12. Accessory Authentication

Accessory authentication provides a mechanism for Apple devices to trust the identities and feature sets of accessories that are compliant with this specification.

It is also possible for an accessory to require the Apple device to authenticate itself to the accessory. For more details, see Device Authentication (page 523).

12.1 Accessory Authentication Requirements

The accessory hardware must incorporate an Apple Authentication Coprocessor, version 2.0B or 2.0C. Version 2.0C is strongly recommended and specified in Apple Authentication Coprocessor 2.0C (page 66). Earlier versions of the authentication coprocessor are not allowed.

An accessory may use one authentication processor to authenticate itself to more than one Apple device. Conversely, multiple accessories must not share one authentication coprocessor or authenticate on behalf of each other to the same Apple device.

12.1.1 Accessory Authentication over iAP2 Requirements

The accessory must have successfully established an iAP2 link over an iAP2 transport.

All accessories that support the Accessory Authentication feature via iAP2 must send or receive the following iAP2 control session message(s):

RequestAuthenticationCertificate (page 805)

AuthenticationCertificate (page 805)

RequestAuthenticationChallengeResponse (page 805)

AuthenticationResponse (page 806)

AuthenticationFailed (page 806)

12.2 Accessory Authentication Usage

12.2.1 Accessory Authentication over iAP2 Usage

Authentication over iAP2 is always initiated with the transmission of a RequestAuthenticationCertificate (page 805) message from the Apple device to the accessory.

If control session version 2 is in use and the RequestAuthenticationCertificate (page 805) message contains a RequestAuthenticationCertificateSerialNumber parameter, the accessory should retrieve the X.509 certificate serial number from its Apple authentication coprocessor and transmit it to the Apple device using an AccessoryAuthenticationSerialNumber (page 807) message.

If the serial number is not available or the RequestAuthenticationCertificateSerialNumber parameter is not present, the accessory must retrieve the X.509 certificate from its Apple authentication coprocessor and transmit it to the Apple device using an AuthenticationCertificate (page 805) message. The certificate must be sent within 1 second of receiving the RequestAuthenticationCertificate (page 805).

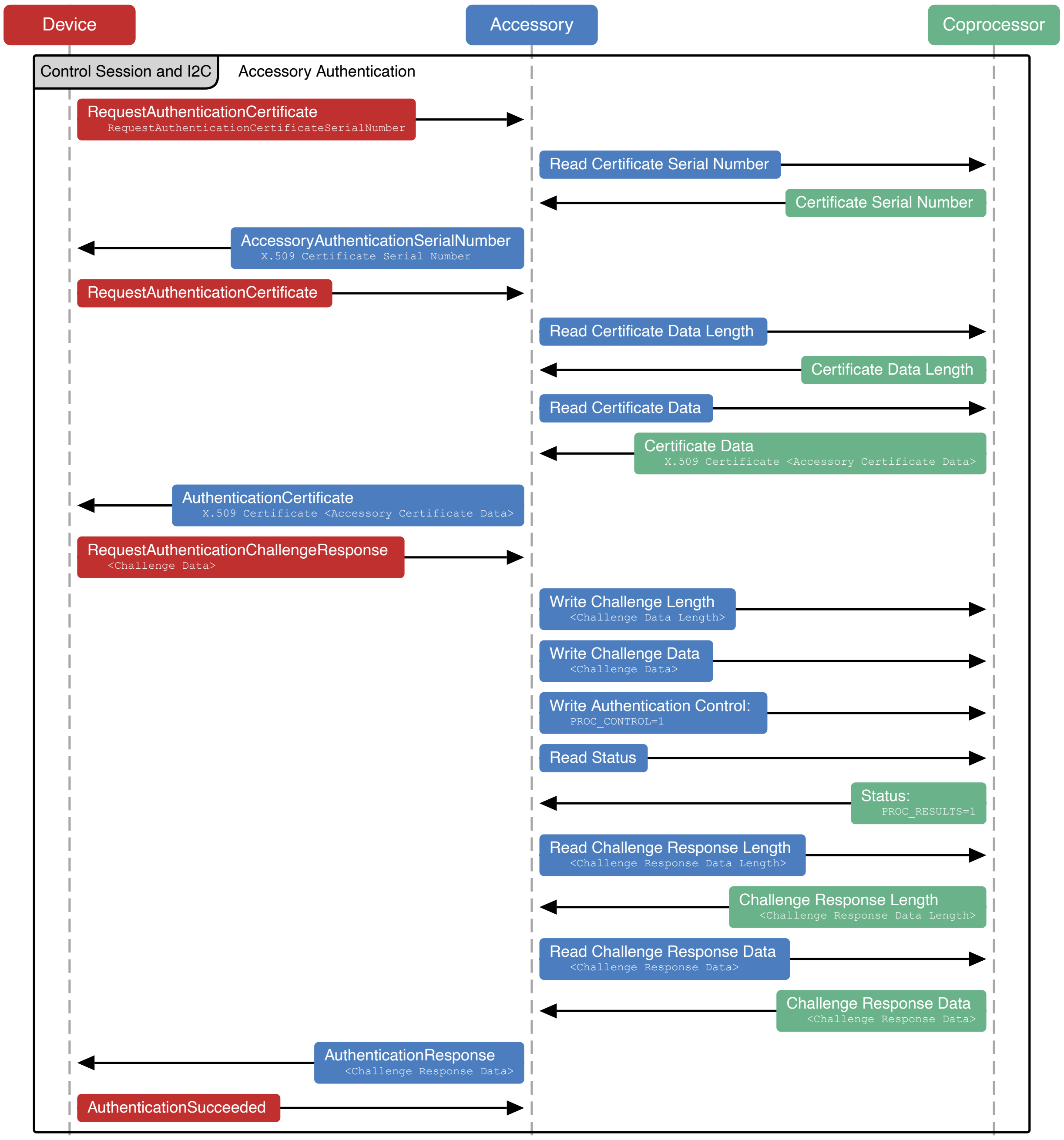
If the certificate is valid, the Apple device will respond with a RequestAuthenticationChallengeResponse (page 805) message. The accessory then must use the Apple authentication coprocessor to compute a response to the authentication challenge and send an AuthenticationResponse (page 806) message with the response. This response must be sent within 2 seconds of receiving the RequestAuthenticationChallengeResponse (page 805) message.

