National ITMX Co., Ltd.

ITMX Local Switching Secure EMVCo Authentication System

Member Banks
Technical Specifications

Release Date: April 26, 2021
Version 1.2

Prepared by

DOCUMENT REVISION HISTORY

VERSION	AUTHOR	DATE	CHANGE DESCRIPTION
0.1	ITMX	Apr 30, 2020	Draft Version
1.0	ITMX	June 19, 2020	Initial Version
1.1	ITMX	Mar 25, 2021	• Revise Wording ITMX LSS to ITMX LSS EMVCo • Update ITMX not support Attempts Server • Revise Wording in 1.3.1 Frictionless Transaction Flow and 1.3.2 Challenge Transaction Flow • Revise Wording in 2.5.1 CReq Message Timeouts on the ACS, 2.5.2 READ TIMEOUT and 2.5.3 MESSAGE ERROR HANDING • Add 3.10 ERROR MESSAGE DATA ELEMENT INFORMATION • Update 5.3 AUTHENTICATION AND AUTHORIZATION flow and table (ECI=06) • Update 5.5 PSEUDO CODE FOR E-COMMERCE by Removing case ECI 81
1.2	ITMX	April 26, 2021	• Update 5.3 AUTHENTICATION AND AUTHORIZATION flow and table (ECI=06) and LSS EMVCo Transaction Status Reason Code

TABLE OF CONTENTS

Chapter 1

1. INTRODUCTION

1.1 OVERVIEW

ITMX has launched an authentication protocol known as "Local Switch Secure (LSS) protocol" to manage the local switching process in collaboration with local banks and authenticate local card's online e-commerce transactions which are acquired by local acquirers.

ITMX LSS EMVCo is a new e-commerce authentication protocol that enables the secure processing of payment, non-payment and account confirmation card transactions. To promotes frictionless customer authentication and enables consumers to authenticate themselves with their card issuer when making card-not-present (CNP) e-commerce purchase.

ITMX LSS EMVCo is **one of EMV 3-D Secure components** as an Interoperability domain or Directory Server that 3-D Secure transactions are switched between the **Acquirer Domain and Issuer Domain**. 3-D Secure transactions are initiated from the Acquirer Domain and are authenticated in the Issuer Domain.

Merchant/Acquirer Domain

- **Merchant's E-Commerce Checkout Environment**

Merchants can invoke ITMX LSS EMVCo for consumers shopping on the Merchant's e-commerce websites using a browser or shopping via the merchant's e-commerce application on a device (e.g., mobile application or mobile browser).

- **Authentication Components**

A merchant's authentication components enable the merchant to send and receive LSS messages and include:

- 3DS Server software enables the merchant to use 3DS authentication with consumers using browser-based devices
- 3DS Software Development Kit (SDK) software enables the merchant to use 3DS with consumers using device-based applications (e.g., a mobile application)

- **Authorization Components**

A merchant's authorization components enable the merchant to interface with ITMX Local Switching (LSW) t to process payment transactions.

Issuer Domain

- **Authentication Components**

Issuer ACS—The Issuer ACS responds to authentication requests and performs cardholder authentication.

- The issuer defines the authentication methods and the criteria that the ACS uses to determine what type of authentication is needed.
- The Issuer's ACS creates the LSAV and provides it to the merchant in the authentication response.

The merchant includes the LSAV in the authorization request message to the issuer as proof that authentication was performed.

- **Authorization Components**

Issuer Host—The Issuer Host system communicates with ITMX Local Switching (LSW) to process payment transactions.

- The Issuer Host receives the ITMX authorization message and uses ITMX LSS Authentication Data included in the transaction to approve or decline the purchase.
- The LSS Authentication Data in the authorization message includes the LSAV. The issuer can use the LSAV to support the authorization decision process.
- ITMX requires the Issuer verify the LSAV sent in the authorization request by the Merchant/Acquirer. ITMX offers a service where LSW can perform LSAV verification on behalf of the Issuer.

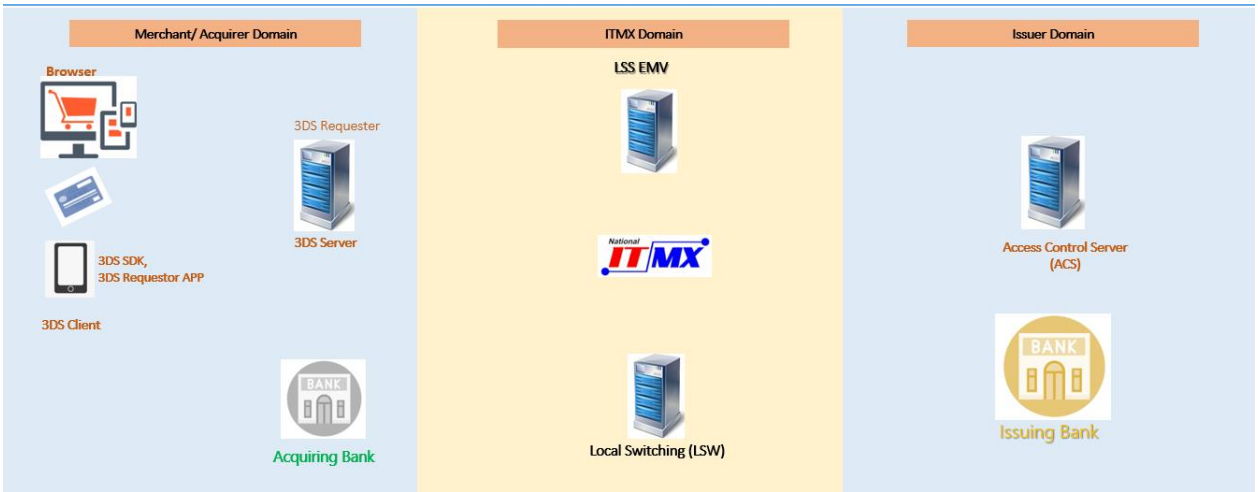
ITMX Domain

ITMX operates the LSS Server and the ITMX LSS EMVCo Attempts Service.

- LSS EMVCo Directory Server—The ITMX Directory Server routes EMV 3DS messages between Merchant 3DS Servers and the Issuer ACS or Attempts Server (Not support yet).
- LSS Attempts Service—Is a ITMX service that responds to authentication request messages on behalf of the Issuer when either the Issuer does not participate in LSS EMVCo or the Issuer participates but their ACS is unavailable. The ITMX LSS EMVCo Attempts Service provides proof, in the form of a LSAV, in the authentication response that the merchant attempted to obtain authentication. (Not support yet)

ITMX's authorization component is LSW and LSW's LSAV verification service.

- LSW—Is a collection of systems and services where ITMX offers online financial processing, authorization, clearing, and settlement services to issuers and acquirers.
- LSAV Verification— LSW can perform LSAV verification on behalf of the issuer during authorization processing. The issuer must provide their LSAV keys to ITMX LSW to participate in these services.



Domains and Components

1.2 REQUIREMENT

LSS EMVCo authentication process comprises the following messages for 3-D Secure to authenticate local card's holder's authenticity when they make e-commerce transactions through LSS local acquirer members.

Required LSS EMV message:

- Authentication Request Message (AReq)
- Authentication Response Message (ARes)
- Challenge Request Message (CReq)
- Challenge Response Message (CRes)
- Results Request Message (RReq)
- Results Response Message (RRes)
- Preparation Request Message (PReq)
- Preparation Response Message (PRes)
- Error Message

Refer to 2. Process Description for detailed information about message layouts.

1.3 TRANSACTION FLOW

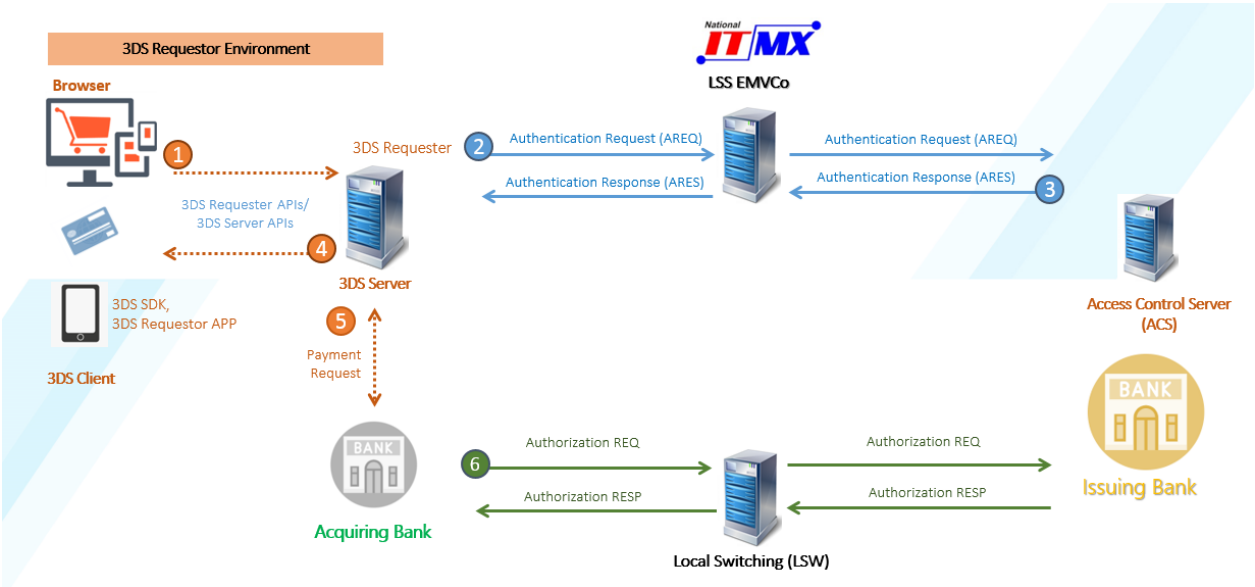
ITMX LSS EMVCo supports two primary authentication flows:

- Frictionless Flow
- Challenge Flow

1.3.1 FRICTIONLESS FLOW

A Frictionless Flow occurs when the Issuer authenticates the cardholder without cardholder involvement by evaluating the transaction's risk level.

- The Merchant environment sends an authentication request message to the Issuer with all the required data to facilitate risk-based authentication.
- The Issuer receives the authentication request message, and uses the data to evaluate the transaction's risk. The Issuer returns an authentication response, and no additional cardholder verification is required.
- The merchant includes the Authentication Data received from the authentication process in the authorization message.



Frictionless Transaction Flow

The Frictionless Flow comprises the following Steps:
Start: Cardholder—Cardholder initiates a transaction on a Consumer Device. The Cardholder provides the information necessary for the authentication (Cardholder entry or already on file with the Merchant).

1. **3DS Requestor Environment**—Within the 3DS Requestor Environment, the necessary 3-D Secure information is gathered and provided to the 3DS Server for inclusion in the AReq message.

How information is provided, and from which component, depends on the following:
• Device Channel—App-based or Browser-based
• Message Category—Payment or Non-Payment
• 3DS Requestor 3-D Secure implementation

2. **3DS Server through ITMX LSS EMVCo to ACS**—Using the information provided by the Cardholder and data gathered within the 3DS Requestor Environment, the 3DS Server creates and sends an AReq message to the **ITMX LSS EMVCo** , which then forwards the message to the appropriate ACS

3. **ACS through ITMX LSS EMVCo to 3DS Server**—In response to the AReq message, the ACS returns an ARes message to the **ITMX LSS EMVCo**, which then forwards the message to the initiating 3DS Server. Before returning the response, the ACS evaluates the data provided in the AReq message. In a Frictionless Flow, the ACS determines that further Cardholder interaction is not required to complete the authentication.

4. **3DS Requestor Environment**—The 3DS Server communicates the result of the ARes message to the 3DS Requestor Environment which then informs the Cardholder.

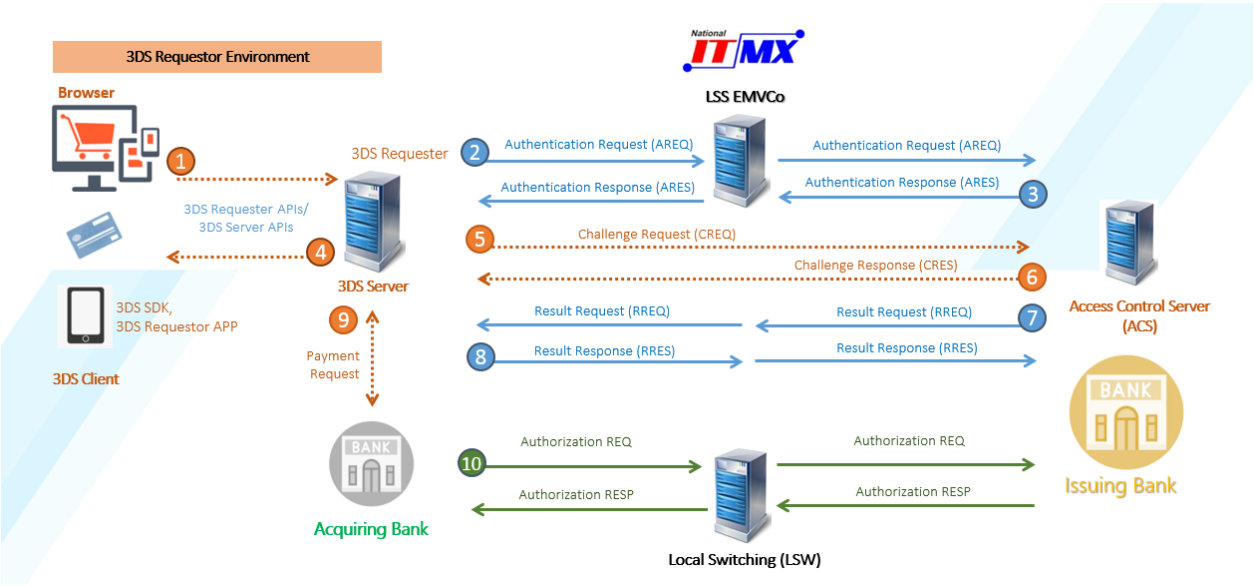
5. **Merchant and Acquirer**—The Merchant proceeds with authorization exchange with its Acquirer. If appropriate, the Merchant, Acquirer, or Payment Processor can submit a standard authorization request.

6. **Payment Authorization**—The Acquirer can process an authorization with the Issuer through the **ITMX Local Switching** system and return the authorization results to the Merchant.

1.3.2 CHALLENGE FLOW

A Challenge Flow occurs where the issuer interacts with the cardholder to complete authentication. The Challenge Flow occurs when the issuer assesses the risk of the transaction during the frictionless flow and determines that the transaction requires additional cardholder authentication; the frictionless flow transitions into the challenge flow:

- The issuer communicates to the merchant environment that additional authentication is required to verify the identity of the cardholder.
- The merchant environment sends the issuer a challenge request message.
- The issuer authenticates the cardholder using their chosen challenge method such as through a one-time passcode.
- Assuming authentication is successful, the transaction proceeds to authorization and the merchant includes key data elements from the authentication process in the authorization message.



Challenge Transaction Flow

The Challenge Flow comprises the following Steps:
Start: Cardholder—Same as the Frictionless Flow.

1. **3DS Requestor Environment**—Same as the Frictionless Flow

2. **3DS Server through ITMX LSS EMVCo DS to ACS**—Same as the Frictionless Flow

3. **ACS through ITMX LSS EMVCo DS to 3DS Server**—Same as the Frictionless Flow except that the ARes message indicates that further Cardholder interaction is required to complete the authentication.

4. **3DS Server to 3DS Requestor Environment**—Same as the Frictionless Flow except that further Cardholder interaction is required to complete the authentication

5. **3DS Client to ACS**—The 3DS Client initiates a CReq message based on information received in the ARes message. The manner in which this is done depends on the model:

   • **App-based**—A CReq message is formed by the 3DS SDK and is posted to the ACS URL received from the ARes message.

   • **Browser-based**—A CReq message is formed by the 3DS Server and is posted through the Cardholder's browser by the 3DS Requestor to the ACS URL received from the ARes message.

6. **ACS to 3DS Client**—The ACS receives the CReq message and interfaces with the 3DS Client to facilitate Cardholder interaction. The manner in which this is done depends on the model:

   • **App-based**—The ACS utilises pairs of CReq and CRes messages to perform the challenge. In response to the CReq message, the CRes message requesting the Cardholder to enter the authentication data is formed by the ACS and sent to the 3DS SDK.

• **Browser-based**—The ACS sends the authentication user interface to the Cardholder browser. The Cardholder enters the authentication data via the browser to be checked by the ACS. In response to the CReq message, the CRes message is formed by the ACS and sent to the 3DS Server to indicate the result of the authentication.
Note: For the App-based model, Step 5 and Step 6 will be repeated until the ACS makes a determination.

Note: For Decoupled Authentication, instead of utilising the CReq and CRes messages, the ACS authenticates the cardholder outside of the EMV 3-D Secure protocol.

7. **ACS through ITMX LSS EMVCo DS to 3DS Server**—The ACS sends an RReq message that can include the Authentication Value (AV) to the DS, which then routes the message to the appropriate 3DS Server using the 3DS Server URL received from the AReq message.

8. **3DS Server through ITMX LSS EMVCo DS to ACS**—The 3DS Server receives an RReq message and in response, returns an RRes message to the DS, which then routes the message to the ACS.

9. **Merchant** and **Acquirer**—Same as the Frictionless Flow.

10. **Payment Authorization**—Same as the Frictionless Flow.

EMV 3DS Message Type	Frictionless Flow	Challenge Flow	Operations
Authentication Request/Response	✓	✓	
Challenge Request/Response		✓	
Results Request/Response		✓	
Preparation Request/Response			✓
Error Message			✓

Chapter 2

2. PROCESS DESCRIPTION

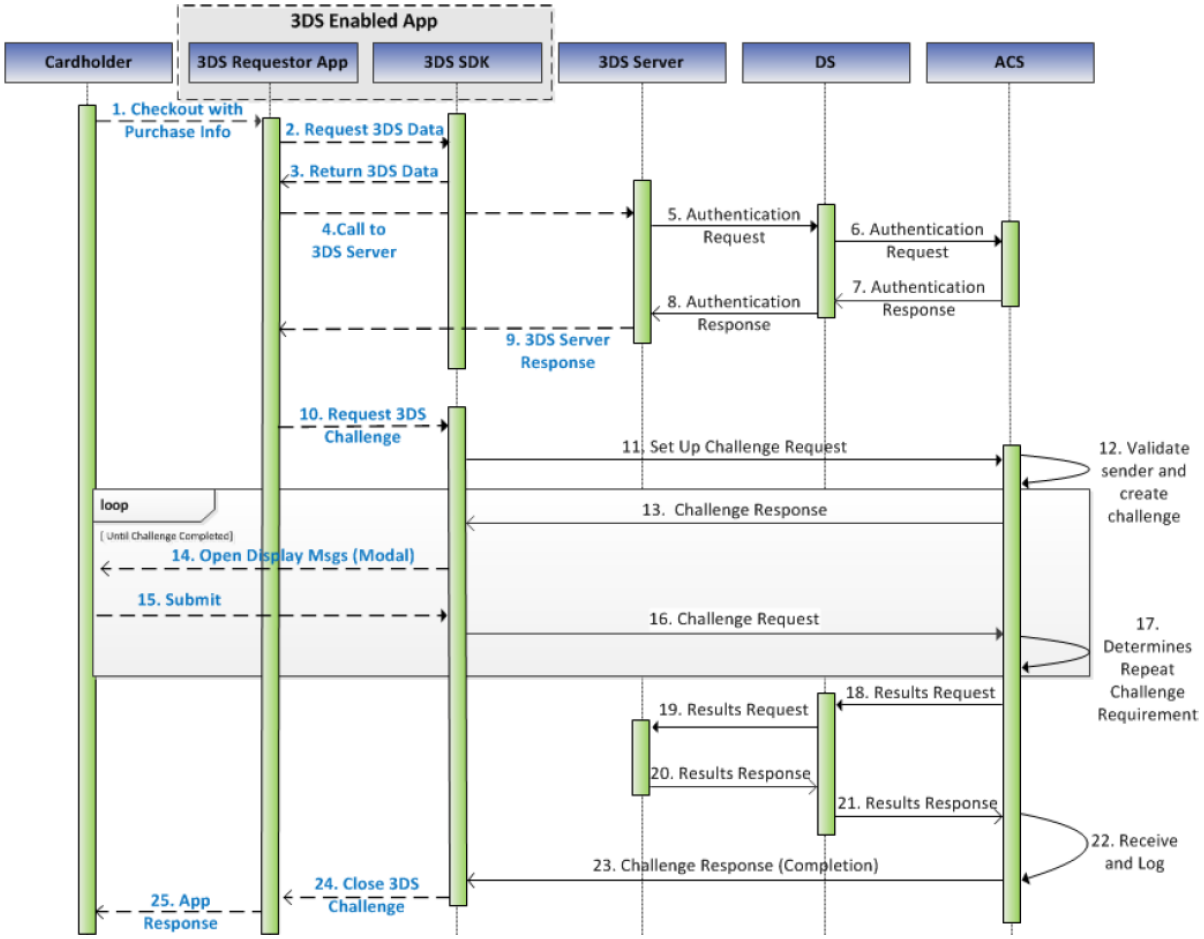
The EMV 3-D Secure processing flow for LSS EMVCo have 3 authentication flows. There are App-based, Browser-based and 3RI-based processing flow. For clarity, the actions for all components involved in the 3-D Secure authentication process are described:

- 3DS Client
- 3DS SDK
- 3DS Method
- 3DS Server
- **DS is ITMX LSS EMVCo**
- Access Control Server (ACS)

2.1 APP-BASED PROCESSING FLOW

In an App-based model, the communication flows between the 3DS SDK/3DS Requestor App and the 3DS Server/3DS Requestor using the APIs made available from the 3DS Server/3DS Requestor.

Start: Cardholder and 3DS Requestor App—The Cardholder initiates a transaction using a 3DS Requestor App on a Consumer Device.



Note. Dashed arrows are not part of the 3-D Secure 2.0 protocol but are shown for clarity. This flow shows the CReq/Res message exchange if there is need to retry or request additional inform from the cardholder

Step 1: The Cardholder
The Cardholder interacts with the 3DS Requestor using the 3DS Requestor App and confirms the applicable business logic. For example, the Cardholder makes an e-commerce purchase using a 3DS Requestor App on a Consumer Device

Step 2: The 3DS Requestor App
Depending on the 3DS Requestor Environment, additional information may be obtained. For example, payment and shopping cart information for Payment Authentication.
The 3DS Requestor App uses the Cardholder Account Number and optionally other cardholder information to identify the Payment System. Payment Systems are identified by their ISO RID

Step 3: The 3DS SDK
The 3DS SDK returns data required for the AReq message

Step 4: The 3DS Requestor Environment
The 3DS Requestor Environment is responsible for gathering the information for the AReq message assembled by the 3DS Server

Step 5: The 3DS Server
Verify the authenticity of the 3DS SDK. Generate the 3DS Server Transaction ID. Obtain the 3DS Requestor ID, the 3DS Server Reference Number, and conditionally the Acquirer BIN. Determine which DS the authentication transaction needs to be sent based on the BIN and optionally other Cardholder account information. Establish a

secure link with the DS. Format the AReq message as defined in Section 3.1. Send the AReq message to the DS using the secured link

Note: The 3DS Server can use the ACS Start Protocol Version, ACS End Protocol Version, DS Start Protocol Version and DS End Protocol Version obtained from the PRes message to verify that the ACS and DS support the protocol version used by the 3DS Server

Step 6: The DS
Receive the AReq message from the 3DS Server and Validate. Generate the DS Transaction ID. Decrypt the SDK Encrypted Data data element of the AReq message and Base64url encode resulting content and move the encoded content to the Device Information data element of the AReq message for the ACS. Check that the Message Version Number is supported by the DS and the ACS. Check the data elements in the AReq message. Determine if the Cardholder Account Number received in the AReq message is in a participating account range. Determine if the Cardholder Account Number is in an account range that has an ACS capable of processing 3-D Secure messages. Store the 3DS Server URL with the DS Transaction ID (for possible RReq processing). Establish a secure link with the ACS. The DS may maintain multiple ACS URLs. If the first URL attempted is not available, then the DS will attempt to connect to one of the alternate URLs. Send the AReq message to the ACS using the secured link.

Step 7: The ACS
The ACS receive the AReq message from the DS and Validate. Check whether the Consumer Device is supported. The ACS uses the Device Information received in the AReq message to recognise the device, assess transaction risk, and determine if it can complete the authentication with this device. The ACS shall not initiate an interaction with the Cardholder as part of a Frictionless transaction. Cardholder interaction shall be done as part of a Challenge flow. Generate the ACS Transaction ID. Use the Cardholder Account Number from the AReq message to determine whether authentication is available or can be completed for the Cardholder.

Evaluate the values received in the AReq message and determine whether the transaction1is:
- authenticated (Transaction Status = Y)
- requiring a Cardholder challenge to complete authentication (Transaction Status = C)
- not authenticated (Transaction Status = N)
- not authenticated, but a proof of authentication attempt (Authentication Value) was generated (Transaction Status = A)
- not authenticated, as authentication could not be performed due to technical or other issue (Transaction Status = U)
- not authenticated because the Issuer is rejecting authentication and requesting that authorization not be attempted (Transaction Status = R)

If a transaction is deemed authenticated (Transaction Status = Y or A) the ACS performs the following:
- For a Payment Authentication (Message Category = 01-PA), the ECI value and Authentication Value shall be generated and included in the ARes message as defined by the DS.
- For a Non-Payment Authentication (Message Category = 02-NPA), the ACS *may*:
    - Generate the ECI value and Authentication Value and include in the ARes message as defined by the DS.
    - Assign an appropriate Transaction Status Reason Code value as defined by the specific DS and include in the ARes message.

Note: Whether the ECI Indicator/Authentication Value and/or the Transaction Status Reason value is included in the ARes message is defined by the DS.

If a challenge is deemed necessary (Transaction Status = C), the ACS determines whether an acceptable challenge method is supported by the 3DS SDK based in part on the following data elements received in the AReq message: Device Channel, Device Rendering Options Supported, and SDK Maximum Timeout.
Then Complete formatting of the ARes message and end the ARes message to the DS using the secure link

Step 8: The DS
Receive the ARes message or Error message from the ACS. Check the data elements in the ARes message. Log transaction information as required by the DS rules. Send the ARes message to the 3DS Server as received from the ACS using the secure link

Step 9: The 3DS Server
Receive the ARes message or Error Message from the DS and Validate

For an authenticated transaction (Transaction Status = Y or A):
For Payment Authentication (Message Category = 01-PA), ensure that the Transaction Status, ECI value, and Authentication Value as generated by the ACS are provided for the authorization process with ITMX Local Switching service.

For a transaction with a challenge (Transaction Status = C):
Evaluate based in part on the 3DS Requestor Challenge Indicator, the ACS Challenge Mandated Indicator and the ACS Rendering Type whether to perform the requested challenge.

For a transaction not authenticated (Transaction Status = N, U, or R)
Send necessary information from the ARes message to the 3DS Requestor Environment.

Notes: For a Frictionless Flow, the next step is conveying the appropriate response to the 3DS Requestor App.
Step 10 through Step 23 and the first requirements of Step 24 are applicable only for a Challenge Flow.

Step 10 The 3DS Requestor App
The 3DS Requestor Environment receives the necessary ARes data elements from the 3DS Server and makes the data elements available to the 3DS SDK for execution of a Challenge Flow.
The 3DS Requestor App Invokes the "doChallenge method" by making a call to the 3DS SDK. Refer to the *EMV 3-D Secure—SDK Specification* for additional information about this method.

Step 11: The 3DS Requestor Environment
The 3DS SDK check the received 3-D Secure data elements. Complete the ACS to 3DS SDK secure channel. Establish a secure link to the ACS. Format the CReq message as defined in Section 3.3.
Send the CReq message to the ACS using the secure link**.**

Step 12: The ACS
The ACS receive the CReq message or Error Message from the 3DS SDK and Validate as defined. Set the Interaction Counter to zero. Set the Challenge Completion Indicator = N. Obtain the information needed to display a Challenge on the Consumer Device per the selected challenge method and ACS UI Type.

Step 13: The ACS
The ACS format the CRes message. Protect the content in the CRes message. Send the CRes message to the 3DS SDK through the secure link. for an initial interaction with the 3DS SDK or for a continued interaction with the 3DS SDK.

