



EMV® Specification Bulletin No. 204 v5

May 2019

EMV® 3-D Secure Updates, Clarifications & Errata

This Specification Bulletin No. 204v5 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.

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

May 2019



Contents

EMV® 3-D Secure Updates, Clarifications & Errata	1
Applicability	1
Related Documents.....	1
Effective Date	1
May 2019 v5	9
Chapter 1 Introduction	9
1.5 Definitions	9
Table 1.3 Definitions.....	9
Chapter 4 EMV 3-D Secure User Interface Templates, Requirements, and Guidelines	9
4.1 3-D Secure User Interface Templates	9
[Req 358]	9
[Req 342]	9
[Req 359]	9
4.2 App-based User Interface Overview	10
4.2.1 Processing Screen Requirements	10
4.2.1.1 3DS SDK/3DS Requestor App.....	11
[Req 143]	11
[Req 145]	11
[Req 388]	11
[Req 360]	11
[Req 361]	11
[Req 389]	11
4.2.2.1 3DS SDK/ACS	14
[Req 362]	14
[Req 369]	14
[Req 387]	14
4.2.4.1 3DS SDK	14
[Req 153]	14
4.2.5.1 3DS SDK/ACS	14
[Req 373]	14
[Req 374]	14
4.2.7.3 3DS SDK	14
[Req 171]	14
4.3.1 Processing Screen Requirements	15
[Req 177]	15
[Req 379]	15
4.3 Browser-based User Interface Overview	15
4.3.2.1 ACS	15
[Req 380]	15



Chapter 5 EMV 3-D Secure Message Handling	15
5.1.3 Base64/Base64url Encoding.....	15
[Req 193]	15
5.1.6 Message Content Validation	15
[Req 309]	15
Chapter 6 EMV 3-D Secure Security Requirements	16
6.2.2.1 3DS SDK Encryption.....	16
6.2.2.2 DS Decryption	17
6.2.3.2 ACS Secure Channel Setup	18
6.2.3.3 3DS SDK Secure Channel Setup	18
6.2.4.1 3DS SDK—CReq	19
6.2.4.2 3DS SDK—CRes	19
6.2.4.3 ACS—CReq	19
6.2.4.4 ACS—CRes	19
Annex A 3-D Secure Data Elements	20
A.4 EMV 3-D Secure Data Elements	20
Table A.1 EMV 3-D Secure Data Elements	20
A.5.7 Card Range Data.....	22
Table A.6 Card Range Data	22
A.7.3 3DS Requestor Information	23
A.7.4 3DS Requestor Prior Transaction Authentication Information	23
A.7.5 ACS Rendering Type.....	23
Table A.12: ACS Rendering Type	23
A.8 UI Data Elements.....	23
Table A.18 UI Data Elements	23
October 2018 v4.....	27
Chapter 1 Introduction	27
1.9 Terminology and Conventions	27
Chapter 3 EMV 3-D Secure Authentication Flow Requirements	27
3.1 App-based Requirements	27
Step 7 The ACS.....	27
[Req 386]	27
Chapter 4 EMV 3-D Secure User Interface Templates, Requirements, and Guidelines	27
4.1 3-D Secure User Interface Templates	27
Figure 4.1: UI Template Zones (NEW).....	28
[Req 358]	28
[Req 359]	28
Figure 4.2 UI Template Examples—All Device Channels (NEW).....	29
4.2.1.1 3DS SDK/3DS Requestor App.....	29
[Req 143]	29



[Req 360]	29
[Req 151]	29
[Req 361]	29
4.2.2 Native UI Templates-Display Requirements	30
4.2.2.1 3DS SDK/ACS	30
[Req 362]	30
[Req 363]	30
[Req 364]	30
[Req 365]	30
[Req 366]	30
[Req 367]	30
[Req 368]	30
[Req 369]	30
[Req 370]	31
4.2.3 Native UI Templates	31
4.2.4 Native UI Message Exchange Requirements	31
4.2.5 HTML UI Templates-Display Requirements	31
4.2.5.1 3DS SDK/ACS	31
[Req 371]	31
[Req 372]	31
[Req 373]	31
[Req 375]	31
[Req 376]	32
[Req 377]	32
[Req 378]	32
4.2.6 HTML UI Templates	32
4.3 Browser-based User Interface Overview	32
4.3.1.1 3DS Requestor Website	32
[Req 172]	32
[Req 173]	32
[Req 174]	32
4.3.1.2 ACS	32
[Req 379]	32
[Req 179]	32
[Req 180]	32
[Req 182]	33
4.3.2 Browser Display Requirements	33
4.3.2.1 ACS	33
[Req 380]	33
[Req 381]	33
[Req 382]	33

[Req 383]	33
[Req 384]	33
4.3.3 Browser UI Templates	33
Figure 4.19: App-based HTML and Browser UI Comparison (NEW).....	34
Figure 4.22: Sample Browser with Lightbox UI—PA (NEW).....	35
Figure 4.23 Sample Browser with Inline UI—PA (NEW).....	36
Chapter 5 EMV 3-D Secure Message Handling Requirements	37
5.1.6 Message Content Validation	37
[Req 209]	37
5.5.1 Transaction Timeouts	37
[Req 221]	37
[Req 224]	37
5.6 PReq/PRes Message Handling Requirements.....	38
[Req 250]	38
[Req 304]	38
[Req 385]	38
Annex A 3-D Secure Data Elements.....	39
A.4 EMV 3-D Secure Data Elements	39
Table A.1 EMV 3-D Secure Data Elements	39
A.5.3 3DS Method Data	42
3DS Method Data Examples	42
A.5.5 Error Code, Error Description, and Error Detail.....	42
Table A.4 Error Code, Error Description, and Error Detail	42
A.7.1 Cardholder Account Information.....	43
Table A.8 Cardholder Account Information	43
A.7.2 Merchant Risk Indicator	44
Table A.9 Merchant Risk Indicator	44
A.7.3 3DS Requestor Authentication Information	44
Table A.10 3DS Requestor Authentication Information	44
A.7.4 3DS Requestor Prior Transaction Authentication Information	44
Table A.11: 3DS Requestor Prior Transaction Authentication Information.....	44
A.7.7 Challenge Data Entry	45
Table A.14: Challenge Data Entry.....	45
A.8 UI Data Elements.....	46
Table A.1: UI Data Elements.....	46
Annex B Message Format	48
B.4 CRes Message Data Elements	48
Table B.4 CRes Data Elements	48
August 2018 v3	49
Chapter 3 EMV 3-D Secure Authentication Flow Requirements	49
3.3 Browser-based Requirements	49



Step 15 The ACS	49
[Req 123]	49
Chapter 4 EMV 3-D Secure UI Templates, Requirements, and Guidelines	49
4.2 App-based User Interface Overview	49
4.2.5.3 3DS SDK	49
[Req 171]	49
Chapter 5 EMV 3-D Secure Message Handling Requirements	49
5.5.1 Transaction Timeouts	49
[Req343]	49
[Req 344]	49
5.8.1 3DS Message Handling	50
[Req 315]	50
Chapter 6 EMV 3-D Secure Security Requirements	50
6.2.4.2 3DS SDK—CRes	50
6.2.4.3 ACS—CReq	50
Annex A 3-D Secure Data Elements	51
A.4 EMV 3-D Secure Data Elements	51
Table A.1 EMV 3-D Secure Data Elements	51
A.5.2 Browser Information—02-BRW Only	52
A.5.5 Error Code, Error Description, and Error Detail	53
Table A.4 Error Code, Error Description, and Error Detail	53
A.7.7 Issuer Image	53
Table A.14 Issuer Image	53
A.7.8 Payment System Image	54
Table A.15 Payment System Image	54
June 2018 v2	55
Chapter 3 EMV 3-D Secure Authentication Flow Requirements	55
3.3 Browser-based Requirements	55
Step 10 The 3DS Server	55
[Req 118]	55
Chapter 4 EMV 3-D Secure UI Templates, Requirements, and Guidelines	55
4.2.5.3 3DS SDK	55
[Req 171]	55
Chapter 5 EMV 3-D Secure Message Handling Requirements	55
5.1.2 HTTP Header—Content Type	55
[Req 190]	55
[Req 191]	55
5.1.3 Base64/Base64url Encoding	56
[Req 193]	56
5.8.2 Browser Challenge Window Requirements	56



