EMV® Specification Bulletin No. 214 v3
June 2023

EMV® 3-D Secure Protocol and Core Functions Specification version 2.2.0 – Updates, Clarifications & Errata

This Specification Bulletin No. 214 v3 provides updates, clarifications and errata incorporated into the EMV 3-D Secure Protocol and Core Functions Specification since version 2.2.0.

Applicability

This Specification Bulletin applies to:

- *EMV® 3-D Secure Protocol and Core Functions Specifications*, *Version 2.2.0*

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.

Effective Date

June 2023

Contents

EMV® 3-D Secure Updates, Clarifications & Errata 1
Applicability 1
Effective Date 1
June 2023 v3 6
Overview and Objectives 6
Chapter 1 Introduction 6
1.5 Definitions 6
Table 1.3 Definitions 6
1.8 Supporting Documentation 7
Chapter 3 EMV 3-D Secure Authentication Flow Requirements 7
3.3 Browser-based Requirements 7
Step 10: The 3DS Server 7
[Req 117] 7
Step 11: The ACS 7
[Req 119] 7
[Req 442] 8
Step 12: The ACS and Browser 8
[Req 307] 8
[Req 122] 8
Chapter 5 EMV 3-D Secure Message Handling Requirements 8
5.8 Browser-based Message Handling 8
5.8.1 3DS Method Handling 8
[Req 261] 8
[Req 263] 9
5.8.2 Browser Challenge ~~Window~~ iframe Requirements 9
[Req 265] 9
[Req 266] 9
[Req 267] 9
[Req 268] 9
[Req 324] 9
[Req 269] 9
[Req 270] 10
Annex A 3-D Secure Data Elements 11
A.4 EMV 3-D Secure Data Elements 11
Table A.1 EMV 3-D Secure Data Elements 11
A.8 UI Data Elements 13
Table A.18: UI Data Elements 13

A.9 iframe and Sandbox Attributes....14
February 2020 v2....16
Chapter 1 Introduction....16
1.10 Constraints....16
Chapter 3 3-D Secure Authentication Flow Requirements....16
Step 14: The 3DS SDK....16
[Req 55]....16
Chapter 4 EMV 3-D Secure User Interface Templates, Requirements, and Guidelines....16
4.1.3 3-D Secure Interface Templates....16
[Req 395]....17
[Req 358]....17
[Req 359]....17
4.2.1 Processing Screen Requirements....18
4.2.1.1 3DS SDK/3DS Requestor App....19
[Req 143]....19
[Req 145]....19
[Req 389]....19
4.2.2.1 3DS SDK/ACS....19
[Req 362]....19
[Req 398]....19
[Req 369]....19
4.2.3 Native UI Templates....20
4.2.4.3 3DS SDK....23
[Req 154]....23
[Req 157]....23
4.2.5.1 3DS SDK/ACS....23
[Req 373]....23
[Req 374]....23
4.2.6 HTML UI Templates....23
4.2.6.1 HTML Other UI Template....24
4.2.7.3 3DS SDK....25
[Req 171]....25
4.3.2.1 ACS....25
[Req 380]....25
Chapter 5 EMV 3-D Secure Message Handling Requirements....25
5.1.5 Data Version Numbers....25
[Req 396]....25
[Req 397]....25
5.8.1 3DS Method Handling....26

[Req 263] 26
Chapter 6 EMV 3-D Secure Security Requirements 26
6.1.4.1 For App-based CReq/CRes 26
Annex A 3-D Secure Data Elements 27
A.4 EMV 3-D Secure Data Elements 27
Table A.1 EMV 3-D Secure Data Elements 27
A.5.4 Browser CReq and CRes POST 29
A.8 UI Data Elements 29
Table A.18: UI Data Elements 29
June 2019 v1 31
Chapter 1 Introduction 31
1.5 Definitions 31
Table 1.3 Definitions 31
Chapter 3 EMV 3-D Secure Authentication Flow Requirements 31
3.1 App-based Requirements 31
[Req 355] 31
[Req 345] 31
[Req 346] 32
3.3 Browser-based Requirements 32
[Req 347] 32
[Req 348] 32
3.4 3RI-based Requirements 32
[Req 353] 32
[Req 354] 33
Chapter 4 EMV 3-D Secure User Interface Templates, Requirements, and Guidelines 33
4.1 3-D Secure User Interface Templates 33
[Req 342] 33
4.2 App-based User Interface Overview 33
4.2.1 Processing Screen Requirements 33
4.2.1.1 3DS SDK/3DS Requestor App 34
[Req 143] 34
[Req 145] 34
[Req 388] 34
~~[Req 360]~~ 34
~~[Req 361]~~ 34
[Req 389] 34
4.2.3 Native UI Templates 39
(Original) Figure 4.14 Sample Whitelisting Information Text—PA 39
(Updated) Figure 4.14 Sample Whitelisting Information Text—PA 40

4.2.4.1 3DS SDK 40
[Req 153] 40
4.2.7.3 3DS SDK 40
[Req 171] 40
Chapter 5 EMV 3-D Secure Message Handling 41
5.1.3 Base64/Base64url Encoding 41
[Req 193] 41
5.1.6 Message Content Validation 41
[Req 309] 41
Example: 41
Chapter 6 EMV 3-D Secure Security Requirements 42
6.2.2.1 3DS SDK Encryption 42
6.2.2.2 DS Decryption 43
6.2.3.2 ACS Secure Channel Setup 44
6.2.3.3 3DS SDK Secure Channel Setup 44
6.2.4.1 3DS SDK—CReq 45
6.2.4.2 3DS SDK—CRes 45
6.2.4.3 ACS—CReq 45
6.2.4.4 ACS—CRes 46
Annex A 3-D Secure Data Elements 47
A.4 EMV 3-D Secure Data Elements 47
Table A.1 EMV 3-D Secure Data Elements 47
A.5.7 Card Range Data 53
Table A.6 Card Range Data 53
A.7.3 3DS Requestor Information 54
A.7.4 3DS Requestor Prior Transaction Authentication Information 54

Overview and Objectives

The 3-D Secure Browser Flow is used to process transactions where the Cardholder has initiated interactions with the 3DS Requestor through a Browser. Adherence by the 3DS Requestor and the ACS to the iframe requirements documented in this 3-D Secure (3DS) Specification Bulletin is critical to the successful processing of the Browser Flow.

The new requirements for 3-D Secure (3DS) Specification v2.2.0 are the same as for v2.3.1.1, therefore, the 3DS Requestor and ACS should have a consistent implementation for the Browser flow and the Challenge.

The bulletin also provides clarifications and additional requirements in case the ACS receives multiple CReq messages for the Browser flow – for example, if the 3DS Requestor refreshes the Merchant web page during the Challenge, in case the Cardholder requested a page refresh on their Browser.

Chapter 1 Introduction

1.5 Definitions

Table 1.3 Definitions

Term	Definition
Browser	A Browser is a dedicated software application for accessing information on the World Wide Web, for example Chrome, Safari, Edge, Firefox. When a user requests a web page from a particular website, the Browser retrieves the necessary content from a web server and then displays the page on the consumer's screen. In the context of 3-D Secure, the Browser is a conduit to transport messages between the Acquirer Domain and the Issuer Domain. A Browser is distinguished from a UI component, for example, a WebView, or Custom Tabs, which can be used to display content within an App on a mobile device. The Browser flow is invoked by a Browser whereas the EMVCo specification does not support a UI component within an app invoking the Browser flow. ~~In the context of 3-D Secure, the browser is a conduit to transport messages between the 3DS Server (in the Acquirer Domain) and the ACS (in the Issuer Domain).~~
Universal App Link	Operating System-registered HTTPS links for opening a specific mobile app, installed on a device. The implementation is platform-specific. • Android App Links: https://developer.android.com/training/app-links • iOS Universal Links: https://developer.apple.com/ios/universal-links

1.8 Supporting Documentation

- *EMV® 3-D Secure Browser Flow Best Practices*

Chapter 3 EMV 3-D Secure Authentication Flow Requirements

3.3 Browser-based Requirements

Step 10: The 3DS Server

The 3DS Server shall:

[Req 117]

For a transaction with a challenge (Transaction Status = C):

a. Evaluate, based in part on the 3DS Requestor Challenge Indicator, the ACS Challenge Mandated Indicator and the ACS Rendering Type whether to perform the requested challenge.
   - If the 3DS Requestor accepts the challenge:
     - Send necessary information (as defined in Table B.2) from the ARes message to the 3DS Requestor Environment.
     - Continue with step b through e of this requirement and then Step 11.
   - If the 3DS Requestor continues without performing the requested challenge, receive the RReq message from the DS and Validate as defined in Section 5.9.9. If the message is in error, the 3DS Server **ends processing**. Format the RRes message as defined in Table B.9 and send to the DS. Further processing is outside the scope of 3-D Secure processing. The 3DS Server may continue with Step 22.

b. Format the CReq message according to the format specified in Table B.3 for a Browser-based implementation.

c. Base64url-encode the CReq message.

d. Construct a form containing the CReq message, and if provided by the 3DS Requestor, the 3DS Requestor Session Data (as defined in Table A.3).

e. ~~Pass~~ Send the CReq message using an HTTP POST through the Cardholder Browser HTML iframe as defined in Section 5.8.2 and Section A.5.4 (Browser CReq and CRes POST) to the ACS URL received in the ARes message,~~by causing the cardholder browser to POST the form to the ACS URL~~ using a server-authenticated TLS link as defined in Section 6.1.4.2.

Step 11: The ACS

The ACS shall:

[Req 119]

Receive the CReq message from the Browser ~~and~~:

- Accept a Base64url-encoded CReq message with or without padding,
- Validate the message as defined in Section 5.9.6,
- Accept the Base64url-encoded Session Data with or without padding.

If the message is in error, the ACS **ends processing**.

New Requirement 442 was added at the end of Step 11, directly after Requirement 121.

[Req 442]

If the ACS receives more than one CReq message, the ACS either:

- Restarts or continues the challenge with the Cardholder, OR
- Returns an Error Message if it is not possible to continue or restart the authentication.

Step 12: The ACS and Browser

The ACS shall:

[Req 307]

The ACS shall not lead the Cardholder outside of the authentication flow by redirecting to any registration or marketing pages. Any redirection shall be used for authentication purposes only and within the iframe. The ACS shall only load external resources that are needed to improve the Cardholder authentication experience and security (e.g., logos).

[Req 122]

Send the ACS UI to the Cardholder over the channel established by the HTTP POST in Step 10. The ACS shall allow the content of the UI to be framed. The Browser displays the ACS UI to the Cardholder.

Chapter 5 EMV 3-D Secure Message Handling Requirements

5.8 Browser-based Message Handling

5.8.1 3DS Method Handling

The 3DS Requestor shall:

[Req 261]

~~Render a hidden HTML iframe in the Cardholder browser and send a form with a field named `threeDSMethodData` containing the JSON Object via HTTP POST to the ACS 3DS Method URL obtained from the PRes message cache data.~~

Open a hidden HTML iframe in the Cardholder Browser with:

- the iframe attributes set as defined in Table A.19.
- the sandbox attributes set as defined in Table A.20.

For Browser compatibility, iframe shall be made hidden with the following style setting: “visibility: hidden”. For example: style="`visibility:hidden`".

Send a form with a field named `threeDSMethodData` containing the JSON object via HTTP POST to the ACS 3DS Method URL obtained from the PRes message cache data.

The ACS shall:

[Req 263]

~~Recall the 3DS Server Transaction ID received in the initial 3DS Method POST~~Validate the Base64url-encoded `threeDSMethodData` with or without padding from the initial 3DS POST method, and retrieve the 3DS Server Transaction ID, then Base64url-encode the JSON object and send via a form with a field named `threeDSMethodData` in the Cardholder Browser HTML iframe using an HTTP POST method to the 3DS Method Notification URL. Refer to Table A.2 for detailed information about 3DS Method Data.

5.8.2 Browser Challenge ~~Window~~ iframe Requirements

The Browser challenge will occur within the Cardholder Browser, and the ACS will provide a formatted challenge UI to the Cardholder within the Browser challenge ~~window~~iframe.

The 3DS Requestor shall:

[Req 265]

Select the size of the HTML iframe to be generated by the 3DS Requestor from one of the ~~window~~iframe sizes specified in the Challenge Window Size data element.

[Req 266]

Use a server-authenticated TLS session as defined in Section 6.4.1.2.

[Req 267]

Create a 3-D Secure challenge ~~window~~iframe by generating a CReq message, creating an HTML iframe in the Cardholder Browser, with the following settings:

- iframe attributes as defined in Table A.19
- sandbox attributes as defined in Table A.20

and ~~generating~~generate an HTTP POST through the iframe to the ACS URL that was received in the ARes message.

[Req 268]

Post the CReq message containing the selected size in the Challenge Window Size data element to the ACS as defined in Table A.1.