Step 14: The 3DS SDK
The 3DS SDK receive the CRes message or Error Message from the ACS and Validate. Display the UI based upon the ACS UI Type selected and the data elements populated.

Step 15: The Cardholder Interaction with the 3DS SDK
The Cardholder interacts with the UI (for example, enters the data and presses Submit).

Step 16: The 3DS SDK
The 3DS SDK establish a secure link to the ACS and format the CReq message. Send the CReq message to the ACS. If the Cardholder abandons the challenge during the processing of Step 12 through Step 15 the 3DS SDK sets the Challenge Cancelation Indicator to the appropriate value in the CReq message and sends the CReq message to the ACS.

Step 17: The ACS
The ACS receive the CReq message or Error Message from the 3DS SDK and Validate. If Challenge Cancelation Indicator has a value, then continue with Step 18.

Check the authentication data entered by the Cardholder:
• 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
- Sets the Challenge Completion Indicator = Y
- Continues with Step 18

• If incorrect and authentication has failed, then the ACS:
  - Increments the Interaction Counter and compares it to the ACS maximum challenges.
  - If the Interaction Counter ≥ ACS maximum challenges 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

• If the Interaction Counter < ACS maximum challenges 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
The ACS format the RReq message. Establish a secure link with the DS. If the Cardholder abandons the challenge during the processing of Step 16 and Step 17, or if the ACS receives an abandonment CReq message from the 3DS SDK**.** Then the ACS sets the Challenge Completion Indicator = Y in the CRes message and sets the Challenge Cancellation Indicator to the appropriate value in the RReq message. Send the RReq message to the DS. Ensure that one RReq message is sent to the DS for each ARes message with a Transaction Status = C.

Step 19: The DS
Receive the RReq message from the ACS and Validate**.** Establish a secure link with the 3DS Server using the 3DS Server URL extracted from the AReq message and Store the 3DS Server URL with the DS Transaction ID. Send the RReq message to the 3DS Server.

Step 20: The 3DS Server
Receive the RReq message or Error Message from the DS and Validate. Format the RRes message in Section 3.5 and send to the DS.

Note: For Payment Authentication, the Merchant can now proceed with Authorization processing with its Acquirer. However, the Merchant may first want to receive confirmation that the Cardholder has not abandoned the transaction.

Step 21: The DS
Receive the RRes message or Error Message from the 3DS Server and Validate. Log transaction information as required by the DS. Send the RRes message to the ACS as received from the 3DS Server

Step 22: The ACS
Receive the RRes message or Error Message from the DS and Validate.

Step 23: The ACS
Format the final CRes message and protect the contents. Send the final CRes message to the 3DS SDK.

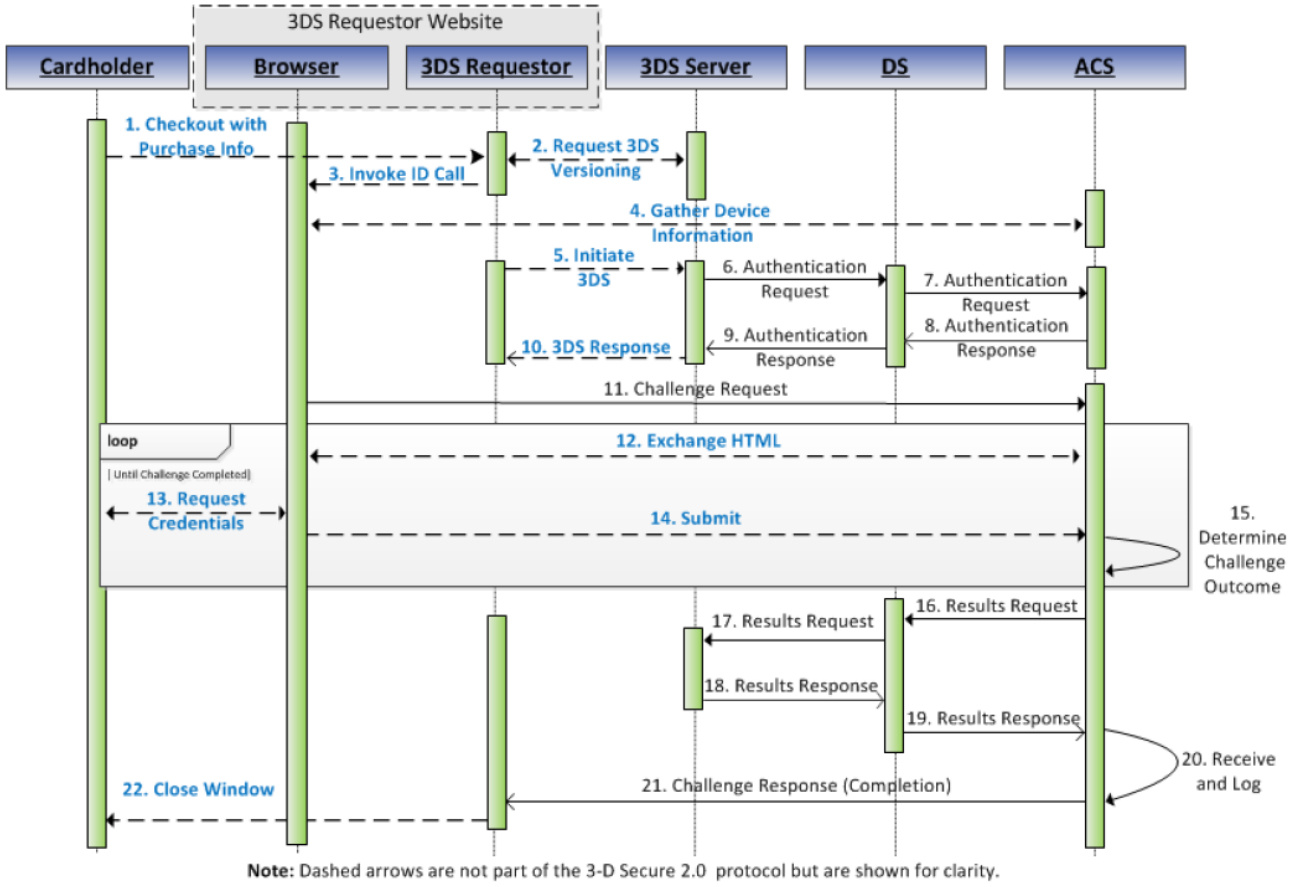
Step 24: The 3DS Requestor Environment
Receive the final CRes message or Error Message from the ACS and Validate. Convey the appropriate response to the 3DS Requestor App. Then 3-D Secure processing completes.

Step 25: The 3DS Requestor App
The 3DS Requestor App displays the appropriate result to the Cardholder.

2.2 BROWSER-BASED PROCESSING FLOW

In a Browser-based model, the communication flows between the Consumer Device and the 3DS Server/3DS Requestor using a TLS browser connection.

Start: Cardholder—The Cardholder initiates the transaction using a browser on a Consumer Device using a website operated by the 3DS Requestor.



The 3-D Secure processing flow for Browser-based implementations contains the following Steps:

Step 1: The Cardholder
The Cardholder interacts with the 3DS Requestor using a browser on a Consumer Device and confirms the applicable business logic. For example, the Cardholder makes an e-commerce purchase on a Merchant website using a Consumer Device.

Step 2: 3DS Server/3DS Requestor
The 3DS Requestor initiates communications with the 3DS Server and provides the necessary 3-D Secure information to the 3DS Server to initiate Cardholder authentication.
The 3DS Requestor uses the Cardholder Account Number and optionally other cardholder information to request the ACS Start Protocol Version, ACS End Protocol Version, DS Start Protocol Version and DS End Protocol Version and if present, the 3DS Method URL for that BIN range from the 3DS Server.

The 3DS Server:
Retrieve the ACS Start Protocol Version and ACS End Protocol Version, DS Start Protocol Version and DS End Protocol Version and, if present, the 3DS Method URL (stored from a previously received PRes message) for that BIN range. Generate the 3DS Server Transaction ID.

Pass the 3DS Server Transaction ID, ACS Start Protocol Version, ACS End Protocol Version, DS Start Protocol Version, DS End Protocol Version and if present, the 3DS Method URL back through the 3DS Requestor Environment to the 3DS Requestor.

If the DS Start Protocol Version and DS End Protocol Version are not present for the BIN range, then the default values for the DS Start Protocol Version and DS End Protocol Version located in the PRes message shall be utilised

Step 3: The 3DS Requestor Environment
For each transaction ensure that the 3DS Server Transaction ID used in the 3DS Method on the 3DS Requestor website is the same 3DS Server Transaction ID used in the AReq message. Ensure the 3DS Method is executed on the 3DS Requestor website if a 3DS Method URL exists for this transaction.

Step 4: Browser and the ACS
The browser connects to the ACS using a secure link. Ensure that the communication between the Browser and the ACS is established using a server authenticated TLS session

Step 5: The 3DS Requestor Environment
The 3DS Requestor Environment is responsible for gathering the information for the AReq message assembled by the 3DS Server.

Ensure that the communication between client (Browser) and server (3DS Requestor) has been established using a server authenticated TLS

Step 6: The 3DS Server
Obtain the 3DS Requestor ID, the 3DS Server Reference Number, and conditionally the Acquirer BIN.
Ensure availability of the necessary information for the AReq message gathered by components within the 3DS Requestor Environment.

Determine which DS the authentication transaction needs to be sent based on the BIN and optionally other Cardholder account information. Establish a secure link with the DS. Format the AReq message and Send the AReq message to the DS.

Step 7: The DS
Receive the AReq message from the 3DS Server and Validate. Generate the DS Transaction ID. Check that the Message Version Number is supported by the DS and the ACS. Check the data elements in the AReq message. Determine if the Cardholder Account Number received in the AReq message is in a participating account range. Determine if the Cardholder Account Number is in an account range that has an ACS capable of processing 3-D Secure messages. Store the 3DS Server URL with the DS Transaction ID (for possible RReq processing). Establish a secure link with the ACS. The DS may maintain multiple ACS URLs. If the first URL attempted is not available, then the DS will attempt to connect to one of the alternate URLs. Send the AReq to the ACS using the secured link**.**

Step 8: The ACS
The ACS receive the AReq message from the DS and Validate. Check whether the Consumer Device is supported. The ACS shall not initiate an interaction with the Cardholder as part of a Frictionless transaction. Cardholder interaction shall be done as part of a Challenge flow. Generate the ACS Transaction ID. Use the Cardholder Account Number from the AReq message to determine whether authentication is available or can be completed for the Cardholder.

Evaluate the values received in the AReq message and determine whether the transaction1is:
- authenticated (Transaction Status = Y)
- requiring a Cardholder challenge to complete authentication (Transaction Status = C)
- not authenticated (Transaction Status = N)
- not authenticated, but a proof of authentication attempt (Authentication Value) was generated (Transaction Status = A)
- not authenticated, as authentication could not be performed due to technical or other issue (Transaction Status = U)
- not authenticated because the Issuer is rejecting authentication and requesting that authorisation not be attempted (Transaction Status = R)

If a transaction is deemed authenticated (Transaction Status = Y or A) the ACS performs the following:

- For a Payment Authentication (Message Category = 01-PA), the ECI value and Authentication Value shall be generated and included in the ARes message as defined by the DS.
- For a Non-Payment Authentication (Message Category = 02-NPA), the ACS *may*:
    - Generate the ECI value and Authentication Value and include in the ARes message as defined by the DS.
    - Assign an appropriate Transaction Status Reason Code value as defined by the specific DS and include in the ARes message.

Note: Whether the ECI Indicator/Authentication Value and/or the Transaction Status Reason value is included in the ARes message is defined by the DS.

If a challenge is deemed necessary (Transaction Status = C), the ACS determines whether an acceptable challenge method is supported by the 3DS SDK based in part on the following data elements received in the AReq message: Device Channel, Device Rendering Options Supported, and SDK Maximum Timeout.
Then Complete formatting of the ARes message and end the ARes message to the DS using the secure link

Step 9: The DS
Receive the ARes message or Error message from the ACS . Check the data elements in the ARes message. Log transaction information as required by the DS rules. Send the ARes message to the 3DS Server as received from the ACS using the secure link

Step 10: The 3DS Server
Receive the ARes message or Error Message from the DS and Validate

For an authenticated transaction (Transaction Status = Y or A):
For Payment Authentication (Message Category = 01-PA), ensure that the Transaction Status, ECI value, and Authentication Value as generated by the ACS are provided for the authorization process with ITMX Local Switching service.

For a transaction with a challenge (Transaction Status = C):
Evaluate based in part on the 3DS Requestor Challenge Indicator, the ACS Challenge Mandated Indicator and the ACS Rendering Type whether to perform the requested challenge. Format the CReq message for a Browser-based implementation and base64url encode the CReq message. Pass the CReq message through the cardholder browser to the ACS URL received in the ARes message, by causing the cardholder browser to POST the form to the ACS URL using a server authenticated TLS link

For a transaction not authenticated (Transaction Status = N, U, or R)
Send necessary information from the ARes message to the 3DS Requestor Environment.

Note: For a Frictionless Flow, the next step is Step 22. Step 11 through Step 21 are applicable only for a Challenge Flow.

Step 11: The ACS
Receive the CReq message from the Browser and Validate. Prepare the authentication User Interface (ACS UI) to the Cardholder browser. Set the Interaction Counter to zero.

Step 12: The ACS and Browser
Embed all resources in the ACS-provided HTML and do not fetch via external URLs.
Send the ACS UI to the Cardholder over the channel established by the HTTP POST in Step 10. The browser displays the ACS UI to the Cardholder.

Step 13: The Cardholder
The Cardholder enters the authentication data as required by the ACS UI.

Step 14: The Browser
The Browser sends the entered authentication data to the ACS over the channel established by the HTTP POST in Step 10.

Step 15: The ACS
Check the authentication data entered by the Cardholder:

• 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
- Sets the Challenge Completion Indicator = Y
- Continues with Step 16

• If incorrect and authentication has failed, then the ACS:
- Increments the Interaction Counter and compares it to the ACS maximum challenges.
- If the Interaction Counter ≥ ACS maximum challenges 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 16

• If the Interaction Counter < ACS maximum challenges 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 12

Step 16: The ACS
The ACS format the RReq message. Establish a secure link with the DS. If the Cardholder abandons the challenge during the processing of Step 12 and Step 14, then the ACS sets the Challenge Cancelation Indicator to the appropriate value in the RReq message. Send the RReq message to the DS. Send one RReq message to the DS for each ARes message with an ARes Transaction Status = C.

Step 17: The DS
Receive the RReq message from the ACS and Validate**.** Establish a secure link with the 3DS Server using the 3DS Server URL extracted from the AReq message and Store the 3DS Server URL with the DS Transaction ID. Send the RReq message to the 3DS Server.

Step 18: The 3DS Server
Receive the RReq message or Error Message from the DS and Validate. Format the RRes message in Section 3.5 and send to the DS.

Note: For Payment Authentication, the Merchant can now proceed with Authorization processing with its Acquirer. However, the Merchant may first want to receive confirmation that the Cardholder has not abandoned the transaction.

Step 19: The DS
Receive the RRes message or Error Message from the 3DS Server and Validate. Log transaction information as required by the DS. Send the RRes message to the ACS as received from the 3DS Server

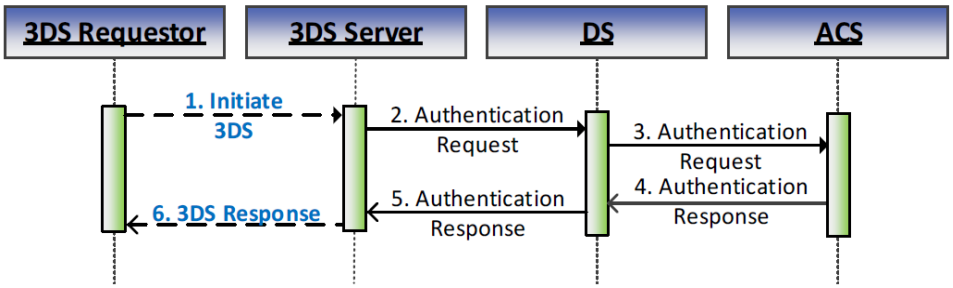
Step 20: The ACS
Receive the RRes message or Error Message from the DS and Validate.

Step 21: The ACS
Format the final CRes message. Base64url encode the final CRes message Send the final CRes message via an HTTP POST (for example, utilising JavaScript) through the browser to the Notification URL that was sent in the initial AReq message using the secure link established in Step 10.

Step 22: The 3DS Requestor Environment
The 3DS Requestor Environment continues with the checkout process and takes the appropriate action.
The 3DS Requestor should notify their 3DS Server and DS if invalid CRes messages are being received.
3-D Secure processing completes.

2.3 3RI-BASED PROCESSING FLOW

For 3RI transactions, the 3DS Requestor Environment utilizes only the 3DS Requestor and the 3DS Server as no cardholder is present during the transaction. The communication flows between the 3DS Requestor and 3DS Server are defined by the specific 3DS Integrator implementation. 3RI-based implementations shall support only Non-Payment Authentication (NPA) transactions.



Step 1: The 3DS Requestor
The 3DS Requestor is responsible for gathering the information for the AReq message assembled by the 3DS Server.

The information is available for the 3DS Server when the AReq message is built. This information can include:
Cardholder Account Information

- Merchant Risk Indicator
- DS Requestor Prior Transaction Authentication Information
- Payment Information
- Non-Payment Information
- Cardholder information

If the 3DS Requestor and 3DS Server are separate components, that data transferred between the components is protected at a level that satisfies Payment System security requirements with mutual authentication of both servers.

Step 2: The 3DS Server
Obtain the 3DS Requestor ID, the 3DS Server Reference Number, and conditionally the Acquirer BIN**.** Generate the 3DS Server Transaction ID.

Ensure availability of the necessary information for the AReq message gathered by components within the 3DS Requestor Environment. Determine which DS the authentication transaction needs to be sent based on the BIN and optionally other Cardholder account information. Establish a secure link with the DS

Format the AReq message and Send the AReq message to the DS.

Step 3: The DS
Receive the AReq message from the 3DS Server and Validate. Generate the DS Transaction ID. Check that the Message Version Number is supported by the DS and the ACS. Check the data elements in the AReq message. Determine if the Cardholder Account Number received in the AReq message is in a participating account range. Determine if the Cardholder Account Number is in an account range that has an ACS capable of processing 3-D Secure messages. Establish a secure link with the ACS. The DS may maintain multiple ACS URLs. If the first URL attempted is not available, then the DS will attempt to connect to one of the alternate URLs. Send the AReq to the ACS.

Step 4: The ACS
Receive the AReq message from the DS and Validate. Generate the ACS Transaction ID. Use the Cardholder Account Number from the AReq message to determine whether authentication is available for the Cardholder. Use the values of the 3RI Indicator and the 3DS Requestor Prior Transaction Authentication Information received in the AReq message when evaluating the transaction disposition

Evaluate the values received in the AReq message and determine whether the transaction1is:

- authenticated (Transaction Status = Y)
- requiring a Cardholder challenge to complete authentication (Transaction Status = C)
- not authenticated (Transaction Status = N)
- not authenticated, but a proof of authentication attempt (Authentication Value) was generated (Transaction Status = A)
- not authenticated, as authentication could not be performed due to technical or other issue (Transaction Status = U)
- not authenticated because the Issuer is rejecting authentication and requesting that authorisation not be attempted (Transaction Status = R)

If a transaction is deemed authenticated (Transaction Status = Y or A) the ACS performs the following:

- For a Non-Payment Authentication (Message Category = 02-NPA), the ACS *may*:
    - Generate the ECI value and Authentication Value and include in the ARes message as defined by the DS.
    - Assign an appropriate Transaction Status Reason Code value as defined by the specific DS and include in the ARes message.

- Whether the ECI Indicator/Authentication Value and/or the Transaction Status Reason value is included in the ARes message is defined by the DS.

Complete formatting of the ARes message and send the ARes message to the DS

Step 5: The DS

Check the data elements in the ARes message. Receive the ARes message or Error message from the ACS and Validate. Log the transaction information. Send the ARes message to the 3DS Server as received from the ACS using the secure link

Step 6: The 3DS Server

Receive the ARes message or Error Message from the DS and Validate**.** Send necessary information from the ARes message to the 3DS Requestor Environment.

2.4 PREPARATION FLOW

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 data will be organised by card range as configured by a DS. 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 flows.

The 3DS Server formats a PReq message (as defined in Sections 3.7,3.8) 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 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 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).

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

If the Serial Number has not changed, the DS would not provide back the Card Range Data element in the PRes message (i.e., it should be absent). Card Range data should not be in the PRes; however Serial Number should be included.

2.5 EXCEPTIONAL FLOW

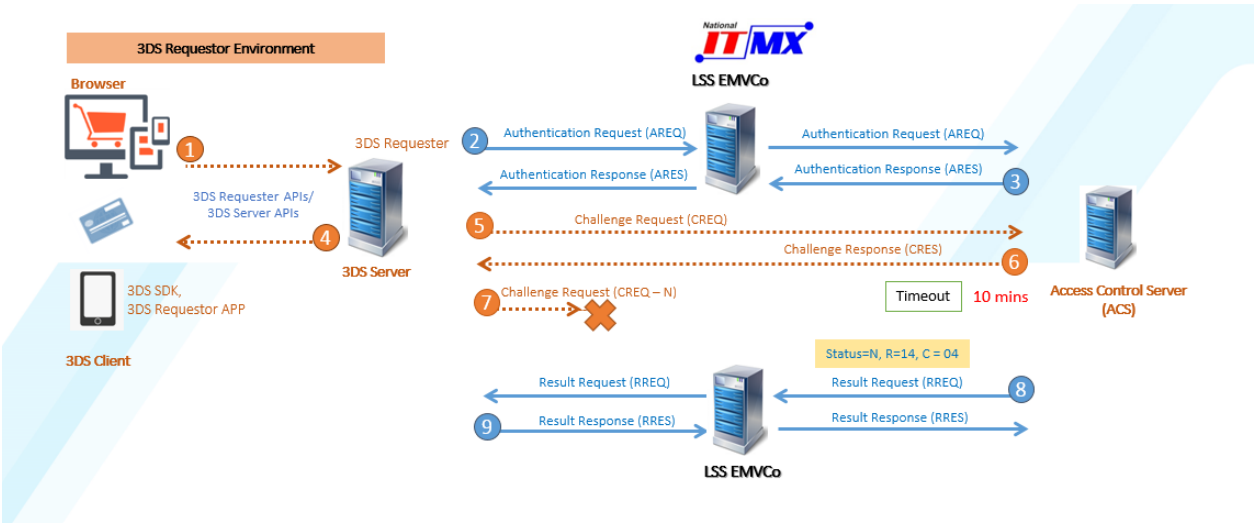
2.5.1 TRANSACTION TIMEOUT

Transaction Timeouts provide the maximum amount of time that a LSS transaction should be considered valid.

For App-based transactions, once a transaction has been established with the initial CReq/CRes message exchange between the ACS and the 3DS SDK, and when the ACS sends a CRes message to the 3DS SDK that requires an additional CReq message to continue or complete the Cardholder challenge (Challenge Completion Indicator = N), the ACS set a timeout value of 10 min (or 600 sec) after successfully sending each CRes message to the 3DS SDK

If the timeout expires before receiving the next CReq message from the 3DS SDK, the ASC 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),** and Challenge Cancelation Indicator = 04 and then clear any ephemeral key generated and stored for use in the CReq/CRes message exchange for this transaction.

Upon receiving the CReq message for a transaction that has timed out, the ASC send an Error Message with Error Component = A and Error Code = 402 to the 3DS SDK.



CReq Message Timeouts on the ACS

For Browser-based transactions, once a transaction has been established with a successful CReq POST to the ACS from the 3DS Requestor, and when the ACS sends the challenge interface to the challenge window, the ACS will Set a timeout value of 10 min (or 600 sec) after successfully sending each challenge interface to the challenge window

If the timeout expires before cardholder authentication can complete, the ASC send an RReq message to the DS to be passed to the 3DS Server with the Transaction Status = N and Transaction Status Reason = 14.

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

2.5.2 READ TIMEOUT

To ensure that 3-D Secure messages are handled across components in a timely manner, all components set read timeouts as appropriate for the implementation (i.e., programming language function, infrastructure components, etc.).

- **Read timeouts occur while waiting on a transaction response after the connection has been established and the message is sent to the recipient.**

  - The 3DS SDK shall set the SDK Maximum Timeout value from the time the TLS handshake has completed and the first CReq message is sent for processing to the ACS.

  Note: The SDK max time out element is in the AReq and can be used by the ACS to know how much time is available to complete the challenge.

  - If the ACS does not respond with the final CRes message before the SDK Maximum Timeout expiry, the 3DS SDK ends 3-D Secure processing, passes an Error Message to the ACS with Error Component = C and Error Code = 402 and passes an error message to the calling app, ending the 3-D Secure transaction.

  - The SDK Maximum Timeout shall not have a value less than five minutes.

- **Connection timeouts occur while making the initial connection (i.e., completing the TCP/IP connection and TLS handshake)**.

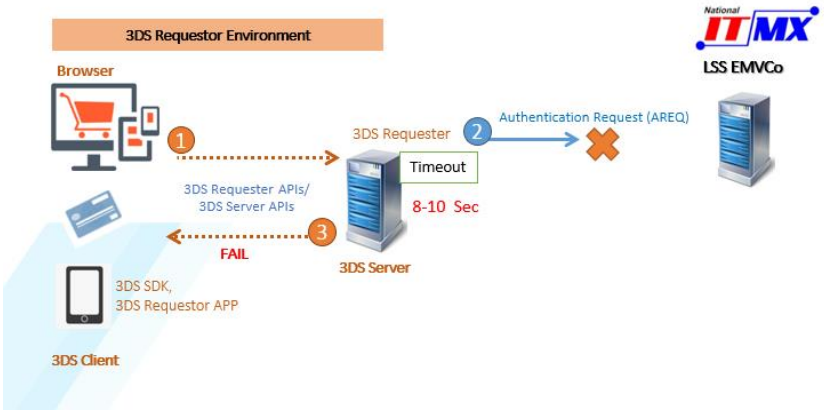
Timeouts are defined for each 3DS message type and 3DS component in the EMV 3DS specification, including:

- AReq/ARes Message Timeouts
- CReq/CRes Message Timeouts
- RReq/RRes Message Timeouts
- PReq/PRes Message Timeouts

2.5.2.1 AReq/Ares Message Timeouts

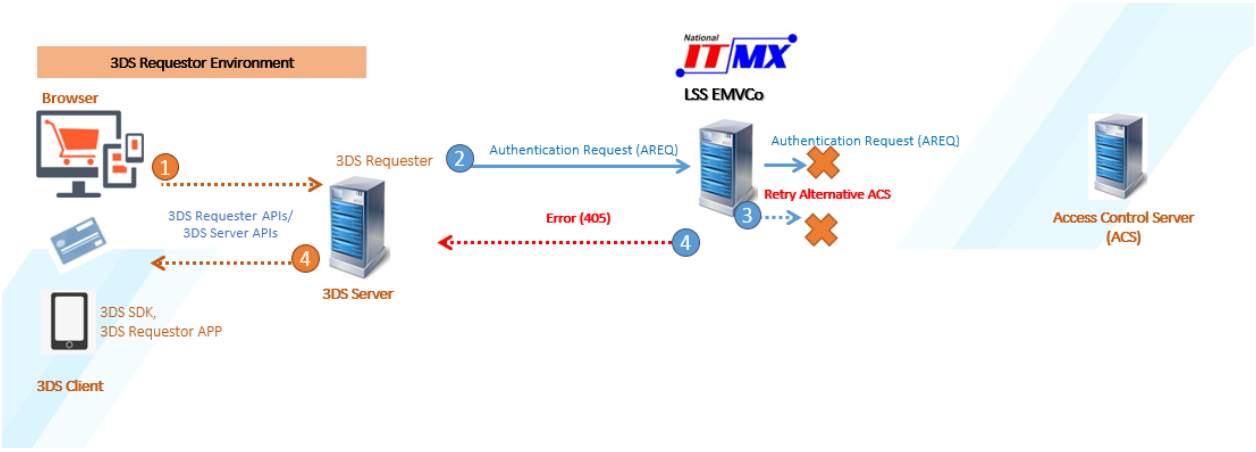
The 3DS Server: Any failure to complete the initial TCP/IP connection and TLS handshake to the DS shall result in an immediate retry or the 3DS Server shall try an alternate DS (if available). Upon second failure, the 3DS Server shall send an error to the 3DS Requestor to complete the transaction.

