EMV®

3-D Secure

Protocol and Core Functions Specification

Version 2.0.0

October 2016

Legal Notice

The EMV® Specifications are provided "AS IS" without warranties of any kind, and EMVCo neither assumes nor accepts any liability for any errors or omissions contained in these Specifications. EMVCO DISCLAIMS ALL REPRESENTATIONS AND WARRANTIES, EXPRESS OR IMPLIED, INCLUDING WITHOUT LIMITATION IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT, AS TO THESE SPECIFICATIONS.

EMVCo makes no representations or warranties with respect to intellectual property rights of any third parties in or in relation to the Specifications. EMVCo undertakes no responsibility to determine whether any implementation of the EMV® Specifications may violate, infringe, or otherwise exercise the patent, copyright, trademark, trade secret, know-how, or other intellectual property rights of third parties, and thus any person who implements any part of the EMV® Specifications should consult an intellectual property attorney before any such implementation.

Without limiting the foregoing, the Specifications may provide for the use of public key encryption and other technology, which may be the subject matter of patents in several countries. Any party seeking to implement these Specifications is solely responsible for determining whether its activities require a license to any such technology, including for patents on public key encryption technology. EMVCo shall not be liable under any theory for any party's infringement of any intellectual property rights in connection with the EMV® Specifications.

Contents

Figures

Tables

This page intentionally left blank.

1 Introduction

The 3-D Secure authentication protocol is based on a three-domain model where the Acquirer Domain and Issuer Domain are connected by the Interoperability Domain for the purpose of authenticating a Cardholder during an electronic commerce (e-commerce) transaction or to provide identity verification.

The 3-D Secure authentication protocol supports both:

- **Payment Authentication**—Cardholder authentication during an e-commerce transaction.
- **Non-Payment Authentication—**Identity verification.

The 3-D Secure authentication protocol can be both:

- **App-based—**Authentication during a transaction on a Consumer Device that originates from an App provided by a registered agent 3DS Requestor (merchant, digital wallet, et al). For example, an e-commerce transaction originating during a check-out process within a merchant’s app.
- **Browser-based—**Authentication during a transaction on a Consumer device that originates from a website utilising a browser. For example, an e-commerce transaction originating during a check-out process within a website on a Consumer Device.

1.1 Purpose

The purpose of this *EMV 3-D Secure Protocol and Core Functions Specification* is to describe the EMV 3-D Secure infrastructure and components, and to specify the requirements for each component within the infrastructure and their interaction.

For purposes of this document, when the phrase 3-D Secure, and/or 3DS is utilised, the intent is EMV 3-D Secure.

1.2 Audience

This document is intended for stakeholders developing EMV 3-D Secure products and supporting 3-D Secure implementations.

1.3 Normative References

The following standards contain provisions that are referenced in this specification. The latest version including all published amendments shall apply unless a publication date is explicitly stated.

Table 1.1: Normative References

Reference	Publication Name	Bookmark
ITU; ITU-T. E.164	*The International Public Telecommunication Numbering Plan*	https://www.itu.int/rec/T-REC-E.164/en
RFC 2045	*Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies*	https://tools.ietf.org/html/rfc2045
RFC 2616	*Hypertext Transfer Protocol -- HTTP/1.1*	https://tools.ietf.org/html/rfc2616
RFC 3447	*PKCS #1: RSA Cryptography Specifications*	https://www.ietf.org/rfc/rfc3447.txt
RFC 4122	*A Universally Unique IDentifier (UUID) URN Namespace*	https://tools.ietf.org/html/rfc4122
RFC 4158	Internet X.509 Public Key Infrastructure: certification Path Building	https://tools.ietf.org/html/rfc4158
RFC 5322	*Internet Message Format*	https://tools.ietf.org/html/rfc5322
RFC 7159	*The JavaScript Object Notation (JSON) Data Interchange Format*	https://tools.ietf.org/html/rfc7159
RFC 7231	*Hypertext Transfer Protocol (HTTP/1.1): Semantics and Content*	https://tools.ietf.org/html/rfc7231
RFC 7515	JSON Web Signatures (JWS)	https://tools.ietf.org/html/rfc7515
RFC 7516	JSON Web Encryption (JWE)	https://tools.ietf.org/html/rfc7516
RFC 7517	JSON Web Key (JWK)	https://tools.ietf.org/html/rfc7517
RFC 7518	JSON Web Algorithms (JWA)	https://tools.ietf.org/html/rfc7518

1.4 Acknowledgment

The following ISO Standards are referenced in this specification.

Table 1.2: ISO Standards

Reference	Publication Name	Bookmark
ISO 3166	Country Codes—ISO 3166	http://www.iso.org/iso/country_codes
ISO 4217	*Currency Codes—ISO 4217*	http://www.iso.org/iso/home/standards/currency_codes.htm
ISO/IEC 7812-1:2015	*ISO/IEC 7812-1:2015 Identification cards—Identification of issuers—Part 1: Numbering system*	http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=66011
ISO/IEC 7813:2016	*ISO/IEC 7813:2016 Information technology—Identification cards—Financial transaction cards*	http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=43317
ISO/IEC 15946 1	ISO/IEC 15946 1 *Information technology—Security techniques—Cryptographic techniques based on elliptic curves—Part 1: General*	http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=65480

1.5 Definitions

The following terms are used in this specification:

Table 1.3: Definitions

Term	Definition
3DS Client	The consumer-facing component allowing consumer interaction with the 3DS Requestor for initiation of the EMV 3-D Secure protocol.
3DS Integrator	An EMV 3-D Secure participant that facilitates and integrates the 3DS Requestor Environment, and optionally facilitates integration between the Merchant and the Acquirer.
3DS Method	A scripting call provided by the 3DS Integrator that is placed on the 3DS Requestor website. Optionally used to obtain additional browser information to facilitate risk-based decisioning.
3DS Requestor	The initiator of the EMV 3-D Secure Authentication Request. For example, this may be a merchant or a digital wallet requesting authentication within a purchase flow.
3DS Requestor App	An App on a Consumer Device that can process a 3-D Secure transaction through the use of a 3DS SDK. The 3DS Requestor App is enabled through integration with the 3DS SDK.
3DS Requestor Environment	The 3DS Requestor-controlled components (3DS Requestor App, 3DS SDK, and 3DS Server) are typically facilitated by the 3DS Integrator. Implementation of the 3DS Requestor Environment will vary as defined by the 3DS Integrator.
3DS Requestor Website	Component that provides the website that requests Cardholder credentials (whether on file or entered by Cardholder).
3DS SDK	3-D Secure Software Development Kit (SDK). A component that is incorporated into the 3DS Requestor App. The 3DS SDK performs functions related to 3-D Secure on behalf of the 3DS Server.
3DS Server	Refers to the 3DS Integrator's server or systems that handle online transactions and facilitates communication between the 3DS Requestor and the DS.
3-D Secure (3DS)	An e-commerce authentication protocol that enables the secure processing of payment and non-payment card transactions.
Abandon	The act of a Cardholder leaving a transaction by use of the Cancel action while in the process of a challenge. For example, using the Cancel button in the App challenge UI.

Term	Definition

Absent	Used in this specification to indicate that an element is absent when the name/value pair does not occur in the message.
	For example, element "firstName" is absent in the following JSON instance:
	{
	“lastName”:”Smith”
	}

Access Control Server (ACS)	A component that operates in the Issuer Domain, that verifies whether authentication is available for a card number and device type, and authenticates specific Cardholders.
Access Control Server User Interface (ACS UI)	The ACS UI is generated during a Cardholder challenge and is rendered by the ACS within a Browser challenge window.
Acquirer	A financial institution that establishes a contractual service relationship with a Merchant for the purpose of accepting payment cards. In the context of 3-D Secure, in addition to the traditional role of receiving and sending authorisation and settlement messages (enters transaction into interchange), the Acquirer also determines whether a Merchant is eligible to support the Merchant’s participation in 3-D Secure.
Acquirer Domain	Contains the systems and functions of the 3DS Requestor Environment and, optionally the Acquirer.
Attempts	In this specification, used to indicate the process by which proof of an authentication attempt is generated when payment authentication is not available. Support for Attempts is determined by each DS.
Authentication	In the context of 3-D Secure, the process of confirming that the person making an e-commerce transaction is entitled to use the payment card.
Authentication Request (AReq) Message	An EMV 3-D Secure message sent by the 3DS Server via the DS to the ACS to initiate the authentication process.
Authentication Response (ARes) Message	An EMV 3-D Secure message returned by the ACS via the DS in response to an Authentication Request message.
Authentication Value (AV)	A cryptographic value generated by the ACS to provide a way, during authorisation processing, for the authorisation system to validate the integrity of the authentication result. The AV algorithm is defined by each Payment System.
Authorisation	A process by which an Issuer, or a processor on the Issuer's behalf, approves a transaction for payment.
Authorisation System	The systems and services through which a Payment System delivers online financial processing, authorisation, clearing, and settlement services to Issuers and Acquirers.

Term	Definition
Bank Identification Number (BIN)	The first six digits of a payment card account number that uniquely identifies the issuing financial institution. Also referred to as Issuer Identification Number (IIN) in ISO 7812.
Base64	Encoding applied to the AReq, ARes, CReq, and CRes messages as defined in RFC 2045.
Browser	In the context of 3-D Secure, the browser is a conduit to transport messages between the 3DS Server (in the Acquirer Domain) and the ACS (in the Issuer Domain).
Card	In this specification, synonymous to the account of a payment card.
Cardholder	An individual to whom a card is issued or who is authorised to use that card.
Certificate	An electronic document that contains the public key of the certificate holder and which is attested to by a Certificate Authority (CA) and rendered not forgeable by cryptographic technology (signing with the private key of the CA).
Certificate Authority (CA)	A trusted party that issues and revokes certificates. Refer also to DS Certificate Authority.
Challenge	The process where the ACS is in communication with the 3DS Client to obtain additional information through Cardholder interaction.
Challenge Flow	A 3-D Secure flow that involves Cardholder interaction as defined in Section 2.5.2.
Challenge Request (CReq) Message	An EMV 3-D Secure message sent by the 3DS SDK or 3DS Server where additional information is sent from the Cardholder to the ACS to support the authentication process.
Challenge Response (CRes)	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.
Consumer Device	Device used by a Cardholder such as a smartphone, laptop, or tablet that the Cardholder uses to conduct payment activities including authentication and purchase.
Device Channel	Indicates the channel from which the transaction originated. Either: • App-based • Browser-based
Device Information	Data provided by the Consumer Device that is used in the authentication process.

Term	Definition
Digital signature	An asymmetric cryptographic method whereby the recipient of the data can prove the origin and integrity of data, thereby protecting the sender of the data and the recipient against modification or forgery by third parties and the sender against forgery by the recipient.
Digital wallet	A software component that allows a user to make an electronic payment with a financial instrument (such as a credit card) while hiding the low level details of executing the payment protocol, including such tasks as entering an account number and providing shipping information and Cardholder identifying information.
Directory Server (DS)	A server component operated in the Interoperability Domain; it performs a number of functions that include: authenticating the 3DS Server, routing messages between the 3DS Server and the ACS, and validating the 3DS Server, the 3DS SDK, and the 3DS Requestor.
Directory Server Certificate Authority (DS CA)	A component that operates in the Interoperability Domain; generates and distributes selected digital certificates to components participating in 3-D Secure. Typically, the Payment System to which the DS is connected operates the CA.
Electronic Commerce Indicator (ECI)	Payment System-specific value provided by the ACS to indicate the results of the attempt to authenticate the Cardholder.

Empty	An element is empty if the field name is present and the value is empty. For example, element "firstName" has no data in the following JSON instance.
	{
	“firstName”:””, “lastName”:”Smith”
	}

EMV	A term referring to EMVCo’s specifications for global interoperability and acceptance of secure payment transactions and/or products and services complying with such specifications.
EMVCo	EMVCo, LLC, a limited liability company incorporated in Delaware, USA.
Ends 3-D Secure Processing	In the 3-D Secure processing flow, this indicates that no further processing as defined by this specification will be performed. Per merchant preferences, an authorisation transaction may still be performed although it will happen without a successful 3-D Secure authentication outcome.

Term	Definition
Ends processing	In the 3-D Secure processing flow, this indicates that an error has been found by a specific 3-D Secure component, which reports the error via the appropriate Error Message or message with an IReq code as defined in this specification. The specific 3-D Secure component reports the error to the component from which the erroneous message was received, and may inform other components about the error and will stop further 3-D Secure processing. The subsequent 3-D Secure components in the authentication flow will still perform further execution of the received message with an IReq code or Error message to close the error situation diligently.
FIDO Authenticator	An authentication entity that meets the FIDO Alliance's requirements and which has related metadata. A FIDO Authenticator is responsible for user verification, and maintaining the cryptographic material required for the relying party authentication. For additional information, refer to: https://fidoalliance.org.
Frictionless	The process of authentication achieved without Cardholder interaction.
Frictionless Flow	A 3-D Secure flow that involves Cardholder interaction as defined in Section 2.5.1.
Interaction Counter	The number of interactions for each transaction is tracked by the ACS and sent with the RReq message to the Directory Server (DS). Used by the ACS to set a maximum number of cardholder interactions as determined by the selected Challenge Flows and security requirements to allow an appropriate number of cardholder retries without going beyond a pre-set maximum.
Interoperability Domain	Facilitates the transfer of information between the Issuer Domain and Acquirer Domain systems.
Invalid Request Code (IReq)	Code returned in a message response that indicates a problem identified in a 3DS request message.
Issuer	A financial institution that issues payment cards, contracts with Cardholders to provide card services, determines eligibility of Cardholders to participate in 3-D Secure, and identifies for the Directory Server card number ranges eligible to participate in 3-D Secure.
Issuer Domain	Contains the systems and functions of the Issuer and its customers (Cardholders).
JavaScript Object Notation (JSON)	An open standard format that uses human-readable text to transmit data objects consisting of attribute-value pairs. It is typically used to transmit data between a server and web application. Refer to Table 1.1 for RFC references.
Key	In cryptography, the value needed to encrypt and/or decrypt a value.

Term	Definition
Key management	The handling of cryptographic keys and other security parameters during the entire lifetime of the keys, including generation, storage, entry and use, deletion or destruction, and archiving.
MAC	Message Authentication Code. A symmetric (secret key) cryptographic method that protects the sender and recipient against modification and forgery of data by third parties.
Merchant	Entity that contracts with an Acquirer to accept payment cards. Manages the online shopping experience with the Cardholder, obtains card number, and then transfers control to the 3DS Server, which conducts payment authentication.
Message Category	Indicates the category of the EMV 3-D Secure message. Either: • Payment (PA) • Non-Payment (NPA)

Missing	An element is missing either if it is absent (that is the name/value pair does not occur in the message) or if the field name is present and value is empty. For example, element "firstName" has no data in both of the following JSON instances.
	Example of empty field name present and value empty:
	{
	“firstName”:””, “lastName”:”Smith”
	}
	Example of absent name/value pair:
	{
	“lastName”:”Smith”
	}

NameValuePair (NVP)	A simple class encapsulating an attribute/value pair.
Native	Refers to the original method for a device display, utilising its own APIs.
One-Time Passcode (OTP)	A passcode that is valid for only one login session or transaction, on a computer system or other digital device.
Out-of-Band (OOB)	A Challenge activity that is completed outside of, but in parallel to, the 3-D Secure flow. The final Challenge Request is not used to carry the data to be checked by the ACS but signals only that the authentication has been completed. ACS authentication methods or implementations are not defined by the 3-D Secure specification.
Payment System	A Payment System defines the operating rules and conditions, and the requirements for card issuance and Merchant acceptance.

Term	Definition
Preparation Request (PReq) Message	3-D Secure message sent from the 3DS Server to the DS to request the ACS versions that correspond to the DS card ranges as well as an optional 3DS Method URL to update the 3DS Server's internal storage information.
Preparation Response (PRes) Message	Response to the PReq message that contains the DS Card Ranges and 3DS Method URL so that updates can be made to the 3DS Server's internal storage.
Private key	Part of an asymmetric cryptographic system. The key that is kept secret and known only to an owner.
Proof of authentication attempt	Refer to Attempts.
Public key	Part of an asymmetric cryptographic system. The key known to all parties.
Public key pair	Two mathematically related keys—a public key and a private key—that are used with a public key (asymmetric) cryptographic algorithm to permit the secure exchange of information without the necessity for a secure exchange of a secret.
Results Request (RReq) Message	Message sent by the ACS via the DS to transmit the results of the authentication transaction to the 3DS Server.
Results Response (RRes) Message	Message sent by the 3DS Server to the ACS via the DS to acknowledge receipt of the Results Request message.
Secret key	A key used in a symmetric cryptographic algorithm such as DES which, if disclosed publicly, would compromise the security of the system.
Transport Layer Security (TLS)	A cryptographic protocol developed by the IETF (Internet Engineering Task Force) to confidentially transmit information over open networks, such as the Internet. Refer to Table 1.1 for RFC references.
Uniform Resource Locator (URL)	Address scheme for pages on the World Wide Web usually in the format http://www.example.com or https://www.example.com.
Universally Unique Identifier (UUID)	Identifier standard used in software construction. In its canonical form, a UUID is represented by 32 lowercase hexadecimal digits, displayed in five groups separated by hyphens, in the form 8-4-4-4-12 for a total of 36 characters (32 alphanumeric characters and four hyphens). Refer to Table 1.1 for RFC references.
Validate	In this specification, the process of checking a message against the requirements for presence and format of each data element in the message as defined in Table A.1 and detailed outline in Section 5.1.6. Refer to Section 5.9 for additional information.
Verify	In this specification, the process of checking a message cryptographically as defined in Section 6.2. Refer to Section 5.9 for additional information.
Wallet	Refer to Digital wallet.

Term	Definition
X.509	Certificate format as defined in RFC 4158.

1.6 Abbreviations

The abbreviations listed in Table 1.4 are used in this specification.

Table 1.4: Abbreviations

Abbreviation	Description
3DS	Three Domain Secure
3DS SDK	Three Domain Secure Software Development Kit
ACS	Access Control Server
AReq	Authentication Request
ARes	Authentication Response
AV	Authentication Value
BIN	Bank Identification Number
CA	Certificate Authority
CA DS	Certificate Authority Directory Server
CReq	Challenge Request
CRes	Challenge Response
DS	Directory Server
ECI	Electronic Commerce Indicator
IReq	Invalid Request Code
JSON	JavaScript Object Notation
MAC	Message Authentication Code
NPA	Non-Payment Authentication
NVP	NameValuePair
OOB	Out-of-Band

Abbreviation	Description
PA	Payment Authentication
OTP	One-time Passcode
PReq	Preparation Request Message
PRes	Preparation Response Message
RReq	Results Request Message
RRes	Results Response Message
SDK	Software Development Kit
TLS	Transport Layer Security
URL	Uniform Resource Locator
UUID	Universally Unique Identifier

1.7 Supporting Documentation

The following documents are specific to the EMV 3-D Secure protocol and should be used in conjunction with this specification. These documents are located on www.emvco.com.

- *EMV 3-D Secure SDK Specification*
- *EMV 3-D Secure SDK—Implementation Guide*
- *EMV 3-D Secure SDK—Device Information*

1.8 Terminology and Conventions

The following words are used often in this specification and have a specific meaning:

Shall

Defines a product or system capability which is mandatory.

May

Defines a product or system capability which is optional or a statement which is informative only and is out of scope for this specification.

Should

Defines a product or system capability which is recommended.