[Req 324]

Provide a fallback mechanism for redirection in environments that do not support JavaScript.

Note: If the Cardholder initiates a page refresh, the 3DS Requestor repeats the previous steps starting from [Req 265] using the same iframe size and attributes.

The ACS shall:

[Req 269]

Receive the CReq message and respond with the HTML to render the challenge user interface within the iframe.

Note: During completion of the challenge by the Cardholder, there may be several interactions required.

After challenge completion, the ACS generates the RReq message. After receiving the corresponding RRes message, the ACS generates the CRes message and invokes the Browser to send an HTTP POST (for example, using JavaScript) to the Notification URL containing the CRes message as defined in Table A.1. This completes the challenge.

The 3DS Requestor shall:

[Req 270]

Close the challenge ~~window~~iframe upon receiving the CRes message by refreshing the parent page and removing the HTML iframe.

Annex A 3-D Secure Data Elements

Some sections of Annex A were moved within the annex and new sections were also added. Please be advised that heading and table numbering have been updated since previous versions of the specification.

A.4 EMV 3-D Secure Data Elements

Table A.1 EMV 3-D Secure Data Elements

Data Element/ Field Name	Description	Source	Length/Format/ Values	Device Channel	Message Category	Message Inclusion	Conditional Inclusion
3DS Requestor App URL Field Name: `threeDSRequestorAppURL`	3DS Requestor App declaring ~~their~~its URL within the CReq message so that the Authentication App can call the 3DS Requestor App after OOB authentication has occurred. Each transaction would require a unique Transaction ID by using the SDK Transaction ID.		Length: Variable, maximum 256 characters JSON Data Type: String Value accepted: • Fully Qualified URL OR • Universal App Link **Note: it is recommended to use Universal App Link.** Example value:				Required ~~if 3DS Requestor App URL is provided by the 3DS Requestor App.~~ in all CReq messages in 3DS Requestor App URL is provided by the 3DS SDK.

Data Element/ Field Name	Description	Source	Length/Format/ Values	Device Channel	Message Category	Message Inclusion	Conditional Inclusion
	Note: When providing the 3DS Requestor App URL as a Universal App Link, the 3DS Requestor needs to properly register the URL with the Operating System.		https://appname.com~~?transID=b2385523-a66c-4907-ac3c-91848e8c0067~~ Refer to Table 1.3 for Universal App Link definition.				
Whitelisting Data Entry							~~If Whitelisting Information Text was present in the CRes message, SDK must provide his data element to the ACS in the CReq message.~~ Required if • Whitelisting Information Text was present in the preceding CRes message AND

Data Element/ Field Name	Description	Source	Length/Format/ Values	Device Channel	Message Category	Message Inclusion	Conditional Inclusion
							• Challenge Cancelation Indicator is not present

A.8 UI Data Elements

Table A.18: UI Data Elements

Data Element	Field Name	Zone	Portrait Top-down Display Order	Landscape Top-down Display Order	ACS UI Type			
					OTP	Single Select	Multi Select	OOB
Expandable Information Label	expandInfoLabel	4	12	10 or 9	O	O	O	O
Expandable Information Text	expandInfoText	4	13	11 or 10	O	O	O	O
Why Information Label	whyInfoLabel	4	10	8 or 7	O	O	O	O
Why Information Text	whyInfoText	4	11	9 or 8	O	O	O	O

A.9 iframe and Sandbox Attributes

Table A.19 specifies the iframe attributes that the 3DS Requestor uses when it creates the challenge or 3DS Method iframe.

Table A.19: iframe Attributes

Attribute[14]	Value
`allowfullscreen`	false
`allowpaymentrequest`	false
`height`	as per Challenge Window Size
`sandbox`	refer to Table A.20
`srcdoc`	may be used to initialise redirection content
`width`	as per Challenge Window Size
`allow="payment *; publickey-credentials-get *"`[15]	enable access to WebAuthn and SPC (Secure Payment Confirmation) API and Payment Request API

Table A.20 specifies the sandbox attributes that the 3DS Requestor uses when it creates the challenge or 3DS Method iframe.

Table A.20: Sandbox Attributes

Attribute	Description	Inclusion
`allow-forms`	Allows the processing of forms.	R
`allow-scripts`	Allows the processing of scripts. Note the `allow-scripts` permission does not give the iframe the ability to create pop-ups or modal windows, which can help prevent clickjacking attacks from occurring.	R
`allow-same-origin`	Gives the iframe permission to only use the data from the same ACS domain.	R

[14] Attributes not listed in Table A.19 should not be present.

[15] Use the following syntax <iframe src="https://www.foo.com" allow="payment; publickey-credentials-get *"></iframe>, if supported by the Browser.

Attribute	Description	Inclusion
allow-pointer-lock	Gives access to the mouse position and events. Note: this attribute is not needed for the 3DS Method iframe.	R
allow-downloads-without-user-activation	Prevents downloads to be initiated for content in the iframe without user action.	Not Allowed
allow-downloads	Prevents downloads to be initiated for content in the iframe.	Not Allowed
allow-modals	Prevents to open modal window from the iframe.	Not Allowed
allow-orientation-lock	Prevents to lock the screen orientation.	Not Allowed
allow-popups	Prevents pop-up windows.	Not Allowed
allow-popups-to-escape-sandbox	Prevents pop-ups to open new windows without inheriting the sandboxing.	Not Allowed
allow-presentation	Prevents to initiate a presentation session.	Not Allowed
allow-storage-access-by-user-activation	Prevents access to the parent's storage capabilities.	Not Allowed
allow-top-navigation	Prevents access to the top-level browsing context.	Not Allowed
allow-top-navigation-by-user-activation	Prevents access to the top-level browsing context also with user interaction.	Not Allowed

February 2020 v2

Chapter 1 Introduction

1.10 Constraints

The Specification or any implementation of the Specification is not intended to replace or interfere with any international, regional, national or local laws and regulations; those governing requirements supersede any industry standards.

Chapter 3 3-D Secure Authentication Flow Requirements

Step 14: The 3DS SDK

[Req 55]

Display the UI based upon the ACS UI Type selected and the data elements populated. Refer to Section 4.2 and to the applicable 3DS SDK specification for UI details.

If the CRes message for a Native UI contains a URL(s) directing the 3DS SDK to fetch data from an external server (i.e., an Issuer Image or Payment System Image for use with a Native UI), establish an additional secure link to the external server as defined in Section 6.1.4.1 and fetch and display the received data within the UI.

If a secure link cannot be established, the 3DS SDK proceeds with the challenge displays all other provided mandatory and optional UI data elements and does not send an error message to the ACS.

Chapter 4 EMV 3-D Secure User Interface Templates, Requirements, and Guidelines

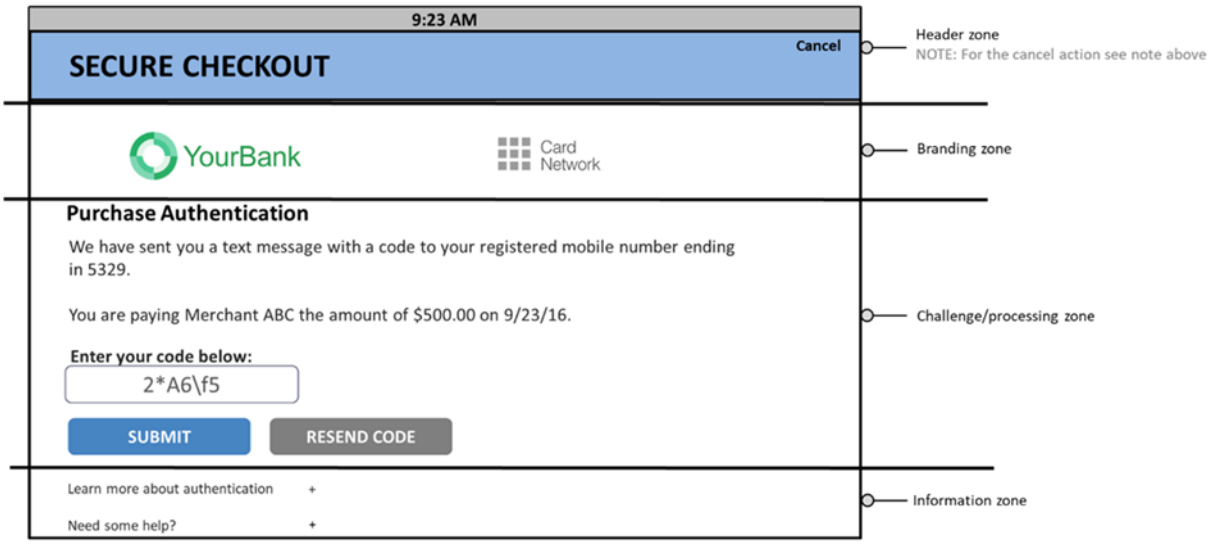
This SB 214v2 contains new Figures added to Chapter 4 of the 3-D Secure Protocol and Core Functions specification. Existing graphics with no updates were renumbered accordingly and are not included in this bulletin.

4.1.3 3-D Secure Interface Templates

Figure 4.2 illustrates the zones and placement of UI data elements within the zones in landscape mode.

Figure 4.2: UI Template Zones—Landscape

Note: The Cancel action can be implemented as a function on a controller for the platform.



The 3DS SDK shall:

[Req 395]
Support the UI template orientation(s) (i.e., portrait and landscape) according to the device capabilities.

[Req 358]
For the Native UI Type, display UI data elements provided by the ACS within the applicable zones as defined in Table A.18 and depicted in Figure 4.1 and Figure 4.2. The expected format is depicted in Sections 4.2.3 and 4.2.6.

The ACS shall:

[Req 359]
For the App-based HTML UI Type and Browser-based UI, create HTML with form elements within the applicable zones as outlined in Figure 4.1 and Figure 4.2. The format is outlined in Sections 4.2.6 and 4.3.3.

Figure 4.3~~Figure 4.2~~ through Figure 4.4 illustrates the consistency of the look and feel across device channels and implementations.

Figure 4.4 illustrates the consistency of the UI in landscape mode.

Figure 4.4: UI Template Examples—App-based—Landscape

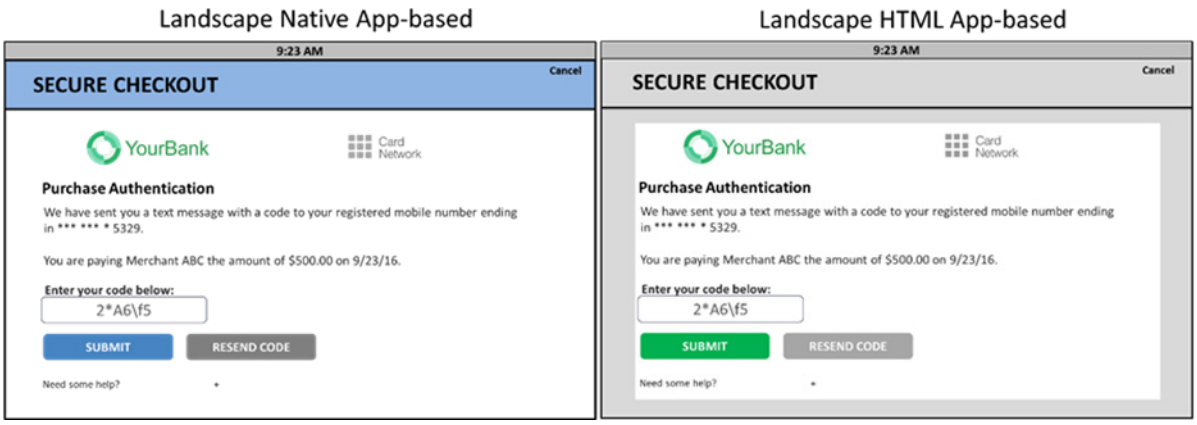
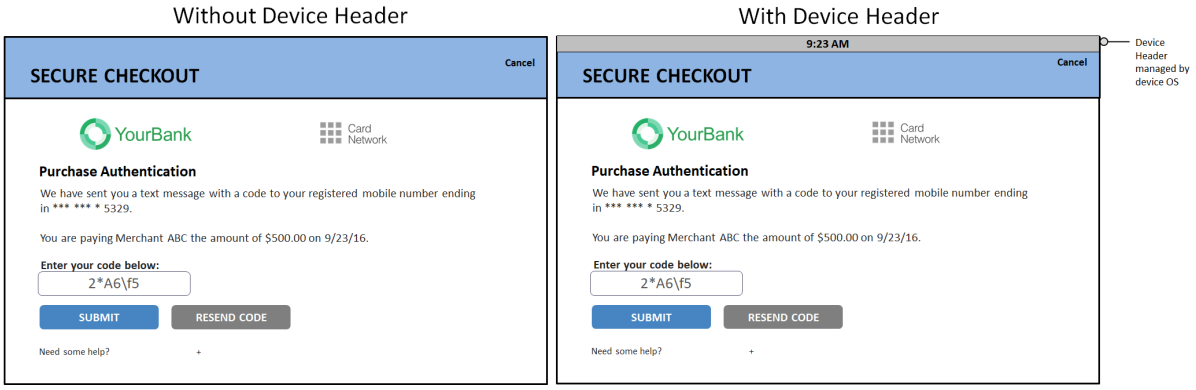


