X

4.6. Organizational Security Policies

The organizations associated with the Apple Pay service shall comply with the following Organizational Security Policies as security rules, procedures, practices, or guidelines imposed by an organization upon its operations.

OSP	Definition
P.UPDATE	Apple ensures that only an authenticated user can update their device's operating system and firmware to a newly released version. Apple also informs the Apple Pay card issuers and PNOs of new applicable features of new releases.
P.DYN_LINK	Apple maintains the enforcement of Dynamic Linking for e-commerce payments using Apple Pay cards, on device side and server side. This Organizational Security Policy guarantees that Apple preserves the following properties from design to feature release for Apple Pay e-commerce payments: (a) The payer is made aware of the amount of the payment transaction and of the payee (b) The authentication code generated is specific to the amount of the payment transaction and the payee agreed to by the payer when initiating the transaction (c) The authentication code accepted by the payment service provider corresponds to the original specific amount of the payment transaction and to the identity of the payee agreed to by the payer (d) any change to the amount or the payee results in the invalidation of the authentication code.

OSP	Definition
P.WATCH	The user can enable auto unlock with Apple Watch. The user can also use their Apple Watch to approve other requests (related to the security functions of the TOE) to enter their administrator password. These requests do not include user authentication for Apple Pay transactions.

5. Security Objectives

5.1. Security Objectives for the TOE

Security Objectives	Definition
OT.User_Auth	The TOE enforces the following authentication policy: • Password only: - Add, update, or delete biometrics - Update password - Modify authentication policy - Update the OS and MK firmware to a new version signed by Apple • Password or biometric (if enrolled) - Unlock of the device - Payments confirmation • Paired Apple Watch (optional): - Unlock of the device - Approve other requests (related to the security functions of the TOE) to enter their administrator password. These requests do not include user authentication for Apple Pay transactions.
OT.Card_Data	The TOE enforces that sensitive card data: • Is encrypted before being sent to the Apple servers • Is not accessible after being sent to the Apple server
OT.Password_Delete	The TOE enforces that removing the password: • Disables biometric authentication • Disables Apple Pay
OT.Card_Delete	The TOE securely triggers the delete of each individual Apple Pay card when: • A card is removed from Apple Wallet, • The TOE handles the revocation of a card by its issuer, • The use of Apple Pay is disabled (from iCloud, the System Settings or password removal), • The iCloud account is no longer associated with the device. When a card is removed, the TOE also instructs the Secure Element to mark it as deleted.
OT.Auth_SE	The TOE provides the Secure Element with password/biometric user authentication feature for Apple Pay payment approval.
OT.Payment	For e-commerce transactions, the TOE enforces that transaction details are displayed to the User (including the card to be used from the Wallet, the amount, and the payee) before the User shows their intent to pay and authenticates for payment validation. The TOE ensures that these details cannot be corrupted between the payment validation and the moment when details are sent to the Secure Element. The TOE also requests explicit intent from the User for Apple Pay transactions.
OT.Bio_Delete	TOE Biometrics delete is a secure erase of the enrolled biometric data.
OT.Disk_Erase	TOE securely deletes all User data when the User launches an “Erase All Content and Settings” on the OS, sets the OS to recovery mode, or does a device erase from iCloud. This also initiates the disabling of Apple Pay and plans the destruction of Apple Pay cards that will be effective when the device is restored.

Security Objectives	Definition
OT.Anti_Replay	The TOE ensures that each payment processed by an Apple Pay card holds the unique identifier.
OT.OS_Update	TOE enforces security measures ensuring preservation of User data when an OS/MK firmware update is installed in the TOE. This objective protects the user authentication credentials (biometrics and password), the Apple Pay card data, and more.
OT.Watch	The TOE allows the user to do the following operations using their Apple Watch through a secure channel: • unlock the TOE • approve other requests (related to the security functions of the TOE) to enter their administrator password. These requests do not include user authentication for Apple Pay transactions.

5.2. Security Objectives for the environment

Environment Security Objectives	Definition
OE.Card_Data	The card issuer is responsible for using the appropriate security measures to protect the confidentiality and the integrity of the sensitive card data and guaranteeing the authenticity of the card during enrolment. The Secure Element is responsible for securing the card validation exchanges with the card issuer's TSM and for ensuring confidentiality and integrity of each card's sensitive data during storage and use.
OE.Perso	The card issuer is responsible for verifying that the device User is authorized to perform a transaction on the account of the card used as a reference, before allowing the card personalization. The card issuer also ensures that the robustness of the personalization data, to prevent attacks like forgery, counterfeit or corruption.
OE.Card_Delete	The issuers of all payment cards provisioned on a device are informed after the User removes a card from that device, removes that device from the iCloud account, or performs a device "Erase All Content and Settings". The card issuers ensure these cards are removed from the User's account. (i.e. the unlinking process of the DPAN from the FPAN, which is done by the card issuer or the corresponding TSP). The Secure Element is responsible for securely deleting the stored card sensitive data (private/secret).
OE.Anti_Replay	The Apple Pay server verifies that each payment (e-commerce Apple Pay transaction) is not replayed. The payment is invalidated if this verification fails.
OE.Transaction_Verification	For Apple Pay, the cryptogram released by the Secure Element for an Apple Pay transaction is verified by the card issuer (or its service provider). The cryptogram validation result allows the card issuer to approve or reject the transaction. The payment is invalidated if this verification fails.
OE.Dynamic_Linking	For e-commerce transactions, the Apple Pay server preserves and the card issuer server verifies the cryptographic based dynamic linking of the transaction data (including amount and payee). The payment is invalidated if this verification fails.

Environment Security Objectives	Definition
OE.Statement	The card issuers ensure that the statement associated to the card (list of transactions) is fully accurate and includes, but is not restricted to, the amount and recipient of each transaction.
OE.Genuine_Wallet	The Wallet application is provided and signed by Apple.
OE.Watch	The S.Apple_Watch is responsible for ensuring the confidentiality of the unlock secret provided by the Mac device during all its lifetime: from in reception at enabling of the “Unlock with Apple Watch” feature, during its storage, during its release for unlocking the Mac device, and when it is deleted when the feature is disabled. The S.Apple_Watch is responsible for ensuring that it is protected by a password and the wrist detection feature is turned on in order to enable the feature “Unlock Mac with Apple Watch”.
OE.User	The S.User is responsible for ensuring that: • Other users of the device with root access are trusted and competent to prevent inadvertent malware installation • They authorize all the Apple Pay activities that are performed on the device • The password is robust and protected • Only their own biometrics credentials are enrolled (they do not enroll biometrics of someone else) • The Mac mini is paired with the Magic keyboard with Touch ID • Only their own Apple Watch is paired with the TOE and the paired Apple Watch is protected. This includes abiding by the watchOS Software License Agreement and protecting the confidentiality of the paired Apple Watch’s passcode

5.3. Rationale of the Security objectives for the security problem definition

The following table details the rationale for each element of the security problem definition. For all the objectives for the TOE, OT.OS_Update also ensures the authentication configuration is preserved during the OS/Mk firmware update.

Element	Rationale
A.DEVICE_AUTH	OE.User and OE.Watch cover A.DEVICE_AUTH ensuring the User owns and protects all the authentication credentials and the User authorizes the Apple Pay activities that are performed on the device.
A.PERSO	OE.Perso covers A.PERSO ensuring the provisioning process verifies the ownership of the provisioned cards.
A.WATCH_USER	OE.User and OE.Watch covers A.WATCH_USER ensuring the User of the device owns and protects their Apple Watch and related credentials.
A.NO_EVIL_ROOT_USER	OE.User ensures other users of the device with root access are trusted and competent to prevent inadvertent malware installation.