Ends 3-D Secure Processing

As outlined in Chapter 3, defines a specific exception scenario in the 3-D Secure authentication flows where further processing is outside the scope of this specification. Refer to Table 1.3 for additional information.

Ends Processing

As outlined in Chapter 3, defines a specific exception scenario in the 3-D Secure authentication flows where a 3-D Secure component experiences an error and does not process the transaction normally. Therefore, subsequent components take action on the error instance. Refer to Table 1.3 for additional information.

This page intentionally left blank.

2 EMV 3-D Secure Overview

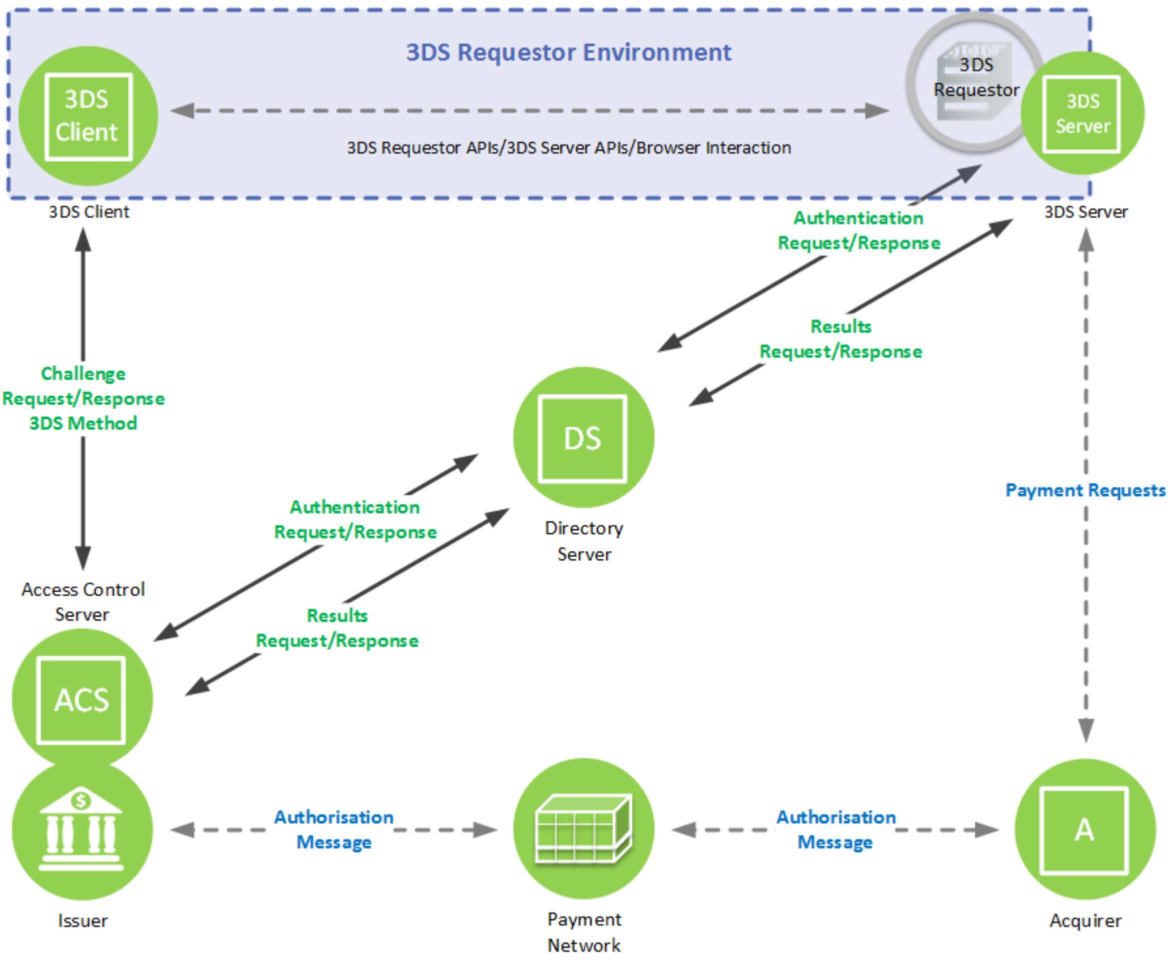
This overview describes the components, the systems, and the functions necessary to implement 3-D Secure. Descriptions are divided into the following domains:

- **Acquirer Domain**—3-D Secure transactions are initiated from the Acquirer Domain
- **Interoperability Domain**—3-D Secure transactions are switched between the Acquirer Domain and Issuer Domain
- **Issuer Domain**—3-D Secure transactions are authenticated in the Issuer Domain

Figure 2.1 depicts the interaction of the three domains and the components of each.

Because the implementation of the 3DS Requestor Environment may vary, the diagram purposefully does not imply a specific implementation of these components or how they interoperate. For example, the 3DS Client may communicate directly with the 3DS Server, or the 3DS Server and 3DS Requestor may be functionally combined.

Figure 2.1: 3-D Secure Domains and Components



Note: Dashed arrows and 3DS Requestor are not part of 3DS specification and are shown for clarity only

2.1 Acquirer Domain

The Acquirer Domain has the following components:

- 3DS Requestor Environment
    - 3DS Requestor
    - 3DS Client
    - 3DS Server
- 3DS Integrator
- Acquirer (for Payment Authorisation)

2.1.1 3DS Requestor Environment

The 3DS Requestor Environment is a collective term for components under the 3DS Requestor's control that support 3-D Secure. The 3DS Requestor Environment components include:

- 3DS Requestor
- 3DS Client
- 3DS Server

2.1.1.1 3DS Requestor

The 3DS Requestor initiates the AReq message and is the conduit for the 3-D Secure data from the Consumer Device. For example, in payment authentication, the 3DS Requestor typically represents the existing Merchant web server for online shopping.

The 3DS Requestor has a relationship with the 3DS Client either via the 3DS Requestor App, or the 3DS Method/Browser on the Consumer Device. The 3DS Requestor has a link to or integration with the 3DS Server.

To process 3-D Secure transactions:

- **App-based**—3DS Requestor App integrates the 3DS SDK as defined in the *EMV 3-D Secure SDK Specification*. The 3DS SDK displays the User Interface (UI) to Cardholders.
- **Browser-based**—3DS Requestor Browser-based solution utilises the 3DS Method to gather browser information/device details and the ACS provides HTML to the Browser to display the UI to the Cardholder when a challenge is necessary.

2.1.1.2 3DS Client

The 3DS Client is the component on a Consumer Device that initiates a 3-D Secure authentication. For example, in payment authentication, the 3DS Client is integrated with the Merchant checkout as part of an online shopping experience.

The 3DS Client can be implemented in two models:

- **App-based**—The 3DS Client is the 3DS SDK that is integrated with the 3DS Requestor App and facilitates the Cardholder interaction.

  The 3DS SDK gathers 3-D Secure information from the Consumer Device, supports the authentication of the Access Control Server (ACS), and protects the Cardholder authentication data flow.

- **Browser-based**—The 3DS Client is the 3DS Method that is integrated with the 3DS Requestor's website and is invoked within a browser on the Consumer Device.

  The 3DS Method is a scripting call provided by the 3DS Integrator placed on the website on which the Cardholder is interacting, such as a Merchant checkout page in a payment transaction. The purpose of the 3DS Method is to obtain additional browser information to help facilitate risk-based decisioning.

2.1.1.3 3DS Server

The 3DS Server provides the functional interface between the 3DS Requestor Environment flows and the DS. The 3DS Server is responsible for:

- Collecting necessary data elements for 3-D Secure messages
- Authenticating the DS
- Validating the DS, the 3DS SDK, and the 3DS Requestor
- Ensuring that message contents are protected

To initiate a 3-D Secure authentication, the 3DS Server collects the necessary data elements from any or all of the components within the 3DS Requestor Environment. For example, in payment authentication, the Cardholder could provide account information using a Consumer Device, or the information could be held on file within the 3DS Requestor Environment. Device Information is obtained by the 3DS Client and forwarded to the 3DS Server.

> **Note: Following payment authentication, depending on the 3DS Requestor configuration, the 3DS Server may also link to the Acquirer and initiate authorisation requests.**

2.1.2 3DS Integrator (3DS Server and 3DS Client)

The 3DS Integrator role provides the functional interface between the 3DS Requestor Environment and the 3-D Secure messages.

The role of the 3DS Integrator is to provision the 3DS Server and the 3DS Client, and to integrate the 3-D Secure functionality with the 3DS Requestor business functionality. This function is critical to interfacing with the DS and the ACS within the authentication messaging, while also acting as the conduit for challenge results when performed. The 3DS Integrator provides the approved 3DS SDK component or the 3DS Method functionality to 3DS Requestors for integration into their 3DS Requestor App and/or website.

The 3DS Integrator is responsible for registration of Merchants with all required DSs.

> **Note: Following payment authentication, the 3DS Integrator can offer authorisation functionality with an Acquirer.**

2.1.3 Acquirer (Payment Authorisation)

An Acquirer is a financial institution that:

- Enters into a contractual relationship with a Merchant for the purpose of accepting payment card transactions
- Supports the Merchant's participation in 3-D Secure

Following a 3-D Secure payment authentication, the Acquirer performs its traditional role, which involves:

- Receiving authorisation requests from the Merchant
- Sending authorisation requests to the authorisation system
- Providing authorisation responses to the Merchant
- Submitting the completed transactions to the settlement system

2.2 Interoperability Domain

The Interoperability Domain has the following components:

- Directory Server (DS)
- Directory Server Certificate Authority (DS CA)
- Authorisation System

2.2.1 Directory Server

The DS performs a number of functions that include:

- Authenticating the 3DS Server and the ACS
- Routing messages between the 3DS Server and the ACS
- Validating the 3DS Server, the 3DS SDK, and the 3DS Requestor
- Defining specific programme rules (for example, logos, time-out values, etc.)
- Onboarding 3DS Servers and ACSs
- Maintaining ACS versions and 3DS Method URLs

2.2.2 Directory Server Certificate Authority

The DS CA generates the DS Public Key to the 3DS SDK and generates Transport Layer Security (TLS) certificates for use by 3-D Secure components. The DS CA is typically operated by the Payment System responsible for a specific DS.

These certificates include:

- TLS client and server certificates used in the communication between the 3DS Server and the DS, and between the DS and the ACS
- Signing certificates used to sign messages passed from the ACS to the 3DS SDK and from the ACS to the 3DS Server

Refer to Chapter 6 for detailed information about certificates.

2.2.3 Authorisation System (Payment Authentication)

The Authorisation System performs its traditional role after payment authentication, which involves:

- Receiving authorisation requests from the Acquirer
- Sending authorisation requests to the Issuer
- Providing authorisation responses to the Acquirer
- Providing clearing and settlement services to the Acquirer and the Issuer

2.3 Issuer Domain

The Issuer Domain has the following components:

- Cardholder
- Consumer Device
- Issuer
- Access Control Server (ACS)

2.3.1 Cardholder

The Cardholder provides account information using a Consumer Device. If necessary, the Cardholder is prompted to provide additional information for authentication.

2.3.2 Consumer Device

The Consumer Device has the capability to run a 3DS Requestor App or present a website on a browser that can be used for 3-D Secure authentication.

The Consumer Device-based components of the 3DS Requestor Environment depend on the model:

- **App-based**—the 3DS SDK integrated with the 3DS Requestor App
- **Browser-based**—a browser utilising the 3DS Method

These components have a specific relationship with the 3DS Requestor and 3DS Server.

2.3.3 Issuer

An Issuer is a financial institution that:

- Enters into a contractual relationship with the Cardholder for issuance of one or more payment cards
- Defines card number ranges eligible to participate in 3-D Secure
- Provides card number ranges to be added to the applicable DS

2.3.4 Access Control Server

The ACS contains the authentication rules and is controlled by the Issuer. ACS functions include:

- Verifying whether a card number is eligible for 3-D Secure authentication
- Verifying whether a Consumer Device type is eligible for 3-D Secure authentication
- Authenticating the Cardholder for a specific transaction

While these functions may belong to a single logical ACS, implementations may divide the processing by function or other characteristics (for example, card number range) among multiple physical servers.

2.4 3-D Secure Messages

This section introduces the messages defined for 3-D Secure. Refer to Chapter 5 for detailed information about message handling, and Annex B for message data elements.

2.4.1 Authentication Request Message (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.

2.4.2 Authentication Response Message (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.

2.4.3 Challenge Request Message (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.

2.4.4 Challenge Response Message (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.

2.4.5 Results Request Message (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.

2.4.6 Results Response Message (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.

2.4.7 Preparation Request Message (PReq)

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

2.4.8 Preparation Response Message (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 versions supported by available ACSs, and if one exists, about the corresponding 3DS Method URL. This message is not part of the 3-D Secure authentication message flow.

2.4.9 Error Message

Error messages provide additional information about an error that occurred during message processing between the 3DS Server, the DS, and the ACS.

Chapter 5, Annex A, and Annex B provide additional information about Error messages.

2.5 Authentication Flows

This section introduces the authentication flows defined for EMV 3-D Secure. Refer to Chapter 3 of this specification for authentication flow requirements.

2.5.1 Frictionless Flow

The Frictionless Flow initiates a 3-D Secure authentication flow and consists of an AReq message and an ARes message.

The Frictionless Flow does not require further Cardholder interaction to achieve a successful authentication and complete the 3-D Secure authentication process.

2.5.2 Challenge Flow

In addition to the AReq and ARes messages that comprise the Frictionless Flow, the Challenge Flow consists of CReq, CRes, RReq, and RRes messages.

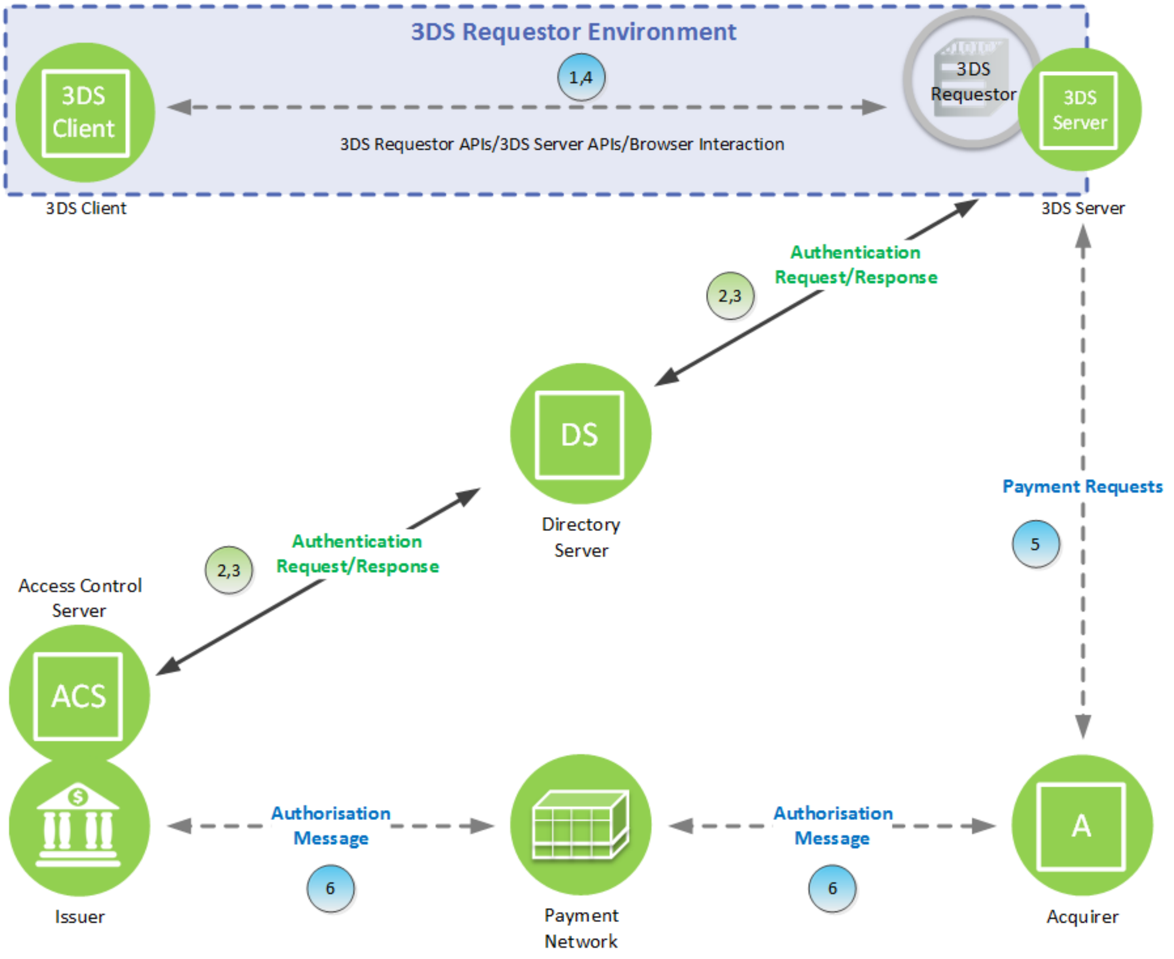
If the ACS determines that further Cardholder interaction is required to complete the authentication, the Frictionless Flow transitions into the Challenge Flow. For example, a challenge may be necessary because the transaction is deemed high-risk, is above certain thresholds, or requires a higher level of authentication due to country mandates (or regulations).

3DS Requestors decide whether to proceed with the challenge, or to terminate the 3-D Secure authentication process.

2.6 Frictionless Flow Outline

Figure 2.2 depicts the steps of the Frictionless Flow.

Figure 2.2: Frictionless Flow



Note: Dashed arrows and 3DS Requestor are not part of 3DS specification and are shown for clarity only

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 (Section 2.6.1) or Browser-based (Section 2.6.2)
   - Message Category—Payment or Non-Payment
   - 3DS Requestor 3-D Secure implementation (Section 2.1.1)

2. **3DS Server through DS 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 DS, which then forwards the message to the appropriate ACS.

3. **ACS through DS to 3DS Server**—In response to the AReq message, the ACS returns an ARes message to the DS, 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.

   As defined in Section 2.1.1, the 3DS Requestor determines how the interaction between these components is implemented. Refer to Sections 2.6.1 and 2.6.2 for additional information.

   Note: 3-D Secure processing ends here. For Payment Authorisation, the subsequent steps apply:

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

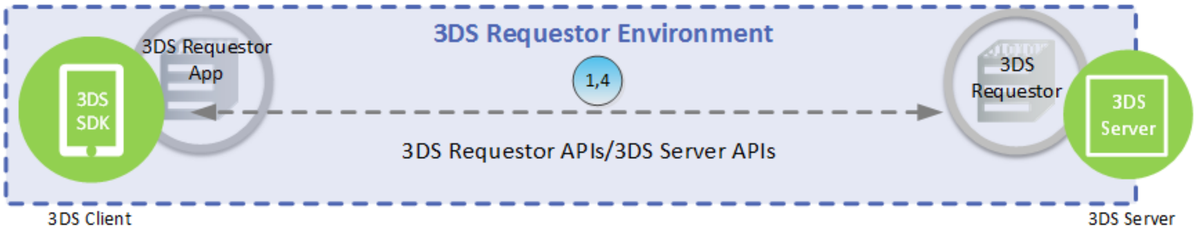
6. **Payment Authorisation**—The Acquirer can process an authorisation with the Issuer through the Payment System and return the authorisation results to the Merchant.

2.6.1 3DS Requestor Environment—App-based

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.

Figure 2.3 depicts the 3DS Requestor Environment in an App-based model.