Figure 4.5: Sample Native UI OTP/Text Template with/without Device Header

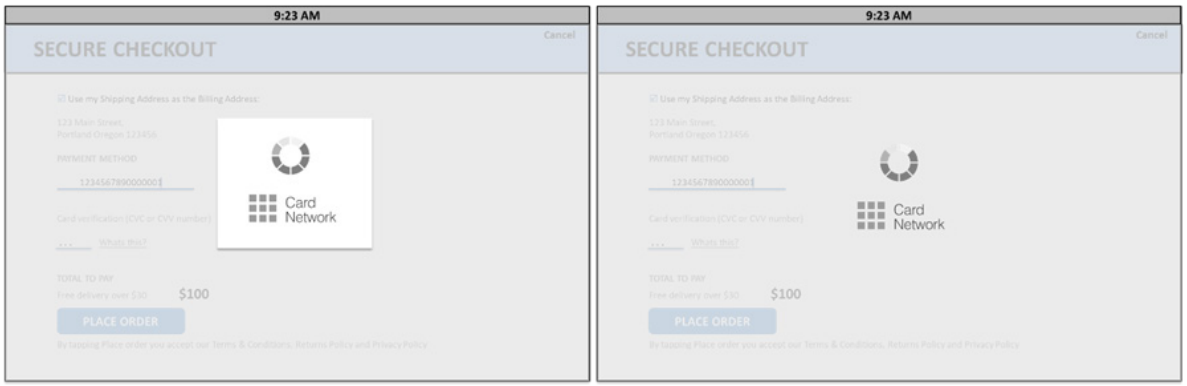
Note: The device header is optional and may not be present depending on the 3DS Requestor implementation and OS constraints. Figure 4.5 depicts sample formats with or without a device header.



4.2.1 Processing Screen Requirements

Figure 4.7~~Figure 4.4~~ and Figure 4.8 provides a sample formats for the App-based Processing screen that contains both the Processing Graphic and the Logo.

Figure 4.8: Sample App-based Processing Screen—Landscape



4.2.1.1 3DS SDK/3DS Requestor App

The 3DS SDK shall for the AReq/ARes message exchange:

[Req 143]

Integrate the Processing Graphic and if requested, the DS logo into the centre of the Processing screen as depicted in Figure 4.7~~Figure 4.4~~ and Figure 4.8 with or without a white box.

The 3DS Requestor App shall for the AReq/ARes message exchange:

[Req 145]

Display the Processing screen supplied by the 3DS SDK during the entire AReq/ARes message cycle and overlay on the merchant checkout page including the overlay the Header Zone as depicted in Figure 4.7~~Figure 4.4~~ and Figure 4.8.

The 3DS SDK shall for the CReq/CRes message exchange:

[Req 389]

Ensure that the Cancel action is not actionable while displaying ~~on~~ the Processing screen.

Figure 4.9 provides a sample format for the App-based processing flow. Figure 4.10 provides a sample format for the Out of Band template and 3DS Requestor App on the same device for an App-based processing flow. Figure 4.11 provides a sample format for the Decoupled Authentication Flow.

4.2.2.1 3DS SDK/ACS

The 3DS SDK shall for the CReq/CRes message exchange:

[Req 362]

For the ACS UI Type and the device screen orientation, display the supported UI template in its applicable orientation and the supported data elements in their applicable zones and order as defined in Table A.18 and depicted in Figure 4.1 and Figure 4.2. The expected format is depicted in Sections 4.2.3 and 4.2.6.

If the 3DS SDK receives an unsupported UI data element(s) for this ACS UI Type, the 3DS SDK does not display the UI data elements, proceeds with the challenge and does not send an error message to the ACS.

Req 398 is a new requirement to align with the existing implementation of the 3DS SDK; no impact is expected to the 3DS SDK.

[Req 398]

For the ACS UI Type, the 3DS SDK returns to the ACS an Error Message (as defined in Section A.5.5) with Error Component = S and Error Code = 201 if any mandatory UI data elements are missing as defined in Table A.18.

[Req 369]

~~Display~~ Provide the Cancel action. This can be implemented as a button in the top corner of the header zone as depicted in Figure 4.1 and Figure 4.2 and/or as a function on a controller for the platform.

The ACS shall for the CReq/CRes message exchange:

[Req 387]
Only include the mandatory and the optional ACS-chosen UI data elements for the selected ACS UI Type as defined in Table A.18.

4.2.3 Native UI Templates

Figure 4.12~~Figure 4.4~~ through Figure 4.23~~Figure 4.9~~ depict sample Native UI Templates. The UI content is provided by the ACS in the CRes message which contains the information needed to properly display the UI.

Figure 4.12~~Figure 4.8~~ and Figure 4.13 provide sample formats for a one-time passcode (OTP)/Text during a Payment Authentication transaction. This sample UI provides a format using expandable fields for additional information.

Figure 4.13: Sample Native UI OTP/Text Template—PA—Landscape

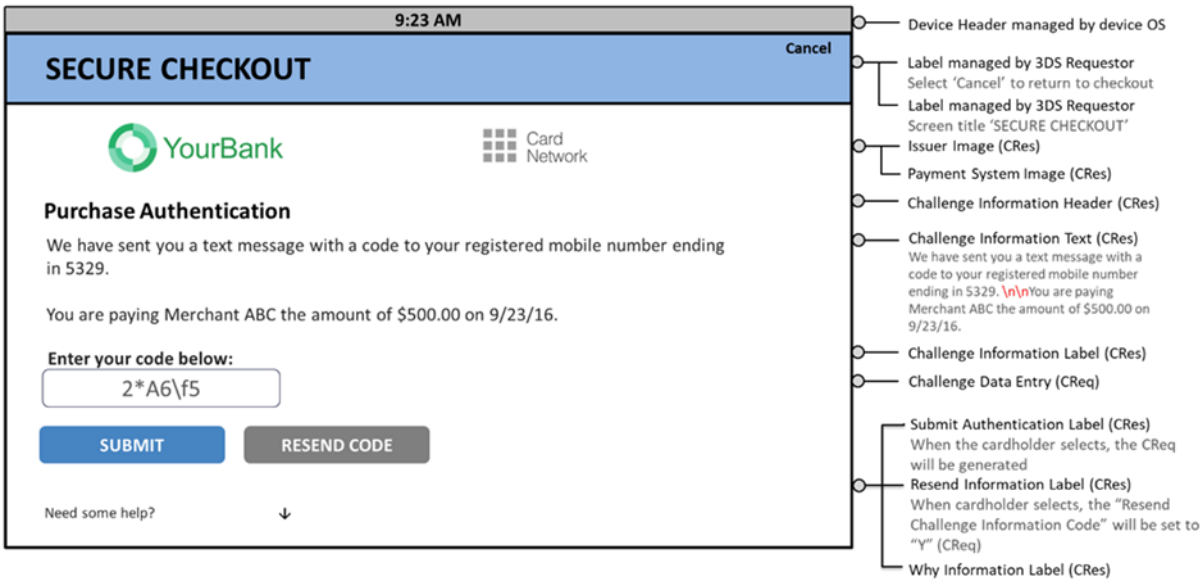


Figure 4.15~~Figure 4.10~~ and Figure 4.16 provide sample formats that allows multiple options to be presented to the Cardholder to obtain single response. For example, asking the Cardholder if they prefer the OTP to be sent to the Consumer Device or to the email address on file.

Note: To optimise the Cardholder experience, the Challenge Selection Information can be displayed horizontally or vertically in landscape.

Figure 4.16: Sample Native UI—Single-select Information—PA—Landscape

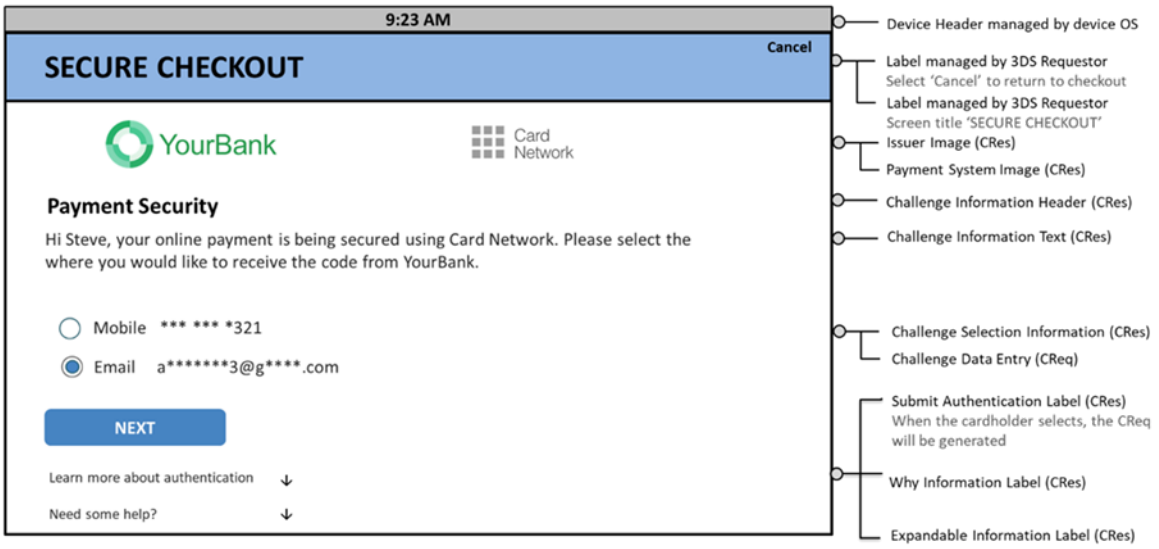


Figure 4.17~~Figure 4.11~~ and Figure 4.18 provide sample formats that allows multiple options to be presented to the Cardholder to obtain multiple responses on a single screen. For example, asking the Cardholder to select the cities where they have lived. This example also depicts a screen with no Issuer or Payment System branding.

Note: To optimise the Cardholder experience, the Challenge Selection Information can be displayed horizontally or vertically in landscape.

Figure 4.18: Sample Native UI—Multi-select Information—PA—Landscape

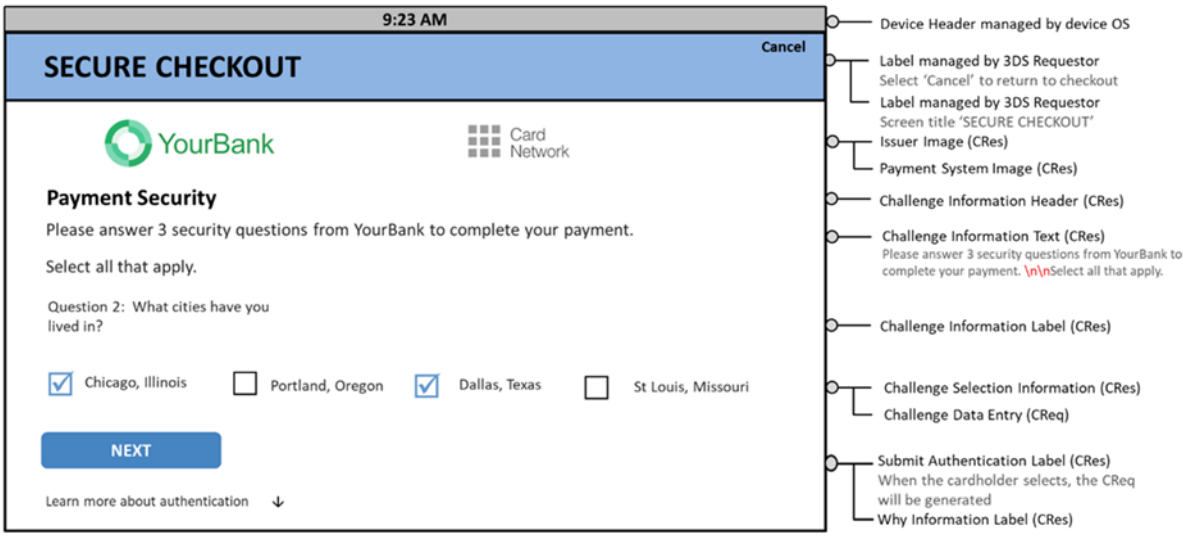


Figure 4.19~~Figure 4.12~~ and Figure 4.20 provide sample OOB formats to display instructions to the Cardholder.