If the authentication response is correct, then the Apple device will send an AuthenticationSucceeded (page 806) message. The accessory must immediately proceed to Accessory Identification (page 265) unless it is using control session version 2 and has already identified (see Control Session (page 789)). With control session version 2, once the Apple device starts authentication, the accessory must complete the process within 15 seconds.

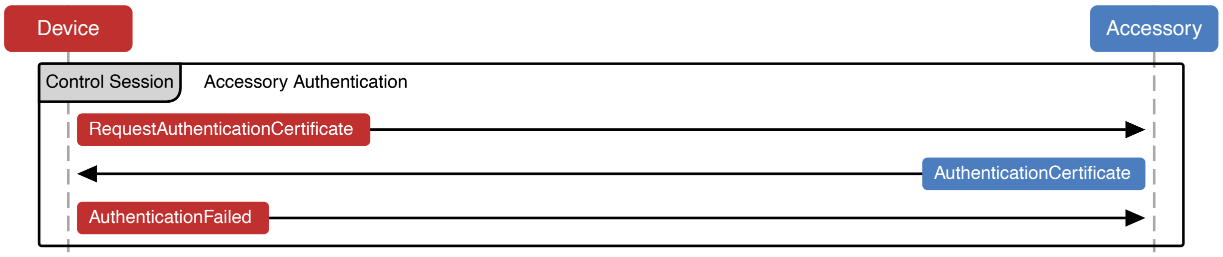
To demonstrate correct handling of the AuthenticationFailed (page 806) message during self-certification, accessories must send an empty or invalid certificate when the Apple Authentication Coprocessor is not present.

12.3 Accessory Authentication Examples

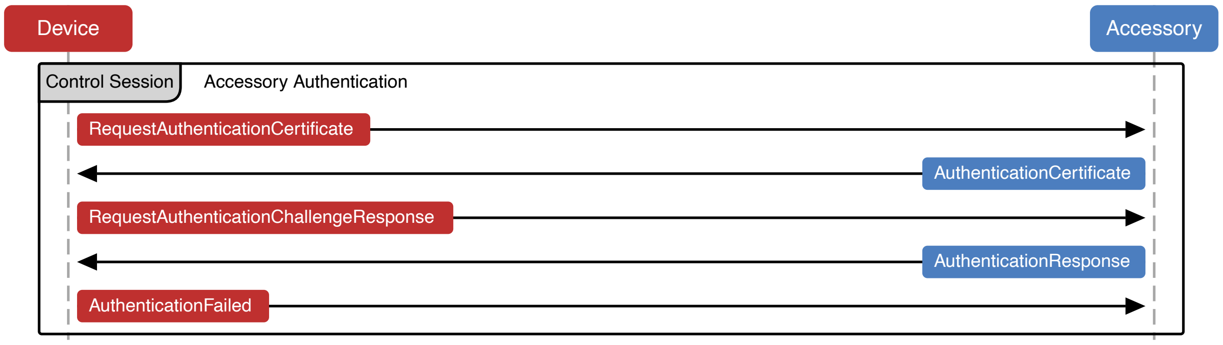
12.3.1 Typical Accessory Authentication over iAP2