Set the read timeout value as defined by the DS receiving the request from the time the TLS handshake has completed and the full AReq message is sent for processing. If the DS has not responded with the ARes message before the 3DS Server read time expiry, the 3DS Server close the connection and result in a failed 3-D Secure transaction.



ARes Message Timeouts on the 3DS Server

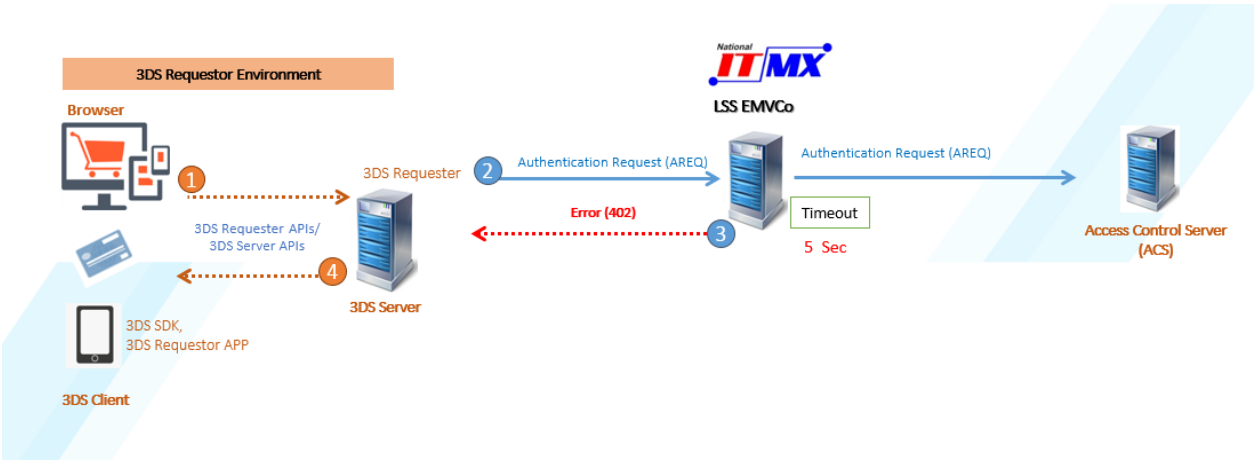
The DS: Any failure to complete the initial TCP/IP connection and TLS handshake to the ACS shall result in an immediate retry or the DS try an alternate ACS (if available). Upon second failure, the DS sends an Error Message with Error Component



AReq Message Timeouts on the DS

The DS set the ACS timeout value from the time the TLS handshake has completed and the full AReq message is sent for processing to the ACS.

If the DS has not received the ARes message from the ACS before the read timeout expiry, the DS sends an Error Message with Error Component.



ARes Message Timeouts on the DS

2.5.2.2 CReq/Cres Message Timeouts

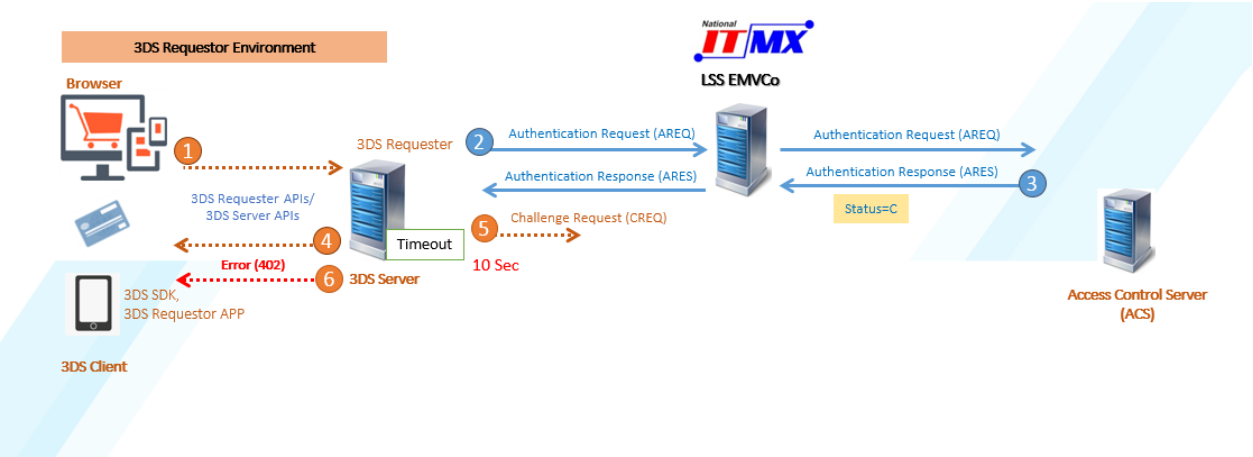
The 3DS SDK shall set the SDK Maximum Timeout value from the time the TLS handshake has completed and the first CReq message is sent for processing to the ACS.

Any failure to complete the initial connection and TLS handshake to the ACS shall result in an immediate retry. Upon second failure, the 3DS SDK shall send an error to the 3DS Requestor App to complete the transaction

The 3DS SDK shall set a 10 second timeout value from the time the TLS handshake has completed and the full CReq message is sent for processing to the ACS

If the ACS does not respond with the CRes message before the 3DS SDK 10-second read timeout expiry, the 3DS SDK **ends 3-D Secure processing**, passes an Error Message to the ACS with Error Component = C and Error Code = 402 and passes an error message to the calling app, ending the 3-D Secure transaction.

If the ACS does not respond with the **final CRes** message before the SDK Maximum Timeout expiry, the 3DS SDK **ends 3-D Secure processing**, passes an Error Message to the ACS with Error Component = C and Error Code = 402 and passes an error message to the calling app, ending the 3-D Secure transaction. The SDK Maximum Timeout shall not have a value less than five minutes.



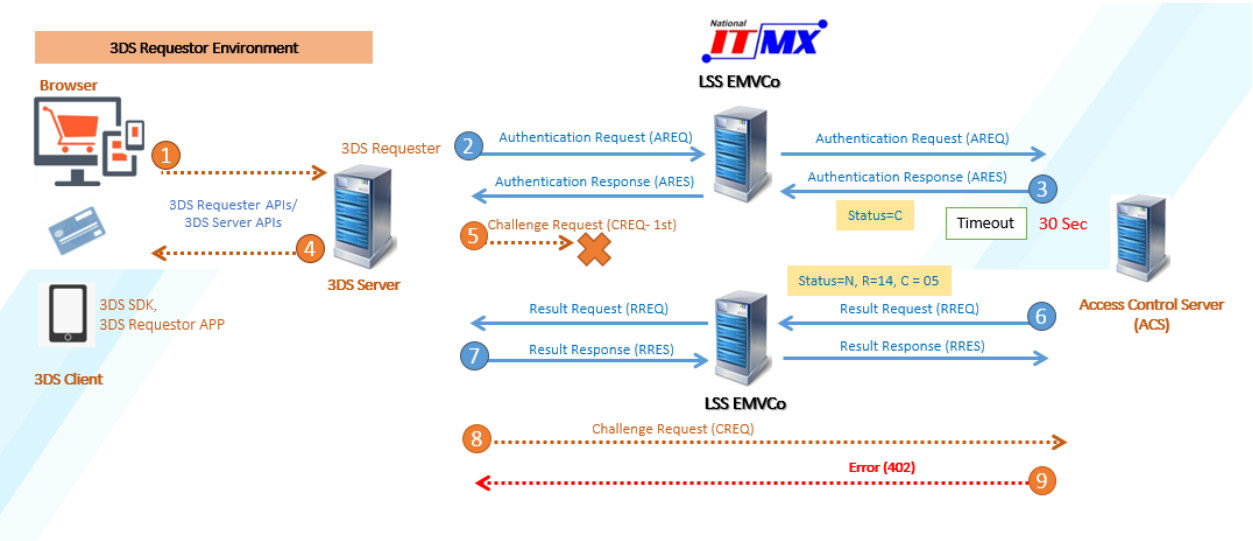
CReq Message Timeouts on the 3DS SDK

The ACS: Maintain the state of a 3-D Secure transaction when the ARes message contains the Transaction Status = C so that the corresponding CReq message can be processed and timeout values can be enforced.

Upon sending an ARes message with the Transaction Status = C, set a timeout value of 30 seconds for the receipt of the initial corresponding CReq message from the 3DS SDK or 3DS Requestor.

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), and Challenge Cancelation Indicator = 05 (Transaction timed out at the ACS—First CReq not received). Clear the ephemeral key generated and stored for use in the CReq/CRes message exchange for the current transaction

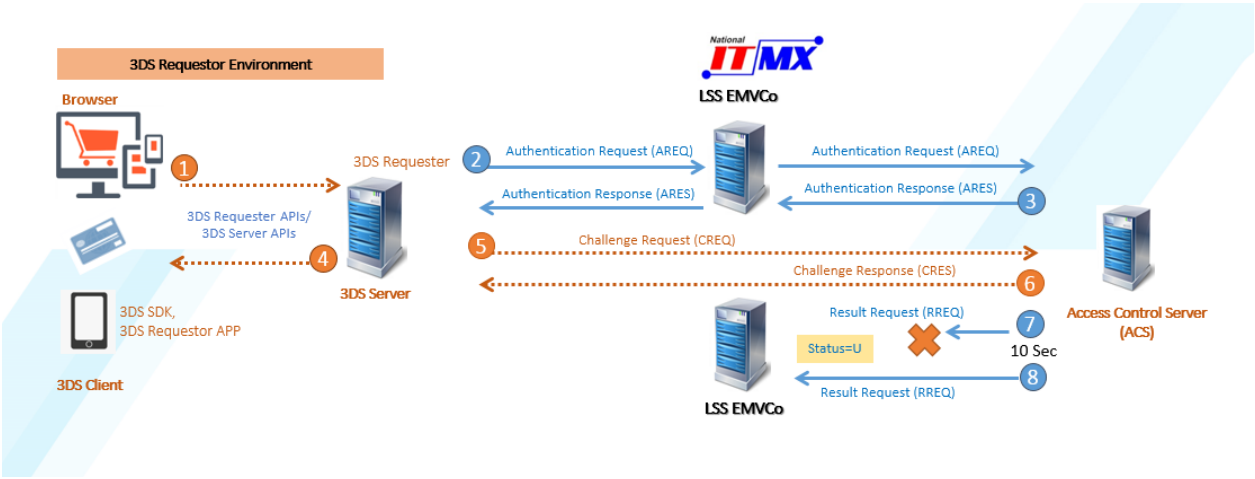
Upon receiving a CReq message for a transaction that has timed-out, send an Error Message with Error Component = A and Error Code = 402 to the 3DS SDK (for an App-based transaction) or 3DS Requestor (for a Browser-based transaction).



CReq Message Timeouts on the ACS

2.5.2.3 RReq/RRes Message Timeouts

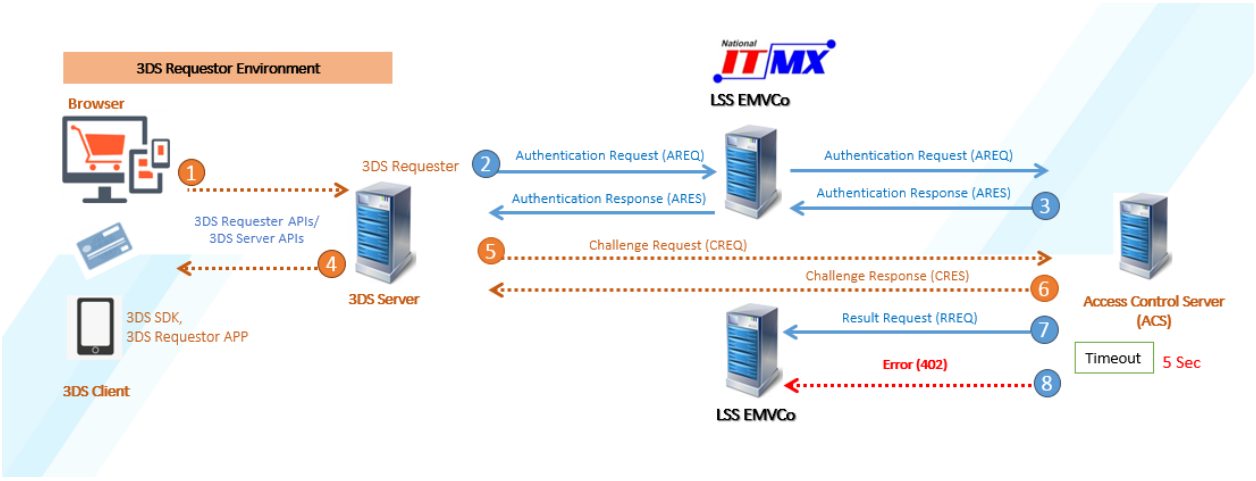
On ACS, sny failure to complete the initial connection and TLS handshake to the DS shall result in an immediate retry. Upon second failure, the ACS will wait 10 seconds and retry to connect to the DS until the message is delivered with Transaction Status = U.



RReq Message Connection failure on the ACS

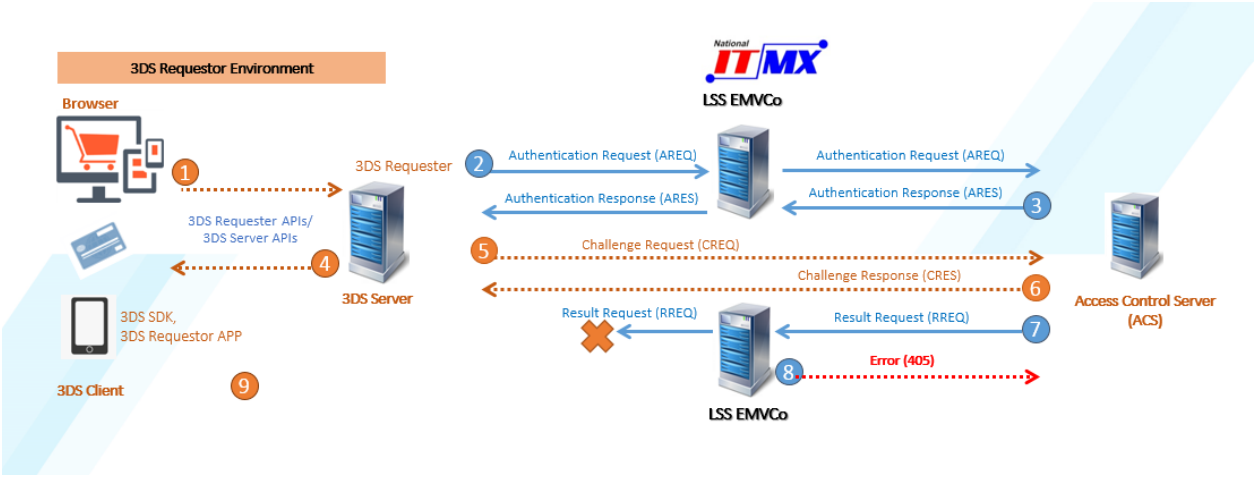
The ACS shall set a 5-second timeout value from the time the TLS handshake has completed and the full RReq message is sent for processing to the DS.

If the DS has not responded with the RRes message or an Error message before the 5-second read timeout expiry, the ACS shall return to the DS an Error Message (as defined in A.5.5) with Error Component = A and Error Code = 402.



RReq Message Timeouts on the ACS

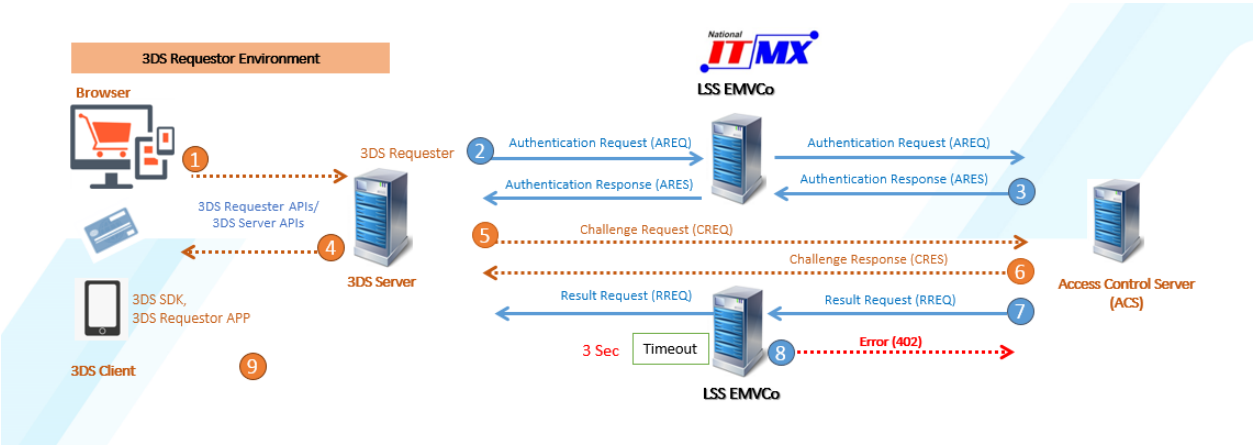
Any failure to complete the initial connection and TLS handshake to the 3DS Server shall result in an immediate retry. Upon second failure, the DS shall send an Error Message with Error Component = D and Error Code = 405 to the ACS to complete the transaction.
Note: No further processing shall occur between the DS and 3DS Server as the SDK has timed out.



RReq Message Connection failure on the DS

The DS shall set a 3-second timeout value from the time the TLS handshake has completed and the full RReq message is sent for processing to the 3DS Server URL.

If the 3DS Server has not sent the RRes message before the 3- second read timeout expiry, the DS shall send an Error Message with Error Component = D and Error Code = 402 to the ACS to complete the transaction.
Note: No further processing shall occur between the DS and 3DS Server as the SDK has timed out

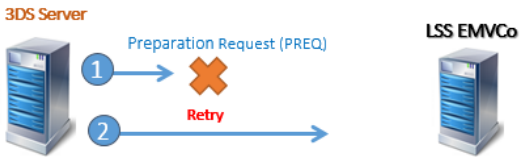


RReq Message Timeouts on the DS

2.5.2.4 PReq/PRes Message Timeouts

3DS Servers shall make a call to each registered DS every 24 hours at a minimum, and once per hour at a maximum to refresh their cache. Send the PReq message via a secure link with the DS established. Immediately retry a connection upon any failure to complete the initial TCP/IP connection and TLS handshake to the DS.

Upon the second failure to complete the TCP/IP connection and TLS handshake to the DS, end the transaction, and periodically retry within the 24-hour window at 60-second intervals until the transaction completes successfully.



PRes Message Timeouts on the DS

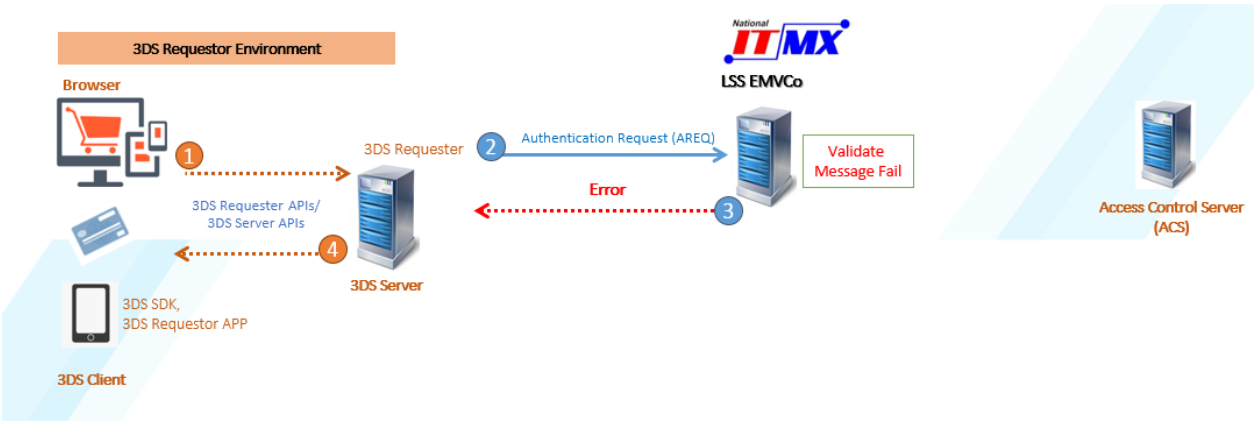
2.5.3 MESSAGE ERROR HANDING

The 3-D Secure component will receive and validate the message, verify and decrypt the message before performing further processing. The validation process will check the message against the specification for presence and format of each data element in the message as defined in section 3. The verification and decryption process will perform the cryptographic process against the message as defined in Section 4.
The outcome of the validation can be:

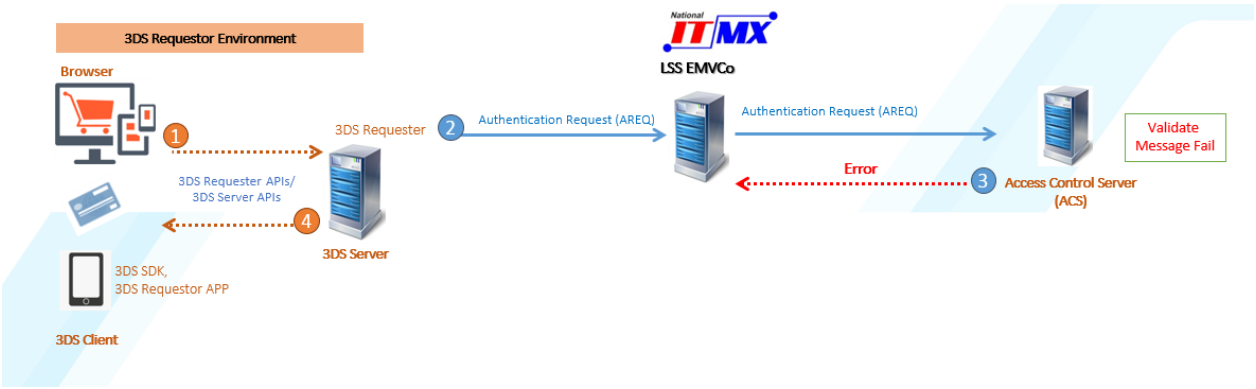
- Message is valid and therefore standard processing is followed, OR
- Message is invalid and therefore an exception scenario is processed, OR
- Message received is an Error Message and therefore an exception scenario is processed.

This section describes only the process where a message is invalid or an Error Message is received.

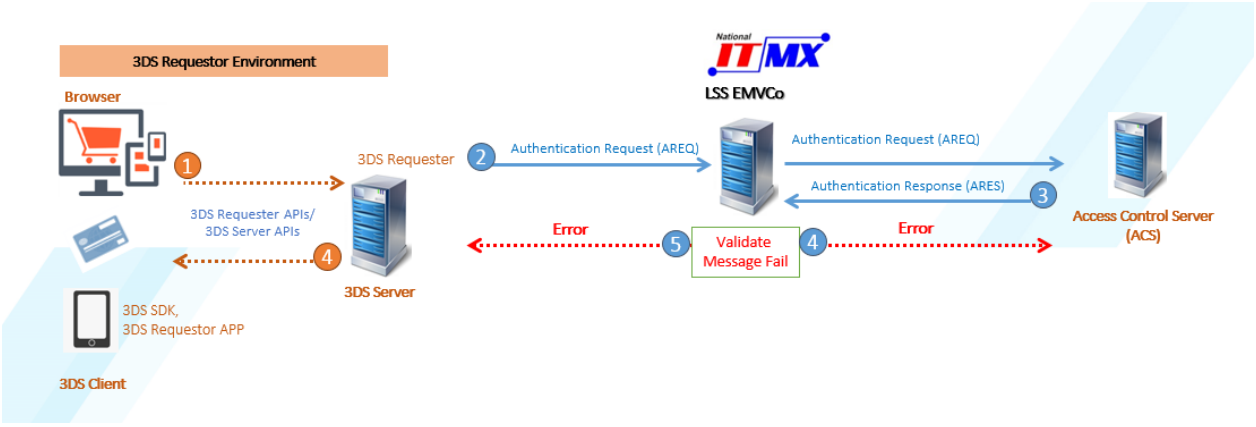
2.5.3.1 DS AReq Message Error Handling



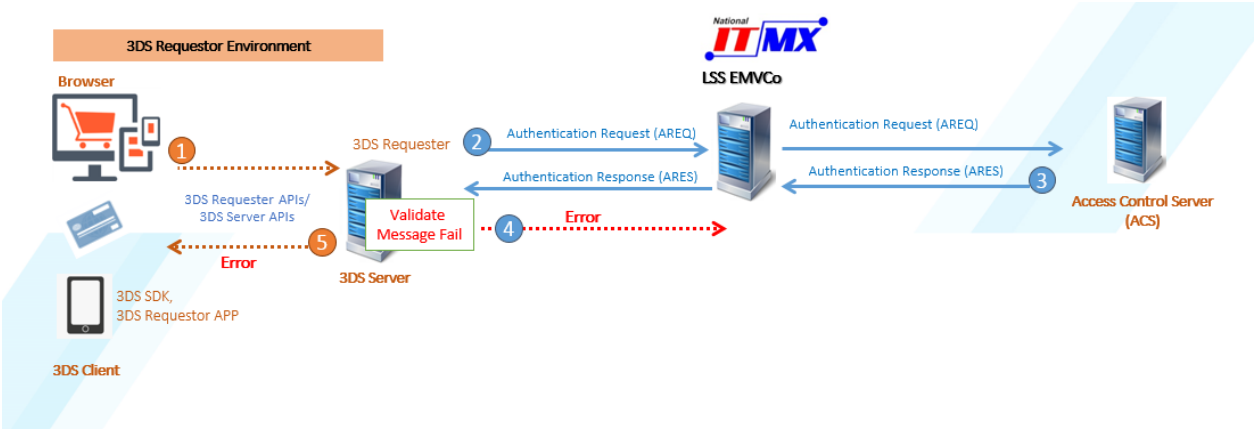
2.5.3.2 ACS AReq Message Error Handling



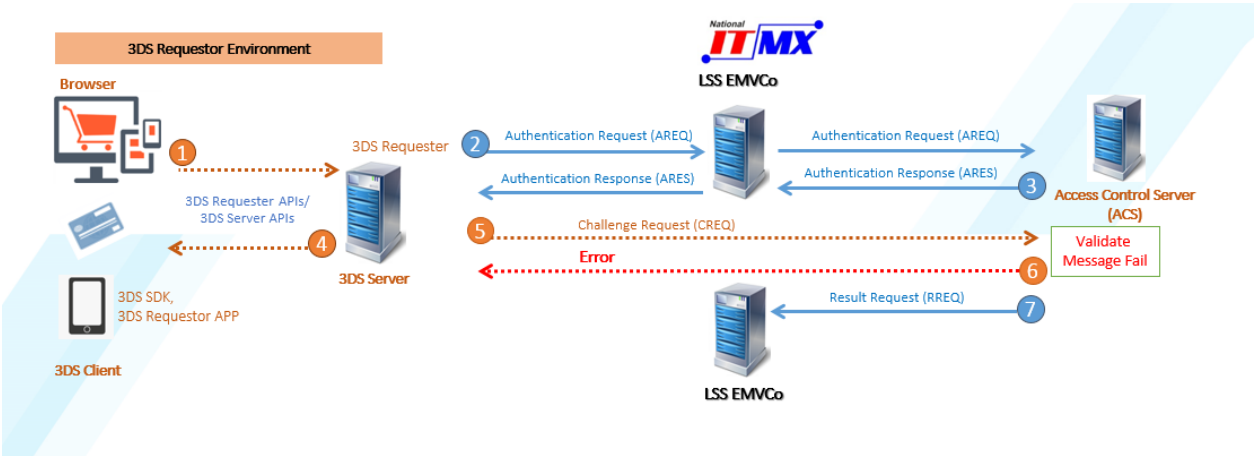
2.5.3.3 DS ARes Message Error Handling



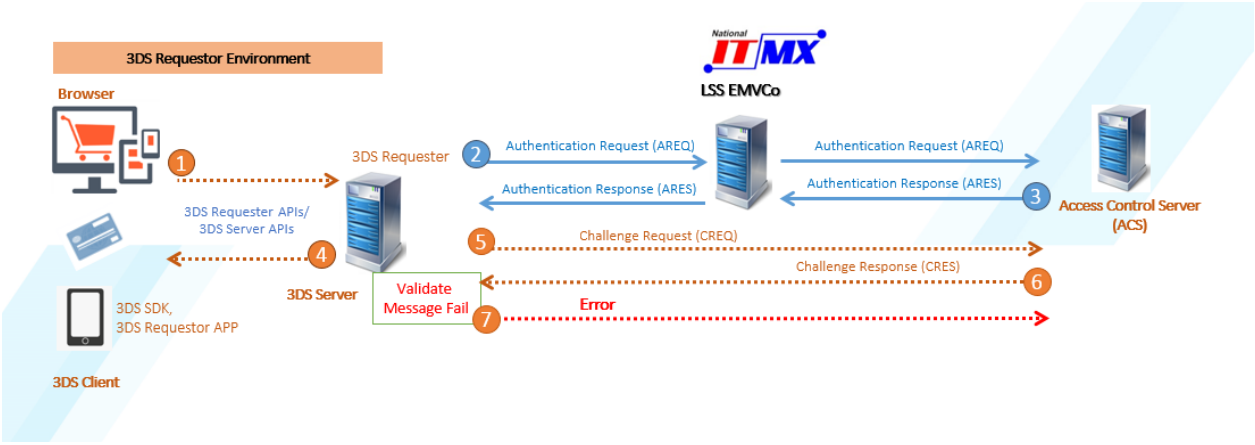
2.5.3.4 3DS Server ARes Message Error Handling