Figure 4.20: Sample OOB Native UI Template—PA—Landscape

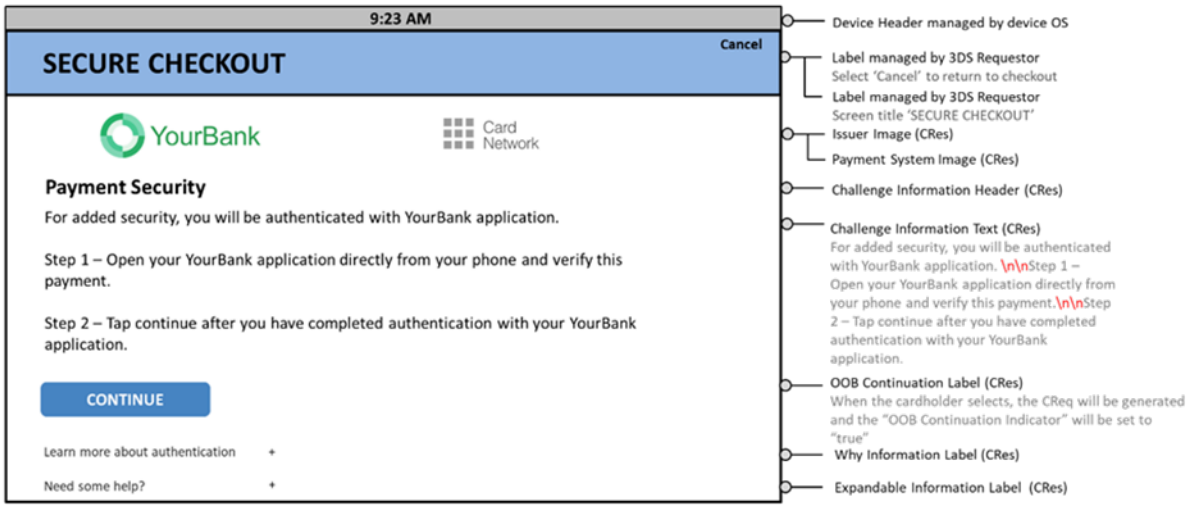


Figure 4.21 Sample Challenge Information Text Indicator—PA (Updated)

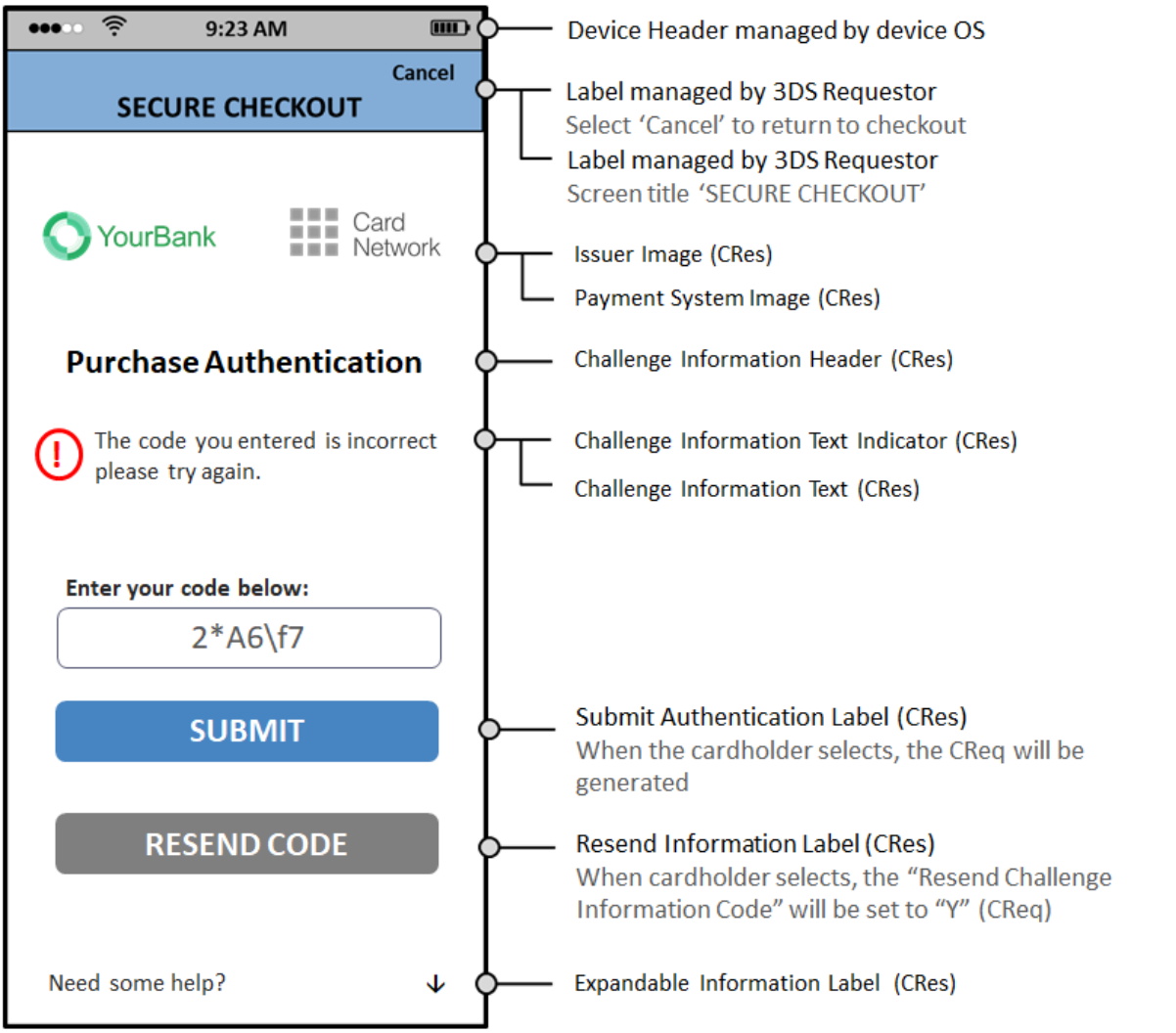
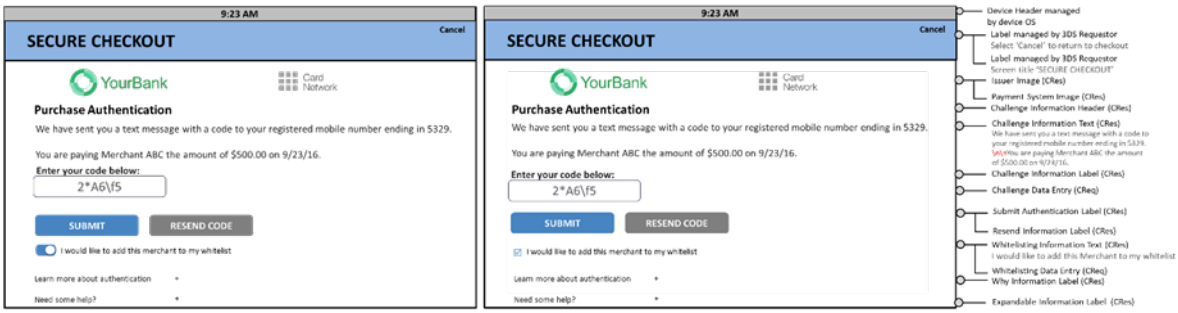


Fig 4.22 and 4.23 provides a sample format that that displays the Whitelisting option to the Cardholder during a purchase authentication.

Figure 4.23 Sample Whitelisting Information Text—PA—Landscape



4.2.4.3 3DS SDK

The 3DS SDK shall:

[Req 154]

~~Control the label for the~~Act upon any action ~~(for example, the Cancel action)~~ to exit the 3DS SDK (for example, the Cancel action on-screen or through an external controller) and return to the 3DS Requestor App.

[Req 157]

Return control to the 3DS Requestor App when the Cancel action ~~in the 3DS Requestor header~~ is ~~selected~~activated.

4.2.5.1 3DS SDK/ACS

The 3DS SDK shall for the CReq/CRes message exchange:

[Req 373]

~~Display~~ Provide the Cancel action. This can be implemented as a button in the top corner of the header zone as depicted in Figure 4.1 and Figure 4.2 and/or as a function on a controller for the platform.

[Req 374]

Create HTML with form elements in the applicable zones as outlined in Figure 4.1 and Figure 4.2 to support both portrait and landscape UI templates. The expected format is outlined in Section 4.2.6.

4.2.6 HTML UI Templates

The HTML UI templates provide the ACS the ability to include Issuer-specific design elements (e.g. branding, colours, fonts) as shown in the figures below. Figure 4.24~~Figure 4.15~~ and Figure 4.25 provides sample Payment Authentication HTML OTP UI templates that includes Issuer branding.

4.25 Sample HTML UI/OTP/Text Template—PA—Landscape

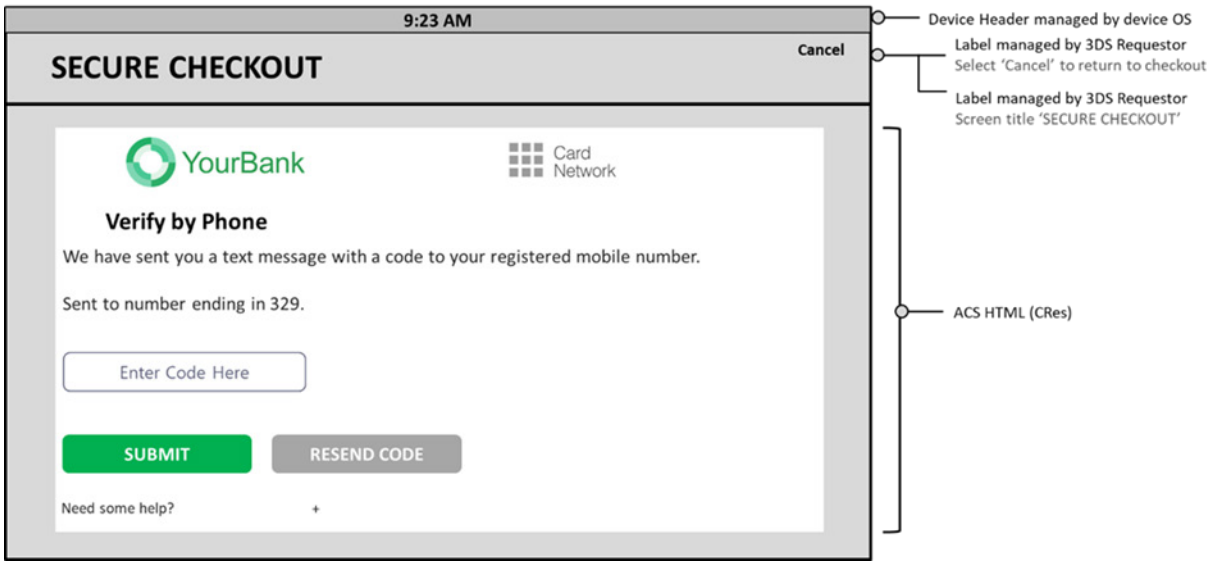
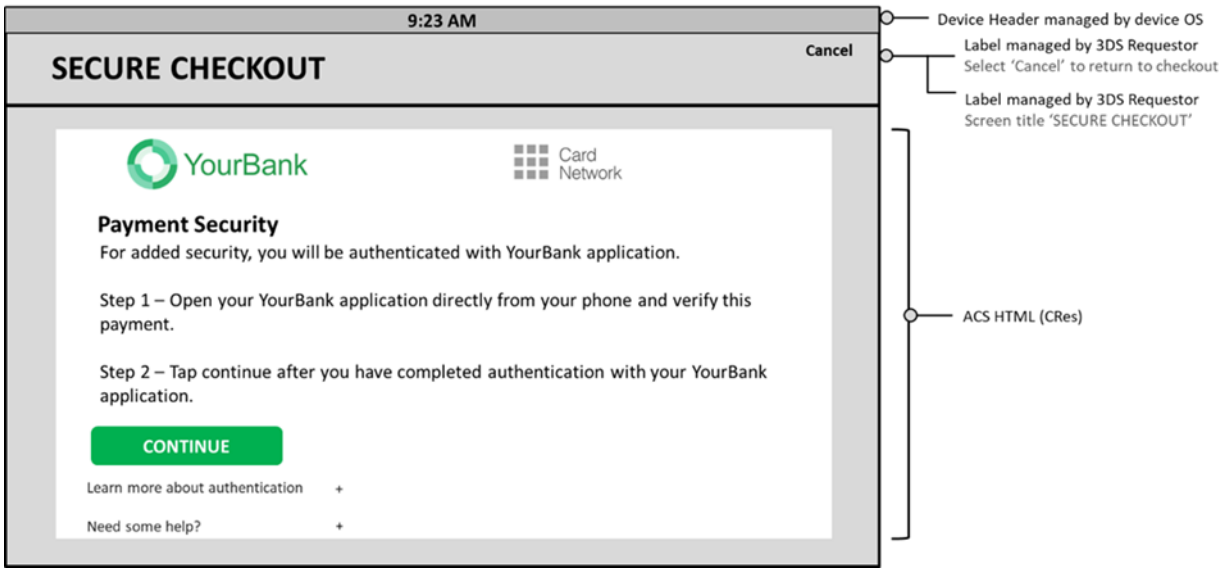


Figure 4.27~~Figure 4.17~~ and Figure 4.28 provide sample templates illustrating the OOB HTML UI.