Element	Rationale
T.SKIMMING	OT.User_Auth, OT.Auth_SE and OT.Payment ensure that an Apple Pay transaction is always authenticated and authorized by the User. OT.User_Auth and OE.Genuine_Wallet ensure that the Apple release of a new OS or Apple Wallet application implementing the authentication functions and payment functions are controlled, and that it preserves the confidentiality and integrity of the authentication and payment data. OT.Password_Delete ensures that if the password is turned off, the Apple Pay and Biometric authentication are not available anymore.
T.USURP	OT.Card_Data, OT.OS_Update and OE.Card_Data ensure that the Apple Pay card data is kept confidential and not altered from an attacker during storage and use in the Secure Element. OT.User_Auth, OT.Auth_SE and OT.Payment prevent the attacker from attempting to perform Apple Pay transactions, enforcing user authentication with set credentials (which are not known or owned by the attacker according to OE.User) and preventing the attacker from replacing a User's password or biometrics with their own. OT.User_Auth and OE.Genuine_Wallet additionally ensure that the installation of a new Apple released OS or Apple Wallet application preserves the confidentiality and integrity of the payment data. OT.Password_Delete ensures that if password login is disabled, the Apple Pay and Biometric authentication are not available anymore.
T.RECOVER	OT.Bio_Delete ensures that a removed password or biometric credential cannot be recovered. OT.Card_Data, OT.OS_Update and OE.Card_Data ensure that the confidential card data is only stored by the Secure Element which protects them from disclosure. OT.Disk_Erase covers the physical erase of the content and settings of the device where the TOE will trigger secure erase all provisioned card data. OE.Card_Delete ensures that the card issuers of provisioned cards are securely removing the deleted cards from the User's account so that no transaction can further proceed.
T.REPLAY	OE.Anti_Replay and OT.Anti_Replay ensure that each Apple Pay transaction cannot be replayed. OT.User_Auth and OE.Genuine_Wallet additionally ensure that the installation of a new Apple released OS or Apple Wallet application preserves transaction replay protection.
T.CORRUPT	OT.Payment enforces the dynamic linking of the Apple Pay transaction data, ensuring that the critical content (such as emitter, recipient, amount) cannot be changed after the transaction was processed using a provisioned card. OE.Dynamic_Linking ensures that the Apple Pay server is verifying the integrity of the Apple Pay transaction data Dynamic Linking. OE.Statement provides an additional verification point for the account holder as the card issuer ensures that all processed Apple Pay transactions appear on the statement of the account associated to the card. OT.User_Auth and OE.Genuine_Wallet ensure that the Apple release of a new OS or Apple Wallet application implementing the payment functions are controlled.

Element	Rationale
T.SILENT	OT.User_Auth and OT.Payment ensure that an Apple Pay transaction is always authorized by the User. OE.Statement ensures that the Apple Pay card issuers provide account holder verification material (in the form of transaction statements) allowing them to identify any fraudulent activity on their account. OT.User_Auth and OE.Genuine_Wallet ensure that the Apple release of a new OS or Apple Wallet application implementing the authentication functions and payment functions is controlled. OT.Password_Delete ensures that if password is turned off, the Apple Pay and Biometric authentication are not available anymore.
T.PHYSICAL	The User credentials maintained by the TOE are secured by OT.User_Auth enforcing the secure verification process of biometrics or password. The TOE lifecycle is secure through the management of the device within the Apple iCloud environment where the User is able to remove the device from its account (OT.Card_Delete) and erase the disk content (OT.Disk_Erase). This binding ensures that the User's critical data is safe in case device is lost or stolen. OT.Card_Data, OT.OS_Update and OE.Card_Data ensure that the provisioned card data is kept confidential in the Secure Element and cannot be extracted.
P.UPDATE	OT.User_Auth and OE.Genuine_Wallet ensure that the Apple release of a new OS or Apple Wallet application implementing the authentication functions and payment functions are controlled, and that it preserves the confidentiality and integrity of the authentication and payment data.
P.DYN_LINK **(Dynamic Linking)**	P.DYN_LINK is covered by OT.Payment, OT.User_Auth, OT.Anti_Replay, OE.Anti_Replay, OE.Transaction_Verification, and OE.Dynamic_Linking, which all participate in the enforcement of the Dynamic Linking requirements on e-commerce payments. The mapping is as follows: (a) OT.Payment ensures that there is a step, part of the user intent confirmation phase, when the User (payer) is made aware of the amount of the payment transaction and payee (b) OT.User_Auth ensures that the user authentication was performed to ensure agreement by the payer to authorize the Apple Pay transaction data (specific to the payer, amount and payee), OT.Anti_Replay ensures that each payment is uniquely identified, and OT.Payment ensures that the payment data is integrity protected (c) OE.Anti_Replay, OE.Transaction_Verification and OE.Dynamic_Linking ensure that the received Apple Pay transaction data correspond to what was agreed to by the payer: The unique identifier to prevent replay is verified, the cryptogram for the data is verified, and the dynamic linking of the user authentication and the payment data integrity is verified (d) OE.Anti_Replay, OE.Transaction_Verification and OE.Dynamic_Linking ensure that any change to the amount or the payee results in the invalidation of the payment and its unique identifier so no replay is attempted.
P.WATCH	P.WATCH is covered by OT.User_Auth, OT.Watch, OE.Watch, and OE.User, which ensure that only the user's Apple Watch can be used for the dedicated Apple Watch features.

Security Objectives mapping table:

	T.CORRUPT	T.PHYSICAL	T.RECOVER	T.REPLAY	T.SILENT	T.SKIMMING	T.USURP	P.DYN_LINK	P.UPDATE	P.WATCH	A.PERSO	A.DEVICE_AUTH	A.WATCH_USER	A.NO_EVIL_ROOT_USER
OT.Anti_Replay				x				x						
OT.Card_Data		x	x				x							
OT.Payment	x				x	x	x	x						
OT.Card_Delete		x												
OT.OS_Update		x	x				x							
OT.Auth_SE						x	x							
OT.User_Auth	x	x		x	x	x	x	x	x	x				
OT.Watch										x				
OT.Password_Delete					x	x	x							
OT.Bio_Delete			x											
OT.Disk_Erase		x	x											
OE.Anti_Replay				x				x						
OE.Card_Data		x	x				x							
OE.Perso											x			
OE.Dynamic_Linking	x							x						
OE.Statement	x				x									
OE.Transaction_Verification								x						
OE.Watch										x		x	x	
OE.User							x			x		x	x	x
OE.Genuine_Wallet	x			x	x	x	x		x					
OE.Card_Delete			x											

6. Security Functional Requirements

6.1. SFR supporting definitions

6.1.1. Security Functional Policies (SFP)

Access Control SFPs are given in the table below:

Authentication_SFP	Authentication policy enforcing authentication, re-authentication and authorization rules as defined by OT.User_Auth. This SFP includes information flows between the Secure Enclave and the biometric sensor (for biometric authentication), between the TOE and S.Apple_Watch and between the Magic Keyboard and the Mac mini.
Payment_SFP	Security policy enforcing that processing a payment requires the User to confirm the intent to pay and being re-authenticated (password, or, if configured, biometrics) to allow processing of the related data and its exportation.
Card_Perso_SFP	Security policy enforcing that: • Importing D.Card_Data is only allowed if: a password is configured • Confidential parts of the card data are protected before being sent to the Apple servers • Confidential parts of the card data are not imported in Apple Wallet

6.1.2. Subjects and Objects

Objects are the Assets identified in the section 4.1.
Subjects are listed in the sections 4.2 and 4.4

6.1.3. Security Attributes

Security Attribute		
Card Data Confidential parts	Parts of the Card Data that should stay confidential and not been stored in the Wallet	- Secret parts of the card number - CVV - …
User Authorization	Part of D.Payment_Data specifying the explicit authorization of the S.User	- "yes" - "no"
BioAuth Unlock	Part of D.User_Configuration, authorization to use biometric authentication for unlocking a locked device, "selected" or "not selected" by the User.	- "selected" - "not selected" (default)
BioAuth AP	Part of D.User_Configuration, authorization to use biometric authentication for Apple Pay operations, "selected" or "not selected" by the User.	- "selected" (default) - "not selected"
Watch Unlock	Authorization to use the paired Apple Watch to unlock the TOE.	- "enabled" - "disabled" (default)
Apple OS public key	The public key used to check the authenticity of a new version of D.OS.	- Part of installed D.OS

Security Attribute		
OS_signature	Part of the OS/MK firmware file that is checked by the TOE before updating D.OS	- Part of the update file
Password_off	Part of D.SEP_Configuration, configuration of the OS allowing to use the device without any authentication. When enabled, the TSF should disable biometric authentication and Apple Pay.	- “disabled” (default) - “enable”

6.1.4. Writing conventions for the SFR operations

Iterations are identified by a slash character “/” followed by the name of the iteration.
Assignments and selections are done with *italicized text*.
Refinements are identified with the prefix “Refinement:” just before.