[Req 324]	56
Chapter 6 EMV 3-D Secure Security Requirements	56
6.1.8 Link h: Browser—ACS (for 3DS Method)	56
6.2.4.1 3DS SDK—CReq	56
6.2.4.4 ACS—CRes	56
Annex A 3-D Secure Data Elements	57
A.4 EMV 3-D Secure Data Elements	57
Table A.1 EMV 3-D Secure Data Elements	57
April 2018 v1	58
Throughout specification:	58
Chapter 1 Introduction	58
Table 1.4: Abbreviations (New)	58
Chapter 3 EMV 3-D Secure Authentication Flow Requirements	58
3.3 Browser-based Requirements	58
Step 12 The ACS and Browser	58
[Req 307]	58
Chapter 4 EMV 3-D Secure UI Templates, Requirements, and Guidelines	58
4.1 3-D Secure User Interface Templates	58
Updated: Figure 4.1: UI Template Examples—All Device Channels	59
4.2.2 Native UI Templates	60
Updated: Figure 4.7: Sample Native UI OTP/Text Template—NPA	60
4.2.4 HTML UI Templates	61
Updated: Figure 4.13: Sample HTML UI OTP/Text Template—PA	61
Chapter 5 EMV 3-D Secure Message Handling Requirements	62
5.1.2 HTTP Header—Content Type	62
[Req 190]	62
[Req 191]	62
5.1.6 Message Content Validation	62
[Req 309]	62
5.5.2.3 RReq/RRes Message Timeouts	62
[Req 243]	62
[Req 245]	62
5.7.1 App-based CReq/CRes Message Handling	62
5.8.2 Browser Challenge Window Requirements	62
[Req 269]	62
5.9.9 3DS Server RReq Message Error Handling	63
Chapter 6 EMV 3-D Secure Security Requirements	63
6.2.3.2 ACS Secure Channel Setup	63
6.2.3.3 3DS SDK Secure Channel Setup	63
6.2.4.4 ACS—CRes	63



Annex A 3-D Secure Data Elements.....	65
A.4 EMV 3-D Secure Data Elements	65
Table A.1 EMV 3-D Secure Data Elements	65
A.6 Message Extension Data.....	69
A.7.3 3DS Requestor Authentication Information	70
Table A.10 3DS Requestor Authentication Information	70
A.7.4 3DS Requestor Prior Transaction Authentication Information	70
Table A.11: 3DS Requestor Prior Transaction Authentication Information.....	70
A.7.6 Device Rendering Options Supported	70
JSON Object Example:	70
Annex B Message Format	71
B.4 CRes Message Data Elements	71
Table B.4 CRes Data Elements	71

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.

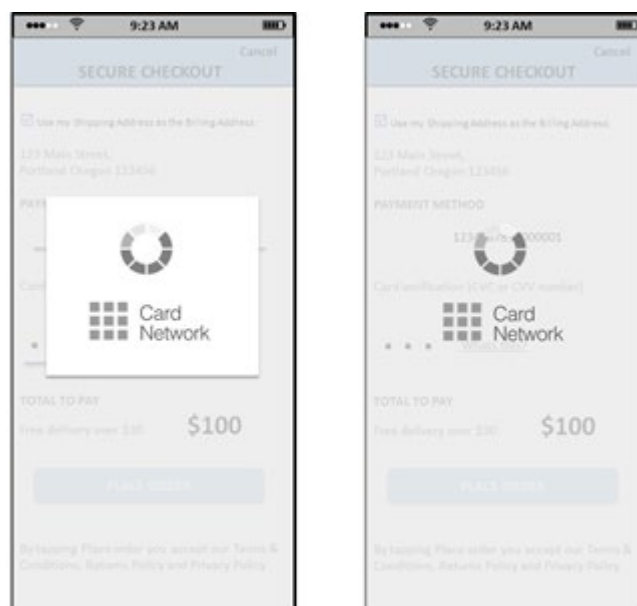
4.2.1 Processing Screen Requirements

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]

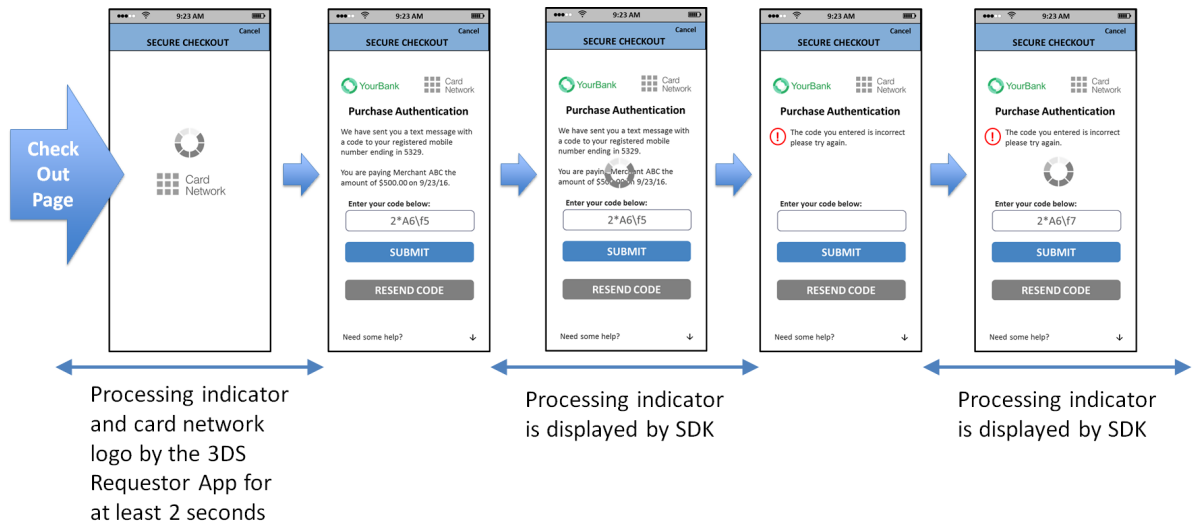
~~Display the Cancel action in the top right corner of the Header zone.~~

[Req 389]