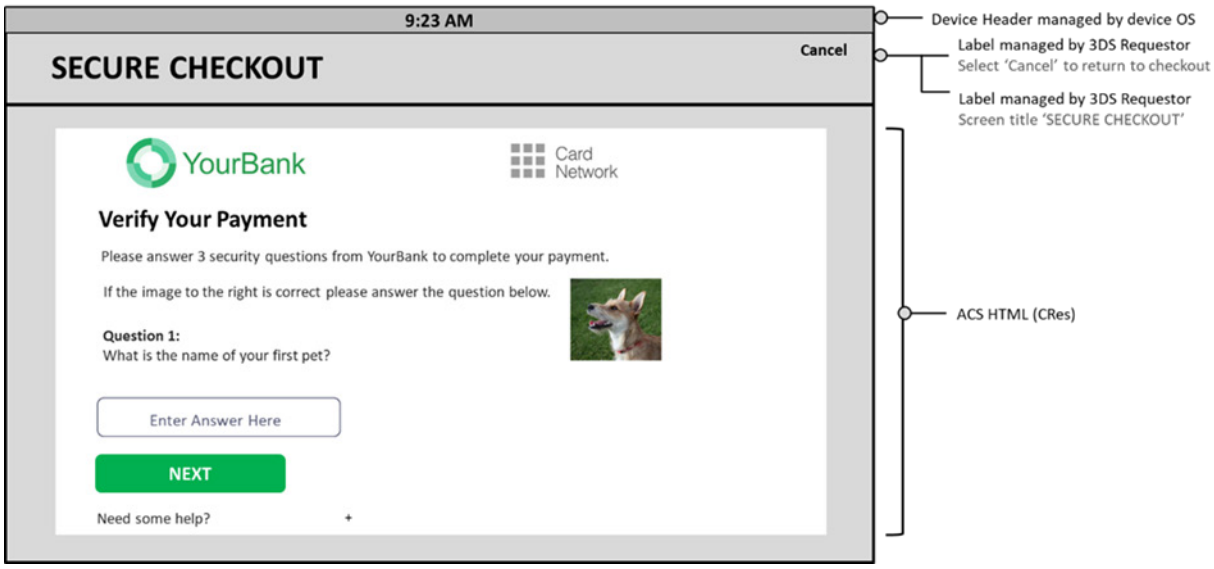
Figure 4.28: Sample OOB HTML UI Template—PA—Landscape



4.2.6.1 HTML Other UI Template

Figure 4.29~~Figure 4.18~~ and Figure 4.30 provide sample HTML Other templates asking the Cardholder to answer questions and confirm an image. There is not an existing data element in the Native format that supports the presentation of an image during authentication, however, the HTML Other will allow for this authentication experience.

Figure 4.30: Sample HTML Other UI Template—PA—Landscape



4.2.7.3 3DS SDK

The 3DS SDK shall:

[Req 171]

Return control to the 3DS Requestor App when the Cancel action ~~in the 3DS Requestor header~~ is selected.

4.3.2.1 ACS

The ACS shall for the CReq/CRes exchange:

[Req 380]

Create HTML with form elements in the applicable zones as outlined in Figure 4.1 and Figure 4.2 to support both portrait and landscape UI templates. The format is outlined in the UI templates in Section 4.3.3.

Chapter 5 EMV 3-D Secure Message Handling Requirements

Section 5.1.5 is a new section. Subsequent headings were renumbered as applicable.

5.1.5 Data Version Numbers

[Req 396]

The 3DS SDK shall implement the latest Data Version of the 3DS SDK Device Information.

[Req 397]

The ACS shall implement all active Data Versions of the 3DS SDK Device Information.

Note: Refer to EMV® 3-D Secure SDK—Device Information.

5.8.1 3DS Method Handling

[Req 263]

Recall the 3DS Server Transaction ID received in the initial 3DS Method POST, then Base64url encode the JSON object and send via a form with a field named `threeDSMethodData` in the Cardholder Browser HTML iframe using an HTTP POST method to the 3DS Method Notification URL. Refer to Table A.2 for detailed information about 3DS Method Data.

Chapter 6 EMV 3-D Secure Security Requirements

6.1.4.1 For App-based CReq/CRes

New paragraph at the end of Section 6.1.4.1.

If the CRes message contains a URL(s) directing the 3DS SDK to fetch data from an external server, an additional link is established using a TLS protocol, with server authentication by the 3DS SDK based on a commercial server certificate.

Annex A 3-D Secure Data Elements

Throughout Annex A, all instances of http have been replaced with https for all domain examples.

A.4 EMV 3-D Secure Data Elements

Table A.1 EMV 3-D Secure Data Elements

Data Element/ Field Name	Description	Source	Length/Format/ Values	Device Channel	Message Category	Message Inclusion	Condition Inclusion
3DS Requestor App URL	~~Merchant app~~ 3DS Requestor App declaring their URL within the CReq message so that the Authentication app can call the ~~merchant app~~3DS Requestor App after OOB authentication has occurred. Each transaction would require a unique Transaction ID by using the SDK Transaction ID.		Example value: https://appname.com?transID=b2385523-a66c-4907-ac3c-91848e8c0067 ~~merchantScheme://appName?transID=b2385523-a66c-4907-ac3c-91848e8c0067~~				~~R~~equired if 3DS Requestor App URL is ~~supported~~ provided by the 3DS Requestor App.

Data Element/ Field Name	Description	Source	Length/Format/ Values	Device Channel	Message Category	Message Inclusion	Condition Inclusion
OOB Continuation Label							~~Note: If present, either of the following must also be present:~~ • ~~Challenge Information Header, OR~~ • ~~Challenge Information Text~~ Refer to Table A.18 for additional information.
Submit Authentication Label							~~Note: If present, either of the following must also be present:~~ • ~~Challenge Information Header, OR~~ • ~~Challenge Information Label, OR~~ • ~~Challenge Information Text~~ Refer to Table A.18 for additional information.

A.5.4 Browser CReq and CRes POST

The following table defines the data elements sent in the Browser POST to the ACS for the CReq flow, and to the Notification URL in the CRes flow. An HTML form is utilised within the Cardholder Browser and the data is ~~sent~~ redirected via the Cardholder Browser in an HTTP POST.

Note: The end result of the redirection must be similar as if an HTML tag was utilised.

A.8 UI Data Elements

Table A.18 specifies the placement and the presence of UI data elements on the UI with respect to the zones defined in Section 4.1.

- M = Mandatory presence
- O = Optional presence
- N = Not present

Table A.18: UI Data Elements

Data Element	Field Name	Zone	Portrait Top-down Display Order	Landscape Top-down Display Order	ACS UI Type			
					OTP	Single Select	Multi Select	OOB
Challenge Information Header	challengeInfoHeader	3	2	2	M	M	M	M
Challenge Information Label	challengeInfoLabel	3	4	4	M	M	M	O
Challenge Information Text	challengeInfoText	3	3	3	M	M	M	M
Challenge Information Text Indicator	challengeInfoTextIndicator	3	3	3	O	O	O	O
Challenge Selection Information	challengeSelectInfo	3	5	5	N	M	M	N
Expandable Information Label	expandInfoLabel	4	12	10	O	O	O	O
Expandable Information Text	expandInfoText	4	13	11	O	O	O	O

Data Element	Field Name	Zone	Portrait Top-down Display Order	Landscape Top-down Display Order	ACS UI Type			
					OTP	Single Select	Multi Select	OOB
Issuer Image	issuerImage	2	1	1	O	O	O	O
~~OOB App Label~~	~~oobAppLabel~~							
OOB Continuation Label	oobContinueLabel	3	6	6	N	N	N	M
Payment System Image	psImage	2	1	1	O	O	O	O
Resend Information Label	resendInformationLabel	3	8	6	O	N	N	N
Submit Authentication Label	submitAuthenticationLabel	3	7	6	M	M	M	N
Whitelisting Information Text	whitelistingInfoText	3	9	7	O	O	O	O
Why Information Label	whyInfoLabel	4	10	8	O	O	O	O
Why Information Text	whyInfoText	4	11	9	O	O	O	O

Chapter 1 Introduction

1.5 Definitions

Table 1.3 Definitions

Term	Definition
Directory Server ID (`directoryServerID`)	Registered Application Provider Identifier (RID) that is unique to the Payment System. RIDs are defined by the ISO 7816-5 standard. The Directory Server ID is a hex value encoded as a 10-character text. For example, 0x'A000000003' is encoded as 'A000000003'.

Chapter 3 EMV 3-D Secure Authentication Flow Requirements

3.1 App-based Requirements

The 3DS Server shall:

[Req 355]

If the Cardholder Information Text has been provided by the ACS for this transaction the 3DS Server shall ensure the Cardholder Information Text is displayed on the 3DS Requestor ~~website~~App.

The ACS shall for all Challenge flow transactions (ARes Transaction Status = C) and for Decoupled Authentication transactions (ARes Transaction Status = D) once the authentication as defined in [Req 322].b has completed or the timer as defined in [Req 322].a has expired do the following:

[Req 345]

Ensure for a Decoupled Authentication transaction that:

- An RReq message is sent immediately upon obtaining an authentication result (whether successful or not).
- An RReq message without an authentication result (Transaction Status = U) is sent when the 3DS Requestor Decoupled Max Time expires—with a grace period of 1 hour.

  Note: It is recommended that an RReq message with Transaction Status = U contains Transaction Status Reason = 24 or 26 and Challenge Cancelation Indicator = 03.

The 3DS Server shall:

[Req 346]

For a Decoupled Authentication transaction, at a minimum wait the specified 3DS Requestor Decoupled Max Time plus 1 hour, 30 seconds for the RReq message. If an RReq message is never received, further processing is outside the scope of 3-D Secure processing.

3.3 Browser-based Requirements

The ACS shall for all Challenge Flow transactions (ARes Transaction Status = C) and for a Decoupled Authentication transaction (ARes Transaction Status = D) once the authentication as defined in [Req 326].b has completed or the timer as defined in [Req 326].a has expired, do the following:

[Req 347]

Ensure for a Decoupled Authentication transaction that:

- An RReq message is sent immediately upon obtaining an authentication result (whether successful or not).
- An RReq message without an authentication result (Transaction Status = U) is sent when the 3DS Requestor Decoupled Max Time expires—with a grace period of 1 hour.

  Note: It is recommended that an RReq message with Transaction Status = U contains Transaction Status Reason = 24 or 26 and Challenge Cancelation Indicator = 03.

The 3DS Server shall:

[Req 348]

For a Decoupled Authentication transaction, at a minimum wait the specified 3DS Requestor Decoupled Max Time plus 1 hour, 30 seconds for the RReq, If an RReq message is never received, further processing is outside the scope of 3-D Secure processing.

3.4 3RI-based Requirements

The ACS shall for a Decoupled Authentication transaction (initial Transaction Status = D) once the authentication as defined in [Req 330].b has completed, or the timer as defined in [Req 330].a has expired, do the following:

[Req 353]

Ensure that:

- An RReq message is sent immediately upon obtaining an authentication result (whether successful or not).
- An RReq message without an authentication result (Transaction Status = U) is sent when the 3DS Requestor Decoupled Max Time expires—with a grace period of 1 hour.

  Note: It is recommended that an RReq message with Transaction Status = U contains Transaction Status Reason = 24 or 26 and Challenge Cancelation Indicator = 03.

The 3DS Server shall:

[Req 354]

For a Decoupled Authentication transaction, at a minimum wait the specified 3DS Requestor Decoupled Max Time plus 1 hour, 30 seconds for the RReq, If an RReq message is never received, further processing is outside the scope of 3-D Secure processing.

Chapter 4 EMV 3-D Secure User Interface Templates, Requirements, and Guidelines

4.1 3-D Secure User Interface Templates

The ACS shall:

[Req 342]

Support all ACS Rendering Types for the ACS supported authentication methods, at a minimum at least one ACS UI Template for each ACS Interface~~Native Device Rendering Option and HTML~~.

4.2 App-based User Interface Overview

The supported digital image file types are png, jpeg, tiff and bmp. Any other image types implemented by the ACS may not be supported by the 3DS SDK.

