EMV®
3-D Secure

Version Number Management—Protocol Version 2.3.0 & above

Version 1.0

September 2021

Legal Notice

The EMV® Specifications are provided “AS IS” without warranties of any kind, and EMVCo neither assumes nor accepts any liability for any errors or omissions contained in these Specifications. EMVCO DISCLAIMS ALL REPRESENTATIONS AND WARRANTIES, EXPRESS OR IMPLIED, INCLUDING WITHOUT LIMITATION IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT, AS TO THESE SPECIFICATIONS.

EMVCo makes no representations or warranties with respect to intellectual property rights of any third parties in or in relation to the Specifications. EMVCo undertakes no responsibility to determine whether any implementation of the EMV® Specifications may violate, infringe, or otherwise exercise the patent, copyright, trademark, trade secret, know-how, or other intellectual property rights of third parties, and thus any person who implements any part of the EMV® Specifications should consult an intellectual property attorney before any such implementation.

Without limiting the foregoing, the Specifications may provide for the use of public key encryption and other technology, which may be the subject matter of patents in several countries. Any party seeking to implement these Specifications is solely responsible for determining whether its activities require a license to any such technology, including for patents on public key encryption technology. EMVCo shall not be liable under any theory for any party’s infringement of any intellectual property rights in connection with the EMV® Specifications.

Contents

1 Introduction

The EMV 3-D Secure protocol is aimed at securing authentication in browser-based and app-based transactions. Due to new technologies, regulatory requirements, or identified security vulnerabilities, it is important for the EMV 3-D Secure protocol to control different versions and manage how those versions get deployed into production and used during the authentication process.

1.1 Application

This document applies to specifications release and management practice of EMV 3-D Secure Protocol versions 2.3.0 and onward.

1.2 Audience

This document is intended for use by implementers of the EMV 3-D Secure protocol.

1.3 Terminology and Conventions

The terminology used in this document and their specific meaning is described in the following sections.

1.3.1 Specification Version

Specification Version defines the iteration of EMV 3-D Secure specification-related documents. For example, 2.3.0.1

Specification Version format: MAJOR.MINOR.PATCH.EDITORIAL

- MAJOR version for significant specification changes
- MINOR version for added functionality
- PATCH version for specification fixes
- EDITORIAL version for specification update related to a Minor release that does not impact interoperability.

Note: First three positions of Specification Version, Major/Minor/Patch, are always identical to the Protocol Version that released within the same specification-related document.

1.3.2 Protocol Version

Protocol Version defines the interoperability between the EMV 3-D Secure components (3DS Server, DS, ACS, 3DS SDK). For example, 2.3.0.

Protocol Version format: MAJOR.MINOR.PATCH

- MAJOR version for significant specification changes

- MINOR version for added functionality
- PATCH version for specification fixes

1.3.3 Specification Bulletin

Specification Bulletins provide a reference of updates, clarifications and errata incorporated into the EMV 3-D Secure specifications. For example, EMV Specification Bulletin No. 227 v2.

Note: All updates, clarifications and errata are also covered in the latest specification-related documents, so specification Bulletins are non-essential reference documents.

Specification Bulletin format: BULLETIN NUMBER + BULLETIN REVISION VERSION

- BULLETIN NUMBER is 3-digit number assigned by EMVCo
- BULLETIN REVISION VERSION for iterative updates, clarifications and errata to the EMV 3D Secure Specifications

1.3.4 Message Version Number

Message Version Number is a 3-D Secure data element that refers to Protocol Version number. Message Version Number is set by the 3DS Server in the AReq message. For example, 2.3.0.

- The 3DS Server determines the highest common Message Version Number based on its evaluation of the Protocol Version(s) supported by the DS, ACS, 3DS SDK and assigns the Message Version.
- The Message Version Number for a specific transaction is consistent following the Message Version Number in the AReq transaction across all 3-D Secure protocol messages.

1.3.5 Data Version

Data Version defines the set of device identification parameters that the 3DS SDK shall collect. The Data Version is a means for participating EMV 3-D Secure components to know which set of device identification parameters is being transferred. For example,1.4.

Note: Data Version is independent of the Protocol Version.

Data Version format: MAJOR.MINOR

- MAJOR version for significant parameter changes
- MINOR version for iterative changes on the same parameter structure

1.3.6 Definitions

For the definition of the terms used in this document, refer to the *EMV 3-D Secure Protocol and Core Functions Specification,* Table 1.3.

