EMV® Specification Bulletin No. 204 v7
June 2023

EMV® 3-D Secure Protocol and Core Functions Specification version 2.1.0 – Updates, Clarifications & Errata

This Specification Bulletin (SB) No. 204v7 provides updates, clarifications and errata incorporated into the EMV 3-D Secure Protocol and Core Functions Specification since the October 2017 v2.1.0 publication. This SB 204v7 also includes v2.1.0 content consolidated from SB 224v1 for clarity.

Applicability

This Specification Bulletin applies to:

- *EMV® 3-D Secure Protocol and Core Functions Specifications, Version 2.1.0*

  Updates are provided by draft date, 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

The following publications should also be referenced for specification updates:

- *EMV® 3-D Secure JSON Message Samples*
- *EMV® 3-D Secure FAQs*

Effective Date

June 2023

Contents

June 2023 v7

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 v2.1.0 and 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).~~

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]

Utilise a server-authenticated TLS session as defined in Section 6.1.4.2.

[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.

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

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

4.1.3 3-D Secure Interface Templates

The 3DS SDK shall:

[Req 395]

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

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~~ provided mandatory and optional UI 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.

Requirement 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 on 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.

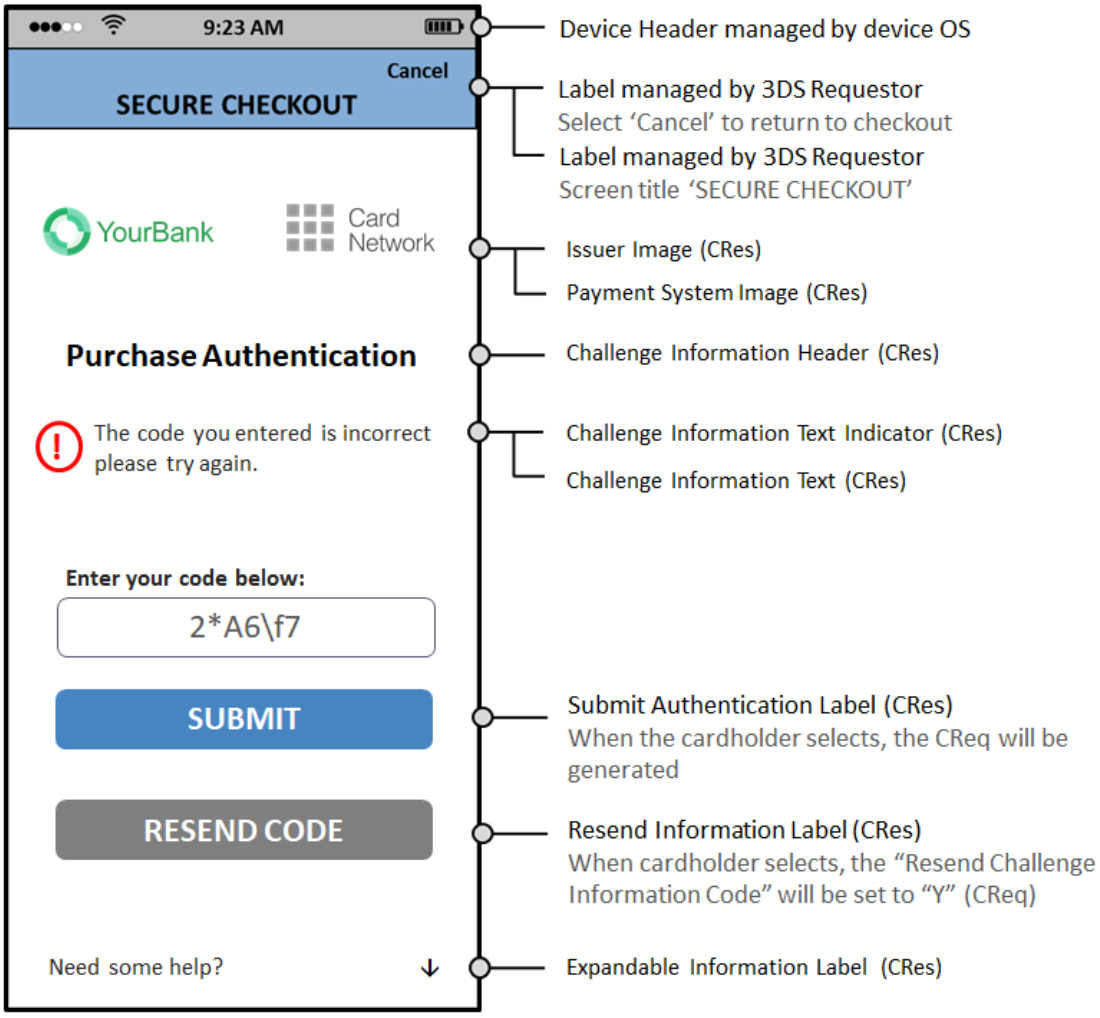
The ACS shall for the CReq/CRes message exchange:

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



4.2.5 HTML UI Display Requirements

[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.

4.3.2.1 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 added 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

Update notice: This 11 February 2020 version of SB 204v6 removes the 3DS Requestor App URL that was incorrectly included in the version published 7 February 2020.

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
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.

Data Element/ Field Name	Description	Source	Length/Format/ Values	Device Channel	Message Category	Message Inclusion	Conditional Inclusion
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 Additional Information Text	challengeAddInfo	3	3	3	N	N	N	O
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	12	O	O	O	O
Expandable Information Text	expandInfoText	4	13	13	O	O	O	O
Issuer Image	issuerImage	2	1	1	O	O	O	O
OOB Continuation Label	oobContinueLabel	3	6	6	N	N	N	M
Payment System Image	psImage	2	1	1	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
Resend Information Label	`resendInformationLabel`	3	8	6	O	N	N	N
Submit Authentication Label	`submitAuthenticationLabel`	3	7	6	M	M	M	N
Why Information Label	`whyInfoLabel`	4	10	10	O	O	O	O
Why Information Text	`whyInfoText`	4	11	11	O	O	O	O

[August 2019 SB224v1]

The 3-D Secure version 2.1.0 content in this August 2019 SB224v1 section was originally published in Specification Bulletin 224v1. SB 224v1 is now consolidated into this SB204v6 version for clarity. Therefore, this SB 204v6 now contains all updates to version 2.1.0 since the October 2017 publication of the EMV 3-D Secure Protocol and Core Functions specification.

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

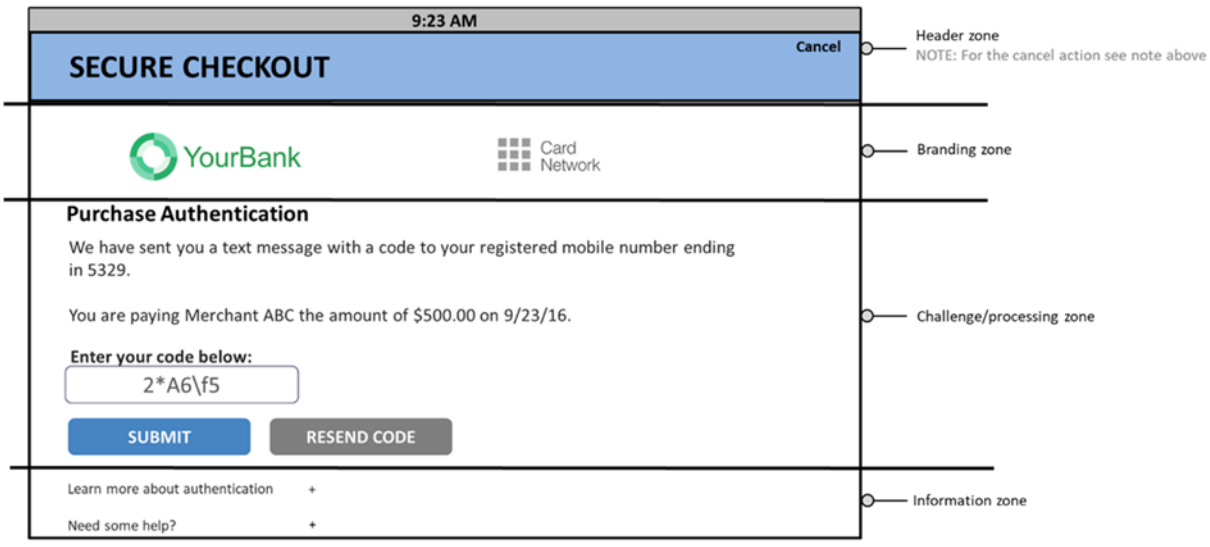
This SB 224v1 contains new Figures added to Chapter 4 of the 3-D Secure Protocol and Core Functions specification. Existing graphics with no updates were renumbered accordingly and are not included in this draft bulletin.

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 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 illustrate 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

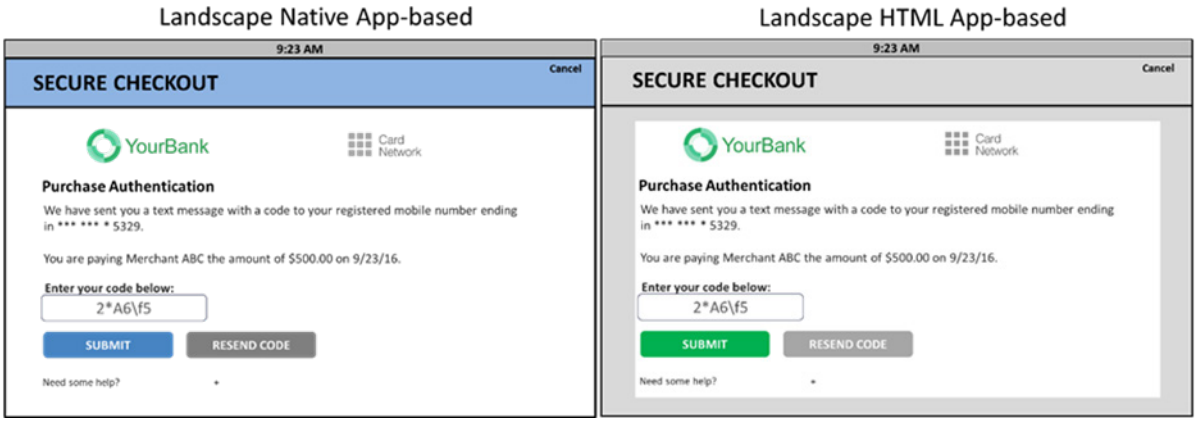
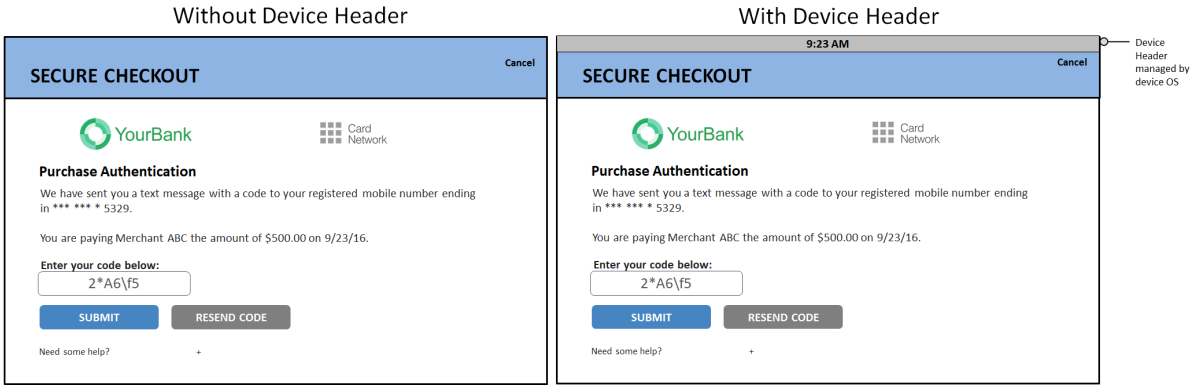


Figure 4.5: Sample Native UI OTP/Text Template with/without Device Header

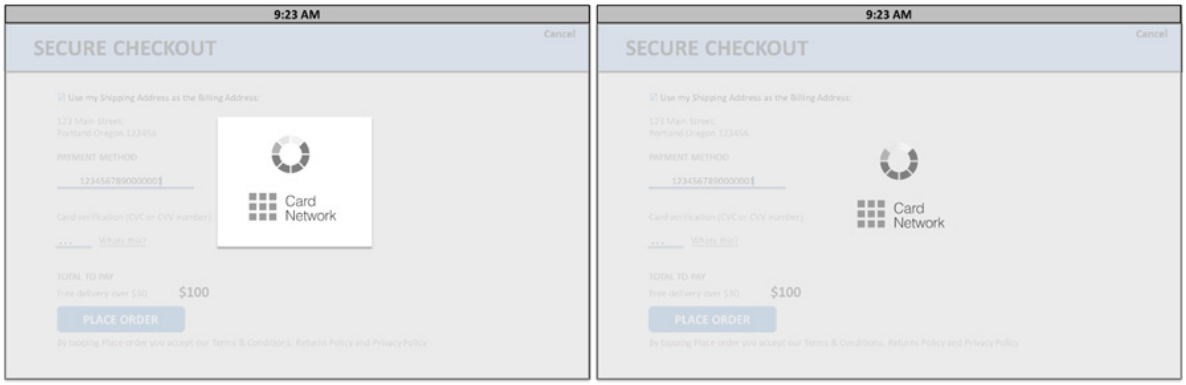
Note: The device header is optional and may not be present depending on the 3DS Requestor implementation and OS constraints. Figure 4.5 depicts sample formats with or without a device header.



