EMV® Specification Bulletin No. 271

August 2022

EMV® 3-D Secure Split-SDK Specification v2.3.1.0

This Specification Bulletin No. 271 provides the updates, clarifications and errata incorporated into the EMV® 3-D Secure Split-SDK Specification since version 2.3.0.0

Applicability

This Specification Bulletin applies to:

- *EMV® 3-D Secure Split-SDK Specification, Version 2.3.1.0*

Updates are provided in the order in which they appear in the specification. Deleted text is identified using strikethrough, and red font is used to identify changed text. Unedited text is provided only for context.

Related Documents

EMV® 3-D Secure Split-SDK Specification, Version 2.3.0.0

Effective Date

- *August 2022*

Contents

EMV® 3-D Secure Split-SDK Specification v2.3.1.0* *1

Applicability 1

Related Documents 1

Effective Date 1

Throughout Specification* *5

Chapter 1 Introduction* *6

1.1 Purpose 7

1.4 Definitions 7

Table 1.2 Definitions 7

1.5 Abbreviations 7

Table 1.3: Abbreviations 8

Chapter 2 Getting Started with the Split-SDK* *9

2.1 Component Architecture 9

2.3 Limited Split-SDK 9

2.4 Protocol Version Support 9

Chapter 3 Message Processing Requirements* *10

3.2 Split-SDK Message Flow Requirements 10

Step 8 Assemble AReq Data 10

[Req 813] 10

3.3 Limited Split-SDK Requirements 10

[Req 33] 10

[Req 34] 11

Chapter 4 SDK Security* *12

4.4 Split-SDK Security Requirements 12

[Req 55] 12

[Req 46] 12

[Req 814] 12

Chapter 5 Split-SDK/Browser~~-SDK~~ Requirements* *13

5.1 Split-SDK/Browser~~-SDK~~ Architecture 13

Figure 5-1: Split-SDK/Browser~~-SDK~~ Component Architecture 13

5.2 Changes to the 3-D Secure Core Protocol Message Flow for a Split-SDK/Browser~~-SDK~~ Flow 13

Step 2: The 3DS Requestor/3DS Server 13

[Req 805] 13

[Req 806] 13

Step 9: The 3DS Server 14
[Req 807] 14
[Req 808] 14
Step 24: The 3DS Requestor Environment 14
[Req 809] 14
[Req 815] 14
[Req 816] 14
5.3 Changes to the Split-SDK Message Flow Requirements for a Split-SDK/Browser~~-SDK~~ 15
Step 8: Assemble AReq Message Data 15
5.4 Split-SDK/Browser Security 15
5.4.1 SDK Initialisation Security Checks 15
Table 5.1: Split-SDK/Browser~~-SDK~~ Initialisation Security Checks 15
5.4.2 SDK Server CSP and CORS Guidelines 15
5.4.3 Iframe and Sandbox Attributes 15
Table 5.2: iframe Attributes 15
5.5 Split-SDK/Browser User Interface 15
Chapter 6 Split-SDK/Shell~~-SDK~~ Requirements* *16
6.1 Split-SDK/Shell~~-SDK~~ Architecture 16
Figure 6-1: Split-SDK/Shell~~-SDK~~ Component Architecture 16
6.2 Changes to the 3-D Secure Core Protocol User Interface Requirements and Guidelines for a Split-SDK/Shell~~-SDK f~~Flow 16
6.2.1 Processing Screen Requirements 16
~~[Req 145]~~ [Req 822] 16
~~[Req 146]~~ [Req 823] 16
6.3 Changes to the Split-SDK Message Flow Requirements for a Split-SDK/Shell~~-SDK~~ 16
Step 2 Get SDK Ephemeral Public Key 17
[Req 48] 17
[Req 49] 17
[Req 50] 17
Step 8: Assemble AReq Message Data 17
Step 11b Close (Frictionless only) 17
~~[Req 8]~~ [Req 818] 17
[Req 51] 17
~~[Req 32]~~ [Req 819] 17