Figure 2.3: 3DS Requestor Environment—Frictionless Flow—App-based



Functionality for Step 1 and Step 4 is defined in Section 2.6, with the following clarifications:

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

1. **3DS SDK and 3DS Server**—The 3DS SDK/3DS Requestor App communicates with the 3DS Server/3DS Requestor. Sensitive information from the Consumer Device is encrypted before being sent in the AReq message to the DS.

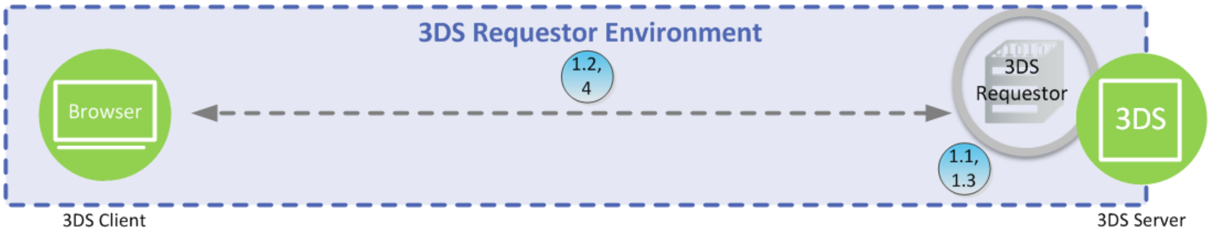
4. **3DS Server and 3DS SDK**—The 3DS Server/3DS Requestor communicates the result of the ARes message to the 3DS SDK/3DS Requestor App, which then informs the Cardholder.

2.6.2 3DS Requestor Environment—Browser-based

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

Figure 2.4 depicts the 3DS Requestor Environment.

Figure 2.4: 3DS Requestor Environment—Browser-based



Functionality for Step 1 and Step 4 is defined in Section 2.6, with the following clarifications:

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

1.1 **3DS Requestor and 3DS Server—**The 3DS Requestor communicates with the 3DS Server. The 3DS Server determines the ACS version and, if present obtains the 3DS Method URL for the requested card range and returns the information to the 3DS Requestor. The ACS version and 3DS Method URL data was previously received by the 3DS Server via a PRes message.

1.2 **3DS Method on the 3DS Requestor checkout page**—The 3DS Requestor checkout page loads the 3DS Method URL, if present, which allows the ACS to obtain additional browser information for risk-based decisioning.

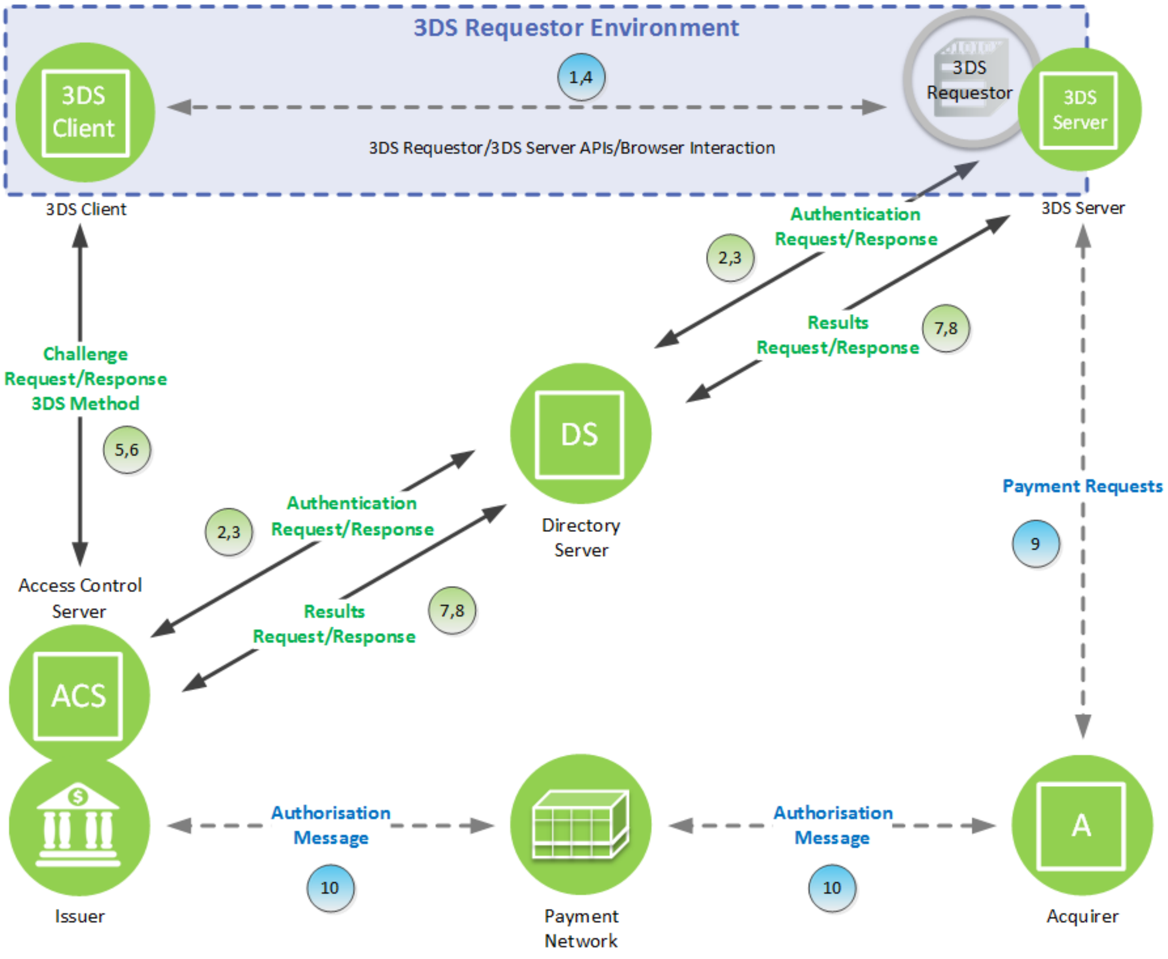
1.3 **3DS Requestor and 3DS Server**—The 3DS Requestor provides the necessary 3-D Secure information for the transaction to the 3DS Server.

4. **3DS Server and 3DS Requestor**—The 3DS Server communicates the result of the ARes message to the 3DS Requestor and completes the transaction. The 3DS Integrator determines how the interaction between these components is implemented.

2.7 Challenge Flow Outline

Figure 2.5 depicts the steps of the Challenge Flow.

Figure 2.5: Challenge Flow



Note: Dashed arrows and 3DS Requestor are not part of 3DS specification and are shown for clarity only

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 DS to ACS**—Same as the Frictionless Flow.
3. **ACS through 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.
7. **ACS through 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 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.

    Note: 3-D Secure processing ends here. For Payment Authorisation, the subsequent steps apply.
9. **Merchant** and **Acquirer—**Same as the Frictionless Flow.
10. **Payment Authorisation**—Same as the Frictionless Flow.

This page intentionally left blank.

3 EMV 3-D Secure Authentication Flow Requirements

This chapter provides the requirements for the EMV 3-D Secure processing flow.

For clarity, the actions for all components involved in the 3-D Secure authentication process are described, however, the four components covered by the *EMV 3-D Secure Protocol and Core Functions Specification* are the:

- 3DS Client
    - 3DS SDK
    - 3DS Method
- 3DS Server
- Directory Server (DS)
- Access Control Server (ACS)

The aforementioned components are required to follow the specifications as written.

For an App-based model, also refer to the *EMV 3-D Secure SDK Specification* for detailed requirements and implementation guidelines.

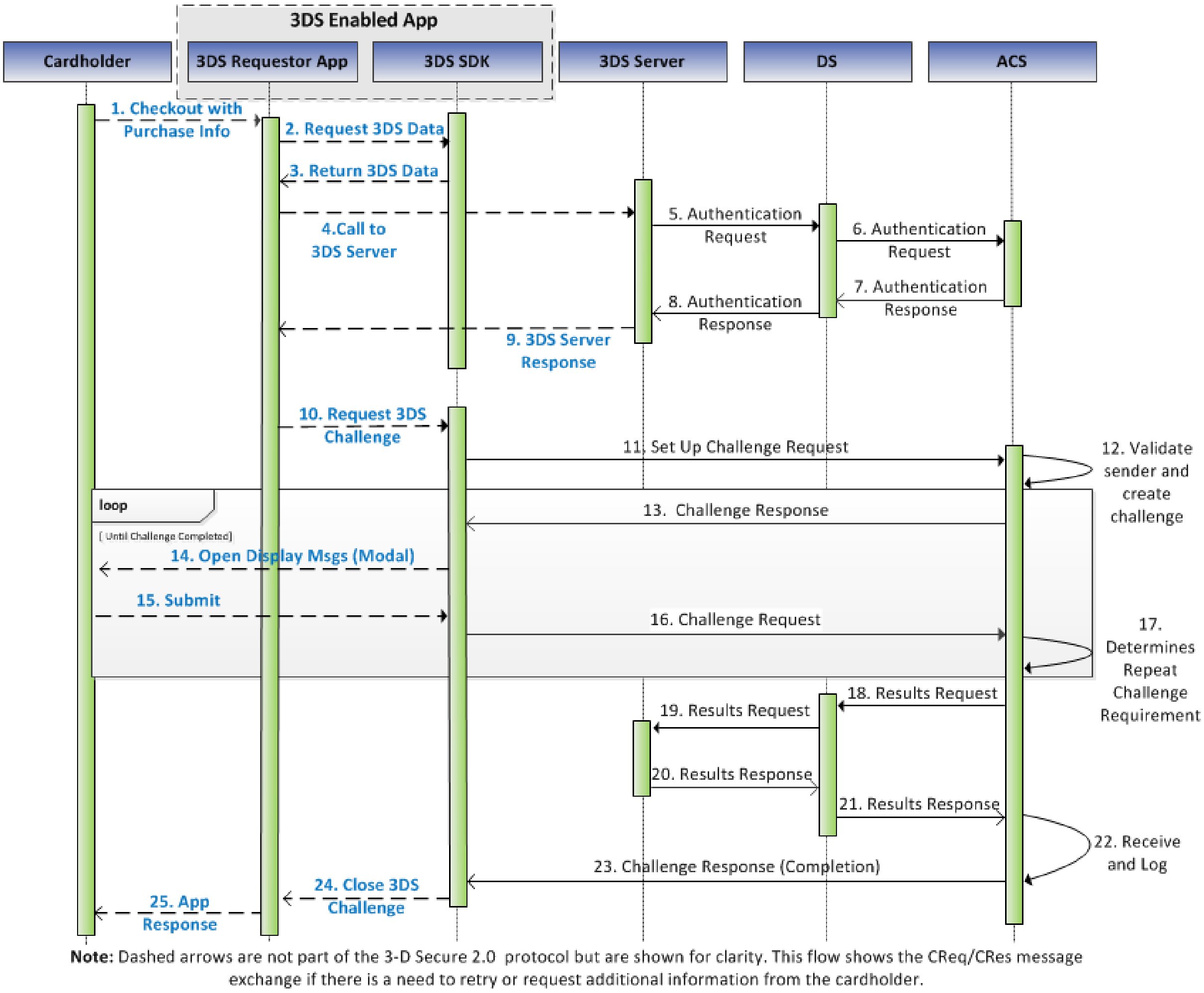
As introduced in Chapter 2, a Challenge flow is initially identical to a Frictionless Flow, thus the two flows in this section are described in one processing flow.

> **Note: The term “Validate” is used throughout this section and is a requirement for the 3-D Secure component to validate a received message. The details of the actual validation process for a given situation are defined in a specific sub-section of Section 5.9. A validation process will include a check for the presence and format of each data element based on the data elements defined in Table A.1 and the functionality outlined in Section 5.1.6, and will also include the actual error handling, should a message be in error. In addition, the validation process may include cryptographic verification and decryption of the message.**

3.1 App-based Requirements

The Steps in the Authentication Flow are outlined in Figure 3.1 with detailed requirements following the figure.

Figure 3.1: 3-D Secure Processing Flow Steps—App-based


3DS Enabled App
Cardholder
3DS Requestor App
3DS SDK
3DS Server
DS
ACS
1. Checkout with Purchase Info
2. Request 3DS Data
3. Return 3DS Data
4.Call to 3DS Server
5. Authentication Request
6. Authentication Request
7. Authentication Response
8. Authentication Response
9. 3DS Server Response
10. Request 3DS Challenge
11. Set Up Challenge Request
12. Validate sender and create challenge
loop
[Until Challenge Completed]
13. Challenge Response
14. Open Display Msgs (Modal)
15. Submit
16. Challenge Request
17. Determines Repeat Challenge Requirement
18. Results Request
19. Results Request
20. Results Response
21. Results Response
22. Receive and Log
23. Challenge Response (Completion)
24. Close 3DS Challenge
25. App Response

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

Figure 3.1 portrays a possible flow for components within the 3DS Requestor Environment and does not preclude a specific implementation. Refer to Section 2.1.1 for additional information about the 3DS Requestor Environment.

Note: Step 10 through Step 25 are applicable only for a Challenge Flow

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

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 (as outlined in Section 2.1.1), 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 DS.

The 3DS Requestor App invokes a method within the 3DS SDK to initiate 3-D Secure Cardholder authentication by obtaining the SDK Transaction ID, SDK App ID, and the Device Information.

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.

As introduced in Section 2.1.1, this specification does not require a specific component to gather the information only that the information is available for the 3DS Server when the AReq message is built. This information can include:

- Cardholder Account information
- Merchant Risk Indicator
- 3DS Requestor Authentication information
- Payment information (for Payment Authentication)
- Non-Payment information
- Cardholder information

This information together with the information gathered by the 3DS SDK (as outlined in the requirements within this step) is made available to the 3DS Server.

Note: UI requirements for this step are defined in Section 4.2.

Req. 3.1 The communication between 3DS Requestor App and Server (3DS Requestor or 3DS Server) shall be established using a server authenticated TLS session as defined in Section 6.1.1.

If the communication channel(s) within the 3DS Requestor Environment makes it impossible to have the 3DS Server receive the information within a reasonable amount of time, then this needs to be reported to the Cardholder via the 3DS Requestor App without further 3-D Secure processing.

During the execution of this Step, the 3DS SDK shall:

Req. 3.2 Obtain the Device Information, SDK Reference Number, and SDK App ID. Refer to the *EMV 3-D Secure 3DS SDK Specification* and Annex A for additional detail.

Req. 3.3 Generate the SDK Transaction ID for the 3-D Secure authentication.

This ID will, for the 3DS SDK, uniquely identify this transaction within all messages in the authentication process (AReq/ARes, CReq/CRes and RReq and RRes).

Req. 3.4 Set the Device Rendering Options Supported data element as defined in Table A.1.

Req. 3.5 Encrypt the Device Information (using the public key of the DS) as defined in Section 6.2.2.1.

Req. 3.6 Prepare the 3DS SDK to ACS secure channel as defined in Section 6.2.3.1.

Step 5: The 3DS Server

The 3DS Server shall:

Req. 3.7 Verify the authenticity of the 3DS SDK as defined in Section 6.2.1.

If the authenticity of the 3DS SDK cannot be verified by the 3DS Server, the 3DS Server **ends 3-D Secure processing**.

Req. 3.8 Generate the 3DS Server Transaction ID.

This ID will, for the 3DS Server, uniquely identify this transaction within all messages in the authentication process (AReq/ARes, CReq/CRes, and RReq/RRes).

Req. 3.9 Obtain the 3DS Requestor ID, conditionally the Acquirer BIN, and the 3DS Server Reference Number.

Req. 3.10 Ensure availability of the necessary information for the AReq message (as defined in Table B.1) gathered by components within the 3DS Requestor Environment.

If the necessary information is not available, the 3DS Server **ends 3-D Secure processing**.

Req. 3.11 Determine which DS the authentication transaction needs to be sent based on the BIN (as defined in ISO 7812) and optionally other Cardholder account information.

Req. 3.12 Establish a secure link with the DS as defined in Section 6.1.2.1.

If the connection cannot be established with the DS, the 3DS Server **ends 3-D Secure processing**.

Req. 3.13 Format the AReq message as defined in Table B.1.

Req. 3.14 Send the AReq message to the DS using the secured link established in Req. 3.12.

If the 3DS Server receives a failure when communicating with the DS or does not get a response (as defined in Section 5.5), then the 3DS Server **ends 3-D Secure processing**.

Step 6: The DS

The DS shall:

Req. 3.15 Receive the AReq message from the 3DS Server and Validate as defined in Section 5.9.1.

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

Req. 3.16 Generate the DS Transaction ID.

For the DS, the DS Transaction ID uniquely identifies the transaction in all subsequent messages in the authentication process (AReq/ARes, and RReq/RRes).

Req. 3.17 Move the SDK Encrypted Data data content (decrypted during the validation process defined in Section 5.9.1) to the Device Information data element of the AReq message for the ACS.

Req. 3.18 Check that the Message Version Number is supported by the DS.

If not, the DS returns to the 3DS Server an ARes message (as defined in Table B.2) with Transaction Status = U and IReq Code = 06, and **ends processing**.

Req. 3.19 Check the data elements in the AReq message as follows.

- If the:
  - 3DS Server Reference Number does not represent a participating 3DS Server, OR
  - SDK Reference Number does not represent a participating 3DS SDK, OR
  - Acquirer BIN does not represent a participating Acquirer, OR
  - Acquirer Merchant ID is not related to the Acquirer BIN

  Then the DS returns to the 3DS Server an ARes message (as defined in Table B.2) with Transaction Status = U and IReq Code = 50 and **ends processing**.
- If Merchant Category Code (MCC) is not valid for the specific DS, then the DS returns to the 3DS Server an ARes message (as described in Table B.2) with the Transaction Status = U and IReq Code = 59 and **ends processing**.

Req. 3.20 Determine if the Cardholder Account Number received in the AReq message is in a participating account range.

If not, the DS returns to the 3DS Server an ARes message (as defined in Table B.2 with Transaction Status set to the appropriate value as defined by the specific DS and **ends processing**

Req. 3.21 Determine if the Cardholder Account Number is in an account range that has an ACS capable of processing 3-D Secure messages.

If not, the DS returns to the 3DS Server an ARes message (as defined in Table B.2) with the Transaction Status set to the appropriate value as defined by the specific DS and **ends processing**.

Req. 3.22 Store the 3DS Server URL with the DS Transaction ID (for possible RReq processing).

Req. 3.23 Establish a secure link with the ACS as defined in Section 6.1.3.1.

If the connection cannot be established with the ACS, the DS returns to the 3DS Server an ARes message (as defined in Table B.2) with EITHER:

- Transaction Status set to the appropriate response as defined by the specific DS and **ends processing**, OR
- Transaction Status = U and IReq Code = 98 with detailed information about the issue in Invalid Request Detail and **ends processing**.

Note: 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.

Req. 3.24 Send the AReq message to the ACS using the secured link established in Req. 3.23.

If the DS does not receive an ARes message from the ACS (as defined in Section 5.5), the DS returns to the 3DS Server an ARes message (as defined in Table B.2) with EITHER:

- Transaction Status set to the appropriate response as defined by the specific DS and **ends processing**, OR
- Transaction Status = U and an IReq Code = 98 with detailed information about the issue in Invalid Request Detail and **ends processing**.

Step 7: The ACS

The ACS shall:

Req. 3.25 Receive the AReq message from the DS and Validate as defined in Section 5.9.2.

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

Req. 3.26 Check whether the Consumer Device is supported.