4.2.1 Processing Screen Requirements

Figure 4.7~~Figure 4.4~~ and Figure 4.8 provide 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.

4.2.2.1 3DS SDK/ACS

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

[Req 362]
For the ACS UI Type, display the supported UI 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.

[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.

4.2.3 Native UI Templates

Figure 4.11~~Figure 4.4~~ through Figure 4.21~~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 4.11~~Figure 4.8~~ and Figure 4.12 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 4.12: Sample Native UI OTP/Text Template—PA—Landscape

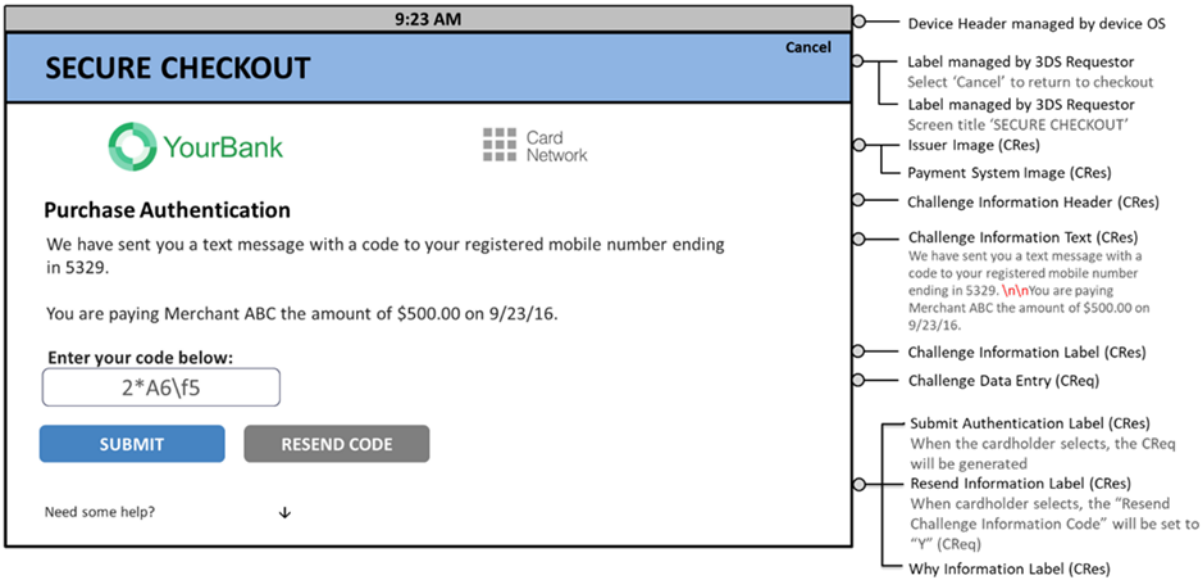


Figure 4.14~~Figure 4.10~~ and Figure 4.15 provide sample formats that allow multiple options to be presented to the Cardholder to obtain single response. For example, asking the Cardholder if they prefer the OTP to be sent to the Consumer Device or to the email address on file.

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

Figure 4.15: Sample Native UI—Single-select Information—PA—Landscape

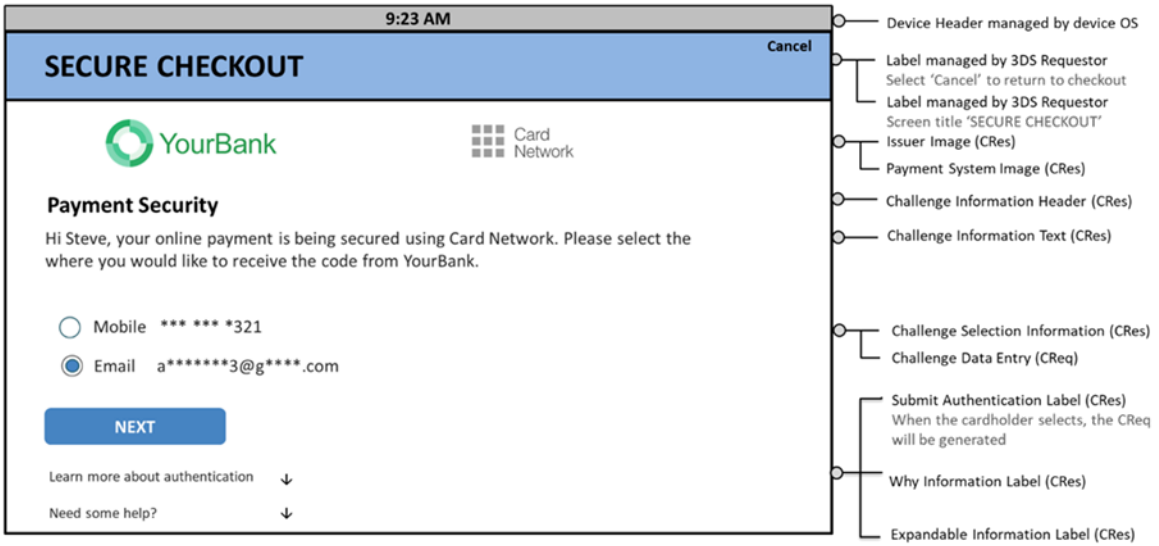


Figure 4.16~~Figure 4.11~~ and Figure 4.17 provide sample formats that allows multiple options to be presented to the Cardholder to obtain multiple responses on a single screen. For example, asking the Cardholder to select the cities where they have lived. This example also depicts a screen with no Issuer or Payment System branding.

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

Figure 4.17: Sample Native UI—Multi-select Information—PA—Landscape

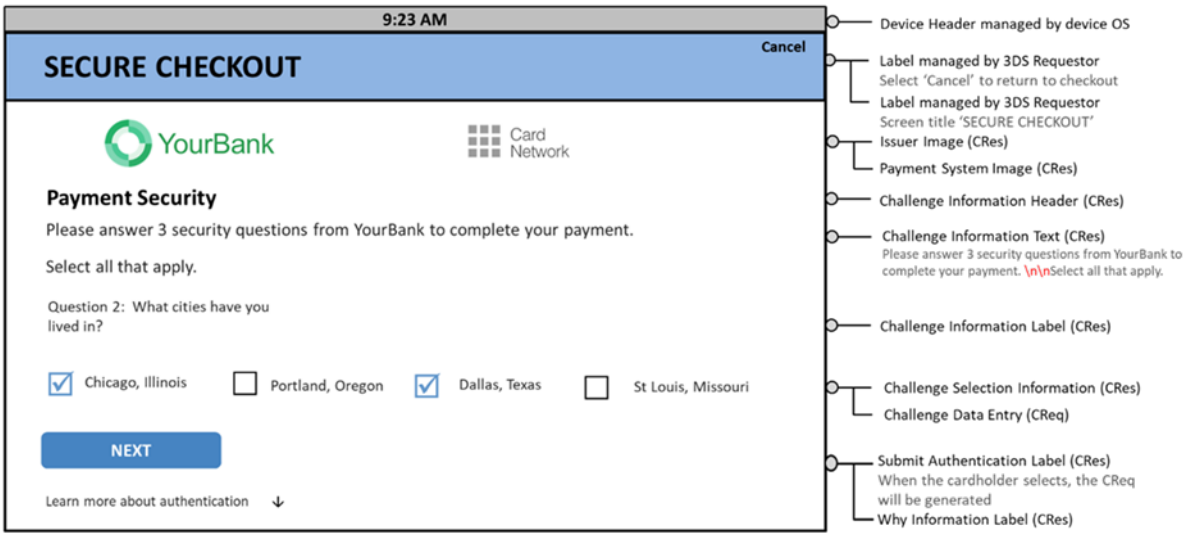
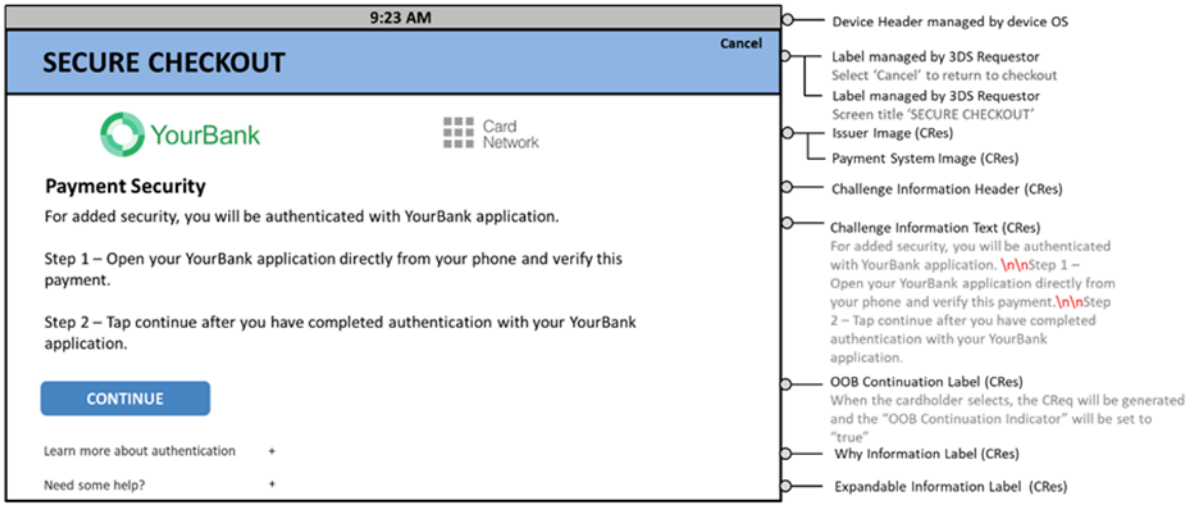


Figure 4.18~~Figure 4.12~~ and Figure 4.19 provide sample OOB formats to display instructions to the Cardholder.

Figure 4.19: Sample OOB Native UI Template—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. 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 4.22~~Figure 4.15~~ and Figure 4.23 provide sample Payment Authentication HTML OTP UI templates that includes Issuer branding.

Figure 4.23: Sample HTML UI/OTP/Text Template—PA—Landscape

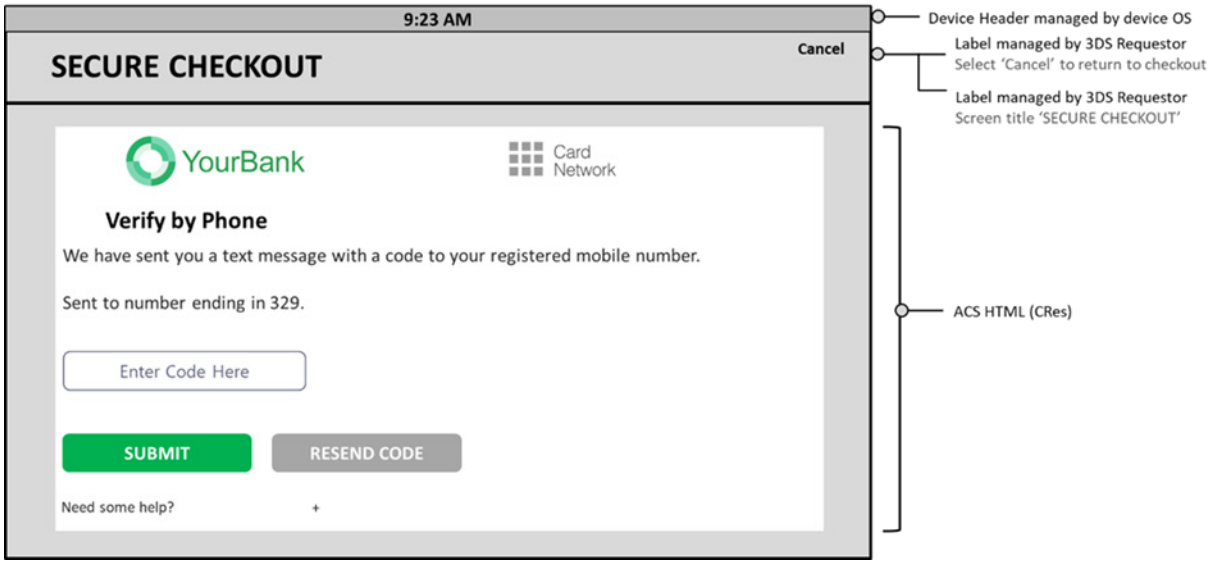
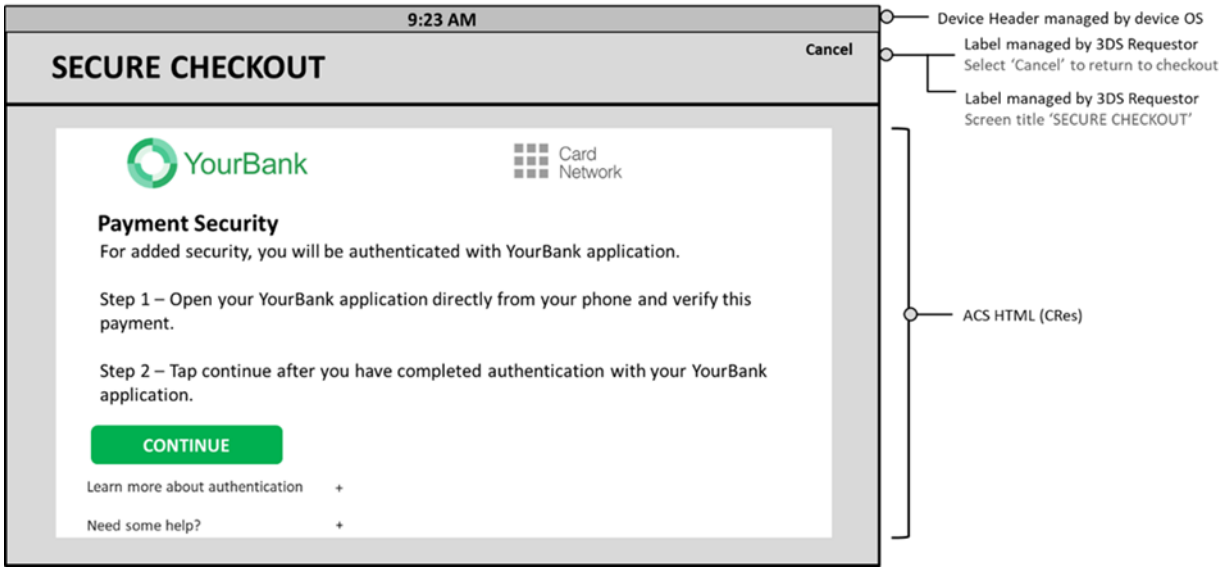


Figure 4.25~~Figure 4.17~~ and Figure 4.26 provide sample templates illustrating the OOB HTML UI.

