EMV® Specification Bulletin No. 256

August 2022

EMV® 3-D Secure Protocol and Core Functions Specification v2.3.1.0

This Specification Bulletin No. 256 provides the updates, clarifications and errata incorporated into the EMV®3-D Secure Protocol and Core Functions Specification since version 2.3.0.0.

Applicability

This Specification Bulletin applies to:

- *EMV® 3-D Secure Protocol and Core Functions Specification, Version 2.3.1.0*

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

Related Documents

EMV® 3-D Secure Protocol and Core Functions Specification, Version 2.3.0.0

Effective Date

- *August 2022*

Contents

Throughout Specification

- Revisions added to improve grammar, consistency, clarity and readability without any effect on the meaning or interpretation of the specification are not included in this bulletin.
- Updates made to defined abbreviations, such as EC(C) and DH (Diffie–Hellman), have no substantive effect on the use of the underlying specification and are not reflected in this bulletin.
- In Section 5.9, references to Section 5.1.6 were replaced with references to Section 5.1.7.
- Instances of SDK have been replaced with 3DS SDK.

1.3 Normative References

Table 1.1 Normative References

Reference	Publication Name	Bookmark
RFC 7233	*Hypertext Transfer Protocol (HTTP/1.1): Range Requests*	https://datatracker.ietf.org/doc/html/rfc7233

1.4 Acknowledgement

Reference	Publication Name	Bookmark
ISO 8583-1	ISO 8583-1 *Financial transaction card originated messages — Interchange message specifications — Part 1: Messages, data elements and code values*	https://www.iso.org/standard/31628.html

1.5 Definitions

Table 1.3 Definitions

Term	Definition
Card Range Data File	The file containing the JSON Card Range Data object. The Card Range Data provides to the 3DS Server the 3DS protocol versions supported by the card ranges hosted by the ACS, and other optional information (e.g. 3DS Method, Message Extension).
Decoupled Authentication Fallback	An additional challenge option for an ACS during the Challenge process. By returning Transaction Status = D in the RReq message, the ACS requests that the 3DS Server initiate a subsequent 3DS authentication with Decoupled Authentication supported.
Platform Provider	An entity that provides a digital ecosystem consisting of an operating system and/or hardware components, capable of uniquely identifying the consumer and their device through a user ID and a hardware-derived device ID, and sharing these IDs for the purposes of risk assessment and fraud prevention.
Preparation Response (PRes) Message	Response to the PReq message that contains the DS Card Ranges, active Protocol Versions for the ACS and DS and 3DS Method URL, or a Card Range Data File URL to download this information, so that updates can be made to the 3DS Server's internal storage.

Term	Definition
Secure Payment Confirmation	FIDO-based authentication to securely confirm payments initiated via the Payment Request API on a Browser (refer to ~~Web Payments Working Group (w3.org)~~w3.org for additional information).

1.6 Abbreviations

Table 1.4 Abbreviations

Abbreviation	Description
~~CA DS~~	~~Certificate Authority Directory Server~~
CEK	Content Encryption Key
DH	Diffie–Hellman
DS CA	Directory Server Certificate Authority
ECC	Elliptic Curve Cryptography

1.8 Supporting Documentation

- *EMV® 3-D Secure Message Extensions*
  - *EMV® 3-D Secure Bridging Message Extension*
  - *EMV® 3-D Secure Device Acknowledgement Message Extension*
  - *EMV® 3-D Secure Payment Token Message Extension*
  - *EMV® 3-D Secure Travel Industry Message Extension*

1.9 Terminology and Conventions

3DS SDK

2. **Split-SDK**—Client-server implementation of the 3DS SDK. Some functions of the Split-SDK entity can be performed by either a Split-SDK Client or a Split-SDK Server or, in some situations, both. The Split-SDK has multiple variants depending on the Consumer Device and the 3DS Requestor ~~e~~Environment. These variants include the ~~Limited SDK~~Split-SDK/Native, Split-SDK/Shell ~~SDK~~ and Split-SDK/Browser, ~~SDK~~ and each ~~are~~is defined in the *EMV® 3-D Secure—Split-SDK Specification.*

2.4 3-D Secure Messages

2.4.1 Authentication Request Message (AReq)

There is only one AReq message per authentication, except for the 3DS Requestor-Initiated SPC Authentication and Decoupled Authentication Fallback.

2.4.2 Authentication Response Message (ARes)

There is only one ARes message per ~~transaction~~authentication, except for the 3DS Requestor-Initiated SPC Authentication and Decoupled Authentication Fallback.

2.4.3 Challenge Request Message (CReq)

- **Browser-based** – The CReq message is ~~sent~~ formed by the 3DS Server and is posted through the Cardholder Browser. There is only one CReq message per challenge.

2.4.5 Results Request Message (RReq)

There is only one RReq message per ~~AReq message~~3DS transaction.

Chapter 3 EMV 3-D Secure Authentication Flow Requirements

3.1 App-based Requirements

Step 6: The DS

[Req 390]

If ~~the~~ SDK Type = 02, ~~03, 04 or 05,~~ verify the signature in the SDK Server Signed Content, as defined in Section 6.2.2.4.

[Req 420]

Identifi~~es~~y the DS public key used by the 3DS SDK to encrypt Device Information from the key identifier (kid) in the SDK Encrypted Data.

Step 7: The ACS

[Req 321]

If a Decoupled Authentication challenge is deemed necessary (Transaction Status = D) and 3DS Requestor Decoupled Request Indicator = Y or B, the ACS determines whether an acceptable challenge method is supported by the ACS based in part on the following data element received in the AReq message: 3DS Requestor Decoupled Max Time. The ACS performs the following:

a. Sets Transaction Status = D for Decoupled Authentication.

b. ~~Sets~~Includes Decoupled (= 12) in the Authentication Method.

c. Sets ACS Decoupled Confirmation Indicator = Y.

d. Stores the 3DS Server Transaction ID and DS Transaction ID (for subsequent RReq processing).

Step 17: The ACS

New Requirement 461 was added after Requirement 310.

[Req 461]

If the ACS determines that Decoupled Authentication Fallback is necessary and 3DS Requestor Decoupled Request Indicator = F or B, inform the Cardholder of Decoupled Authentication using the Information UI template as defined in Chapter 4 with additional CRes messages, then continue to **[Req 61]** to prepare the final CRes message.

[Req 61]

Check the ~~authentication~~ data ~~entered by the Cardholder:~~received in the CReq message and assess the status of the authentication.

- If ~~correct~~the authentication is successful, then the ACS:
    - Increments the Interaction Counter
    - Sets ~~the~~ Transaction Status = Y
    - Sets the ECI value as defined by the specific DS

  - Generates the Authentication Value as defined by the DS
  - Sets ~~the~~ Challenge Completion Indicator = Y
  - Continues with Step 18
- If ~~incorrect and~~the authentication has failed or is not completed, then the ACS:
  - Increments the Interaction Counter and compares it to the ACS maximum challenges.
  - If the Interaction Counter ≥ ACS maximum challenges or the authentication has failed, the ACS:
    - Sets ~~the~~ Transaction Status = N
    - Sets ~~the~~ Transaction Status Reason = 19
    - Sets the ECI value as defined by the specific DS
    - Sets ~~the~~ Challenge Completion Indicator = Y
    - Continues with Step 18
  - Else if~~If~~ the Interaction Counter < ACS maximum challenges and the authentication is not completed, the ACS:
    - Obtains the information needed to display a repeat Challenge on the Consumer~~'s~~ Device per the selected challenge method and ACS UI Type.
    - Continues with Step 13.

Step 18: The ACS

New Requirement 462 was added directly before Requirement 62.

[Req 462]

For a Challenge Flow (ARes Transaction Status = C), if 3DS Requestor Decoupled Request Indicator = F or B, and if the ACS has determined that Decoupled Authentication Fallback is necessary,

- Set Transaction Status = D.
- Set Transaction Status Reason = 29 or 30.