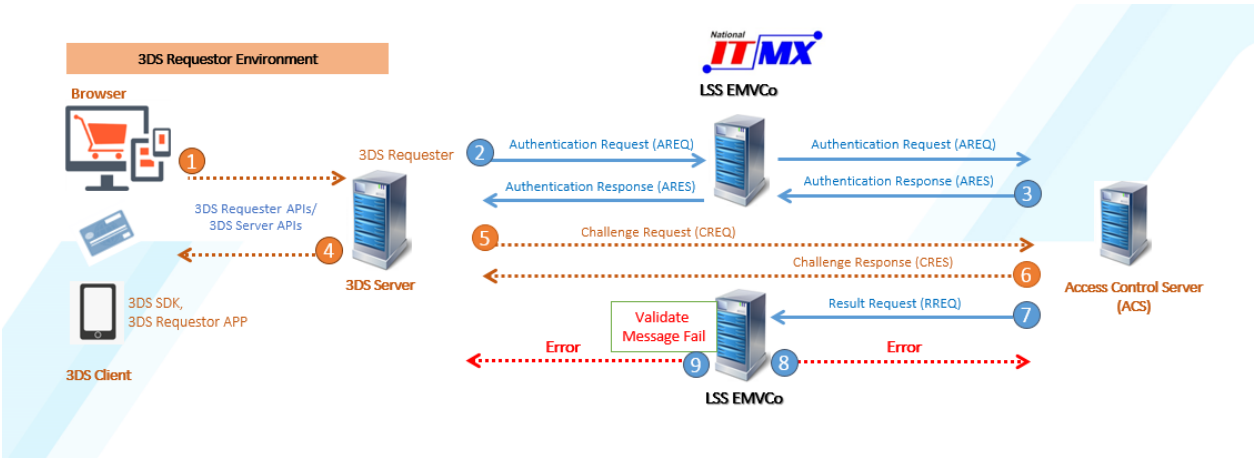
2.5.3.5 ACS CReq Message Error Handling



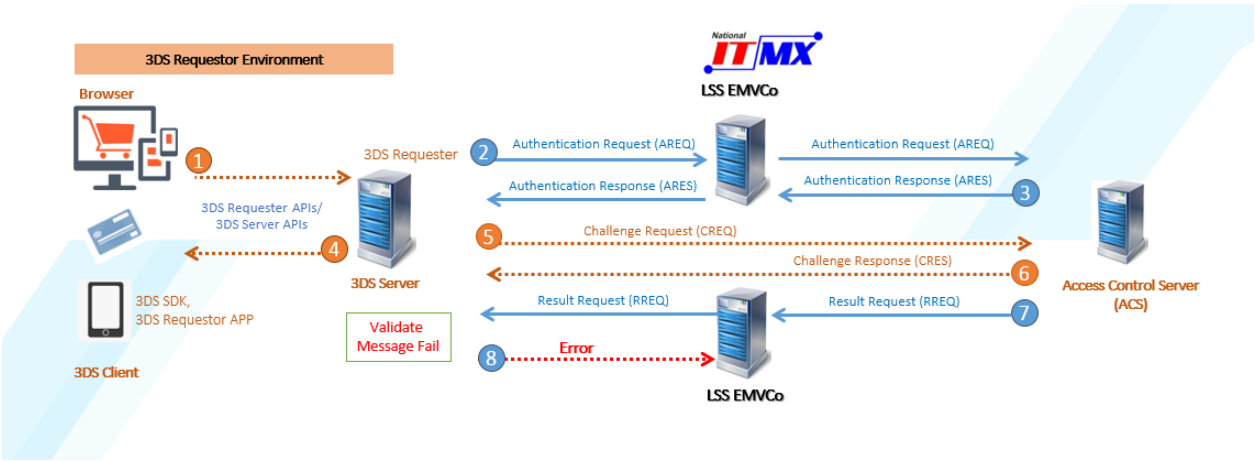
2.5.3.6 3DS SDK CRes Message Error Handling



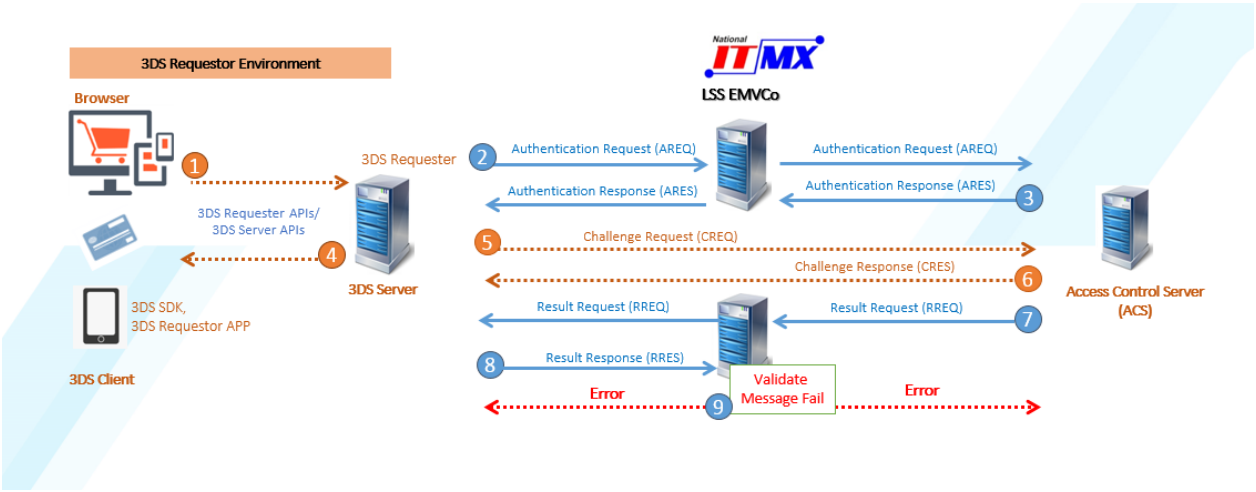
2.5.3.7 DS RReq Message Error Handling



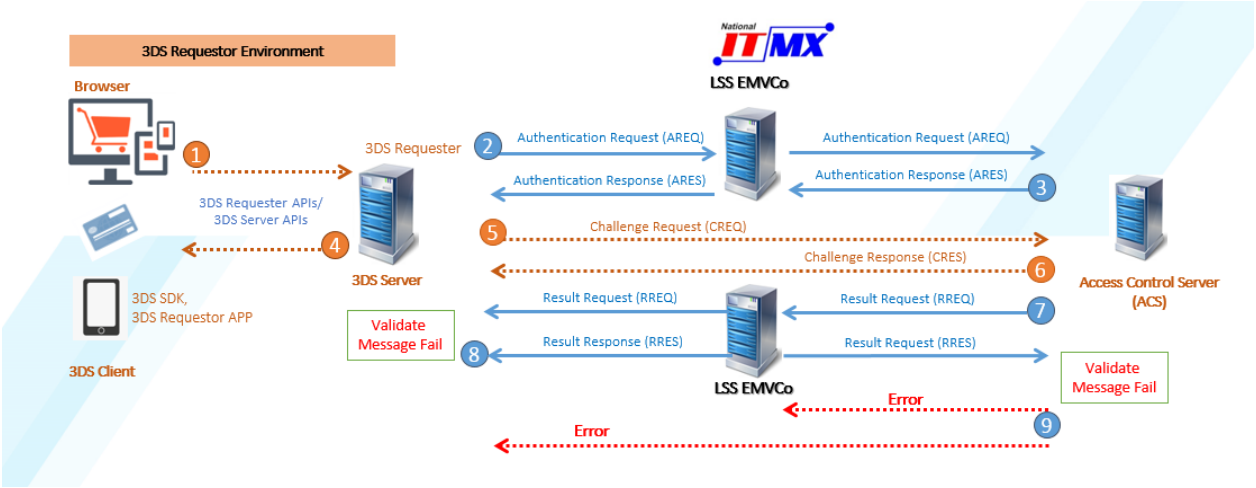
2.5.3.8 3DS Server RReq Message Error Handling



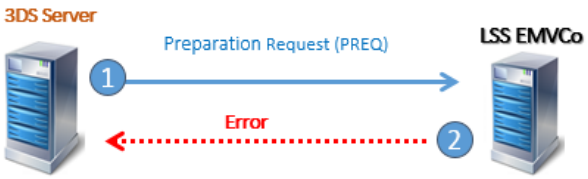
2.5.3.9 DS RRes Message Error Handling



2.5.3.10 ACS RRes Message Error Handling



2.5.3.11 DS PReq Message Error Handling



Chapter 3

3. MESSAGE LAYOUT

Data element information for each message layout and the Standards used to identify the information are as follows:

- **Data Element Name**—Identifies the data element
- **Field Name**—Identifies the field name for the data element
- **Description**—Identifies the data element purpose, and additional detail as applicable
- **Source**—Identifies the 3-D Secure component that is responsible to provide the data element in the message

- **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.

- **Device Channel**—Identifies the inclusion of a data element in a message based on the Device Channel used for a specific transaction. The following Standard is used to identify the Device Channel type:

  01-APP—App-based Authentication

  02-BRW—Browser-based Authentication

  03-3RI—Verification of Account

- **Message Category**—Identifies the inclusion of a data element in a Message based on the type of transaction. The following Standard is used to identify the Authentication type:

  01-PA—Payment Authentication

  02-NPA—Non-Payment Authentication

- **Message Inclusion**—Identifies the Message Type(s) that the data element is included in, and whether the inclusion of the data element in the Message Type is Required, Optional, or Conditional for both Message Categories.

3.1 AUTHENTICATION REQUEST (AREQ)

The AReq message is the initial message in the 3-D Secure authentication flow. The 3DS Server forms the AReq message when requesting authentication of the Cardholder. It can contain Cardholder, payment, and Device information for the transaction. There is only one AReq message per authentication.

Step. Number	Field Name	Description	Source	Length/Format/Values	Device Channel	Message Category	Message Inclusion	Conditional Inclusion
1	threeDSCompInd	Indicates whether the 3DS Method successfully completed.	3DS Server	Length: 1 character JSON Data Type: String Values accepted: • Y = Successfully completed • N = Did not successfully complete • U = Unavailable—3DS Method URL was not present in the PRes message data for the card range associated with the Cardholder Account Number.	02-BRW	01-PA 02-NPA	AReq = R	
2	threeDSRequestorAuthenticationInd	Indicates the type of Authentication request. This data element provides additional information to the ACS to determine the best approach for handing an authentication request.	3DS Server	Length: 2 characters JSON Data Type: String Values accepted: • 01 = Payment transaction • 02 = Recurring transaction • 03 = Instalment transaction • 04 = Add card • 05 = Maintain card • 06 = Cardholder verification as part of EMV token ID&V • 07–79 = Reserved for EMVCo future use (values invalid until defined by EMVCo) • 80-99 = Reserved for DS use	01-APP 02-BRW	01-PA 02-NPA	AReq = R	

3	threeDSRequestorAuthenticationInfo	Information about how the 3DS Requestor authenticated the cardholder before or during the transaction.	3DS Server	Length: Variable JSON Data Type: Object Refer to 3.9.14 for data elements to include. Note: Data will be formatted into a JSON object prior to being placed into the 3DS requestor Authentication Information field of the message.	01-APP 02-BRW	01-PA 02-NPA	AReq = O	Optional, recommended to include.
4	threeDSRequestorChallengeInd	Indicates whether a challenge is requested for this transaction. For example: For 01-PA, a 3DS Requestor may have concerns about the transaction, and request a challenge. For 02-NPA, a challenge may be necessary when adding a new card to a wallet. For local/regional mandates or other variables.	3DS Server	Length: 2 characters JSON Data Type: String Values accepted: • 01 = No preference • 02 = No challenge requested • 03 = Challenge requested: 3DS Requestor Preference • 04 = Challenge requested: Mandate • 05–79 = Reserved for EMVCo future use (values invalid until defined by EMVCo) • 80-99 = Reserved for DS use Note: If the element is not provided, the expected action is that the ACS would interpret as 01 = No preference.	01-APP 02-BRW	01-PA 02-NPA	AReq = O	
5	threeDSRequestorID	DS assigned 3DS Requestor identifier. Each DS will provide a unique ID to each 3DS Requestor on an individual basis.	3DS Server	Length: Variable, maximum 35 characters JSON Data Type: String Value accepted: Any individual DS may impose specific formatting and character requirements on the contents of this field.	01-APP 02-BRW 03-3RI	01-PA 02-NPA	AReq = R	

6	threeDSRequestorName	DS assigned 3DS Requestor name. Each DS will provide a unique name to each 3DS Requestor on an individual basis.	3DS Server	Length: Variable, maximum 40 characters JSON Data Type: String Values accepted: Any individual DS may impose specific formatting and character requirements on the contents of this field.	01-APP 02-BRW 03-3RI	01-PA 02-NPA	AReq = R	
7	threeDSRequestorPriorAuthenticationInfo	Information about how the 3DS Requestor authenticated the cardholder as part of a previous 3DS transaction.	3DS Server	Length: Variable Format: String JSON Data Type: Object Values accepted: Refer to 3.9.15 for data elements to include. Note: Data will be formatted into a JSON object prior to being placed into the 3DS Requestor Prior Transaction Authentication Information field of the message.	01-APP 02-BRW 03-3RI	01-PA 02-NPA	AReq = O	Optional, recommended to include.
8	threeDSRequestorURL	Fully qualified URL of 3DS Requestor website or customer care site. This data element provides additional information to the receiving 3-D Secure system if a problem arises and should provide contact information.	3DS Server	Length: Variable, maximum 2048 characters JSON Data Type: String Value accepted: Fully qualified URL For example: http://server.domainname.com	01-APP 02-BRW 03-3RI	01-PA 02-NPA	AReq = R	
9	threeDSServerRefNumber	Unique identifier assigned by the EMVCo secretariat upon testing and approval.	3DS Server	Length: Variable, maximum 32 characters JSON Data Type: String Value accepted: Set by the EMVCo Secretariat	01-APP 02-BRW 03-3RI	01-PA 02-NPA	AReq = R	

10	threeDSServerOperatorID	DS assigned 3DS Server identifier. Each DS can provide a unique ID to each 3DS Server on an individual basis.	3DS Server	Length: Variable, maximum 32 characters JSON Data Type: String Value accepted: Any individual DS may impose specific formatting and character requirements on the contents of this field.	01-APP 02-BRW 03-3RI	01-PA 02-NPA	AReq = C	Requirements for the presence of this field are DS specific.
11	threeDSServerTransID	Universally unique transaction identifier assigned by the 3DS Server to identify a single transaction.	3DS Server	Length: 36 characters JSON Data Type: String Value accepted: Canonical format as defined in IETF RFC 4122. May utilise any of the specified versions if the output meets specified requirements.	01-APP 02-BRW 03-3RI	01-PA 02-NPA	AReq = R	Required in Error Message if available (e.g. can be obtained from a message or is being generated).
12	threeDSServerURL	Fully qualified URL of the 3DS Server to which the DS will send the RReq message after the challenge has completed. Incorrect formatting will result in a failure to deliver the transaction results via the RReq message.	3DS Server ACS	Length: Variable, maximum 2048 characters JSON Data Type: String Value accepted: Fully qualified URL Example value: https://server.adomainname.net	01-APP 02-BRW	01-PA 02-NPA	AReq = R	
13	threeRIInd	Indicates the type of 3RI request. This data element provides additional information to the ACS to determine the best approach for handing a 3RI request.	3DS Server	Length: 2 characters JSON Data Type: String Values accepted: • 01 = Recurring transaction • 02 = Instalment transaction • 03 = Add card • 04 = Maintain card information • 05 = Account verification • 06–79 = Reserved for EMVCo future use (values invalid until defined by	03-3RI	02-NPA	AReq = R	

				EMVCo) • 80-99 = Reserved for DS use				
14	acctType	Indicates the type of account. For example, for a multi-account card product.	3DS Server	Length: 2 characters JSON Data Type: String Values accepted: • 01 = Not Applicable • 02 = Credit • 03 = Debit • 04–79 = Reserved for EMVCo future use (values invalid until defined by EMVCo) • 80–99 = DS or Payment System-specific	01-APP 02-BRW 03-3RI	01-PA 02-NPA	AReq = C	Required if 3DS Requestor is asking Cardholder which Account Type they are using before making the purchase. Required in some markets (for example, for Merchants in Brazil). Otherwise, it is optional.
15	acquirerBIN	Acquiring institution identification code as assigned by the DS receiving the AReq message.	3DS Server	Length: Variable, maximum 11 characters JSON Data Type: String Value accepted: This value correlates to the Acquirer BIN as defined by each Payment System or DS.	01-APP 02-BRW 03-3RI	01-PA 02-NPA	01-PA: AReq = R 02-NPA: AReq = O	
16	acquirerMerchantID	Acquirer-assigned Merchant identifier. This may be the same value that is used in authorisation requests sent on behalf of the 3DS Requestor and is represented in ISO 8583 formatting requirements.	3DS Server	Length: Variable, maximum 35 characters JSON Data Type: String Value accepted: Individual Directory Servers may impose specific format and character requirements on the contents of this field.	01-APP 02-BRW 03-3RI	01-PA 02-NPA	01-PA: AReq = R 02-NPA AReq = O	

17	addrMatch	Indicates whether the Cardholder Shipping Address and Cardholder Billing Address are the same.	3DS Server	Length: 1 character JSON Data Type: String Values accepted: • Y = Shipping Address matches Billing Address • N = Shipping Address does not match Billing Address	01-APP 02-BRW	01-PA 02-NPA	AReq = O	
18	broadInfo	Unstructured information sent between the 3DS Server, the DS and the ACS.	3DS Server DS ACS	Length: 4096 characters JSON Data Type: Object	01-APP 02-BRW 03-3RI	01-PA 02-NPA	AReq = C	Requirements for the presence of this field are DS specific.
19	browserAcceptHeader	Exact content of the HTTP accept headers as sent to the 3DS Requestor from the Cardholder's browser.	3DS Server	Length: Variable, maximum 2048 characters JSON Data Type: String Value accepted: If the total length of the accept header sent by the browser exceeds 2048 characters, the 3DS Server truncates the excess portion. Refer to 3.9.17 for additional detail.	02-BRW	01-PA 02-NPA	AReq = R	
20	browserIP	IP address of the browser as returned by the HTTP headers to the 3DS Requestor.	3DS Server	Length: Variable, maximum 45 characters JSON Data Type: String Value accepted: • IPv4 address is represented in the dotted decimal format of 4 sets of decimal numbers separated by dots. The decimal number in each and every set is in the range 0 to 255. Example IPv4 address: 1.12.123.255 • IPv6 address is represented as eight groups of four hexadecimal digits, each group representing 16 bits (two	02-BRW	01-PA 02-NPA	AReq = C	Shall include this field where regionally acceptable.

				octets. The groups are separated by colons (:). Example IPv6 address:2011:0db8:85 a3:0101:0101:8a2e:03 70:7334 Refer to 3.9.17 for additional detail.				
21	browserJavaEnabled	Boolean that represents the ability of the cardholder browser to execute Java. Value is returned from the navigator.javaEnabled property. Refer to 3.9.17 for additional detail.	3DS Server	JSON Data Type: Boolean Values accepted: • true • false	02-BRW	01-PA 02-NPA	AReq = R	
22	browserLanguage	Value representing the browser language as defined in IETF BCP47. Returned from navigator.language property. Refer to 3.9.17 for additional detail.	3DS Server	Length: Variable, 1–8 characters JSON Data Type: String	02-BRW	01-PA 02-NPA	AReq = R	
23	browserColorDepth	Value representing the bit depth of the colour palette for displaying images, in bits per pixel. Obtained from Cardholder browser using the screen.colorDepth property. Refer to 3.9.17 for additional detail.	3DS Server	Length: 1–2 characters JSON Data Type: String Values accepted: • 1 = 1 bit • 4 = 4 bits • 8 = 8 bits • 15 = 15 bits • 16 = 16 bits • 24 = 24 bits • 32 = 32 bits • 48 = 48 bits	02-BRW	01-PA 02-NPA	AReq = R	

24	browserScreenHeight	Total height of the Cardholder's screen in pixels. Value is returned from the screen.height property. Refer to 3.9.17 for additional detail.	3DS Server	Length: Variable, 1–6 characters JSON Data Type: String	02-BRW	01-PA 02-NPA	AReq = R	
25	browserScreenWidth	Total width of the cardholder's screen in pixels. Value is returned from the screen.width property. Refer to 3.9.17 for additional detail.	3DS Server	Length: Variable, 1–6 characters JSON Data Type: String	02-BRW	01-PA 02-NPA	AReq = R	
26	browserTZ	Time difference between UTC time and the Cardholder browser local time, in minutes.	3DS Server	Length: 1–5 characters JSON Data Type: String Value accepted: Value is returned from the getTimezoneOffset() method. Refer to 3.9.17 for additional detail.	02-BRW	01-PA 02-NPA	AReq = R	
27	browserUserAgent	Exact content of the HTTP user-agent header.	3DS Server	Length: Variable, maximum 2048 characters JSON Data Type: String Value accepted: Note: If the total length of the User-Agent sent by the browser exceeds 2048 characters, the 3DS Server truncates the excess portion. Refer to 3.9.17 for additional detail.	02-BRW	01-PA 02-NPA	AReq = R	

28	cardExpiryDate	Expiry Date of the PAN or token supplied to the 3DS Requestor by the Cardholder.	3DS Server	Length: 4 characters JSON Data Type: String Format accepted: YYMM	01-APP 02-BRW 03-3RI	01-PA 02-NPA	AReq = C	The requirements for the presence of this field are DS specific.
29	browserJavaEnabled	Boolean that represents the ability of the cardholder browser to execute Java. Value is returned from the navigator.javaEnabled property. Refer to 3.9.17 for additional detail.	3DS Server	JSON Data Type: Boolean Values accepted: • true • false	02-BRW	01-PA 02-NPA	AReq = R	
30	acctNumber	Account number that will be used in the authorisation request for payment transactions. May be represented by PAN, token.	3DS Server	Length: Variable, 13–19 characters JSON Data Type: String Value accepted: Format represented ISO 7812.	01-APP 02-BRW 03-3RI	01-PA 02-NPA	AReq = R	
31	acctID	Additional information about the account optionally provided by the 3DS Requestor.	3DS Server	Length: Variable, maximum 64 characters JSON Data Type: String	01-APP 02-BRW 03-3RI	01-PA 02-NPA	AReq = O	

32	billAddrCity	The city of the Cardholder billing address associated with the card used for this purchase.	3DS Server	Length: Variable, maximum 50 characters JSON Data Type: String	01-APP 02-BRW 03-3RI	01-PA 02-NPA	AReq = C	01-PA: Required unless market or regional mandate restricts sending this information. 02-NPA: Required (if available) unless market or regional mandate restricts sending this information.
33	billAddrCountry	The country of the Cardholder billing address associated with the card used for this purchase.	3DS Server	Length: 3 characters JSON Data Type: String Value accepted: Shall be the ISO 3166-1 numeric three-digit country code, other than exceptions listed in 3.9.2	01-APP 02-BRW 03-3RI	01-PA 02-NPA	AReq = C	Required if Cardholder Billing Address State is present. 01-PA: Required unless market or regional mandate restricts sending this information. 02-NPA: Required (if available) unless market or regional mandate restricts sending this information.

34	billAddrLine1	First line of the street address or equivalent local portion of the Cardholder billing address associated with the card used for this purchase.	3DS Server	Length: Variable, maximum 50 characters JSON Data Type: String	01-APP 02-BRW 03-3RI	01-PA 02-NPA	AReq = C	01-PA: Required unless market or regional mandate restricts sending this information. 02-NPA: Required (if available) unless market or regional mandate restricts sending this information.
35	billAddrLine2	Second line of the street address or equivalent local portion of the Cardholder billing address associated with the card used for this purchase.	3DS Server	Length: Variable, maximum 50 characters JSON Data Type: String	01-APP 02-BRW 03-3RI	01-PA 02-NPA	AReq = C	01-PA: Required unless market or regional mandate restricts sending this information. 02-NPA: Required (if available) unless market or regional mandate restricts sending this information.

36	billAddrLine3	Third line of the street address or equivalent local portion of the Cardholder billing address associated with the card used for this purchase.	3DS Server	Length: Variable, maximum 50 characters JSON Data Type: String	01-APP 02-BRW 03-3RI	01-PA 02-NPA	AReq = C	01-PA: Required unless market or regional mandate restricts sending this information. 02-NPA: Required (if available) unless market or regional mandate restricts sending this information.
37	billAddrPostCode	ZIP or other postal code of the Cardholder billing address associated with the card used for this purchase.	3DS Server	Length: Variable, maximum 16 characters JSON Data Type: String	01-APP 02-BRW 03-3RI	01-PA 02-NPA	AReq = C	01-PA: Required unless market or regional mandate restricts sending this information. 02-NPA: Required (if available) unless market or regional mandate restricts sending this information.

38	billAddrState	The state or province of the Cardholder billing address associated with the card used for this purchase.	3DS Server	Length: Variable, maximum 3 characters JSON Data Type: String Value accepted: Should be the country subdivision code defined in ISO 3166-2	01-APP 02-BRW 03-3RI	01-PA 02-NPA	AReq = C	01-PA: Required unless market or regional mandate restricts sending this information, or State is not applicable for this country. 02-NPA: Required (if available) unless market or regional mandate restricts sending this information, or State is not applicable for this country.
39	email	The email address associated with the account that is either entered by the Cardholder, or is on file with the 3DS Requestor.	3DS Server	Length: Variable, maximum 254 characters JSON Data Type: String Value	01-APP 02-BRW 03-3RI	01-PA 02-NPA	AReq = C	Required unless market or regional mandate restricts sending this information.
40	homePhone	The home phone number provided by the Cardholder.	3DS Server	Length: Variable • cc: 1–3 characters • subscriber: variable, maximum 15 characters Format: JSON Object; strings	01-APP 02-BRW 03-3RI	01-PA 02-NPA	AReq = C	Required (if available) unless market or regional mandate

				Values accepted:
				Country Code and Subscriber sections of the number, represented by the following named fields:
				• cc
				• subscriber
				Refer to ITU-E.164 for additional information on format and length.
				Example:
				“homePhone”: {
				“cc”: “1” ,
				“subscriber”:
				“1234567899”
				}
								restricts sending this information.

41	mobilePhone	The mobile phone number provided by the Cardholder.	3DS Server	Length: Variable
				• cc: 1–3 characters
				• subscriber: variable, maximum 15 characters
				Format: JSON Object; strings
				Values accepted:
				Country Code and Subscriber sections of the number, represented by the following named fields:
				• cc
				• subscriber
				Refer to ITU-E.164 for additional information on format and length.
				Example:
				“mobilePhone”: {
				“cc”: “1” ,
				“subscriber”:
				“1234567899”
				}
					01-APP
					02-BRW
					03-3RI
						01-PA
						02-NPA
							AReq = C	Required (if available) unless market or regional mandate restricts sending this information.