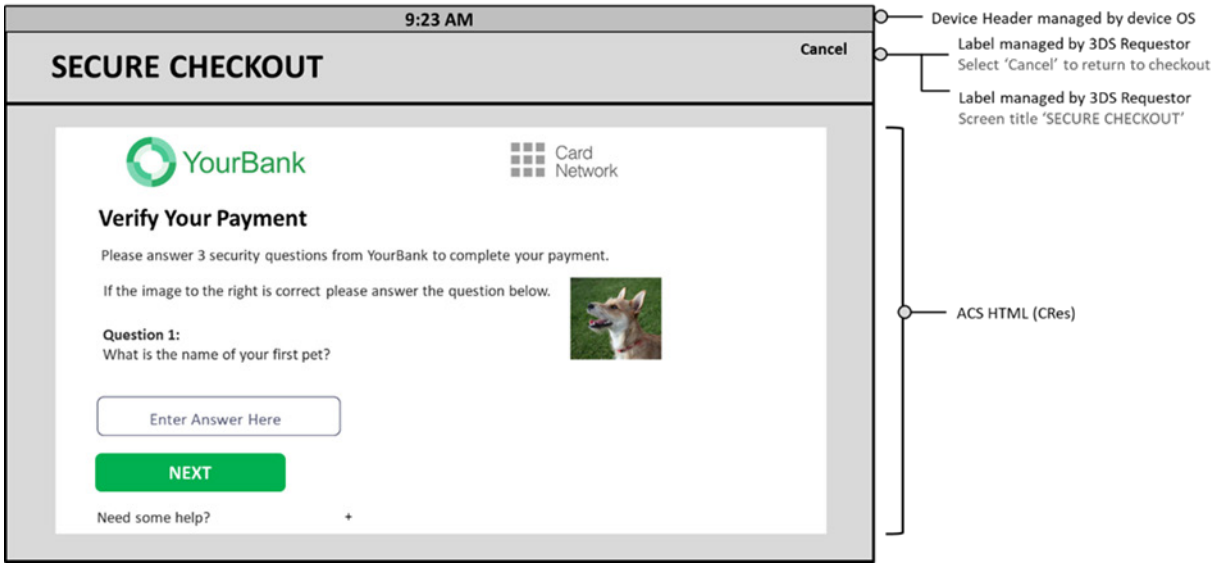
Figure 4.26: Sample OOB HTML UI Template—PA—Landscape



4.2.6.1 HTML Other UI Template

Figure 4.27~~Figure 4.18~~ and Figure 4.28 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 4.28: 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.

Annex A 3-D Secure Data Elements

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
Challenge Additional Information Text	challengeAddInfo	3	3	3	N	N	N	U
Challenge Information Header	challengeInfoHeader	3	2	2	Y	Y	Y	Y
Challenge Information Label	challengeInfoLabel	3	4	4	Y	Y	Y	N
Challenge Information Text	challengeInfoText	3	3	3	Y	Y	Y	Y
Challenge Information Text Indicator	challengeInfoTextIndicator	3	3	3	Y	Y	Y	Y
Challenge Selection Information	challengeSelectInfo	3	5	5	N	Y	Y	N
Expandable Information Label	expandInfoLabel	4	12	12	Y	Y	Y	Y
Expandable Information Text	expandInfoText	4	13	13	Y	Y	Y	Y
Issuer Image	issuerImage	2	1	1	Y	Y	Y	Y

Data Element	Field Name	Zone	Portrait Top-down Display Order	Landscape Top-down Display Order	ACS UI Type			
					OTP	Single Select	Multi Select	OOB
OOB Continuation Label	`oobContinueLabel`	3	6	6	N	N	N	Y
Payment System Image	`psImage`	2	1	1	Y	Y	Y	Y
Resend Information Label	`resendInformationLabel`	3	8	6	Y	N	N	N
Submit Authentication Label	`submitAuthenticationLabel`	3	7	6	Y	Y	Y	N
Why Information Label	`whyInfoLabel`	4	10	10	Y	Y	Y	Y
Why Information Text	`whyInfoText`	4	11	11	Y	Y	Y	Y

May 2019 v5

The following text is provided for clarification and is not included in the 3-D Secure specification.

A 3DS Server supporting only version 2.1.0 of the specification that receives a Preparation Response (PRes) Message should be aware that the ACS Start Protocol Version, ACS End Protocol Version, DS Start Protocol Version and DS End Protocol Version could contain any active Protocol Version Number listed in Table 1.5 of the most recent EMV® 3-D Secure Protocol and Core Functions Specification.

For example, since the release of the version 2.2.0 specification, the DS Start Protocol Version and DS End Protocol Version could now be equal to 2.2.0 in a 2.1.0 PRes message. Therefore, the start and end protocol version can be any active protocol version number for an EMV® 3DS Specification issued by EMVCo.

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

4.1 3-D Secure User Interface Templates

The 3DS SDK shall:

[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.

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~~.

[Req 359]

For the App-based HTML UI Type and Browser-based UI, create HTML with form ~~UI data~~ elements within the applicable zones as outlined ~~defined in Table A.18 and depicted~~ in Figure 4.1. The ~~expected~~ format is outlined~~depicted~~ in Sections 4.2.6 and 4.3.3.

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.

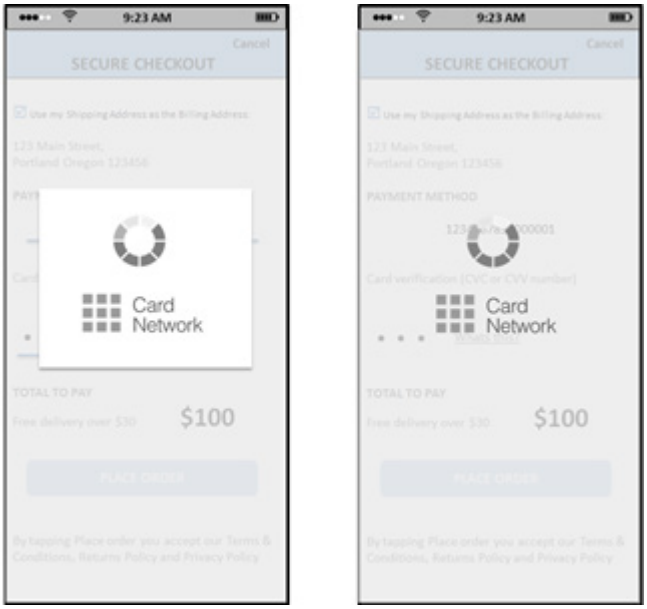
Note: Some platforms may not natively support all image types.

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 as an 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]**~~

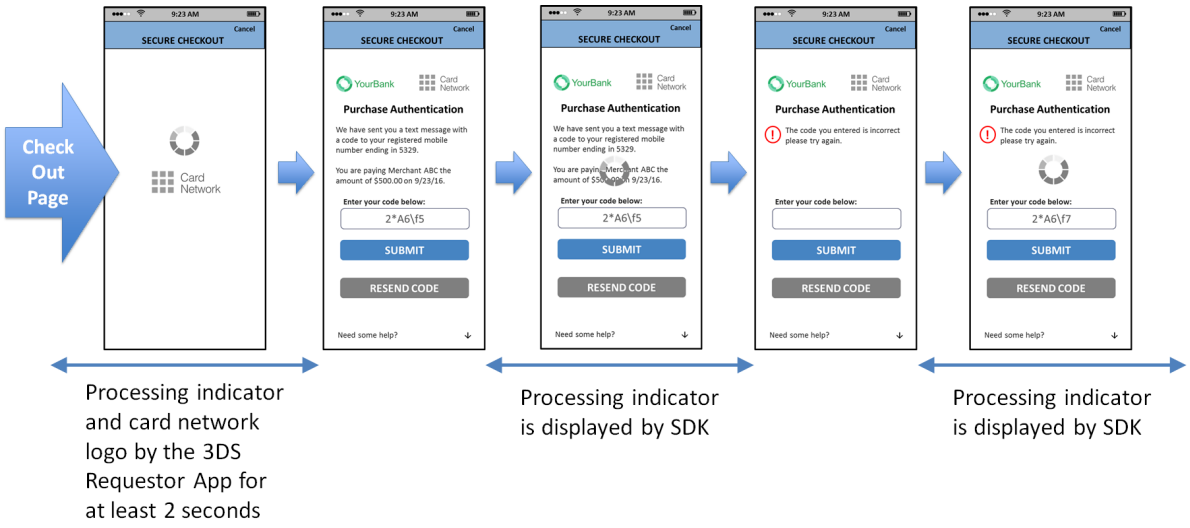
~~Display the Cancel action in the top right corner of the Header zone.~~

[Req 389]

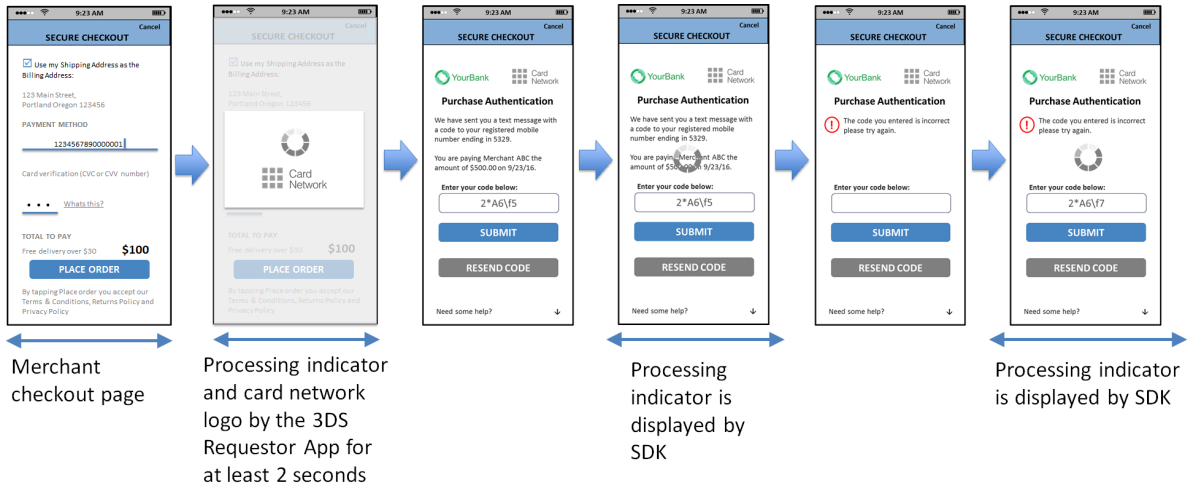
Ensure that the Cancel action is not actionable on the Processing screen.