Device
Accessory
Coprocessor
Control Session and I2C
Accessory Authentication
RequestAuthenticationCertificate
RequestAuthenticationCertificateSerialNumber
Read Certificate Serial Number
Certificate Serial Number
AccessoryAuthenticationSerialNumber
X.509 Certificate Serial Number
RequestAuthenticationCertificate
Read Certificate Data Length
Certificate Data Length
Read Certificate Data
Certificate Data
X.509 Certificate <Accessory Certificate Data>
AuthenticationCertificate
X.509 Certificate <Accessory Certificate Data>
RequestAuthenticationChallengeResponse
<Challenge Data>
Write Challenge Length
<Challenge Data Length>
Write Challenge Data
<Challenge Data>
Write Authentication Control:
PROC_CONTROL=1
Read Status
Status:
PROC_RESULTS=1
Read Challenge Response Length
<Challenge Response Data Length>
Challenge Response Length
<Challenge Response Data Length>
Read Challenge Response Data
<Challenge Response Data>
Challenge Response Data
<Challenge Response Data>
AuthenticationResponse
<Challenge Response Data>
AuthenticationSucceeded

12.3.2 Accessory Authentication over iAP2 Failure Due To Invalid Certificate



12.3.3 Accessory Authentication over iAP2 Failure Due To Invalid Response



13. Accessory Identification

All accessory interface features other than charging require that the identification feature be supported.

13.1 Accessory Identification Requirements

The accessory must have successfully completed Accessory Authentication (page 261) unless using control session version 2. See Control Session (page 789).

Control session version 2 allows the Apple device to choose the order of identification and authentication, so the accessory must be ready for either StartIdentification (page 807) or RequestAuthenticationCertificate (page 805) after successful link layer negotiation.

13.2 Accessory Identification Usage

All accessories that support the Accessory Identification feature via iAP2 must send or receive the following iAP2 control session message(s):

- StartIdentification (page 807)
- IdentificationInformation (page 807)
- IdentificationAccepted (page 816)
- IdentificationRejected (page 817)
- CancelIdentification (page 819)

All accessories that support the Accessory Identification feature via iAP2 may also send or receive the following iAP2 control session message(s):

- IdentificationInformationUpdate (page 819)

Identification is always initiated with the transmission of a StartIdentification (page 807) message from the Apple device to the accessory. The accessory must respond with a IdentificationInformation (page 807) message. The IdentificationInformation (page 807) message is evaluated by the Apple device to determine whether all of the requested iAP2 connection features are supported. If the parameters are valid and the Apple device can support all of the requested messages, it will send a IdentificationAccepted (page 816) message to the accessory.

Otherwise, a IdentificationRejected (page 817) message is sent detailing which invalid parameters and unsupported messages were present in the previous IdentificationInformation (page 807) message. Note that it is possible for the list of unsupported messages to be empty; that is an indication of incorrect accessory implementation, rather than unsupported features on a specific device.

Accessory Test System (ATS) output and Apple device console logs (see 'Getting Crash Logs and Console Output' at http://developer.apple.com/library/ios/#qa/qa1747/ will also provide additional supporting information which is useful for accessory development. Apple strongly recommends that accessory developers pay close attention to both ATS and the device logs.

If the identification information has been rejected, the accessory must send another IdentificationInformation (page 807) message with different parameters or abort the identification process using CancelIdentification (page 819). Cancellation of the identification process will cause the Apple device to inform the user that the accessory is not supported.

Failure of the accessory to send IdentificationInformation (page 807) within 1 second of either StartIdentification (page 807) or IdentificationRejected (page 817) is grounds for failing to pass self certification.

Once IdentificationAccepted (page 816) is received, the accessory must not send any more identification messages other than IdentificationInformationUpdate (page 819).

It is possible for an accessory to update its identification information after IdentificationAccepted (page 816) by sending an IdentificationInformationUpdate (page 819) message. For example, the user may change the accessory's active language. When that occurs, the accessory must send an IdentificationInformationUpdate (page 819) message containing both the new active language and accessory information strings in the new language.

If the identification process fails or the accessory cancels identification, the Apple device may choose to restart the identification process or terminate the iAP2 link.