6.2. Identification and authentication

6.2.1. User authentication

FIA_UID.2 User identification before any action

FIA_UID.2.1	The TSF shall require each user to be successfully identified before allowing any other TSF-mediated actions on behalf of that user.

FIA_UAU.2 User authentication before any action

FIA_UAU.2.1	The TSF shall require each user to be successfully authenticated before allowing any other TSF-mediated actions on behalf of that user.

Note: this gives S.User the role of “Authenticated User”. According to this requirement, S.User is authenticated by the TSF before performing any of the operations listed in the following requirements.

FIA_UAU.5 Multiple authentication mechanisms

FIA_UAU.5.1	The TSF shall provide *password authentication, biometric authentication (fingerprint), Watch unlock (paired Apple Watch)* to support user authentication.
FIA_UAU.5.2	The TSF shall authenticate any user's claimed identity according to the: • *Password authentication as default authentication* • *Biometric authentication for:* - *Unlock if selected in the System Settings (BioAuth Unlock = selected)* - *Transaction authorization if selected in the System Settings (BioAuth AP = selected)* • *Watch unlock (if Watch Unlock = enabled) for:* - *Unlock the TOE* • *Rules defined in FIA_UAU.6 Re-authenticating and FIA_AFL.1 Authentication failure handling*.

Note: The TOE allows other authorizations not related to Apple Pay to be performed from optional paired Apple Watch.

FIA_AFL.1/Biometric Authentication failure handling

FIA_AFL.1.1 /Biometric	The TSF shall detect when 5 *(five)* unsuccessful authentication attempts occur related to *Biometric validation*.
FIA_AFL.1.2 /Biometric	When the defined number of unsuccessful authentication attempts has been *met*, the TSF shall *require Password validation, blocking further Biometric validation attempts*.

FIA_AFL.1/Recovery Authentication failure handling

FIA_AFL.1.1 /Recovery	The TSF shall detect when 10 *(ten)* unsuccessful authentication attempts occur related to *first password validation after boot or reboot*.
FIA_AFL.1.2 /Recovery	When the defined number of unsuccessful authentication attempts has been *met*, the TSF shall *block further password validation attempts until the user boots into recoveryOS*

FIA_AFL.1/Delay Authentication failure handling

FIA_AFL.1.1 /Delay	The TSF shall detect when *4 (four)* unsuccessful authentication attempts occur related to *first password validation after boot or reboot.*
FIA_AFL.1.2 /Delay	When the defined number of unsuccessful authentication attempts has been *met*, the TSF shall *start delaying further password validation attempts and require password validation.*

FIA_UAU.6 Re-authenticating

FIA_UAU.6.1	The TSF shall re-authenticate the user under the conditions *that the user requests:* - *OS update (change of D.OS)* - *Erasing the device* - *"Touch ID & Password" configuration change in System Settings (once for all "Touch ID & Password" parameters until "Touch ID & Password" interface is closed), including "BioAuth Unlock" enablement, "BioAuth AP" enablement, and changes to the password and Biometric patterns)* - *Unlocking the Users & Groups pane in System Settings* - *Transaction validation (export of D.Payment_Data), the re-authentication should be done during the 60 second after the transaction validation request.*

6.2.2. Data Authentication

FDP_DAU.1 Basic Data Authentication

FDP_DAU.1.1	The TSF shall provide a capability to generate evidence that can be used as a guarantee of the validity of *D.Payment_Data (including S.Merchant and S.User data)*.
FDP_DAU.1.2	The TSF shall provide *S.SE* with the ability to verify evidence of the validity of the indicated information.

6.2.3. User attribute definition

FIA_ATD.1 User attribute definition

FIA_ATD.1.1	The TSF shall maintain the following list of security attributes belonging to individual users: *D.User_Password, D.User_Bio, D.Card_Data, BioAuth Unlock, BioAuth AP, Watch Unlock.*
Application Note: The update of D.OS shall not modify these user attributes.	

6.2.4. Specification of secrets

FIA_SOS.2 TSF Generation of secrets

FIA_SOS.2.1	The TSF shall provide a mechanism to generate secrets that meet *FIPS 140-2 validated cryptographic module.*
FIA_SOS.2.2	The TSF shall be able to enforce the use of TSF generated secrets for *creation of the D.Unlock_Secret during the Unlock with Apple Watch Setup.*

6.3. Access/Flow Control SFRs

6.3.1. Authentication_SFP

FDP_ACC.2/Authentication_SFP Complete access control

FDP_ACC.2.1/ Authentication_SFP	The TSF shall enforce the *Authentication_SFP* on: Subjects: *S.User, S.Apple_Watch* Objects: *the TSF* and all operations among subjects and objects covered by the SFP.
FDP_ACC.2.2/ Authentication_SFP	The TSF shall ensure that all operations between any subject controlled by the TSF and any object controlled by the TSF are covered by an access control SFP.

FDP_ACF.1/Authentication_SFP Security attribute-based access control

FDP_ACF.1.1/ Authentication_SFP	The TSF shall enforce the *Authentication_SFP* to objects based on the following: Subjects: *S.User, S.Apple_Watch* Objects: *the TSF* Security attributes: *"Watch Unlock"*
FDP_ACF.1.2/ Authentication_SFP	The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed: *S.User is authenticated according to FIA_UAU.5*
FDP_ACF.1.3/ Authentication_SFP	The TSF shall explicitly authorise access of subjects to objects based on the following additional rules: *If "Watch Unlock" set to "enabled", S.Apple_Watch gives S.User the capability to:* - *Unlock the TOE* - *Approve other requests (related to the security functions of the TOE) to enter their administrator password.*
FDP_ACF.1.4/ Authentication_SFP	The TSF shall explicitly deny access of subjects to objects based on the following additional rules: *"Password_off" is enabled.*

6.3.2.Payment_SFP

FDP_ETC.2/Transaction Export of user data with security attributes

FDP_ETC.2.1 /Transaction	The TSF shall enforce the *Payment_SFP* when exporting user data, controlled under the SFP(s), outside of the TOE.
FDP_ETC.2.2 /Transaction	The TSF shall export the user data with the user data's associated security attributes.
FDP_ETC.2.3 /Transaction	The TSF shall ensure that the security attributes, when exported outside the TOE, are unambiguously associated with the exported user data.
FDP_ETC.2.4 /Transaction	The TSF shall enforce the following rules when user data is exported from the TOE: - *exported Payment_SFP details (the card to be used from the Wallet, the amount and the payee in the case of e-commerce payments) are those displayed to S.User during re-authentication request (FIA_UAU.6 Re-authenticating)* - *D.Payment_Data includes a unique identifier (An Apple Pay server nonce).*

FDP_ACC.2/Payment_SFP Complete access control

<table>
<tr><td>FDP_ACC.2.1/ Payment_SFP</td><td>The TSF shall enforce the <i>Payment_SFP</i> on<i>:</i>
<table>
<tr><td>Subjects:</td><td><i>S.User</i></td></tr>
<tr><td>Objects:</td><td><i>D.Payment_Data (including S.User and S.Merchant Data)</i></td></tr>
</table>
and all operations among subjects and objects covered by the SFP.</td></tr>
<tr><td>FDP_ACC.2.2/ Payment_SFP</td><td>The TSF shall ensure that all operations between any subject controlled by the TSF and any object controlled by the TSF are covered by an access control SFP.</td></tr>
</table>

Note: the only possible operation on D.Payment_Data is to export it as a transaction order to the Secure Element.