Step 20 The 3DS Server

The 3DS Server shall:

[Req 70]

Receive the RReq message or Error Message from the DS and Validate as defined in Section 5.9.9.

If the message is in error, the 3DS Server **ends processing**.

[Req 463]

If Transaction Status = D and if 3DS Requestor Decoupled Request Indicator = F or B, set Results Message Status to 04.

Note: The 3DS Server initiates a 3RI transaction with Decoupled Authentication as defined in Section 3.4.

3.3 Browser-based Requirements

Step 8 The ACS

Note: The ACS uses the ~~Device~~ Browser Information (as defined in Section A.6) received in the AReq message and the 3DS Method to recognise the device, assess transaction risk, and determine if it can complete the authentication.

[Req 325]

If a Decoupled Authentication challenge is deemed necessary (Transaction Status = D) and 3DS Requestor Decoupled Request Indicator = Y or B, the ACS determines whether an acceptable challenge method is supported by the ACS based in part on the following data element received in the AReq message: 3DS Requestor Decoupled Max Time.

Step 15 The ACS

New Requirement 464 was added at the beginning of Step 15, directly before Requirement 123.

The ACS shall:

[Req 464]

If the ACS determines that Decoupled Authentication Fallback is necessary, inform the Cardholder of Decoupled Authentication in the final CRes message using the Information UI template as defined in Chapter 4.

[Req 123]

Check the authentication data ~~entered by the Cardholder~~received and assess the status of the authentication:

- If correct, then the ACS:
  - Increments the Interaction Counter
  - Sets ~~the~~ Transaction Status = Y
  - Sets the ECI value as defined by the specific DS
  - Generates the Authentication Value as defined by the DS
  - Continues with Step 16
- If ~~incorrect and~~ the authentication has failed, is not completed or the Cardholder has selected to cancel the authentication, then the ACS:
  - Increments the Interaction Counter and compares it to the ACS maximum challenges
  - If the Interaction Counter ≥ ACS maximum challenges or the authentication has failed or the Cardholder has selected to cancel the authentication, the ACS:
    - Sets ~~the~~ Transaction Status = N

    - Sets ~~the~~ Transaction Status Reason = 19
    - Sets the ECI value as defined by the specific DS
    - Continues with Step 16
  - Else ~~(~~if the Interaction Counter < ACS maximum challenges or the authentication is not completed, the ACS:
    - Obtains the information needed to display a repeat Challenge on the Consumer~~’s~~ Device per the selected challenge method and ACS UI Type.
    - Prepares the authentication User Interface (ACS UI) to the Cardholder Browser which may contain HTML, JavaScript, etc.
    - Continues with Step 12.

The process of exchanging HTML will repeat until a determination is made by the ACS.

Step 16 The ACS

New Requirement 465 was added in Step 16 directly before Requirement 124.

[Req 465]

For a Challenge Flow (ARes Transaction Status = C), if 3DS Requestor Decoupled Request Indicator = F or B, and if the ACS has determined that Decoupled Authentication Fallback is necessary, set Transaction Status = D.

Step 18 The 3DS Server

The 3DS Server shall:

[Req 132]

Receive the RReq message or Error Message from the DS and Validate as defined in Section 5.9.9.

If the message is in error, the 3DS Server **ends processing**.

[Req 466]

If Transaction Status = D and if 3DS Requestor Decoupled Request Indicator = F or B, set Results Message Status to 04.

Note: The 3DS Server initiates a 3RI transaction with Decoupled Authentication as defined in Section 3.4.

Step 21: The ACS

[Req 139]

Base64url-encode the final CRes message and include, if present in the CReq message, the 3DS Requestor Session Data in HTML form (as defined in Table A.3).

3.4 3RI-based Requirements

Step 2 The 3DS Server

New Requirement 467 was added in Step 2, between Requirement 423 and Requirement 276.

[Req 467]

In the case of Decoupled Authentication Fallback, the 3DS Server initiates a 3RI authentication within 60 seconds of receiving the RReq message from the previous transaction, containing:

- 3DS Requestor Decoupled Request Indicator = Y
- 3DS Requestor Prior Transaction Authentication Information object:
    - 3DS Requestor Prior Transaction Reference = ACS Transaction ID from the RReq message indicating that Decoupled Authentication is to be performed
    - 3DS Requestor Prior Transaction Authentication Method = 02 (Cardholder challenge occurred by ACS).

3.5 SPC-based Authentication Requirements

3.5.1 3DS Requestor Initiates SPC Authentication

Several new requirements have been added in this section.

Step 5:

[Req 447]

The 3DS Requestor website shall display the Processing screen as per the requirements in Section 4.3.1.1 until the SPC API is invoked in Step 10a.

Step 6:

The 3DS Server recognises that the ACS supports SPC-based authentication in the ACS Information Indicator~~and reports that to the ACS (3DS Requestor SPC Support = Y).~~

[Req 448]

The 3DS Server shall set the 3DS Requestor SPC Support = Y in the AReq message.

Step 8:

The ACS recognises that ~~an~~ SPC-based ~~interaction with the Cardholder~~authentication is ~~required and returns~~supported.

[Req 449]

If the ACS determines that SPC is the selected authentication method, the ACS shall return the following:

The following note was added at the end of Step 8:

Note: Depending on implementation, it is possible that the DS performs the~~se~~ functions described in Step 8 on behalf of the ACS. In ~~this~~ such cases, the~~se~~ ~~parameters~~ data are provided ~~returned~~ by the DS ~~and reported~~ to the ACS ~~/3DS Server~~ in Step 7 and to the 3DS Server in Step 9.

Step 10a~~:~~ The 3DS Requestor

In this step, the note was converted into a requirement.

~~Note:~~

[Req 450]

The 3DS Requestor ~~is not allowed to~~shall not change or store any of the data received for the SPC authentication from the 3DS Server.

Step 10b~~:~~ The Cardholder

The Cardholder authenticates using the FIDO authenticator on his/her device.

Step 10c~~:~~ The 3DS Requestor

The 3DS Requestor retrieves the Assertion Data from the SPC API call.

Step 10d~~:~~ The 3DS Requestor

The 3DS Requestor initiates a second 3DS Authentication Request to return Assertion Data to the ACS, and displays the processing screen as defined in Section 4.3.1.1 during the AReq message processing.

Step 10e The 3DS Server

In this step~~, the 3DS Server~~, in addition to performing the requirements defined in Step 6 of the Browser flow, the 3DS Server shall:

Step 10f~~:~~ The DS

The ~~3DS~~ DS ~~components~~ performs the requirements as defined in Step 7 of the Browser-based flow.

In this step, the note was converted into a requirement.

[Req 451]

~~Note:~~

If the DS evaluates the Assertion Data on behalf of the ACS, the DS ~~includes~~ shall include the verification result in the 3DS Requestor Authentication Method Verification Indicator

Step 10g~~:~~ The ACS

Depending on:

- the verification of the signature in the Assertion Data ~~and other transaction information~~;
- the consistency of the transaction data between the first and second AReq message; AND
- the consistency of the Assertion Data with the data from the AReq message,

the ACS ~~decides~~determines the ~~final~~disposition of the transaction, as defined in **[Req 107]** (e.g. authenticated [Transaction Status = Y]~~,~~ or to be further ~~C~~challenged [Transaction Status = C]~~, etc.~~).

Step 10h~~:~~ The DS

The DS performs the requirements as defined in Step 9 of the Browser flow.

Step 10i~~:~~ The 3DS Server

The 3DS Server performs the requirements as defined in Step 10 of the Browser flow.