New Graphic for Figure 4.5

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

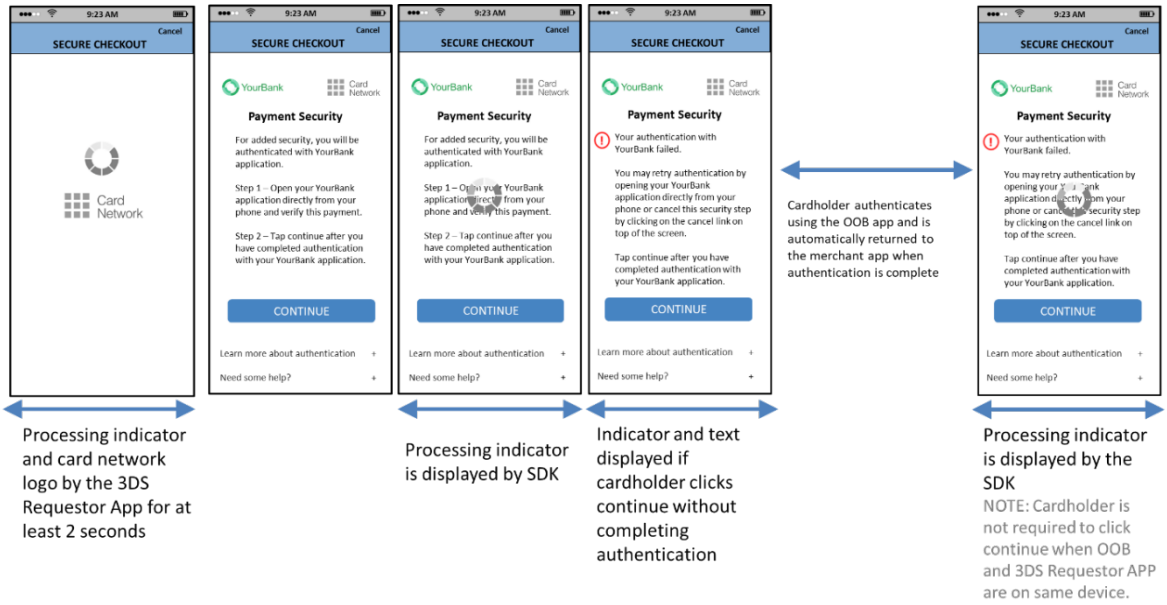


***(Updated)* Figure 4.5: Sample OTP/Text Template—App-based Processing Flow**

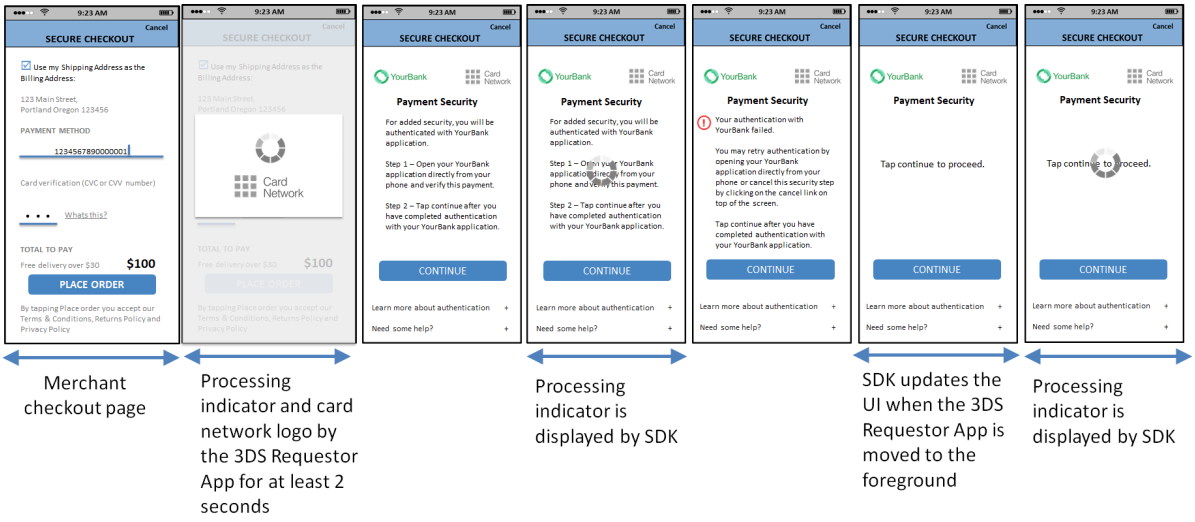


New Graphic for Figure 4.6

***(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**



4.2.2.1 3DS SDK/ACS

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

[Req 362]

~~Display all UI elements provided by the ACS within the applicable zones as defined in Table A.18 and depicted in Figure 4.1. The expected format is depicted in section 4.2.3.~~

For the ACS UI Type, display the supported UI data elements in their applicable zones and order as defined in Table A.18 and depicted in Figure 4.1. The expected format is depicted in sections 4.2.3 and 4.2.6.

If the 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 369]

Display the Cancel action in the top ~~right~~ corner of the header zone as depicted in Figure 4.1.

The ACS shall for the CReq/CRes message exchange:

[Req 387]

Only include the UI data elements supported for the selected ACS UI Type as defined in Table A.18.

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.

4.2.5.1 3DS SDK/ACS

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

[Req 373]

Display the Cancel action in the top ~~right~~ corner of the header zone as depicted in Figure 4.1.

The ACS shall for the CReq/CRes exchange:

[Req 374]

Create HTML with ~~the UI~~form elements in the applicable zones ~~as defined in Table A.18 and depicted~~ outlined in Figure 4.1. The expected format is ~~depicted~~ outlined in Section 4.2.6.

4.2.7.3 3DS SDK

The 3DS SDK shall:

[Req 171]

- The web view will return~~, either~~ a parameter string (HTML Action = GET) ~~or form data (HTML Action = POST)~~ containing the cardholder's data input.

4.3.1 Processing Screen Requirements

The ACS shall:

[Req 177]

Create and maintain versions of the HTML that correspond to the sizes of the Challenge Window Size data element as defined in Table A.1 and provide the appropriate size in the CRes message based upon the Challenge Window Size that was provided by the 3DS Server in the ~~ARes~~AReq message.

~~**[Req 379]**~~

~~Create HTML with the UI elements in the Branding, Challenge/Processing and Information zones as defined in Table A.18 and depicted in the UI templates in Section 4.3.3.~~

4.3 Browser-based User Interface Overview

4.3.2.1 ACS

The ACS shall for the CReq/CRes exchange:

[Req 380]

Create HTML with t~~he UI~~form elements in the applicable zones as ~~defined in Table A.18 and depicted~~outlined in Figure 4.1. The ~~expected~~ format is ~~depicted~~ outlined in the UI templates in Section 4.3.3.

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.

For 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.

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~~

- 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~~

  - 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:

- 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 $CEK_{S\text{-}A}$ 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 $CEK_{A\text{-}S}$ 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	Conditional Inclusion
ACS Rendering Type	Identifies the ACS ~~UI~~ Interface and ACS UI Template that the ACS will first present to the consumer.						
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	Conditional Inclusion
Cardholder Email Address							Required (if available) unless market or regional mandate restricts sending this information.
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.						
DS Start Protocol Version	The ~~most recent~~ earliest (i.e. oldest) active protocol version that is supported for the DS.						

Data Element/ Field Name	Description	Source	Length/Format/ Values	Device Channel	Message Category	Message Inclusion	Conditional Inclusion
OOB App Label							Note: This element has been defined to support future enhancements to the OOB message flow. An ACS will not provide this value and a 3DS SDK will not perform any processing and will not display the OOB App Label in this version of the specification.
OOB App URL							Note: this element has been defined to support future enhancements to the OOB message flow. An ACS will not provide this value and a 3DS SDK will not perform any processing of the OOB App URL in this version of the specification.

Data Element/ Field Name	Description	Source	Length/Format/ Values	Device Channel	Message Category	Message Inclusion	Conditional Inclusion
Serial Number			Length: Variable, maximum 20 alphanumeric characters	~~01-APP~~ ~~02-BRW~~ N/A	~~01-PA~~ ~~02-NPA~~ N/A		

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.	
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.

A.7.5 ACS Rendering Type

Table A.12 ACS Rendering Type

Data Element/Field Name	Description	Length/Format/Values
ACS UI Template	Note: HTML Other is only valid in combination with 02 = HTML UI. If used with 01 = Native UI, the DS will respond with Error = 203 as described in Sections 5.9.3 and 5.9.8.	

A.8 UI Data Elements

Table A.18 UI Data Elements

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

Data Element	Field Name	Zone	Top-down Display Order	ACS UI Type			
				OTP	Single Select	Multi Select	OOB
~~ACS HTML~~	~~acsHTML~~	~~The placement of UI data elements in the HTML is identical to the Native UI elements described in this table~~.					

Data Element	Field Name	Zone	Top-down Display Order	ACS UI Type			
				OTP	Single Select	Multi Select	OOB
~~ACS HTML Refresh~~	~~acsHTMLRefresh~~	~~The placement of UI data elements in the HTML is identical to the Native UI elements described in this table~~					
Challenge Additional Information Text	challengeAddInfo	3	3	N	N	N	Y
Challenge Information Header	challengeInfoHeader	3	2	Y	Y	Y	Y
Challenge Information Label	challengeInfoLabel	3	4	Y	Y	Y	N
Challenge Information Text	challengeInfoText	3	3	Y	Y	Y	Y
Challenge Information Text Indicator	challengeInfoTextIndicator	3	3	Y	Y	Y	Y
Challenge Selection Information	challengeSelectInfo	3	5	N	Y	Y	N
Expandable Information Label	expandInfoLabel	4	12	Y	Y	Y	Y
Expandable Information Text	expandInfoText	4	13	Y	Y	Y	Y
Issuer Image	issuerImage	2	1	Y	Y	Y	Y

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

Chapter 1 Introduction

1.9 Terminology and Conventions

Increment(s)

A 3DS component may be required to increment a counter in which case the increment is increasing the counter by one.

Chapter 3 EMV 3-D Secure Authentication Flow Requirements

3.1 App-based Requirements

Step 7: The ACS

[Req 386]

Check whether the SDK Device Information data version number is recognised.

If not recognised, the ACS proceeds with processing the transaction and does not error due to the unrecognised Data Version Number.

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

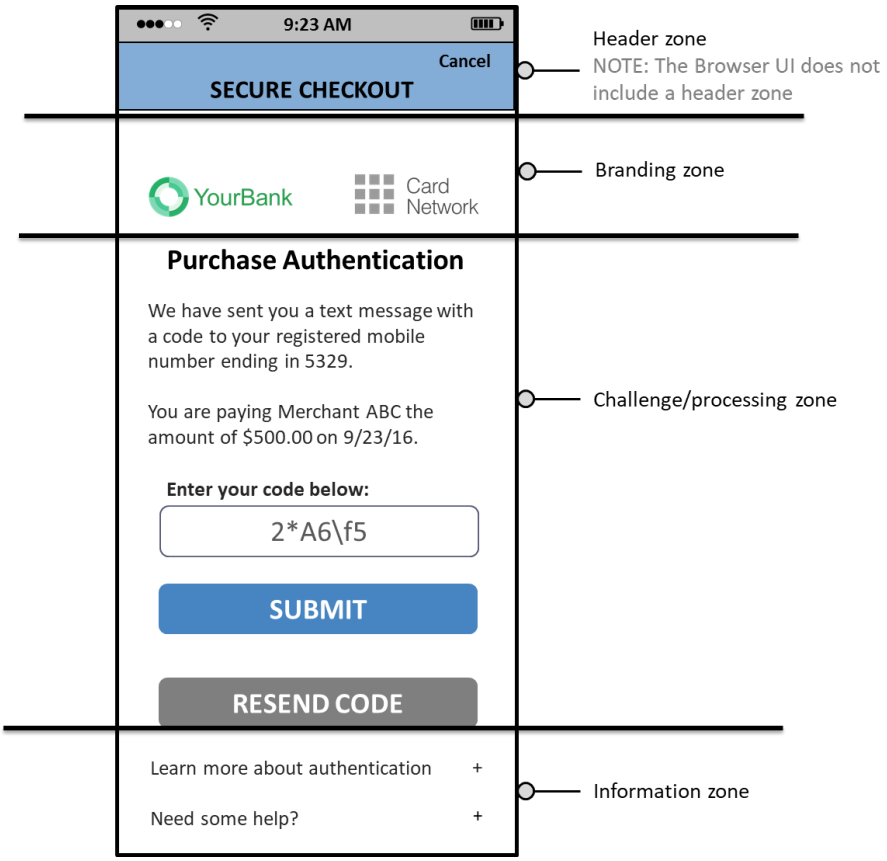
4.1 3-D Secure User Interface Templates

To facilitate this consistency, the UI layout is defined in zones as follows:

- **Header zone (Zone 1)**—Contains all labels managed by the 3DS Requestor and is located at the top of the screen.
- **Branding zone (Zone 2)**—Contains all logos and is located between the Header and Challenge zone.
- **Challenge/Processing zone (Zone 3)**—Contains processing and challenge information and is located between the Branding zone and the Information zone.
- **Information zone (Zone 4)**—Contains additional information for the cardholder and is located at the bottom of the screen.

Figure 4.1 illustrates the zones and placement of UI data elements within the zones.

Figure 4.1: UI Template Zones (NEW)



Note: With the addition of a new Figure 4.1, all subsequent Chapter 4 figures were renumbered.