42	cardholderName	Name of the Cardholder	3DS Server	Length: Variable, 2–45 characters JSON Data Type: String Value accepted: Alphanumeric special characters, listed in EMV Book 4, “Appendix B”.	01-APP 02-BRW 03-3RI	01-PA 02-NPA	AReq = C	Required unless market or regional mandate restricts sending this information.
43	shipAddrCity	City portion of the shipping address requested by the Cardholder.	3DS Server	Length: Variable, maximum 50 characters JSON Data Type: String	01-APP 02-BRW 03-3RI	01-PA 02-NPA	AReq = C	Required (if available) unless market or regional mandate restricts sending this information.
44	shipAddrCountry	Country of the shipping address requested by the Cardholder.	3DS Server	Length: 3 characters JSON Data Type: String Value accepted: ISO 3166-1 three-digit country code, other than those listed in 3.9.2 .	01-APP 02-BRW 03-3RI	01-PA 02-NPA	AReq = C	Required if Cardholder Shipping Address State is present. Required (if available) unless market or regional mandate restricts sending this information.
45	shipAddrLine1	First line of the street address or equivalent local portion of the shipping address requested by the Cardholder.	3DS Server	Length: Variable, maximum 50 characters JSON Data Type: String	01-APP 02-BRW 03-3RI	01-PA 02-NPA	AReq = C	Required (if available) unless market or regional mandate restricts sending this information.

46	shipAddrLine2	The second line of the street address or equivalent local portion of the shipping address requested by the Cardholder.	3DS Server	Length: Variable, maximum 50 characters JSON Data Type: String	01-APP 02-BRW 03-3RI	01-PA 02-NPA	AReq = C	Required (if available) unless market or regional mandate restricts sending this information.
47	shipAddrLine3	The third line of the street address or equivalent local portion of the shipping address requested by the Cardholder.	3DS Server	Length: Variable, maximum 50 characters JSON Data Type: String	01-APP 02-BRW 03-3RI	01-PA 02-NPA	AReq = C	Required (if available) unless market or regional mandate restricts sending this information.
48	shipAddrPostCode	The ZIP or other postal code of the shipping address requested by the Cardholder.	3DS Server	Length: Variable, maximum 16 characters JSON Data Type: String	01-APP 02-BRW 03-3RI	01-PA 02-NPA	AReq = C	Required (if available) unless market or regional mandate restricts sending this information.
49	shipAddrState	The state or province of the shipping address associated with the card being used for this purchase.	3DS Server	Length: Variable: maximum 3 characters JSON Data Type: String Value accepted: Should be the country subdivision code defined in ISO 3166-2.	01-APP 02-BRW 03-3RI	01-PA 02-NPA	AReq = C	Required (if available) unless market or regional mandate restricts sending this information, or State is not applicable for this country.

50	workPhone	The work phone number provided by the Cardholder.	3DS Server	Length: Variable
				• cc: 1–3 characters
				• subscriber: variable, maximum 15 characters
				JSON Data Type: String
				Values accepted:
				Country Code and Subscriber sections of the number, represented by the following named fields:
				• cc
				• subscriber
				Refer to ITU-E.164 for additional information on format and length.
				Example:
				“workPhone”: {
				“cc”: “1” ,
				“subscriber”:
				“1234567899”
				}
					01-APP
					02-BRW
					03-3RI
						01-PA
						02-NPA
							AReq = C	Required (if available), unless market or regional mandate restricts sending this information.

51	deviceChannel	Indicates the type of channel interface being used to initiate the transaction.	3DS Server	Length: 2 characters JSON Data Type: String Values accepted: • 01 = App-based (APP) • 02 = Browser (BRW) • 03 = 3DS Requestor Initiated (3RI) • 04–79 = Reserved for EMVCo future use (values invalid until defined by EMVCo) • 80–99 = Reserved for DS use	01-APP 02-BRW 03-3RI	01-PA 02-NPA	AReq = R	

52	deviceInfo	Device information gathered by the 3DS SDK from a Consumer Device. This is JSON name/value pairs that as a whole is Base64url encoded. This will be populated by the DS as unencrypted data to the ACS obtained from SDK Encrypted Data.	DS	Length: Variable, maximum 64000 characters JSON Data Type: Object Value accepted: Base64url encoded JSON Object Refer to EMV 3-D Secure SDK Specification for values.	01-APP	01-PA 02-NPA	AReq = C	Required between the DS and ACS but will not be present from 3DS Server to DS.
53	deviceRenderOptions	Defines the SDK UI types that the device supports for displaying specific challenge user interfaces within the SDK. Note: As established in [Req 314], all Device Rendering Options must be supported by the SDK and ACS components.	3DS Server	Length: Variable JSON Data Type: Object Refer to 3.9.3 for data elements to include. Note: Data will be formatted into a JSON object prior to being placed into the Device Rendering Options Supported field of the message.	01-APP	01-PA 02-NPA	AReq = R	
54	dsReferenceNumber	EMVCo-assigned unique identifier to track approved DS.	DS	Length: Variable, maximum 32 characters JSON Data Type: String	01-APP 02-BRW 03-3RI	01-PA 02-NPA	AReq = C	The DS will populate the AReq with this data element prior to passing to the ACS.

55	dsTransID	Universally unique transaction identifier assigned by the DS to identify a single transaction.	DS	Length: 36 characters JSON Data Type: String Value accepted: Canonical format as defined in IETF RFC 4122. May utilise any of the specified versions as long as the output meets specified requirements.	01-APP 02-BRW 03-3RI	01-PA 02-NPA	AReq = C ARes = R	The DS will populate the AReq with this data element prior to passing to the ACS. Required in Error Message if available (e.g. can be obtained from a message or is being generated).
56	dsURL	URL of the DS to which the ACS will send the RReq if a challenge occurs. The ACS is responsible for storing this value for later use in the transaction for sending the RReq to the DS.	DS	Length: Variable, maximum 2048 characters JSON Data Type: String Value accepted: Fully qualified URL. Example: http://server.domainname.com	01-APP 02-BRW	01-PA 02-NPA	AReq = C	Required between the DS and ACS but will not be present from 3DS Server to DS.
57	payTokenInd	A value of True indicates that the transaction was de-tokenised prior to being received by the ACS. This data element will be populated by the system residing in the 3-D Secure domain where the de-tokenisation occurs (i.e., the 3DS	3DS Server DS	JSON Data Type: Boolean Value accepted: • true	01-APP 02-BRW 03-3RI	01-PA 02-NPA	AReq = C	Required if there is a detokenisation of an Account Number.

		Server or the DS). Note: The Boolean value of true is the only valid response for this field when it is present.						
58	purchaseInstalData	Indicates the maximum number of authorisations permitted for instalment payments.	3DS Server	Length: Variable, maximum 3 characters JSON Data Type: String Values accepted: Value shall be greater than 1	01-APP 02-BRW	01-PA 02-NPA	AReq = C	Required if the Merchant and Cardholder have agreed to instalment payments, i.e. if 3DS Requestor Authentication Indicator = 03. Omitted if not an instalment payment authentication.
59	mcc	DS-specific code describing the Merchant's type of business, product or service.	3DS Server	Length: 4 characters JSON Data Type: String This value correlates to the Merchant Category Code as defined by each Payment System or DS.	01-APP 02-BRW 03-3RI	01-PA 02-NPA	01-PA: AReq = R 02-NPA: AReq = O	Optional but strongly recommended to include for 02NPA if the merchant is also the 3DS Requestor.

60	merchantCountryCode	Country Code of the Merchant. This value correlates to the Merchant Country Code as defined by each Payment System or DS.	3DS Server	Length: 3 characters JSON Data Type: String ISO 3166-1 numeric three-digit country code, other than those listed in 3.9.2 . Note: The same value must be used in the authorisation request.	01-APP 02-BRW 03-3RI	01-PA 02-NPA	01-PA: AReq = R 02-NPA: AReq = O	Optional but strongly recommended to include for 02NPA if the merchant is also the 3DS Requestor.
61	merchantName	Merchant name assigned by the Acquirer or Payment System.	3DS Server	Length: Variable, maximum 40 characters JSON Data Type: String Same name used in the authorisation message as defined in ISO 8583.	01-APP 02-BRW 03-3RI	01-PA 02-NPA	01-PA: AReq = R 02-NPA: AReq = O	Optional but strongly recommended to include for 02NPA if the merchant is also the 3DS Requestor.
62	merchantRiskIndicator	Merchant's assessment of the level of fraud risk for the specific authentication for both the cardholder and the authentication being conducted.	3DS Server	Length: Variable JSON Data Type: Object Refer to 3.9.6 for data elements. Note: Data will be formatted into a JSON object prior to being placed into the Device Merchant Risk Indicator field of the message.	01-APP 02-BRW 03-3RI	01-PA 02-NPA	AReq = O	Optional, but strongly recommended to include.
63	messageCategory	Identifies the category of the message for a specific use case.	3DS Server	Length: 2 characters JSON Data Type: String Values accepted: • 01 = PA • 02 = NPA • 03–79 = Reserved for EMVCo future use (values invalid until defined by EMVCo) • 80-99 = Reserved for DS use	01-APP 02-BRW 03-3RI	01-PA 02-NPA	AReq = R	

64	messageExtension	Data necessary to support requirements not otherwise defined in the 3-D Secure message are carried in a Message Extension.	3DS Server	Length: Variable, maximum 81920 bytes JSON Data Type: Array Values accepted: Refer to 3.9.4 for data elements.	01-APP 02-BRW 03-3RI	01-PA 02-NPA	AReq = C	Conditions to be set by each DS.
65	messageType	Identifies the type of message that is passed.	3DS Server 3DS SDK DS ACS	Length: 4 characters JSON Data Type: String Values accepted: • AReq	01-APP 02-BRW 03-3RI	01-PA 02-NPA	AReq = R	
66	messageVersion	Protocol version identifier This shall be the Protocol Version Number of the specification utilised by the system creating this message. The Message Version Number is set by the 3DS Server which originates the protocol with the AReq message. The Message Version Number does not change during a 3DS transaction.	3DS Server	Length: Variable, 5–8 characters JSON Data Type: String Value accepted: Refer to 3.9.1	01-APP 02-BRW 03-3RI	01-PA 02-NPA	AReq = R	

67	notificationURL	Fully qualified URL of the system that receives the CRes message or Error Message. The CRes message is posted by the ACS through the Cardholder browser at the end of the challenge and receipt of the RRes message.	3DS Server	Length: Variable, maximum 256 characters JSON Data Type: String Value accepted: Fully Qualified URL	02-BRW	01-PA 02-NPA	AReq = R	
68	purchaseAmount	Purchase amount in minor units of currency with all punctuation removed. When used in conjunction with the Purchase Currency Exponent field, proper punctuation can be calculated.	3DS Server	Length: Variable, maximum 48 characters JSON Data Type: String Example: If the purchase amount is USD 123.45, element will contain the value 12345.	01-APP 02-BRW	01-PA 02-NPA	01-PA: AReq = R 02-NPA: AReq = C	Required for 02NPA if 3DS Requestor Authentication Indicator = 02 or 03.
69	purchaseCurrency	Currency in which purchase amount is expressed.	3DS Server	Length: 3 characters; Numeric JSON Data Type: String ISO 4217 three-digit currency code, other than those listed in 3.9.2 .	01-APP 02-BRW	01-PA 02-NPA	01-PA: AReq = R 02-NPA: AReq = C	Required for 02NPA if 3DS Requestor Authentication Indicator = 02 or 03.
70	purchaseExponent	Minor units of currency as specified in the ISO 4217 currency exponent. Example: • USD = 2 • Yen = 0	3DS Server	Length: 1 character JSON Data Type: String	01-APP 02-BRW	01-PA 02-NPA	01-PA: AReq = R 02-NPA: AReq = C	Required for 02NPA if 3DS Requestor Authentication Indicator = 02 or 03.

71	purchaseDate	Date and time of the purchase, expressed in UTC.	3DS Server	Length: 14 characters JSON Data Type: String Format accepted: YYYYMMDDHHMMSS	01-APP 02-BRW	01-PA 02-NPA	01-PA: AReq = R 02-NPA: AReq = C	Required for 02NPA if 3DS Requestor Authentication Indicator = 02 or 03.
72	recurringExpiry	Date after which no further authorisations shall be performed.	3DS Server	Length: 8 characters JSON Data Type: String Format accepted: YYYYMMDD	01-APP 02-BRW	01-PA 02-NPA	01-PA: AReq = C 02-NPA: AReq = C	Required if 3DS Requestor Authentication Indicator = 02 or 03.
73	recurringFrequency	Indicates the minimum number of days between authorisations.	3DS Server	Length: Variable, maximum 4 characters JSON Data Type: String	01-APP 02-BRW	01-PA 02-NPA	01-PA: AReq = C 02-NPA: AReq = C	Required if 3DS Requestor Authentication Indicator = 02 or 03.
74	sdkAppID	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.	3DS SDK (sent via 3DS Server)	Length: 36 characters JSON Data Type: String Canonical format as defined in IETF RFC 4122. This may utilise any of the specified versions as long as the output meets specified requirements.	01-APP	01-PA 02-NPA	AReq = R	

75	sdkEncData	JWE Object as defined in 4.1.1 containing data encrypted by the SDK for the DS to decrypt. Note: This element is the only field encrypted in this version of the EMV 3-D Secure specification.	3DS SDK (sent via 3DS Server)	Length: Variable, maximum 64000 characters JSON Data Type: Object	01-APP	01-PA 02-NPA	AReq = C	Required 3DS Server to DS, but will not be present DS to ACS.
76	sdkEphemPubKey	Public key component of the ephemeral key pair generated by the 3DS SDK and used to establish session keys between the 3DS SDK and ACS. In AReq, this data element is present as its own object. In ARes, this data element is contained within the ACS Signed Content JWS Object. See In 4.2.2 for additional detail.	3DS SDK	Length: Variable, maximum 256 characters JSON Data Type: Object JWK	01-APP	01-PA 02-NPA	AReq = R	For ARes, see ACS Signed Content.
77	sdkMaxTimeout	Indicates maximum amount of time (in minutes) for all exchanges.	3DS SDK	Length: 2 characters JSON Data Type: String Values accepted: Greater than or = 05	01-APP	01-PA	AReq = R	

78	sdkReferenceNumber	Identifies the vendor and version for the 3DS SDK that is integrated in a 3DS Requestor App, assigned by EMVCo when the 3DS SDK is approved.	3DS SDK (sent via 3DS Server)	Length: Variable maximum 32 characters JSON Data Type: String	01-APP	01-PA 02-NPA	AReq = R	
79	sdkTransID	Universally unique transaction identifier assigned by the 3DS SDK to identify a single transaction.	3DS SDK (sent via 3DS Server)	Length: 36 characters JSON Data Type: String Canonical format as defined in IETF RFC 4122. This may utilise any of the specified versions as long as the output meets specified requirements.	01-APP	01-PA 02-NPA	AReq = R	Required in Error Message if available (e.g. can be obtained from a message or is being generated).
80	transType	Identifies the type of transaction being authenticated.	3DS Server	Length: 2 characters JSON Data Type: String Values accepted: • 01 = Goods/ Service Purchase • 03 = Check Acceptance • 10 = Account Funding • 11 = Quasi-Cash Transaction • 28 = Prepaid Activation and Load Note: Values derived from the 8583 ISO Standard.	01-APP 02-BRW	01-PA	AReq = C	This field is required in some markets (e.g. for Merchants in Brazil). Otherwise, optional.

3.2 AUTHENTICATION RESPONSE (ARES)

The ARes message is the Issuer's ACS response to the AReq message. It can indicate that the Cardholder has been authenticated, or that further Cardholder interaction is required to complete the authentication. There is only one ARes message per transaction.

Step. Number	Field Name	Description	Source	Length/Format/Values	Device Channel	Message Category	Message Inclusion	Conditional Inclusion
1	threeDSServerTransID	Universally unique transaction identifier assigned by the 3DS Server to identify a single transaction.	3DS Server	Length: 36 characters JSON Data Type: String Value accepted: Canonical format as defined in IETF RFC 4122. May utilise any of the specified versions if the output meets specified requirements.	01-APP 02-BRW 03-3RI	01-PA 02-NPA	ARes = R	Required in Error Message if available (e.g. can be obtained from a message or is being generated).
2	acsChallengeMandated	Indication of whether a challenge is required for the transaction to be authorised due to local/regional mandates or other variable.	ACS	Length: 1 character JSON Data Type: String Values accepted: • Y = Challenge is mandated • N = Challenge is not mandated	01-APP 02-BRW	01-PA 02-NPA	ARes = C	Required if Transaction Status = C.
3	acsOperatorID	DS assigned ACS identifier. Each DS can provide a unique ID to each ACS on an individual basis.	ACS	Length: Variable, maximum 32 characters JSON Data Type: String Value accepted: Any individual DS may impose specific formatting and character requirements on the contents of this field.	01-APP 02-BRW 03-3RI	01-PA 02-NPA	ARes = C	Requirements for the presence of this field are DS specific.
4	acsReferenceNumber	Unique identifier assigned by the EMVCo Secretariat upon Testing and Approval.	ACS	Length: Variable, maximum 32 characters JSON Data Type: String Value accepted: Set by the EMVCo Secretariat.	01-APP 02-BRW 03-3RI	01-PA 02-NPA	ARes = R	

5	acsRenderingType	Identifies the ACS UI Template that the ACS will first present to the consumer.	ACS	JSON Data Type: Object Values accepted: Refer to 3.9.5 for data elements to include. Note: Data will be formatted into a JSON object prior to being placed into the acsRenderingType field of the message.	01-APP	01-PA 02-NPA	ARes = C	For ARes, required if Transaction Status = C.
6	acsSignedContent	Contains the JWS object created by the ACS for the ARes message. See In 4.2.2 for details.	ACS	Length: Variable JSON Data Type: Object Value accepted: The body of JWS object will contain the following data elements : • ACS URL • ACS Ephemeral Public Key (QT) • SDK Ephemeral Public Key (QC)	01-PA 02-NPA	01-PA 02-NPA	ARes = C	Required if the Transaction Status = C.
7	acsTransID	Universally Unique transaction identifier assigned by the ACS to identify a single transaction.	ACS	Length: 36 characters JSON Data Type: String Value accepted: Canonical format as defined in IETF RFC 4122. May utilise any of the specified versions if the output meets specified requirements.	01-APP 02-BRW 03-3RI	01-PA 02-NPA	ARes = R Erro = C	Required in Error Message if available (e.g. can be obtained from a message or is being generated).
8	acsURL	Fully qualified URL of the ACS to be used for the challenge. 01-APP—SDK will send the Challenge Request to this URL 02-BRW—3DS Requestor will post the CReq to this URL via the challenge window	ACS	Length: Variable, maximum 2048 characters JSON Data Type: String Value accepted: Fully qualified URL. For example: https://server.acsdomainname.com	01-APP 02-BRW	01-PA 02-NPA	01-APP: see ACS Signed Content 02-BRW: ARes = C	For 01-APP, see ACS Signed Content. For 02-BRW, required if Transaction Status = C.

		For App-based, this data element is contained within the ACS Signed Content JWS Object For Browser-based, this data element is present as its own object.						
9	authenticationType	Indicates the type of authentication method the Issuer will use to challenge the Cardholder, whether in the ARes message or what was used by the ACS when in the RReq message.	ACS	Length: 2 characters JSON Data Type: String Values accepted: • 01 = Static • 02 = Dynamic • 03 = OOB • 04–79 = Reserved for EMVCo future use (values invalid until defined by EMVCo) • 80–99 = Reserved for DS use	01-APP 02-BRW	01-PA 02-NPA	ARes = C	Required in the ARes message if the Transaction Status = C in the ARes. Required in the RReq message if the Transaction Status = Y or N in the RReq.
10	authenticationValue	Payment System-specific value provided as part of the ACS registration for each supported DS. Authentication Value may be used to provide proof of authentication.	ACS	Length: 28 characters JSON Data Type: String Value accepted: A 20-byte value that has been Base64 encoded, giving a 28-byte result.	01-APP 02-BRW 03-3RI	01-PA 02-NPA	ARes = C	01-PA: Required if Transaction Status = Y or A. Omitted from the RReq message when sent as an abandonment notification. 02-NPA: Conditional based on DS rules.
11	broadInfo	Unstructured information sent between the 3DS Server, the DS and the ACS.	3DS Server DS ACS	Length: 4096 characters JSON Data Type: Object	01-APP 02-BRW 03-3RI	01-PA 02-NPA	ARes = C	Requirements for the presence of this field are DS specific.

12	cardholderInfo	Text provided by the ACS/Issuer to Cardholder during a Frictionless transaction that was not authenticated by the ACS. The Issuer can optionally provide information to Cardholder. For example, “Additional authentication is needed for this transaction, please contact (Issuer Name) at xxx-xxx-xxxx.”	ACS	Length: Variable, maximum 128 characters JSON Data Type: String Note: If field is populated this information shall optionally be displayed to the cardholder by the merchant.	01-APP 02-BRW	01-PA 02-NPA	ARes = O	
13	dsReferenceNumber	EMVCo-assigned unique identifier to track approved DS.	DS	Length: Variable, maximum 32 characters JSON Data Type: String	01-APP 02-BRW 03-3RI	01-PA 02-NPA	ARes = R	The DS will populate the AReq with this data element prior to passing to the ACS.
14	dsTransID	Universally unique transaction identifier assigned by the DS to identify a single transaction.	DS	Length: 36 characters JSON Data Type: String Value accepted: Canonical format as defined in IETF RFC 4122. May utilise any of the specified versions as long as the output meets specified requirements.	01-APP 02-BRW 03-3RI	01-PA 02-NPA	ARes = R Erro = C	The DS will populate the AReq with this data element prior to passing to the ACS. Required in Error Message if available (e.g. can be obtained from a message or is being generated).
15	eci	Payment System-specific value provided by the ACS to indicate the results of the attempt to authenticate the Cardholder.	ACS	Length: 2 characters JSON Data Type: String Values accepted: Payment System specific	01-APP 02-BRW 03-3RI	01-PA 02-NPA	ARes = C	The requirements for the presence of this field are DS specific.

16	messageExtension	Data necessary to support requirements not otherwise defined in the 3-D Secure message are carried in a Message Extension.	3DS Server	Length: Variable, maximum 81920 bytes JSON Data Type: Array Values accepted: Refer to 3.9.4 for data elements.	01-APP 02-BRW 03-3RI	01-PA 02-NPA	ARes = C	Conditions to be set by each DS.
17	messageType	Identifies the type of message that is passed.	3DS Server 3DS SDK DS ACS	Length: 4 characters JSON Data Type: String Values accepted: • ARes	01-APP 02-BRW 03-3RI	01-PA 02-NPA	ARes = R	
18	messageVersion	Protocol version identifier This shall be the Protocol Version Number of the specification utilised by the system creating this message. The Message Version Number is set by the 3DS Server which originates the protocol with the AReq message. The Message Version Number does not change during a 3DS transaction.	3DS Server	Length: Variable, 5–8 characters JSON Data Type: String Value accepted: Refer to 3.9.1	01-APP 02-BRW 03-3RI	01-PA 02-NPA	ARes = R Erro = R	
19	sdkTransID	Universally unique transaction identifier assigned by the 3DS SDK to identify a single transaction.	3DS SDK (sent via 3DS Server)	Length: 36 characters JSON Data Type: String Canonical format as defined in IETF RFC 4122. This may utilise any of the specified versions as long as the output meets specified requirements.	01-APP	01-PA 02-NPA	ARes = R Erro = C	Required in Error Message if available (e.g. can be obtained from a message or is being generated).

20	transStatus	Indicates whether a transaction qualifies as an authenticated transaction or account verification. Note: The CRes message can contain only a value of Y or N.	ACS DS	Length: 1 character JSON Data Type: String • Y = Authentication/ Account Verification Successful • N = Not Authenticated /Account Not Verified; Transaction denied • U = Authentication/ Account Verification Could Not Be Performed; Technical or other problem, as indicated in ARes or RReq • A = Attempts Processing Performed; Not Authenticated/Verified , but a proof of attempted authentication/verificat ion is provided • C = Challenge Required; Additional authentication is required using the CReq/CRes • R = Authentication/ Account Verification Rejected; Issuer is rejecting authentication/verificat ion and request that authorisation not be attempted.	01-APP 02-BRW 03-3RI	01-PA 02-NPA	01-PA: ARes = R 02-NPA: ARes = C	For the CRes, only present in the final CRes message.

21	transStatusReason	Provides information on why the Transaction Status field has the specified value.	ACS DS	Length: 2 characters JSON Data Type: String Values accepted: • 01 = Card authentication failed • 02 = Unknown Device • 03 = Unsupported Device • 04 = Exceeds authentication frequency limit • 05 = Expired card • 06 = Invalid card number • 07 = Invalid transaction • 08 = No Card record • 09 = Security failure • 10 = Stolen card • 11 = Suspected fraud • 12 = Transaction not permitted to cardholder • 13 = Cardholder not enrolled in service • 14 = Transaction timed out at the ACS • 15 = Low confidence • 16 = Medium confidence • 17 = High confidence • 18 = Very High confidence • 19 = Exceeds ACS maximum challenges • 20 = Non-Payment transaction not supported • 21 = 3RI transaction not supported • 22–79 = Reserved for EMVCo future use (values invalid until defined by EMVCo) • 80–99 = Reserved for DS use	01-APP 02-BRW 03-3RI	01-PA 02-NPA	ARes = C	For 01-PA, required if the Transaction Status field = N, U, or R. For 02-NPA, Conditional as defined by the DS.

3.3 CHALLENGE REQUEST (CREQ)

The CReq message initiates Cardholder interaction in a Challenge Flow and can be used to carry authentication data from the Cardholder.

• App-based—The CReq message is sent by the 3DS SDK. There are two or more CReq messages per challenge as multiple back-and-forth attempts between the ACS and the Cardholder may be required to complete the authentication.

• Browser-based—The CReq message is sent by the 3DS Server. There is only one CReq message per challenge.