3.5.2 The ACS Initiates SPC Authentication

Step 8: In this step, the ACS recognises a pre-registered FIDO authenticator on the device for this Cardholder and ~~uses the~~selects Authentication Method = 14 (SPC), in combination with Transaction Status = C to indicate to the 3DS Server that the ACS intends to perform SPC authentication as the challenge method.

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

4.1 3-D Secure User Interface Templates

[Req 418]

Support full-screen vertical and horizontal scrolling for HTML UI, and at minimum, support full-screen vertical scrolling for the Native UI, for all ACS-provided content. The Header zone may remain anchored at the top of the display.

4.2 App-based User interface Overview

4.2.1.1 3DS SDK/3DS Requestor App

Figure 4.10 provides a sample format for the OOB template for a manual transfer to and from the OOB Authentication App for an App-based processing flow.

4.2.7.2 ACS

[Req 164]

Include in the ACS HTML an action which triggers a location change to a specified (HTTPS://EMV3DS/challenge) URL upon the Cardholder completing data input and pressing Submit, for ACS UI Type = 05, and only if manual app switching is used for OOB for ACS UI Type = 06.

Chapter 5 EMV 3-D Secure Message Handling

5.1 General Message Handling

5.1.1 HTTP POST

[Req 186]

The body of the HTTP message shall contain the JSON message properly formatted utilising the JSON required UTF-8 character set as defined in RFC 7159, or JWE/JWS object format as defined in RFC 7516/RFC 7515. To maximise the message content, it is recommended to remove any whitespace (space, carriage return, line feed etc.) characters outside of quoted strings of the JSON data.

5.1.2 HTTP Header—Content-Type

New wording added directly after Requirement 191.

The HTTP headers contain additional information for the support of 3-D Secure messaging:

[Req 468]

For the AReq, CReq, RReq, OReq or PReq messages, the 3DS component sending the message shall include its own Transaction ID using the X-Request-ID in the HTTP header, as defined in Table A.29.

[Req 469]

For the ARes, CRes, RRes, ORes or PRes messages, the 3DS component sending the message shall include its own Transaction ID using X-Response-ID and the X-Request-ID received in the Request message in the HTTP header, as defined in Table A.29.

5.1.4 Protocol and Message Version Numbers

[Req 320]

The Message Version Numbe~~rs shall be validated to ensure that they are consistent across a 3-D Secure transaction. The~~ is set by the 3-D Secure component initiating the 3DS message and the 3-D Secure components shall validate that the Message Version Number remains unchanged throughout the 3-D Secure transaction. The 3-D Secure component that identifies a validation error shall return an Error Message with the applicable Error Component and Error Code = 203.

~~For example, when the DS receives an RReq message, the DS will validate that the Message Version Number matches the AReq message.~~

Requirement 311 follows unchanged.

~~**Note: For all 3-D Secure transactions, the 3DS Server sets the Message Version Number that all components will utilise.**~~

5.1.6 Message Parsing

[Req 430]

If the 3DS Server receives more than one ~~ARes message and/or~~ RReq message during a transaction, then it shall return Error Code = 312.

[Req 431]

If the 3DS Server receives an RReq message and the Transaction Status does not = C or D or S in the corresponding ARes message, then it shall return Error ~~=~~Code = 313.

[Req 433]

If the DS receives an RReq message and the Transaction Status does not = C or D or S in the corresponding ARes message, then it shall return Error Code = 313.

5.1.7 Message Content Validation

[Req 434]

If the value of a data element is in the range of ~~“Reserved for DS use” and not recognised, or in the range of~~ “Reserved for EMVCo future use”, all 3DS components shall return an Error Message (as defined in Section A.9) with the applicable Error Component and Error Code = 207.

Note: The error requirements and the use of Error Code = 207 for the values in the range of “Reserved for DS use” are defined by the DS.

5.5 Timeouts

5.5.1 Transaction Timeouts

New verbiage, which includes four new requirements, has been added after Requirement 344.

For an SPC challenge (Transaction Status = S), the ACS shall:

[Req 452]

Set a timeout value of 10 minutes (or 600 seconds) after sending the first ARes message.

[Req 453]

If the timeout expires before the second AReq message is received, send an RReq message within 60 seconds to the DS, to be passed to the 3DS Server, with Transaction Status = N and Transaction Status Reason = 14.

This completes the SPC challenge for the ACS. In a timeout situation, the 3DS Server proceeds as defined in Section 3.3, Step 18 for the RReq and RRes messages. At the end of Step 18, the 3DS Server notifies the 3DS Requestor of the timeout. The method used to notify the 3DS Requestor is outside the scope of this specification.

When notified of the timeout, the 3DS Requestor takes the appropriate action and message to the Cardholder.

For an SPC challenge (Transaction Status = S), the 3DS Server shall:

[Req 454]

Set a timeout value of 9 minutes (or 540 seconds) after successfully providing the SPC data to the 3DS Requestor.

[Req 455]

If the timeout expires before receiving the Assertion Data from the 3DS Requestor, send the second AReq message to the ACS with SPC Incompletion Indicator = 03.

5.6 PReq/PRes Message Handling Requirements

The PReq/PRes messages are utilised by the 3DS Server to cache information about the Protocol Version Numbers(s) supported by available ACSs, the DS, and also any URL to be used for the 3DS Method call. The information provided on the Protocol Version Number(s) supported by ACSs and the DS can be utilised in ~~the App-based, Browser-based and 3RI~~all 3DS flows.

The 3DS Server has the option to receive the Card Range Data in the PRes message or, if optionally supported by the DS, to receive a URL to download a file containing the Card Range Data.

The 3DS Server formats a PReq message (as defined in Table B.6) and sends the request to the DS. If this is the first time that the cache is being loaded (or if the cache has been flushed and needs to be reloaded, or if the DS does not support partial cache updates), the Serial Number data element is not included in the request, which will result in the DS returning the entire list of participating card range information

Otherwise, the 3DS Server ~~should include~~includes the Serial Number from the most recently processed PRes message, which will result in the DS returning only the changes since the previous PRes message. The Serial Number is not used or provided when the DS and 3DS Server use the Card Range Data File download option.

The DS manages the Serial Number to ensure that the response to a PReq message for a particular Serial Number includes all updates posted since that Serial Number was issued. If the Serial Number provided in the PReq message is invalid (for example, ~~if~~ too old and can no longer be found), the response should be an Error Message with ~~an~~ Error Code = 307.

If the PReq message does not include a Serial Number, the DS PRes message response ~~shall contain~~or file contains all card range entries.

If the Serial Number has not changed, the DS ~~would~~does not provide back the Card Range Data element but ~~would include~~includes the Serial Number in the PRes message.

The 3DS Server shall:

[Req 246]

Call each registered DS for:

- An update for all Card Range Data (Serial Number not provided) every 12 hours at maximum. If there is an error in the received Card Range Data, the 3DS Server calls each registered DS once per hour at ~~a~~ maximum for a complete or partial update.
- A partial update providing the Serial Number once per hour at maximum.

Requirement 425 follows with no wording change. Requirement 426 (with some revisions) and Requirement 428 have been moved.

~~The DS shall:~~

~~**[Req 426]**~~

~~Send a response with either:~~

- ~~A compressed response body and a Content-Encoding header that specifies that gzip encoding was used (“Content-Encoding: gzip “), OR~~
- ~~An uncompressed response body.~~

~~**[Req 428]**~~

~~If the DS receives a request without the “Accept-Encoding: gzip” in the header of an 3DS Server HTTP request, then the DS does not return an Error Message and returns the data in an uncompressed format.~~

~~The 3DS Server shall:~~

[Req 456]

Support Range Requests, as defined by RFC 7233, if the Card Range Data Download Indicator is present in the PReq message.

Requirements 247, 248 and 249 follow with no wording change.

[Req 428]

If the DS receives a request without the “Accept-Encoding: gzip” in the header of an 3DS Server HTTP request, then the DS does not return an Error Message and returns the data in an uncompressed format.