13.2.1 Accessory Identification of Manufacturing Information

Accessories must provide values for the following IdentificationInformation (page 807) message parameters that match the accessory's marking and packaging:

- `Name`
- `ModelIdentifier`
- `Manufacturer`
- `SerialNumber`

Additionally, the `FirmwareVersion` and `HardwareVersion` parameter values must uniquely reflect the current revision of the accessory's firmware and hardware respectively.

Accessories must not provide blank or generic string values (such as those provided by a reference design or component vendor) for any manufacturing information parameters.

The SerialNumber parameter should be unique for each accessory (i.e. serialized); Apple devices may use this to differentiate multiple accessories of the same type.

13.2.2 Accessory Identification of Sent/Received iAP2 Control Session Messages

The `MessagesSentByAccessory` and `MessagesReceivedFromDevice` parameters in IdentificationInformation (page 807) are critical to accessory identification. Accessories must exhaustively enumerate all iAP2 control session messages that they can send or receive. During the self certification process, the accessory must be used in a manner that demonstrates correct implementation of all declared messages.

There is only one exception to this requirement. Messages associated with the Authentication and Identification features do not need to be in the `MessagesSentByAccessory` or `MessagesReceivedFromDevice` lists, as all accessories that implement iAP2 must support those features.

13.2.3 Accessory Identification of Power Capabilities

All accessories must provide values for the `PowerProvidingCapability` and `MaximumCurrentDrawnFromDevice` parameters in their IdentificationInformation (page 807) messages.

If an accessory does not provide power to the Apple device, `PowerProvidingCapability` must be set to `None`.

If an accessory does not draw power from the Apple device, `MaximumCurrentDrawnFromDevice` must be set to `0`.

Otherwise, the accessory must set those parameters as required by Power (page 660).

13.2.4 Accessory Identification of iAP2 Transport Components

Every accessory that implements iAP2 must identify one and only one transport component that supports an iAP2 connection. That transport component must contain a `TransportSupportsiAP2Connection` parameter. Depending on the type of transport, additional identifying information, such as a Media Access Control (MAC) address, may be required. The iAP2 connection must be implemented using the identified transport component.

Some accessory interface features require a specific transport component to be implemented in order to function. This requirement is independent of whether or not an iAP2 connection is implemented on that transport. For example, an accessory must identify a `USBDeviceModeTransportComponent` for Digital Audio over the USB Device Mode transport. Every identified transport component must have a unique identifier and human-readable name that reflects the intended purpose of the component, such as 'iAP2' or 'Digital Audio Speakers'.

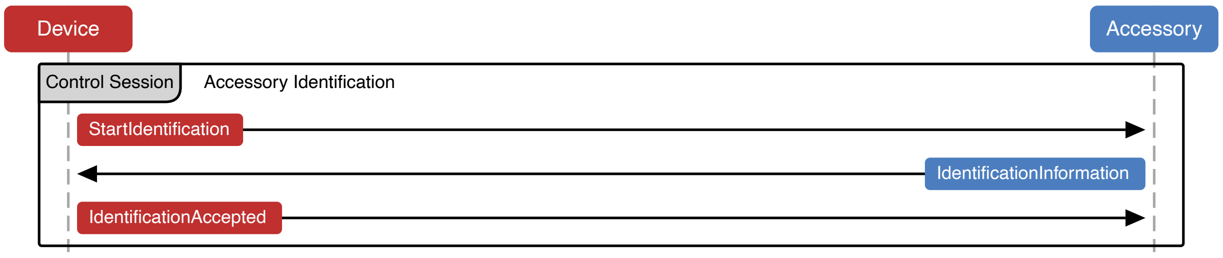
All transport components must have unique identifiers.

13.2.5 Additional Accessory Identification Parameters

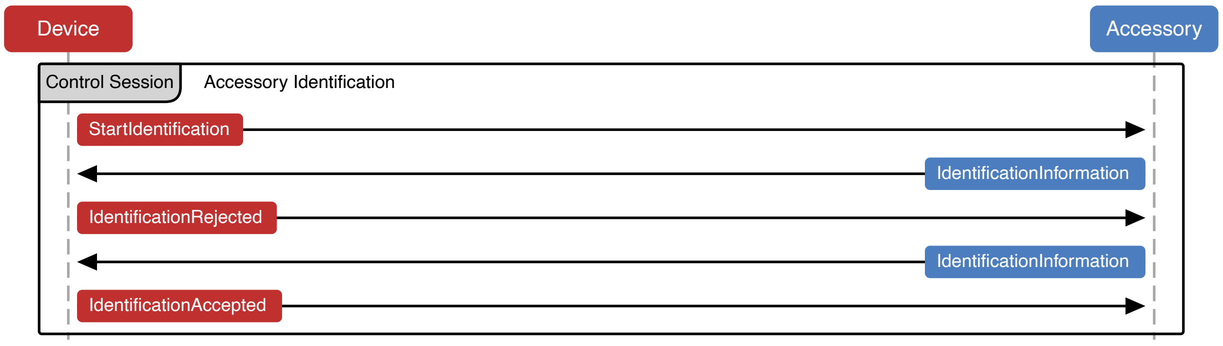
Several accessory interface features require the accessory to supply additional identification parameters. See the appropriate sections in the specification for details.