1.4 Supporting Documentation

The following documents are specific to the EMV 3-D Secure protocol and should be used in conjunction with this document. These documents as well as EMV 3-D Secure FAQs are located on the EMVCo website under the 3-D Secure heading.

- *EMV 3-D Secure—Protocol and Core Functions Specification* versions 2.3.0.0 and above
- *EMV 3-D Secure SDK Specification* versions 2.3.0.0 and above
- *EMV 3-D Secure Split-SDK Specification* versions 2.3.0.0 and above
- *EMV 3-D Secure SDK—Device Information*

2 Specification Version Management

This chapter provides an overview of 3-D Secure specification version management. Specification version update includes major/minor specification releases, and/or patch releases that impact interoperability, as well as editorial updates that do not impact interoperability.

EMV 3-D Secure specifications identify the Specification Version using a four-part version number:

- Specification Version number format: MAJOR.MINOR.PATCH.EDITORIAL
- Specification Version number example: 2.3.0.0

2.1 Major Releases

A Major Release occurs when EMVCo has made non-interoperable software changes to the 3-D Secure specification that signals to the industry that the specification has significant changes. For example, the Specification Version number would be updated from 2.3.1.0 to 3.0.0.0.

Note: Major Release changes are currently not applicable and are therefore not addressed in detail in this document.

Specification Version impact:

- Position 1 is incremented
- Position 2, 3 and 4 are reset to “0”

Example:

- Previous Specification Version number: 2.3.1.0
- New Specification Version number: 3.0.0.0

2.2 Minor Releases

A Minor Release indicates that EMVCo has made non-interoperable software changes to the 3-D Secure specification that signals to the industry that the 3-D Secure specification has new features and/or new functionalities for implementation. For example, the Specification Version number would be updated from 2.3.1.0 to 2.4.0.0.

2.2.1 Numbering

Specification Version number impact

- Position 1, no change
- Position 2 is incremented
- Position 3 and 4, are reset to “0”

Example:

- Previous Specification Version: 2.3.1.0
- New Specification Version: 2.4.0.0

2.2.2 Release Plan

2.2.2.1 Draft Release for Associate and Subscriber Input and Feedback

Note: All Minor Release changes to the 3-D Secure specification are first released as drafts to allow EMVCo Associates and Subscribers to provide input and receive working group feedback on the draft documentation. After the input and resulting feedback is finalised, the documentation is then published as public to the industry.

Each draft specification release will include the Specification Version Number and the draft number in the Specification and the accompanying Draft Specification Bulletin.

2.2.2.2 Documentation

3-D Secure specification-related documentation will be updated to the new Specification Version number and will be published to the EMVCo website. These documentation updates include:

- *EMV 3-D Secure Protocol and Core Functions Specification*
- *EMV 3-D Secure Split-SDK Specification*
- *EMV 3-D Secure SDK Specification*

Immediately after the release, these documents will carry the same Specification Version number, regardless of whether changes were made that impact only one of the documents. If changes were not made to the document, it will be noted under the document's Revision Log. All other 3-D Secure documentation will be updated if changes are deemed necessary for that Minor Release.

2.2.3 Communication

EMVCo will publish a Bulletin providing the scope of the updates for implementation.

2.3 Patch Release

A Patch Release indicates that EMVCo has made non-interoperable changes to the 3-D Secure specification that signals to the industry that the 3-D Secure specification has added functional fixes or security updates following, but related to, a Minor Release.

Patch Releases will be used for Protocol Version 2.3.0 and onward. For example, the Specification Version number would be updated from 2.3.0.0 to 2.3.1.0.

2.3.1 Numbering

Specification Version number impact

- Position 1, no change
- Position 2, no change

For protocol 2.3.0 and higher:

- Position 3 is incremented
- Position 4, is reset to “0”

Example

- Previous Specification Version: 2.3.0.0
- New Specification Version: 2.3.1.0

2.3.2 Release Plan

2.3.2.1 Documentation

3-D Secure specification-related documentation will be updated to the new Specification Version number and will be published to the EMVCo website. These documentation updates include:

- *EMV 3-D Secure Protocol and Core Functions Specification*
- *EMV 3-D Secure Split SDK Specification*
- *EMV 3-D Secure SDK Specification*

Immediately after the release, these documents will carry the same Specification Version number, regardless of whether changes were made that impact only one of the documents. If changes were not made to the document, it will be noted under the document’s Revision Log. All other 3-D Secure documentation will be updated if changes are deemed necessary for that Patch Release.