Requirement 303 remains unchanged and follows Requirement 428 in this Section.

~~**[Req 414]**~~

~~Prepare the PRes message and ensure~~

[Req 457]

Support Range Requests, as defined by 7233, for the Card Range Data File download.

Requirement 414 has been deleted.

[Req 458]

Ensure that there is no overlap or conflict in the card ranges contained in the Card Range Data (Start and End of the card range).

[Req 459]

Provide in the Card Range Data data element only information about card ranges that are participating in EMV 3-D Secure and are registered with the DS that is responding to the request.

[Req 460]

Prepare the Card Range Data File if supported by the DS and if the Card Range Data Download Indicator was present in the PReq message. The Card Range Data File contains the entire list of participating card ranges, e.g. the JSON object Card Range Data; {"cardRangeData": [...] }.

[Req 426]

Prepare the PRes message and s~~S~~end a response with either:

- A compressed response body and a Content-Encoding header that specifies that gzip encoding was used (“Content-Encoding: gzip”), OR
- An uncompressed response body. The DS shall use the uncompressed response format when it provides the Card Range Data File URL in the PRes message.

[Req 250]

Send the PRes message containing the DS card range information as defined in the Card Range Data data element defined in Table A.1 or the Card Range Data File URL, if supported.

- 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~~the entire list of participating card ranges in the Card Range Data using only Action Indicator = A.

Requirement 251 has been deleted.

~~**[Req 251]**~~

~~Send the PRes message containing only information about card ranges that are participating in EMV 3-D Secure and are registered with the DS that is responding to the request.~~

Requirement 304 remains unchanged and is directly followed by new verbiage:

The 3DS Server retrieves the Card Range Data data element from the PRes message or from the file downloaded from the Card Range Data File URL. For the file download, it is recommended that the 3DS Server and the DS support a compressed format (“Accept-Encoding: gzip” in the HTTP request).

In Requirement 385, the following change has been made to the third second-level bullet:

[Req 385]

- ○ Discard~~s~~ all updates contained in the ~~PRes message~~Card Range Data, and use~~s~~ previously stored cache information or alternatively, ignore~~s~~ all existing cache information.

5.9 Message Error Handling

5.9.5 ACS Creq Message Error Handling—01-APP

- ○ If SDK Type = 02~~, 03, 04 or 05~~ (as received in the AReq message) AND the CReq message is not protected using A128GCM ("enc" as defined in section 6.2.4.3 is not A128GCM), the ACS:

5.11 OReq/ORes Message Handling Requirements

[Req 435]

Prepare and send the OReq message or the sequence of OReq messages in the sequence number order via a secure link with the recipient (3DS Server or ACS) established as defined in Table B.10.

Chapter 6 EMV 3-D Secure Security Requirements

6.1 Link

6.1.1 Link a: Consumer Device—3DS Requestor

The following sentence is added at the end of the section:

It is recommended to use TLS 1.2 or higher for all these links.

6.1.4.1 For App-based CReq/CRes

- Protocol—TLS ~~Internet~~1.2 or higher

6.1.4.2 For Browser-based CReq/CRes

- Protocol—TLS ~~Internet~~1.2 or higher

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

- Protocol—TLS ~~Internet~~1.2 or higher

6.2 Security Functions

6.2.2 Function I: 3DS SDK Device Information Encryption and Split-SDK Server Signature to DS

6.2.2.3 Split-SDK Server Signature

- "x5c": X.5C v3: Cert (*Pb*$_{SDK}$) and chaining certificates, if present (without any DS CA public key certificate)

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

6.2.3.2 ACS Secure Channel Setup

- "x5c": X.5C v3: Cert(*Pb*$_{ACS}$) and chaining certificates, if present (without any DS CA public key certificate)

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

6.2.4.1 3DS SDK—CReq

- "enc":
  - For SDK Type = 01: either A128CBC-HS256 or A128GCM:
  - For SDK Type = 02~~, 03, 04 or 05~~: A128GCM

Annex A 3-D Secure Data Elements

A.4 EMV 3-D Secure Data Elements

This annex contains an alphabetical listing of all EMV 3-D Secure data elements. Data element information and the Standards used to identify the information are as follows:

- Length/Format/Values—Identifies the value length detail, JSON data format, and if applicable, the values associated with the data element. The term "character" in the Length Edit criteria refers to one UTF-8 character. The length of a JSON object is the length in characters of the overall string representing the object.

Some rows have been rearranged in alphabetical order, without any effect on the meaning, and are therefore not replicated in this specification bulletin.

Table A.1 EMV 3-D Secure Data Elements

Data Element/ Field Name	Description	Source	Length/Format/ Values	Device Channel	Message Category	Message Inclusion	Condition Inclusion
3DS Requestor App URL Indicator Field Name: `threeDSRequestorAppURLInd`	Indicates whether the OOB Authentication App used by the ACS during a challenge supports the 3DS Requestor App URL.	ACS	Length: 1 character JSON Data Type: String Values accepted: • Y = 3DS Requestor App URL is supported by the OOB Authentication App	01-APP	01-PA 02-NPA	ARes = R	

Data Element/ Field Name	Description	Source	Length/Format/ Values	Device Channel	Message Category	Message Inclusion	Condition Inclusion
			• N = 3DS Requestor App URL is NOT supported by the OOB Authentication App				
3DS Requestor Authentication Indicator			Values accepted: • 08 = Split shipment • 09 = Delayed shipment • 10 = Split payment • ~~08~~11–79 = Reserved for EMVCo future use (values invalid until defined by EMVCo)				
3DS Requestor Decoupled Max Time							Required if 3DS Requestor Decoupled Request Indicator = Y or F or B.
3DS Requestor Decoupled Request Indicator	Note: if the element is not provided, the expected action is for the ACS to interpret as N (Do not use Decoupled Authentication).		Values accepted:				

Data Element/ Field Name	Description	Source	Length/Format/ Values	Device Channel	Message Category	Message Inclusion	Condition Inclusion
			• Y = Decoupled Authentication is supported and is preferred as a primary challenge method if a challenge is necessary (Transaction Status = D in ARes). • N = Do not use Decoupled Authentication • F = Decoupled Authentication is supported and is to be used only as a fallback challenge method if a challenge is necessary (Transaction Status = D in RReq). • B = Decoupled Authentication is supported and can be used as a primary or fallback challenge method if a challenge is necessary (Transaction Status = D in either ARes or RReq).				

Data Element/ Field Name	Description	Source	Length/Format/ Values	Device Channel	Message Category	Message Inclusion	Condition Inclusion
			~~Note: if the element is not provided, the expected action is for the ACS to interpret as N, do not use Decoupled Authentication.~~				
3DS Requestor Prior Transaction Authentication Information						AReq = ~~O~~C	~~Optional, recommended to include~~ Required for 3RI in the case of Decoupled Authentication Fallback or for SPC
3RI Indicator			Values accepted: • 06 = Split/~~delayed~~ shipment • 15 = Delayed shipment • 16 = Split payment • 17–79 = Reserved for EMVCo future use (values invalid until defined by EMVCo)				
Acquirer Country Code			Values accepted:				

Data Element/ Field Name	Description	Source	Length/Format/ Values	Device Channel	Message Category	Message Inclusion	Condition Inclusion
			• ISO 3166-1 numeric three-digit country codes, other than exceptions listed in Table A.5.				
Authentication Method			If SDK Type = ~~03~~and Split-SDK Type/Limited Indicator = Y, a value of 01 or 06 is not valid.				
Browser User Device ID Field Name: `deviceId`	Unique and immutable identifier linked to a device that is consistent across 3DS transactions for the specific user device. Examples: • Hardware Device ID • Platform-calculated device fingerprint Refer to D021 in the SDK Device Information	3DS Server	Length: Variable, maximum 64 characters JSON Data Type: String	02-BRW	01-PA 02-NPA	AReq = C	Required if available.
Browser User ID Field Name: `userId`	Identifier of the transacting user's Browser Account ID.	3DS Server	Length: Variable, maximum 64 characters JSON Data Type: String	02-BRW	01-PA 02-NPA	AReq = C	Required if available.