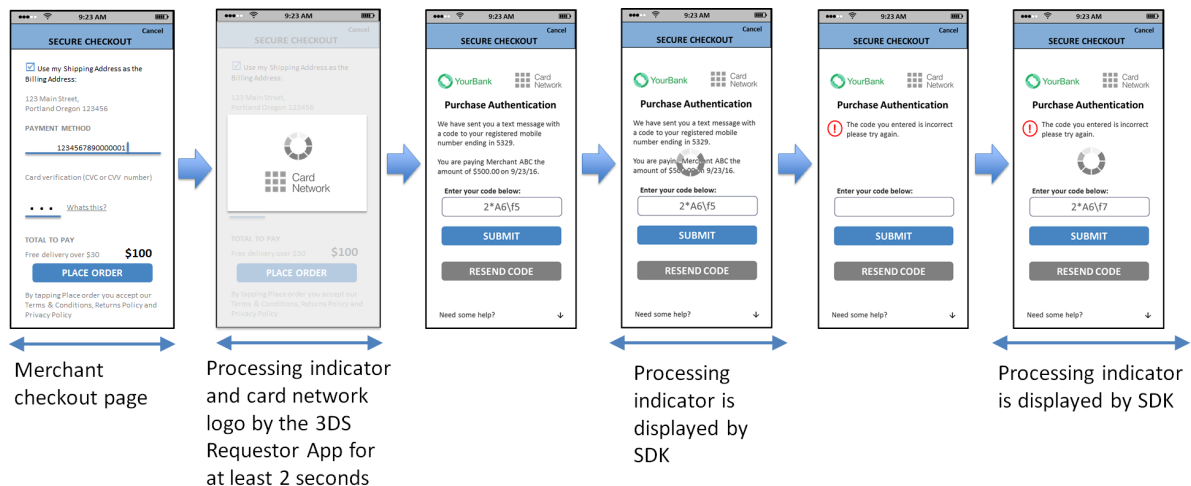
Ensure that the Cancel action is not actionable on the Processing screen.

New Graphic for Figure 4.5

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

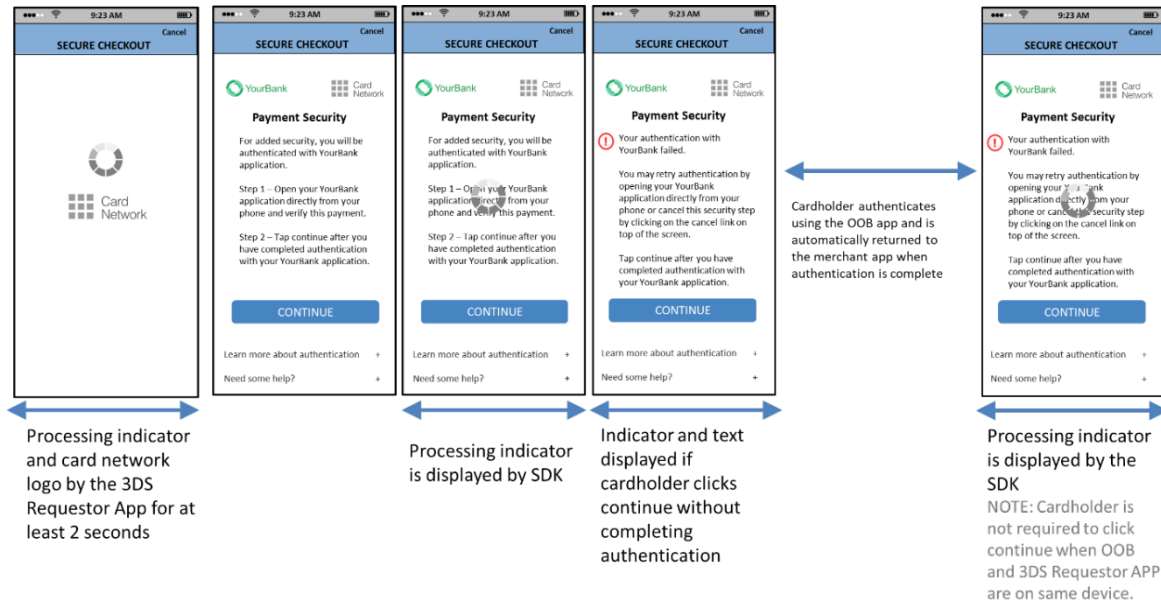


(Updated) Figure 4.5: Sample OTP/Text Template—App-based Processing Flow

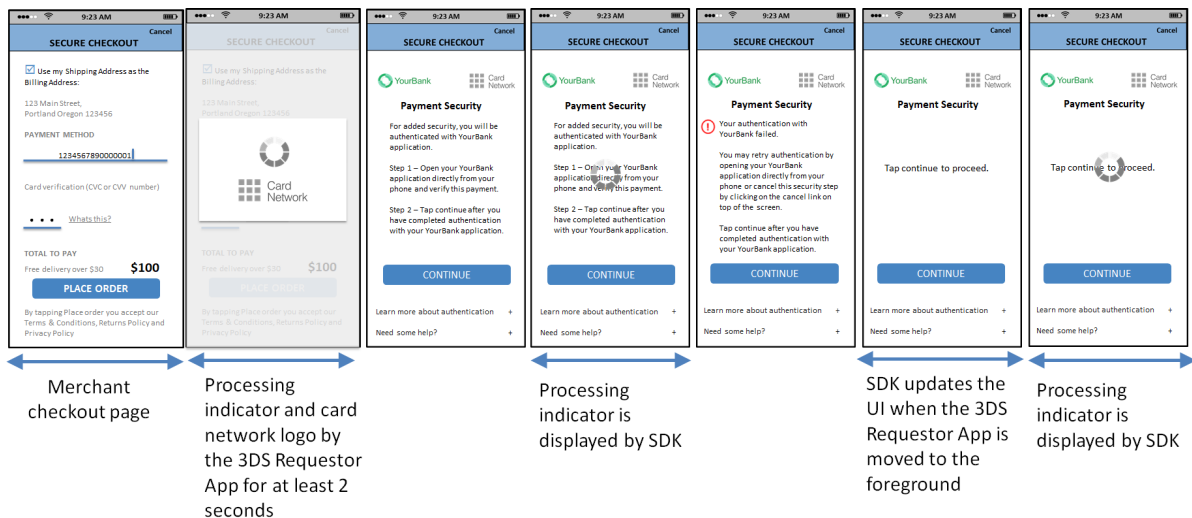


New Graphic for Figure 4.6

(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 AResAReq 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 the UIform elements in the applicable zones as defined in Table A.18 and depictedoutlined 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. Click here to view the “clean” version of these edits.

6.2.2.1 3DS SDK Encryption

The 3DS SDK:

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

— "alg": ECDH-ES

— "apv": DirectoryServerID

— "epk": P_{DS} , in JSON Web Key (JWK) format

{ "kty": "EC"

"crv": "P-256" }

— All other parameters: not present

— Keydatalen = 256

— AlgorithmID = empty string (length = 0x00000000)

— PartyUInfo = empty string (length = 0x00000000)

— PartyVInfo = directoryServerID (length || ascii string)

— SuppPubInfo = Keydatalen (0x00000100)

— SuppPrivInfo = empty octet sequence

— CEK: "kty": oct-256 bits

- Generates 128-bit random data as IV (included in the JWE)

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

— "alg": 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~~
 - ~~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 CEKs (one for each direction) which are 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 and assume the algorithmID to be null for the KDF⁸. (Footnote 8 also deleted: ⁸Note this is using RFC 7518 only for key derivation.). The Concat KDF parameter values supported infor this version of the specification are:
 - "alg":ECDH-ES
 - "apv": SDK Reference Number
 - "epk": Q_C (received in the AReq message as sdkEphemKey)
 - {"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:
- Generates a digital signature of the full JSON object according to JWS (RFC 7515) using JWS Compact Serialization. The parameter values supported infor 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 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 and assume the algorithmID to be null for the KDF¹⁰. (Footnote 10 also deleted: ¹⁰Note this is using RFC 7518 only for key derivation.). The Concat KDF parameter values supported infor this version of the specification are:



- ~~"alg":ECDH-ES~~
- ~~"apv": SDK Reference Number~~
- ~~"epk": Q_r (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)
 - PartyUIInfo = empty string (length = 0x00000000)
 - PartyVInfo = sdkReferenceNumber (length || ascii string)
 - SuppPubInfo = Keydatalen (0x00000100)
 - SuppPrivInfo = empty octet sequence
 - CEK: ~~"kty":oct-256 bits extracted~~ allocated as:

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

6.2.4.1 3DS SDK—CReq

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

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

6.2.4.2 3DS SDK—CRes

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

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

6.2.4.3 ACS—CReq

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

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

6.2.4.4 ACS—CRes

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

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

6.2.2.1 3DS SDK Encryption

The 3DS SDK:

- Identifies the DS public key P_{DS} , associated attributes, and encryption function (relating to the BIN and optionally other information) from information provided by the 3DS Requestor Environment. If the public key cannot be identified, ceases processing and report error.
- Creates a JSON Object of Device Information.
- If P_{DS} is an RSA key:
 - Encrypts the JSON object according to JWE (RFC 7516) using JWE Compact Serialization. The parameter values for this version of the specification and to be included in the JWE protected header are:
 - "alg": "RSA-OAEP-256"
 - "enc": "A128CBC-HS256 or A128GCM"
 - All other parameters not present
- Else if P_{DS} is an EC key:
 - Encrypts the JSON object as follows:
 - Generates a fresh ephemeral key pair (Q_{SDK} , d_{SDK}) as described in Annex C.
 - 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 for this version of the specification are:
 - Keydatalen = 256
 - AlgorithmID = empty string (length = 0x00000000)
 - PartyUInfo = empty string (length = 0x00000000)
 - PartyVInfo = directoryServerID (length || ascii string)
 - SuppPubInfo = Keydatalen (0x00000100)
 - SuppPrivInfo = empty octet sequence
 - 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 for this version of the specification and to be included in the JWE protected header are:
 - "alg": "ECDH-ES"
 - "epk": Q_{SDK} ,
{"kty": "EC",
"crv": "P-256"
"x": x coordinate of Q_{SDK}
"y": y coordinate of Q_{SDK} }
 - "enc": either "A128CBC-HS256" or "A128GCM"
 - All other parameters: not present

- If the algorithm is A128CBC-HS256 use the full CEK or if the algorithm is A128GCM use the leftmost 128 bits of the CEK.
- Deletes the ephemeral key pair (Q_{SDK} , d_{SDK})
- Makes the resulting JWE available to the 3DS Server as SDK Encrypted Data.

6.2.2.2 DS Decryption

The DS:

- If the protected header of the JWE in the SDK Encrypted Data field indicates that 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.
- Else, if the protected header of the JWE in the SDK Encrypted Data field indicates that 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 for this version of the specification are:
 - Keydatalen = 256
 - AlgorithmID = empty string (length = 0x00000000)
 - PartyUInfo = empty string (length = 0x00000000)
 - PartyVInfo = directoryServerID (length || ascii string)
 - SuppPubInfo = Keydatalen (0x00000100)
 - SuppPrivInfo = empty octet sequence
 - 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. If decryption fails, ceases processing and reports error.
- Insert the result into Device Information for the AReq message to the ACS.

6.2.3 Function J: 3DS SDK—ACS Secure Channel Set-Up

Using data transferred in the AReq/ARes messages the 3DS SDK and ACS execute a Diffie-Hellman key exchange protocol to establish keys for a secure channel that will later be used to protect the CReq/CRes messages in the Challenge Flow if the transaction is challenged.

6.2.3.1 3DS SDK Preparation for Secure Channel

The 3DS SDK:

- Generates a fresh ephemeral key pair (Q_C , d_C) as described in Annex C and provides Q_C as a JWK for inclusion in the AReq message as `sdkEphemPubKey`.

6.2.3.2 ACS Secure Channel Setup

If the ACS determines that a challenge is required to secure the direct link between the 3DS SDK and ACS for the CReq/CRes messages, it completes the security function during the AReq/ARes exchange as follows:

As a prerequisite, the ACS has a key pair (Pb_{ACS} , Pv_{ACS}) with a certificate Cert (Pb_{ACS}). This certificate is an X.509 certificate signed by a DS CA whose public key is known to the 3DS SDK.

The ACS receives Q_C from the 3DS SDK in the AReq message (via the 3DS Server and the DS).

The ACS signs its own ephemeral public key Q_T together with Q_C received from the 3DS SDK and the ACS URL (to be used by the 3DS SDK for the CReq message). The resulting signature is sent back to the 3DS SDK together with Cert (Pb_{ACS}) for the ACS public key.

The ACS:

- Generates a fresh ephemeral key pair (Q_T , d_T) as described in Annex C.
- Checks that Q_C is a point on the curve P-256.
- 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 256-bit CEK which is identified to the ACS Transaction ID. The Concat KDF parameter values for this version of the specification are:
 - 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: 256 bits allocated as:
 - CEK_{A-S}: 256 bits
 - CEK_{S-A}: 256 bits

Note: Key separation is good practice for opposite directions of the 3DS SDK to ACS link. In this version of the specification CEK_{A-S} and CEK_{S-A} are extracted with the same value. Thus, for A128CBC-HS256 the same 256-bit key will be used in both directions. For A128GCM the key is split as two 128 bit components, resulting in separate keys for each direction.

- Creates a JSON object of the following data as the JWS payload to be signed:
 - {"acsEphemPubKey": Q_T , "sdkEphemPubKey": Q_C , "acsURL": "https://mybank.com/acs"}
- Generates a digital signature of the full JSON object according to JWS (RFC 7515) using JWS Compact Serialization. The parameter values for this version of the specification and to be included in the JWS header are:

- "alg": PS256⁸ or ES256
- "x5c": X.5C v3: Cert(Pb_{ACS}) and chaining certificates if present
- All other parameters: not present
- Includes the resulting JWS in the ARes message as ACS Signed Content
- Deletes the ephemeral key pair (Q_T , d_T)
- Zeros the channel counters ACSCounterAtoS (:oct – 8 bits) and ACSCounterStoA (:oct – 8 bits)

6.2.3.3 3DS SDK Secure Channel Setup

The 3DS SDK receives the necessary data elements from the ARes message (extracted by the 3DS Server), including Q_T , ACS Signature, ACS Public Key Certificate, and ACS_URL.

The 3DS SDK:

- Using the CA public key of the DS CA identified from information provided by the 3DS Server, validates the JWS from the ACS according to JWS (RFC7515) using either PS256 or ES256 as indicated by the "alg" parameter in the header. 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 256-bit CEK, which is identified to the ACS Transaction ID received in the ARes message. The Concat KDF parameter values supported for this version of the specification are:
 - 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: 256 bits allocated as:
 - CEK_{A-S}: 256 bits
 - CEK_{S-A}: 256 bits
- Deletes the ephemeral key pair (Q_C , d_C)
- Zeros the channel counters SDKCounterAtoS (:oct – 8 bits) and SDKCounterStoA (:oct – 8 bits)

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.

⁸ PS256 (RSA-PSS) is specified in preference to RS256 (RSASSA-PKCS1-v1_5) following the recommendation in RFC 3447 (2003).

6.2.4 Function K: 3DS SDK—ACS (CReq, CRes)

6.2.4.1 3DS SDK—CReq

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

- Creates a JSON object of the data elements identified in the CReq message defined in Table B.3.
- 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 for this version of the specification and to be included in the JWE protected header are:
 - "alg": dir
 - "enc": either: A128CBC-HS256 or A128GCM
 - "kid": ACS Transaction ID
 - All other parameters: not present

If the algorithm is A128CBC-HS256 use the full CEK_{S-A} and a fresh 128-bit random data as IV or if the algorithm is A128GCM use the leftmost 128 bits of CEK_{S-A} with SDKCounterStoA (padded to the left with '00' bytes) as the IV.