FDP_ACF.1/Payment_SFP Security attribute based access control

<table>
<tr><td>FDP_ACF.1.1/ Payment_SFP</td><td>The TSF shall enforce the <i>Payment_SFP</i> to objects based on the following:
<table>
<tr><td>Subjects:</td><td><i>S.User</i></td></tr>
<tr><td>Objects:</td><td><i>D.Payment_Data (including S.User and S.Merchant Data),
the Secure Element (through a trusted channel)</i></td></tr>
<tr><td>Security attributes:</td><td><i>“User Authorization”, “BioAuth AP”, “Password_off”</i></td></tr>
</table></td></tr>
<tr><td>FDP_ACF.1.2/ Payment_SFP</td><td>The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed:
- <i>Export of D.Payment_Data with “User Authorization” set to “yes” to the Secure Element is allowed if:</i>
- <i>S.User shows their intent to pay (using the gesture of activating the Touch ID sensor combined with successfully matching the user’s fingerprint or double pressing of touch ID sensor when user doesn't have touch ID set up)</i>
- <i>S.User had been successfully re-authenticated for transaction validation (FIA_UAU.6).</i></td></tr>
<tr><td>FDP_ACF.1.3/ Payment_SFP</td><td>The TSF shall explicitly authorise access of subjects to objects based on the following additional rules: <i>none</i>.</td></tr>
<tr><td>FDP_ACF.1.4/ Payment_SFP</td><td>The TSF shall explicitly deny access of subjects to objects based on the following additional rules: <i>“Password_off” is “enabled”</i>.</td></tr>
</table>

6.3.3.Card_Perso_SFP

FDP_ACC.2/Card_Perso_SFP Complete access control

FDP_ACC.2.1 /Card_Perso_SFP	The TSF shall enforce the *Card_Perso_SFP* on Subjects: *S.User, S.Apple_Servers* Objects: *D.Card_Data* and all operations among subjects and objects covered by the SFP.
FDP_ACC.2.2 /Card_Perso_SFP	The TSF shall ensure that all operations between any subject controlled by the TSF and any object controlled by the TSF are covered by an access control SFP.

FDP_ACF.1/Card_Perso_SFP Security attribute based access control

FDP_ACF.1.1 /Card_Perso_SFP	The TSF shall enforce the *Card_Perso_SFP* to objects based on the following*:* Subjects: *S.User, S.Apple_Servers* Objects: *D.Card_Data* Security attributes: *Password_off*
FDP_ACF.1.2 /Card_Perso_SFP	The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed: - *S.User is connected to its iCloud account, and is authenticated on the TOE*.
FDP_ACF.1.3 /Card_Perso_SFP	The TSF shall explicitly authorise access of subjects to objects based on the following additional rules: *none*.
FDP_ACF.1.4 /Card_Perso_SFP	The TSF shall explicitly deny access of subjects to objects based on the following additional rules: *"Password_off" is enabled.*

FDP_ETC.2/Card_Perso_SFP Export of user data with security attributes

FDP_ETC.2.1 /Card_Perso_SFP	The TSF shall enforce the *Card_Perso_SFP* when exporting user data, controlled under the SFP(s), outside of the TOE.
FDP_ETC.2.2 /Card_Perso_SFP	The TSF shall export the user data with the user data's associated security attributes.
FDP_ETC.2.3 /Card_Perso_SFP	The TSF shall ensure that the security attributes, when exported outside the TOE, are unambiguously associated with the exported user data.
FDP_ETC.2.4 /Card_Perso_SFP	The TSF shall enforce the following rules when user data is exported from the TOE: - *D.Card_Data is encrypted before being exported to S.Apple_Servers* - *"Card Data Confidential parts" are not kept on the TOE after being exported.*

FPT_ITC.1 Inter-TSF confidentiality during transmission

FPT_ITC.1.1	The TSF shall protect all TSF data transmitted from the TSF to another trusted IT product from unauthorised disclosure during transmission.

FDP_ITC.1 Import of user data without security attributes

FDP_ITC.1.1	The TSF shall enforce the *Card_Perso_SFP* when importing user data, controlled under the SFP, from outside of the TOE.
FDP_ITC.1.2	The TSF shall ignore any security attributes associated with the user data when imported from outside the TOE.
FDP_ITC.1.3	The TSF shall enforce the following rules when importing user data controlled under the SFP from outside the TOE: *None*.

Note: security attributes of the Card Data are "Card Data Confidential parts" in section 6.1.3. These requirements specify that the confidential part of the cards data is used for card enrollment (FDP_ETC.2/Card_Perso_SFP) but are not stored on the TOE.

6.4. Magic Keyboard Trusted Channels

FDP_ITT.1/HID Basic internal transfer protection

FDP_ITT.1.1	The TSF shall enforce the *Authentication_SFP,* to prevent the *modification and disclosure* of user data when it is transmitted between physically-separated parts of the TOE.

Note: This requirement concerns the protection of human interface device data (key pressing and releasing including D.User_Password) sent by the Magic Keyboard to the Mac mini. Protection against modification also includes protection against replay.

FDP_ITT.1/Bio Basic internal transfer protection

FDP_ITT.1.1	The TSF shall enforce the *Authentication_SFP* to prevent the *modification and disclosure* of user data when it is transmitted between physically-separated parts of the TOE.

Note: This requirement concerns the protection of biometric data sent by the biometric sensor to the Secure Enclave. Protection against modification also includes protection against replay.

6.5. Secure Enclave/Secure Element Trusted Channel

FTP_ITC.1/SE Inter-TSF trusted channel

FTP_ITC.1.1/SE	The TSF shall provide a communication channel between itself and another trusted IT product that is logically distinct from other communication channels and provides assured identification of its end points and protection of the channel data from modification or disclosure.
FTP_ITC.1.2/SE	The TSF shall permit *the TSF* to initiate communication via the trusted channel.
FTP_ITC.1.3/SE	The TSF shall initiate communication via the trusted channel for *Payment initiation and transmission of D.Payment_Data*.

FDP_UCT.1/SE Basic data exchange confidentiality

FDP_UCT.1.1/SE	The TSF shall enforce the *Payment_SFP,* to *transmit* user data in a manner protected from unauthorised disclosure.

FDP_UIT.1/SE Data exchange integrity

FDP_UIT.1.1/SE	The TSF shall enforce the *Payment_SFP*, to *transmit and receive* user data in a manner protected from *modification, insertion and replay* errors.
FDP_UIT.1.2/SE	The TSF shall be able to determine on receipt of user data, whether *modification, insertion or replay* has occurred.

FPT_RPL.1/SE Replay detection

FPT_RPL.1.1/SE	The TSF shall detect replay for the following entities: *S.SE*.
FPT_RPL.1.2/SE	The TSF shall perform *reject data* when replay is detected.

6.6. Secure Enclave/Apple Watch Trusted Channel

FTP_ITC.1/Watch Inter-TSF trusted channel

FTP_ITC.1.1 /Watch	The TSF shall provide a communication channel between itself and another trusted IT product that is logically distinct from other communication channels and provides assured identification of its end points and protection of the channel data from modification or disclosure.
FTP_ITC.1.2 /Watch	The TSF shall permit *the TSF* to initiate communication via the trusted channel.
FTP_ITC.1.3 /Watch	The TSF shall initiate communication via the trusted channel for: *unlock the TOE*.