Step. Number	Field Name	Description	Source	Length/Format/Values	Device Channel	Message Category	Message Inclusion	Conditional Inclusion
1	threeDSServerTransID	Universally unique transaction identifier assigned by the 3DS Server to identify a single transaction.	3DS Server	Length: 36 characters JSON Data Type: String Value accepted: Canonical format as defined in IETF RFC 4122. May utilise any of the specified versions if the output meets specified requirements.	01-APP 02-BRW 03-3RI	01-PA 02-NPA	CReq = R Erro = C	Required in Error Message if available (e.g. can be obtained from a message or is being generated).
2	acsTransID	Universally Unique transaction identifier assigned by the ACS to identify a single transaction.	ACS	Length: 36 characters JSON Data Type: String Value accepted: Canonical format as defined in IETF RFC 4122. May utilise any of the specified versions if the output meets specified requirements.	01-APP 02-BRW 03-3RI	01-PA 02-NPA	CReq = R Erro = C	Required in Error Message if available (e.g. can be obtained from a message or is being generated).
3	challengeCancel	Indicator informing the ACS and the DS that the authentication has been cancelled.	3DS SDK ACS	Length: 2 characters JSON Data Type: String Values accepted: • 01 = Cardholder selected “Cancel” • 02 = 3DS Requestor cancelled Authentication. • 03 = Transaction Abandoned • 04 = Transaction Timed	01-APP 02-BRW	01-PA 02-NPA	CReq = C	Required in CReq for 01-APP if the authentication transaction was cancelled by user interaction with the cancelation button in the UI or for other reasons as

				Out at ACS— other timeouts • 05 = Transaction Timed Out at ACS— First CReq not received by ACS • 06 = Transaction Error • 07 = Unknown • 08-79 = Reserved for future EMVCo use (values invalid until defined by EMVCo) • 80–99 = Reserved for future DS use				indicated. Required in the RReq if the ACS identifies that the authentication transaction was cancelled for reasons as indicated.
4	challengeDataEntry	Contains the data that the Cardholder entered into the Native UI text field. Note: ACS UI Type = 05 is not supported.	3DS SDK	Length: Variable, maximum 45 characters JSON Data Type: String	01-APP	01-PA 02-NPA	CReq = C	Required when ACS UI Type = 01, 02, or 03 and challenge data has been entered into the UI.
5	challengeHTMLDataEntry	Data that the Cardholder entered into the HTML UI. Note: ACS UI Types 01, 02, 03, and 04 are not supported.	3DS SDK	Length: Variable, maximum 256 characters JSON Data Type: String	01-APP	01-PA 02-NPA	CReq = C	Required when ACS UI Type = 05 and challenge data has been entered into the UI.
6	challengeWindowSize	Dimensions of the challenge window that has been displayed to the Cardholder. The ACS shall reply with content that is formatted to appropriately render in this window to provide the best possible user experience. Preconfigured sizes are width x height in pixels of the window displayed in the Cardholder browser window.	3DS Requestor	Length: 2 characters JSON Data Type: String Value accepted: • 01 = 250 x 400 • 02 = 390 x 400 • 03 = 500 x 600 • 04 = 600 x 400 • 05 = Full screen	02-BRW	01-PA 02-NPA	CReq = R	

7	messageExtension	Data necessary to support requirements not otherwise defined in the 3-D Secure message are carried in a Message Extension.	3DS Server	Length: Variable, maximum 81920 bytes JSON Data Type: Array Values accepted: Refer to 3.9.4 for data elements.	01-APP 02-BRW 03-3RI	01-PA 02-NPA	CReq = C	Conditions to be set by each DS.
8	messageType	Identifies the type of message that is passed.	3DS Server 3DS SDK DS ACS	Length: 4 characters JSON Data Type: String Values accepted: • CReq	01-APP 02-BRW 03-3RI	01-PA 02-NPA	CReq = R Erro = R	
9	messageVersion	Protocol version identifier This shall be the Protocol Version Number of the specification utilised by the system creating this message. The Message Version Number is set by the 3DS Server which originates the protocol with the AReq message. The Message Version Number does not change during a 3DS transaction.	3DS Server	Length: Variable, 5–8 characters JSON Data Type: String Value accepted: Refer to 3.9.1	01-APP 02-BRW 03-3RI	01-PA 02-NPA	CReq = R Erro = R	
10	oobContinue	Indicator notifying the ACS that Cardholder has completed the authentication as requested by selecting the Continue button in an Out-of-Band (OOB) authentication method. Note: The Boolean value of true is the only valid response for this field when it is present.	3DS SDK	JSON Data Type: Boolean Value accepted: • true	01-APP	01-PA 02-NPA	CReq = C	Required when ACS UI Type = 04 when the Cardholder has selected that option on the device.

11	resendChallenge	Indicator to the ACS to resend the challenge information code to the Cardholder.	3DS SDK	Length: 1 character JSON Data Type: String Values accepted: • Y = Resend • N = Do not Resend	01-APP	01-PA 02-NPA	CReq = C	Required for Native UI if the Cardholder is requesting the ACS to resend challenge information.
12	sdkTransID	Universally unique transaction identifier assigned by the 3DS SDK to identify a single transaction.	3DS SDK (sent via 3DS Server)	Length: 36 characters JSON Data Type: String Canonical format as defined in IETF RFC 4122. This may utilise any of the specified versions as long as the output meets specified requirements.	01-APP	01-PA 02-NPA	CReq = R Erro = C	Required in Error Message if available (e.g. can be obtained from a message or is being generated).
13	sdkCounterStoA	Counter used as a security measure in the 3DS SDK to ACS secure channel.	3DS SDK	Length: 3 characters JSON Data Type: String	01-APP	01-PA 02-NPA	01-PA: CReq = R 02-NPA CReq = R	

3.4 CHALLENGE RESPONSE (CRES)

The CRes message is the ACS response to the CReq message. It can indicate the result of the Cardholder authentication or, in the case of an App-based model, also signal that further Cardholder interaction is required to complete the authentication.

• App-based—Elements of the CRes message provide the necessary data for the 3DS SDK to generate and display the user interface (UI) for the challenge. There are two or more CRes messages per transaction to complete Cardholder authentication.

• Browser-based—The CRes message contains the authentication result and completes the Cardholder challenge. There is only one CRes message per challenge.

Step. Number	Field Name	Description	Source	Length/Format/Values	Device Channel	Message Category	Message Inclusion	Conditional Inclusion
1	threeDSServerTransID	Universally unique transaction identifier assigned by the 3DS Server to identify a single transaction.	3DS Server	Length: 36 characters JSON Data Type: String Value accepted: Canonical format as defined in IETF RFC 4122. May utilise any of the specified versions if the output meets specified requirements.	01-APP 02-BRW 03-3RI	01-PA 02-NPA	CRes = R Erro = C	Required in Error Message if available (e.g. can be obtained from a message or is being generated).
2	acsCounterAtoS	Counter used as a security measure in the ACS to 3DS SDK secure channel.	ACS	Length: 3 characters JSON Data Type: String	01-APP	01-PA 02-NPA	01-PA: CRes = R 02-NPA CRes = R	
3	acsTransID	Universally Unique transaction identifier assigned by the ACS to identify a single transaction.	ACS	Length: 36 characters JSON Data Type: String Value accepted: Canonical format as defined in IETF RFC 4122. May utilise any of the specified versions if the output meets specified requirements.	01-APP 02-BRW 03-3RI	01-PA 02-NPA	CRes = R Erro = C	Required in Error Message if available (e.g. can be obtained from a message or is being generated).
4	acsHTML	HTML provided by the ACS in the CRes message. Utilised when HTML is specified in the ACS UI Type during the Cardholder challenge.	ACS	Length: Variable, maximum 100KB JSON Data Type: String Value accepted: Base64url encoded HTML This value will be Base64url encoded prior to being placed into the CRes message.	01-APP	01-PA 02-NPA	CRes = C	Conditional upon selection of the ACS UI Type = 5 (HTML) by the ACS.

5	acsUiType	User interface type that the 3DS SDK will render, which includes the specific data mapping and requirements.	ACS	Length: 2 characters JSON Data Type: String Values accepted • 01 = Text • 02 = Single Select • 03 = Multi Select • 04 = OOB • 05 = HTML • 06–79 = Reserved for EMVCo future use (values invalid until defined by EMVCo) • 80–99 = Reserved for DS use	01-APP	01-PA 02-NPA	CRes = R	
6	challengeAddInfo	Text provided by the ACS/Issuer to Cardholder during OOB authentication to replace Challenge Information Text and Challenge Information Text Indicator in the OOB Template.	ACS	Length: Variable, maximum 256 characters JSON Data Type: String If field is populated this information is displayed to the Cardholder by the SDK when the 3DS Requestor App is brought to the foreground.	01-APP	01-PA 02-NPA	CRes = O	
7	challengeCompletionInd	Indicator of the state of the ACS challenge cycle and whether the challenge has completed or will require additional messages. Shall be populated in all CRes messages to convey the current state of the transaction. Note: If set to Y, the ACS will populate the Transaction Status in the CRes message.	ACS	Length: 1 character JSON Data Type: String Values accepted: • Y = Challenge completed and no further challenge message exchanges are required • N = Challenge not completed and there shall be additional challenge message exchanges required	01-APP	01-PA 02-NPA	CRes = R	

8	challengeInfoHeader	Header text that for the challenge information screen that is being presented.	ACS	Length: Variable, maximum 45 characters JSON Data Type: String If field is populated this information is displayed to the cardholder.	01-APP	01-PA 02-NPA	CRes = O	
9	challengeInfoLabel	Label to modify the Challenge Data Entry field provided by the Issuer.	ACS	Length: Variable, maximum 45 characters JSON Data Type: String If field is populated this information is displayed to the cardholder.	01-APP	01-PA 02-NPA	CRes = O	
10	challengeInfoText	Text provided by the ACS/Issuer to Cardholder during the Challenge Message exchange.	ACS	Length: Variable, maximum 350 characters JSON Data Type: String If field is populated this information is displayed to the cardholder. Note: Carriage return is supported in this data element and is represented by an “\n”.	01-APP	01-PA 02-NPA	CRes = O	
11	challengeInfoTextIndicator	Indicates when the Issuer/ACS would like a warning icon or similar visual indicator to draw attention to the “Challenge Information Text” that is being displayed.	ACS	Length: 1 JSON Data Type: String Value accepted: • Y = Display indicator • N = Do not display indicator Note: If field is populated this information is displayed to the cardholder.	01-APP	01-PA 02-NPA	CRes = O	
12	challengeSelectInfo	Selection information that will be presented to the Cardholder if the option is single or multi-select. The variables will be sent in a JSON Array and parsed by the SDK for display in the user interface.	ACS	Length: Variable, each name/value pair maximum 45 characters JSON Data Type: Array Note: If field is populated this information is displayed to	01-APP	01-PA 02-NPA	CRes = O	

		Example: “challengeSelectInfo”: [{"mobile": "**** **** 123"}, {"email": "s******k**@g***.com"}]						
13	expandInfoLabel	Label displayed to the Cardholder for the content in Expandable Information Text.	ACS	Length: Variable, maximum 45 characters JSON Data Type: String	01-APP	01-PA 02-NPA	CRes = O	
14	expandInfoText	Text provided by the Issuer from the ACS to be displayed to the Cardholder for additional information and the format will be an expandable text field.	ACS	Length: Variable, maximum 256 characters JSON Data Type: String Note: Carriage return is supported in this data element and is represented by an “\n”.	01-APP	01-PA 02-NPA	CRes = O	
15	issuerImage	Sent in the initial CRes message from the ACS to the 3DS SDK to provide the URL(s) of the Issuer logo or image to be used in the Native UI.	ACS	Format: JSON Object Values accepted: Refer to 3.9.7 for data elements.	01-APP	01-PA 02-NPA	CRes = C	Presence of this field is Payment System specific.
16	messageExtension	Data necessary to support requirements not otherwise defined in the 3-D Secure message are carried in a Message Extension.	3DS Server	Length: Variable, maximum 81920 bytes JSON Data Type: Array Values accepted: Refer to 3.9.4 for data elements.	01-APP 02-BRW 03-3RI	01-PA 02-NPA	CRes = C	Conditions to be set by each DS.
17	messageType	Identifies the type of message that is passed.	3DS Server 3DS SDK DS ACS	Length: 4 characters JSON Data Type: String Values accepted: • CRes	01-APP 02-BRW 03-3RI	01-PA 02-NPA	CRes = R Erro = R	

18	messageVersion	Protocol version identifier This shall be the Protocol Version Number of the specification utilised by the system creating this message. The Message Version Number is set by the 3DS Server which originates the protocol with the AReq message. The Message Version Number does not change during a 3DS transaction.	3DS Server	Length: Variable, 5–8 characters JSON Data Type: String Value accepted: Refer to 3.9.1	01-APP 02-BRW 03-3RI	01-PA 02-NPA	CRes = R Erro = R	
19	oobAppURL	Mobile Deep link to an authentication app used in the out-of-band authentication. The App URL will open the appropriate location within the authentication app.	ACS	Length: Variable, maximum 256 characters JSON Data Type: String Value accepted: Fully Qualified URL	01-APP	01-PA 02-NPA	CRes = O	Note: this element has been defined to support future enhancements to the OOB message flow. The 3DS SDK will not perform any processing of the OOP App URL in this version of the specification.
20	oobAppLabel	Label to be displayed for the link to the OOB App URL. For example: “oobAppLabel”: “Click here to open Your Bank App”	ACS	Length: Variable, maximum 45 characters JSON Data Type: String	01-APP	01-PA 02-NPA	CRes = C	
21	oobContinueLabel	Label to be used in the UI for the button that the user selects when they have completed the OOB authentication.	ACS	Length: Variable, maximum 45 characters JSON Data Type: String	01-APP	01-PA 02-NPA	CRes = C	Required when ACS UI Type = 04 in when the Cardholder has selected that option on the device.

22	psImage	Sent in the initial CRes message from the ACS to the 3DS SDK to provide the URL(s) of the DS or Payment System logo or image to be used in the Native UI.	ACS	JSON Data Type: Object Values accepted: Refer to 3.9.8 for data elements.	01-APP	01-PA 02-NPA	CRes = C	Presence of this field are Payment System specific.
23	resendInformationLabel	Label to be used in the UI for the button that the user selects when they would like to have the authentication information resent.	ACS	Length: Variable maximum 45 characters JSON Data Type: String	01-APP	01-PA 02-NPA	CRes = C	Required for Native UI if the ACS is allowing the Cardholder to request resending authentication information.
24	sdkTransID	Universally unique transaction identifier assigned by the 3DS SDK to identify a single transaction.	3DS SDK (sent via 3DS Server)	Length: 36 characters JSON Data Type: String Canonical format as defined in IETF RFC 4122. This may utilise any of the specified versions as long as the output meets specified requirements.	01-APP	01-PA 02-NPA	CRes = R Erro = C	Required in Error Message if available (e.g. can be obtained from a message or is being generated).
25	submitAuthenticationLabel	Label to be used in the UI for the button that the user selects when they have completed the authentication. Note: This is not used for OOB authentication.	ACS	Length: Variable maximum 45 characters JSON Data Type: String	01-APP	01-PA 02-NPA	CRes = C	Required when the ACS UI Type = 01, 02, or 03.
26	transStatus	Indicates whether a transaction qualifies as an authenticated transaction or account verification. Note: The CRes message can contain only a value of Y or N.	ACS DS	Length: 1 character JSON Data Type: String • Y = Authentication/ Account Verification Successful • N = Not Authenticated /Account Not Verified; Transaction denied • U = Authentication/ Account Verification Could Not Be	01-APP 02-BRW 03-3RI	01-PA 02-NPA	01-PA: CRes = C 02-NPA: CRes = C	For the CRes, only present in the final CRes message.

				Performed; Technical or other problem, as indicated in ARes or RReq • A = Attempts Processing Performed; Not Authenticated/Verified , but a proof of attempted authentication/verificat ion is provided • C = Challenge Required; Additional authentication is required using the CReq/CRes • R = Authentication/ Account Verification Rejected; Issuer is rejecting authentication/verificat ion and request that authorisation not be attempted.				
27	whyInfoLabel	Label to be displayed to the Cardholder for the "why" information section.	ACS	Length: Variable, maximum 45 characters JSON Data Type: String	01-APP	01-PA 02-NPA	CRes = O	
28	whyInfoText	Text provided by the Issuer to be displayed to the Cardholder to explain why the Cardholder is being asked to perform the authentication task.	ACS	Length: Variable, maximum 256 characters JSON Data Type: String Note: Carriage return is supported in this data element and is represented by an “\n”.	01-APP	01-PA 02-NPA	CRes = O	

The final CRes message sent upon completion of the challenge from the ACS.

Step. Number	Field Name	Description	Source	Length/Format/Values	Device Channel	Message Category	Message Inclusion	Conditional Inclusion

1	threeDSServerTransID	Universally unique transaction identifier assigned by the 3DS Server to identify a single transaction.	3DS Server	Length: 36 characters JSON Data Type: String Value accepted: Canonical format as defined in IETF RFC 4122. May utilise any of the specified versions if the output meets specified requirements.	01-APP 02-BRW 03-3RI	01-PA 02-NPA	CRes = R Erro = C	Required in Error Message if available (e.g. can be obtained from a message or is being generated).
2	acsCounterAtoS	Counter used as a security measure in the ACS to 3DS SDK secure channel.	ACS	Length: 3 characters JSON Data Type: String	01-APP	01-PA 02-NPA	01-PA: CRes = R 02-NPA CRes = R	
3	acsTransID	Universally Unique transaction identifier assigned by the ACS to identify a single transaction.	ACS	Length: 36 characters JSON Data Type: String Value accepted: Canonical format as defined in IETF RFC 4122. May utilise any of the specified versions if the output meets specified requirements.	01-APP 02-BRW 03-3RI	01-PA 02-NPA	CRes = R Erro = C	Required in Error Message if available (e.g. can be obtained from a message or is being generated).
4	challengeCompletionInd	Indicator of the state of the ACS challenge cycle and whether the challenge has completed or will require additional messages. Shall be populated in all CRes messages to convey the current state of the transaction. Note: If set to Y, the ACS will populate the Transaction Status in the CRes message.	ACS	Length: 1 character JSON Data Type: String Values accepted: • Y = Challenge completed and no further challenge message exchanges are required • N = Challenge not completed and there shall be additional challenge message exchanges required	01-APP	01-PA 02-NPA	CRes = R	
5	messageExtension	Data necessary to support requirements not otherwise defined in the 3-D Secure message are carried in a Message Extension.	3DS Server	Length: Variable, maximum 81920 bytes JSON Data Type: Array Values accepted: Refer to 3.9.4 for data elements.	01-APP 02-BRW 03-3RI	01-PA 02-NPA	CRes = C	Conditions to be set by each DS.

6	messageType	Identifies the type of message that is passed.	3DS Server 3DS SDK DS ACS	Length: 4 characters JSON Data Type: String Values accepted: • CRes	01-APP 02-BRW 03-3RI	01-PA 02-NPA	CRes = R Erro = R	
7	messageVersion	Protocol version identifier This shall be the Protocol Version Number of the specification utilised by the system creating this message. The Message Version Number is set by the 3DS Server which originates the protocol with the AReq message. The Message Version Number does not change during a 3DS transaction.	3DS Server	Length: Variable, 5–8 characters JSON Data Type: String Value accepted: Refer to 3.9.1	01-APP 02-BRW 03-3RI	01-PA 02-NPA	CRes = R Erro = R	
8	sdkTransID	Universally unique transaction identifier assigned by the 3DS SDK to identify a single transaction.	3DS SDK (sent via 3DS Server)	Length: 36 characters JSON Data Type: String Canonical format as defined in IETF RFC 4122. This may utilise any of the specified versions as long as the output meets specified requirements.	01-APP	01-PA 02-NPA	CRes = R Erro = C	Required in Error Message if available (e.g. can be obtained from a message or is being generated).
9	transStatus	Indicates whether a transaction qualifies as an authenticated transaction or account verification. Note: The CRes message can contain only a value of Y or N.	ACS DS	Length: 1 character JSON Data Type: String • Y = Authentication/ Account Verification Successful • N = Not Authenticated /Account Not Verified; Transaction denied • U = Authentication/ Account Verification Could Not Be Performed; Technical or other problem, as indicated	01-APP 02-BRW 03-3RI	01-PA 02-NPA	01-PA: CRes = C 02-NPA: CRes = C	For the CRes, only present in the final CRes message.

				in ARes or RReq • A = Attempts Processing Performed; Not Authenticated/Verified , but a proof of attempted authentication/verificat ion is provided • C = Challenge Required; Additional authentication is required using the CReq/CRes • R = Authentication/ Account Verification Rejected; Issuer is rejecting authentication/verificat ion and request that authorisation not be attempted.				

3.5 RESULT REQUEST (RREQ)

The RReq message communicates the results of the authentication. The message is sent by the ACS through the DS to the 3DS Server. There is only one RReq message per authentication. The RReq message is present only in an authentication requiring a Cardholder challenge.

Step. Number	Field Name	Description	Source	Length/Format/Values	Device Channel	Message Category	Message Inclusion	Conditional Inclusion
1	threeDSServerTransID	Universally unique transaction identifier assigned by the 3DS Server to identify a single transaction.	3DS Server	Length: 36 characters JSON Data Type: String Value accepted: Canonical format as defined in IETF RFC 4122. May utilise any of the specified versions if the output meets specified requirements.	01-APP 02-BRW 03-3RI	01-PA 02-NPA	RReq = R	Required in Error Message if available (e.g. can be obtained from a message or is being generated).

2	acsTransID	Universally Unique transaction identifier assigned by the ACS to identify a single transaction.	ACS	Length: 36 characters JSON Data Type: String Value accepted: Canonical format as defined in IETF RFC 4122. May utilise any of the specified versions if the output meets specified requirements.	01-APP 02-BRW 03-3RI	01-PA 02-NPA	RReq = R	Required in Error Message if available (e.g. can be obtained from a message or is being generated).
3	acsRenderingType	Identifies the ACS UI Template that the ACS will first present to the consumer.	ACS	JSON Data Type: Object Values accepted: Refer to 3.9.5 for data elements to include. Note: Data will be formatted into a JSON object prior to being placed into the acsRenderingType field of the message.	01-APP	01-PA 02-NPA	RReq = R	For ARes, required if Transaction Status = C.
4	authenticationMethod	Authentication approach that the ACS used to authenticate the Cardholder for this specific transaction. Note: This is in the RReq message from the ACS only. It is not passed to the 3DS Server URL.	ACS	Length: 2 characters JSON Data Type: String Values accepted: • 01 = Static Passcode • 02 = SMS OTP • 03 = Key fob or EMV card reader OTP • 04 = App OTP • 05 = OTP Other • 06 = KBA • 07 = OOB Biometrics • 08 = OOB Login • 09 = OOB Other • 10 = Other • 11–79 = Reserved for future EMVCo use (values invalid until defined by EMVCo) • 80–99 = Reserved for future DS use	01-APP 02-BRW	01-PA 02-NPA	RReq = C	Required to be sent by the ACS. This field is not present in the RReq message from the DS to the 3DS Server URL.

5	authenticationType	Indicates the type of authentication method the Issuer will use to challenge the Cardholder, whether in the ARes message or what was used by the ACS when in the RReq message.	ACS	Length: 2 characters JSON Data Type: String Values accepted: • 01 = Static • 02 = Dynamic • 03 = OOB • 04–79 = Reserved for EMVCo future use (values invalid until defined by EMVCo) • 80–99 = Reserved for DS use	01-APP 02-BRW	01-PA 02-NPA	RReq = C	Required in the ARes message if the Transaction Status = C in the ARes. Required in the RReq message if the Transaction Status = Y or N in the RReq.
6	authenticationValue	Payment System-specific value provided as part of the ACS registration for each supported DS. Authentication Value may be used to provide proof of authentication.	ACS	Length: 28 characters JSON Data Type: String Value accepted: A 20-byte value that has been Base64 encoded, giving a 28-byte result.	01-APP 02-BRW 03-3RI	01-PA 02-NPA	RReq = C	01-PA: Required if Transaction Status = Y or A. Omitted from the RReq message when sent as an abandonment notification. 02-NPA: Conditional based on DS rules.
7	challengeCancel	Indicator informing the ACS and the DS that the authentication has been cancelled.	3DS SDK ACS	Length: 2 characters JSON Data Type: String Values accepted: • 01 = Cardholder selected “Cancel” • 02 = 3DS Requestor cancelled Authentication. • 03 = Transaction Abandoned • 04 = Transaction Timed Out at ACS— other timeouts • 05 = Transaction Timed Out at ACS— First CReq not received by ACS • 06 = Transaction Error • 07 = Unknown	01-APP 02-BRW	01-PA 02-NPA	RReq = C	Required in CReq for 01-APP if the authentication transaction was cancelled by user interaction with the cancelation button in the UI or for other reasons as indicated. Required in the RReq if the ACS identifies that the authentication transaction was

				• 08-79 = Reserved for future EMVCo use (values invalid until defined by EMVCo) • 80–99 = Reserved for future DS use				cancelled for reasons as indicated.
8	dsTransID	Universally unique transaction identifier assigned by the DS to identify a single transaction.	DS	Length: 36 characters JSON Data Type: String Value accepted: Canonical format as defined in IETF RFC 4122. May utilise any of the specified versions as long as the output meets specified requirements.	01-APP 02-BRW 03-3RI	01-PA 02-NPA	RReq = R Erro = C	The DS will populate the AReq with this data element prior to passing to the ACS. Required in Error Message if available (e.g. can be obtained from a message or is being generated).
9	eci	Payment System-specific value provided by the ACS to indicate the results of the attempt to authenticate the Cardholder.	ACS	Length: 2 characters JSON Data Type: String Values accepted: Payment System specific	01-APP 02-BRW 03-3RI	01-PA 02-NPA	RReq = C	The requirements for the presence of this field are DS specific.
10	interactionCounter	Indicates the number of authentication cycles attempted by the Cardholder. Value to be tracked by the ACS.	ACS	Length: 2 characters JSON Data Type: String	01-APP 02-BRW	01-PA 02-NPA	RReq = R	
11	messageCategory	Identifies the category of the message for a specific use case.	3DS Server	Length: 2 characters JSON Data Type: String Values accepted: • 01 = PA • 02 = NPA • 03–79 = Reserved for EMVCo future use (values invalid until defined by EMVCo) • 80-99 = Reserved for DS use	01-APP 02-BRW 03-3RI	01-PA 02-NPA	RReq = R	

12	messageExtension	Data necessary to support requirements not otherwise defined in the 3-D Secure message are carried in a Message Extension.	3DS Server	Length: Variable, maximum 81920 bytes JSON Data Type: Array Values accepted: Refer to 3.9.4 for data elements.	01-APP 02-BRW 03-3RI	01-PA 02-NPA	RReq = C	Conditions to be set by each DS.
13	messageType	Identifies the type of message that is passed.	3DS Server 3DS SDK DS ACS	Length: 4 characters JSON Data Type: String Values accepted: • RReq	01-APP 02-BRW 03-3RI	01-PA 02-NPA	RReq = R Erro = R	
14	messageVersion	Protocol version identifier This shall be the Protocol Version Number of the specification utilised by the system creating this message. The Message Version Number is set by the 3DS Server which originates the protocol with the AReq message. The Message Version Number does not change during a 3DS transaction.	3DS Server	Length: Variable, 5–8 characters JSON Data Type: String Value accepted: Refer to 3.9.1	01-APP 02-BRW 03-3RI	01-PA 02-NPA	RReq = R Erro = R	
15	transStatus	Indicates whether a transaction qualifies as an authenticated transaction or account verification. Note: The CRes message can contain only a value of Y or N.	ACS DS	Length: 1 character JSON Data Type: String • Y = Authentication/ Account Verification Successful • N = Not Authenticated /Account Not Verified; Transaction denied • U = Authentication/ Account Verification Could Not Be Performed; Technical or other problem, as indicated in ARes or RReq	01-APP 02-BRW 03-3RI	01-PA 02-NPA	01-PA: RReq = R 02-NPA: RReq = C	For the CRes, only present in the final CRes message.

				• A = Attempts Processing Performed; Not Authenticated/Verified , but a proof of attempted authentication/verificat ion is provided • C = Challenge Required; Additional authentication is required using the CReq/CRes • R = Authentication/ Account Verification Rejected; Issuer is rejecting authentication/verificat ion and request that authorisation not be attempted.				
16	transStatusReason	Provides information on why the Transaction Status field has the specified value.	ACS DS	Length: 2 characters JSON Data Type: String Values accepted: • 01 = Card authentication failed • 02 = Unknown Device • 03 = Unsupported Device • 04 = Exceeds authentication frequency limit • 05 = Expired card • 06 = Invalid card number • 07 = Invalid transaction • 08 = No Card record • 09 = Security failure • 10 = Stolen card • 11 = Suspected fraud • 12 = Transaction not permitted to cardholder • 13 = Cardholder not enrolled in service • 14 = Transaction timed out	01-APP 02-BRW 03-3RI	01-PA 02-NPA	RReq = C	For 01-PA, required if the Transaction Status field = N, U, or R. For 02-NPA, Conditional as defined by the DS.

				at the ACS • 15 = Low confidence • 16 = Medium confidence				

3.6 RESULT RESPONSE (RRES)

The RRes message acknowledges receipt of the RReq message. The message is sent by the 3DS Server through the DS to the ACS. There is only one RRes message per authentication. The RRes message is present only in an authentication requiring a Cardholder challenge.

Step. Number	Field Name	Description	Source	Length/Format/Values	Device Channel	Message Category	Message Inclusion	Conditional Inclusion
1	threeDSServerTransID	Universally unique transaction identifier assigned by the 3DS Server to identify a single transaction.	3DS Server	Length: 36 characters JSON Data Type: String Value accepted: Canonical format as defined in IETF RFC 4122. May utilise any of the specified versions if the output meets specified requirements.	01-APP 02-BRW 03-3RI	01-PA 02-NPA	RRes = R Erro = C	Required in Error Message if available (e.g. can be obtained from a message or is being generated).
2	acsTransID	Universally Unique transaction identifier assigned by the ACS to identify a single transaction.	ACS	Length: 36 characters JSON Data Type: String Value accepted: Canonical format as defined in IETF RFC 4122. May utilise any of the specified versions if the output meets specified requirements.	01-APP 02-BRW 03-3RI	01-PA 02-NPA	RRes = R Erro = C	Required in Error Message if available (e.g. can be obtained from a message or is being generated).