- Sends the resulting JWE to the ACS as the protected CReq message.
- Increments SDKCounterStoA. If SDKCounterStoA = zero, ceases processing and reports error.

6.2.4.2 3DS SDK—CRes

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

- Decrypts the message according to JWE (RFC 7516) using either A128CBC-HS256 or A128GCM and the CEK_{A-S} obtained in Section 6.2.3.3 as identified by the "enc" and "kid" parameters in the protected header. If decryption fails, ceases processing and reports error.
- Checks that ACSCounterAtoS in the decrypted message numerically equals SDKCounterAtoS. If not ceases processing and reports error.
- Increments SDKCounterAtoS. If SDKCounterAtoS = zero, 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.
- Checks that SDKCounterStoA in the decrypted message numerically equals ACSCounterStoA. If not ceases processing and reports error.
- Increments ACSCounterStoA. If ACSCounterStoA = zero, ceases processing and reports error.

6.2.4.4 ACS—CRes

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

- Creates a JSON object of the data elements identified in the CRes message defined in Table B.4.
- 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 for this version of the specification and to be included in the JWE protected header are:
 - "alg": dir
 - "enc": either A128CBC-HS256 or A128GCM
 - "kid": ACS Transaction ID
 - All other parameters: not present

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 rightmost 128 bits of CEK_{A-S} with ACSCounterAtoS (padded to the left with 'FF' bytes) as the IV.

- Sends the resulting JWE to the 3DS SDK as the protected CRes message.
- Increments ACSCounterAtoS. If ACSCounterAtoS = zero, 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	Condition 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.
Cardholder Email Address							Required (if available) unless market or regional mandate restricts sending this information.



Data Element/ Field Name	Description	Source	Length/Format/ Values	Device Channel	Message Category	Message Inclusion	Condition Inclusion
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.						
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.



Data Element/ Field Name	Description	Source	Length/Format/ Values	Device Channel	Message Category	Message Inclusion	Condition Inclusion
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.
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.					
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



Data Element	Field Name	Zone	Top-down Display Order	ACS UI Type			
				OTP	Single Select	Multi Select	OOB
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
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

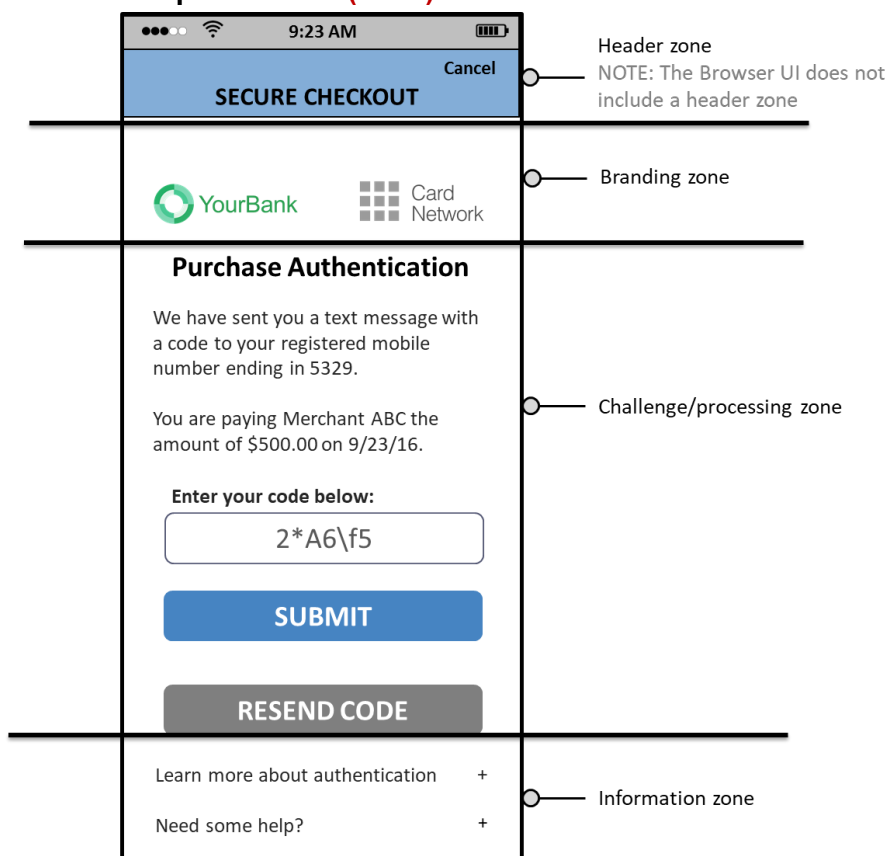
4.1 3-D Secure User Interface Templates

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 Ch 4 Figures were renumbered.

The SDK shall:

[Req 358]

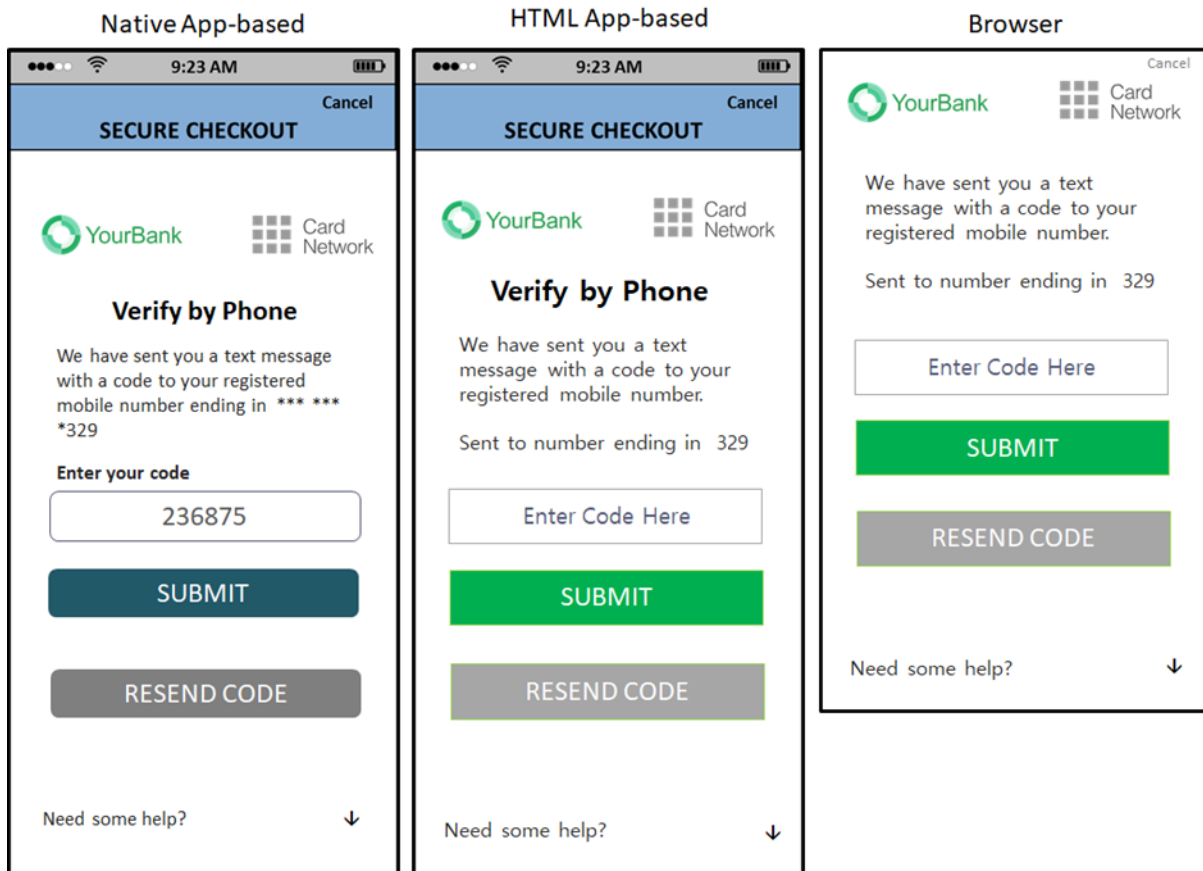
Display UI data elements provided by the ACS within the applicable zones as defined in Table A.18 and depicted in Figure 4.1. The expected format is depicted in sections 4.2.3 and 4.2.6.