Data Element/ Field Name	Description	Source	Length/Format/ Values	Device Channel	Message Category	Message Inclusion	Condition Inclusion
	This identifier is a unique immutable hash of the user's account identifier for the given Browser, provided as a string. Note: Cardholders may have more than one account on a given Browser. Refer to D026 in the SDK Device Information						
Card Range Data							Required if the Serial Number has changed in the prior PRes message or is absent in the PReq message AND Not present if the Card Range Data File URL is present
Card Range Data Download Indicator Field Name: `cardRangeDataDownloadInd`	Indicates if the 3DS Server supports Card Range Data from a file. Note: If present, this field contains the value Y.	3DS Server	Length: 1 character JSON Data Type: String Value accepted: Y = Download supported	N/A	N/A	PReq = C	Present only if the 3DS Server supports the Card Range Data File download

Data Element/ Field Name	Description	Source	Length/Format/ Values	Device Channel	Message Category	Message Inclusion	Condition Inclusion
Card Range Data File URL Field Name: `cardRangeDataFileURL`	Fully Qualified URL of the DS File containing the Card Range Data for download.	DS	Length: Variable, maximum 2048 characters JSON Data Type: String Value accepted: • Fully Qualified URL Example: https://server.dsdomainname.com/cardfile.json	N/A	N/A	PRes = C	Present only if the 3DS Server and the DS are using the Card Range Data File download
Card Security Code Status							Conditional ~~if Card Security Code received in AReq message~~based on DS rules
Challenge HTML Data Entry							Required when: • ACS UI Type = 05 or 06, AND • Challenge Cancelation Indicator is not present, AND • OOB Continuation Indicator is NOT = 02

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

Default-SDK Type
Field Name: `defaultSdkType`
	Indicates the characteristics of a Default-SDK.
	SDK Variant: SDK implementation characteristics
	Wrapped Indicator: If the Default-SDK is embedded as a wrapped component in the 3DS Requestor App
	Example:
	`"defaultSdkType":{`
	`"sdkVariant":"01",`
	`"wrappedInd":"Y"`
	`}`
		3DS Server	JSON Data Type: Object
			`sdkVariant`
			Length: 2 characters
			JSON Data Type: String
			Values accepted:
			• 01 = Native
			• 02–79 = Reserved for EMVCo future use (values invalid until defined by EMVCo)
			• 80–99 = Reserved for DS use
			`wrappedInd`
			Length: 1 character
			JSON Data Type: String
			Value accepted:
			• Y = Wrapped
			Only present if value = Y
				01-APP	01-PA
					02-NPA
						AReq = C	Required if SDK Type = 01

Device Information Recognised Version	Indicates the highest Data Version of the Device Information ~~that the ACS recognised from the AReq for this message pair~~supported by the ACS.		Length: Variable, minimum 3 characters JSON Data Type: String Values accepted:				

Data Element/ Field Name	Description	Source	Length/Format/ Values	Device Channel	Message Category	Message Inclusion	Condition Inclusion
			- ~~Note:~~ Any active Device Information Data Version is considered a valid value. Refer to *EMV® ~~Secure SDK—Device Information~~ Specification Bulletin 255* for values.				
Merchant Risk Indicator			Note: Data will be formatted into a JSON object prior to being placed into the ~~Device~~ Merchant Risk Indicator field of the message.				
Message Extension		3DS Server 3DS SDK ACS DS					
OOB App Label							Required if ~~`oobAppURL`~~ the OOB App URL is available and ACS UI Type = 04.

Data Element/ Field Name	Description	Source	Length/Format/ Values	Device Channel	Message Category	Message Inclusion	Condition Inclusion
OOB App URL							Required for ACS UI type = 04 or 06 if: • OOB App URL Indicator = 01 in the CReq message; AND • the ACS utilises the OOB Authentication App automatic switching feature.
OOB App URL Indicator Field Name: `oobAppURLInd`	Indicates if the 3DS SDK supports the OOB App URL.	3DS SDK	Length: Variable, maximum 48 characters JSON Data Type: String Values accepted: • 01 = Supported • 02 = Not supported by the device • 03 = Not supported by the 3DS Requestor	01-APP	01-PA 02-NPA	CReq = R	

Data Element/ Field Name	Description	Source	Length/Format/ Values	Device Channel	Message Category	Message Inclusion	Condition Inclusion
			• 04–79 = Reserved for EMVCo future use (values invalid until defined by EMVCo) • 80–99 = Reserved for DS use If SDK Type = 02 and Split-SDK Type = 02, the OOB App URL Indicator is set to 02. Note: OOB App URL does not work for the Split-SDK/Browser.				
Operation Prior Transaction Reference						OReq = ~~R~~O	
Operation Sequence			• `seqNum`: 2 characters Values accepted: ○ 01–99 • `seqTotal`: 2 characters Values accepted ○ 01–99				

Data Element/ Field Name	Description	Source	Length/Format/ Values	Device Channel	Message Category	Message Inclusion	Condition Inclusion
Payee Origin Field Name: `payeeOrigin`	The origin of the payee that will be provided in the SPC Transaction Data Refer to Secure Payment Confirmation.	3DS Server	Length: Variable, maximum 2048 characters JSON Data Type: String Value accepted: • Fully Qualified URL	02-BRW	01-PA 02-NPA	AReq = C	Required if 3DS Requestor SPC Support = Y
Purchase Amount							• Required for 02-NPA if 3DS Requestor Authentication Indicator = 02, ~~or~~ 03, 07, 08, 09 • Required for 02-NPA if 3RI Indicator = 01, 02, 06, 07, 08, 09, 11, 15
Purchase Currency							• Required for 02-NPA if 3DS Requestor Authentication Indicator = 02, ~~or~~ 03, 07, 08, 09

Data Element/ Field Name	Description	Source	Length/Format/ Values	Device Channel	Message Category	Message Inclusion	Condition Inclusion
							- Required for 02-NPA if 3RI Indicator = 01, 02, 06, 07, 08, 09, 11, 15
Purchase Currency Exponent							- Required for 02-NPA if 3DS Requestor Authentication Indicator = 02, ~~or~~ 03, 07, 08, 09 - Required for 02-NPA if 3RI Indicator = 01, 02, 06, 07, 08, 09, 11, 15
Purchase Date & Time							- Required for 02-NPA if 3DS Requestor Authentication Indicator = 02, ~~or~~ 03, 07, 08, 09 - Required for 02-NPA if 3RI Indicator = 01, 02, 06, 07, 08, 09, 11, 15

Data Element/ Field Name	Description	Source	Length/Format/ Values	Device Channel	Message Category	Message Inclusion	Condition Inclusion
Recurring Amount							• Required if: [3DS Requestor Authentication Indicator = 02 or 03; OR • 3RI Indicator = 01 or 02] • AND • Recurring Indicator/ Amount Indicator = 01
Recurring Date	Effective date of the new authorised amount following the first/promotional payment in a recurring or instalment transaction.						•
Recurring Frequency	Indicates the minimum number of days between authorisations for a recurring or instalment transaction.		Values accepted: • Numeric values between 1 and 9999				
Recurring Indicator			~~Length: 43 characters~~ **Frequency Indicator**				

Data Element/ Field Name	Description	Source	Length/Format/ Values	Device Channel	Message Category	Message Inclusion	Condition Inclusion
			• 02 = Variable or Unknown Frequency				
Results Message Status			• 04 = 3DS Server will process Decoupled Authentication in a subsequent authentication • 05–79 = Reserved for EMVCo future use (values invalid until defined by EMVCo)				
SDK Server Signed Content							Required if SDK Type = 02~~, 03, 04 or 05~~.
SDK Type		3DS ~~SDK~~Server	• ~~03 = Limited-SDK~~ • ~~04 = Browser-SDK~~ • ~~05 = Shell-SDK~~ • 0~~6~~3–79 = Reserved for EMVCo future use (values invalid until defined by EMVCo)				
Serial Number	*The following note was added at the end of the description:*					PRes = ~~O~~C	PRes: Absent if the Card Range Data File URL is present.