FDP_UCT.1/Watch Basic data exchange confidentiality

FDP_UCT.1.1 /Watch	The TSF shall enforce the *Authentication_SFP*, to *transmit and receive* user data in a manner protected from unauthorised disclosure.

FDP_UIT.1/Watch Data exchange integrity

FDP_UIT.1.1 /Watch	The TSF shall enforce the *Authentication_SFP* to *transmit and receive* user data in a manner protected from *modification, insertion and replay* errors.
FDP_UIT.1.2 /Watch	The TSF shall be able to determine on receipt of user data, whether *modification, insertion or replay* has occurred.

FPT_RPL.1/Watch Replay detection

FPT_RPL.1.1/Watch	The TSF shall detect replay for the following entities: *S.Apple_Watch*.
FPT_RPL.1.2/Watch	The TSF shall perform *reject data* when replay is detected.

6.7. Local data protection

FPR_UNO.1 Unobservability

FPR_UNO.1.1	The TSF shall ensure that *S.Attacker* are unable to observe the operation *Password set, Password check, Password Update, Password removal, Biometrics set, Biometrics check, Biometrics update, Biometrics delete, Apple Pay provisioning, Apple Watch Unlock activation, Magic Keyboard Pairing on D.User_Bio, D.User_Password, D.Card_Data, D.SEP_Watch, D.Unlock_Secret, D.SEP_Bio, D.Keyboard_Secret* by *S.User*.

FDP_RIP.1 Subset residual information protection

<table>
<tr><td>FDP_RIP.1.1</td><td>The TSF shall ensure that any previous information content of a resource is made unavailable upon the <i>deallocation of the resource from</i> the following objects: <i>D.User_Bio, D.User_Password, “Card Data Confidential parts” and iCloud Account (in D.User_Configuration)</i>.</td></tr>
<tr><td colspan="2"><i>Application Note</i>:
- The removal of the iCloud Account by the S.User triggers the deallocation of all the Apple Pay data/configuration.
- The revocation of individual card by its issuer triggers the deallocation of the related card data.
- The procedure of erase on the device triggers the deallocation of all security attributes except the D.OS.
- The removal of “Card Data Confidential parts” is done by instructing the Secure Element to mark the card as deleted.
- The removal of the password by the S.User disables the actual password; and triggers the deallocation of all the Apple Pay data.</td></tr>
</table>

FDP_SDI.1 Stored data integrity monitoring

FDP_SDI.1.1	The TSF shall monitor user data stored in containers controlled by the TSF for *integrity errors* on all objects, based on the following attributes: *D.User_Bio, D.Card_Data, D.OS*.

6.8. TSF management

6.8.1. Roles and Management Functions

FMT_SMR.1 Security roles

FMT_SMR.1.1	The TSF shall maintain the roles *Authenticated User*.
FMT_SMR.1.2	The TSF shall be able to associate users with roles.

FMT_SMF.1 Specification of Management Functions

FMT_SMF.1.1	The TSF shall be capable of performing the following management functions: *management of security attributes (see FMT_MSA.1)*.

6.8.2. Management of security attributes

FMT_MSA.3 Static attribute initialization

<table>
<tr><td>FMT_MSA.3.1</td><td>The TSF shall enforce the <i>Payment_SFP, Card_Perso_SFP</i> to provide
<table>
<tr><td><b><i>“BioAuth Unlock”</i></b></td><td>restrictive (disabled)</td></tr>
<tr><td><b><i>“BioAuth AP”</i></b></td><td>permissive (selected)</td></tr>
<tr><td><b><i>“Password_off”</i></b></td><td>restrictive (disabled)</td></tr>
<tr><td><b><i>“Watch Unlock”</i></b></td><td>restrictive (disabled)</td></tr>
</table>
default values for security attributes that are used to enforce the SFP.</td></tr>
<tr><td>FMT_MSA.3.2</td><td>The TSF shall allow <i>nobody</i> to specify alternative initial values to override the default values when an object or information is created.</td></tr>
</table>

FMT_MSA.1 Management of security attributes

FMT_MSA.1.1	The TSF shall enforce the *Authentication_SFP, Card_Perso_SFP* to restrict the ability to *modify* the security attributes *“, “BioAuth Unlock”, “BioAuth AP”, “Password_off”, “Watch Unlock”* to *“Authenticated User”*.

Application Note:

- When “Password_off” is enabled, the TSF removes the D.Card_Data according to FDP_RIP.1.

6.8.3.Management of TSF Data

FMT_MTD.1 Management of TSF data

FMT_MTD.1.1	The TSF shall restrict the ability to *update* the *D.OS* to *“Authenticated user”*.

FMT_MTD.3 Secure TSF data

FMT_MTD.3.1	The TSF shall ensure that only secure values are accepted for *D.OS*.

Application Note: secure value is defined by “OS_signature” is valid and signed by “Apple OS public key”.

6.9. Security Requirements Rationale

6.9.1. Security Functional Requirements (SFR) Dependencies

TOE SFR	Required dependencies	Covered by
FIA_UAU.2	FIA_UID.1	FIA_UID.2
FIA_AFL.1(Biometric/Recovery/Delay)	FIA_UAU.1	FIA_UAU.2
FDP_ACC.2/Authentication_SFP	FDP_ACF.1	FDP_ACF.1/Authentication_SFP
FDP_ACF.1/Authentication_SFP	FDP_ACC.1 FMT_MSA.3	FDP_ACC.2/Authentication_SFP FMT_MSA.3
FDP_ACC.2/Payment_SFP	FDP_ACF.1	FDP_ACF.1/Payment_SFP
FDP_ACF.1/Payment_SFP	FDP_ACC.1 FMT_MSA.3	FDP_ACC.2/Payment_SFP FMT_MSA.3
FDP_ITT.1/HID	FDP_ACC.1, or FDP_IFC.1	FDP_ACC.2/Authentication_SFP
FDP_ITT.1/BIO	FDP_ACC.1, or FDP_IFC.1	FDP_ACC.2/Authentication_SFP
FDP_ETC.2/Transaction	FDP_ACC.1, or FDP_IFC.1	FDP_ACC.2/Payment_SFP
FDP_ACC.2/Card_Perso_SFP	FDP_ACF.1	FDP_ACF.1/Card_Perso_SFP
FDP_ACF.1/Card_Perso_SFP	FDP_ACC.1 FMT_MSA.3	FDP_ACC.2/Card_Perso_SFP FMT_MSA.3
FDP_ITC.1	FDP_ACC.1, or FDP_IFC.1 FMT_MSA.3	FDP_ACC.2/Card_Perso_SFP FMT_MSA.3
FDP_ETC.2/Card_Perso_SFP	FDP_ACC.1, or FDP_IFC.1	FDP_ACC.2/Card_Perso_SFP
FDP_UCT.1/SE	FDP_ACC.1, or FDP_IFC.1 FTP_ITC.1, or FTP_TRP.1	FDP_ACF.1/Payment_SFP, and FDP_ACF.1/Card_Perso_SFP FTP_ITC.1/SE
FDP_UIT.1/SE	FDP_ACC.1, or FDP_IFC.1 FTP_ITC.1, or FTP_TRP.1	FDP_ACF.1/Payment_SFP, and FDP_ACF.1/Card_Perso_SFP FTP_ITC.1/SE
FDP_UCT.1/Watch	FDP_ACC.1, or FDP_IFC.1 FTP_ITC.1, or FTP_TRP.1	FDP_ACF.1/Authentication_SFP FTP_ITC.1/Watch
FDP_UIT.1/Watch	FDP_ACC.1, or FDP_IFC.1 FTP_ITC.1, or FTP_TRP.1	FDP_ACF.1/Authentication_SFP FTP_ITC.1/Watch
FMT_SMR.1	FIA_UID.1	FIA_UID.2
FMT_MSA.3	FMT_MSA.1, FMT_SMR.1	FMT_MSA.1, FMT_SMR.1
FMT_MSA.1	FDP_ACC.1, or FDP_IFC.1 FMT_SMR.1, SMT_SMF.1	FDP_ACC.2/Authentication_SFP, FDP_ACC.2/Authentication_SFP, and FDP_ACC.2/Card_Perso_SFP FMT_SMR.1, FMT_SMF.1
FMT_MTD.1	FMT_SMR.1, SMT_SMF.1	FMT_SMR.1, SMT_SMF.1
FMT_MTD.3	FMT_MTD.1	FMT_MTD.1