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 Templates-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** seconds.

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]

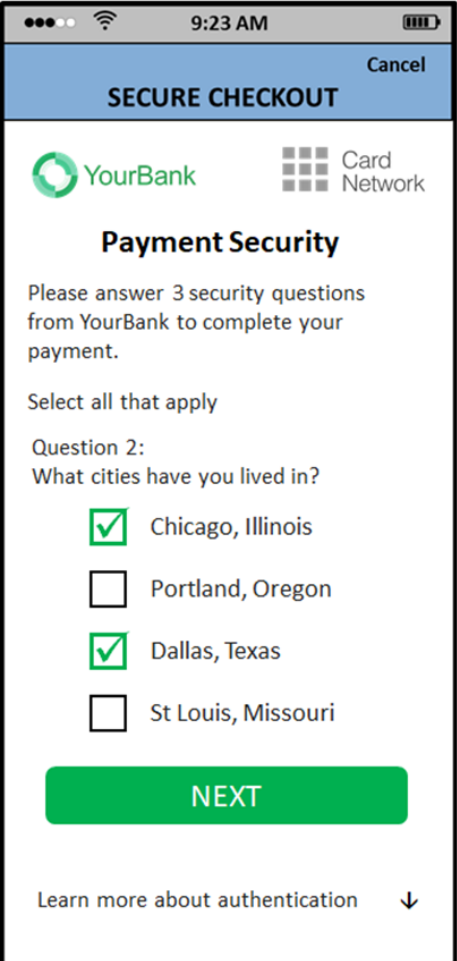
Why Information Text for display in the Information zone when the user selects the Why Information Label.

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.

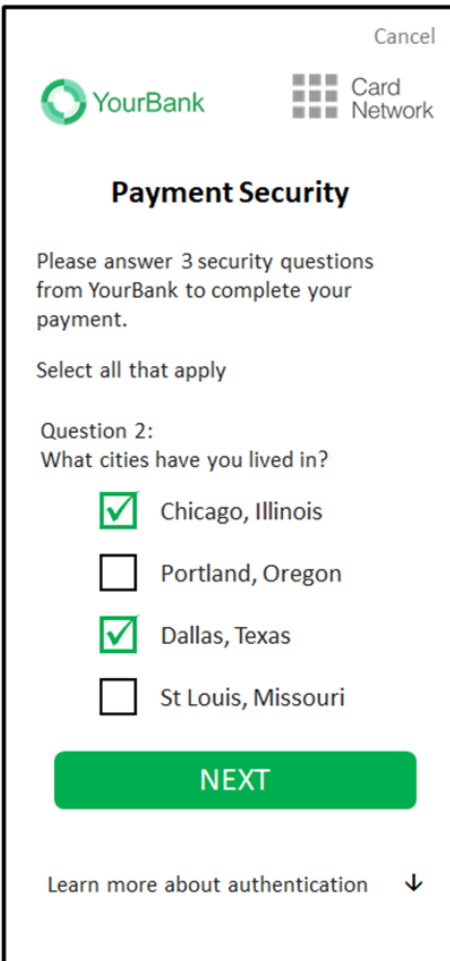
Figure 4.19: App-based HTML and Browser UI Comparison (NEW)

HTML App-based Template



The mobile app UI features a status bar at the top with signal, Wi-Fi, and battery icons, and the time 9:23 AM. Below is a blue header bar with 'Cancel' and 'SECURE CHECKOUT'. The main content area includes the 'YourBank' logo, 'Card Network' logo, and the title 'Payment Security'. A message asks the user to answer 3 security questions. Below this, it says 'Select all that apply' and 'Question 2: What cities have you lived in?'. There are four options: 'Chicago, Illinois' (checked), 'Portland, Oregon' (unchecked), 'Dallas, Texas' (checked), and 'St Louis, Missouri' (unchecked). A green 'NEXT' button is at the bottom, followed by a link 'Learn more about authentication' with a downward arrow.

Browser Template



The browser UI has a 'Cancel' link at the top right. It features the 'YourBank' logo and 'Card Network' logo. The title 'Payment Security' is centered. A message asks the user to answer 3 security questions. Below this, it says 'Select all that apply' and 'Question 2: What cities have you lived in?'. There are four options: 'Chicago, Illinois' (checked), 'Portland, Oregon' (unchecked), 'Dallas, Texas' (checked), and 'St Louis, Missouri' (unchecked). A green 'NEXT' button is at the bottom, followed by a link 'Learn more about authentication' with a downward arrow.