Step 31 Close 18
~~[Req 31]~~ [Req 820] 18
[Req 52] 18
~~[Req 32]~~ [Req 821] 18
6.4 SDK Security 18
6.4.1 SDK Initialisation Security Checks 18
6.4.2 Split-SDK/Shell~~-SDK~~ Security 18
[Req 53] 18
[Req 54] 18
[Req 817] 19

Throughout Specification

- Revisions added to improve grammar, consistency, clarity and readability without any effect on the meaning or interpretation of the specification are not included in this bulletin.
- For consistency and ease of reference, the *EMV® 3-D Secure Protocol and Core Functions Specification* is referred to as the *Core Specification*.
- Updates made to defined abbreviations, such as DH (Diffie–Hellman), have no substantive effect on the use of the underlying specification and are not reflected in this bulletin.

Chapter 1 Introduction

The Split-SDK functions much like the 3DS Default-SDK described in the *EMV® 3-D Secure—SDK Specification* and follows the App-based flow. The distinction of the Split-SDK is that some client functionality does not run on the device, but on a server component, thus implementing a model that splits the functionality between a Split-SDK Client (client side) and a Split-SDK Server (server side). SDK Type indicates whether a Default-SDK or Split-SDK is involved.

Although other 3DS components (for example, the ACS) have the same interaction model as for a 3DS Default-SDK, the Split-SDK integration model with the 3DS Requestor (for example, with Merchants) differs ~~because~~in that the Split-SDK may be integrated within a server component rather than within a 3DS Requestor App.

~~While there are many implementation options for a Split-SDK, the fundamental principle of a Split-SDK model is that the Split-SDK Client encrypts the CReq message. An implementation, where the Split-SDK Client cannot support that principle, can have only limited support of SDK Authentication Types and is not allowed to support SDK Authentication Types using static information. This type of implementation option is not considered a Split-SDK but is categorised as a Limited-SDK.~~

~~For an entity offering a Split-SDK architecture, one option is to use the Split-SDK Server with a JavaScript Client running in a Browser. This is referred to as a Browser-SDK.~~

This *EMV® 3-D Secure Split-SDK Specification* covers the basic functionality of a split architecture and identifies three implementation variants.

- Split-SDK/Native: the Client functionality is implemented using native platform code embedded within a 3DS Requestor App, as defined in Chapters 3 and 4.
- Split-SDK/Browser: the Client functionality is implemented using JavaScript running in a device Browser. The JavaScript is delivered from the Split-SDK Server at the time of the authentication. A Split-SDK/Browser-~~SDK~~based Client utilises the Split-SDK ~~c~~Client/~~s~~Server flow as defined in Figure 3-1, but with a JavaScript Client as defined in Chapter 5.
- ~~Another option to support the App-based flow is to use the Split-SDK Server with a JavaScript running in a secured WebView in an App. This is referred to as a Shell-SDK.~~Split-SDK/Shell: the Client functionality is implemented using JavaScript running in a secured WebView opened by the Split-SDK/Shell that is embedded in the 3DS Requestor App. The JavaScript is delivered from the Split-SDK Server at the time of the authentication. A ~~Shell-SDK~~Split-SDK/Shell-based Client utilises the Split-SDK Client/Server flow as defined in Figure 3-1, but with a ~~wrapped WebView~~JavaScript Client as defined in Chapter 6.

If the Client cannot securely encrypt the CReq message, then the Split-SDK is considered Limited, as defined in Section 3.3. For a Limited Split-SDK, the range of allowed Authentication Methods is limited to those that are dynamic (static Authentication Methods are not supported).

In the *Core Specification*, the Split-SDK Type data element indicates the implementation variants (Split-SDK Variant) and whether the Split-SDK is Limited (Limited Indicator).

1.1 Purpose

Additionally, this document describes the Split-SDK variants ~~which include~~ and their specific requirements.

- ~~Limited-SDK, where certain essential Split-SDK Client functions are implemented on the Split-SDK Server~~
- ~~Browser-SDK, with a JavaScript implementation for a Split-SDK Client~~
- ~~Shell-SDK, with a wrapped WebView component for a JavaScript Split-SDK Client~~

1.4 Definitions

~~For the definition of the terms used in this specification, refer to Table 1.3: Definitions in the *EMV 3-D Secure Protocol and Core Functions Specification*. Additionally, the definitions below are used in this specification.~~ In addition to the definitions listed in Table 1.3 in the *Core Specification*, this specification uses the definitions listed in Table 1.2 below.