Requirements without dependency: FIA_UID.2, FIA_SOS.2, FIA_UAU.6, FIA_UAU.5, FDP_DAU.1, FIA_ATD.1, FPT_ITC.1, FTP_ITC.1/SE, FPT_RPL.1/SE, FTP_ITC.1/Watch, FPT_RPL.1/Watch, FPR_UNO.1, FDP_RIP.1, FMT_SMF.1 and FDP_SDI.1

6.9.2.Rationale SFR/Security Objectives for the TOE

Objective reference	Rationale
OT.User_Auth	FIA_UID.2 and FIA_UAU.2 enforce each user of the device is authenticated before being able to do any action in the user interface. FIA_UAU.5 specifies different authentication methods. FIA_UAU.6 enforces a re-authentication for the OS/MK firmware update, for changing the authentication configuration, including biometric data management and biometric authentication policy or for a transaction validation. FMT_SMR.1 and FMT_SMF.1 SF ensure that the TSF shall maintain the roles Authenticated User and be capable of performing the management functions. FIA_ATD.1, FMT_MSA.3, FMT_MSA.1 and FMT_MTD.1 specify the management of related information. FMT_MTD.3 specifies the signature verification on the OS/MK firmware update. FIA_AFL.1/Biometric, FIA_AFL.1/Recovery, and FIA_AFL.1/Delay specify the failure handling in case of wrong biometric or password authentication. FPR_UNO.1 ensures the non-observability of the password. FDP_ITT.1/BIO and FDP_ITT.1/HID also support this objective ensuring the biometric data is not modified or disclosed during internal transfer.
OT.Card_Data	The SFP Card_Perso_SFP and the associated SFRs (FDP_ACC.2/Card_Perso_SFP, FDP_ACF.1/Card_Perso_SFP, FDP_ITC.1 and FDP_ETC.2/Card_Perso_SFP, FPT_ITC.1) enforce card data stored in Apple Wallet does not include sensitive card data as this one is only sent to the Secure Element. Also, FTP_ITC.1/SE, FDP_UCT.1/SE and FDP_UIT.1/SE enforce that all the data exchanged between the TSF and S.SE is protected (with their corresponding security property) by a trusted channel. FPR_UNO.1 ensures the non-observability of secret card data. FDP_SDI.1 ensures card data stored in containers controlled by the TSF are monitored for integrity errors.
OT.Password_Delete	FDP_ACC.2/Card_Perso_SFP and FDP_ACF.1.4/Card_Perso_SFP enforce the card enrollment is not possible if Password_off is enabled. FIA_UAU.6 enforces a re-authentication for changing the Password_off configuration. FPR_UNO.1 ensures the non-observability of the password.
OT.Card_Delete	FDP_RIP.1 ensures the confidential parts of the card data are securely removed by the Secure Element. FMT_MSA.1.1 ensures the secure delete in case of password turning off. FPR_UNO.1 ensures the non-observability of sensitive card data.
OT.Auth_SE	FIA_UID.2, FIA_UAU.2, FIA_UAU.5 specify the base of the authentication feature. FIA_AFL.1/Biometric enhance the biometric security limiting the authentication attempts before requiring password authentication. FIA_AFL.1/Recovery and FIA_AFL.1/Delay protect the TSF and the user data against brute force attacks. FIA_UAU.6 specifies the re-authentication for some functions of the TOE.

Objective reference	Rationale
OT.Payment	FDP_ETC.2.4/Transaction ensures the transaction details are displayed before a transaction is validated by the User. FIA_UAU.6 ensures each transaction is validated by a re-authentication. FDP_DAU.1 provides evidence of the validity of Apple Pay Transaction Data, verifiable by the card issuer. FDP_ACC.2/Payment_SFP and FDP_ACF.1/Payment_SFP ensure user authorization is done before each transaction. FPT_RPL.1/SE ensures transaction replay detection.
OT.Bio_Delete	FDP_RIP.1 ensures biometric data is erased securely.
OT.Disk_Erase	FDP_RIP.1 ensures all sensitive data is securely removed during a disk erase.
OT.Anti_Replay	FDP_ETC.2.4/Transaction ensures a unique identifier provided by the Apple server is included in the transaction data, avoiding any replay attack.
OT.OS_Update	FIA_UAU.6.1 ensures S.User is re-authenticated before the OS/MK firmware update proceeds. FIA_ATD.1.1 ensures that no user data with impact on the TSF behavior are modified during the OS/MK firmware update process. FDP_SDI.1 ensures D.OS stored in containers controlled by the TSF is monitored for integrity errors.
OT.Watch	FTP_ITC.1/Watch, FDP_UCT.1/Watch, FDP_UIT.1/Watch, FIA_SOS.2, FPR_UNO.1 and FPT_RPL.1/Watch enforce that all the data exchanged between the TSF and S.Apple_Watch is protected by a trusted channel regarding the confidentiality and the integrity. The protected operations are specified by FDP_ACC.2/Authentication_SFP and FDP_ACF.1/Authentication_SFP.

6.9.3.SAR Dependencies

TOE SAR	Dependencies
ADV_ARC.1	ADV_FSP.1, ADV_TDS.1
ADV_FSP.3	ADV_TDS.1
ADV_TDS.1	ADV_FSP.2
AGD_OPE.1	ADV_FSP.1
AGD_PRE.1	No dependencies
ALC_CMC.2	ALC_CMS.1
ALC_CMS.2	No dependencies
ALC_DEL.1	No dependencies
ALC_FLR.3	No dependencies
ASE_CCL.1	ASE_INT.1, ASE_ECD.1, ASE_REQ.1
ASE_ECD.1	No dependencies
ASE_INT.1	No dependencies
ASE_OBJ.2	ASE_SPD.1

TOE SAR	Dependencies
ASE_REQ.2	ASE_OBJ.2, ASE_ECD.1
ASE_SPD.1	No dependencies
ASE_TSS.1	ASE_INT.1, ASE_REQ.1, ADV_FSP.1
ATE_COV.1	ADV_FSP.2, ATE_FUN.1
ATE_FUN.1	ATE_COV.1
ATE_IND.2	ADV_FSP.2, AGD_OPE.1, AGD_PRE.1, ATE_COV.1, ATE_FUN.1
AVA_VAN.2	ADV_ARC.1, ADV_FSP.2, ADV_TDS.1, AGD_OPE.1, AGD_PRE.1

All the dependencies are covered.

6.9.4. SAR Rationale

For this evaluation, the SARs of EAL2 have been chosen as they provide assurance by a full security target and an analysis of the SFRs in that ST, using a functional and interface specification, guidance documentation and a basic description of the architecture of the TOE, to understand the security behavior. It was established that this level is appropriate for the model of attacker of Apple Pay.

ADV_FSP.3 augmentation has been chosen to enhance the level of information provided related to SFR-enforcing TSFIs and provide a complete summary of the TOE.

ALC_FLR.3 augmentation has been chosen to guarantee the security of the security functions in the scope of this security target during the maintenance of the TOE.

7. TOE Summary Specification

This section describes the security functions of the TOE covering the SFR of the previous chapter.