The SDK shall:

[Req 358]

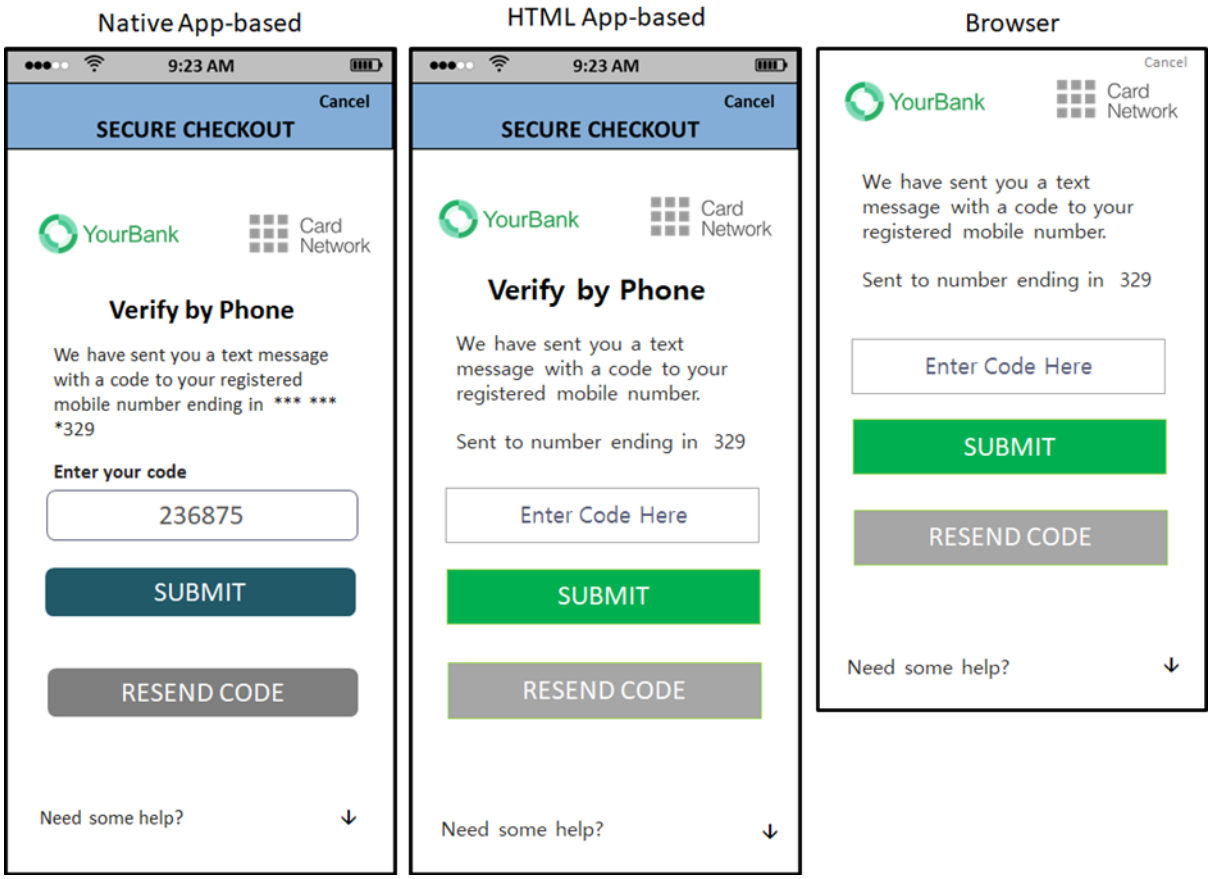
Display UI data elements provided by the ACS within the applicable zones as defined in Table A.18 and depicted in Figure 4.1. The expected format is depicted in sections 4.2.3 and 4.2.6.

The ACS shall:

[Req 359]

Create HTML with UI data elements within the applicable zones as defined in Table A.18 and depicted in Figure 4.1. The expected format is depicted in Section 4.3.3.

Figure 4.2 UI Template Examples—All Device Channels (NEW)



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, ~~integrate~~ the DS logo into the centre of the Processing screen.

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

[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 151]

Display the Processing screen for a minimum of one second during the second and subsequent CReq/CRes message cycle (For the first CReq/CRes message cycle see [Req 153]).

[Req 361]

Display the Cancel action in the top right corner of the Header zone.

4.2.2 Native UI ~~Templates~~ Display Requirements

4.2.2.1 3DS SDK/ACS

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

[Req 362]
Display all UI elements provided by the ACS within the applicable zones as defined in Table A.18 and depicted in Figure 4.1. The expected format is depicted in Section 4.2.3.

[Req 363]
Interpret and place the carriage return on the screen when provided in the CRes message by the ACS.

[Req 364]
Not display any UI elements other than those provided in the CRes message by the ACS in the Branding, Challenge/Processing and Information zones.

[Req 365]
Display the Expandable Information Label as a graphical control element that can be expanded (for example, an accordion).

[Req 366]
Display the Expandable Information Text only when the user selects the Expandable Information Label.

[Req 367]
Display the Why Information Label as a graphical control element that can be expanded (for example, an accordion).

[Req 368]
Display the Why Information Text only when the user selects the Why Information Label.

[Req 369]
Display the Cancel action in the top right corner of the header zone as defined in Figure 4.1.

The ACS shall for the CReq/CRes message exchange:

[Req 370]
If a carriage return is used, then represent the carriage return as specified in Table A.1 for the following data elements:

- Challenge Information Text
- Expandable Information Text
- Why Information Text

4.2.3 Native UI Templates

4.2.4 Native UI Message Exchange Requirements

The 3DS SDK shall:

[Req 153]

After submitting the CReq message to the ACS, display the ~~3DS Requestor App~~ Processing screen until the CRes message is received, or timeout is exceeded. Refer to Section 5.5.2.2 for CReq/CRes message Timeout requirements.

4.2.5 HTML UI T~~emplates~~ Display Requirements

Details of the HTML UI and the rendering process are separately described in the EMV 3-D Secure SDK Specification and in the documentation provided by each DS. ~~The HTML UI templates provide Issuers the ability to include Issuer-specific design elements (e.g. branding, colours, fonts) as shown in the following figures:~~

4.2.5.1 3DS SDK/ACS

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

[Req 371]

Display the HTML as provided by the ACS.

[Req 372]

Display only the UI elements provided by the ACS in the Branding, Challenge/Processing and Information zones.

[Req 373]

Display the Cancel action in the top right corner of the header zone as defined in Figure 4.1. Note: The functionality of UI elements in the header zone are managed by the 3DS SDK.

The ACS shall for the CReq/CRes exchange:

[Req 374]

Create HTML with the UI elements in the applicable zones as defined in Table A.18 and depicted in Figure 4.1. The expected format is depicted in Section 4.2.6.

The ACS shall, if using the following optional data elements, provide the:

[Req 375]

Expandable Information Label for display as a graphical control element that can be expanded (for example, an accordion) in the Information zone.

[Req 376]

Expandable Information Text for display in the Information zone only when the user selects the Expandable Information Label.

[Req 377]

Why Information Label for display as a graphical control element that can be expanded (for example, an accordion) in the Information zone.

[Req 378]

Why Information Text for display in the Information zone only when the user selects the Why Information Label.

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.

4.3 Browser-based User Interface Overview

~~The figures provided in this section depict examples of the Issuer content and format, as well as 3DS Requestor website placement.~~

4.3.1.1 3DS Requestor Website

The 3DS Requestor website shall:

[Req 172]

Create a Processing screen with a Processing Graphic (for example, a progress bar or a spinning wheel) for display during the AReq/ARes message cycle.

[Req 173]

Display the Processing screen ~~a graphical element (for example, a progress bar or a spinning wheel)~~ that conveys ~~to indicate~~ to the Cardholder that processing is occurring (Refer to Figure 4.20~~Figure 4-17~~ and Figure 4.21~~Figure 4-18~~ for examples).

[Req 174]

Include the DS logo for display at the centre of the screen unless specifically requested not to include.

4.3.1.2 ACS

The ACS shall:

[Req 379]

Create HTML with the UI elements in the Branding, Challenge/Processing and Information zones as defined in Table A.18 and depicted in the UI templates in Section 4.3.3.

[Req 179]

Display a graphical element (for example, a progress bar or a spinning wheel) within the Challenge/Processing zone that conveys to the consumer that processing is occurring.

[Req 180]

Include the DS logo for display during the challenge flow, (with the exception of the Processing screen) unless specifically requested not to include.

[Req 182]

Display the Processing screen for a minimum of ~~two~~one second~~s~~.

4.3.2 Browser Display Requirements

The browser will display the HTML as provided by the ACS. As such, it is the ACS responsibility to format the HTML to best display on the Consumer Device.

4.3.2.1 ACS

The ACS shall for the CReq/Cres exchange:

[Req 380]
Create HTML with the UI elements in the applicable zones as defined in Table A.18 and depicted in Figure 4.1. The expected format is depicted in the UI templates in Section 4.3.3.

The ACS shall, if using the following optional data elements, provide the:

[Req 381]
Expandable Information Label for display as a graphical control element that can be expanded (for example, an accordion) in the Information zone.

[Req 382]
Expandable Information Text for display in the Information zone only when the user selects the Expandable Information Label.

[Req 383]
Why Information Label for display as a graphical control element that can be expanded (for example, an accordion) in the Information zone.

[Req 384]
Why Information Text for display in the Information zone when the user selects the Why Information Label.

4.3.3 Browser UI Templates

The figures provided in this section depict examples of the Issuer content and format, as well as the 3DS Requestor website placement.

Figure 4.19: App-based HTML and Browser UI Comparison (NEW)

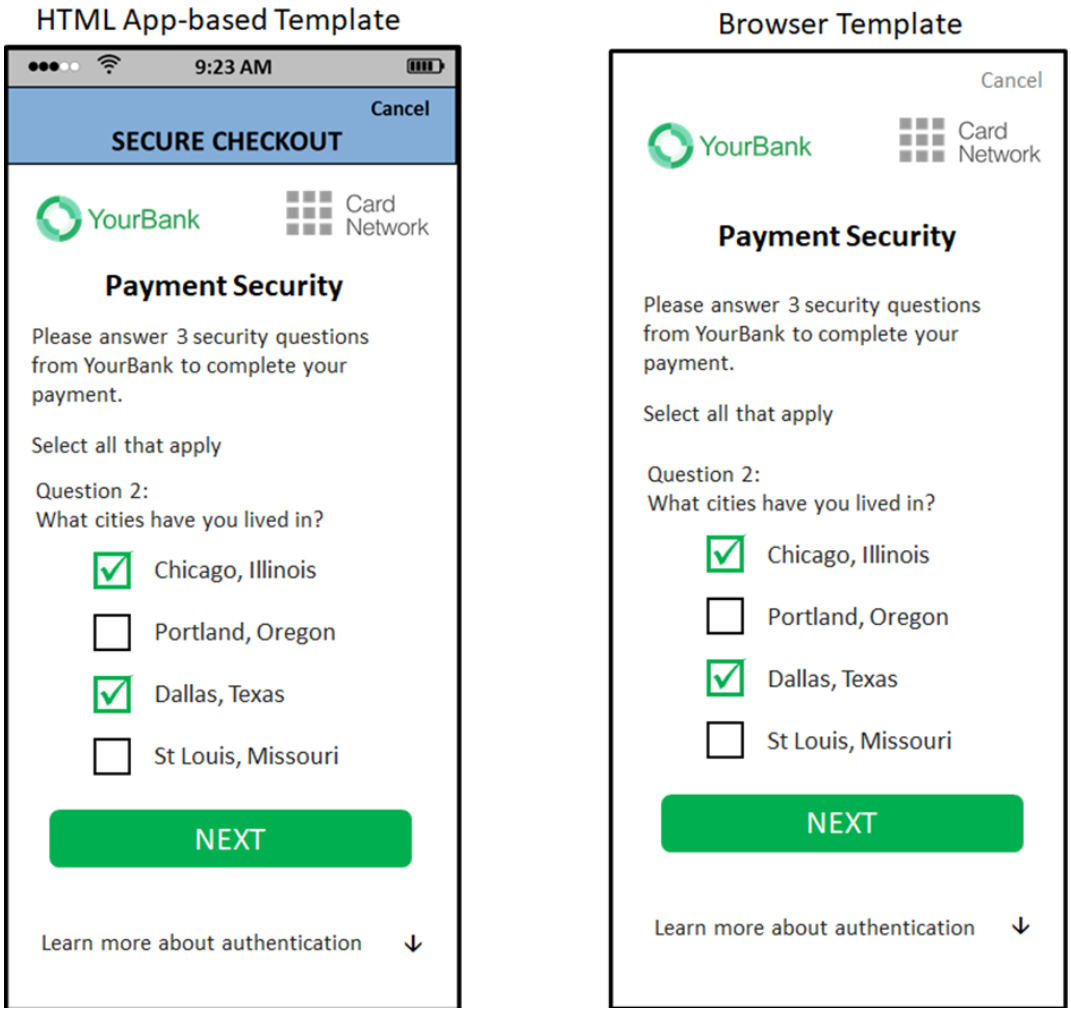


Figure 4.22: Sample Browser with Lightbox UI—PA (NEW)