If not, the ACS returns to the DS an ARes message (as defined in Table B.2) with Transaction Status = U, and IReq Code = 10 and **ends processing**.

Req. 3.27 Generate the ACS Transaction ID.

Req. 3.28 Use the Cardholder Account Number from the AReq message to determine whether authentication is available or can be completed for the Cardholder.

If the authentication for the Cardholder Account Number is not available, then the ACS returns to the DS an ARes message (as defined in Table B.2) with Transaction Status, and Transaction Status Reason Code set to the appropriate response as defined by the specific DS and **ends processing**.

The ACS should:

Req. 3.29 Use the values of the 3DS Requestor Challenge Indicator received in the AReq message when evaluating the transaction disposition as defined in Req. 3.30.

The ACS shall:

Req. 3.30 Evaluate the values received in the AReq message and determine whether the transaction[1]:

- is authenticated (Transaction Status = Y)
- requires a Cardholder challenge to complete authentication (Transaction Status = C)
- is not authenticated (Transaction Status = N)

[1] The decisioning process for this action is outside the scope of this specification.

- is not authenticated, but a proof of authentication attempt (Authentication Value) was generated (Transaction Status = A)[2]
- is not authenticated, as authentication could not be performed due to technical or other problem (Transaction Status = U)
- is not authenticated because the Issuer is rejecting authentication and requesting that authorisation not be attempted (Transaction Status = R)

Req. 3.31 If a transaction is deemed authenticated (Transaction Status is = Y or A), the ACS performs the following:

- For a Payment Authentication (Message Category = 01), 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), 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 specific DS.

Req. 3.32 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 and Device Rendering Options Supported received in the AReq message. The ACS performs the following:

a. Sets the Transaction Status = C for Challenge

b. Sets the ACS Rendering Type

c. Sets up the ACS to 3DS SDK secure channel (as defined in Section 6.2.3.2)

d. Stores the SDK Transaction ID (for subsequent CReq processing)

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

Req. 3.33 Complete formatting of the ARes message as defined in Table B.2.

Req. 3.34 Send the ARes message to the DS using the secure link established in Req. 3.23.

Step 8: The DS

The DS shall:

Req. 3.35 Receive the ARes message or Error message from the ACS and Validate as defined in Section 5.9.3.

If the message is in error the DS **ends processing**.

Req. 3.36 Log transaction information as required by the DS rules.

[2] Support for Attempts is determined by each DS.

Req. 3.37 Send the ARes message to the 3DS Server as received from the ACS using the secure link established in Req. 3.12.

Step 9: The 3DS Server

The 3DS Server shall:

Req. 3.38 Receive the ARes message or Error Message from the DS and Validate as defined in Section 5.9.4.

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

Req. 3.39 For an authenticated transaction (Transaction Status = Y or A):

a. For Payment Authentication, ensure that the Transaction Status, ECI value, and Authentication Value as generated by the ACS are provided for the authorisation process.

b. Send necessary information from the ARes message to the 3DS Requestor Environment.

c. Continue with Req. 3.79.

Req. 3.40 For a transaction with a challenge (Transaction Status = C):

Evaluate based in part on the 3DS Requestor Challenge Indicator preference whether to perform the requested challenge.

- If the 3DS Requestor accepts the challenge:
  - Send necessary information from the ARes message to the 3DS Requestor Environment:
  - Continue with Step 10.
- If the 3DS Requestor continues without performing the requested challenge, further processing is outside the scope of 3-D Secure processing. The 3DS Server may continue with Req. 3.79 and **ends 3-D Secure processing**.

Req. 3.41 For a transaction not authenticated (Transaction Status = N, U, or R) or for an Error Message:

a. Send necessary information from the ARes message to the 3DS Requestor Environment.

b. Continue with Req. 3.79.

Notes: For a Frictionless Flow, the next step is Req. 3.79.

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 them 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 shall:

Req. 3.42 Check the received 3-D Secure data elements as defined in Table A.1.

Req. 3.43 Complete the ACS to 3DS SDK secure channel as defined in Section 6.2.3.3.

If the secure channel cannot be completed, the 3DS SDK reports the error to the 3DS Requestor App and **ends 3-D Secure processing**.

Req. 3.44 Establish a secure link to the ACS as defined in Section 6.1.4.1.

The link is established using the ACS URL received from the 3DS Server and verified as part of Req. 3.43.

If the secure channel cannot be completed, the 3DS SDK reports the error to the 3DS Requestor App and **ends 3-D Secure processing**.

Req. 3.45 Format the CReq message as defined in Table B.3 and protect the content as defined in Section 6.2.4.1.

Req. 3.46 Send the CReq message to the ACS using the secure link established in Req. 3.44.

Step 12: The ACS

The ACS shall:

Req. 3.47 Receive the CReq message or Error Message from the 3DS SDK and Validate as defined in Section 5.9.5.

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

Req. 3.48 Set the Interaction Counter = 0.

Req. 3.49 Set the Challenge Completion Indicator = N.

Req. 3.50 Obtain the information needed to display a Challenge on the Consumer Device per the selected challenge method and ACS UI Type. Refer to Section 4.2 for information about UI data elements.

Step 13: The ACS

The ACS shall:

Req. 3.51 Format the CRes message as defined in Table B.4.

Req. 3.52 Protect the content in the CRes message as defined in Section 6.2.4.4.

Req. 3.53 Send the CRes message to the 3DS SDK through the secure link established in Req. 3.44 for an initial interaction with the 3DS SDK, or Req. 3.56 for a continued interaction with the 3DS SDK.

Step 14: The 3DS SDK

The 3DS SDK shall:

Req. 3.54 Receive the CRes message or Error Message from the ACS and Validate as defined in Section 5.9.7.

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

Req. 3.55 Display the UI based upon the ACS UI Type selected and the data elements populated. Refer to Section 5.1.2, and refer to *EMV 3-D Secure SDK Specification* for UI details.

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 shall:

Req. 3.56 Establish a secure link to the ACS as defined in Section 6.1.4.2.

Req. 3.57 Format the CReq message as defined in Section B.3 and protect the contents as defined in Section 6.2.4.1.

Req. 3.58 Send the CReq message to the ACS using the secure link established in Req. 3.56.

Req. 3.59 If during the processing of Step 12 through Step 15 the Cardholder abandons the challenge, then the 3DS SDK sets the Challenge Cancelation Indicator to the appropriate value in the CReq message and sends the CReq message to the ACS using the secure link established in Req. 3.56.

Step 17: The ACS

The ACS shall:

Req. 3.60 Receive the CReq message or Error Message from the 3DS SDK and Validate as defined in Section 5.9.5.

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

Req. 3.61 Check the authentication data entered by the Cardholder:

- If correct, then the ACS:
  - 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, then the ACS:
    - Sets the Transaction Status = N
    - Sets the Transaction Status Reason = 19
    - Sets the ECI value as defined by the specific DS
    - Sets the Challenge Completion Indicator = Y
    - Continues with Step 18
  - Else (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 shall:

Req. 3.62 Format the RReq message as defined in Section B.8.

Req. 3.63 Establish a secure link with the DS as defined in Section 6.1.3.2.

Req. 3.64 If the Cardholder abandons the challenge during the processing of Step 16 and Step 17, or the ACS has received an abandonment CReq message from the 3DS SDK (as defined in Req. 3.61), then the ACS sets the Challenge Completion Indicator = N in the CRes message and sets the Challenge Cancelation Indicator to the appropriate value in the RReq message. Refer to Annex A for the specific values.

Req. 3.65 Send the RReq message to the DS using the secure link established in Req. 3.63.

Req. 3.66 Ensure that one RReq message is sent to the DS for each ARes message with a Transaction Status = C.

Step 19: The DS

The DS shall:

Req. 3.67 Receive the RReq message from the ACS and Validate as defined in Section 5.9.8.

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

Req. 3.68 Establish a secure link with the 3DS Server as defined in Section 6.1.2.2 using the 3DS Server URL extracted from the AReq message and stored in Req. 3.22.

Req. 3.69 Send the RReq message to the 3DS Server using the secure link established in Req. 3.68.

Step 20 The 3DS Server

The 3DS Server shall:

Req. 3.70 Receive the RReq message or Error Message from the DS and Validate as defined in Section 5.9.9.

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

Req. 3.71 Format the RRes message as defined in Table B.9 and send it to the DS using the secure link established in Req. 3.68.

Note: For Payment Authentication, the Merchant can now proceed with Authorisation 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

The DS shall:

Req. 3.72 Receive the RRes message or Error Message from the 3DS Server and Validate as defined in Section 5.9.10.

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

Req. 3.73 Log transaction information as required by the DS.

Req. 3.74 Send the RRes message to the ACS as received from the 3DS Server using the secure link established in Req. 3.63.

Step 22 The ACS

The ACS shall:

Req. 3.75 Receive the RRes message or Error Message from the DS and Validate as defined in Section 5.9.11.

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

Step 23 The ACS

The ACS shall (as a continuation of receiving the CReq message in Step 17):

Req. 3.76 Format the final CRes message (as defined in Table B.5) and protect the contents as defined in Section 6.2.4.4.

Req. 3.77 Send the final CRes message to the 3DS SDK using the secure link established in Req. 3.56.

Step 24 The 3DS Requestor Environment

The 3DS SDK shall:

Req. 3.78 Receive the final CRes message or Error Message from the ACS and Validate as defined in Section 5.9.7.

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

Req. 3.79 Convey the appropriate response to the 3DS Requestor App.

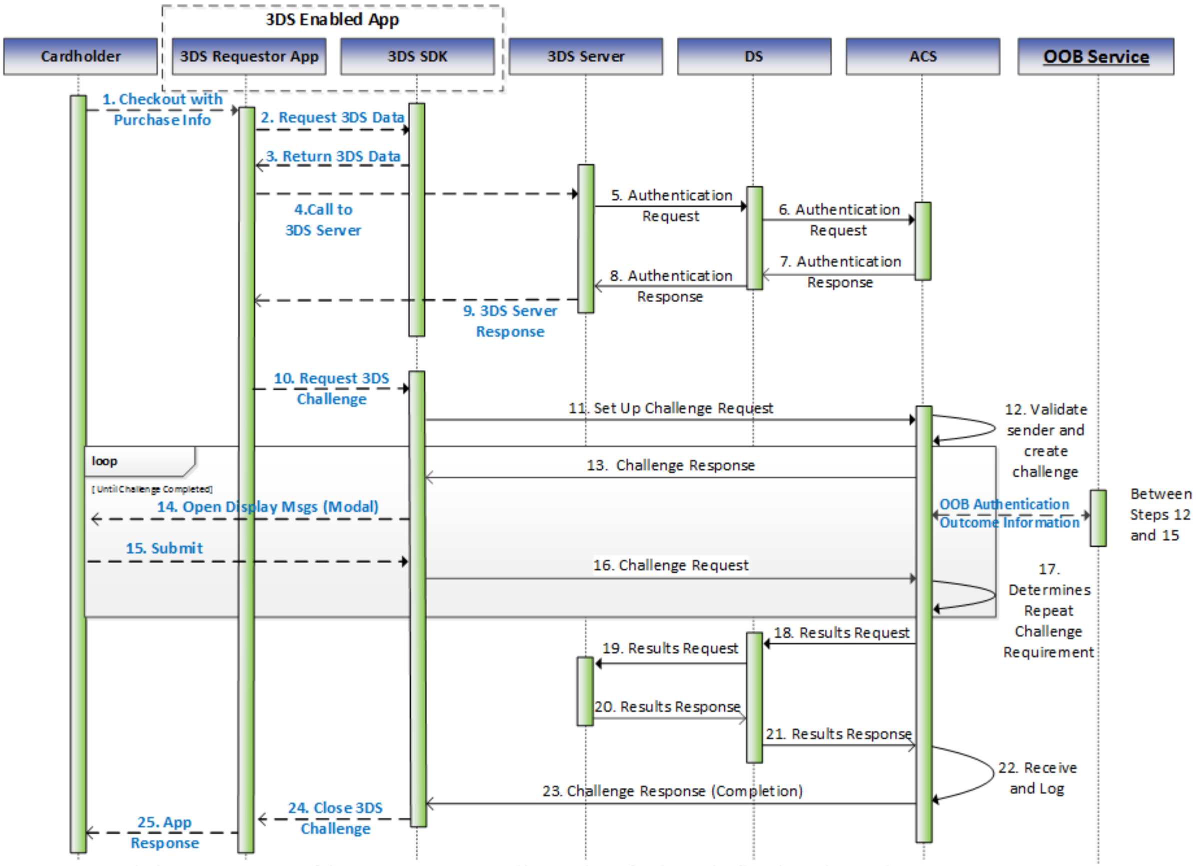
Note: 3-D Secure processing completes.

Step 25 The 3DS Requestor App

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

3.2 Challenge Flow with OOB Authentication Requirements

Figure 3.2: Out-of-Band Processing Flow


3DS Enabled App
Cardholder
3DS Requestor App
3DS SDK
3DS Server
DS
ACS
OOB Service
1. Checkout with Purchase Info
2. Request 3DS Data
3. Return 3DS Data
4.Call to 3DS Server
5. Authentication Request
6. Authentication Request
7. Authentication Response
8. Authentication Response
9. 3DS Server Response
10. Request 3DS Challenge
11. Set Up Challenge Request
12. Validate sender and create challenge
loop
[Until Challenge Completed]
13. Challenge Response
14. Open Display Msgs (Modal)
OOB Authentication Outcome Information
Between Steps 12 and 15
15. Submit
16. Challenge Request
17. Determines Repeat Challenge Requirement
18. Results Request
19. Results Request
20. Results Response
21. Results Response
22. Receive and Log
23. Challenge Response (Completion)
24. Close 3DS Challenge
25. App Response

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

An Out-of-Band (OOB) Challenge Flow is identical to a standard 3-D Secure Processing Flow as defined in Section 3.1 with the following exceptions:

Step 6: The ACS recognises that an OOB interaction with the Cardholder is required.

Step 12 and Step 15: Between Step 12 and Step 15, the ACS initiates an OOB interaction with the Cardholder rather than interacting with the Cardholder via the 3DS SDK. During the OOB authentication, the Cardholder authenticates to the ACS or a service provider/Issuer interacting with the ACS.

The method used for the OOB communication and the authentication method itself is outside the scope of this specification. An example of an OOB communication could be a push notification to a banking app that completes authentication, then sends the results to the ACS.

Step 13: The challenge information in the CRes message consists only of Cardholder instructions on how to perform the OOB authentication.

Step 16: The ACS receives only an acknowledgement that the Cardholder may have performed the OOB authentication.

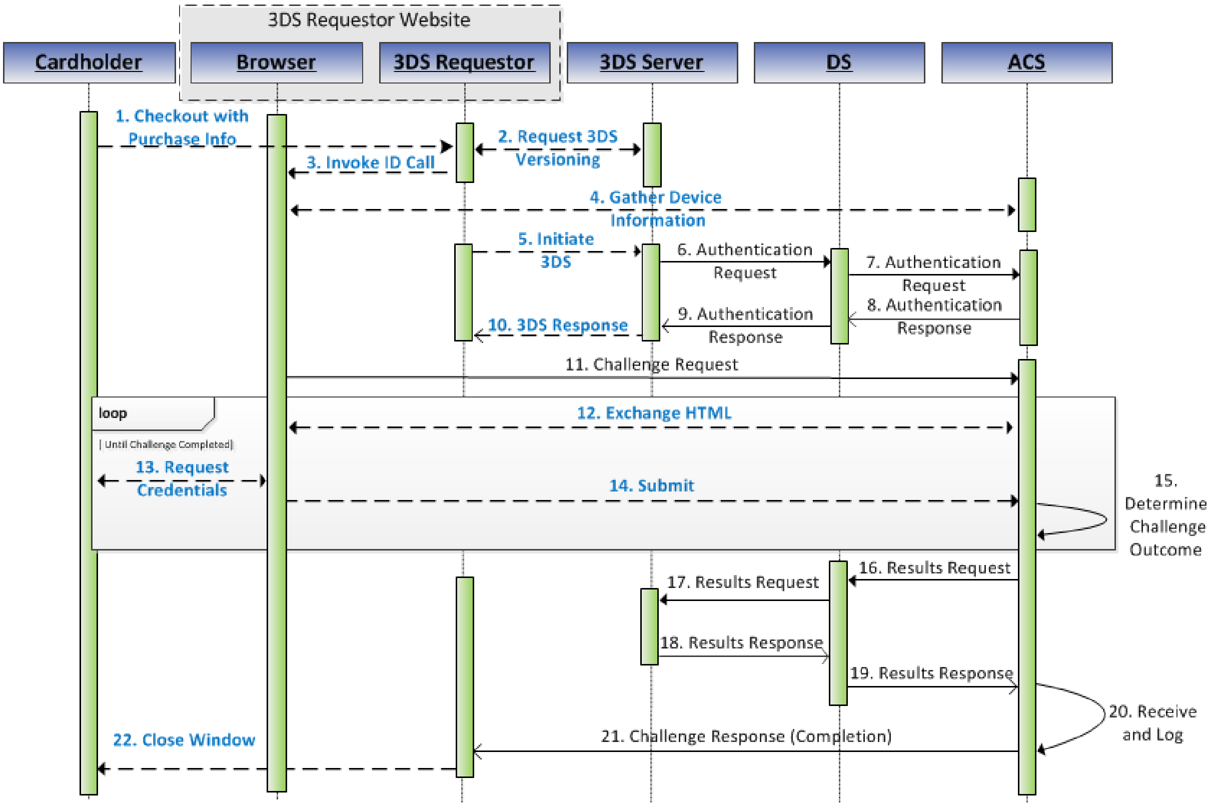
Step 17: In Req. 3.61 of the Challenge Flow, the ACS gathers the information on whether the authentication was successful from the OOB interaction with the Cardholder rather than from the CReq message.

How an authentication decision is made for an OOB authentication is outside the scope of this specification, however the ACS needs to have access to the result of the OOB authentication before Step 18.

3.3 Browser-based Requirements

The Steps in the authentication flow are outlined in Figure 3.3 with detailed requirements following the figure.

Figure 3.3: 3-D Secure Processing Flow Steps—Browser-based



Note: Dashed arrows are not part of the 3-D Secure 2.0 protocol but are shown for clarity.

Figure 3.3 portrays a possible flow for components within the 3DS Requestor Environment and does not preclude a specific implementation. Refer to Section 2.1.1 for additional information about the 3DS Requestor Environment.

Note: Step 10 through Step 21 are applicable only for the Challenge Flow.

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.

Depending on the 3DS Requestor Environment (as outlined in Section 2.6.2) additional information may be obtained.

The 3DS Requestor uses the Cardholder Account Number and optionally other cardholder information to request the ACS Version Number and if present, the 3DS Method URL, for that BIN range from the 3DS Server.

The 3DS Server shall:

Req. 3.80 Retrieve the ACS Version Number, and, if present, the 3DS Method URL (stored from a previously received PRes message) for that BIN range.

Req. 3.81 Generate the 3DS Server Transaction ID.

Req. 3.82 Pass the 3DS Server Transaction ID, ACS Version Number and, if present, the 3DS Method URL back through the 3DS Requestor Environment to the 3DS Requestor.

Step 3 The 3DS Requestor Environment

The 3DS Server shall:

Req. 3.83 Ensure for each transaction, 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.

Req. 3.84 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

If a 3DS Method URL was present in the response from the 3DS Server in Step 2, the Browser will connect via the 3DS Method to the ACS or an entity designated by the ACS to gather browser and Device Information.