Data Element/ Field Name	Description	Source	Length/Format/ Values	Device Channel	Message Category	Message Inclusion	Condition Inclusion
	Note: Serial Number is not provided when the DS and the 3DS Server select the Card Range Data File download option.						
SPC Incompletion Indicator			• 03 = SPC timed out • 04–99 = Reserved for EMVCo future use (values invalid until defined by EMVCo)				

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

Split-SDK Type
Field Name: `splitSdkType`
	Indicates the characteristics of a Split-SDK.
	Split-SDK Variant: Implementation characteristics of the Split-SDK client
	Limited Split-SDK Indicator: If the Split-SDK client has limited capabilities
	Example:
	`"splitSdkType":{`
	`"sdkVariant":"01",`
	`"limitedInd":"Y"`
	`}`
		3DS Server	Length: Variable
			JSON Data Type: Object
			`sdkVariant`
			Length: 2 characters
			JSON Data Type: String
			Values accepted:
			• 01 = Native Client
			• 02 = Browser
			• 03 = Shell
			• 04–79 = Reserved for EMVCo future use (values invalid until defined by EMVCo)
			• 80–99 = Reserved for DS use
			`limitedInd`
			Length: 1 character
			JSON Data Type: String
			Value accepted:
			• Y = Limited
			Only present if value = Y
				01-APP	01-PA
					02-NPA
						AReq = C	Required if SDK Type = 02

Data Element/ Field Name	Description	Source	Length/Format/ Values	Device Channel	Message Category	Message Inclusion	Condition Inclusion
Transaction Status	*The following second note was added at the end of the description:* ~~Note:~~ The Final CRes message can only contain ~~only~~ a value of Y or N. Transaction Status = C or S is not allowed for Device Channel = 3RI.						
Transaction Status Reason			• 29 = Authentication attempted but not completed by the Cardholder. Fall back to Decoupled Authentication • 30 = Authentication completed successfully but additional authentication of the Cardholder required. Reinitiate as Decoupled Authentication • 31–79 = Reserved for EMVCo future use (values invalid until defined by EMVCo)				

Data Element/ Field Name	Description	Source	Length/Format/ Values	Device Channel	Message Category	Message Inclusion	Condition Inclusion
WebAuthn Credential List			JSON Data Type: Array of ~~String~~Objects ~~Base64url encoded~~ The object contains: • Relying Party ID Field Name: `rpID` Length: Variable, maximum 2048 characters • WebAuthn Credential Field Name: `credentialIds` Length: Variable, ~~String:~~ 16–1000 characters Base64url-encoded	02-BRW ~~03-3RI~~		ARes = ~~O~~C	Required when Transaction Status = S

A.8 Browser CReq and CRes POST

Table A.3: 3DS CReq/CRes POST Data

Data Element / Field Name	Description	Recipient	Length/Format/Values	Message Inclusion
3DS Requestor Session Data	The 3DS Requestor may provide the 3DS Requestor Session Data ~~in~~with the CReq message to the ACS. The ACS returns the 3DS Requestor session data ~~in~~with the CRes message POST to the 3DS Requestor.			~~CReq = O~~ ~~CRes = C. Required if present in the CReq message.~~ • O in HTML form with the CReq message • R in HTML form with the CRes message if received with the CReq message
CReq			Base64url-encoded	

Browser CReq - CRes Data Examples

- **Example 1:** `threeDSSessionData` sent by the 3DS Requestor in the CReq message to the ACS

```
3DS Requestor Session Data from the 3DS Requestor = "merchant.com-ID-adac2434-df78-4bfa-bcd9-11ca4ccd5dca"

3DS Requestor Session Data base64URL encoded =
"bWVyY2hhbnQuY29tLUlELWFkYWMyNDM0LWRmNzgtNGJmYS1iY2Q5LTExY2E0Y2NkNWRjYQ"
```

```
CReq message
{
      "threeDSServerTransID":"8a880dc0-d2d2-4067-bcb1-b08d1690b26e",
      "acsTransID":"d7c1ee99-9478-44a6-b1f2-391e29c6b340",
      "threeDSRequestorURL":"https://merchant.com/url",
      "messageType":"CReq",
      "messageVersion":"2.3.0"
}
CReq message base64URL encoded
"eyJ0aHJlZURTU2VydmVyVHJhbnNJRCI6IjhhODgwZGMwLWQyZDItNDA2Ny1iY2IxLWIwOGQxNjkwYjI2ZSIsCSJhY3NUcmFuc0lEIjoiZDdjMWVlOTktOTQ3OC00NGE2LWIxZjItMzkxZTI5YzZiMzQwIiwidGhyZWVEU1JlcXVlc3RvclVybCI6Imh0dHBzOi8vbWVyY2hhbnQuY29tL3VybCIsIm1lc3NhZ2VUeXBlIjoiQ1JlcSIsIm1lc3NhZ2VWZXJzaW9uIjoiMi4zLjAifQ"


HTML form
"htmlCreq": "<form action=\'https://acs.com.creq\' method=\'post\'>
<input type=\'hidden\' name=\'creq\' value=\
'eyJ0aHJlZURTU2VydmVyVHJhbnNJRCI6IjhhODgwZGMwLWQyZDItNDA2Ny1iY2IxLWIwOGQxNjkwYjI2ZSIsCSJhY3NUcmFuc0lEIjoiZDdjMWVlOTktOTQ3OC00NGE2LWIxZjItMzkxZTI5YzZiMzQwIiwidGhyZWVEU1JlcXVlc3RvclVybCI6Imh0dHBzOi8vbWVyY2hhbnQuY29tL3VybCIsIm1lc3NhZ2VUeXBlIjoiQ1JlcSIsIm1lc3NhZ2VWZXJzaW9uIjoiMi4zLjAifQ' />
<input type=\'hidden\' name=\'threeDSSessionData\' value=\'b\'
bWVyY2hhbnQuY29tLUlELWFkYWMyNDM0LWRmNzgtNGJmYS1iY2Q5LTExY2E0Y2NkNWRjYQ'\' /></form>"
```