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

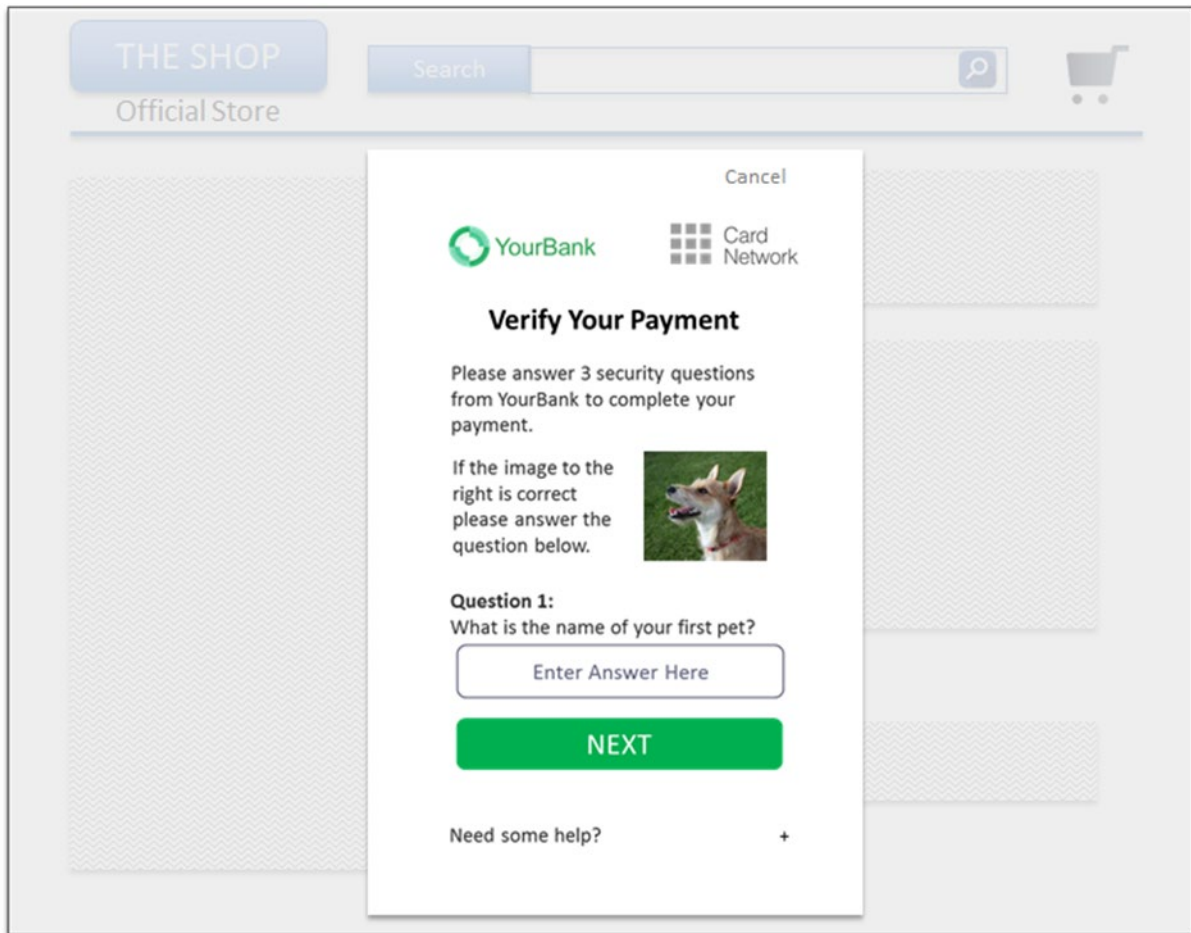


Figure 4.23 Sample Browser with Inline UI—PA (NEW)

THE SHOP

Official Store

Search

Cancel


YourBank

Card Network

Verify Your Payment

Please answer 3 security questions from YourBank to complete your payment.

If the image to the right is correct please answer the question below.



Question 1:
What is the name of your first pet?

Enter Answer Here

NEXT

Need some help? +

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 Out at the ACS~~), and Challenge Cancellation 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 Out at the ACS~~), and Challenge Cancellation 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	Condition Inclusion
Challenge Cancellation Indicator							Value of 04 or 05 is required when Transaction Status Reason = 14.
Challenge Data Entry							<p>Required when:</p> <ul style="list-style-type: none"> ACS UI Type = 01, 02, or 03, AND Challenge data has been entered in the UI, AND Challenge Cancellation Indicator, AND Resend Challenge Information Code <p>Are not present</p> <p>See Table A.14 for Challenge Data Entry conditions.</p>



Data Element/ Field Name	Description	Source	Length/Format/ Values	Device Channel	Message Category	Message Inclusion	Condition Inclusion
Challenge HTML Data Entry							Required when <ul style="list-style-type: none"> ACS UI Type = 05. AND challenge data has been entered into the UI. Challenge Cancellation Indicator is not present.
OOB Continuation Label							Note: If present, either of the following must also be present: <ul style="list-style-type: none"> Challenge Information Header, OR Challenge Information Text
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 .						



Data Element/ Field Name	Description	Source	Length/Format/ Values	Device Channel	Message Category	Message Inclusion	Condition Inclusion
Submit Authentication Label							<p>Note: If present, either of the following must also be present:</p> <ul style="list-style-type: none"> Challenge Information Header, OR Challenge Information Label, OR Challenge Information Text
3DS Requestor Prior Transaction Authentication Information			Format: String Values accepted:				
Transaction Status	Note: The Final CRes message can contain only a value of Y or N.					01-PA: Final CRes = GR 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="eyJ0aHJlZURTU2VydmVyVHJhbnNJRCI6IjNhYzdjYWE3LWFhNDItMjY2My03OTFiLTJhYzA1YTU0MmM0YSIsInRocmVlRFNnZXRob2Rob3RpZmlj
YXRpb25VUkwiOiJ0aHJlZURTU2V0aG9kTm90aWZpY2F0aW9uVGVJMin0">
</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: <ul style="list-style-type: none">• 2• 02• 002	JSON Data Type: String
Number of Transactions Per Day	Example values: <ul style="list-style-type: none">• 2• 02• 002	
Number of Transactions Per Year	Example values: <ul style="list-style-type: none">• 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: <ul style="list-style-type: none">123012300123	
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 <ul style="list-style-type: none">Value = Y	The ACS sends the 3DS SDK a CRes message.
Missing	01, 02, or 03	Missing	Present <ul style="list-style-type: none">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
OOB App Label	oobAppLabel	Zone 3



Data Element	Field Name	Zone
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 \geq 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:

[Req343]

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



Data Element/ Field Name	Description	Source	Length/Format/ Values	Device Channel	Message Category	Message Inclusion	Condition Inclusion
Instalment Payment Data			<p>Example values accepted:</p> <ul style="list-style-type: none"> • 2 • 02 • 002 				
Purchase Amount			<p>Example: purchase amount is USD 123.45:</p> <p>Values accepted:</p> <ul style="list-style-type: none"> • 12345 • 012345 • 0012345 <p>If the purchase amount is USD 123.45, element will contain the value 12345.</p>				
Recurring Frequency			<p>Example values accepted:</p> <ul style="list-style-type: none"> • 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 shall be~~ **It shall 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 <ul style="list-style-type: none">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: <pre>"issuerImage" :{ "medium": "http://acs.com/medium_ image.svgpng", "high": "http://acs.com/high_image. svgpng", "extraHigh": "http://acs.com/extraHigh_image. svgpng" }</pre>	



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.svgpng", "high": "http://ds.com/high_image.svgpng", "extraHigh": "http://ds.com/extraHigh_image.svgpng" }	

June 2018 v2

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-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 fall-back 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 check-out page. It is used for the ACS to load JavaScript which gathers 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	Condition Inclusion
3DS Server URL		ACS					

April 2018 v1

Throughout specification:

Updated all instances of:

- Transaction Status Reason ~~Code~~ 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]

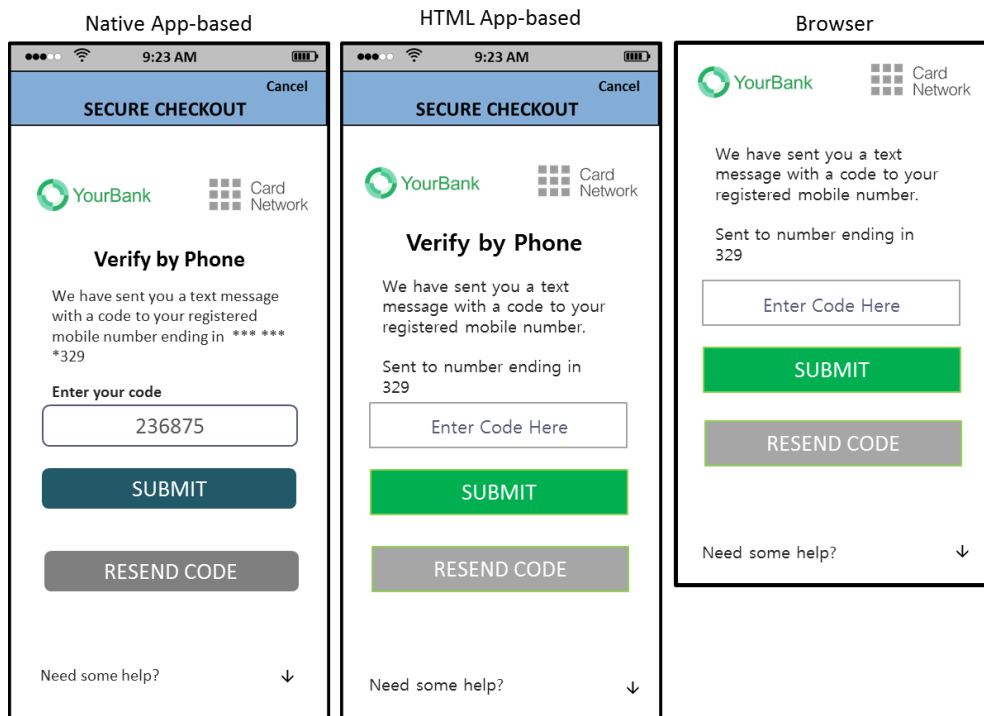
~~Embed all resources in the ACS provided HTML and do not fetch via external URLs.~~ The ACS shall not lead the Cardholder outside of the authentication flow by redirecting to any registration or marketing pages. Any redirection shall be used for authentication purposes only. The ACS shall only load 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