Table 1.2 Definitions

Term	Definition
3DS Default-SDK	Software embedded in a 3DS Requestor App that operates on the user device as defined by in the *EMV® 3-D Secure—SDK Specification*.
~~Browser-SDK~~	~~Implementation choice for a Split-SDK Server, where the interaction with the Client utilises a Browser and JavaScript (or similar web technology).~~
~~Limited-SDK~~	~~Variant of a Split-SDK where certain essential Split-SDK Client functions are implemented on the Split-SDK Server.~~
~~Shell-SDK~~	~~Implementation choice for a Split-SDK Server, where the interaction with the Client utilises a WebView in a 3DS Requestor App and JavaScript (or similar web technology).~~
Split-SDK/Browser	Implementation choice for a Split-SDK Server, where the Client functionality is implemented using JavaScript running in a Browser (or similar web technology).
Limited Split-SDK	A Split-SDK implementation where the CReq encryption is not performed by the Split-SDK Client but by the Split-SDK Server.
Split-SDK/Native	Implementation choice for a Split-SDK Server, where the Client functionality is implemented using native platform code embedded within a 3DS Requestor App.
Split-SDK/Shell	Implementation choice for a Split-SDK Server, where the Client functionality is implemented as a JavaScript running in a WebView secured by the Split-SDK Shell that is embedded in a 3DS Requestor App.

1.5 Abbreviations

~~The~~In addition to the abbreviations listed in Table 1.4 in the Core Specification, this specification uses the abbreviations listed in Table 1.3 ~~are used in this specification~~below.

Table 1.3: Abbreviations

Abbreviation	Description
~~3DS~~	~~Three Domain Secure~~
~~3DS SDK~~	~~Three Domain Secure Software Development Kit~~
~~ACS~~	~~Access Control Server~~
~~CA~~	~~Certificate Authority~~
~~CA DS~~	~~Certificate Authority Directory Server~~
CORS	Cross-Origin Resource Sharing
CSP	Content-Security-Policy
~~DH~~	~~Diffie Hellman~~
~~DS~~	~~Directory Server~~
~~JSON~~	~~JavaScript Object Notation~~
~~JWE~~	~~JSON Web Encryption~~
~~MAC~~	~~Message Authentication Code~~
~~OOB~~	~~Out-of-Band~~
~~OTP~~	~~One-time Passcode~~
~~SDK~~	~~Software Development Kit~~
UI	User Interface
~~URL~~	~~Uniform Resource Locator~~
~~UUID~~	~~Universally Unique Identifier~~

Chapter 2 Getting Started with the Split-SDK

2.1 Component Architecture

Figure 2-1 depicts a typical Split-SDK/Native component architecture. The 3DS Requestor has embedded a Split-SDK Client in its 3DS Requestor App (typically, an SDK embedded in a Merchant mobile application).

Figures 2-1 and 2-2 were replaced and are not replicated in this bulletin.

2.3 Limited Split-SDK

A ~~Limited-SDK is a variant of the~~ Split-SDK ~~where~~is defined as Limited when one or more functions (outlined in ~~s~~Section 2.2) destined for the Split-SDK Client is~~ being~~/are performed by the Split-SDK Server. ~~A Limited SDK is defined as such in SDK Type.~~

It is up to implementers of a Limited Split-SDK to define which functions are moved to the Split-SDK Server. If any of the required Split-SDK Client functions (as outlined in ~~s~~Section 3.3) is not performed by the Split-SDK Client, the Split-SDK Type is inherently defined as ~~a~~ Limited~~-SDK~~ (Limited Indicator = Y) and can therefore support only a limited set of SDK Authentication Types – specifically the types that do not include static information.

2.4 Protocol Version Support

As defined in Req 311 in the *Core Specification*, the Split-SDK supports all lower active protocol versions (Protocol Version Status set to Active in *EMV® Specification Bulletin 255*).