- **Example 2:** `threeDSSessionData` sent by the ACS in the CRes message to the 3DS Requestor

```
Base64url decoded 3DS Requestor Session Data: "merchant.com-ID-adac2434-df78-4bfa-bcd9-11ca4ccd5dca"
```

```
CRes message
{
      "threeDSServerTransID":"8a880dc0-d2d2-4067-bcb1-b08d1690b26e",
      "acsTransID":"d7c1ee99-9478-44a6-b1f2-391e29c6b340",
      "transStatus":"Y",
      "messageType":"CRes",
      "messageVersion":"2.3.0
}


Base64 URL encoded CRes message
"eyJ0aHJlZURTU2VydmVyVHJhbnNJRCI6IjhhODgwZGMwLWQyZDItNDA2Ny1iY2IxLWIwOGQxNjkwYjI2ZSIsImFjc1RyYW5zSUQiOiJkN2M
xZWU5OS05NDc4LTQ0YTYtYjFmMi0zOTFlMjljNmIzNDAiLCJ0cmFuc1N0YXR1cyI6IlkiLCJtZXNzYWdlVHlwZSI6IkNSZXMiLCJtZXNzYWd
lVmVyc2lvbiI6IjIuMy4wIix9"


3DS Requestor Session Data base64URL encoded =
"bWVyY2hhbnQuY29tLUlELWFkYWMyNDM0LWRmNzgtNGJmYS1iY2Q5LTExY2E0Y2NkNWRjYQ"


HTML form
"htmlCres": "<form action=\'https://3dss.com.cres\' method=\'post\'>
<input type=\'hidden\' name=\'cres\'
value=\'eyJ0aHJlZURTU2VydmVyVHJhbnNJRCI6IjhhODgwZGMwLWQyZDItNDA2Ny1iY2IxLWIwOGQxNjkwYjI2ZSIsImFjc1RyYW5zSUQi
OiJkN2MxZWU5OS05NDc4LTQ0YTYtYjFmMi0zOTFlMjljNmIzNDAiLCJ0cmFuc1N0YXR1cyI6IlkiLCJtZXNzYWdlVHlwZSI6IkNSZXMiLCJt
ZXNzYWdlVmVyc2lvbiI6IjIuMy4wIix9'/>
<input type=\'hidden\' name=\'threeDSSessionData\' value=\'b\'
bWVyY2hhbnQuY29tLUlELWFkYWMyNDM0LWRmNzgtNGJmYS1iY2Q5LTExY2E0Y2NkNWRjYQ'\' /></form>"
```

A.9 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			The Transaction ID received was invalid. Invalid meaning Transaction ID not recognised~~, or Transaction ID is recognised as a duplicate~~.

A.11 Card Range Data

Previously placed at the end of Section A.11, the following Card Range Data Example has been moved, and it now directly follows Table A.6.

Card Range Data Example

```
{"cardRangeData": [
                    {"ranges": [
      {"start": "1000000000000000",
       "end": "1000000000005000"},
      {"start": "1000000000006000",
       "end": "1000000000007000"}
                      ],
                     "actionInd": "A",
                     "issuerCountryCode": "356",
                     "dsProtocolVersions": ["2.2.0", "2.3.0", "2.3.1"],
                     "acsProtocolVersions": [
      {"version": "2.2.0",
      "acsInfoInd": ["01", "02"],
      "threeDSMethodURL": "https://www.acs.com/script1","supportedMsgExt": [
           {"id": "A000000802-001","version": "2.0"},
           {"id": "A000000802-004","version": "1.0"}
      ]},
      {"version": "2.3.0",
      "acsInfoInd": ["01", "02", "03", "04", "80"],
      "threeDSMethodURL": "https://www.acs.com/script2"
      },
      {"version": "2.3.1",
      "acsInfoInd": ["01", "02", "03", "04", "81"],
      "threeDSMethodURL": "https://www.acs.com/script3"
      }
                        ]
                     }
                 ]
}
```

A.13 3DS Requestor Risk Information

A.13.1 Cardholder Account Information

Table A.10: Cardholder Account Information

Data Element/Field Name	Description	Length/Format/Values
Cardholder Account Requestor ID	This identifier is a coded as the SHA-256 + Base64url of the account identifier for the 3DS Requestor and is provided as a String.	
Cardholder Account Purchase Count	If the Cardholder Account Purchase Count reaches the value 999, it remains set at 999.	Values accepted: • 0–999
Number of Transactions Per Year	If the maximum value is reached, the Number of Transactions Per Year remains set at 999.	Values accepted: • 0–999

A.13.3 3DS Requestor Authentication Information

Table A.12: 3DS Requestor Authentication Information Object

Data Element/Field Name	Description	Length/Format/Values
3DS Requestor Authentication Data	For example, 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 Assertion and/or Attestation ~~or Assertion~~ Data. • 07, then this element can carry FIDO Assertion and/or Attestation ~~or Assertion~~ Data with the FIDO Assurance Data signed by a trusted third party. • 08, then this element can carry the SRC Assurance Data. For 3DS Requestor Authentication Method = 06 or 07, refer to EMV® 3-D Secure White Paper – Use of FIDO® Data in 3-D Secure Messages for the 3DS Requestor Authentication Data content and format.	JSON Data Type: String or Object
3DS Requestor Authentication Method	Note: For 09 = SPC Authentication, the Assertion Data is provided as a JSON object returned by the SPC API.	

A.13.4 3DS Requestor Prior Transaction Authentication Information

The 3DS Requestor Prior Authentication Information format is an array of object, the object contains the optional data elements as outlined in Table A.13.

Table A.13: 3DS Requestor Prior Transaction Authentication Information Object

Data Element/Field Name	Description	Length/Format/Values
3DS Requestor Prior DS Transaction ID Field Name: `threeDSReqPriorDsTransId`	This data element provides the prior DS Transaction ID to the ACS to determine the best approach for handling a request.	Length: 36 characters JSON Data Type: String Value accepted: • This data element contains a DS Transaction ID for a prior authenticated transaction (for example, the first recurring transaction that was authenticated with the Cardholder).

A.13.7 Challenge Data Entry

Table A.16: Challenge Data Entry

Challenge Data Entry	ACS UI Type	Challenge Cancelation Indicator	Resend Challenge Information Code	Challenge Additional Code	Challenge No Entry	Response
~~Missing~~	~~01, 02, or 03~~	~~Missing~~	~~Present~~ • ~~Value = N~~	~~Missing~~	~~Present~~ • ~~Value = Y~~	~~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.~~

The following note has been added directly below Table A.16:

Note: For all the combinations of Challenge Data Entry, Challenge Cancelation Indicator, Resend Challenge Information Code, Challenge Additional Code and Challenge No Entry not present in Table A.16, the ACS sends an Error Message with Error Code = 203 to the 3DS SDK.

A.13.8 Transaction Status Conditions

Table A.17: Transaction Status Conditions

Transaction Status	ARes	Final CRes	RReq	Error Response
D = Challenge Required; Decoupled Authentication confirmed	Valid[10]		~~Inv~~Valid[11]	• ~~RReq: Refer to Section 5.9.8 and use Error Code = 203~~
S = Challenge using SPC	Valid[13]	Invalid	Invalid	• ARes: Refer to Section 5.9.3 and use Error Code = 203 if Condition not met • Final CRes: End processing (no Error) • RReq: Refer to Section 5.9.8 and use Error Code = 203

Footnote 10: This indicator (D) can be sent only if 3DS Requestor Decoupled Request Indicator = Y or B within the AReq message.

Footnote 11: This indicator (D) can be sent only if 3DS Requestor Decoupled Request Indicator = F or B within the AReq message.

Footnote 13: This indicator (S) can be sent only if 3DS Requestor SPC Support = Y within the AReq message.

A.17 EMV Payment Token Information

Table A.25: Token Information

Data Element/Attribute Name	Description	Source	Length/Format/Values	Inclusion
Token Status Indicator			Length: Variable, maximum 40 characters	

A.20 Cardholder Information Text

Figures A.1 and A.2 have been replaced and are not replicated in this specification bulletin.

A.21 SPC Transaction Data

The field names of the SPC Transaction Data match the names used in the SPC API (refer to SPC API for additional information).