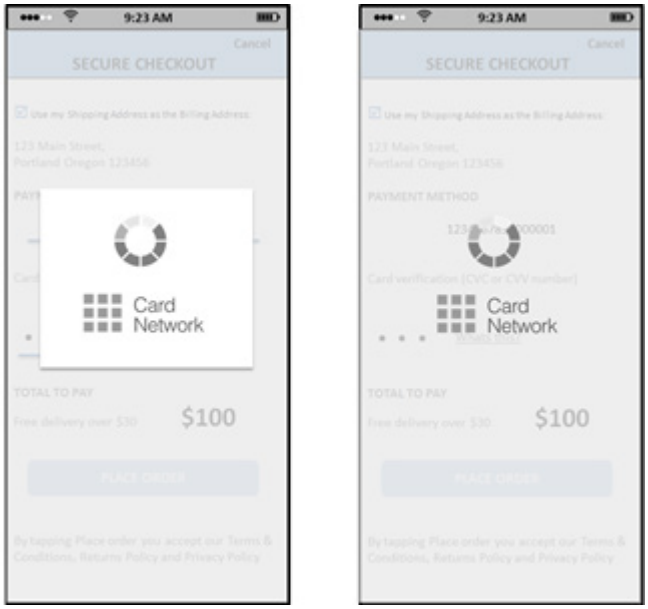
Note: Some platforms may not natively support all image types.

4.2.1 Processing Screen Requirements

New graphic for Figure 4.4

***(Original)* Figure 4.4 Sample App-based Processing Screen**

(Updated) **Figure 4.4 Sample App-based Processing Screen**



4.2.1.1 3DS SDK/3DS Requestor App

The 3DS SDK shall for the AReq/ARes message exchange:

[Req 143]
Integrate the Processing Graphic and if requested, the DS logo into the centre of the Processing screen as depicted in Figure 4.4 with or without a white box.

The 3DS Requestor App shall for the AReq/ARes message exchange:

[Req 145]
Display the Processing screen supplied by the 3DS SDK during the entire AReq/ARes message cycle and overlay on the merchant checkout page as depicted in Figure 4.4.

The 3DS Requestor App shall in case of challenge:

[Req 388]

Set the Header zone text and the Cancel action name to be displayed by the SDK.

~~**[Req 360]**~~
~~Display the Cancel action in the top right corner of the Header zone.~~

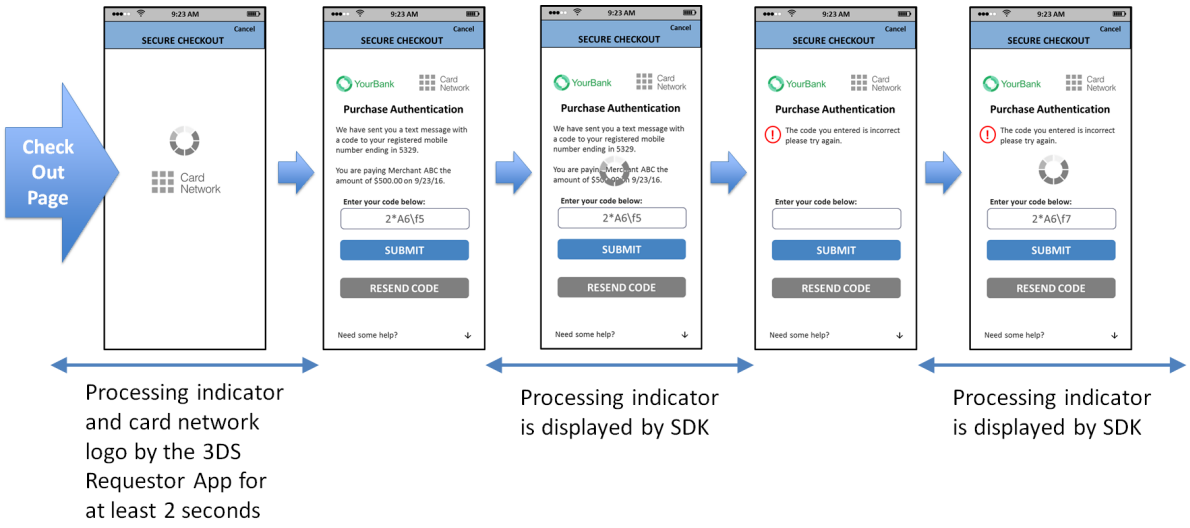
The 3DS SDK shall for the CReq/CRes message exchange:

~~**[Req 361]**~~
~~Display the Cancel action in the top right corner of the Header zone.~~

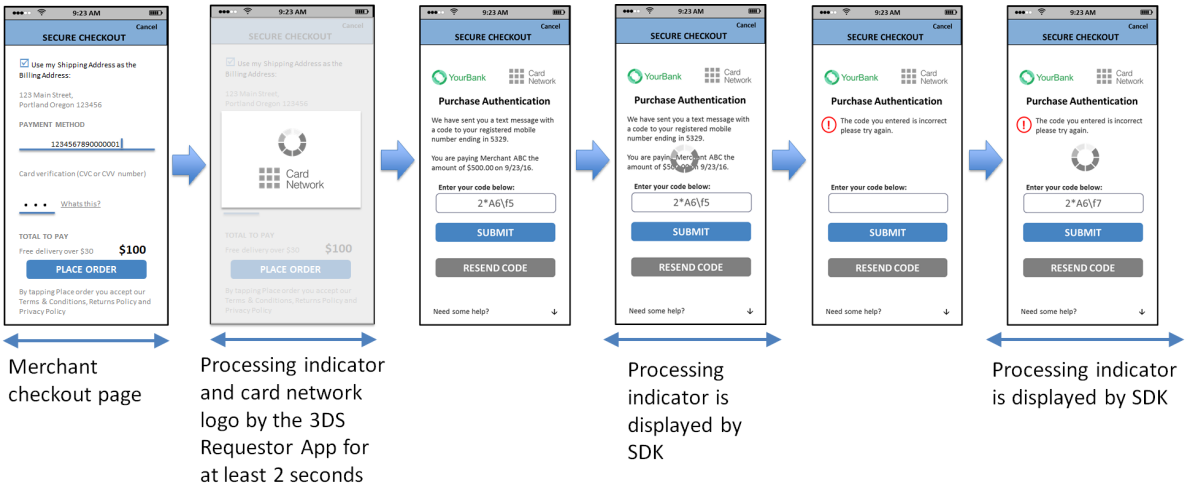
[Req 389]
Ensure that the Cancel action is not actionable on the Processing screen.

New Graphic for Figure 4.5

***(Original)* Figure 4.5: Sample OTP/Text Template—App-based Processing Flow**

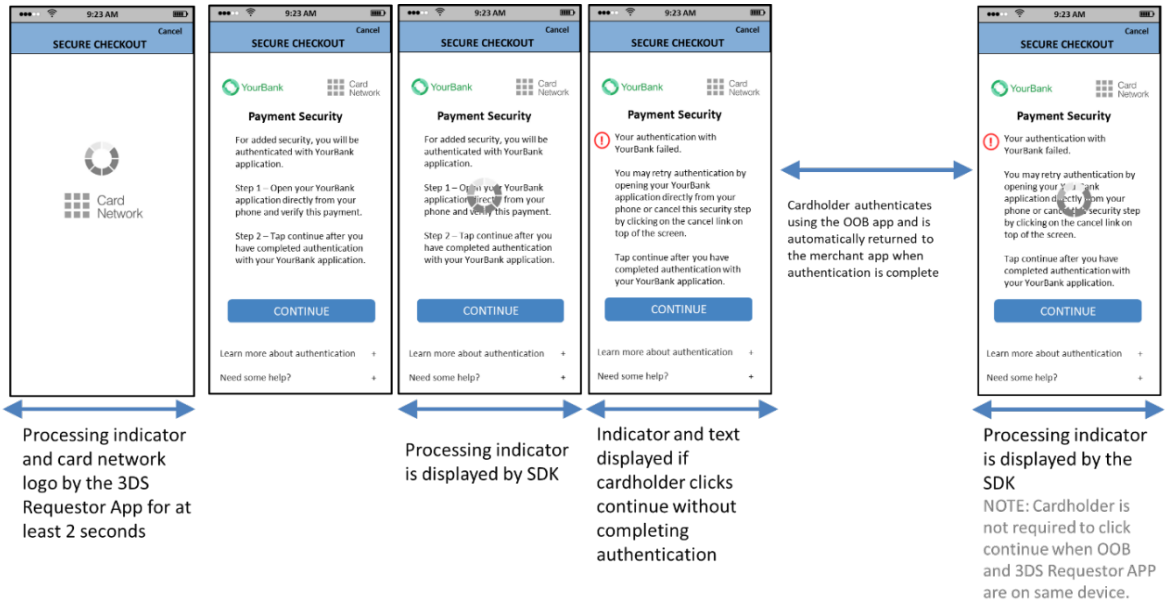


***(Updated)* Figure 4.5: Sample OTP/Text Template—App-based Processing Flow**

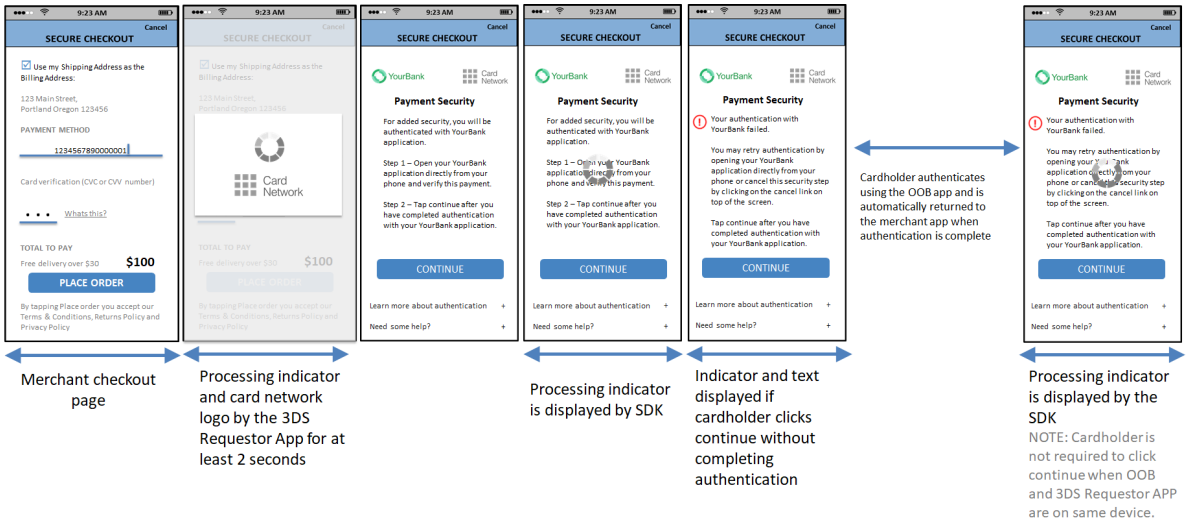


New Graphic for Figure 4.6

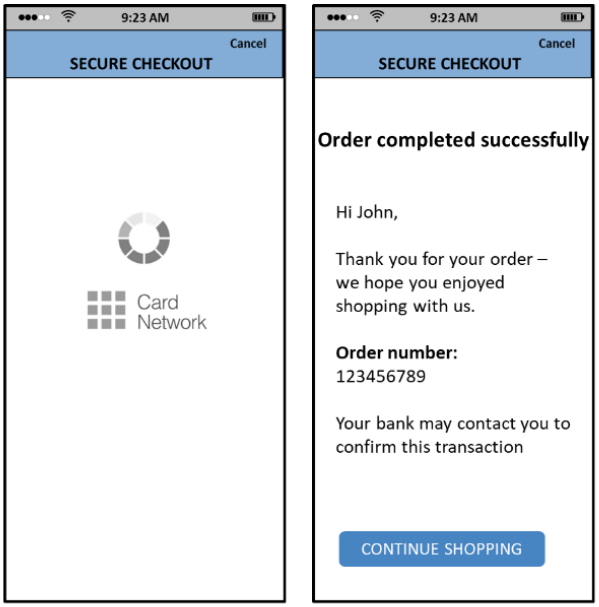
***(Original)* Figure 4.6: Sample OOB Template (OOB App and 3DS Requestor App on same device)—App-based Processing Flow**