The Split-SDK flow as described in this document continues to apply for protocol versions 2.1.0 and 2.2.0, and the Split-SDK implements the requirements of the respective *Core Specification* versions, in particular, for the UI and the OOB authentication.

Note: 3DS Server operators should consider the use of the Device Acknowledgement Message Extension to provide better information on the Split-SDK to the ACS.

Chapter 3 Message Processing Requirements

This chapter provides the Split-SDK processing flow for both Frictionless and Challenge Flows for a Split-SDK/Native.

3.2 Split-SDK Message Flow Requirements

Step 8 Assemble AReq Data

~~Note:~~ The ~~reference for the Device Information for a~~ Split-SDK ~~is~~shall:

[Req 813]

Provide the Platform Provider-specific Parameters ~~(See~~in the SDK Encrypted Data (refer to Section 2.8 in the *EMV® 3-D Secure SDK—Device Information*).

The following changes were made to the wording directly following Requirement 7 in Step 8 in this section:

The Split-SDK Server assembles the following data:

- SDK T~~t~~ype (Split-SDK = 02) ~~OR~~
  - ~~Limited-SDK = 03 (See [Req 33] in Section 3.3) OR~~
  - ~~Browser-SDK = 04 (See Section 5) OR~~
  - ~~Shell-SDK = 05 (See Section 6)~~
- Split-SDK Type

3.3 Limited Split-SDK Requirements

While the requirements in Chapters 2, 3 and 4 of this document apply to a ~~Limited~~Split-SDK, this section specifically defines a Limited Split-SDK and its additional requirements.

[Req 33]

If the Split-SDK Client cannot perform one or more of the functions defined in ***[Req 3]***, ***[Req 4]***, ***[Req 8]***, ***[Req 12]***, ***[Req 13]***, ***[Req 14]***, ***[Req 15]***, ***[Req 16]***, ***[Req 19]***, ***[Req 20]***, ***[Req 27]***, ***[Req 28]***, or ***[Req 31]***, then the Split-SDK is categorised as a Limited Split-SDK. ~~Meaning~~This means that if the SDK Client ~~cannot complete~~is not capable of performing any one of the following ~~then it is categorised as a Limited-SDK~~:

~~•~~ ~~Verify~~Verification of the ACS Signed Content; OR

~~•~~ ~~Complete~~Completion of the ~~Diffie-Hellman~~DH exchange and ~~generate~~generation of the session keys used for the encryption/decryption of the CReq and CRes messages; OR

~~•~~ ~~Encrypt~~Encryption of the CReq message,

then it is categorised as a Limited Split-SDK.

A Limited Split-SDK shall:

[Req 34]

Set ~~the~~ SDK Type = ~~03 (Limited-SDK)~~(02 = Split-SDK), Split-SDK Variant, and Limited Indicator = Y.

4.4 Split-SDK Security Requirements

[Req 55]

Stop the transaction if it is not able to authenticate the Split-SDK -~~c~~Client or detects that the Split-SDK -Client is compromised.

The Split-SDK Server shall:

[Req 46]

Not store the content of a received CRes message beyond the current transaction.~~ or~~

[Req 814]

Not alter the content of a received CRes message.

Chapter 5 Split-SDK/Browser~~-SDK~~ Requirements

The Split-SDK/Browser~~-SDK~~ variant implements the same functions as the Split-SDK Client, as defined in Section 3.2.

The following sections describe only the differences between the Split-SDK/Browser~~-SDK~~ and the Split-SDK/Native in terms of transaction flow and security.

5.1 Split-SDK/Browser~~-SDK~~ Architecture

Figure 5-1 depicts a typical Split-SDK/Browser~~-SDK~~ component architecture. The architecture is identical to the Split-SDK/Native architecture (Figure 2-1), except that the Client is executed as JavaScript in a Browser.

Figure 5-1: Split-SDK/Browser~~-SDK~~ Component Architecture

Figure 5-1 was replaced and is not replicated in this bulletin.