3	dsTransID	Universally unique transaction identifier assigned by the DS to identify a single transaction.	DS	Length: 36 characters JSON Data Type: String Value accepted: Canonical format as defined in IETF RFC 4122. May utilise any of the specified versions as long as the output meets specified requirements.	01-APP 02-BRW 03-3RI	01-PA 02-NPA	RRes = R Erro = C	The DS will populate the AReq with this data element prior to passing to the ACS. Required in Error Message if available (e.g. can be obtained from a message or is being generated).
4	messageExtension	Data necessary to support requirements not otherwise defined in the 3-D Secure message are carried in a Message Extension.	3DS Server	Length: Variable, maximum 81920 bytes JSON Data Type: Array Values accepted: Refer to 3.9.4 for data elements.	01-APP 02-BRW 03-3RI	01-PA 02-NPA	RRes = C	Conditions to be set by each DS.
5	messageType	Identifies the type of message that is passed.	3DS Server 3DS SDK DS ACS	Length: 4 characters JSON Data Type: String Values accepted: • AReq • ARes • CReq • CRes • PReq • PRes • RReq • RRes • Erro	01-APP 02-BRW 03-3RI	01-PA 02-NPA	RRes = R Erro = R	
6	messageVersion	Protocol version identifier This shall be the Protocol Version Number of the specification utilised by the system creating this message. The Message Version Number is set by the 3DS Server which originates the protocol with the AReq message.	3DS Server	Length: Variable, 5–8 characters JSON Data Type: String Value accepted: Refer to 3.9.1	01-APP 02-BRW 03-3RI	01-PA 02-NPA	RRes = R Erro = R	

		The Message Version Number does not change during a 3DS transaction.						
7	resultsStatus	Indicates the status of the Results Request message from the 3DS Server to provide additional data to the ACS. This will indicate if the message was successfully received for further processing or will be used to provide more detail on why the Challenge could not be completed from the 3DS Client to the ACS.	3DS Server	Length: 2 characters JSON Data Type: String Values accepted: • 01 = Results Request Received for further Processing • 02 = Challenge Request not sent to ACS by 3DS Requestor (3DS Server or 3DS Requestor opted out of the challenge) • 03 = ARes challenge data not delivered to the 3DS Requestor due to technical error • 04–79 = Reserved for EMVCo future use (values invalid until defined by EMVCo) • 80-99 = Reserved for DS use	01-APP 02-BRW	01-PA 02-NPA	RRes = R	

3.7 PREPARATION REQUEST (PREQ)

The PReq message is sent from the 3DS Server to the DS to request information about the Protocol Version Number(s) supported by available ACSs and the DS and if one exists, any corresponding 3DS Method URL. This message is not part of the 3-D Secure authentication message flow.

Step. Number	Field Name	Description	Source	Length/Format/Values	Device Channel	Message Category	Message Inclusion	Conditional Inclusion
1	threeDSServerRefNumber	Unique identifier assigned by the EMVCo secretariat upon testing and approval.	3DS Server	Length: Variable, maximum 32 characters JSON Data Type: String Value accepted: Set by the EMVCo Secretariat	01-APP 02-BRW 03-3RI	01-PA 02-NPA	PReq = R	
2	threeDSServerOperatorID	DS assigned 3DS Server identifier. Each DS can provide a unique ID to each 3DS Server on an individual basis.	3DS Server	Length: Variable, maximum 32 characters JSON Data Type: String Value accepted: Any individual DS may impose specific formatting and character requirements on the contents of this field.	01-APP 02-BRW 03-3RI	01-PA 02-NPA	PReq = C	Requirements for the presence of this field are DS specific.
3	threeDSServerTransID	Universally unique transaction identifier assigned by the 3DS Server to identify a single transaction.	3DS Server	Length: 36 characters JSON Data Type: String Value accepted: Canonical format as defined in IETF RFC 4122. May utilise any of the specified versions if the output meets specified requirements.	01-APP 02-BRW 03-3RI	01-PA 02-NPA	PReq = R Erro = C	Required in Error Message if available (e.g. can be obtained from a message or is being generated).
4	messageExtension	Data necessary to support requirements not otherwise defined in the 3-D Secure message are carried in a Message Extension.	3DS Server	Length: Variable, maximum 81920 bytes JSON Data Type: Array Values accepted: Refer to 3.9.4 for data elements.	01-APP 02-BRW 03-3RI	01-PA 02-NPA	PReq = C	Conditions to be set by each DS.

5	messageType	Identifies the type of message that is passed.	3DS Server 3DS SDK DS ACS	Length: 4 characters JSON Data Type: String Values accepted: • PReq • Erro	01-APP 02-BRW 03-3RI	01-PA 02-NPA	PReq = R Erro = R	
6	messageVersion	Protocol version identifier This shall be the Protocol Version Number of the specification utilised by the system creating this message. The Message Version Number is set by the 3DS Server which originates the protocol with the AReq message. The Message Version Number does not change during a 3DS transaction.	3DS Server	Length: Variable, 5–8 characters JSON Data Type: String Value accepted: Refer to 3.9.1	01-APP 02-BRW 03-3RI	01-PA 02-NPA	PReq = R Erro = R	
7	serialNum	If present in PReq message, the DS returns Card Range Data that has been updated since the time of the PRes message. If absent, the DS returns all card ranges. If present in the PRes message, indicates the current state of the Card Range Data (the specific value is only meaningful to the DS). The 3DS Server should retain this value for submission in a future PReq message to request only changes that have been made to the card range data since the PRes message was generated.	DS 3DS Server	Length: Variable, maximum 20 characters JSON Data Type: String	01-APP 02-BRW	01-PA 02-NPA	PReq = O	

3.8 PREPARATION RESPONSE (PRES)

The PRes message is the DS response to the PReq message. The 3DS Server can utilise the PRes message to cache information about the Protocol Version(s) supported by available ACSs and the DS, and if one exists, about the corresponding 3DS Method URL. This message is not part of the 3-D Secure authentication message flow.

Step. Number	Field Name	Description	Source	Length/Format/Values	Device Channel	Message Category	Message Inclusion	Conditional Inclusion
1	threeDSServerTransID	Universally unique transaction identifier assigned by the 3DS Server to identify a single transaction.	3DS Server	Length: 36 characters JSON Data Type: String Value accepted: Canonical format as defined in IETF RFC 4122. May utilise any of the specified versions if the output meets specified requirements.	01-APP 02-BRW 03-3RI	01-PA 02-NPA	PRes = R Erro = C	Required in Error Message if available (e.g. can be obtained from a message or is being generated).
2	cardRangeData	Card range data from the DS indicating the most recent protocol versions supported by the ACS, and optionally the DS that hosts that range, and if configured, the ACS URL for the 3DS Method. May be as many JSON Objects as there are stored card ranges in DS.	DS	Length: Variable JSON Data Type: Array Values accepted: See 3.9.9 for Card Range Data elements.	N/A	01-PA 02-NPA	PRes = O	
3	dsEndProtocolVersion	The most recent active protocol version that is supported for the DS. Note: Optional within the Card Range Data (as defined in 3.9.9).	DS	Length: Variable, 5–8 characters JSON Data Type: String	N/A	01-PA 02-NPA	PRes = R	

4	dsStartProtocolVersion	The most recent active protocol version that is supported for the DS. Optional within the Card Range Data (as defined in 3.9.9).	DS	Length: Variable, 5–8 characters JSON Data Type: String	N/A	01-PA 02-NPA	PRes = R	
5	dsTransID	Universally unique transaction identifier assigned by the DS to identify a single transaction.	DS	Length: 36 characters JSON Data Type: String Value accepted: Canonical format as defined in IETF RFC 4122. May utilise any of the specified versions as long as the output meets specified requirements.	01-APP 02-BRW 03-3RI	01-PA 02-NPA	PRes = R Erro = C	The DS will populate the AReq with this data element prior to passing to the ACS. Required in Error Message if available (e.g. can be obtained from a message or is being generated).
6	messageExtension	Data necessary to support requirements not otherwise defined in the 3-D Secure message are carried in a Message Extension.	3DS Server	Length: Variable, maximum 81920 bytes JSON Data Type: Array Values accepted: Refer to 3.9.4 for data elements.	01-APP 02-BRW 03-3RI	01-PA 02-NPA	PRes = C	Conditions to be set by each DS.
7	messageType	Identifies the type of message that is passed.	3DS Server 3DS SDK DS ACS	Length: 4 characters JSON Data Type: String Values accepted: • PRes • Erro	01-APP 02-BRW 03-3RI	01-PA 02-NPA	PRes = R Erro = R	
8	messageVersion	Protocol version identifier This shall be the Protocol Version Number of the specification utilised by the system creating this message. The Message Version Number is set by the 3DS Server which originates the protocol with the AReq	3DS Server	Length: Variable, 5–8 characters JSON Data Type: String Value accepted: Refer to 3.9.1	01-APP 02-BRW 03-3RI	01-PA 02-NPA	PRes = R Erro = R	

		message. The Message Version Number does not change during a 3DS transaction.						
9	serialNum	If present in PReq message, the DS returns Card Range Data that has been updated since the time of the PRes message. If absent, the DS returns all card ranges. If present in the PRes message, indicates the current state of the Card Range Data (the specific value is only meaningful to the DS). The 3DS Server should retain this value for submission in a future PReq message to request only changes that have been made to the card range data since the PRes message was generated.	DS 3DS Server	Length: Variable, maximum 20 characters JSON Data Type: String	01-APP 02-BRW	01-PA 02-NPA	PRes = O	

3.9 DATA ELEMENT INFORMATION

3.9.1 3-D SECURE PROTOCOL VERSION NUMBER

Protocol Version Number	Status
2.0.0	Deprecated
2.1.0	Active

3.9.1 EXCLUDED ISO CURRENCY AND COUNTRY CODE VALUES

Code	Value Not Permitted for 3-D Secure	Definition
ISO 4217	955	European Composite Unit
ISO 4217	956	European Monetary Unit
ISO 4217	957	European Unit of Account 9
ISO 4217	958	European Unit of Account 17
ISO 4217	959	Gold
ISO 4217	960	I.M.F.
ISO 4217	961	Silver
ISO 4217	962	Platinum
ISO 4217	963	Reserved for testing
ISO 4217	964	Palladium
ISO 4217	999	No currency is involved
ISO 3166-1	901–999	Reserved by ISO to designate country names not otherwise defined

3.9.2 DEVICE RENDERING OPTIONS SUPPORTED

Element/Field Name	Description	Length/Format/Values
SDK Interface Field Name: sdkInterface	Lists all of the SDK Interface types that the device supports for displaying specific challenge user interfaces within the SDK.	Length: 2 characters JSON Data Type: String Values accepted: • 01 = Native • 02 = HTML • 03 = Both
SDK UI Type Field Name: sdkUiType	Lists all UI types that the device supports for displaying specific challenge user interfaces within the SDK. Valid values for each Interface: • Native UI = 01–04 • HTML UI = 01–05 Note: Currently, all SDKs need to support all UI Types. In the future, however, this may change (for example, smart watches may support a UI Type not yet defined by this specification).	Length: 2 characters JSON Data Type: Array of String String values accepted: • 01 = Text • 02 = Single Select • 03 = Multi Select • 04 = OOB • 05 = HTML Other (valid only for HTML UI)

3.9.3 MESSAGE EXTENSION ATTRIBUTES

Name	Description	Length/Format/Value	Inclusion
criticalityIndicator	A Boolean value indicating whether the recipient must understand the contents of the extension to interpret the entire message.	JSON Data Type: Boolean Values accepted: • true • false	R
data	The data carried in the extension.	Length: Variable, Maximum 8059 characters • JSON Data Type: Object	R
id	A unique identifier for the extension.	Length: Variable, Maximum 64 characters	R

	Note: Payment System Registered Application Provider Identifier (RID) is required as prefix of the ID.	JSON Data Type: String	
name	The name of the extension data set as defined by the extension owner.	Length: Variable, Maximum 64 characters JSON Data Type: String	R

3.9.4 ACS RENDERING TYPE

Element/Field Name	Description	Length/Format/Values
ACS Interface Field Name: acsInterface	This the ACS interface that the challenge will present to the cardholder.	Length: 2 characters JSON Data Type: String Values accepted: • 01 = Native UI • 02 = HTML UI
ACS UI Template Field Name: acsUiTemplate	Identifies the UI Template format that the ACS first presents to the consumer. Valid values for each Interface: • Native UI = 01–04 • HTML UI = 01–05	Length: 2 characters JSON Data Type: String Values accepted: • 01 = Text • 02 = Single Select • 03 = Multi Select • • 04 = OOB • • 05 = HTML Other (valid only for HTML UI)

3.9.5 MERCHANT RISK INDICATOR

Element/Field Name	Description	Length/Format/Values
deliveryEmailAddress	For Electronic delivery, the email address to which the merchandise was delivered.	Length: maximum 254 characters • JSON Data Type: String

Delivery Timeframe Field name: deliveryTimeframe	Indicates the merchandise delivery timeframe.	Length: 2 characters JSON Data Type: String Values accepted: • 01 = Electronic Delivery • 02 = Same day shipping • 03 = Overnight shipping • 04 = Two-day or more shipping
Gift Card Amount Field Name: giftCardAmount	For prepaid or gift card purchase, the purchase amount total of prepaid or gift card(s) in major units (for example, USD 123.45 is 123).	Length: maximum 15 characters JSON Data Type: String
Gift Card Count Field Name: giftCardCount	For prepaid or gift card purchase, total count of individual prepaid or gift cards/codes purchased.	Length: 2 characters JSON Data Type: String
Gift Card Currency Field Name: giftCardCurr	For prepaid or gift card purchase, the currency code of the card as defined in ISO 4217 other than those listed in 3.9.2.	Length: 3 characters JSON Data Type: String
Pre-Order Date Field Name: preOrderDate	For a pre-ordered purchase, the expected date that the merchandise will be available.	Length: 8 characters JSON Data Type: String Format accepted: Date format = YYYYMMDD
Pre-Order Purchase Indicator Field Name: preOrderPurchaseInd	Indicates whether Cardholder is placing an order for merchandise with a future availability or release date.	Length: 2 characters JSON Data Type: String Values accepted: • 01 = Merchandise available • 02 = Future availability
Reorder Items Indicator Field Name: reorderItemsInd	Indicates whether the cardholder is reordering previously purchased merchandise.	Length: 2 characters JSON Data Type: String Values accepted: • 01 = First time ordered • 02 = Reordered

3.9.6 ISSUER IMAGE

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
	Include up to three fully qualified URLs defined as either; medium density, high density and extra high-density images of the Issuer Image.
	
	Examples:
	
	Images to display:
	
	“issuerImage” :{
	
	“medium”: “http://acs.com/medium_image.svg”,
	
	”high”: “http://acs.com/high_image.svg”,
	
	“extraHigh”: “http://acs.com/extraHigh_image.svg”
	
	}
		Length: Variable, maximum 2048 characters
		
		JSON Data Type: String
		
		Values accepted:
		
		Images to display:
		
		• Fully qualified URL in correct JSON Object format

3.9.7 PAYMENT SYSTEM IMAGE

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
	Include up to three fully qualified URLs defined as either; medium density, high density and extra-high density images of the DS or Payment System image.
	
	Examples:
	
	Images to display:
	
	“psImage” :{
	
	“medium”: “http://ds.com/medium_image.svg”,
	
	”high”: “http://ds.com/high_image.svg”,
	
	“extraHigh”: “http://ds.com/extraHigh_image.svg”
	
	}
		Length: Variable, maximum 2048 characters
		
		JSON Data Type: String
		
		Values accepted:
		
		Images to display:
		
		• Fully qualified URL in correct JSON Object format

3.9.8 CARD RANGE DATA

The Card Range Data data element contains information returned in the PRes message to the 3DS Server from the DS that indicates the most recent EMV 3-D Secure version supported by the ACS that hosts that card range. It also may optionally contain the ACS URL for the 3DS Method if supported by the ACS and the DS Start and End Protocol Versions which support the card range

Name	Description	Length/Format/Value	Inclusion
3DS Method URL Field name: threeDSMethodURL	The ACS URL that will be used by the 3DS Method. Note: The 3DSMethodURL data element may be omitted if not supported by the ACS for this specific card range.	Length: Variable, Maximum 256 characters JSON Data Type: String • Value accepted: Fully qualified URL	O
ACS End Protocol Version Field Name: acsEndProtocolVersion	The most recent active protocol version that is supported for the ACS URL. Refer to 3.9.1 for active protocol version numbers.	Length: Variable, 5–8 characters • JSON Data Type: String	R
ACS Start Protocol Version Field Name: acsStartProtocolVersion	The earliest (i.e. oldest) active protocol version that is supported by the ACS. Refer to 3.9.1 for active protocol version numbers.	Length: Variable, 5–8 characters JSON Data Type: String	R
Indicator Field Name: actionInd	Indicates the action to take with the card range. The card ranges are processed in the order returned. Note: If the Serial Number is not included in the PReq message, then the action is A = Add for all card ranges returned (the Action Indicator is ignored in the PRes message).	Length: 1 character JSON Data Type: String Value accepted: • • A = Add the card range to the cache (default value) • • D = Delete the card range from the cache	O
DS End Protocol Version	The most recent active protocol version that is supported for the DS.	Length: Variable, 5–8 characters	O

Field Name: dsEndProtocolVersion		JSON Data Type: String	
DS Start Protocol Version Field Name: dsStartProtocolVersion	The earliest (i.e. oldest) active protocol version that is supported by the DS.	Length: Variable, 5–8 characters JSON Data Type: String	O
End Card Range Field Name: endRange	End of the card range.	Length: 13–19 characters JSON Data Type: String	R
Start Card Range Field Name: startRange	Start of the card range.	Length: 13–19 characters JSON Data Type: String	R

3.9.9 LOCAL SWITCHING AUTHENTICATION VERIFICATION VALUE (LSAV)

The local switching authentication value is the authentication value (authenticationValue) in Authentication Response or Result Response

Subfield	Position	Length	Definition
1	1	1 Byte	**Secure Authentication Results Code.** The result code is mapped by Transaction Status field (refer to 3.9.11 for mapping table) valid values: • 0: Authentication Successful. • 5: Authentication could not be performed • 7: Attempted Authentication. The attempted authentication response was generated by the issuer's attempts ACS for a non-participating cardholder OR by the ITMX LSS Attempts Service for non-participating issuer. • 8: The attempted authentication response was generated by the ITMX LSS Attempts Service because the Issuer ACS was unavailable. • 9: Authentication failed.

Subfield	Position	Length	Definition
2	2	1 Byte	**Second Factor Authentication Code** The Second Factor Authentication Code represents the additional authentication method used by the Issuer ACS **SUPPORTED VALUES FOR EMV 3DS INCLUDE:** • 01: Challenge flow using Static Passcode • 02: Challenge flow using OTP via SMS method • 03: Challenge flow using OTP via key fob or card reader method • 04: Challenge flow using OTP via App method • 05: Challenge flow using OTP via any other method • 06: Challenge flow using KBA method • 07: Challenge flow using OOB with Biometric method • 08: Challenge flow using OOB with App login method • 09: Challenge flow using OOB with any other method • 10: Challenge flow using any other authentication method • 11: Push Confirmation • 98: Attempts Server responding Note. KBA = Knowledge-Based Authentication—Authentication credentials are entered into a 3DS user interface as part of the 3DS protocol. OOB = Out-of-Band Authentication—Authentication credentials are not entered into a 3DS user interface. OTP = One-Time Passcode—Authentication credentials are entered into a 3DS user interface as part of the 3DS protocol.
3	3	1 Byte	**Key Indicator** The Token Key Indicator identifies the Token key set used to calculate the Token Value. A leading zero is required to pad the first unused half-byte of the Token Key Indicator. valid values: 01- key set 1 02- key set 2
4	4-5	2 Bytes	**Token Value** Value is a 3-digit code generated by the issuer's ACS that may be used by the issuer to validate the authentication response during authorization. A leading zero is required in byte 4 to pad the first unused half-byte of the **Token** For example, a **Token** Value of 789 is represented as 0789.
5	6-7	2 Bytes	**Unpredictable Number** The Unpredictable Number is derived from the Authentication Tracking Number (ATN) which is created by the Issuer ACS or an Attempts Server. The value is a 4-digit code that contains the least four significant digits of the ATN
6	8-15	8 Bytes	**Authentication Tracking Number (ATN)** This 8-byte/16 BCD value is a 16-digit code generated by the Issuer's ACS to identify the transaction
7	16	1 Byte	**Version and Authentication Action** The left nibble of this 1-byte/1-BCD value identifies a version; the right nibble identifies the authentication action. Default: 0
8	17-20	4 Bytes	**Reserved.** This subfield is zero filled .

Token Value

No	Location	Description	Remark
1	Bit 1 – 16	PAN	Cardholder PAN as provided in the Authentication Request from the merchant
2	Bit 17 – 20	Unpredictable Number	The Unpredictable Number is derived from the Authentication Tracking Number (ATN) which is created by the Issuer ACS or an Attempts Server. The value is a 4-digit code that contains the least four significant digits of the ATN
3	Bit 21	Authentication Results Code	Authentication Results Code indicates the results of the ACS authentication of the cardholder and is derived from the Transaction Status field in the authentication response. Refer to 3.9.12.
4	Bit 22 – 23	Authentication Method	The Authentication Method is a 2-digit code that indicates the authentication method used by the Issuer ACS during authentication to authentication the cardholder.(Subfield 2 of LSAV)

3.9.10 ERROR CODE

Value	Error Code	Error Description	Error Detail
101	Message Received Invalid	One of the following: • Message is not AReq, ARes, CReq, CRes, PReq, PRes, RReq, or RRes • Valid Message Type is sent to or from an inappropriate component (such as AReq message being sent to the 3DS Server) • Message not recognised	One of the following: • Invalid Message Type • Invalid Message for the receiving component • Invalid Formatted Message
102	Message Version Number Not Supported	Message Version Number received is not valid for the receiving component.	All supported Protocol Version Numbers in a comma delimited list.
103	Sent Messages Limit Exceeded	Exceeded maximum number of PReq messages sent to the DS.	For example, the 3DS Server sends two PReq messages to the DS within one hour.

201	Required Data Element Missing	A message element required as defined in 3 is missing from the message.	Name of required element(s) that was omitted; if more than one element is detected, this is a comma delimited list. Parent Example: messageType Parent/Child Example: acctInfo.chAccAgeInd ID of critical Message Extension(s) that was not recognised; if more than one extension is detected, this is a comma delimited list of message identifiers that were not recognised.
202	Critical Message Extension Not Recognised	Critical message extension not recognised.	Name of invalid element(s); if more than one invalid data element is detected, this is a comma delimited list.
203	Format of one or more Data Elements is Invalid according to the Specification	• Data element not in the required format or value is invalid as defined in Section 3, or • Message Version Number does not match the value set by the 3DS Server in the AReq message.	Name of invalid element(s); if more than one invalid data element is detected, this is a comma delimited list.
204	Duplicate Data Element	Valid data element presents more than once in the message.	Name of duplicated data element; if more than one duplicate data element is detected, this is a comma delimited list.
301	Transaction ID Not Recognised	Transaction ID received is not valid for the receiving component.	The Transaction ID received was invalid.
302	Data Decryption Failure	Data could not be decrypted by the receiving system due to technical or other reason.	Description of the failure.
303	Access Denied, Invalid Endpoint	Access denied, invalid endpoint.	Description of the failure.
304	ISO Code Invalid	ISO code not valid per ISO tables (for either country or currency), or code is one of the excluded values listed in Table A.5.	Name of invalid element(s); if more than one invalid element is detected this is a comma delimited list If Challenge Request. Purchase.currency and Challenge

			Request.Purchase.exponent form an invalid pair, list both as Error Description.
305	Transaction data not valid	If in response to an AReq message: • Cardholder Account Number is not in a range belonging to Issuer If in response to a CReq, and a CReq message was incorrectly sent, one of the following: • CReq message was received by the wrong ACS • CReq message was not sent, based on the values in the ARes message • CReq message with this ACS Transaction ID has already been received and processed	Name of element(s) that caused the ACS to decide that the AReq message or CReq message was incorrectly sent; if more than one invalid element is detected this is a commadelimited list.
306	Merchant Category Code (MCC) Not Valid for Payment System	Merchant Category Code (MCC) not valid for Payment System.	For example, Invalid MCC received in the AReq message.
307	Serial Number not Valid	Serial Number not valid.	For example, Invalid Serial number in the PReq/PRes message (e.g., too old, not found).
402	Transaction Timed Out	Transaction timed-out.	For example, Timeout expiry reached for the transaction as defined in 2.5.1
403	Transient System Failure	Transient system failure.	For example, a slowly processing back-end system.
404	Permanent System Failure	Permanent system failure.	For example, a critical database cannot be accessed.
405	System Connection Failure	System connection failure.	For example, the sending component is unable to establish connection to the receiving component.

3.9.11 TRANSACTION STATUS

Transaction Status (transStatus) is found in the ARes and RReq message. Transaction Status field is a single alpha character. For use in creating the LSAV, the Transaction Status value must be converted to an Authentication Results Code which is a 1-digit numeric value, as shown below.

Field Name	Value	Description	Authentication Results Code
transStatus	Y	Authentication/ Account Verification Successful	0
	N	Not Authenticate/Account Not Verified, Transaction denied. Authentication Failed	9
	U	Authentication/ Account Verification Could Not Be Performed; Technical or other problem, as indicated in ARes or RReq.	5
	A	Attempts Processing Performed; Not Authenticated/Verified, but a proof of attempted authentication/verification is provided. Non-participating issuer, or Non-participating cardholder of participating issuer	7*
		Attempted. Proof of authentication attempt generated for participating issuer with server not available (Proof of Attempts STIP (Stand In Processor)) .	8*
	R	Authentication/Account Verification Rejected; Issuer is rejecting authentication/verification and request that authorization not be attempted	9

7 and 8 are support when ITMX enable attempt service

3.9.12 TRANSACTION STATUS REASON

Field Name	Value	Description
transStatusReason	01	Card authentication failed
	02	Unknown Device
	03	Unsupported Device
	04	Exceeds authentication frequency limit
	05	Expired card
	06	Invalid card number
	07	Invalid transaction
	08	No Card record
	09	Security failure
	10	Stolen card
	11	Suspected fraud
	12	Transaction not permitted to cardholder
	13	Cardholder not enrolled in service
	14	Transaction timed out at the ACS
	15	Low confidence
	16	Medium confidence
	17	High confidence
	18	Very High confidence
	19	Exceeds ACS maximum challenges
	20	Non-Payment transaction not supported
	21	3RI transaction not supported
	22–79	Reserved for EMVCo future use
	80–99	Reserved for ITMX LSS EMVCO

3.9.13 3DS REQUESTOR AUTHENTICATION INFORMATION

The 3DS Requestor Authentication Information contains optional information about how the cardholder authenticated during login to their 3DS Requestor account.