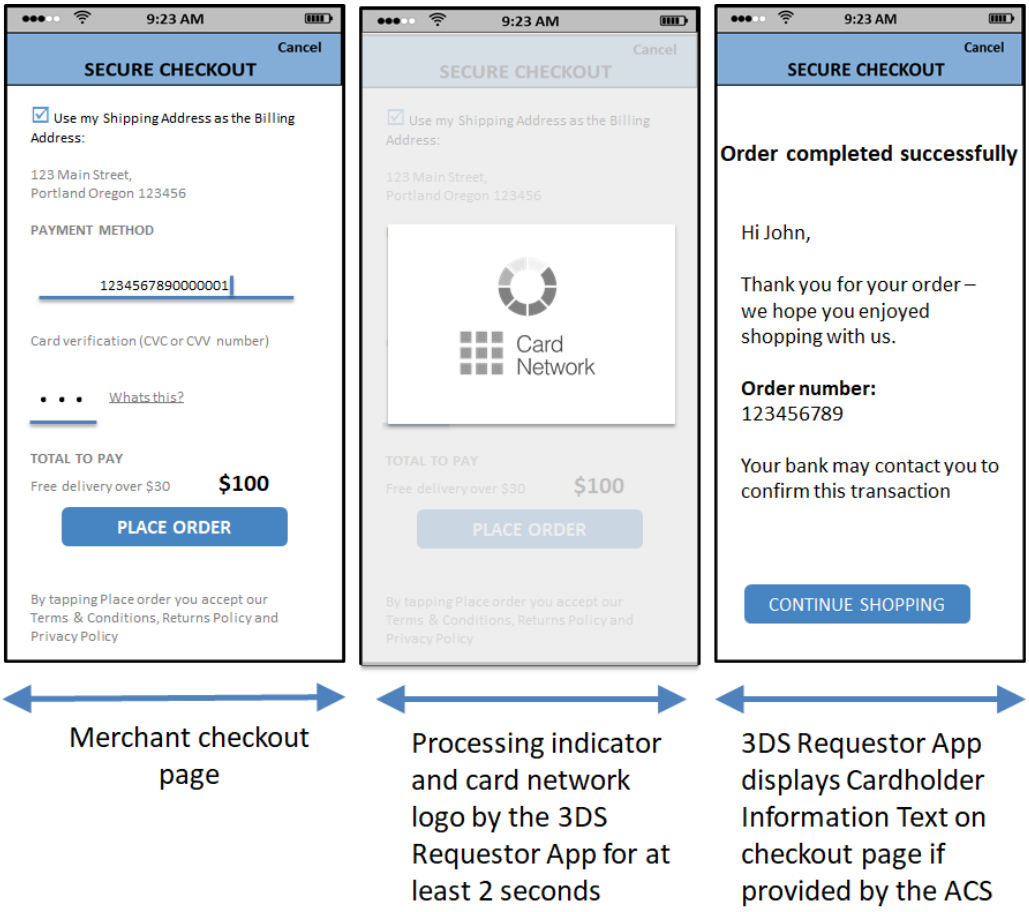
***(Updated)* Figure 4.6: Sample OOB Template (OOB App and 3DS Requestor App on same device)—App-based Processing Flow**



(Original) **Figure 4.7: Sample Decoupled Authentication Template—App-based Processing Flow**



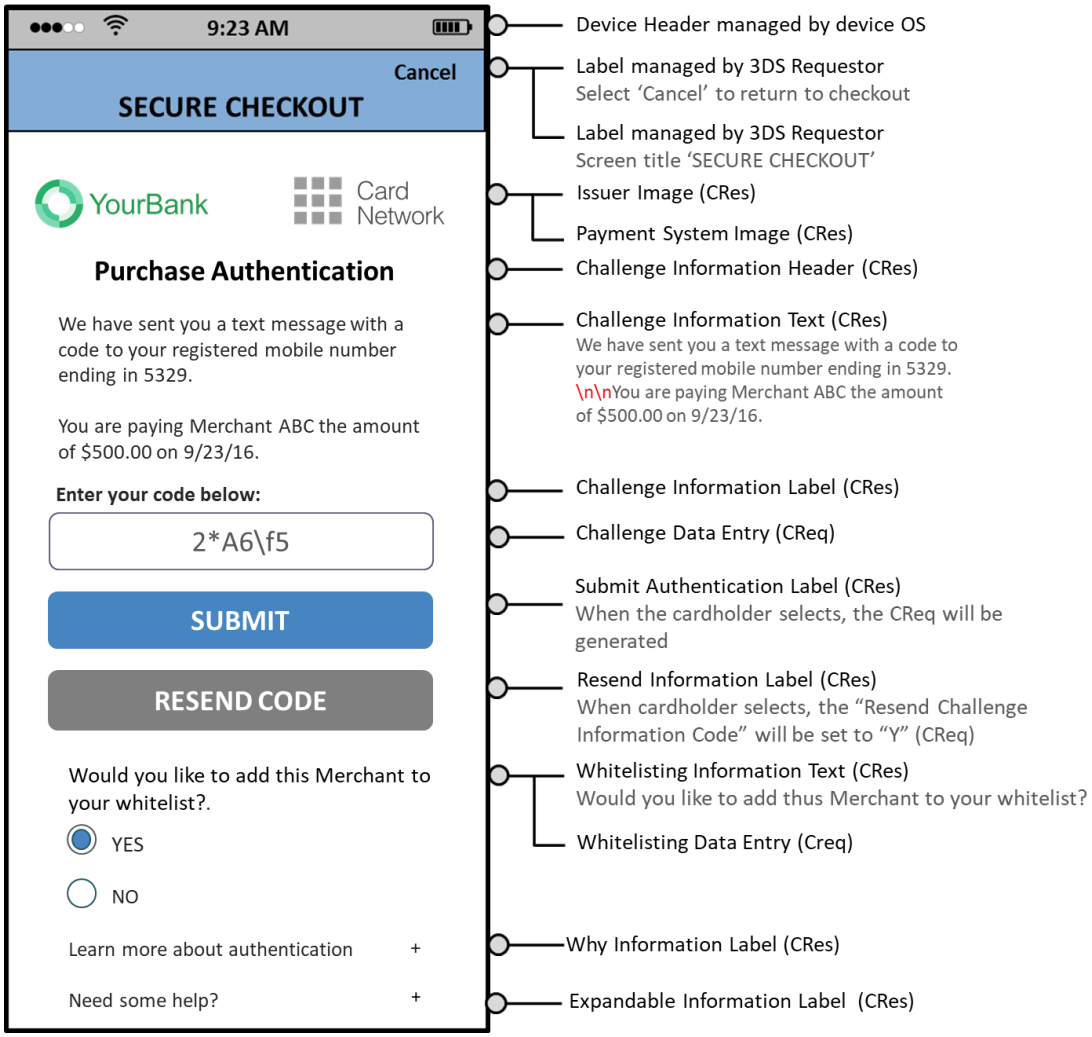
(Updated) Figure 4.7: Sample Decoupled Authentication Template—App-based Processing Flow



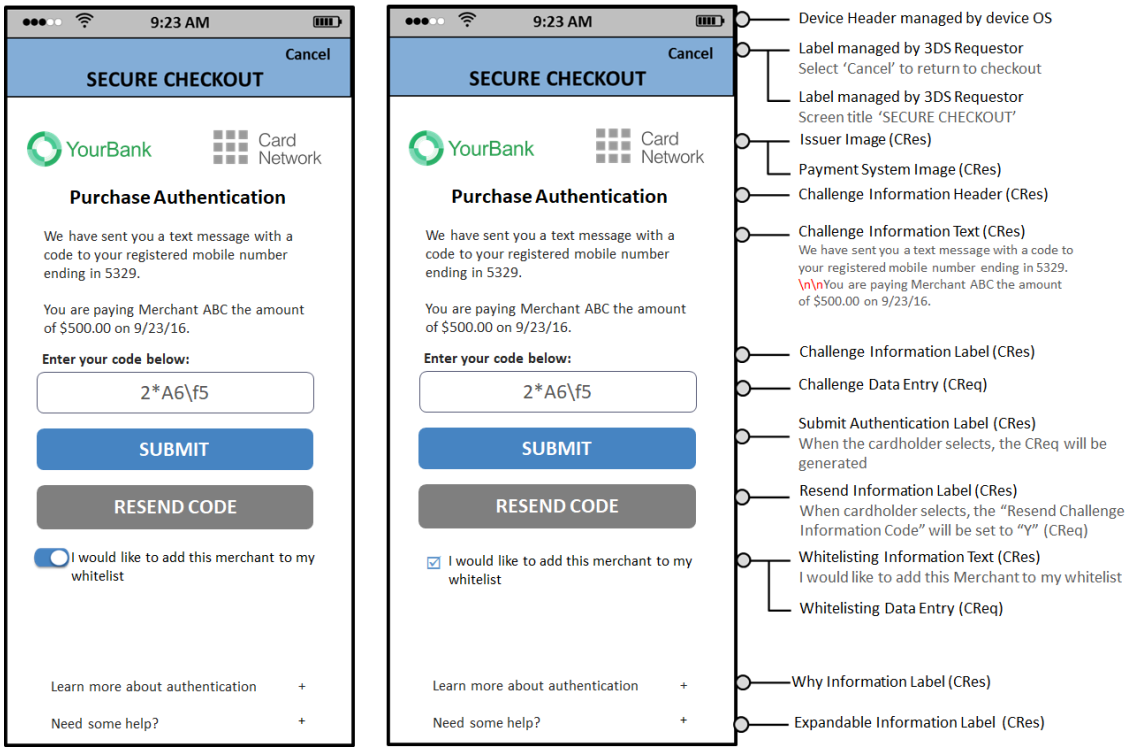
4.2.3 Native UI Templates

New graphic for Figure 4.14

***(Original)* Figure 4.14 Sample Whitelisting Information Text—PA**



***(Updated)* Figure 4.14 Sample Whitelisting Information Text—PA**



4.2.4.1 3DS SDK

The 3DS SDK shall:

[Req 153]
After submitting the CReq message to the ACS, display the same Processing screen as during the AReq/ARes message until the CRes message is received, or timeout is exceeded.

4.2.7.3 3DS SDK

The 3DS SDK shall:

[Req 171]
Return control to the 3DS Requestor App when the Cancel action in the 3DS Requestor header is selected.

On HTML submit:

- The web view will return~~, either~~ a parameter string (HTML Action = GET) ~~or form data (HTML Action = POST)~~ containing the cardholder's data input.
- The SDK passes the received data, unchanged, to the ACS in the Challenge HTML Data Entry data element of the CReq message. The SDK shall not modify or reformat the data.

Chapter 5 EMV 3-D Secure Message Handling

5.1.3 Base64/Base64url Encoding

[Req 193]

Base64 ~~and Base64url~~ decoding software shall ignore any white space (such as carriage returns or line ends) within Base64 ~~and Base64url~~ encoded data and shall not treat the presence of such characters as an error.

5.1.6 Message Content Validation

[Req 309]

Unless explicitly noted, if a conditionally optional or optional field is sent as empty or null, the receiving component shall return an Error Message (as defined in Section A.5.5) with the applicable Error Component and Error Code = 203.

Example:

The DS receives an ARes message from the ACS with an empty conditionally Optional data element that is specified in Table A.1 for the Message Type, Device Channel and Message Category but the condition is not met. Such as, acsChallengeMandated = "" and transStatus = Y. The DS validates the ARes message content and returns an error to the ACS and can return an ARes message or Error to the 3DS Server.

Chapter 6 EMV 3-D Secure Security Requirements

Multiple updates are made to Section 6.2 Security Functions. These edits are included in the following section and additionally for clarity, are included at the end of this section in a "clean" final format with no revision marks. Click here to view the "clean" version of these edits.

6.2.2.1 3DS SDK Encryption

The 3DS SDK:

- If $P_{DS,}$ is an RSA key:
  - Encrypts the JSON object according to JWE (RFC 7516) using JWE Compact Serialization. The parameter values ~~supported in~~for this version of the specification and to be included in the JWE protected header are:
- Else if P_{DS} is an EC key:
  - Conducts a Diffie-Hellman key exchange process according to JWA (RFC 7518) in Direct Key Agreement mode using ECDH-ES, curve P-256, d_{SDK}, ~~and~~ P_{DS} with Concat KDF to produce a 256-bit CEK. The Concat KDF parameter values ~~supported in~~for this version of the specification are:

    ~~"alg":ECDH-ES~~

    ~~"apv":DirectoryServerID~~

    ~~"epk": P_{DS}, in JSON Web Key (JWK) format~~
    ~~{"kty":"EC"~~
    ~~"crv":"P-256"}~~

    ~~All other parameters: not present~~

    - Keydatalen = 256
    - AlgorithmID = empty string (length = 0x00000000)
    - PartyUInfo = empty string (length = 0x00000000)
    - PartyVInfo = `directoryServerID` (length || ascii string)
    - SuppPubInfo = Keydatalen (0x00000100)
    - SuppPrivInfo = empty octet sequence

  ~~- CEK: "kty":oct - 256 bits~~

  - Generates 128-bit random data as IV (included in the JWE)
  - Encrypt the JSON object according to JWE (RFC 7516) using the CEK and JWE Compact Serialization. The parameter values ~~supported~~for this version of the specification and to be included in the JWE protected header are:

    - "alg":~~dir~~"ECDH-ES"
    - "epk": Q_{SDK},
      {"kty": "EC",
      "crv": "P-256"
      "x":x coordinate of Q_{SDK}
      "y":y coordinate of Q_{SDK}}

6.2.2.2 DS Decryption

The DS:

- If the protected header of the JWE in the SDK Encrypted Data field indicates that ~~a RSA key~~RSA-OAEP-256 was used for encryption:
  - Decrypts the SDK Encrypted Data field from the AReq message according to JWE (RFC 7516) using RSA-OAEP-256 and either A128CBC-HS256 or A128GCM as indicated by the "enc" parameter in the protected header. ~~The parameter values supported in this version of the specification are:~~
  - ~~"alg": RSA-OAEP-256~~
  - ~~If the enc parameter of the JWE in the SDK Encrypted Data field indicates that A128CBC-HS256 was used for encryption:~~
  - ~~"enc": A128CBC-HS256~~
  - ~~If the enc parameter of the JWE in the SDK Encrypted Data field indicates that A128GCM was used for encryption:~~
  - ~~"enc": A128GCM~~
  - ~~All other parameters: not present~~
- Else, if the protected header of the JWE in the SDK Encrypted Data field indicates that ~~an EC key~~ECDH-ES was used for encryption:
  - Conducts a Diffie-Hellman key exchange process according to JWA (RFC 7518) in Direct Key Agreement mode using ECDH-ES, curve P-256, Q_{SDK}, and d_{DS} with the parameter values from the protected header and Concat KDF to produce a 256-bit CEK. The Concat KDF parameter values ~~supported in~~for this version of the specification are:
    - ~~"alg":ECDH-ES~~
    - ~~"apv": DirectoryserverID~~
    - ~~"epk": Q_{SDK}~~
    - ~~{"kty":"EC"~~
    - ~~"crv":"P-256"}~~
    - ~~If the enc parameter of the JWE in the SDK Encrypted Data field indicates that A128CBC-HS256 was used for encryption :~~
    - ~~"enc": A128CBC-HS256~~
    - ~~If the enc parameter of the JWE in the SDK Encrypted Data field indicates that A128GCM was used for encryption:~~
    - ~~"enc": A128GCM~~
    - ~~All other parameters: not present~~
    - Keydatalen = 256
    - AlgorithmID = empty string (length = 0x00000000)
    - PartyUInfo = empty string (length = 0x00000000)
    - PartyVInfo = `directoryServerID` (length || ascii string)
    - SuppPubInfo = Keydatalen (0x00000100)

- SuppPrivInfo = empty octet sequence

~~o CEK: "kty":oct - 256 bits~~

o Decrypt the JWE in the SDK Encrypted Data field according to JWE (RFC 7516) using the CEK and either A128CBC-HS256 or A128GCM as indicated by the “enc” parameter in the protected header. If the algorithm is A128GCM the leftmost 128bits of CEK is used ~~with the received IV~~. If decryption fails, ceases processing and reports error.

6.2.3.2 ACS Secure Channel Setup

The ACS:

- Completes the Diffie-Hellman key exchange process as a local mechanism according to JWA (RFC 7518) in Direct Key Agreement mode using ECDH-ES, curve P-256, d_T and Q_C with Concat KDF to produce a ~~pair of~~ 256-bit CEK~~s (one for each direction)~~ which ~~are~~is identified ~~by~~to the ACS Transaction ID. ~~In order to obtain 256 bits of keying material from the included Concat KDF function assume an "enc" parameter of ECDH-ES+A256KW and assume the algorithmID to be null for the KDF[8].~~ (*Footnote 8 also deleted*: [8]~~Note this is using RFC 7518 only for key derivation.~~). The Concat KDF parameter values ~~supported in~~for this version of the specification are:

  ~~o "alg":ECDH-ES~~

  ~~o "apv": SDK Reference Number~~

  ~~o "epk": Q_C (received in the AReq message as `sdkEphemKey`)~~

  ~~o {"kty":"EC"~~
  ~~"crv":"P-256"}~~

  ~~o All other parameters: not present~~

  o Keydatalen = 256

  o AlgorithmID = empty string (length = 0x00000000)

  o PartyUInfo = empty string (length = 0x00000000)

  o PartyVInfo = `sdkReferenceNumber` (length || ascii string)

  o SuppPubInfo = Keydatalen (0x00000100)

  o SuppPrivInfo = empty octet sequence

  ~~o CEK: "kty":oct -~~ 256 bits ~~extracted~~ allocated as:

- Generates a digital signature of the full JSON object according to JWS (RFC 7515) using JWS Compact Serialization. The parameter values ~~supported in~~for this version of the specification and to be included in the JWS header are:

6.2.3.3 3DS SDK Secure Channel Setup

The 3DS SDK:

- Using the CA public key of the DS CA identified from information provided by the 3DS Server, ~~Validate~~validates the JWS from the ACS according to JWS (RFC7515) using either PS256 or ES256 as indicated by the "alg" parameter in the header. ~~The 3DS SDK is required to support both "alg" parameters PS256 and ES256.~~ If validation fails, ceases processing and report error.

- Completes the Diffie-Hellman key exchange process according to JWA (RFC 7518) in Direct Key Agreement mode using curve P-256, d_C and Q_T, with Concat KDF to produce a ~~pair of~~ 256-bit CEK~~s (one for each direction)~~, which ~~are~~is identified to the ACS Transaction ID received in the ARes message. ~~In order to obtain 256 bits of keying material from the included Concat KDF function assume an "enc" parameter of ECDH-ES+A256KW and assume the algorithmID to be null for the KDF[10]. (*Footnote 10 also deleted:* [10] Note this is using RFC 7518 only for key derivation).~~ The Concat KDF parameter values supported ~~in~~for this version of the specification are:
  - ~~"alg":ECDH-ES~~
  - ~~"apv": SDK Reference Number~~
  - ~~"epk": Q_T (received in the AReq message as `acsEphemPubKey`) which is part of ACS Signed Content)~~
  - ~~{"kty":"EC"~~
    ~~"crv":"P-256"}~~
  - ~~All other parameters: not present~~
  - Keydatalen = 256
  - AlgorithmID = empty string (length = 0x00000000)
  - PartyUInfo = empty string (length = 0x00000000)
  - PartyVInfo = `sdkReferenceNumber` (length || ascii string)
  - SuppPubInfo = Keydatalen (0x00000100)
  - SuppPrivInfo = empty octet sequence
  - CEK: ~~"kty":oct –~~256 bits ~~extracted~~ allocated as:

If the ACS signature is valid, the 3DS SDK has confirmed the authenticity of the ACS, that the session keys are fresh, and that the ACS_URL is correct.

6.2.4.1 3DS SDK—CReq

For CReq messages sent from the 3DS SDK to the ACS, the 3DS SDK:

- Encrypts the JSON object according to JWE (RFC 7516) using the CEK_{S-A} obtained in Section 6.2.3.3 and JWE Compact Serialization. The parameter values ~~supported in~~for this version of the specification and to be included in the JWE protected header are:
- Sends the resulting JWE to the ACS as the ~~encrypted~~ protected CReq message.

6.2.4.2 3DS SDK—CRes

For CRes messages received by the 3DS SDK from the ACS, the 3DS SDK:

- Decrypts the message according to JWE (RFC 7516) using either A128CBC-HS256 or A128GCM and the CEK_{A-S} obtained in Section 6.2.3.3 as identified by the "enc" and "kid" parameters in the protected header. If decryption fails, ceases processing and reports error.

6.2.4.3 ACS—CReq

For CReq messages received by the ACS from the 3DS SDK, the ACS:

- Decrypts the message according to JWE (RFC 7516) using either A128CBC-HS256 or A128GCM and the CEK_{S-A} obtained in Section 6.2.3.2 as identified by the "enc" and "kid" parameters in the protected header. If decryption fails, ceases processing and reports error.

6.2.4.4 ACS—CRes

For CRes messages sent from the ACS to the 3DS SDK the ACS:

- Encrypts the JSON object according to JWE (RFC 7516) using the same "enc" algorithm used by the 3DS SDK for the CReq message, the CEK_{A-S} obtained in Section 6.2.3.2 identified by "kid" and JWE Compact Serialization. The parameter values ~~supported in~~for this version of the specification and to be included in the JWE protected header are:
- Sends the resulting JWE to the 3DS SDK as the ~~encrypted~~ protected CRes message.

Annex A 3-D Secure Data Elements

A.4 EMV 3-D Secure Data Elements

Table A.1 EMV 3-D Secure Data Elements

Data Element/ Field Name	Description	Source	Length/Format/ Values	Device Channel	Message Category	Message Inclusion	Condition Inclusion
3DS Requestor App URL						CReq = ~~O~~C	Required if 3DS Requestor App URL is supported.
3DS Requestor Authentication Indicator			07 = Billing Agreement ~~07~~08–79 = Reserved for EMVCo future use (values invalid until defined by EMVCo)				
3DS Requestor Authentication Method Verification Indicator							Conditional based on DS rules. The DS populates the AReq with this data element prior to passing to the ACS.
3DS Requestor Decoupled Max Time						AReq = ~~O~~C	Required if Decoupled Request Indicator = Y.

Data Element/ Field Name	Description	Source	Length/Format/ Values	Device Channel	Message Category	Message Inclusion	Condition Inclusion
3RI Indicator			12 = Billing Agreement ~~12~~13–79 = Reserved for EMVCo future use (values invalid until defined by EMVCo)				
ACS Rendering Type	Identifies the ACS ~~UI~~ Interface and ACS UI Template that the ACS will first present to the consumer.						For RReq, required unless ACS Decoupled Confirmation Indicator = Y.
Authentication Method	Note: This is in the RReq message from the ACS only. It is not passed to the 3DS Server ~~URL~~.						This field is present in the RReq message from the ACS to the DS but is not present in the RReq message from the DS to the 3DS Server. ~~This field is not present in the RReq message from the DS to the 3DS Server URL.~~

Data Element/ Field Name	Description	Source	Length/Format/ Values	Device Channel	Message Category	Message Inclusion	Condition Inclusion
BrowserJavaScript Enabled Field Name: browserJavascriptEnabled *Note: The field name was incorrectly identified in SB 207. There was no change made in the specification.*							
Cardholder Email Address							Required (if available) unless market or regional mandate restricts sending this information

Data Element/ Field Name	Description	Source	Length/Format/ Values	Device Channel	Message Category	Message Inclusion	Condition Inclusion
Challenge Data Entry							Required when: • ACS UI Type = 01, 02, or 03, AND • Challenge data has been entered in the UI, AND • Challenge Cancelation Indicator is not present AND • Resend Challenge Information Code is not present ~~Are not present.~~

Data Element/ Field Name	Description	Source	Length/Format/ Values	Device Channel	Message Category	Message Inclusion	Condition Inclusion
Challenge No Entry							Required when: • ACS UI Type = 01, 02, or 03, AND • Challenge Data Entry is not present, AND • Challenge Cancelation Indicator is not present AND • Resend Challenge Information Code is not present ~~Are not present.~~
Device Rendering Options Supported	~~Defines~~ Identifies the SDK ~~UI types~~Interface and SDK UI Type that the device supports for displaying specific challenge user interfaces within the SDK.						

Data Element/ Field Name	Description	Source	Length/Format/ Values	Device Channel	Message Category	Message Inclusion	Condition Inclusion
DS Start Protocol Version	The ~~most recent~~ earliest (i.e. oldest) active protocol version that is supported for the DS.						
Instalment Payment Data							• Required for 03-3RI if 3RI Indicator = 02.
Interaction Counter	Indicates the number of authentication cycles (excluding Decoupled Authentication) attempted by the Cardholder.					RReq = ~~R~~C	Required unless ACS Decoupled Confirmation Indicator = Y.
Purchase Amount							• Required for 02-NPA if 3RI Indicator = 01, 02, 06, 07, 08, 09, 11.
Purchase Currency							• Required for 02-NPA if 3RI Indicator = 01, 02, 06, 07, 08, 09, 11.

Data Element/ Field Name	Description	Source	Length/Format/ Values	Device Channel	Message Category	Message Inclusion	Condition Inclusion
Purchase Currency Exponent							• Required for 02-NPA if 3RI Indicator = 01, 02, 06, 07, 08, 09, 11.
Purchase Date & Time							• Required for 02-NPA if 3RI Indicator = 01, 02, 06, 07, 08, 09, 11.
Recurring Expiry							• Required for 03-3RI if 3RI Indicator = 01 or 02.
Recurring Frequency							• Required for 03-3RI if 3RI Indicator = 01 or 02.

A.5.7 Card Range Data

Table A.6 Card Range Data

Data Element/Field Name	Description	Length/Format/Values	Inclusion
ACS End Protocol Version		Note: If the ACS End Protocol Version is not available, this value is the DS End Protocol Version for that card range.	

Data Element/Field Name	Description	Length/Format/Values	Inclusion
ACS Start Protocol Version		Note: If the ACS Start Protocol Version is not available, this value is the DS Start Protocol Version for that card range.	

A.7.3 3DS Requestor Information

The 3DS Requestor Authentication Information contains optional information about how the cardholder authenticated during login to their 3DS Requestor account. The detailed data elements, which are optional, are outlined in Table A.10.

A.7.4 3DS Requestor Prior Transaction Authentication Information

The 3DS Requestor Prior Transaction Authentication Information contains optional information about a 3DS cardholder authentication that occurred prior to the current transaction. The detailed data elements, which are optional, are outlined in Table A.11.

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