5.2 Changes to the 3-D Secure Core Protocol Message Flow for a Split-SDK/Browser~~-SDK~~ Flow

~~The following steps replace the steps in the App flow in the EMV 3DS Protocol Specification v2.3.0.0. and later.~~ The authentication flow for a Split-SDK/Browser follows the same steps as the App-based flow defined in the *Core Specification* v2.3.1.0 and higher, with the exception of the following steps:

Step 2: The 3DS Requestor/3DS Server

The 3DS Server provides the following to the Split-SDK Server through the 3DS Requestor Environment:

- the Split-SDK Server URL ~~that is~~ used by the 3DS Requestor to establish ~~to~~the connection between the ~~Cardholder~~ Browser and the Split-SDK Server

The 3DS Requestor opens an iframe and redirects the iframe to the Split-SDK Server URL. The Split-SDK Server loads the Split-SDK/Browser~~-SDK~~ Client in the iframe within the ~~Cardholder~~ Browser, this iframe is used to display the ~~p~~Processing screen and, in case of challenge, the Split-SDK/Browser~~-SDK~~ renders the UI selected by the ACS.

Requirements 802, 803 and 804 follow unchanged.

The Split-SDK Server shall:

[Req 805]

Post the HTML code containing the ~~p~~Processing screen and Split-SDK/Browser~~-SDK~~ Client code to the iframe.

The Split-SDK/Browser~~-SDK~~ Client executes and establishes a secure connection with the Split-SDK Server.

The Split-SDK/Browser~~-SDK~~ Client shall:

[Req 806]

Verify that it is running within an iframe. If not, then the Split-SDK/Browser~~-SDK~~ Client reports an error to the Split-SDK Server and **ends processing**.

The Split-SDK Server provides to the Split-SDK/Browser~~-SDK the~~ Client:

- the Directory Server ID value
- the Message version (obtained from the 3DS Requestor Environment)

Note: After Step 2, the flow is similar to the App-based flow~~ in the Split-SDK~~.

Step 9: The 3DS Server

In a Split-SDK/Browser~~-SDK~~ implementation, the 3DS Server supports the same requirements as those defined in Section 3.1, Step 9 of the ~~EMV 3DS Protocol~~*Core Specification*, except that the next step is executed within a Browser instead of an App.

~~**Note:**~~ The note at the end of Step 9

"**The next step for a:**

- **Decoupled Authentication transaction~~, the next step~~ is Step 18**"

  is replaced ~~by~~with the following wording:

  "For a Decoupled Authentication transaction:

  [Req 807]

  The Split-SDK Server and the Split-SDK/Browser~~-SDK~~ Client shall proceed with Step 31 of the Split-SDK Message Flow.

  [Req 808]

  The 3DS Requestor shall close the challenge window.

  The next step is Step 18."

Step 24: The 3DS Requestor Environment

[Req 809]

Receive the final CRes message or Error Message from the ACS~~ and Validate as defined in Section 5.9.7 of the EMV 3DS Protocol Specification~~.

The following wording was added after Requirement 810 and before Requirement 811 in this section.

The Split-SDK Server closes the challenge process with the Split-SDK Client.

The Split-SDK/Browser Client shall:

[Req 815]

Clean up the SDK Ephemeral Key, the CEK used for CReq encryption and the CEK used for CRes decryption.

The Split-SDK Server shall:

[Req 816]

Clean up the CEK used for CRes decryption.

The Split-SDK Server conveys the result of the challenge process to the 3DS Requestor.

5.3 Changes to the Split-SDK Message Flow Requirements for a Split-SDK/Browser~~-SDK~~

Step 8: Assemble AReq Message Data

In Step 8~~,~~ of this ~~EMV Split SDK Specification, the~~specification, SDK Type = ~~04~~02 (Split-SDK) and Split-SDK Variant = 02 (Browser~~-SDK~~) for a transaction as defined in this ~~C~~chapter.

5.4 Split-SDK/Browser Security

This section describes the differences between the ~~Browser~~Split-SDK/Browser and the Split-SDK for the basic security requirements that are to be implemented by the ~~Browser~~Split-SDK/Browser.

5.4.1 SDK Initialisation Security Checks

The Split-SDK/Browser~~-SDK~~ of the Split-SDK Server shall:

Table 5.1: Split-SDK/Browser~~-SDK~~ Initialisation Security Checks

5.4.2 SDK Server CSP and CORS Guidelines

The protection of Split-SDK content (Split-SDK/Browser~~-SDK~~) from a Merchant is ensured by the iframe used to insulate Split-SDK content from Merchant content.