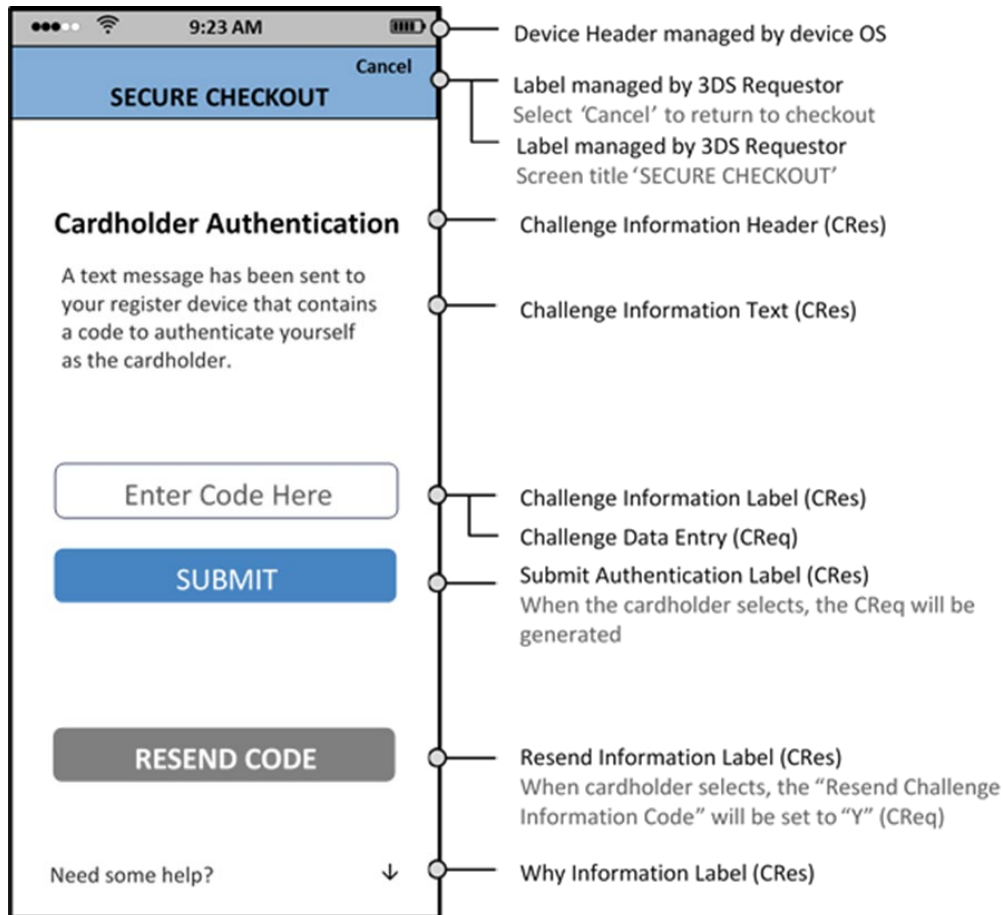
- Updated the **Verify** Button Label to **Submit**.



4.2.2 Native UI Templates

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

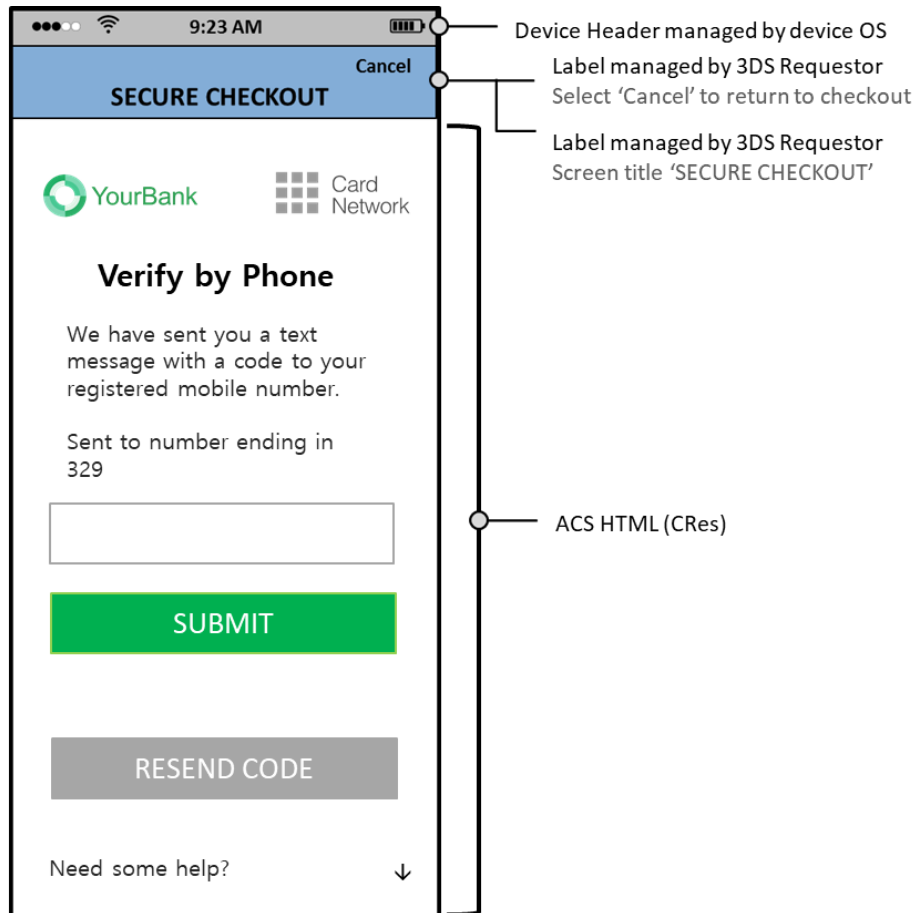
- Updated the ~~Confirm~~ Button Label to **Submit**.



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 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 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 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": "ACSURL": "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	Condition 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	Condition 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	Condition Inclusion
Challenge Cancellation Indicator			<ul style="list-style-type: none"> 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 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				
Results Message Status			<ul style="list-style-type: none"> 03 = ARes (Transaction Status = C) challenge data not delivered to the 3DS Requestor due to technical error 				



Data Element/ Field Name	Description	Source	Length/Format/ Values	Device Channel	Message Category	Message Inclusion	Condition Inclusion
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: ObjectString				
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": "extensionField1",  
    "id": "ID1",  
    "criticalityIndicator": true,  
    "data": {  
      "valueOne": "value"  
    }  
  },  
  {  
    "name": "extensionField2",  
    "id": "ID2",  
    "criticalityIndicator": true,  
    "data": {  
      "valueOne": "value1",  
      "valueTwo": "value2"  
    }  
  },  
  {  
    "name": "sharedData",  
    "id": "ID3",  
    "criticalityIndicator": false,  
    "data": {  
      "value3": "IkpTT05EYXRhIjogew0KImRhdGExIjogInNvbWUgZGF0YSIsDQoiZGF0YT  
IiOiAic29tZSBvdGhlciBkYXRhIjogKfQ=="  
    }  
  }  
]
```

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	<p>For example, for method:if the 3DS Requestor Authentication Method is:</p> <ul style="list-style-type: none"> 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