Table A.28: SPC Transaction Data

Data Element/Field Name	Description	Source	Length/Format/Values	Inclusion
Additional Data Field Name: `additionalData`	For SPC API enhancement, to be defined in a future 3DS specification release	ACS	Length: Variable, maximum 90000 characters JSON Data Type: Object	ARes = O
Challenge Field Name: `challenge`	Random string generated by the ACS to prevent replay attacks.	ACS	Length: Variable, 43–100 characters JSON Data Type: String Base64url-encoded Example: a random 32-byte value that has been Base64url-encoded gives a 43-character string.	ARes = R
Challenge Information Text Field Name: `challengeInfoText`	Text provided by the ACS to be displayed during the SPC authentication.	ACS	Length: Variable, maximum 350 characters JSON Data Type: String	ARes = C Required when supported by the SPC API
Currency Field Name: `currency`	Transaction amount currency to be displayed during the SPC authentication	ACS	Length: 3 characters JSON Data Type: String Values accepted: • ISO 4217 three-digit currency codes, other than those listed in Table A.5.	ARes = R

Data Element/Field Name	Description	Source	Length/Format/Values	Inclusion
Display Name Field Name: `displayName`	Card or product name (Payment Instrument) to be displayed during the SPC authentication.	ACS	Length: Variable, maximum 40 characters JSON Data Type: String	ARes = R
Icon Field Name: `icon`	Card image (Payment Instrument) URL or Data URL to be displayed during the SPC authentication.	ACS	Length: Variable, maximum 4096 characters JSON Data Type: String Values accepted: - Fully Qualified URL or Data URL in correct JSON Object format ○ Fully Qualified URL: Variable length, maximum 2048 characters ○ Data URL: Variable length, maximum 4096 characters. The Data URL embeds the image in Base64-encoded format.	ARes = R

Issuer Image SPC	Includes at minimum ~~one~~the Default Image and at maximum ~~of~~the three ~~f~~Fully ~~q~~Qualified URLs or ~~d~~Data URLs defined as ~~either;~~ default, dark mode or monochrome images of the Issuer Image SPC.
	Example Fully ~~q~~Qualified URL ~~example~~:
	"issuerImageSpc~~" :{~~":{
	        "default~~":~~
	":"https://acs.com/defaultspcimag
	e.png~~}~~"}
	Example Data URL ~~example~~:
	"issuerImageSpc~~" :{~~":{
		ACS		ARes = ~~R~~C
				Required when supported by the SPC API

Data Element/Field Name	Description	Source	Length/Format/Values	Inclusion

	`"default~~": "<img src~~`
	`~~"~~":"data:image/png;base64,iVBORw0KGgoAA..." ~~/>"]~~`			

Payee Name Field Name: `payeeName`	The display name of the payee that this SPC call is for (e.g. the Merchant). Matches the Merchant Name from the AReq message.	ACS	Length: Variable, maximum 40 characters JSON Data Type: String	ARes = C Required if Payee Origin is NOT present
Payee Origin Field Name: `payeeOrigin`	The origin of the payee that this SPC call is for (e.g. the Merchant). Matches the Payee Origin from the AReq message.	ACS	Length: Variable, maximum 2048 characters JSON Data Type: String Value accepted: • Fully Qualified URL	ARes = C Required if Payee Name is NOT present

Payment System Image SPC	Payment System logo or ~~i~~Image URLs to be displayed during the SPC authentication.
	Includes at minimum ~~one~~the Default Image and at maximum ~~of~~the three ~~f~~Fully ~~q~~Qualified URLs defined as ~~either;~~ default, dark mode or monochrome images of the ~~Issuer~~ Payment System Image SPC.
	Example Fully Qualified URL ~~Example~~:
	`"psImageSpc~~" :[~~":{`
	`"default~~":~~":"https://ds.com/defaultspcimage.png~~]:~~"}`
	Example Data URL ~~Example~~:
			○ Data URL: Variable length, maximum 30000 characters. The Data URL embeds the image in Base64url-encoded format.	ARes = ~~R~~C
				Required when supported by the SPC API

Data Element/Field Name	Description	Source	Length/Format/Values	Inclusion
	"psImageSpc":{ "default~~":"<img src "":"~~":"data:image/png;base64,c2RzYWRhc2Q..."~~ />"}~~			
Timeout Field Name: `timeout`	The number of milliseconds before the request to sign the transaction details times out.	ACS	Length: Variable, 5–6 characters JSON Data Type: String Value accepted: • Integer coded as a string in the range 60000–500000	ARes = R
Value Field Name: `value`	Transaction amount as a decimal value to be displayed during the SPC authentication.	ACS	Length: Variable, maximum 40 characters JSON Data Type: String	ARes = R
WebAuthn SPC Extension Indicator Field Name: `extInd`	For SPC and WebAuthn API enhancement.	ACS	Length: 1 character JSON Data Type: String Value accepted: • Y = Extension requested Only present if value = Y	ARes = O
~~Merchant SPC Name~~ ~~Field Name: `merchantSpcName`~~	~~Merchant name to be displayed during the SPC authentication~~	~~ACS~~	~~Length: Variable, maximum 40 characters~~ ~~JSON Data Type: String~~	~~ARes = R~~
~~Amount SPC~~ ~~Field name: `amountSpc`~~	~~Transaction amount as a decimal value to be displayed during the SPC authentication~~	~~ACS~~	~~Length: Variable, maximum 40 characters~~ ~~JSON Data Type: String~~	~~ARes = R~~

Data Element/Field Name	Description	Source	Length/Format/Values	Inclusion
~~Amount Currency SPC~~ ~~Field Name: `amountCurrencySpc`~~	~~Transaction amount currency to be displayed during the SPC authentication~~	~~ACS~~	~~Length: 3 characters~~ ~~JSON Data Type: String~~ ~~Format represented ISO 4217 in alphabetic code~~	~~ARes = R~~
~~Card Art~~ ~~Field Name: `cardArt`~~	~~Card image URL or Data URL to be displayed during the SPC authentication~~	~~ACS~~	~~Length: Variable, maximum 4096 characters~~ ~~JSON Data Type: String~~ ~~Value accepted:~~ • ~~Fully Qualified URL or Data URL in correct JSON Object format~~ ~~Fully qualified URL: Variable length, maximum 2048 characters~~ ~~Data URL: Variable length, maximum 4096 characters. The Data URL embeds the image in Base 64 encoded format.~~	~~ARes = R~~
~~Card Art Name~~ ~~Field Name: `cardArtName`~~	~~Card -product name to be displayed during the SPC authentication.~~	~~ACS~~	~~Length: Variable, maximum 40 characters~~ ~~JSON Data Type: String~~	~~ARes = R~~
~~SPC Unique Random Number~~ ~~Field Name: `spcUniqueRandomNumber`~~	~~Universally unique number assigned by the ACS used by the 3DS Requestor to perform SPC authentication.~~	~~ACS~~	~~Length: 36 characters~~ ~~JSON Data Type: String~~ ~~Canonical format as defined in IETF RFC 4122. This may utilise any of the specified versions if the output meets specified requirements.~~	~~ARes = R~~

A.22 HTTP Headers

The new Table A.29 HTTP Headers and the HTTP Header Examples, added in Section A.22, are not replicated in this bulletin. Please refer to the underlying specification to review the new section.

Annex B Message Format

B.1 AReq Message Data Elements

Table B.1 AReq Data Elements

Data Element	Field Name
App IP Address	`appIp`
Browser User Device ID	`deviceId`
Browser User ID	`userId`
Default-SDK Type	`defaultSdkType`
Payee Origin	`payeeOrigin`
Recurring Currency Exponent	recurring~~Currency~~Exponent
Split-SDK Type	`splitSdkType`
Tax ID	`taxId`

B.2 ARes Message Data Elements

Table B.2 ARes Data Elements

Data Element	Field Name
3DS Requestor App URL Indicator	`threeDSRequestorAppURLInd`

B.3 CReq Message Data Elements

Table B.3 CReq Data Elements

Data Element	Field Name
OOB App URL Indicator	`oobAppURLInd`

B.6 PReq Message Data Elements

Table B.6: PReq Data Elements

Data Element	Field Name
Card Range Data Download Indicator	`cardRangeDataDownloadInd`

B.7 PRes Message Data Elements

Table B.7 PRes Data Elements

Data Element	Field Name
Card Range Data File URL	`cardRangeDataFileURL`

B.8 RReq Message Data Elements

Table B.8 RReq Data Elements

Data Element	Field Name
Trust List Status Source	`trustListStatusSource`

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