5.4.3 Iframe and Sandbox Attributes

Table 5.2 specifies the iframe attributes ~~that~~used by the 3DS Requestor ~~uses~~ when ~~it~~ ~~creates~~creating the Split-SDK/Browser~~-SDK~~ iframe.

Table 5.2: iframe Attributes

Attribute	Value
`~~W~~width`	as per ~~c~~Challenge ~~w~~Window ~~s~~Size

Table 5.3 specifies the sandbox attributes ~~that~~used by the 3DS Requestor ~~uses~~when ~~it~~ ~~creates~~creating the Split-SDK/Browser~~-SDK~~ iframe.

The following wording was added as Section 5.5.

5.5 Split-SDK/Browser User Interface

The Split-SDK/Browser implements the 3-D Secure User Interface Templates as defined in Section 4 of the *Core Specification* v2.3.1.0.

The Split-SDK/Browser displays the different UI Template zones (refer to Figure 4.1) – in particular, the Header zone with the Cancel action button and the Header text.

Chapter 6 Split-SDK/Shell~~-SDK~~ Requirements

The Split-SDK/Shell~~-SDK~~ variant implements the same functions as the Split-SDK Client, as defined in Section 3.2.

The following sections describe only the differences between the Split-SDK/Shell~~-SDK~~ and the Split-SDK/Native in terms of transaction flow and security.

6.1 Split-SDK/Shell~~-SDK~~ Architecture

Figure 6-1 depicts a typical Split-SDK/Shell~~-SDK~~ component architecture.

The Split-SDK/Shell~~-SDK~~ component architecture is identical to the Split-SDK architecture (Figure 2-1), except that the SDK Client utilises a wrapped WebView component for both App-based Native and App-based HTML ~~c~~Challenge ~~f~~Flows.

Unlike the Default-SDK, which renders the Native UI using platform-specific display elements (i.e., button, textbox, text label, etc.), the Split-SDK/Shell~~-SDK~~ renders the Native UI challenges using a JavaScript (JS Split-SDK/Shell~~-SDK~~) running in a WebView.

Figure 6-1: Split-SDK/Shell~~-SDK~~ Component Architecture

Figure 6-1 was replaced and is not replicated in this bulletin.

6.2 Changes to the 3-D Secure Core Protocol User Interface Requirements and Guidelines for a Split-SDK/Shell~~-SDK~~ ~~f~~Flow

The following items replace the requirements for the App-based User Interface in the ~~EMV 3DS Protocol~~Core Specification v2.3.~~0~~1.0 and ~~later~~higher.

6.2.1 Processing Screen Requirements

The 3DS Requestor App or the ~~3DS~~ Split-SDK/Shell~~-SDK~~ shall, for the AReq/ARes message exchange:

~~[Req 145]~~ [Req 822]

Display the Processing screen supplied by the 3DS SDK during the entire AReq/ARes message cycle.

~~[Req 146]~~ [Req 823]

Display the Processing screen for a minimum of two seconds.

6.3 Changes to the Split-SDK Message Flow Requirements for a Split-SDK/Shell~~-SDK~~

The Split-SDK/Shell Server and Client mutually authenticate before the Client loads the JS Split-SDK/Shell. When loaded and running, the JS Split-SDK/Shell mutually authenticates with the Split-SDK Server.

A Split-SDK/Shell authentication flow follows the same steps as the Split-SDK/Native flow defined in this document, with the exception of the following steps:

Step 2 Get SDK Ephemeral Public Key

[Req 48]

The Split-SDK/Shell~~-SDK~~ shall:

- Implement a mutual authentication protocol and an encryption protocol that protects the data exchanged between the Split-SDK/Shell~~-SDK~~ Client and the Split-SDK Server.
- Retrieve the information related to the transaction, the HTML for the ~~p~~Processing screen, and the JavaScript of the JS Split-SDK/Shell~~-SDK~~ from the Split-SDK Server.