The manner in which the 3DS Method obtains Device Information and which information is gathered is outside the scope of this specification, however it is necessary to use the 3DS Server Transaction ID to identify the Browser/Device Information for a later match at the ACS. Refer to Chapter 5 for additional information about message handling.

The manner in which the Device Information is retrieved or used by the ACS or an entity designated by the ACS is outside the scope of this specification, however it is a requirement, that the browser connects to the ACS using a secure link (as defined in Section 6.1.8).

The ACS shall:

Req. 3.85 Ensure that the communication between the Browser and the ACS is established using a server authenticated TLS session (as defined in Section 6.1.8).

If the communication is not established as required, the ACS **ends processing**.

Note: The 3DS Method requirements are defined in Section 5.8.1.

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.

As introduced in Section 2.1.1, this specification does not require a specific component to gather the information only that the information is available for the 3DS Server when the AReq message is built. This information can include:

- Cardholder Account Information
- Merchant Risk Indicator
- 3DS Requestor Authentication Information
- Payment Information
- Non-Payment Information
- Cardholder information

This information is made available to the 3DS Server.

The 3DS Server shall:

Req. 3.86 Ensure that the communication between client (Browser) and server (3DS Requestor) has been established using a server authenticated TLS session as defined in Section 6.1.1.

- If the communication is not established as required, the 3DS Server **ends 3-D Secure processing**.
- If the communication channel(s) within the 3DS Requestor Environment makes it impossible to have the 3DS Server receive the information within a reasonable amount of time, then this needs to be reported to the Cardholder via the browser and **ends 3-D Secure processing**.

Step 6 The 3DS Server

The 3DS Server shall:

Req. 3.87 Obtain the 3Ds Requestor ID, conditionally the Acquirer BIN, and the 3DS Server Reference Number.

Req. 3.88 Ensure availability of the necessary information for the AReq message (as defined in Table B.1) gathered by components within the 3DS Requestor Environment.

If the necessary information is not available, the 3DS Server **ends 3-D Secure processing**.

Req. 3.89 Determine which DS the authentication transaction needs to be sent based on the BIN (as defined in ISO 7812) and optionally other Cardholder account information.

Req. 3.90 Establish a secure link with the DS as defined in Section 6.1.2.1.

If the connection cannot be established with the DS, the 3DS Server **ends 3-D Secure processing**.

Req. 3.91 Format the AReq message as defined in Section B.1.

Req. 3.92 Send the AReq message to the DS using the secured link established in Req. 3.90.

If the 3DS Server receives a failure when communicating with the DS or does not get a response (as defined in Section 5.5), then the 3DS Server **ends 3-D Secure processing**.

Step 7 The DS

The DS shall:

Req. 3.93 Receive the AReq message from the 3DS Server and Validate as defined in 5.9.1.

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

Req. 3.94 Generate the DS Transaction ID.

Req. 3.95 Check that the Message Version Number is supported by the DS.

If not, the DS returns to the 3DS Server an ARes message (as defined in Table B.2) with Transaction Status = U and IReq Code = 06 and **ends processing**.

Req. 3.96 Check further the data elements in the AReq message as follows:

- If the:
  - 3DS Server Reference Number does not represent a participating 3DS Server, OR
  - Acquirer BIN does not represent a participating Acquirer, OR
  - Acquirer Merchant ID is not related to the Acquirer BIN

  Then the DS returns to the 3DS Server an ARes message (as described in Table B.2) with the Transaction Status = U and IReq Code = 50 then **ends processing**.
- If Merchant Category Code (MCC) is not valid for the specific DS, then the DS returns to the 3DS Server an ARes message (as described in Table B.2) with the Transaction Status = U and IReq Code = 59 then **ends processing**.

Req. 3.97 Determine if the Cardholder Account Number received in the AReq message is in a participating account range.

If not, the DS returns to the 3DS Server an ARes message (as described in Table B.2) with the Transaction Status set to the appropriate value as defined by the specific DS and **ends processing**.

Req. 3.98 Determine if the Cardholder Account Number is in an account range that has an ACS capable of processing 3-D Secure messages.

If not, the DS returns to the 3DS Server an ARes message (as described in Table B.2) with the Transaction Status set to the appropriate value as defined by the specific DS and **ends processing**.

Req. 3.99 Store the 3DS Server URL with the DS Transaction ID (for possible RReq processing).

Req. 3.100 Establish a secure link with the ACS as defined in Section 6.1.3.1.

If the connection cannot be established with the ACS, the DS returns to the 3DS Server an ARes with Transaction Status = U and IReq Code = 98 and **ends processing**.

Note: 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.

The DS shall:

Req. 3.101 Send the AReq to the ACS using the secured link established in Req. 3.100.

If the DS does not receive an ARes message from the ACS (as defined in Section 5.5), the DS returns to the 3DS Server an Ares message (as defined in Table B.2) EITHER:

- Transaction Status set to the appropriate response as defined by the specific DS and ends processing, OR
- Transaction Status = U and an IReq Code = 98 with detailed information about the issue in Invalid Request Detail and **ends processing**.

Step 8 The ACS

The ACS shall:

Req. 3.102 Receive the AReq message from the DS and Validate as defined in Section 5.9.2.

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

Req. 3.103 Check whether the Consumer Device on which the authentication is being requested is supported.

Req. 3.104 Generate the ACS Transaction ID.

Req. 3.105 Use the Cardholder Account Number from the AReq message to determine whether authentication is available for the Cardholder.

If the authentication for the Cardholder Account Number is not available, the ACS returns to the 3DS Server an ARes message (as defined in Table B.2) with the Transaction Status, and the Transaction Status Reason Code set to the appropriate response as defined by the specific DS and **ends processing**.

The ACS should:

Req. 3.106 Use the values of the 3DS Requestor Challenge Indicator received in the AReq message when evaluating the transaction disposition as defined in Req. 3.107.

The ACS shall:

Req. 3.107 Evaluate the values received in the AReq message and determine whether the transaction[3]:

- is authenticated (Transaction Status = Y)
- requires a Cardholder challenge to complete authentication (Transaction Status = C)
- is not authenticated (Transaction Status = N)

[3] The decisioning process for this action is outside the scope of this specification.

- is not authenticated, but a proof of authentication attempt (Authentication Value) was generated (Transaction Status = A)[4]
- is not authenticated because authentication could not be performed due to a technical or other problem (Transaction Status = U)
- is not authenticated because the Issuer is rejecting authentication and requesting that authorisation not be attempted (Transaction Status = R)

Req. 3.108 If a transaction is deemed authenticated (Transaction Status is = Y or A), the ACS performs the following:

- For a Payment Authentication (Message Category = 01), 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), the ACS *may*:
  - *Generate the ECI value and Authentication Value and include in the ARes message as defined by the specific Payment System.*
  - *Assign an appropriate Transaction Status Reason 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 specific DS.

Req. 3.109 If a challenge is deemed necessary, (Transaction Status = C) the ACS determines whether an acceptable challenge method is supported by the 3DS Server based on the Device Channel data element received in the AReq message. The ACS performs the following:

a. Sets the Transaction Status = C for Challenge.

b. Sets the ACS URL field in the ARes that will be utilised in the Browser to ACS link as defined in Req. 3.100.

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

Req. 3.110 Complete formatting of the ARes message as defined in Table B.2.

Req. 3.111 Send the ARes message to the DS using the secure link established in Req. 3.100.

Step 9 The DS

The DS shall:

Req. 3.112 Receive the ARes message or Error message from the ACS and Validate as defined in Section 5.9.3.

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

Req. 3.113 Log transaction information as required by the DS rules.

Req. 3.114 Send the ARes message to the 3DS Server as received from the ACS using the secure link established in Req. 3.100.

[4] Support for attempts is determined by each DS.

Step 10 The 3DS Server

The 3DS Server shall:

Req. 3.115 Receive the ARes message or Error Message from the DS and Validate as defined in Section 5.9.4.

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

Req. 3.116 For an authenticated transaction (Transaction Status = Y or A):

a. For a Payment Authentication, ensure that the Transaction Status, ECI value, and Authentication Value as generated by the ACS are provided for the authorisation process.

b. Send necessary information from the ARes message to the 3DS Requestor Environment.

c. Continue with Step 22.

Req. 3.117 For a transaction with a challenge (Transaction Status = C):

a. Evaluate based in part on the 3DS Requestor Challenge Indicator preference whether to perform the requested challenge.

- If the 3DS Requestor accepts the challenge:
  - Validate the signature of the ACS Signed Content data element (as defined in 6.2.3.3) and send necessary information from the ARes message to the 3DS Requestor Environment.
  - Continue with step b through e of this requirement and then Step 11.
- If the 3DS Requestor continues without performing the requested challenge, further processing is outside the scope of 3-D Secure processing. The 3DS Server may continue with Step 22.

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

c. Base64-encode the CReq message.

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

e. 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 as defined in Section 6.1.4.2.

Req. 3.118 For a transaction not authenticated (Transaction Status = N, U, or R):

a. Send necessary information from the ARes message to the 3DS Requestor Environment.

b. Continue with Step 22.

Note: Req. 3.117.d specifies posting the CReq message from the 3DS Server through the cardholder browser to the ACS. This flow is only partially depicted in Step 10 of Figure 3.3.

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

The ACS shall:

Req. 3.119 Receive the CReq message from the Browser and Validate as defined in 5.9.6.

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

Req. 3.120 Prepare the authentication User Interface (ACS UI) to the Cardholder browser.

Req. 3.121 Set the Interaction Counter = 0.

Step 12 The ACS and Browser

The ACS shall:

Req. 3.122 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

The ACS shall:

Req. 3.123 Check the authentication data entered by the Cardholder:

- If correct, then the ACS:
  - 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
  - Else (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.

- Prepare the authentication User Interface (ACS UI) to the Cardholder Browser which may contain HTML, JavaScript, etc.
- Continues with Step 12

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

Step 16 The ACS

The ACS shall:

Req. 3.124 Format the RReq message as defined in Table B.8.

Req. 3.125 Establish a secure link with the DS as defined in Section 6.1.3.2.

Req. 3.126 If the Cardholder abandons the challenge during the processing of Step 12 through Step 14, then the ACS sets the Challenge Completion Indicator =N in the RReq message and sets the Challenge Cancelation Indicator to the appropriate value in the RReq message. Refer to Annex A for the specific values.

Req. 3.127 Send the RReq message to the DS.

Req. 3.128 Send one RReq message to the DS for each ARes message with an ARes Transaction Status = C.

Step 17 The DS

The DS shall:

Req. 3.129 Receive the RReq message from the ACS and Validate as defined in Section 5.9.8.

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

Req. 3.130 Establish a secure link with the 3DS Server as defined in Section 6.1.2.2 using the 3DS Server URL data element extracted from the AReq message and stored in Req. 3.99.

Req. 3.131 Send the RReq message to the 3DS Server using the secure link established in Req. 3.130.

Step 18 The 3DS Server

The 3DS Server shall:

Req. 3.132 Receive the RReq message or Error Message from the DS and Validate as defined in Section 5.9.9.

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

Req. 3.133 Format the RRes message as defined in Table B.9 and send it to the DS using the secure link established in Req. 3.130.

Note: For Payment Authentication, the Merchant can now proceed with Authorisation 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

The DS shall:

Req. 3.134 Receive the RRes message or Error Message from the 3DS Server and Validate as defined in Section 5.9.10.

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

Req. 3.135 Log transaction information as required by the DS rules.

Req. 3.136 Send the RRes message to the ACS through the secure link established in Req. 3.125.

Step 20 The ACS

The ACS shall:

Req. 3.137 Receive the RRes message or Error Message from the DS and Validate as defined in Section 5.9.12.

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

Step 21 The ACS

The ACS shall (as a continuation of receiving the CReq message in Step 11):

Req. 3.138 Format the final CRes message as defined in Table B.5.

Req. 3.139 Base64-encode the final CRes message.

Req. 3.140 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 CReq 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 Environment:

- For a transaction with a Challenge, receives the CRes message at the Notification URL as defined in B.4.
- Conveys the appropriate response to appropriate 3DS Requestor Environment components and closes the authentication window.

Note: 3-D Secure processing completes.

This page intentionally left blank.

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

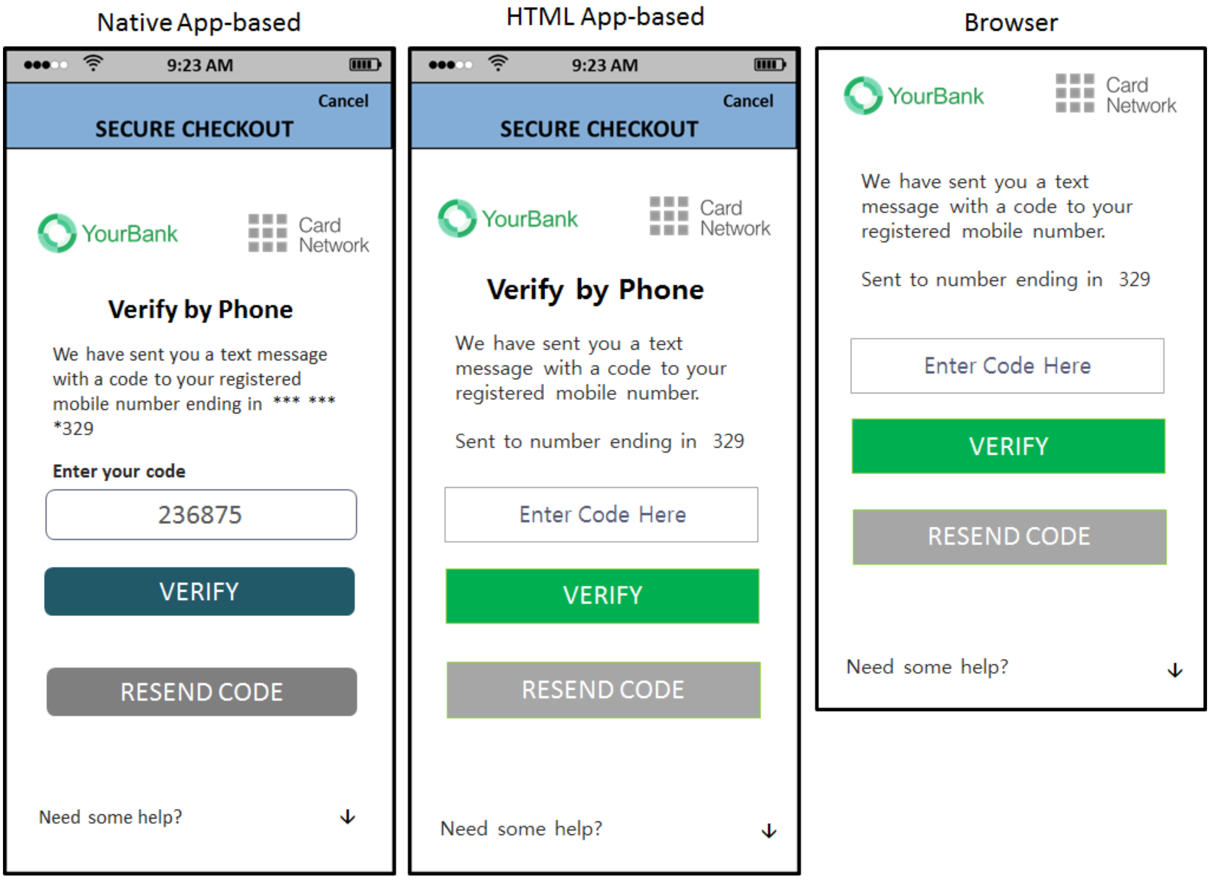
This chapter provides requirements, Template examples, and guidelines for building the User Interface (UI) to support 3-D Secure authentication for both App-based and Browser-based implementations.

4.1 3-D Secure User Interface Templates

3-D Secure UI Templates provide a consistent user experience whether App-based (Native or HTML) or Browser-based. Issuers will work with their ACS to determine their specific UI content, and the UI format is driven by the Templates.

Figure 4.1 illustrates the consistency of the look and feel across device channels and implementations.

Figure 4.1: UI Template Examples—All Device Channels



We have sent you a text message with a code to your registered mobile number ending in *** *** *329

236875

We have sent you a text message with a code to your registered mobile number.

Sent to number ending in 329

We have sent you a text message with a code to your registered mobile number.

Sent to number ending in 329

4.2 App-based User Interface Overview

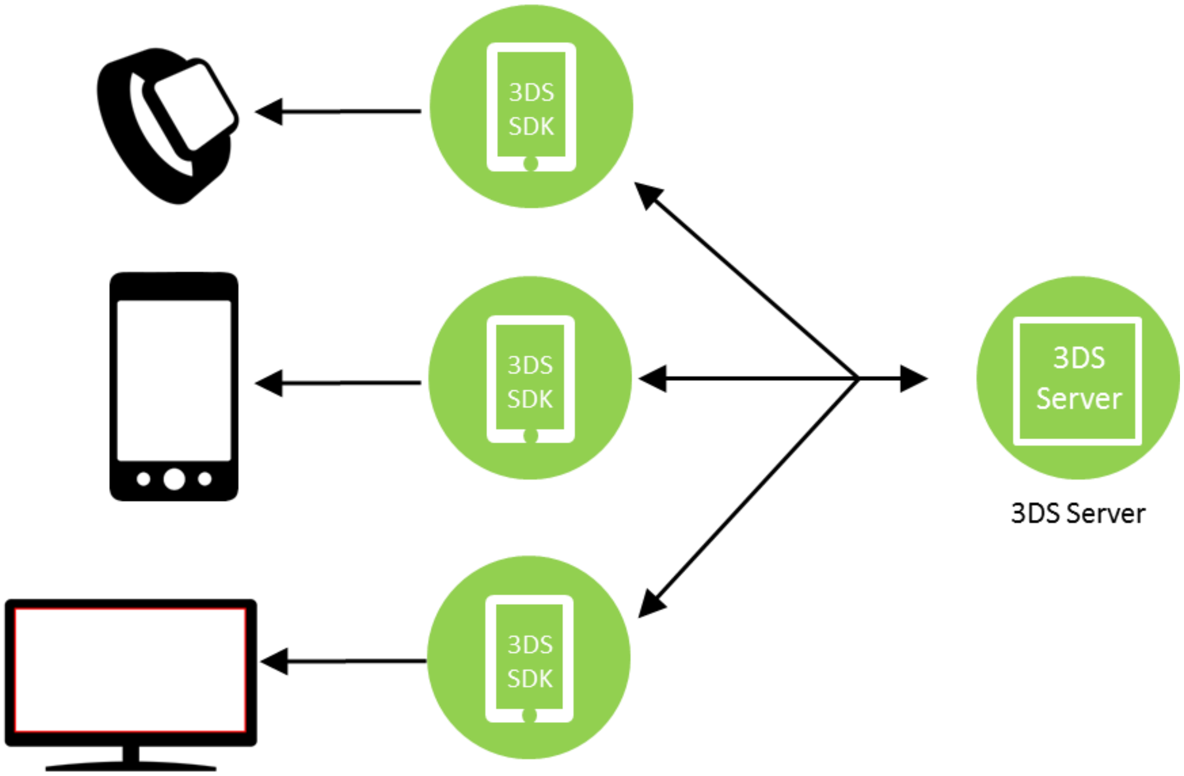
In an App-based implementation, the 3DS Requestor App and the 3DS SDK control the rendering of the UI. Each SDK is responsible for creating the UI elements that are specific to that particular environment, for example, the operating systems (OS) on a Consumer Device. The Consumer Device determines the UI format. Either:

- Native
- HTML
- Both Native and HTML

Note: The UI format must remain consistent during the CReq/CRes message exchange. For example, if starting with a Native format, then every exchange must remain with a Native format.