If patch releases are made, all details would also be captured within the Bulletin for informational purposes.

2.3.2.2 Communication

EMVCo will publish a Bulletin providing the scope of the updates for implementation.

The Bulletin will be published to the public without EMVCo Associates and Subscribers review.

2.4 Editorial Updates

Editorial update indicates that EMVCo has made an interoperable change to the 3-D Secure specification, that signals to the industry that the 3-D Secure specification has added updates and/or clarifications following a Minor release. For example, the Specification Version number would be updated from 2.3.0.0 to 2.3.0.1.

2.4.1 Numbering

Specification Version number impact

- Position 1, no change
- Position 2, no change
- Position 3, no change

For protocol 2.3.0 and higher:

- Position 4 is incremented

Note: The editorial updates for protocol 2.3.0 and onward are different from the practice for protocol 2.1.0 and 2.2.0.

Note: An editorial update for 3DS Core Specification Version does not impact 3DS SDK Specification Versions, and vice versa. For protocol 2.3.0 and onward, for example:

Stage 0, Initial status

Protocol Version (PV)	Specification Name	Specification Version (SV)
2.3.0	Core Spec	2.3.0.2
	Default-SDK	2.3.0.0
	Split-SDK	2.3.0.0

Stage 1, an editorial update applies to core specification

- Previous Core Specification Version number: 2.3.0.2
- New Core Specification Version number: 2.3.0.3

Protocol Version (PV)	Specification Name	Specification Version (SV)
2.3.0	Core Spec	2.3.0.3
	Default-SDK	2.3.0.0
	Split-SDK	2.3.0.0

Stage 2, an editorial update applies to Split-SDK specification

- Previous Split-SDK Specification Version number: 2.3.0.0
- New Split-SDK Specification Version number: 2.3.0.1

Protocol Version (PV)	Specification Name	Specification Version (SV)
2.3.0	Core Spec	2.3.0.3
	Default-SDK	2.3.0.0
	Split-SDK	2.3.0.1

2.4.2 Release Plan

2.4.2.1 Documentation

3-D Secure specification-related documentations will be updated to the new Specification Version numbers. The update would be included within the next Minor Release or Patch Release.

These documentation updates include:

- *EMV 3-D Secure Protocol and Core Functions Specification*
- *EMV 3-D Secure Split-SDK Specification*
- *EMV 3-D Secure—SDK Specification*

Each document would have the Specification Version Number increment independently.

If editorial updates are made, all details would also be captured within the Bulletin for informational purposes.

2.4.3 Communication

EMVCo will publish a Bulletin announcing the scope of the updates for implementation.

The Bulletin will be published without EMVCo Associates and Subscribers draft review.

3 Protocol Version Management

This chapter provides an overview of 3-D Secure protocol version management. Protocol version update includes major/minor specification releases, and/or patch releases that impact interoperability.

EMV 3-D Secure specifications and 3-D Secure components identify the Protocol Version using a three-part version number.

- Protocol Version number format: MAJOR.MINOR.PATCH
- Protocol Version number example: 2.3.0

3.1 Major/ Minor/ Patch Releases

Major/ Minor/ Patch releases of Protocol Version follow the same update process of Specification Version.

3.1.1 Numbering

Protocol Version number impact:

- Position 1, 2, 3 follow the same update of Specification Version

Major Release Example:

- Previous Protocol Version number: 2.3.1
- New Protocol Version number: 3.0.0

Minor Release Example:

- Previous Protocol Version number: 2.3.1
- New Protocol Version number: 2.4.0

Patch Release Example:

- Previous Protocol Version number: 2.3.1
- New Protocol Version number: 2.3.2

4 Identifying Which Message Version to Use

The 3DS Server is responsible for determining which Message Version to use when initiating authentication messages by ensuring each 3DS component supports the Protocol Version for each 3-D Secure transaction.

If the Message Version/Protocol Version is not aligned amongst 3-D Secure components for a 3-D Secure transaction, a Message Error will result.

4.1 DS and ACS

For the DS and ACS, the 3DS Server uses the PReq/PRes message pair to identify the Protocol Versions supported by each DS and ACS. The 3DS Server caches this information to build messages that can be processed by the DS and ACS.

4.1.1 PReq/PRes Message Pair

The 3DS Server can cache information about the Protocol Versions supported by the available ACS and DS. The data will be organised by the card ranges as configured by a DS. The information provided on the Protocol Versions supported by ACS and the DS can be utilised in the App-based, Browser-based and 3RI flows.