[Req 49]

The Split-SDK/Shell~~-SDK~~ shall:

- Initiali~~z~~se for a new 3DS transaction
- Create and protect a WebView
- Post the HTML and JavaScript in WebView

[Req 50]

The JS Split-SDK/Shell~~-SDK~~ shall:

- Implement a mutual authentication protocol and an encryption protocol that protects the data exchanged between the JS Split-SDK/Shell~~-SDK~~ and the Split-SDK Server.
- Send the SDK Ephemeral Public Key to the SDK Server (which could ~~be~~occur through Split-SDK/Shell~~-SDK,~~ or directly through a secured connection to the Split-SDK Server).

Step 8: Assemble AReq Message Data

In Step 8~~,~~ of this ~~EMV Split SDK Specification, the~~specification, SDK Type = ~~05~~02 (Split-SDK) and Split-SDK Variant = 03 (Shell~~-SDK~~) for a transaction as defined in this ~~C~~chapter.

Step 11b Close (Frictionless only)

The JS Split-SDK/Shell~~-SDK~~ shall:

~~[Req 8]~~ [Req 818]

Clean up the SDK Ephemeral Key.

The Split-SDK/Shell~~-SDK~~ shall:

[Req 51]

Close the WebView.

The Split-SDK Server shall:

~~[Req 32]~~ [Req 819]

Clean up the CEK used for CRes decryption.

Step 31 Close

The Split-SDK Server closes the challenge process with the Split-SDK/Shell~~-SDK~~.

The JS Split-SDK/Shell~~-SDK~~ shall:

~~[Req 31]~~ [Req 820]

Clean up the SDK Ephemeral Key, the CEK used for CReq encryption and the CEK used for CRes decryption.

The Split-SDK/Shell~~-SDK~~ shall:

[Req 52]

Close the WebView.

The Split-SDK Server shall:

~~[Req 32]~~ [Req 821]

Clean up the CEK used for CRes decryption.

6.4 SDK Security

This section describes the differences between the Split-SDK/Shell~~-SDK~~ and the Split-SDK for the basic security requirements that are to be implemented by the Split-SDK/Shell~~-SDK~~.

6.4.1 SDK Initialisation Security Checks

The Split-SDK/Shell~~-SDK~~ of the Split-SDK Server shall conduct the security checks as defined in **[Req 42]** and Table 4.1 ~~of~~in Section 4.2 of this ~~EMV Split-SDK S~~specification.

6.4.2 Split-SDK/Shell~~-SDK~~ Security

The protection of Split-SDK content (Split-SDK/Shell~~-SDK~~) from a Merchant is ensured by wrapping the WebView component to prevent direct access by the Merchant.

Additionally, the Split-SDK/Shell~~-SDK~~ ~~c~~Client should only allow the WebView component to run scripts and load other external resources (e.g., CSS files) that have been loaded from the Split-SDK Server or from within the Split-SDK/Shell~~-SDK~~ package itself.

[Req 53]

The Split-SDK/Shell~~-SDK~~ shall set a WebView that:

- allows the execution of the JS Split-SDK/Shell~~-SDK~~
- prevents the 3DS Requestor App or other external application ~~to access~~from accessing the content of the WebView.

[Req 54]

The Split-SDK/Shell~~-SDK~~ shall:

- load and run the JS Split-SDK/Shell~~-SDK~~ provided by the Split-SDK Server
- not accept other external sources for the JS Split-SDK/Shell~~-SDK~~ package, or any other codes (HTML, CSS).

Navigation attempts from within the Split-SDK/Shell~~-SDK~~ WebView are captured by the JS Split-SDK/Shell~~-SDK~~ and processed internally, rather than being passed to the operating system and network stack. In addition to navigation attempts, the 3DS SDK also captures external resource requests (image loads, external JS scripts, CSS, etc.).

For the App-based Native flow, the Split-SDK/Shell~~-SDK~~ should allow loading of image URLs (Issuer Image, Payment System Image) coming from an external source.

[Req 817]

~~In case~~ If the Split-SDK/Shell~~-SDK opens~~ uses an iframe in the WebView, it shall use the setting ~~as~~ defined in Table 5.2 to open the iframe.

Legal Notice