Figure 4.2 depicts three App-based interface scenarios and illustrates a unique SDK for each.

Figure 4.2: 3DS SDK Options

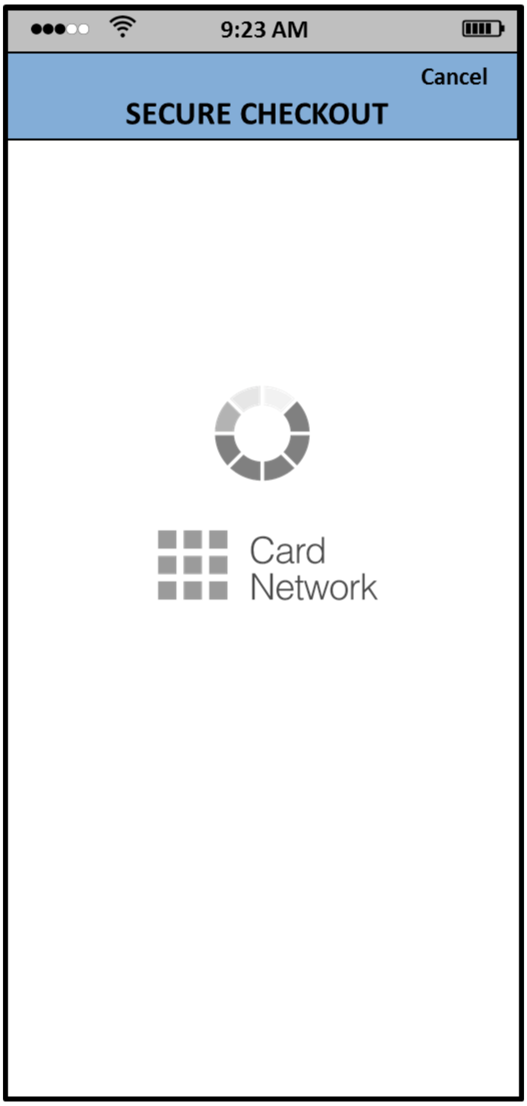


4.2.1 Processing Screen Requirements

During message exchanges, a screen should be displayed to the Cardholder to indicate that 3-D Secure processing is occurring. The Processing screen is applicable to both Native and HTML implementations.

Figure 4.3 provides a sample format for the App-based Processing screen that contains both the Processing Graphic and the Logo.

Figure 4.3: Sample App-based Processing Screen



4.2.1.1 3DS SDK/3DS Requestor App

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

Req. 4.1 Create a Processing screen for display during AReq/ARes message cycle.

Req. 4.2 Not include any other design element in the Processing screen.

Req. 4.3 If requested, integrate the DS logo into the Processing screen.

Req. 4.4 Store the DS logo.

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

Req. 4.5 Display the Processing screen supplied by the 3DS SDK during the entire AReq/ARes message cycle.

Req. 4.6 Display the Processing screen for a minimum of two seconds.

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

Req. 4.7 Create the default Processing Graphic (for example, a progress bar or a spinning wheel) of the Consumer Device OS (refer to Figure 4.3 for an example).

Req. 4.8 Not include any other design element in the Processing screen.

Req. 4.9 If requested, integrate the DS Logo into the Processing screen.

Req. 4.10 Store the DS logo.

Req. 4.11 Display the Processing screen during the entire CReq/CRes message cycle.

Req. 4.12 Display the Processing screen for a minimum of two seconds.

4.2.2 Native UI Templates

The Native UI integrates into the 3DS Requestor App UI to facilitate a consistent user experience. The Native UI has a similar look and feel as the 3DS Requestor's App with the authentication content provided by the Issuer.

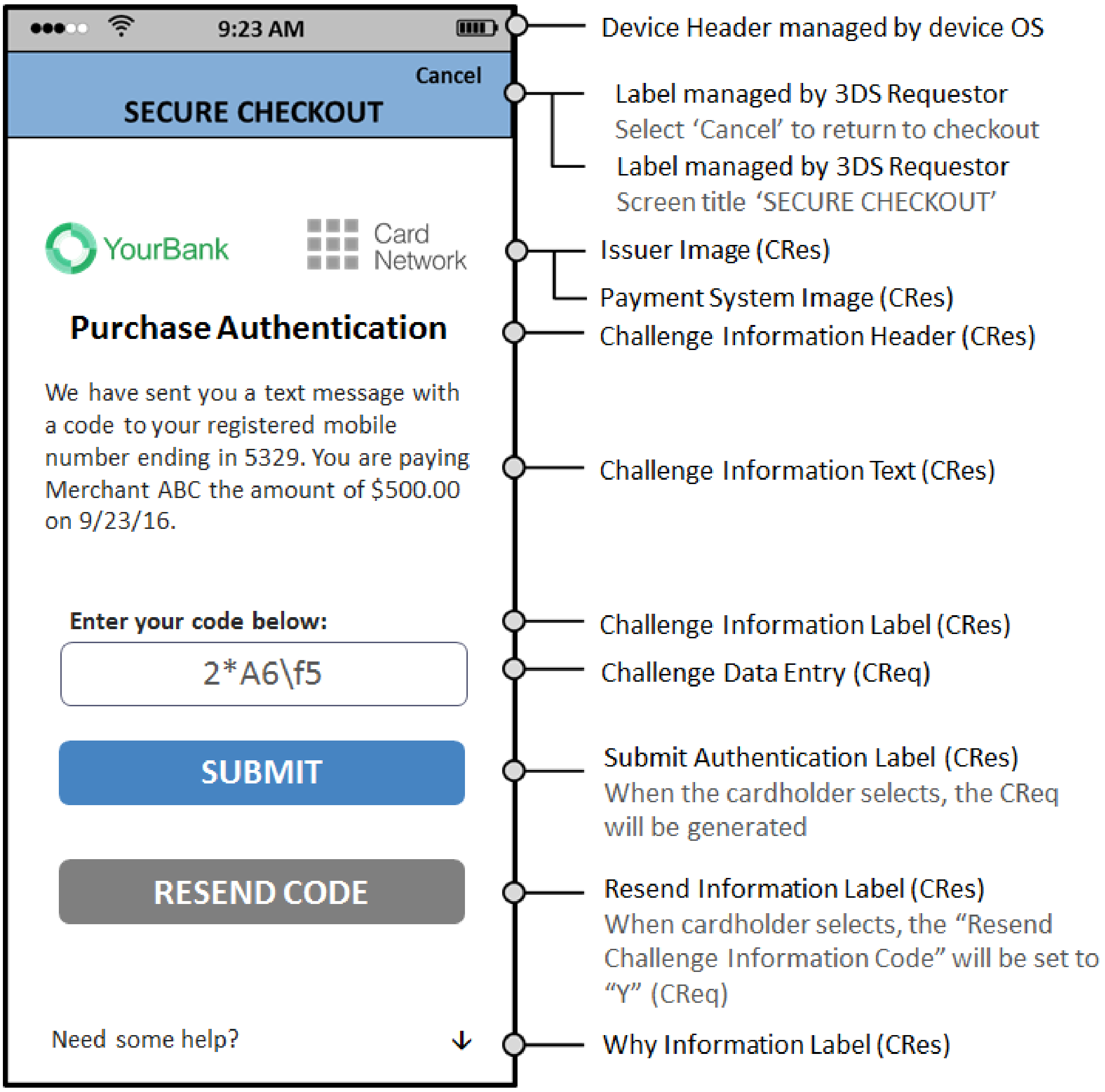
This format also allows for Issuer and Payment System branding. Both the 3DS Requestor App and the 3DS SDK control the rendering of the UI such that the authentication pages inherits the 3DS Requestor's UI design elements. Details of the UI rendering process through an SDK are separately described in the *EMV 3-D Secure SDK—Implementation Guide*.

Figures 4.3–4.8 depict sample Native UI Templates. The UI content is provided by the ACS in the CRes message which contains the information needed to properly display the UI.

UI elements exceptions are depicted in the figures as "Label Managed by the 3DS Requestor" and the functionality of these items are managed by the 3DS SDK.

Figure 4.4 provides a sample format for a one-time passcode (OTP) during a Payment Authentication transaction. This sample UI provides a format using expandable fields for additional information.

Figure 4.4: Sample Native UI OTP Template—Payment Authentication



Purchase Authentication

We have sent you a text message with a code to your registered mobile number ending in 5329. You are paying Merchant ABC the amount of $500.00 on 9/23/16.

Device Header managed by device OS

Label managed by 3DS Requestor
Select 'Cancel' to return to checkout

Label managed by 3DS Requestor
Screen title 'SECURE CHECKOUT'

Issuer Image (CRes)

Payment System Image (CRes)

Challenge Information Header (CRes)

Challenge Information Text (CRes)

Challenge Information Label (CRes)

Challenge Data Entry (CReq)

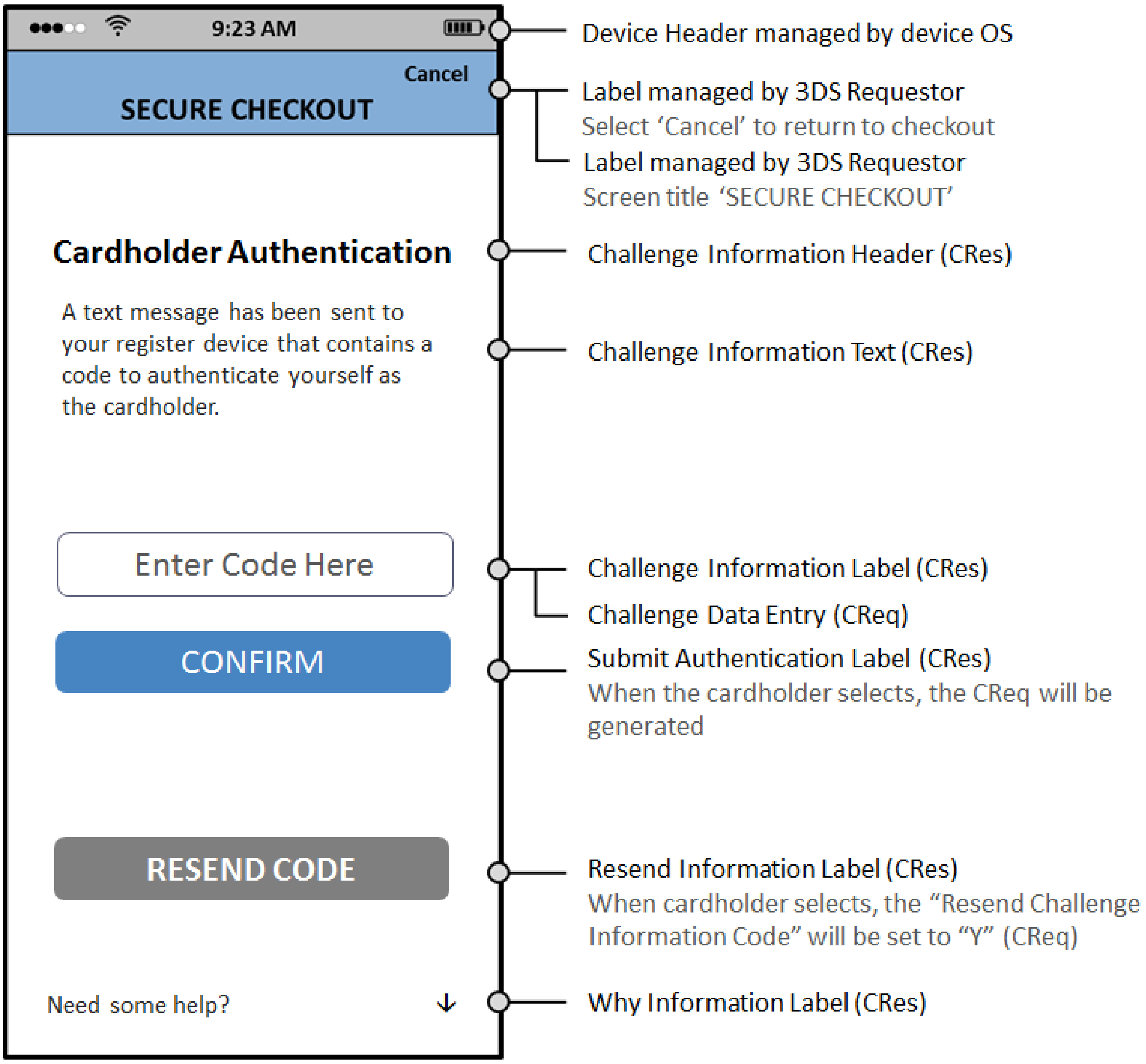
Submit Authentication Label (CRes)
When the cardholder selects, the CReq will be generated

Resend Information Label (CRes)
When cardholder selects, the "Resend Challenge Information Code" will be set to "Y" (CReq)

Why Information Label (CRes)

Figure 4.5 provides a sample format for a one-time passcode (OTP) during a Non-Payment Authentication transaction.

Figure 4.5: Sample Native UI OTP Template—Non-Payment Authentication



A text message has been sent to your register device that contains a code to authenticate yourself as the cardholder.

Device Header managed by device OS

Label managed by 3DS Requestor
Select 'Cancel' to return to checkout

Label managed by 3DS Requestor
Screen title 'SECURE CHECKOUT'

Challenge Information Header (CRes)

Challenge Information Text (CRes)

Challenge Information Label (CRes)

Challenge Data Entry (CReq)

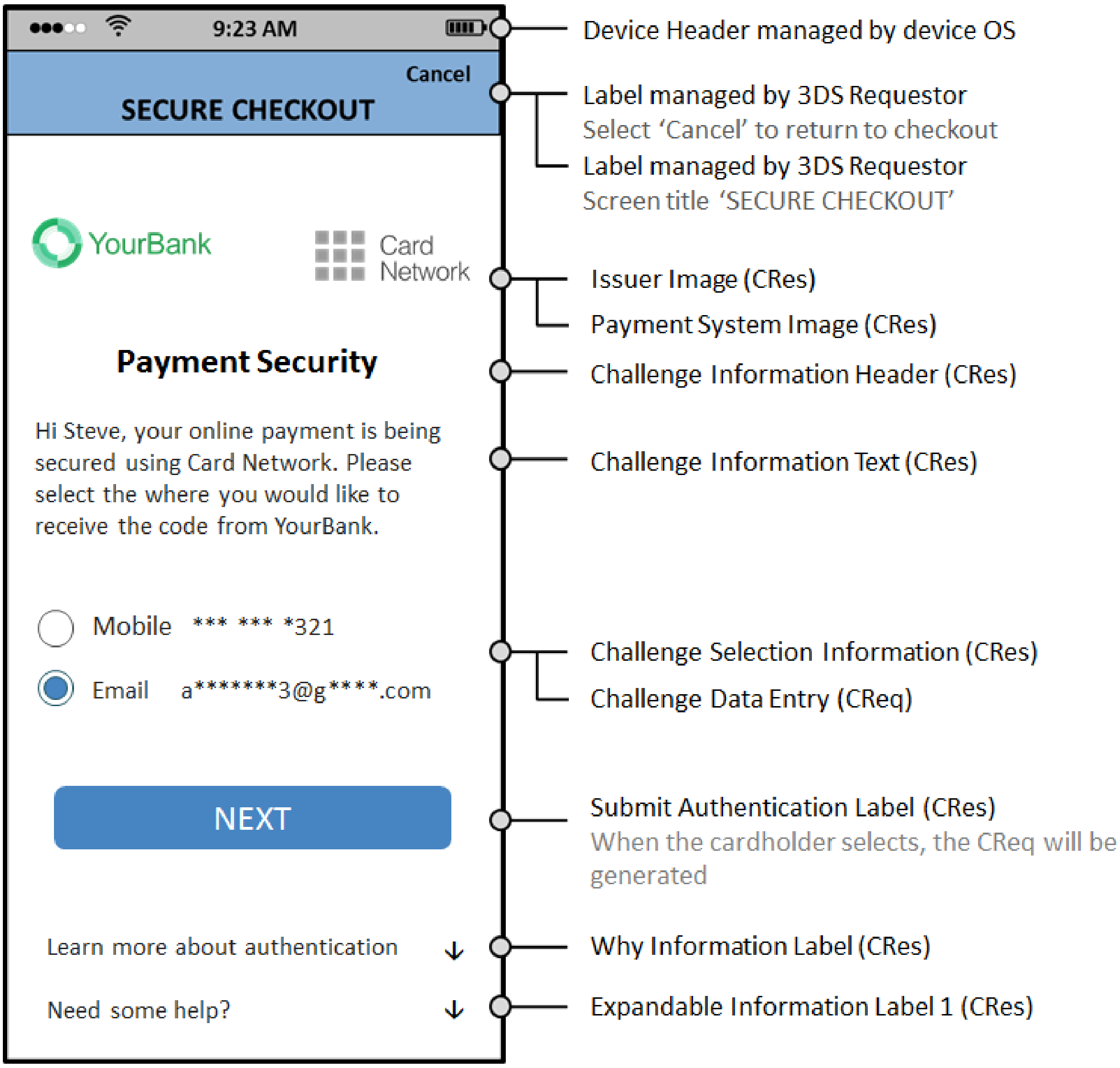
Submit Authentication Label (CRes)
When the cardholder selects, the CReq will be generated

Resend Information Label (CRes)
When cardholder selects, the "Resend Challenge Information Code" will be set to "Y" (CReq)

Why Information Label (CRes)

Figure 4.6 provides a sample format that allows multiple options to be presented to the Cardholder to obtain single response. For example, asking the Cardholder if they prefer the OTP to be sent to the Consumer Device or the email address on file.

Figure 4.6: Sample Native UI—Single-select Information—Payment Authentication



9:23 AM

Cancel

SECURE CHECKOUT

YourBank

Card Network

Payment Security

Hi Steve, your online payment is being secured using Card Network. Please select the where you would like to receive the code from YourBank.