13.3 Accessory Identification Examples

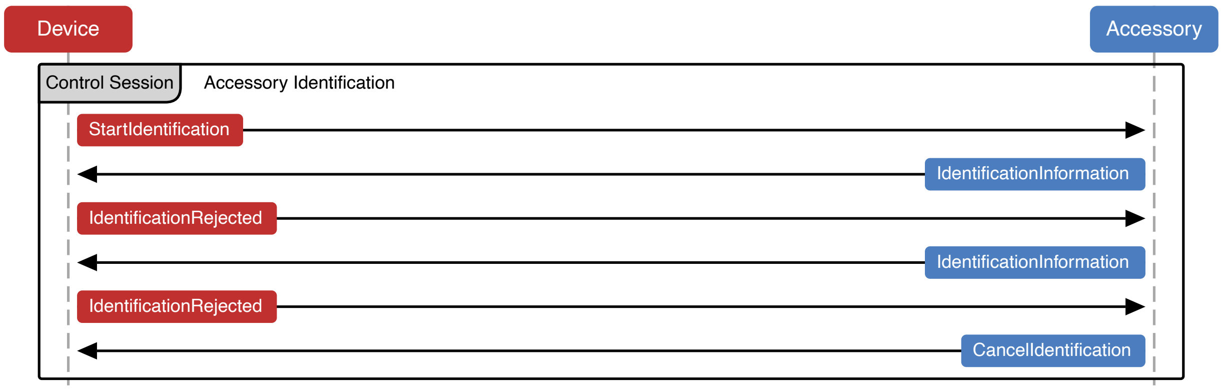
13.3.1 Typical Accessory Identification



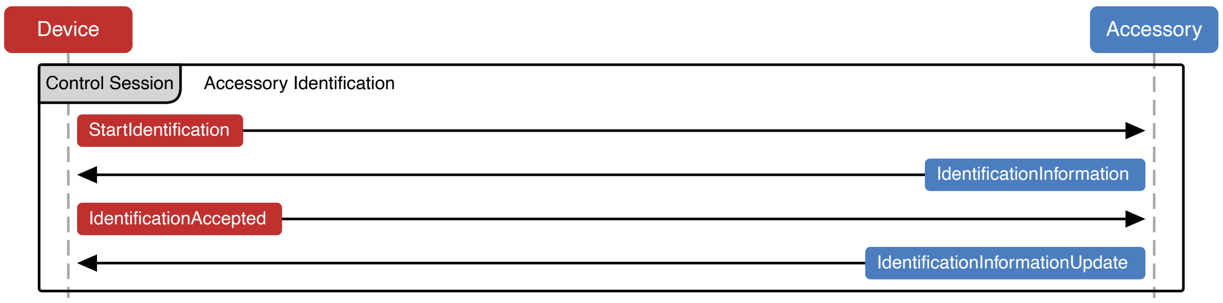
13.3.2 Successful Accessory Identification With Two Tries



13.3.3 Unsuccessful Accessory Identification After Two Tries



13.3.4 Accessory Identification With Information Update



13.4 Test Procedures

Equipment Required

- Latest revision of ATS software
- ATS hardware

1. Verify that the accessory identifies with all of the required accessory information fields in the ATS iAP Summary.
   a. Connect the accessory and the Apple device to ATS.
   b. View iAP Summary page.
   c. Confirm required fields (Name, brand, etc.) are correctly filled in (no "filler" entries, such as ACME, 1234, etc.)
2. Verify that the accessory supports and is connected to a non-IDPS supported device such as iPod Classic, that it properly falls back to Identify Device Lingoes when attached to the non-IDPS device.
   a. Connect the accessory and the non-IDPS Apple device to ATS.

   b. View iAP summary page.
   c. Verify Identification Method is IDL.
3. Verify accessory does not claim a lingo or message that is does not use.
   a. Consult accessory Product Plan for claimed messages/lingoes.
   b. Compare to Summary Page in ATS.