7.1. SF User authentication and management

When using the TOE, before being able to use Touch ID as a Biometrics authentication method, the device must be set up so that a password is required to unlock it.

Touch ID is Apple’s authentication system that enables the secure access to a Mac device equipped with the Touch ID sensor. This technology is based on the Touch ID sensor and the Secure Enclave.

When the Secure Enclave detects a successful match against the stored mathematical representation, the device unlocks without asking for the device password. Touch ID does not replace the password but provides easy access to the device within defined boundaries and time constraints. To use Touch ID, the device must be set up so that a password is required to unlock it.

The probability that a random person in the population could unlock a user’s Mac device paired with a Magic Keyboard is less than 1 in 50,000. This probability increases with multiple enrolled fingerprints (up to 1 in 10,000 with five fingerprints). For additional protection, Touch ID allows only five unsuccessful match attempts before a passcode or password is required to obtain access to the user’s device or account.

The password can always be used instead of Biometrics, and the password is required under the following circumstances:

- The device has just been turned on or restarted.
- The User has not unlocked their device for more than 48 hours.
- The User has not used their password to unlock their device for 156 hours (six and a half days), and the User has not used a biometric to unlock their device in 4 hours.
- The device has received a remote lock command.
- There were five unsuccessful biometric match attempts (though for usability, the device might offer entering a password instead of using biometrics after a smaller number of failures).

These features implement the requirements listed in §6.2 and in §6.3.1 for the authentication, in §6.7 for local data protection and in FMT_SMF.1, FMT_MSA.1, and FMT_MSA.3 for management.

Unlock with a paired Apple (requirements in §6.6 for secure communication with Apple Watch) Watch gives the user the capability to use their optional paired apple Watch in order to:

- Unlock the TOE
- Approve other requests (related to the security functions of the TOE) to enter their administrator password. These requests do not include user authentication for Apple Pay transactions.

The following table summarizes the functions supporting User authentication.

Function	Description
Password authentication	Password_Setup: Setup of the device’s password.
	Password_Update: Update of the device’s password.
	Password_Verify: Authentication of the User using their password.
	Password_Delete: Removal of the password
Biometric authentication	Bio_Enroll: Enrollment of biometrics credential (first, or additional).

	Bio_Update: Update of Biometrics.
	Bio_Verify: Authentication of the User using a Biometrics.
	Bio_Delete: Delete an enrolled Biometrics credential (one, or all).
Unlock with Apple Watch	Unlock_With_Watch_Setup: Enable the feature « Unlock with Apple Watch ».
	Unlock_With_Watch_Check: Secure process to import the Unlock Secret to the TOE’s Secure Enclave for unlocking the device and approvals prompts.

7.1.1. Password_Setup

The TOE offers a configuration setting where the User can set up a password that will be used to perform authentication in order to access restricted services on the device including: first unlock after power-on or reboot, Apple Pay transactions, and more. The TOE enforces strong access control in order to prevent access to these restricted services without authentication (FIA_UID.2).

Note: *The User has the possibility to use the password or any other activated (with enrolled templates) authentication method to perform unlock (beyond first unlock) or Apple Pay transactions.*

The TOE ensures the non-observability of the password during the setting process within the Secure Enclave (FPR_UNO.1).

7.1.2. Password_Verify

The TOE offers password entry to the User as part of the device unlock procedure, or during the User authorization step of an Apple Pay transaction. The User might have selected the default method as being biometrics, but password verification is always possible (as a fallback or by User choice). The TSFs ensure that password verification preserves the secrecy of the password value set by the User, which is expected in order to prevent an attacker from guessing the code by observing the verification process (FPR_UNO.1). The TOE ensures that the verification process would not validate a code that does not match the password set by the User and detect alterations to the configured value (FDP_SDI.1), preventing use of the User account (and related data) and forcing a device panic. To discourage brute-force password attacks, the TOE authentication failure policy is escalating time delays after the entry of an invalid password (FIA_AFL.1/Delay) or after 10 failed validation attempts, blocking further password validation attempts until the user boots into recoveryOS (FIA_AFL.1/Recovery).

7.1.3. Password_Update

The User can update the password value through the device’s settings menu. This functions as a verify operation, as knowledge of the previous password value is required to proceed to the setup step where the User types a new value and effectively updates the value in the Secure Enclave (FIA_UAU.2, FDP_ACC.2/Authentication_SFP and FDP_ACF.1/Authentication_SFP). The TOE ensures the non-observability of the old password during the verification as well as of the new password during the setting process (FPR_UNO.1). The TOE also enforces password alteration detection, preventing a corrupted password from being used to reset the authentication function (FDP_SDI.1).

7.1.4. Password_Delete

If the User wants to fully remove the use of the password on the device, and because this would not allow an adequate security level for the processing of the TSF, the actual password is disabled; and the iCloud Account information and all the Apple Pay data is deallocated. (FDP_RIP.1). To prevent misuse of the Password deletion function, the User is required to first verify the current password before proceeding (FIA_UAU.2, FDP_ACC.2/Authentication_SFP and FDP_ACF.1/Authentication_SFP), and the TOE protects the password secrecy during the verification process in case an attacker was to use this path for attempting an observation-based attack (FPR_UNO.1).

The Secure Element, in the TOE Environment, is responsible for securely deleting the Apple Pay card data during this process.

7.1.5.Bio_Enroll

The TOE offers biometrics authentication through the Touch ID service. To use Touch ID, the User must set up the device so that a password is required to unlock it, and password verification is required to be able to perform any modifications to the biometrics setting. Access to the fingerprint enrollment and the setting for enabling biometrics for Apple Pay, is gated by this password verification (FIA_UAU.2, FDP_ACC.2/Authentication_SFP and FDP_ACF.1/Authentication_SFP). The Secure Enclave inside the TOE ensures the non-observability of the new biometrics during the enrollment process (FPR_UNO.1). A User can enroll up to 5 fingerprints for Touch ID.

7.1.6.Bio_Update

Once a fingerprint is enrolled on a device, the User can update it within the settings of the Touch ID feature, for which access is gated by the password (FIA_UAU.2, FDP_ACC.2/Authentication_SFP and FDP_ACF.1/Authentication_SFP). After a successful Password verification, the User can reset Touch ID, deleting the associated templates, and re-enroll their fingerprint later. This procedure is effectively enforcing the same security principles as a deletion (Bio_Delete), and an enrollment (Bio_Enroll): the Secure Enclave ensures that Biometrics data integrity is preserved, its manipulation is protected from observation, and the deallocation prevents attackers from finding any residual information (FPR_UNO.1, FDP_SDI.1, and FDP_RIP.1).

7.1.7.SF.Bio_Verify

After the disk is erased and password is entered to allow the device's normal mode of operation, the biometrics authentication function is offered to the User as the default means for authentication means, if it is enabled, for device unlock, Apple Pay transactions. The User will present their biometric features to the device's biometric sensor, and the Secure Enclave will securely perform the verification of the submitted template to the enrolled biometric template(s). The Secure Enclave ensures the non-observability and integrity monitoring of the biometric templates during the verification process (FPR_UNO.1 and FDP_SDI.1). In case of verification failure, which means the TOE was not able to find a successful match, an authentication failure policy is enforced and password verification is required before the biometrics authentication function is enabled again (FIA_AFL.1/Biometric).

7.1.8.Bio_Delete

The User has the capability to delete all enrolled biometrics, from the Touch ID settings on the device, using the Reset Touch ID option. This function deletes the associated template of all the fingerprints enrolled. This function, like the others, is gated by the password (FIA_UAU.2, FDP_ACC.2/Authentication_SFP and FDP_ACF.1/Authentication SFP). The deletion process ensures that no residual information of the deallocated data is left behind or leaked to a potential observer (FDP_RIP.1, and FPR_UNO.1).