○ Mobile *** *** *321

● Email a*******3@g****.com

NEXT

Learn more about authentication ↓

Need some help? ↓

Device Header managed by device OS

Label managed by 3DS Requestor
Select 'Cancel' to return to checkout

Label managed by 3DS Requestor
Screen title 'SECURE CHECKOUT'

Issuer Image (CRes)

Payment System Image (CRes)

Challenge Information Header (CRes)

Challenge Information Text (CRes)

Challenge Selection Information (CRes)

Challenge Data Entry (CReq)

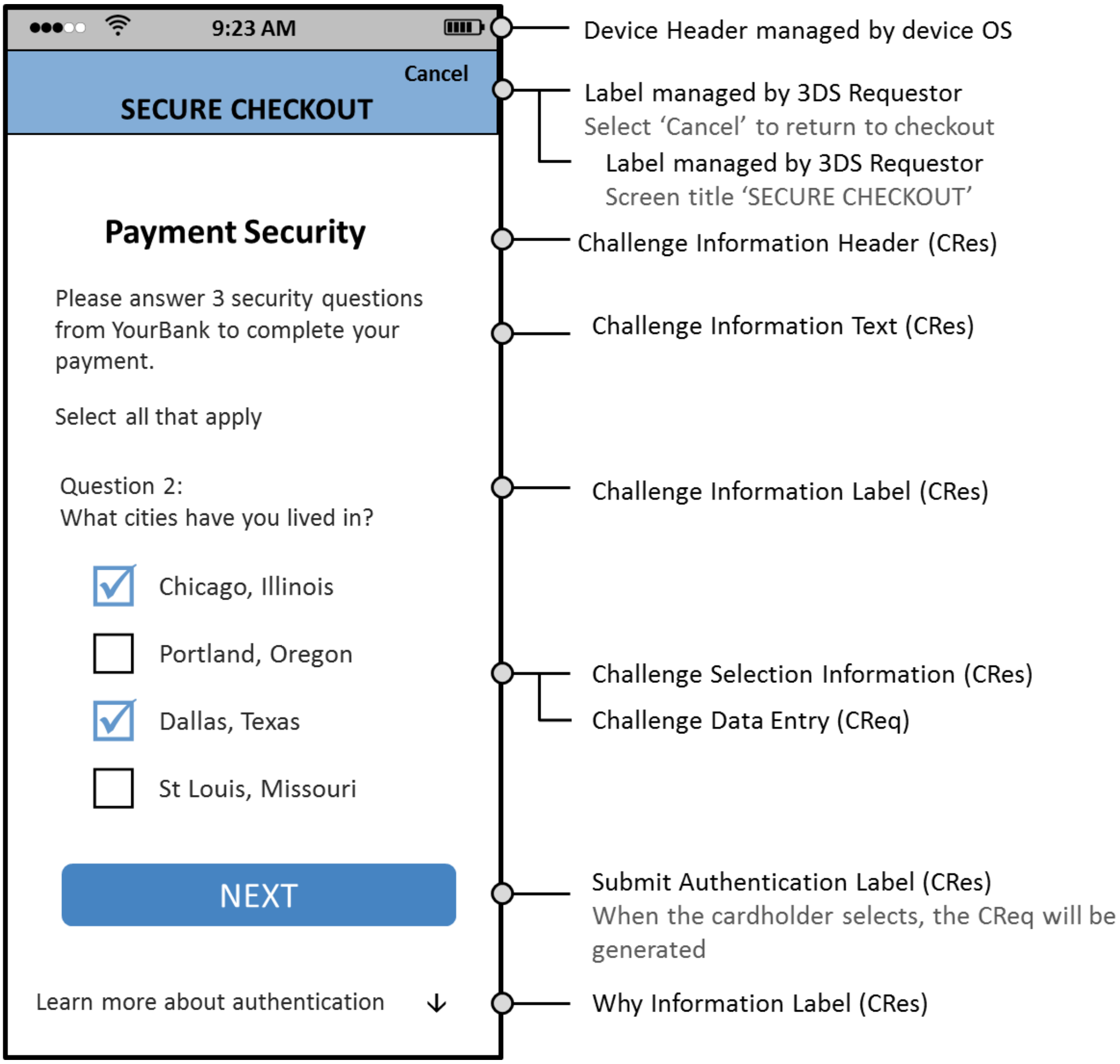
Submit Authentication Label (CRes)
When the cardholder selects, the CReq will be generated

Why Information Label (CRes)

Expandable Information Label 1 (CRes)

Figure 4.7 provides a sample format that allows multiple options to be presented to the Cardholder to obtain multiple responses on a single screen. For example, asking the Cardholder to select the cities where they have lived. This example also depicts a screen with no Issuer or Payment System branding.

Figure 4.7: Sample Native UI—Multi-select Information—Payment Authentication



Payment Security

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

Select all that apply

Question 2:
What cities have you lived in?

☑ Chicago, Illinois

□ Portland, Oregon

☑ Dallas, Texas

□ St Louis, Missouri

Learn more about authentication ↓

Device Header managed by device OS

Label managed by 3DS Requestor
Select ‘Cancel’ to return to checkout

Label managed by 3DS Requestor
Screen title ‘SECURE CHECKOUT’

Challenge Information Header (CRes)

Challenge Information Text (CRes)

Challenge Information Label (CRes)

Challenge Selection Information (CRes)

Challenge Data Entry (CReq)

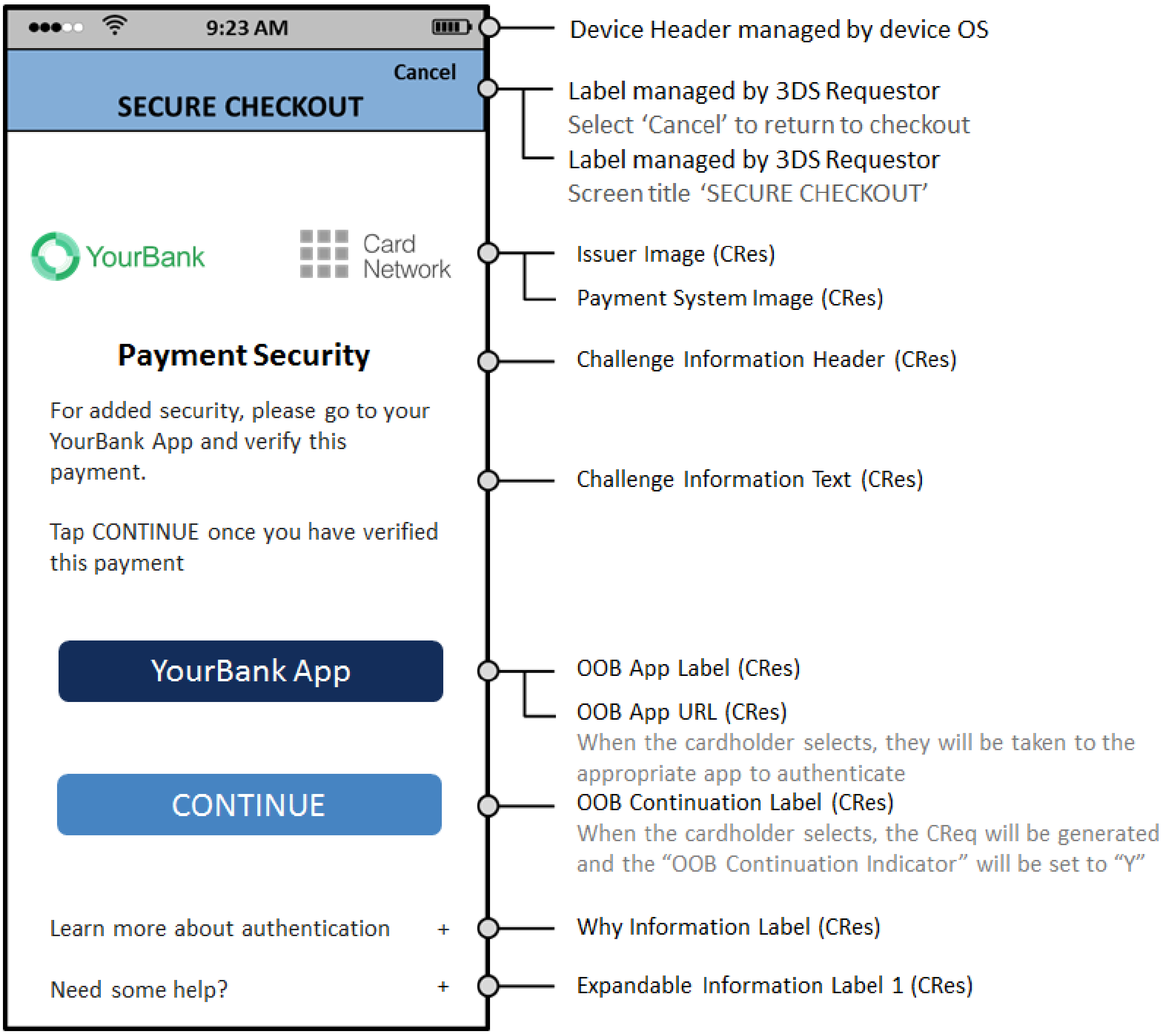
Submit Authentication Label (CRes)
When the cardholder selects, the CReq will be generated

Why Information Label (CRes)

The Out-of-Band (OOB) user interface allows Issuers to utilise authentication methods other than dynamic and static data such as an Issuer's mobile app. When an OOB challenge is necessary, the Issuer/ACS provides instructions to the Cardholder to explain the authentication process.

Figure 4.8 provides a sample OOB format to display instructions to the Cardholder.

Figure 4.8: Sample OOB Native UI Template—Payment Authentication



Payment Security

For added security, please go to your YourBank App and verify this payment.

Tap CONTINUE once you have verified this payment

Learn more about authentication

Need some help?

Device Header managed by device OS

Label managed by 3DS Requestor
Select 'Cancel' to return to checkout

Label managed by 3DS Requestor
Screen title 'SECURE CHECKOUT'

Issuer Image (CRes)

Payment System Image (CRes)

Challenge Information Header (CRes)

Challenge Information Text (CRes)

OOB App Label (CRes)

OOB App URL (CRes)
When the cardholder selects, they will be taken to the appropriate app to authenticate

OOB Continuation Label (CRes)
When the cardholder selects, the CReq will be generated and the "OOB Continuation Indicator" will be set to "Y"

Why Information Label (CRes)

Expandable Information Label 1 (CRes)

4.2.3 Native UI Message Exchange Requirements

While the CReq/CRes message exchange is the same for both Native and HTML, there is a difference in the Data Elements and Templates used. This section identifies the requirements for an App-based Native UI.

4.2.3.1 3DS SDK

The 3DS SDK shall:

Req. 4.13 After submitting the CReq to the ACS, display the 3DS Requestor App Processing screen until the CRes is received, or timeout is exceeded. Refer to Section 5.5.2.2 for CReq/CRes Message Timeout requirements.

4.2.3.2 ACS

As defined via the requirements in Section 3.1, the ACS will:

Use the secure CReq/CRes channel through the SDK for communication with the Consumer Device and populate the applicable ACS UI Type that the 3DS SDK will need to display as identified in Req. 3.55.

The ACS will continue the Challenge message cycle until complete. Whether the cycle is completed is communicated to the 3DS SDK via the Challenge Completion Indicator data element in the CRes message.

The CRes message sent to the 3DS SDK will populate the appropriate data elements to render the screen as designed by the Issuer/ACS.

4.2.3.3 3DS SDK

The 3DS SDK shall:

Req. 4.14 Control the label for the action (for example, the Cancel action) to exit the 3DS SDK and return to the 3DS Requestor App.

After receiving the CRes message from the ACS, the 3DS SDK displays the requested content through the 3DS Requestor App.

The 3DS Requestor shall:

Req. 4.15 Control UI interaction processing, for example, the Cancel action.

Req. 4.16 Have the option to display a screen title.

Req. 4.17 Return control to the 3DS Requestor App when the Cancel action in the 3DS Requestor header is selected.

Req. 4.18 Create and generate the CReq message to return to the ACS in response to Cardholder interaction with the UI.

Note: The CReq/CRes message exchange continues until the Challenge Completion Indicator in the CRes is set to Y by the ACS.

4.2.4 HTML UI Templates

The HTML UI provides Cardholders with an Issuer-consistent App-based experience across Consumer Devices that are able to render HTML. The HTML UI templates provides Issuers the ability to include Issuer-specific design elements (for example, branding, colours, and/or fonts).

The HTML UI implementation establishes a client-server relationship between the ACS-provided HTML document loaded in a 3DS Requestor's web view and the SDK process itself. This is accomplished by intercepting remote URL requests issued by the web view, and handling them within the SDK, rather than allowing them to pass through to the Consumer Device operating system and hence on to the Internet. This has two effects:

- Prevents maliciously formed HTML within the web view flow from requesting external resources or redirecting to an external malicious site (for example, a phishing page).
- Changes the web view form into an extension of the SDK's UI, one that's defined by the remote ACS using HTML, rather than by the SDK or 3DS Requestor's App.

Key HTML UI considerations:

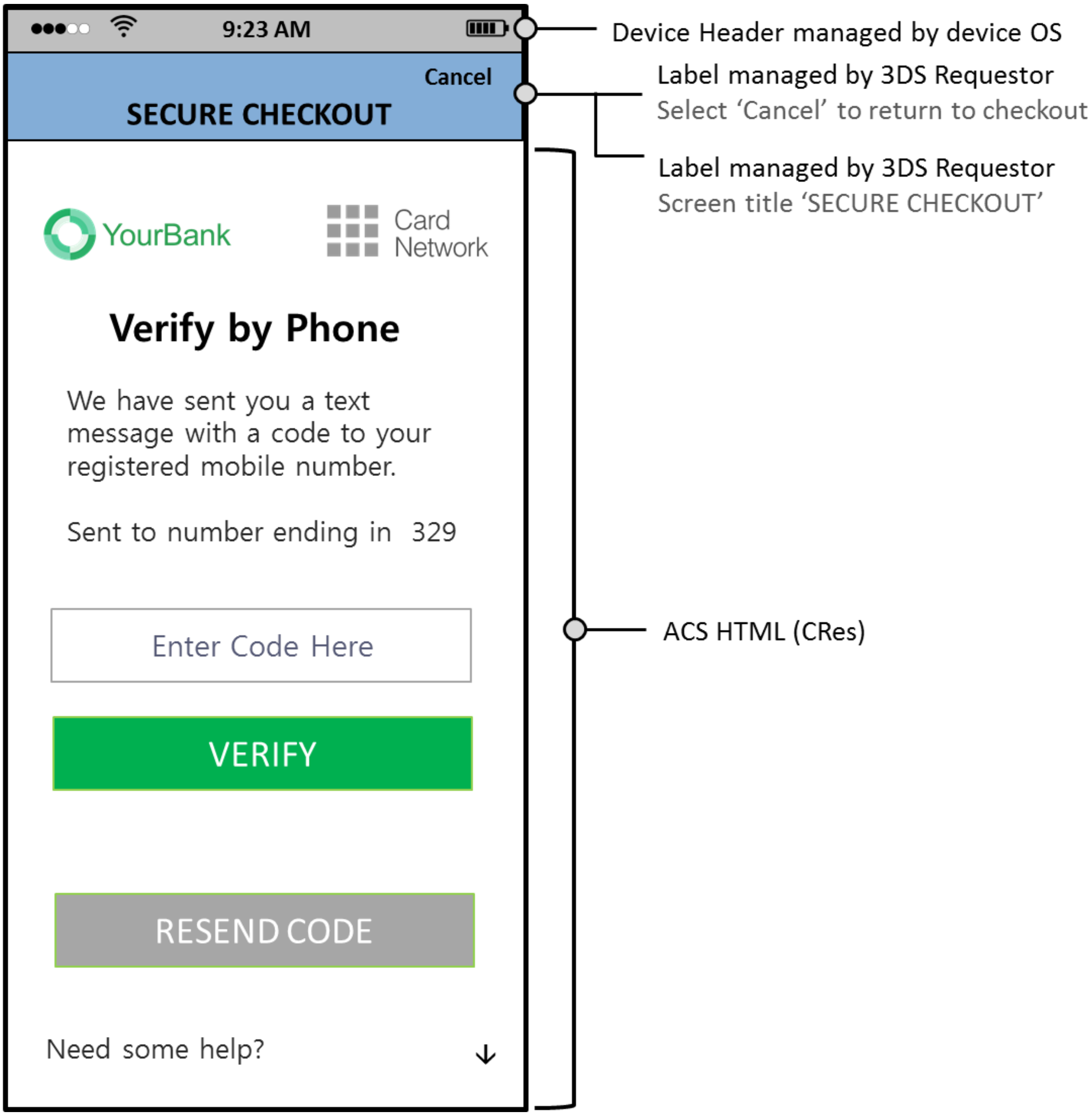
- The contents of the web view are received as responsive HTML directly from the ACS and displayed in the web view by the SDK. Navigation attempts from within the web view are captured by the SDK and processed internally, rather than being passed to the operating system and network stack. In addition to navigation attempts, the SDK also captures external resource requests (image loads, external .JS scripts, CSS, etc.)
- The web view element is not being utilised as a browser, but as a UI element whose content is the HTML and Cascading Style Sheets (CSS) provided by the ACS.
- A secure communication channel is established between the Consumer Device and the ACS for routing all the network communications. By defining all interaction with the web view in terms of the SDK, it clearly indicates that the SDK defines and owns the UI, and that the 3DS Requestor's App is isolated from the challenge interaction.

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

> **Note: App-based HTML will function on all Consumer Devices that support HTML display.**

Figure 4.9 provides a sample Payment Authentication HTML OTP UI template that includes Issuer branding.

Figure 4.9: Sample HTML OTP User Interface Template—Payment Authentication



We have sent you a text message with a code to your registered mobile number.

Sent to number ending in 329

Device Header managed by device OS

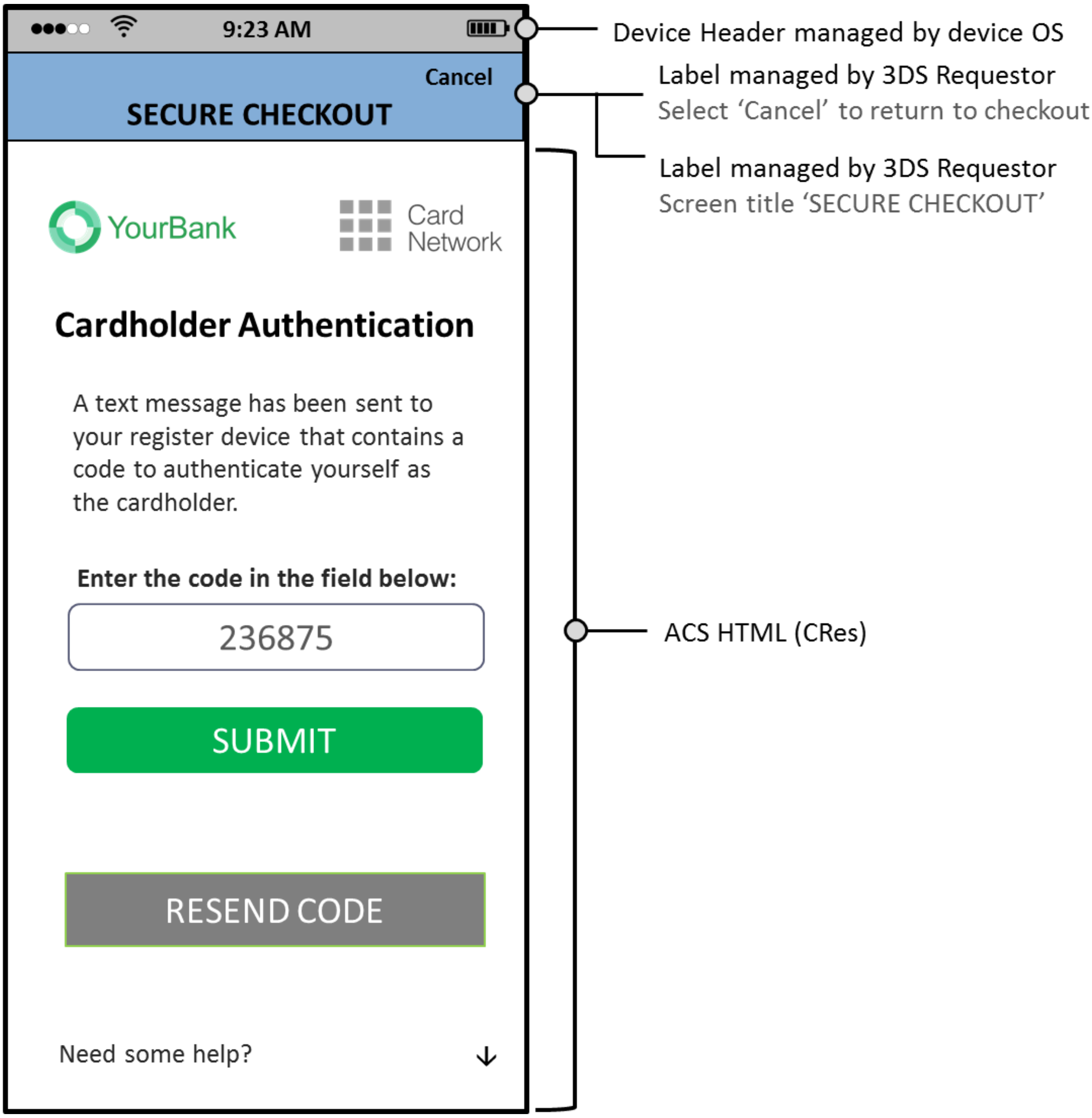
Label managed by 3DS Requestor
Select 'Cancel' to return to checkout

Label managed by 3DS Requestor
Screen title 'SECURE CHECKOUT'

ACS HTML (CRes)

Figure 4.10 provides a sample Non-Payment Authentication HTML OTP UI template that includes Issuer branding. The HTML templates for Single-select and Multi-select are visually similar to the Native UI and therefore not repeated.