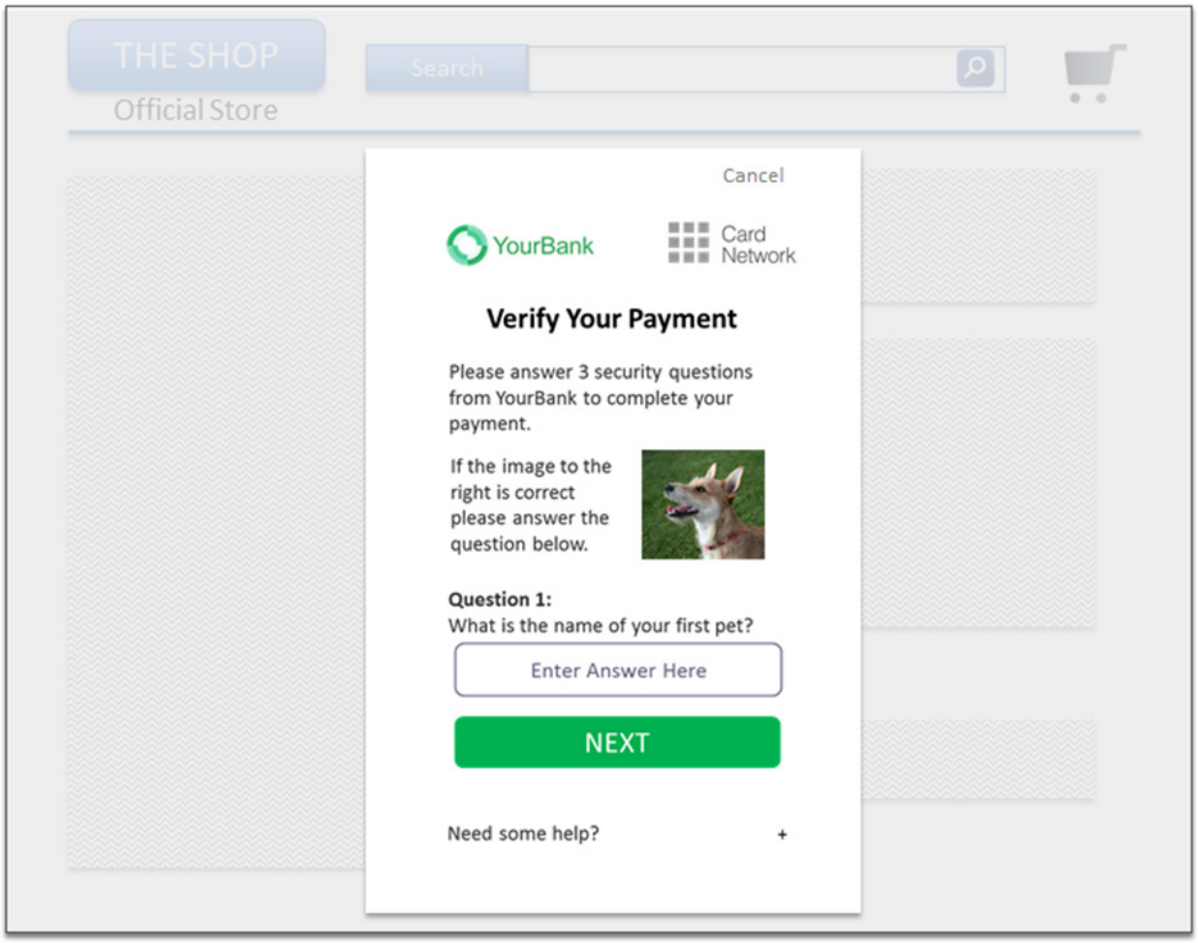
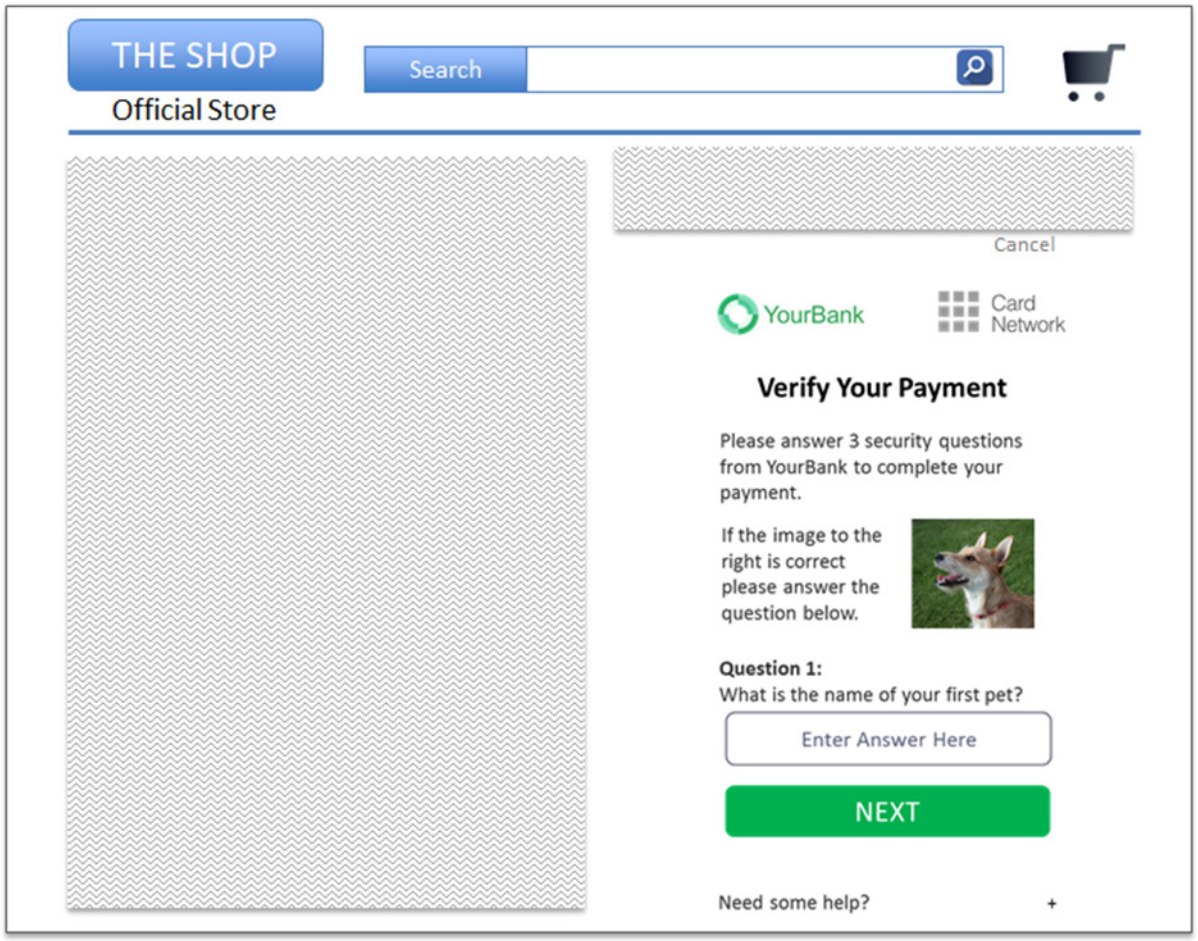


Figure 4.23 Sample Browser with Inline UI—PA (NEW)



Chapter 5 EMV 3-D Secure Message Handling Requirements

5.1.6 Message Content Validation

The message validation criteria are based on the Message Type field and apply as follows:

[Req 209]

If there are additional data elements received that are not specified for the Message Type, Device Channel and Message Category but the message otherwise passes validation, the message shall be considered valid.

~~However, the additional elements (with the exception of data extensions) shall be ignored and shall not be sent to the next 3DS component in the flow. OR,~~ For the additional data elements received (with the exception of data extensions), the receiving 3DS component shall EITHER:

- ~~If the additional data elements in the AReq message do not pass validation criteria, the DS responds with an error message to the 3DS Server.~~Ignore the additional data elements and not send them to the next 3DS component in the flow.

  OR

- Check the format of the additional data elements:
  - If the format is correct, ignore the additional data elements and do not send them to the next 3DS component in the flow.
  - If the format is incorrect, the receiving 3DS component responds with an error message to the sending 3DS component.

    Example:

    The DS receives an AReq message from the 3DS Server with additional data elements that are not specified in Table A.1 for the AReq Message Type, Device Channel and Message Category and the DS validates the AReq content and drops the additional elements when sending the AReq message to the ACS.

    OR

    If the additional data elements in the AReq message do not pass ~~validation criteria~~format checking, the DS can respond with an Error Message to the 3DS Server.

5.5.1 Transaction Timeouts

[Req 221]

If the transaction reaches the 30-second timeout expiry, send an RReq message to the DS to be passed to the 3DS Server with Transaction Status = N, Transaction Status Reason = 14 (~~Challenge~~ Transaction ~~Timed~~ timed ~~Out~~out at the ACS), and Challenge Cancelation Indicator = 05 (Transaction timed out at the ACS—First CReq not received).

[Req 224]

If the timeout expires before receiving the next CReq message from the 3DS SDK, send an RReq message to the DS to be passed to the 3DS Server with Transaction Status = N, Transaction Status Reason = 14 (~~Challenge~~ Transaction ~~Timed~~ timed ~~Out~~out at the ACS), and Challenge Cancelation Indicator = 04 and then clear any ephemeral key generated and stored for use in the CReq/CRes message exchange for this transaction.

5.6 PReq/PRes Message Handling Requirements

[Req 250]

- If the PReq message does not include a Serial Number, or if the DS does not support partial cache update, the DS PRes message response shall contain all Card Range Data using only Action Indicator = A.

[Req 304]

Receive and validate the PRes message as defined in Table B.7:

- If any data element present fails validation, the 3DS Server:
  - Returns to the DS an Error Message (as defined in Section A.5.5) with Error Component = S and Error Code = 203.
- If any required data elements are missing, the 3DS Server:
  - Returns to the DS an Error Message (as defined in Section A.5.5) with Error Component = S and Error Code = 201.

~~- If an error is identified in the Card Range Data, the 3DS Server:~~
  ~~- Resubmits the PReq message without the Serial Number.~~

[Req 385]

Update the cache information for each Card Range Data according to the Action Indicator.

- If the PRes message does not include a Serial Number, the 3DS Server:
  - Replaces all existing Card Range Data for the DS.
- If an error is identified in the Card Range Data, the 3DS Server:
  - Resubmits the PReq message without the Serial Number.

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	Conditional Inclusion
Challenge Cancelation Indicator							Value of 04 or 05 is required when Transaction Status Reason = 14.

Data Element/ Field Name	Description	Source	Length/Format/ Values	Device Channel	Message Category	Message Inclusion	Conditional 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, AND • Resend Challenge Information Code are not present See Table A.14 for Challenge Data Entry conditions.

Data Element/ Field Name	Description	Source	Length/Format/ Values	Device Channel	Message Category	Message Inclusion	Conditional Inclusion
Challenge HTML Data Entry							Required when • ACS UI Type = 05. AND ~~challenge data has been entered into the UI.~~ • Challenge Cancelation Indicator is not present.
OOB Continuation Label							Note: If present, either of the following must also be present: • Challenge Information Header, OR • Challenge Information Text

Data Element/ Field Name	Description	Source	Length/Format/ Values	Device Channel	Message Category	Message Inclusion	Conditional Inclusion
SDK App ID	Universally unique ID created upon all installations ~~and updates~~ of the 3DS Requestor App on a Consumer Device. This will be newly generated and stored by the 3DS SDK for each installation ~~or update~~.						
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
3DS Requestor Prior Transaction Authentication Information			~~Format: String~~ ~~Values accepted:~~				

Data Element/ Field Name	Description	Source	Length/Format/ Values	Device Channel	Message Category	Message Inclusion	Conditional Inclusion
Transaction Status	Note: The Final CRes message can contain only a value of Y or N.					01-PA: Final CRes = ~~C~~R 02-NPA: Final CRes = C	~~For 01-PA, the CRes, only present in the final CRes message.~~ For 02-NPA, Conditional as defined by the DS. See Table A.15 for 01-PA Transaction Status conditions. ~~Note: CRes indicates Final CRes.~~

A.5.3 3DS Method Data

3DS Method Data Examples

- Example 1: `threeDSMethodData` to be sent to ACS in the 3DS Method HTTP form POST from 3DS Requestor

```
<form name="frm" method="POST" action="Rendering URL">
<input type="hidden" name="threeDSMethodData"
value="eyJ0aHJlZURTU2VydmVyVHJhbnNJRCI6IjNhYzdjYWE3LWFhNDItMjY2My03OTFiLTJhYzA1YTU0MmM0YSIsInRocmVlRFNNZXRob
2ROb3RpZmljYXRpb25VUkwiOiJ0aHJlZURTTWV0aG9kTm90aWZpY2F0aW9uVVJMIn0">
</form>

Decoded threeDSMethodData:
{"threeDSServerTransID":"3ac7caa7-aa42-2663-791b-
2ac05a542c4a","threeDSMethodNotificationURL":"threeDSMethodNotificationURL"}
```

- **Example 2**: `threeDSMethodData` to be sent to 3DS Method Notification URL from the ACS

```
<form name="frm" method="POST" action="threeDSMethodNotificationURL">
<input type="hidden" name="threeDSMethodData"
value="eyJ0aHJlZURTU2VydmVyVHJhbnNJRCI6IjNhYzdjYWE3LWFhNDItMjY2My03OTFiLTJhYzA1YTU0MmM0YSJ9">
</form>

Decoded threeDSMethodData:
{"threeDSServerTransID":"3ac7caa7-aa42-2663-791b-2ac05a542c4a"}
```

A.5.5 Error Code, Error Description, and Error Detail

Table A.4 Error Code, Error Description, and Error Detail

Value	Error Code	Error Description	Error Detail
301			Invalid meaning Transaction ID not recognised, or Transaction ID is recognised as a duplicate.

A.7.1 Cardholder Account Information

Table A.8 Cardholder Account Information

Data Element/Field Name	Description	Length/Format/Values
Number of Provisioning Attempts Per Day	Example values: • 2 • 02 • 002	JSON Data Type: String
Number of Transactions Per Day	Example values:	

	• 2 • 02 • 002	
Number of Transactions Per Year	Example values: • 2 • 02 • 002	

A.7.2 Merchant Risk Indicator

Table A.9 Merchant Risk Indicator

Data Element/Field Name	Description	Length/Format/Values
Gift Card Amount	Example: gift card amount is USD 123.45: Values accepted: • 123 • 0123 • 00123	
Gift Card Currency	For prepaid or gift card purchase, ISO 4217 three-digit currency code of the gift card, other than those listed in Table A.5.~~the currency code of the card as defined in ISO 4217 other than those listed in Table A.5.~~	Length: 3 characters; numeric

A.7.3 3DS Requestor Authentication Information

Table A.10 3DS Requestor Authentication Information

Data Element/Field Name	Description	Length/Format/Values
3DS Requestor Authentication Data		~~Value accepted: Any~~

A.7.4 3DS Requestor Prior Transaction Authentication Information

Table A.11: 3DS Requestor Prior Transaction Authentication Information

Data Element/Field Name	Description	Length/Format/Values

3DS Requestor Prior Transaction Authentication Data		JSON Data Type: String ~~Format: Any~~

A.7.7 Challenge Data Entry

The Challenge Data Entry (`challengeDataEntry`) contains the data that the Cardholder entered in the Native UI text field. Table A.14 identifies the 3-D Secure message handling when this element is missing, assuming that no other errors are found.