7.1.9. Unlock_With_Watch_Setup

Optionally, the User can decide to enable a feature called "Unlock with Apple Watch" so that the Mac device can be unlocked by the Watch. Enabling this feature requires the User to enter the password on the Mac device (FDP_ACC.2/Authentication_SFP, FDP_ACF.1/Authentication_SFP, FMT_MSA.1 and FMT_MSA.3) before the setup process is started.

During this process, the TOE Secure Enclave Processor creates a non-predictable Unlock Secret (FIA_SOS.2 and FPR_UNO.1).

The TOE's Secure Enclave securely exports the Unlock Secret to the paired Watch's Secure Enclave applying security measures to the sensitive asset with Secure Enclave/Watch Pairing Secret (FTP_ITC.1/Watch, FDP_UCT.1/Watch, FDP_UIT.1/Watch, FPT_RPL.1/Watch).

Until the feature is disabled, the TOE maintains the security attributes and assets of the Unlock with Apple Watch process, including the Unlock Secret (FIA_ATD.1).

7.1.10. Unlock_With_Watch_Check

Optionally, if the User enabled the feature called "Unlock with Apple Watch", the Mac device user can use its optional a paired watch to:

- Unlock the device
- Approve other requests (related to the security functions of the TOE) to enter their administrator password. These requests do not include user authentication for Apple Pay transactions.

During this process, the paired Watch's Secure Enclave Processor securely imports the Unlock Secret to the TOE's Secure Enclave, applying security measures to the sensitive asset with Secure Enclave Watch Pairing Secret (FTP_ITC.1/Watch, FDP_UCT.1/Watch, FDP_UIT.1/Watch, FPT_RPL.1/Watch).

7.2. SF Mac/Magic Keyboard secure channels

The TSF protects the data exchanged between the Magic Keyboard and the Mac mini with secure channels:

- The human interface device data (key pressing and releasing including D.User_Password) (FDP_ITT.1/HID) against disclosure and modification
- The biometric data against disclosure and modification (FDP_ITT.1/BIO).

The TSF also ensures that secrets used for the data protection are not disclosed (FPR_UNO.1).

7.3. SF Secure Enclave/Secure Element secure channel

The TSF is able to initialize a secure channel with the Secure Element (FTP_ITC.1/SE). This secure channel protects the exchanged data with its corresponding security properties: against disclosure (FDP_UCT.1/SE), modification (FDP_UIT.1/SE) and replay (FPT_RPL.1/SE).

7.4. SF Secure Enclave/Watch secure channel

The TSF is able to initialize a secure channel to S.Apple_Watch (FTP_ITC.1/Watch). This secure channel protects the exchanged data against disclosure (FDP_UCT.1/Watch), modification (FDP_UIT.1/Watch) and replay (FPT_RPL.1/Watch).

7.5. SF Card Data management

The following table summarizes the functions supporting Card Data management.

Function	Description
Apple Pay card data management	AP_Card_Provisioning: Provisioning of a new Card for Apple Pay.
	AP_Cancellation: User removes a card from Apple Wallet.
	AP_Revocation: Card issuer initiated suspend/unlink of a payment card in Apple Wallet.

7.5.1. AP_Card_Provisioning

On the TOE, there are different ways to add a card into Apple Wallet:

- Adding a card manually
- Adding cards on file from an iTunes Store account to Apple Pay
- Adding cards from a card issuer's website
- Adding cards that were provisioned on a different device (multi-device provisioning, available with macOS 14 or later)

The first three modes are only available to an authenticated User on the device, with the password enabled (FIA_UAU.2, FDP_ACC.2/Card_Perso_SFP, and FDP_ACF.1/Card_Perso_SFP).

Multi-device provisioning is available with macOS 14 or later. When a user provisions an eligible payment card, they will also be able to push provision the card to other Apple Pay-capable devices on the same iCloud account. Nothing is copied from the original device; the other devices provision using the same flow they would use during device setup. A macOS device is capable of both initiating the provisioning of cards to other devices and receiving cards that were initiated via multi-device provisioning on another device.

When a user adds a card to Wallet, the TSF encrypts card data (FPT_ITC.1) and sends it to Apple servers (FDP_ETC.2/Card_Perso_SFP). Full card numbers are not stored on the device (FDP_ITC.1) or on Apple servers.

Integrity protections are in place to prevent alteration of the enrolled Apple Pay card data. (FDP_SDI.1).

7.5.2. AP_Cancellation

When the User decides to remove an Apple Pay card from the Apple Wallet, the TSF order the Secure Element to securely invalidate and remove the Device Account Number (FDP_RIP.1).

7.5.3. AP_Revocation

When the card issuer decides to suspend or unlink an Apple Pay card from the Apple Wallet of a TOE User, the TSF order the Secure Element to securely invalidate and remove the Device Account Number (FDP_RIP.1).

7.6. SF Payment management

When a device initiates an Apple Pay transaction, the Secure Element, in the TOE Environment, only allows a payment to be made after it receives authorization from the Secure Enclave. This involves confirming the User has provided intent and has authenticated with biometric authentication, or using the device password (FDP_ACC.2/Payment_SFP, FDP_ACF.1/Payment_SFP, and FMT_SMR.1). Biometric authentication is the default method if available, but the password can be used at any time. A password is automatically offered after three unsuccessful attempts to match biometrics; after five unsuccessful attempts, the password is required. A password is also required when biometric authentication is not configured or not enabled for Apple Pay (FMT_SMF.1, FMT_MSA.1, and FMT_MSA.3).

Apple Pay includes an anti-replay mechanism that prevents transactions from being repeated by including in D.Payment_Data (An Apple Pay server nonce).

The processing of the Apple Pay transaction happens in the TOE Environment, on the Secure Element, using the secret card data, the Transaction Token and producing a payment cryptogram. The TOE ensures that the payment evidence transmitted back to the card issuer for processing (through a Terminal or network) was authorized by the User (FIA_UAU.6 Re-authenticating). In the case of Apple Pay online transactions, which are processed through the Apple Pay server, this integrity protection ensures the Dynamic Linking of the transaction data by a cryptographic based Authentication Code as exposed in the PSD2 regulation. The Apple Pay server ensures the integrity of the Dynamic Linking, and the card issuer verifies that it corresponds to a valid transaction, containing the right Transaction Token and produced by a genuine Apple Pay card (FDP_DAU.1).

The exported user data (transaction data displayed to S.User) is controlled by FDP_ETC.2/Transaction.

The Secure Element, in the TOE Environment, is responsible for ensuring the confidentiality of the Apple Pay Card data during the transaction processing.

The following table summarizes the functions supporting Payment management.

TSF	Description
AP_Transaction	Processing of an Apple Pay transaction.

7.7. SF OS Update

An OS/MK firmware update can be offered at any time by Apple to the User. The User is required to authenticate through a Password verification, or the update cannot be installed (FIA_UAU.6, FIA_UAU.2, FMT_MTD.3).

The OS/MK firmware update preserves the user attributes, especially the card data, the password, the enrolled biometric patterns, and other authentication parameters (FIA_ATD.1).

The ability to update the D.OS is restricted to Authenticated user (FMT_MTD.1).

7.8. SF iCloud logout & disk erase

An iCloud logout is performed when the User unlinks a device from an iCloud account. When this happens, the TOE ensures that the iCloud Account related data is securely deallocated, and that no residual information is left behind (FDP_RIP.1).

The most destructive security function available to the User on a macOS device is the disk erase or device reset, as they erase of all the User data. When this is performed, the TOE ensures that all the sensitive data terminated as part of a disk erase is protected from leaving residual information. This covers the iCloud Account information, all the Apple Pay Card Data, and the User authentication data like password and Biometrics (FDP_RIP.1).

Change History

Date	Version	Author	Comments
2024-02-09	1.0	Apple	Initialization of the Security Target
2024-04-26	2.0	Apple	Minor updates
2024-07-16	3.0	Apple	Minor updates
2024-10-01	4.0	Apple	Minor updates