4.1.2 Responsibilities

4.1.2.1 ACS Responsibility

It is the responsibility of the ACS to indicate its active Protocol Versions that is supported for the ACS URL to each DS.

How each DS manages this functionality may differ and is not within scope of the 3-D Secure specification.

4.1.2.2 DS Responsibility

It is the responsibility of the DS to provide the active Protocol Versions that the ACS and DS support to the 3DS Server.

- For the DS, the DS Protocol Versions data element is used.
- For an ACS, the ACS Protocol Versions data element is used.

4.1.2.3 3DS Server Responsibility

It is the responsibility of the 3DS Server to make PReq message calls to each registered DS every 24 hours at a minimum, and once per hour at a maximum to refresh its cache of Protocol Versions that DS and ACS support for 3-D Secure.

Note: The 3DS Server is responsible to ensure each 3-D Secure component supports the Message Version number via the Protocol Version on every 3-D Secure transaction. Otherwise, the 3-D Secure transaction will result in a Message Error.

4.2 3DS SDK

For a 3DS SDK, the 3DS Server similarly needs to learn which Protocol Version is supported by the 3DS SDK for an In-App authentication. Guidance on how a 3DS Server could obtain the supported Protocol Version of a 3DS SDK is described below.

Note: The term "3DS Requestor" is used when referencing both the Default SDK interfacing with the "3DS Requestor App" and the Split-SDK interfacing with the "3DS Requestor".

4.2.1 Sample Flow—3DS Server determines Protocol Version

The 3DS Requestor knows the Protocol Versions supported by the 3DS SDK during integration of a particular 3DS SDK version. This information is typically provided by the 3DS SDK vendor.

- 3DS Requestor sends the list of supported protocol versions to the 3DS Server (together with the Cardholder Account Number) using proprietary functions.
- 3DS Server decides the Protocol Version used based on DS and ACS supported Protocol Versions and the 3DS SDK supported Protocol Versions.
- 3DS Server passes back the Protocol Version to the Request to be used along with the Directory Server ID.
- 3DS Requestor passes the Protocol Version and the Directory Server ID during the activation step of the SDK. (In the Default-SDK, the activation step refers to the `createTransaction` method. In the Split-SDK, this refers to the `Initiate Authentication Request` step).

Note: A future specification enhancement for the Default-SDK could define a `getSDKProtocolVersions` interface for the `ThreeDS2Service` so that the 3DS Requestor will not need to maintain the list, but instead can pass the returned value to the 3DS Server directly.

4.2.2 Sample Flow—3DS Requestor suggests Protocol Version (Option 1)

The 3DS Requestor, using proprietary functions obtains from the 3DS Server the Directory Server ID and list of Protocol Versions supported by the DS and the ACS for the Cardholder Account Number.

The 3DS Requestor invokes the activation step for the 3DS SDK using the highest protocol version from the list to the 3DS SDK.

- If the 3DS SDK supports the Protocol Version, the 3-D Secure transaction flow would be undisrupted.
- If the 3DS SDK does not support the Protocol Version, the 3DS SDK would return an error, in which case the 3DS Requestor downgrades the Protocol Version and would try to activate the 3DS SDK again (In the Default-SDK, the error returned will be in the form of an `InvalidInputException` thrown when the `createTransaction` method is called. In the Split-SDK, an error as defined in the Split-SDK proprietary implementation will be returned during the `Initiate Authentication Request` step).

- This would continue until the Protocol Version is negotiated between the 3DS Requestor and the 3DS SDK. The 3DS Requestor shares the suggested Protocol Version with the 3DS Server through proprietary interfaces.

4.2.3 Sample Flow—3DS Requestor suggests Protocol Version (Option 2)

The 3DS Requestor, using proprietary functions, obtains from the 3DS Server the Directory Server ID and list of Protocol Versions supported by the DS and the ACS for the Cardholder Account Number.

- Because the 3DS Requestor knows the Protocol Versions supported by the 3DS SDK during the integration of a particular 3DS SDK version, it can select the highest common Protocol Version from the list provided by the 3DS Server and the 3DS SDK.
- The 3DS Requestor invokes the activation step of the 3DS SDK using the highest common Protocol Version determined in the previous step.
- The 3DS Requestor shares the suggested Protocol Version with the 3DS Server through proprietary interfaces.