Element/Field Name	Description	Length/Format/Values
3DS Requestor Authentication Data Field Name: threeDSReqAuthData	Data that documents and supports a specific authentication process. In the current version of the specification, this data element is not defined in detail, however the intention is that for each 3DS Requestor Authentication Method, this field carry data that the ACS can use to verify the authentication process. For example, for method: 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 04—data element can carry the FIDO attestation data (including the signature) In future versions of the specification, these details are expected to be included	Length: maximum 2048 bytes JSON Data Type: String Value accepted: Any
3DS Requestor Authentication Method Field Name: threeDSReqAuthMethod	Mechanism used by the Cardholder to authenticate to the 3DS Requestor.	Length: 2 characters JSON Data Type: String Values accepted: • 01 = No 3DS Requestor authentication occurred (i.e. cardholder “logged in” as guest) • 02 = Login to the cardholder account at the 3DS Requestor system using 3DS Requestor’s own credentials • 03 = Login to the cardholder account at the 3DS Requestor system using federated ID • 04 = Login to the cardholder account at the 3DS Requestor system using issuer credentials • 05 = Login to the cardholder account at the 3DS Requestor

		system using third-party authentication • 06 = Login to the cardholder account at the 3DS Requestor system using FIDO Authenticator • 07–79 = Reserved for EMVCo future use (values invalid until defined by EMVCo) • 80–99 = Reserved for DS use
3DS Requestor Authentication Timestamp Field name: threeDSReqAuthTimestamp	Date and time in UTC of the cardholder authentication.	Length: 12 characters JSON Data Type: String Format accepted: Date format = YYYYMMDDHHMM

3.9.14 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

Element/Field Name	Description	Length/Format/Values
3DS Requestor Prior Transaction Authentication Data Field Name: threeDSReqPriorAuthData	Data that documents and supports a specific authentication process. In the current version of the specification this data element is not defined in detail, however the intention is that for each 3DS Requestor Authentication Method, this field carry data that the ACS can use to verify the authentication process. In future versions of the specification, these details are expected to be included.	Length: maximum 2048 bytes Format: Any

3DS Requestor Prior Transaction Authentication Method Field Name: threeDSReqPriorAuthMethod	Mechanism used by the Cardholder to previously authenticate to the 3DS Requestor.	Length: 2 characters JSON Data Type: String Values accepted: • 01 = Frictionless authentication occurred by ACS • 02 = Cardholder challenge occurred by ACS • 03 = AVS verified • 04 = Other issuer methods • 05–79 = Reserved for EMVCo future use (values invalid until defined by EMVCo) • 80–99 = Reserved for DS use
3DS Requestor Prior Transaction Authentication Timestamp Field name: threeDSReqPriorAuthTimestamp	Date and time in UTC of the prior cardholder authentication.	Length: 12 characters JSON Data Type: String Format accepted: Date format = YYYYMMDDHHMM
3DS Requestor Prior Transaction Reference Field Name: threeDSReqPriorRef	This data element provides additional information to the ACS to determine the best approach for handing a request.	Length: 36 characters JSON Data Type: String Value accepted: This data element contains an ACS Transaction ID for a prior authenticated transaction (for example, the first recurring transaction that was authenticated with the cardholder).

3.9.15 DEVICE INFORMATION—01-APP ONLY

The Device Information is gathered by the 3DS SDK as defined in the *EMV 3-D Secure SDK—Device Information* specification. This data is placed into a JWE object that will encrypt the data using the DS public key.

3.9.16 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 accurately populate the browser information for each transaction. This data may be obtained by 3DS software provided to the 3DS Requestor or through remote JavaScript calls, but 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:

- Browser Accept Headers
- Browser IP Address
- Browser Java Enabled
- Browser Language
- Browser Screen Color Depth
- Browser Screen Height
- Browser Screen Width
- Browser Time Zone
- Browser User-Agent

3.10 ERROR MESSAGE DATA ELEMENT INFORMATION

Data Element	Field Name
3DS Server Transaction ID	threeDSServerTransID
ACS Transaction ID	acsTransID
DS Transaction ID	dsTransID
Error Code	errorCode
Component	errorComponent
Error Description	errorDescription
Error Detail	errorDetail
Error Message Type	errorMessageType
Message Type	messageType
Message Version Number	messageVersion
SDK Transaction ID	sdkTransID

Chapter 4

4. SECURITY

4.1 LINKS SECURITY

4.1.1 3DS SERVER AND DS

4.1.1.1 For AReq/ARes

The 3DS Server to DS link for the AReq/ARes messages is established using a TLS protocol with mutual authentication. The public key certificates of both parties are signed by the DS CA, with the 3DS Server making the necessary selection if it connects to more than one DS.

- TLS 1.2 Protocol
- 3DS Server public key
  Client Certificate format: X.509
- DS public key
  Server Certificate format: X.509
- CA signing 3DS Server key—DS CA
- CA signing DS key—DS CA

4.1.1.2 For RReq/RRes

The DS to 3DS Server link for the RReq/RRes messages is established using a TLS protocol with mutual authentication. The public key certificates of both parties are signed by the DS CA.

- TLS 1.2 Protocol
- DS public key
- Client Certificate format: X.509
- 3DS Server public key
- Server Certificate format: X.509
- CA signing DS key—DS CA
- CA signing 3DS Server key—DS CA

4.1.2 DS AND ACS

4.1.2.1 For AReq/ARes

The DS to ACS link for the AReq/ARes messages is established using a TLS protocol with mutual authentication. The public key certificates of both parties are signed by the DS CA.

- TLS 1.2 Protocol
- DS public key
- Client Certificate format: X.509
- ACS public key
- Server Certificate format: X.509
- CA signing DS key—DS CA
- CA signing ACS key—DS CA

4.1.2.2 For RReq/RRes

The DS to 3DS Server link for the RReq/RRes messages is established using a TLS protocol with mutual authentication. The public key certificates of both parties are signed by the DS CA.

- TLS 1.2 Protocol
- ACS public key
- Client Certificate format: X.509
- DS public key
- Server Certificate format: X.509
- CA signing ACS key—DS CA
- CA signing DS Server key—DS CA

4.2 SECURITY FUNCTIONS

4.2.1 3DS SDK ENCRYPTION TO DS

- 3DS SDK Encryption

  Refer to EMVCo 3DS Protocal and Core Function Specification Section 6.2.2.1

4.2.2 3DS SDK—ACS SECURE CHANNEL SET-UP

Refer to EMVCo 3DS Protocal and Core Function Specification Section 6.2.3

Chapter 5

5. AUTHORIZATION PROCESSING

5.1 ELECTRONIC COMMERCE INDICATOR (ECI)

The Electronic Commerce Indicator (ECI) indicates the level of security used when the cardholder provided payment information to the merchant. It must be set to a value corresponding to the authentication results and the characteristics of the merchant checkout process. The merchant commerce server transmits the authorization request message, including the ECI, to the acquirer or its processor.

Possible ECI data values are:

- ECI = 05: Fully Authentication: This value means that the cardholder was authenticated by the issuer by verifying the cardholder's password or identity information. The value is returned by the ACS in the Payer Authentication Response message when the cardholder successfully passed Secure payment authentication.

- ECI = 06: This value means that the merchant attempted to authenticate the cardholder, but the cardholder was not participating. The value should be returned by the ACS in the Authentication Response message for an Attempt Response. Additionally, merchants may use an ECI 6 in the authorization request when a Verify Enrollment of N is received from the ITMX LSS EMVCo

- ECI = 07: This value is set by the merchant when the payment transaction was conducted over a secure channel (for example, SSL/TLS), but payment authentication was not performed, or when the issuer responded that authentication could not be performed. An ECI 07 applies when either the Verify Enrollment or the Payer Authentication Response contains a U for Unable to Authenticate.

These values and related information are summarized in Section 5.3, Authentication and Authorization.

5.2 LOCAL SWITCHING AUTHENTICATION VERIFICATION VALUE AND AUTHENTICATION IDENTIFIER

Local Switching Authentication Verification Value (LSAV)

The LSAV is a cryptographic value derived by the issuer during authentication that can provide evidence of the results of payment authentication during an online purchase, the detail of LSAV generating refer to 3.9.9. If a merchant receives a LSAV value in a Payer Authentication Response message from the issuer, the LSAV **must** be included in the ITMX Local Switching authorization message in order for the merchant to receive chargeback protection:

• The merchant **must** be able to send the LSAV to its acquirer.

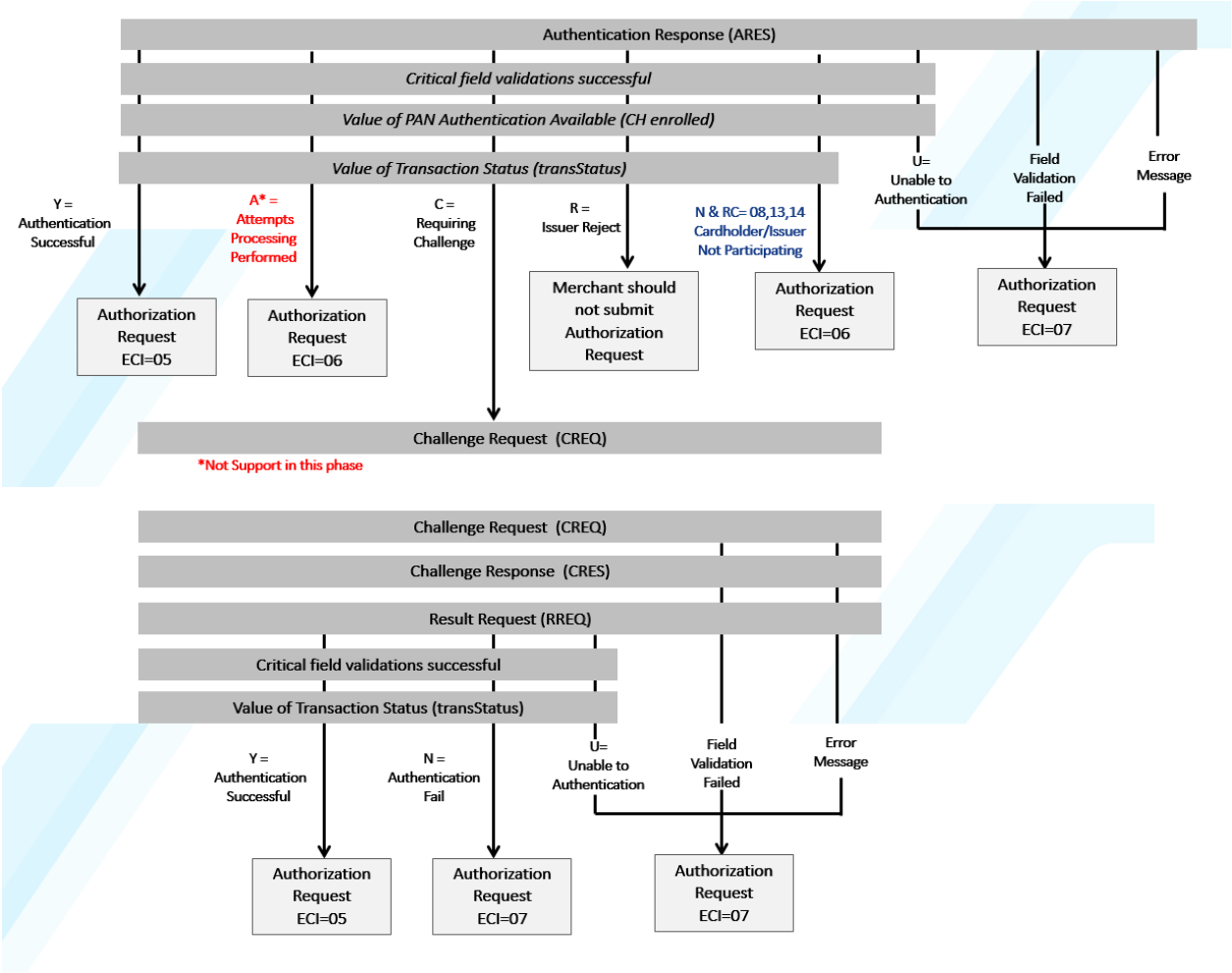
• The acquirer **must** be able to receive the LSAV from the merchant and correctly transmit the data in the authorization request.

Authentication Identifier

The Authentication Identifier is the Authentication Tracking Number (ATN) subfield of the LSAV value transmitted by the issuer ACS. Acquirers and merchants are not required to transmit a separate Authentication Identifier

5.3 AUTHENTICATION AND AUTHORIZATION

Figure below summarizes the possible outcomes of an ITMX LSS EMVCo transaction, illustrating when a merchant's submission of an authorization request.



Only transStatus = N and Reason Code = 08 or 13 or 14 then Acquirer send ECI=06 in authorization message.

<table>
<tr><th>Field Name</th><th>Value</th><th>Description</th><th>ECI</th><th>LSAV</th></tr>
<tr><td rowspan="6">Authentication Request /Response

(AReq/ARes)

The 3DS Server sends the AReq through the ITMX LSS EMVCO to the Issuer ACS or Attempts ACS

Upon receipt, the Issuer ACS or Attempts ACS performs risk-based authentication and provides the results of authentication to the 3DS Server in the ARes</td><td>Y</td><td>Authentication Successful</td><td>05</td><td>LSAV

present</td></tr>
<tr><td>A*</td><td>Attempts Processing Performed</td><td>06</td><td>No LSAV</td></tr>
<tr><td>N**</td><td>Authentication Failed; Not Authenticated; Transaction Denied</td><td>06</td><td>No LSAV</td></tr>
<tr><td>U</td><td>Authentication Could Not Be Performed; Technical or Other Problem</td><td rowspan="4">07</td><td rowspan="4">No LSAV</td></tr>
<tr><td>N</td><td>Authentication Failed; Not Authenticated; Transaction Denied</td></tr>
<tr><td>C</td><td>Challenge Required to authenticate the cardholder</td></tr>
<tr><td>R</td><td>Authentication Rejected</td></tr>
<tr><td rowspan="2">Challenge Request/Response

(CReq/CRes)

The 3DS Server (or 3DS SDK) sends the CReq to the Issuer ACS

Upon receipt, the Issuer ACS challenges the cardholder through an authentication

method such as OTP and responds to the 3DS Server or 3DS SDK with the CRes</td><td>Y</td><td>Authentication Successful</td><td colspan="2" rowspan="2">Results of the challenge are sent in the Results Request (RReq) message by the ACS to the 3DS Server.</td></tr>
<tr><td>N</td><td>Not Authenticated; Transaction Denied</td></tr>
<tr><td>Results Request/Response (RReq/RRes)

The Issuer ACS sends the RReq to the 3DS Server to provide the results of the challenge authentication

The 3DS Server acknowledges the RReq by responding with the RRes</td><td colspan="4">Same set of values as AReq/ARes

A successful challenge is an ECI 05 with a LSAV

An unsuccessful challenge is an ECI 07 with no LSAV</td></tr>
</table>

* For phase1, ITMX not support attempts yet

** Only when transStatus = N and Reason Code = 08 or 13 or 14 Then Acquirer send ECI=06 in authorization message.

LSS EMVCo Transaction Status Reason Code

Message	Transaction Status	Value	Description	ECI	LSAV
Authentication Request/Response (AReq/Ares)	N U R	01	Card authentication failed	07	No LSAV
		02	Unknown Device		
		03	Unsupported Device		
		04	Exceeds authentication frequency limit		
		05	Expired card		
		06	Invalid card number		
		07	Invalid transaction		
	N	08*	No Card record	06	
	N U R	09	Security failure	07	
		10	Stolen card		
		11	Suspected fraud		
		12	Transaction not permitted to cardholder		
	N	13*	Cardholder not enrolled in service	06	
		14*	Transaction timed out at the ACS		
	N U R	15	Low confidence	07	
		16	Medium confidence		
		17	High confidence		
		18	Very High confidence		
		19	Exceeds ACS maximum challenges		
		20	Non-Payment transaction not supported		

		26	Authentication attempted but not performed by the cardholder		
		21	3RI transaction not supported	N/A	
		80	Error Connecting to ACS	07	No LSAV
		81	ACS Timed Out		
		82	Invalid Response from ACS		
		83	System Error Response from ACS		
		84	Internal Error While Generating LSAV		
		85	VMID not eligible for requested program		
		86	Protocol Version Not Supported by ACS		
		87	Transaction is excluded from Attempts Processing		
		88	Requested program not supported by the ACS		

* Only when transStatus = N and Reason Code = 08 or 13 or 14 Then Acquirer send ECI=06 in authorization message.

5.4 AUTHORIZATION FLOW

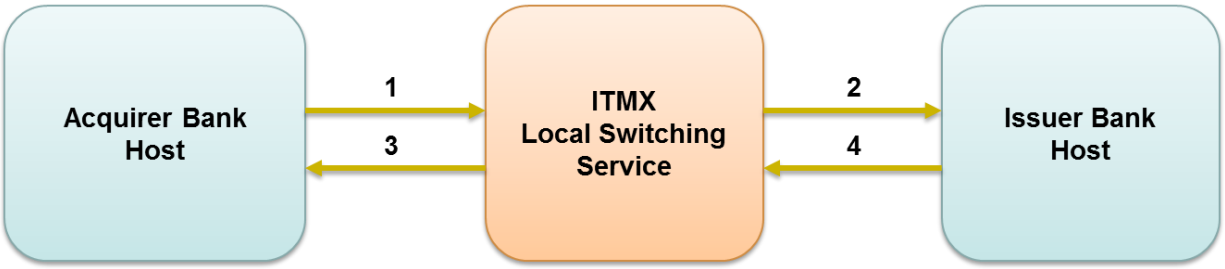


Figure 5–2: Authorization message flow

1. Flow (1) Acquirer Bank send Authorization to ITMX

1.1 If Authentication Success, send TXN ID, Bank Token and ECI=05 - Fully Authentication

1.2 If Authentication not Success, send only ECI=06, or 07.

2. Flow (1) ITMX verify Bank Token with Key A, Key B of issuer and forwards Bank Token result code to Issuer Flow (2).

2.1 If issuer bank chooses Verify Bank Token themselves ITMX will send blank - no check; pass through.

2.2 If acquirer send Bank Token invalid format, ITMX will send 0 - invalid token format.

2.3 If acquirer send Bank Token and ITMX cannot validate, ITMX will send 1 - token validation failed.

2.4 If acquirer send Bank Token and ITMX can validate, ITMX will send 2 - token validation passed.

3. Flow (3) (4) Issuer send response to ITMX and then ITMX send to Acquirer

Two options for member to select.

Option 1: Authentication

- Member generate LSAV.

Authorization

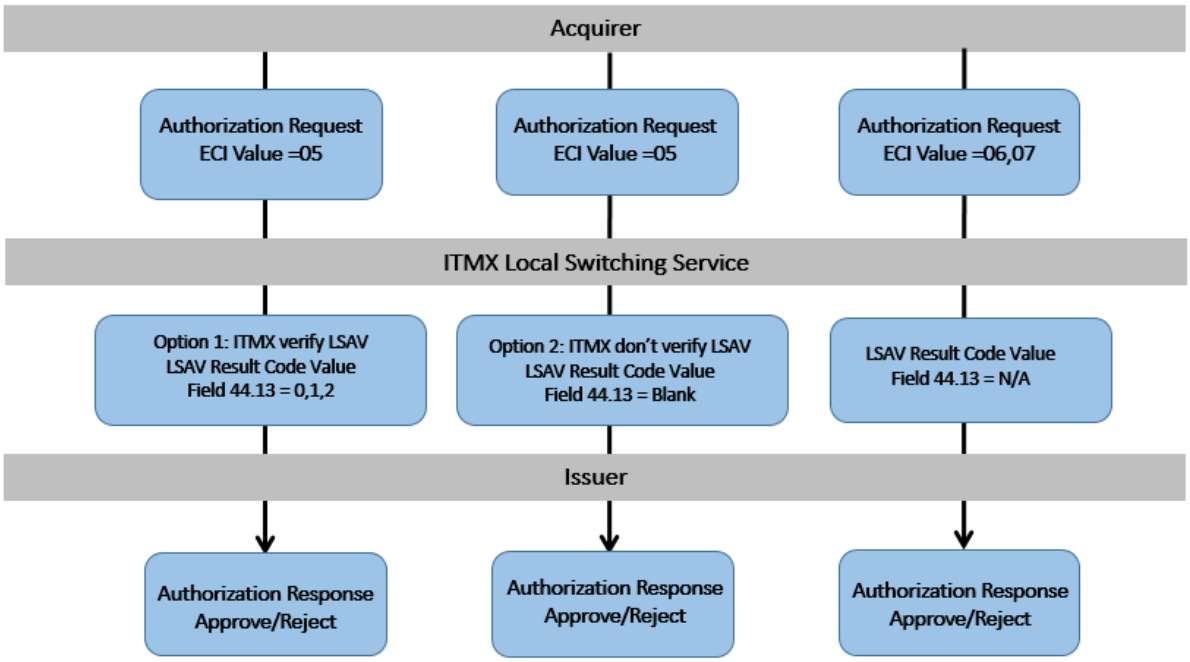
- ITMX verify LSAV with Member Key A, B which eject at ITMX.
- Send verification result to issuing bank.

Option 2: Authentication

- Member generate LSAV.

Authorization

- ITMX don't verify LSAV.



Authorization message flow

5.5 PSEUDO CODE FOR E-COMMERCE

Refer to Section 3.4.2.4 ISO8583 DATA Element Definition in ITMX Local Switching Member Bank Technical Specification.

if ((DE22 == 01 or 00) && DE25 == 59 && DE60.2 == 1 && DE60.8 == 05) # Secure e-commerce

DE35 is off

DE126 is on

DE126.8 is optional

DE126.9 is on

DE126.15 is off

DE126.16 is off

else

if ((DE22 == 01 or 00) && DE25 == 59 && DE60.2 == 1 && (DE60.8 == 06 or 07)) # Non Secure e-commerce

DE35 is off

DE126 is on

DE126.15 is off

DE126.16 is off

end-if

Additional:

- Secure => DE60.8 = 05
  - => DE126.8 Contain Transaction ID/XID (Optional)
  - => DE126.9 Contain LSAV
  - => DE44.13 Contain LSAV Result Code
    - blank -> no check; pass through
    - 0 -> invalid token format
    - 1 -> token validation failed
    - 2 -> token validation passed
- Non-Secure => DE60.8 = 06, 07
  - => DE126.8, DE126.9 = ‘OFF Bit’

Chapter 6

6. CLEARING PROCESSING

6.1 INTERCHANGE TRANSACTION FILES FORMAT

For the Outgoing Transaction File (OTF) and Incoming Transaction File (ITF) has field Mail/Phone/Electronic Commerce and Payment Indicator position 118 in Detail Record Sequence #1 follow by

- Existing valid value for ITMX LSS EMVCo by
  - 5 – Secure E-Commerce Transaction
  - 6 – Not-Authenticated Security, merchant attempted to authenticate
  - 7 – Non-Authenticated Security Transaction, Non-Secure Transaction

Field	Format	Length	Decimal	Position	Remark
Mail/Phone/Electronic Commerce and Payment Indicator	Char	1		118	Indicates transaction performed by mail order, telephone, or electronic commerce. Valid values are: Space = Field not applicable or acquirer did not specify. 1 = Mail/Phone Order (MO/TO). 2 = Recurring transaction 5= Secure Electronic Commerce Transaction. 6 = Non-Authenticated Security Transaction at a 3D Secure-capable merchant, and merchant attempted to authenticate the cardholder using 3-D secure. 7 = Non-Authenticated Security Transaction To support LSS 1.0, ITMX defines ECI as below: A = Non Secure / Merchant not Participate B = Issuer not participate C = Cardholder not enrolment D = Fully Authentication

Existing OTF/ITF Layout

6.2 BIN FILE FORMAT

6.2.1 BIN FILE HEADER RECORD LAYOUT (EXISTING)

Field Name	Position	Length	Format	Description
Table Type	1-2	02	AN	AA File Header Record.
Table Mnemonic	3-10	08	AN	FILEHEAD
Record Type	11	01	AN	A Header Record.
Table Key	12-23	12	AN	EPACRN (left-justified, space-filled).
Effective Date	24-31	08	UN	Date in YYYYMMDD format.
Filler	32	01	AN	Set to spaces.
Filler	33-36	04	AN	Set to spaces.
Filler	37-39	03	UN	Set to spaces.
Separator	40	01	S	Comma.
Filler	41-44	04	AN	Set to spaces.
Release Number	45-47	03	UN	Set to spaces.
Separator	48	01	S	Comma
Filler	49	01	AN	Set to spaces.
Separator	50	01	S	Comma
Filler	51	01	AN	Set to spaces.
Separator	52	01	S	Comma
Filler	53	01	AN	Set to spaces.
Separator	54	01	S	Comma
Filler	55	08	AN	Set to spaces.
Separator	63	01	S	Comma
Filler	64-67	04	UN	Set to spaces.
Reserved	68-100	33	AN	Set to spaces.

BIN Table Header Record Layout

Field Name	Position	Length	Format	Description
Table Type	1-2	02	AN	VL Value Table Record.
Table Mnemonic	3-10	08	AN	ACTRNG
Record Type	11	01	AN	A Table Header Record.
Table Key	12-23	12	AN	Blanks
Effective Date	24-31	08	UN	Date in YYYYMMDD format.
Filler	32	01	AN	Set to spaces.
Incoming Table Load Indicator-	33	01	AN	I Load table for incoming process.
Outgoing Table Load Indicator-	34	01	AN	O Load table for outgoing process.
Edit Table Load Indicator	35	01	AN	E Load table for edit process.

Report Table Load Indicator	36	01	AN	Blank
Separator	37	01	S	Comma
Length of Table Key	38-39	02	UN	09
Separator	40	01	S	Comma
Storage Technique Indicator	41	01	AN	U Regions most used in main storage.
Separator	42	01	S	Comma
Search Technique Indicator	43	01	AN	R Binary search, with range
Length of the Attributes	44 - 45	02	UN	36
Field Name Mnemonic	46-53	08	AN	ARATTR (left-justified, space-filled).
Filler	54	01	UN	Set to Spaces.
Filler	55-56	02	UN	Set to Spaces.
Filler	57-58	02	UN	Set to Spaces.
Reserved	59-100	42	AN	Set to spaces.

BIN Value Table Default Record Layout

Field Name	Position	Length	Format	Description
Table Type	1-2	02	AN	VL Value Table Record.
Table Mnemonic	3-10	08	AN	ACTRNG
Record Type	11	01	AN	D Detail Record.
Filler	12-23	12	AN	Set to spaces.
Effective Date	24-31	08	UN	Date in YYYYMMDD format.
Filler	32	01	AN	Set to spaces.
Reserved	33-100	68	AN	Set to spaces

BIN Value Table Detail Record Layout

Field Name	Position	Length	Format	Description
Table Type	1-2	02	AN	VL Value Table Record.
Table Mnemonic	3-10	08	AN	ACTRNG
Record Type	11	01	AN	R Detail Record.
Table Key	12-23	12	AN	The key for the record is the high range ARDEF value. For example, 401299999.
Effective Date	24-31	08	UN	Date in YYYYMMDD format.
Delete Indicator	32	01	AN	Blank This is not a delete record. D This is a delete record (to be deleted after the effective date specified in the preceding field).
Filler	33-44	12	AN	Set to spaces
Issuer BIN	45-50	06	UN	Reporting BIN number. For example, 401200.
Check Digit Algorithm	51	01	UN	No check digit validation performed.

Account Number Length	52-53	02	UN	Set to zeroes
Card Type	54	01	AN	A : ATM B : Visa Business C : Visa Classic D : Visa Commerce E : Visa Electron F : COPAC G : Visa Travel/Money/Card H : Visa Preferred Card I : Visa Infinite Card J : Visa Platinum K : Visa Signature L : National BIN M : Visa Gold O : Visa Signature Business P : Premier Q : Proprietary R : Corporate T&E Card S : Purchasing Card T : Trav l Voucher V : VPAY W : Debit Union Pay 1 : MC Classic 2 : MC Gold 3 : MC Premium 4 : MC Super Premium 5 : MC Elite
Authentication Effective Date	**55-62**	**08**	**UN**	**Effective Date for LSS Authentication (2 Versions) in YYYYMMDD format.**
Filler	63	01	UN	Set to spaces.
Filler	64-65	02	AN	Set to spaces.
Filler	66	01	AN	Set to spaces.
Filler	67	01	AN	Set to spaces.
Filler	68	01	UN	Set to spaces.
Filler	69-70	02	AN	Set to spaces.
Filler	71	1	AN	Set to spaces.
Filler	72	1	AN	Set to spaces.
Filler	73	1	AN	Set to spaces.
Filler	74	1	AN	Set to spaces.
Filler	75	01	AN	Set to spaces.
Filler	76	01	AN	Set to spaces.
Filler	77	01	AN	Set to spaces.
Filler	78	01	AN	Set to spaces.
Filler	79-80	02	AN	Set to spaces.
Reserved	81-98	18	AN	Set to spaces
Filler	99	01	AN	Set to spaces
Reserved	100	01	AN	Set to spaces

BIN File Trailer Record Layout

Field Name	Position	Length	Format	Description
Table Type	1-2	02	AN	zz File Trailer Record
Table Mnemonic	3-10	08	AN	FILETAIL
Record Type	11	01	AN	z File Trailer Record
Filler	12-23	12	AN	Set to spaces
Effective Date	24-31	08	UN	Date in YYYYMMDD format.
Filler	32	01	AN	Set to spaces
Filler	33-38	06	UN	Set to zeros
Filler	39-63	25	AN	Set to spaces
Filler	64-67	04	UN	Set to spaces
Reserved	68-100	33	AN	Set to spaces.

--- END OF DOCUMENT ---