Table A.14: Challenge Data Entry

Challenge Data Entry	ACS UI Type	Challenge Cancelation Indicator	Resend Challenge Information Code	Response
Missing	01, 02, or 03	Missing	Missing	The ACS assumes that the Cardholder has not entered challenge data in the UI and therefore the ACS does not send the 3DS SDK an Error Message, but instead sends a CRes message.
Missing	01, 02, or 03	Present	Missing	The ACS sends the 3DS SDK a CRes message.
Missing	01, 02, or 03	Missing	Present • Value = Y	The ACS sends the 3DS SDK a CRes message.
Missing	01, 02, or 03	Missing	Present • Value = N	The ACS assumes that the Cardholder has not entered challenge data in the UI and therefore the ACS does not send the 3DS SDK an Error Message, but instead sends a CRes message.
Missing	01, 02, or 03	Present	Present	The ACS sends the 3DS SDK an Error Message.

Note that subsequent sections and tables were renumbered accordingly.

A.8 UI Data Elements

Table A.1 outlines the default validation requirements for the CRes message. Table A.18 specifies the placement of UI data elements on the UI with respect to the zones defined in Section 4.1.

Table A.1: UI Data Elements

Data Element	Field Name	Zone
ACS HTML	`acsHTML`	The placement of UI data elements in the HTML is identical to the Native UI elements described in this table.
ACS HTML Refresh	`acsHTMLRefresh`	The placement of UI data elements in the HTML is identical to the Native UI elements described in this table.
Challenge Additional Information Text	`challengeAddInfo`	Zone 3
Challenge Information Header	`challengeInfoHeader`	Zone 3
Challenge Information Label	`challengeInfoLabel`	Zone 3
Challenge Information Text	`challengeInfoText`	Zone 3
Challenge Information Text Indicator	`challengeInfoTextIndicator`	Zone 3
Challenge Selection Information	`challengeSelectInfo`	Zone 3
Expandable Information Label	`expandInfoLabel`	Zone 4
Expandable Information Text	`expandInfoText`	Zone 4
Issuer Image	`issuerImage`	Zone 2
OOB App URL	`oobAppURL`	Zone 3

Data Element	Field Name	Zone
OOB App Label	oobAppLabel	Zone 3
OOB Continuation Label	oobContinueLabel	Zone 3
Payment System Image	psImage	Zone 2
Resend Information Label	resendInformationLabel	Zone 3
Submit Authentication Label	submitAuthenticationLabel	Zone 3
Why Information Label	whyInfoLabel	Zone 4
Why Information Text	whyInfoText	Zone 4

Annex B Message Format

B.4 CRes Message Data Elements

Table B.4 CRes Data Elements

Data Element	Field Name
~~Transaction Status~~	~~transStatus~~

Chapter 3 EMV 3-D Secure Authentication Flow Requirements

3.3 Browser-based Requirements

Step 15: The ACS

[Req 123]

Check the authentication data entered by the Cardholder:

- If correct, then the ACS:
  - ~~Sets the Challenge Completion Indicator = Y~~
- If incorrect and authentication has failed, then the ACS:
  - If the Interaction Counter ≥ ACS maximum challenges, the ACS:
    - ~~Sets the Challenge Completion Indicator = Y~~

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

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.

4.2.5.3 3DS SDK

[Req 171]

- The web view will return, either a parameter string (HTML Action = GET) or ~~a header/body~~form data (HTML Action = POST) containing the cardholder's data input.

Chapter 5 EMV 3-D Secure Message Handling Requirements

5.5.1 Transaction Timeouts

Existing statement edited to become Req 343:

[Req 343]

The ACS sends a CRes message with a Transaction Status = N to the Notification URL received in the initial AReq message.

This completes the challenge

The 3DS Requestor shall:

[Req 344]

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

5.8.1 3DS Message Handling

The 3DS Server shall:

[Req 315]

~~If the 3DS Method completes within 10 seconds, then the 3DS Requestor will notify the 3DS Server to set the 3DS Method Completion Indicator = Y.~~ Set the 3DS Method Completion Indicator = Y upon notification from the 3DS Requestor. If the 3DS Method does not complete ~~in~~ within 10 seconds, set the 3DS Method Completion Indicator to = N.

Chapter 6 EMV 3-D Secure Security Requirements

6.2.4.2 3DS SDK—CRes

- Checks that ACSCounterAtoS in the decrypted message numerically equals SDKCounterAtoS. If not ceases processing and reports error.

6.2.4.3 ACS—CReq

- Checks that SDKCounterStoA in the decrypted message numerically equals ACSCounterStoA. If not ceases processing and reports error.

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	Conditional Inclusion
Challenge Data Entry	Example: challengeSelectInfo: "challengeDataEntry": "phone"						
Challenge Selection Information	Example: ~~"challengeSelectInfo": {~~ ~~{"mobile": "**** **** 123"},~~ ~~{"email": "s******k**@g***.com"}~~ "challengeSelectInfo": [{"phone": "Mobile **** **** 321"}, {"mail": "Email a*******g**@g***.com"}]						

Data Element/ Field Name	Description	Source	Length/Format/ Values	Device Channel	Message Category	Message Inclusion	Conditional Inclusion
Instalment Payment Data			Example values accepted: • 2 • 02 • 002				
Purchase Amount			Example: purchase amount is USD 123.45: Values accepted: • 12345 • 012345 • 0012345 ~~If the purchase amount is USD 123.45, element will contain the value 12345.~~				
Recurring Frequency			Example values accepted: • 31 • 031 • 0031				

A.5.2 Browser Information—02-BRW Only

Accurate Browser Information is obtained in the AReq message for an ACS to determine the ability to support authentication on a particular Cardholder browser for each transaction. The 3DS Server ~~shall~~ needs to accurately populate the browser information for each transaction. This data ~~may~~ can be obtained by 3DS software provided to the 3DS Requestor or through for example, remote JavaScript calls. ~~but it~~ It ~~shall~~ is ~~be~~ the responsibility of the 3DS Server to ensure that the data is not altered or hard-coded, and that it is unique to each transaction. The specific fields ~~that shall be~~ captured from the Cardholder browser for each transaction are: (No additional edits to section)

A.5.5 Error Code, Error Description, and Error Detail

Table A.4 Error Code, Error Description, and Error Detail

Value	Error Code	Error Description	Error Detail
203		or • Data element is present in a message where the conditional inclusion does not apply.	

A.7.7 Issuer Image

Table A.14 Issuer Image

Data Element/Field Name	Description	Length/Format/Values

Medium Density Image
Field Name: `medium`
 	
High Density Image
Field Name: `high`
 	
Extra High Density Image
Field Name: `extraHigh`
	Examples:
	Images to display:
	`"issuerImage" :{`
	`"medium": "http://acs.com/medium_image.~~svg~~png",`
	`"high": "http://acs.com/high_image.~~svg~~png",`
	`"extraHigh": "http://acs.com/extraHigh_image.~~svg~~png"`
	`}`	

A.7.8 Payment System Image

Table A.15 Payment System Image

Data Element/Field Name	Description	Length/Format/Values

Medium Density Image
Field Name: `medium`
 	
High Density Image
Field Name: `high`
 	
Extra High Density Image
Field Name: `extraHigh`
	Examples:
	Images to display:
	`"psImage" :{`
	`"medium": "http://ds.com/medium_image.~~svg~~png",`
	`"high": "http://ds.com/high_image.~~svg~~png",`
	`"extraHigh": "http://ds.com/extraHigh_image.~~svg~~png"`
	`}`	

Chapter 3 EMV 3-D Secure Authentication Flow Requirements

3.3 Browser-based Requirements

Step 10: The 3DS Server

[Req 118]

Note: ACS implementations that use JavaScript for redirection ~~must~~ will also need to support a fall-back for environments that do not support JavaScript as defined in Req. 324.

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

4.2.5.3 3DS SDK

[Req 171]

- The SDK passes the received data, unchanged, to the ACS in the ~~ACS~~ 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 Requirements

5.1.2 HTTP Header—Content Type

[Req 190]

The HTTP headers shall contain the ~~Content-Type field and have the value:~~ Content-Type Header: application/JSON; and include charset= of UTF~~utf-~~8 for the following messages:

For example, Content-Type: application/JSON; charset = UTF-8

[Req 191]

The Content-Type Header requirements for CReq/CRes are ~~HTTP headers shall contain the Content-Type field and have the value~~:

- For App-based CReq/CRes ~~App-based:~~ the HTTP headers shall contain the Content-Type Header: application/jose; and include a charset of UTF-8.~~charset=utf-8~~

  For example, Content-Type: application/jose; charset = UTF-8

- ~~CRes App-based: application/jose; charset=utf-8~~

- For Browser-based CReq ~~Browser-based~~the HTTP headers shall contain the Content-Type Header: application/x-www-form-urlencoded. ~~charset=utf-8~~

  For example, Content-Type: application/x-www-form-urlencoded

- For Browser-based CRes ~~Browser-based~~the HTTP headers shall contain the ~~the HTTP headers shall contain the~~ Content-Type Header: text/html and include charset of UTF-8~~charset of UTF-8~~.

For example, Content-Type: text/html: charset = UTF-8

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.

5.8.2 Browser Challenge Window Requirements

[Req 324]

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

Chapter 6 EMV 3-D Secure Security Requirements

6.1.8 Link h: Browser—ACS (for 3DS Method)

The link between the Browser and the ACS for the 3DS Method is opened from a hidden iframe loaded by the 3DS Server as part of the checkout page. It is used for the ACS ~~to load JavaScript which~~ gather~~s~~ device information to be returned to the ACS.

6.2.4.1 3DS SDK—CReq

- 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 this version of the specification are:
  - "alg": dir
  - "enc": either:
    - A128CBC-HS256
    - A128GCM

6.2.4.4 ACS—CRes

If the algorithm is A128CBC-HS256, use the full CEK_{A-S} and a fresh 128-bit random data as IV or if the algorithm is A128GCM use the ~~leftmost~~ rightmost 128 bits of CEK_{A-S} with ~~SDKCounterStoA~~ ACSCounterAtoS (padded to the left with 'FF' bytes) as the IV.

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	Conditional Inclusion
3DS Server URL		~~ACS~~					

April 2018 v1

Throughout specification:

Updated all instances of:

- Transaction Status Reason ~~C~~ode to Transaction Status Reason code

Chapter 1 Introduction

Table 1.4: Abbreviations (New)

- AVS—Address Verification Service
- EC—Elliptic Curve
- RSA—Rivest–Shamir–Adleman

Chapter 3 EMV 3-D Secure Authentication Flow Requirements

3.3 Browser-based Requirements

Step 12: The ACS and Browser

[Req 307]

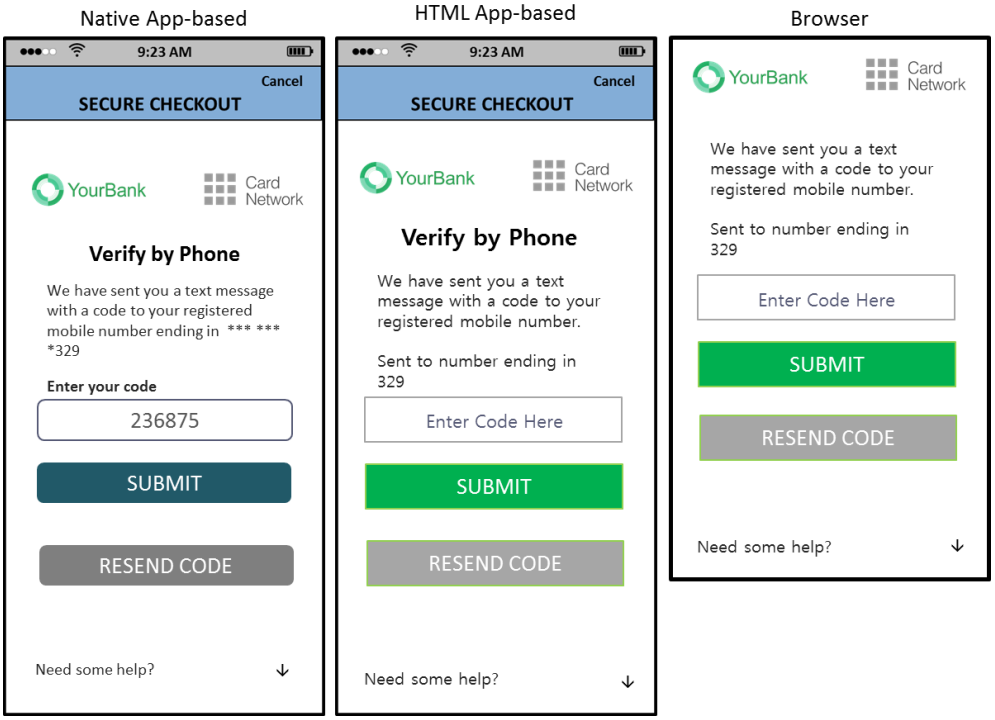
~~Embed all resources in the ACS provided HTML and do not fetch via external URLs.~~ The ACS shall not lead the Cardholder outside of the authentication flow by redirecting to any registration or marketing pages. Any redirection shall be used for authentication purposes only. The ACS shall only load external resources that are needed to improve the cardholder authentication experience and security (e.g., logos).

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

4.1 3-D Secure User Interface Templates

Updated: Figure 4.1: UI Template Examples—All Device Channels

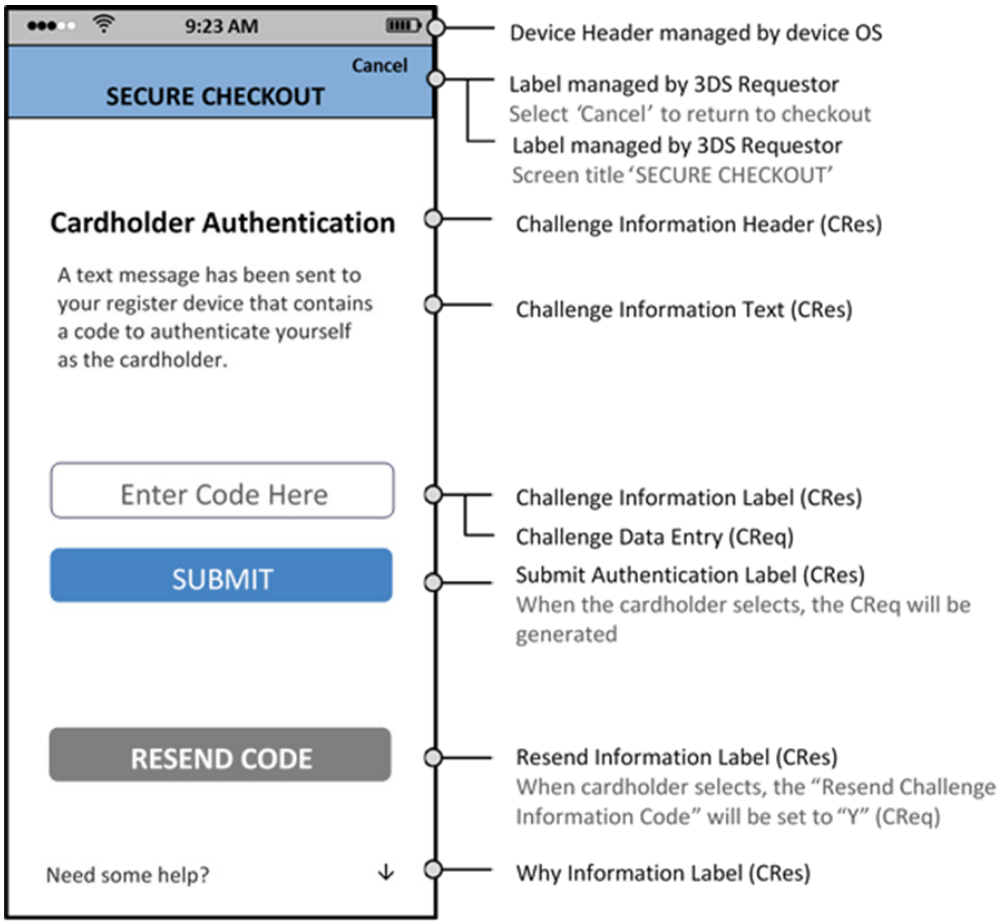
- Updated the ~~**Verify**~~ Button Label to **Submit**.



4.2.2 Native UI Templates

Updated: Figure 4.7: Sample Native UI OTP/Text Template—NPA

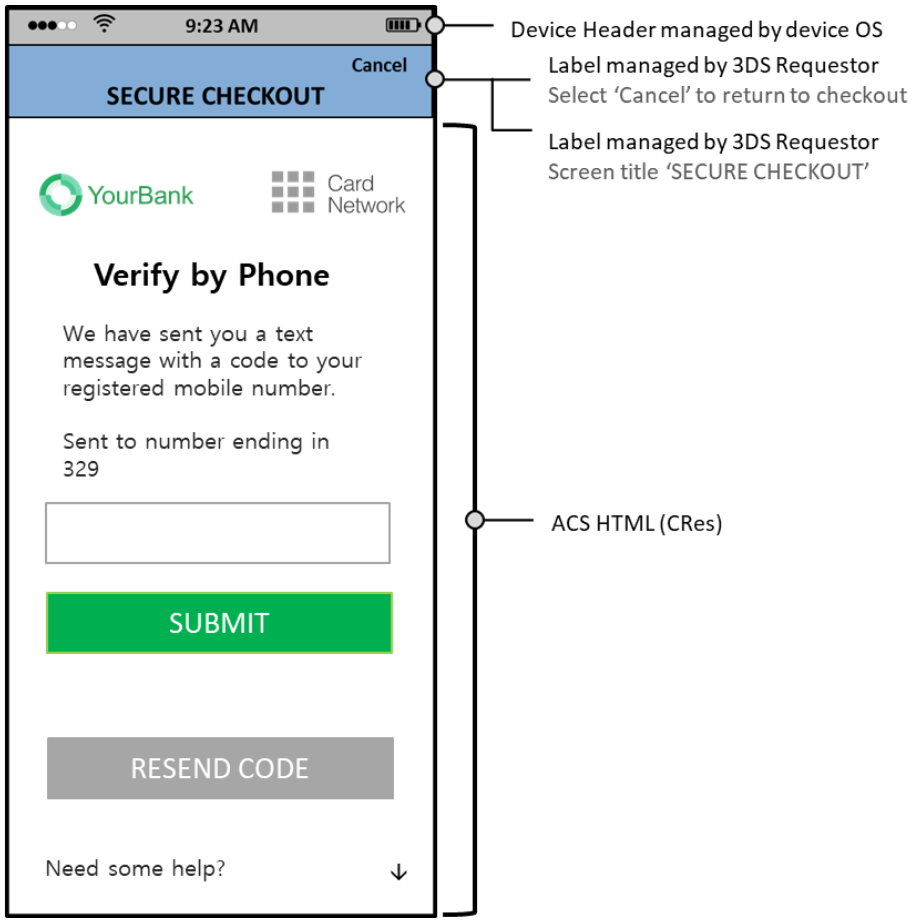
- Updated the ~~**Confirm**~~ Button Label to **Submit**.



4.2.4 HTML UI Templates

Updated: Figure 4.13: Sample HTML UI OTP/Text Template—PA

- Updated the ~~**Verify**~~ Button Label to **Submit**.



Chapter 5 EMV 3-D Secure Message Handling Requirements

5.1.2 HTTP Header—Content Type

[Req 190]

The HTTP headers shall contain the Content-Type field and have the value: Content-Type: application/JSON; charset=utf-8 for the following messages:

- AReq/ARes
- RReq/RRes
- PReq/PRes
- Error Message

[Req 191]

The HTTP headers shall contain the Content-Type field and have the value: ~~Content-Type: application/jose for the CReq/CRes message.~~

- CReq App-based: application/jose; charset=utf-8
- CRes App-based: application/jose; charset=utf-8
- CReq Browser-based: application/x-www-form-urlencoded; charset=utf-8
- CRes Browser-based: text/html; charset=utf-8

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~~s~~ an Error Message (as defined in Section A.5.5) with the applicable Error Component and Error Code = 203.

5.5.2.3 RReq/RRes Message Timeouts

[Req 243]

[Req 245]

Note: No further processing shall occur between the DS and 3DS Server ~~as the SDK has timed out~~.

5.7.1 App-based CReq/CRes Message Handling

Upon receiving the CRes message from the ACS, the 3DS SDK displays the UI to the Cardholder for authentication and communicates the result back to the ACS in the ~~CRes~~ CReq message.

5.8.2 Browser Challenge Window Requirements

[Req 269]

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

5.9.9 3DS Server RReq Message Error Handling

- For a message that cannot be recognised, the 3DS Server:
    - Returns to the DS an Error Message (as defined in Section A.5.5) with Error Component = ~~A~~ S and Error Code = 101.

Chapter 6 EMV 3-D Secure Security Requirements

6.2.3.2 ACS Secure Channel Setup

- Completes the Diffie-Hellman key exchange process as a local mechanism according to JWA (RFC 7518) in Direct Key Agreement mode using curve P-256, dT and QC to produce a pair of CEKs (one for each direction) which ~~are~~is identified by 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, but do not use this as the algorithmID for the KDF. The parameter values supported in this version of the specification are:

Additional 6.2.3.2 update:

- ~~{"acsEphemPubKey": "QT", "sdkEphemPubKey":” QC”, "ACSURL":"ACS URL":"https://mybank.com/acs"}~~

Was updated to:

- {"acsEphemPubKey": QT, "sdkEphemPubKey": QC, "acsURL":"https://mybank.com/acs"}

6.2.3.3 3DS SDK Secure Channel Setup

- Completes the Diffie–Hellman key exchange process according to JWA (RFC 7518) in Direct Key Agreement mode using curve P-256, dC and QT to produce a pair of CEKs (one for each direction), which are 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, but do not use this as the algorithmID for the KDF. The parameter values supported in this version of the specification are:

6.2.4.4 ACS—CRes

If the algorithm is A128CBC-HS256 use the full CEKA-S and a fresh 128-bit random data as IV or if the algorithm is A128GCM use the leftmost 128 bits of CEKA-S with ~~SDKCounterStoA~~ SDKCounterAtoS (padded to the left with ‘FF’ bytes) as the IV.

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	Conditional Inclusion
ACS Signed Content	Contains the JWS object (represented as a string) created by the ACS for the ARes message.		JSON Data Type: ~~Object~~String The body of JWS object (represented as a string) will contain the following data elements as defined in Table A.1:				
Broadcast Information			Length: Variable, maximum 4096 characters				
Browser Screen Height			Length: Variable, 1–6 characters; Numeric JSON Data Type: String				
Browser Screen Width			Length: Variable, 1–6 characters; Numeric JSON Data Type: String				

Data Element/ Field Name	Description	Source	Length/Format/ Values	Device Channel	Message Category	Message Inclusion	Conditional Inclusion
Cardholder Information Text	Text provided by the ACS/Issuer to Cardholder during a Frictionless transaction ~~that was not authenticated by the ACS~~.		Length: Variable, maximum 128 characters JSON Data Type: String If field is populated this information ~~shall~~ can optionally be displayed to the cardholder by the merchant.				

Data Element/ Field Name	Description	Source	Length/Format/ Values	Device Channel	Message Category	Message Inclusion	Conditional Inclusion
Challenge Cancelation Indicator			• 02 = Reserved for future EMVCo use (values invalid until defined by EMVCo)~~3DS Requestor cancelled Authentication.~~ • 03 = Reserved for future EMVCo use (values invalid until defined by EMVCo)~~Transaction Abandoned~~ • 08 = Transaction Timed Out at SDK • ~~08~~09-79 = Reserved for future EMVCo use (values invalid until defined by EMVCo)				
Device Information			JSON Data Type: ~~Object~~String Base64url encoded JSON Object (represented as a string)				
Message Extension			~~bytes~~characters				

Data Element/ Field Name	Description	Source	Length/Format/ Values	Device Channel	Message Category	Message Inclusion	Conditional Inclusion
Results Message Status			• 03 = ARes (Transaction Status = C) ~~challenge data~~ not delivered to the 3DS Requestor due to technical error				
SDK Encrypted Data	JWE Object (represented as a string) as defined in Section 6.2.2.1 containing data encrypted by the SDK for the DS to decrypt.		JSON Data Type: ~~Object~~String				
SDK Maximum Timeout					02-NPA		

A.6 Message Extension Data

A maximum of 10 extensions (objects) are supported within the Message Extension data element, totalling a maximum of 81920 ~~bytes~~characters.

```
"messageExtension":
[
      {
            "name": "extension~~Field~~1",
           "id": "ID1",
           "criticalityIndicator": true,
          "data":{
                "valueOne":"value"
           }
      },
      {
            "name": "extension~~Field~~2",
           "id": "ID2",
           "criticalityIndicator": true,
          "data":{
                "valueOne":"value1",
"valueTwo":"value2"
           }
      },
      {
            "name": "sharedData",
           "id": "ID3",
           "criticalityIndicator": false,
          "data":{

"value3":"IkpTT05EYXRhIjogew0KImRhdGExIjogInNvbWUgZGF0YSIsDQoi
ZGF0YTIiOiAic29tZSBvdGhlciBkYXRhIg0KfQ=="
           }
      }
]
```

A.7.3 3DS Requestor Authentication Information

Table A.10 3DS Requestor Authentication Information

Data Element/Field Name	Description	Length/Format/Value
3DS Requestor Authentication Data	For example, ~~for method:~~if the 3DS Requestor Authentication Method is: • 03, then this element can carry information about the provider of the federated ID and related information. • 06, then this element can carry the FIDO attestation data (including the signature). • ~~02 – field can carry generic 3DS Requestor authentication information~~ • ~~03 – data element can carry information about the provider of the federated ID and related information~~ • ~~0406 – data element can carry the FIDO attestation data (including the signature)~~	2048 ~~bytes~~characters

A.7.4 3DS Requestor Prior Transaction Authentication Information

Table A.11: 3DS Requestor Prior Transaction Authentication Information

Data Element/Field Name	Description	Length/Format/Value
3DS Requestor Prior Transaction Authentication Data		Length: maximum 2048 ~~bytes~~characters

A.7.6 Device Rendering Options Supported

JSON Object Example:

```
{
"deviceRenderOptions":{ "sdkInterface":"03", "sdkUiType":[”01", “02”, “03”, “04”, “05”] }
```

}

Annex B Message Format

B.4 CRes Message Data Elements

Table B.4 CRes Data Elements

Data Element	Field Name
ACS HTML Refresh	acsHTMLRefresh

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