Figure 4.10: Sample OTP UI Template—Non-Payment Authentication



Cardholder Authentication

A text message has been sent to your register device that contains a code to authenticate yourself as the cardholder.

Enter the code in the field below:

236875

Device Header managed by device OS

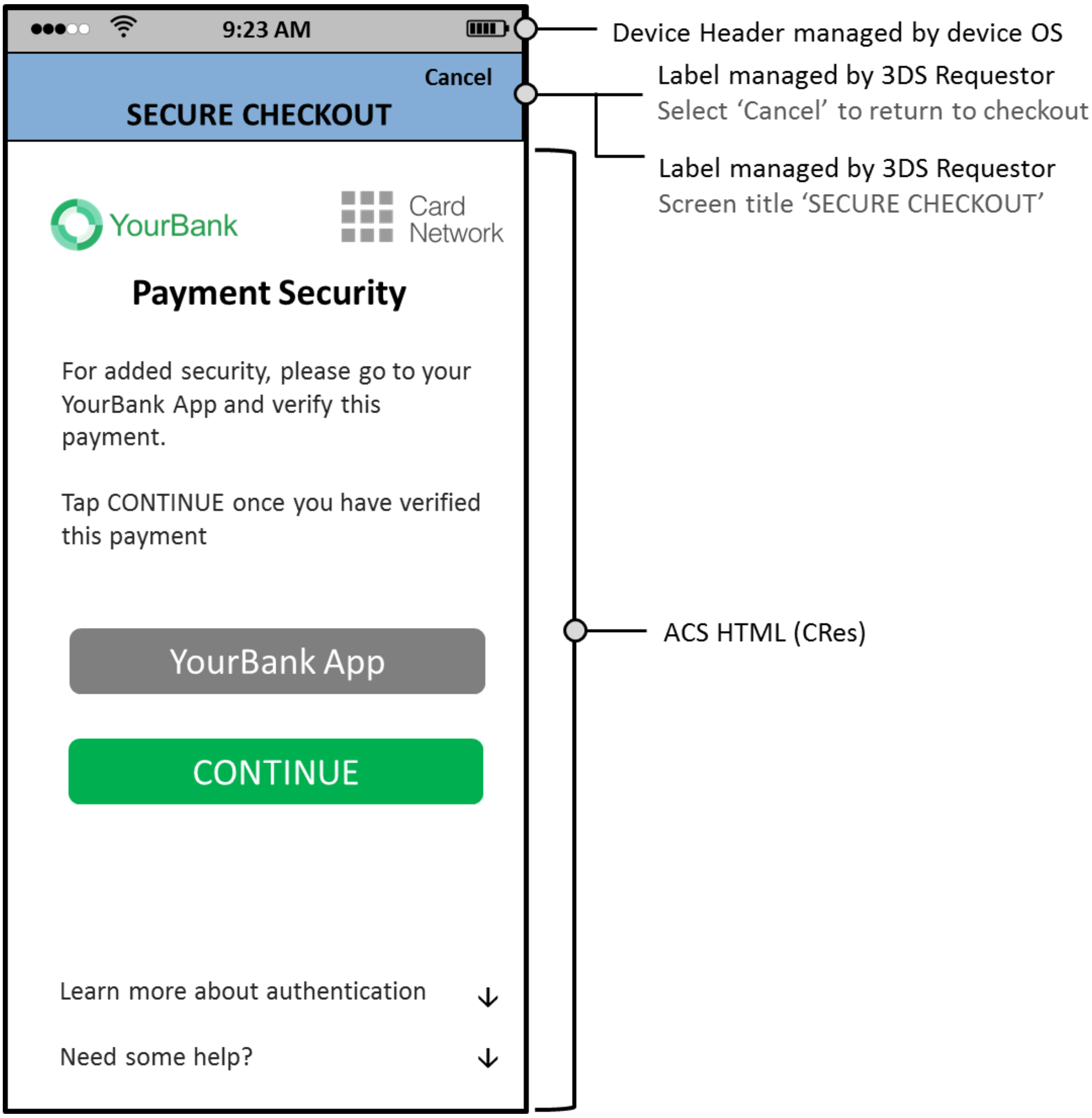
Label managed by 3DS Requestor
Select 'Cancel' to return to checkout

Label managed by 3DS Requestor
Screen title 'SECURE CHECKOUT'

ACS HTML (CRes)

Figure 4.11 provides a sample template illustrating the OOB HTML.

Figure 4.11: Sample OOB HTML UI Template—Payment Authentication



For added security, please go to your YourBank App and verify this payment.

Tap CONTINUE once you have verified this payment

Device Header managed by device OS

Label managed by 3DS Requestor
Select ‘Cancel’ to return to checkout

Label managed by 3DS Requestor
Screen title ‘SECURE CHECKOUT’

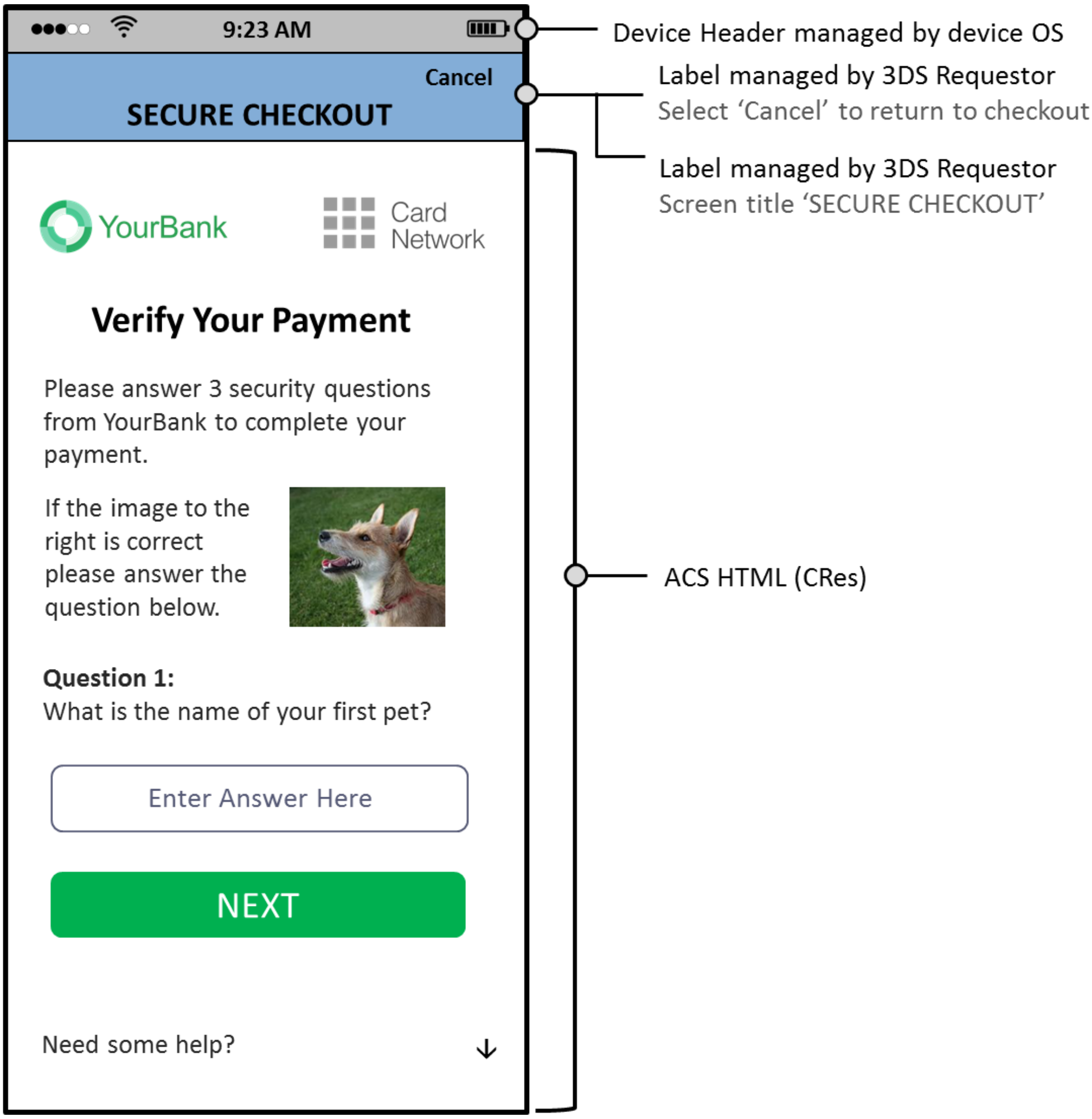
ACS HTML (CRes)

4.2.4.1 HTML Other ACS Rendering Type

The HTML Other ACS Rendering Type allows Issuers to perform authentication functionality other than the existing Native data element options standard templates. This option is exclusive to the HTML UI, adheres to the HTML template guidelines and will also be subject to the rules of the specific DS.

Figure 4.12 provides a sample HTML Other template asking the Cardholder to answer questions and confirm an image. There is not an existing data element in the Native format that supports the presentation of an image during authentication, however, the HTML Other will allow for this authentication experience.

Figure 4.12: Sample HTML Other UI Template—Payment Authentication



Verify Your Payment

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

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

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

Need some help?

Device Header managed by device OS

Label managed by 3DS Requestor
Select 'Cancel' to return to checkout

Label managed by 3DS Requestor
Screen title 'SECURE CHECKOUT'

ACS HTML (CRes)

4.2.5 HTML Message Exchange Requirements

As outlined in Section 4.2.3 Native and HTML utilise different Data Elements and Templates. This section identifies the requirements for an App-based HTML UI.

4.2.5.1 3DS SDK

The 3DS SDK shall:

Req. 4.19 After submitting the CReq to the ACS, display the 3DS App Processing screen until the CRes is received or timeout exceeded.

4.2.5.2 ACS

The ACS shall:

Req. 4.20 Ensure the HTML provided does not contain external references, either navigation attempts or external resource requests.

Req. 4.21 Embed CSS and image data to be processed and rendered within the web view entirely within the HTML.

Req. 4.22 Provide a fully-formed HTML document, including CSS and logos, following responsive design principles.

Req. 4.23 Embed all resources in the ACS-provided HTML and not fetched via external URLs.

Req. 4.24 Include in the ACS HTML an action which triggers a location change to a specified (HTTPS://EMV3DS/challenge) URL upon the Cardholder completing data input and pressing Submit.

Note: The ACS uses the location setting technique described above to communicate back to itself through the secure CReq/CRes channel.

Req. 4.25 Not bypass the SDK, or connect back to itself directly.

As defined via the requirements in Section 3.1, the ACS will then use the secure CReq/CRes channel through the SDK for communication with the Consumer Device.

The ACS will continue a Challenge message cycle until it is complete. Whether the cycle is completed is communicated to the 3DS SDK via the Challenge Completion Indicator data element in the CRes message.

The CRes message sent to the 3DS SDK will include the ACS HTML Data data element to render the screen as designed by the Issuer/ACS.

4.2.5.3 3DS SDK

The 3DS SDK shall:

Req. 4.26 Extract the HTML data from the ACS HTML data element, and display the requested content upon receiving the CRes message from the ACS.

Req. 4.27 Intercept and block any requests by the web view to fetch external resources.

Req. 4.28 Build a view that includes the 3DS Requestor header and place at the top of the view containing the ACS HTML as specified in the HTML UI templates.

Req. 4.29 Use a web view to display the UI to the Cardholder. (WebView, View Controller, etc.).

Req. 4.30 Process Cardholder actions, for example the Cancel action.

Req. 4.31 Return control to the 3DS Requestor App when the Cancel action in the 3DS Requestor header is selected.

- On HTML submit:
  - The web view will return, either a parameter string (HTML Action = GET) or a header/body (HTML Action = POST) containing the cardholders data input.
  - The SDK passes the received data, unchanged, to the ACS in the ACS HTML data element of the CReq message. The SDK shall not modify or reformat the data in any way.

The 3DS Server transmits the CReq message to the ACS.

4.3 Browser-based User Interface Overview

The Browser UI provides Cardholders with an Issuer-specific, consistent Browser-based experience across 3DS Requestors. When a challenge is necessary, the ACS provides HTML to the Browser for display to the Cardholder using the 3-D Secure UI design guidelines. Detailed requirements are described in the documentation provided by each DS.

Note: Browser Flows will function on all devices that support Browser display.

In the Browser UI, no header information is necessary as the UI appears within the browser window; either within a Lightbox or Inline. The 3DS Requestor/3DS Integrator is responsible for providing the size of the iframe as well as the placement in the 3DS Requestor website.

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

4.3.1 Processing Screen Requirements

The Browser Processing screen is displayed at the start of all 3-D Secure Browser-based transaction flows.

4.3.1.1 3DS Requestor Website

The3DS Requestor website shall:

Req. 4.32 Create a Processing screen for display during the AReq/ARes message cycle.

Note: The Processing screen is displayed by the 3DS Requestor website during AReq message processing.

Req. 4.33 Display a graphical element (for example, a progress bar or a spinning wheel) that conveys to indicate to the Cardholder that processing is occurring (Refer to Figure 4.14 and Figure 4.15 for examples).

Req. 4.34 Include the DS logo for display unless specifically requested not to include.

Req. 4.35 Not include any other design element in the Processing screen.

Req. 4.36 Display the Processing screen for a minimum of two seconds.

4.3.1.2 ACS

The ACS shall:

Req. 4.37 Create and maintain versions of the HTML that correspond to the sizes of the Challenge Window Size data element as defined in Table A.1 and provide the appropriate size in the CRes message based upon the Challenge Window Size that was provided by the 3DS Server in the ARes message.

Req. 4.38 Create a Processing screen for display during the CReq/CRes message cycle.

Req. 4.39 Display a graphical element (for example, a progress bar or a spinning wheel) that conveys to the consumer that processing is occurring.

Req. 4.40 Include the DS logo for display unless specifically requested not to include.

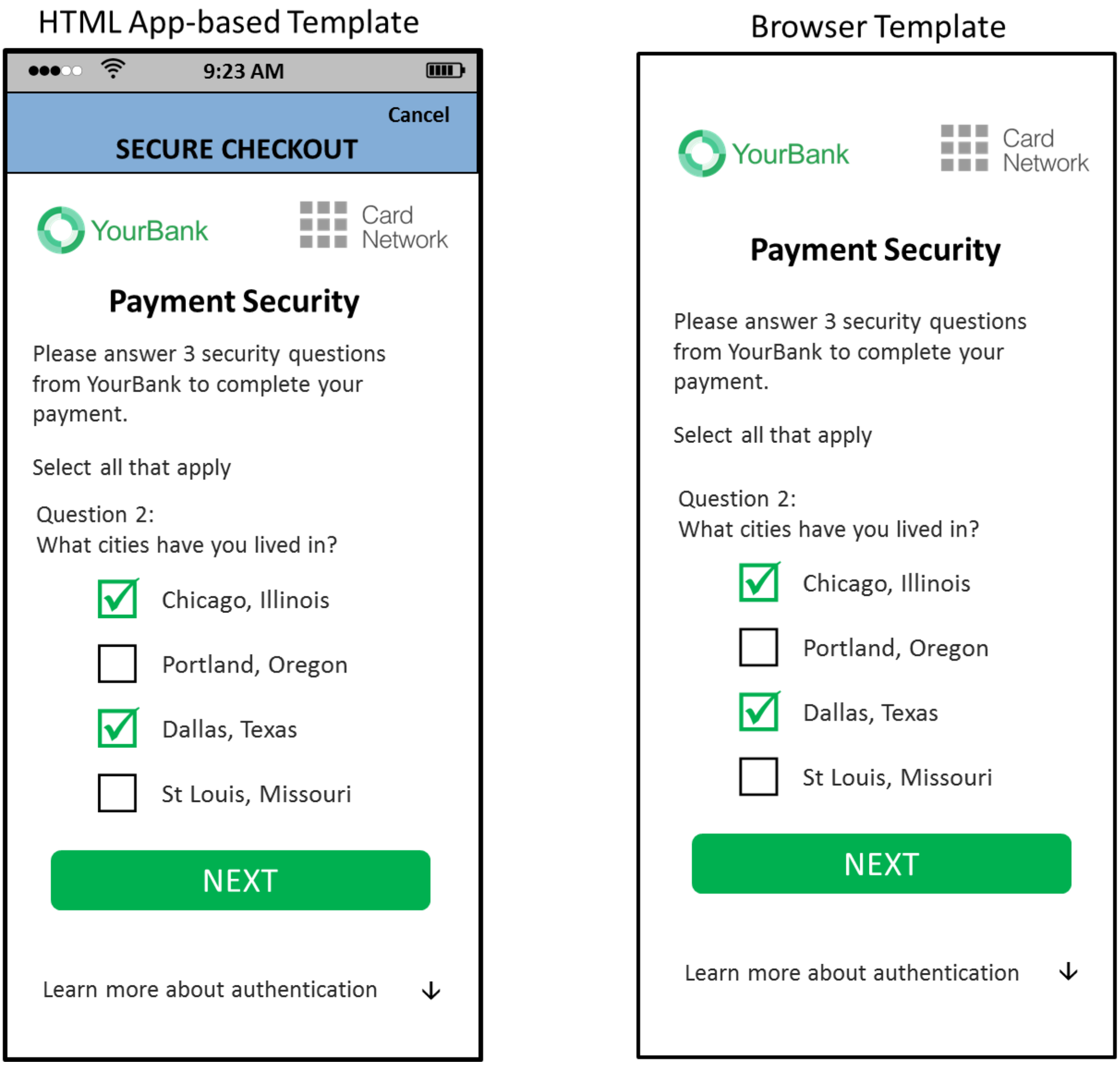
Req. 4.41 Not include any other design element in the Processing screen.

Req. 4.42 Display the processing screen for a minimum of two seconds.

Req. 4.43 Ensure browser compatibility, by using a commercial CA that is supported by major browsers.

Figure 4.13 depicts the consistency between the App-based HTML and the Browser UI.

Figure 4.13: App-based HTML and Browser UI Comparison



HTML App-based Template

9:23 AM

Cancel

SECURE CHECKOUT

YourBank

Card Network

Payment Security

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

Select all that apply

Question 2:
What cities have you lived in?

☑ Chicago, Illinois

□ Portland, Oregon

☑ Dallas, Texas

□ St Louis, Missouri

NEXT

Learn more about authentication ↓

Browser Template

YourBank

Card Network

Payment Security

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

Select all that apply

Question 2:
What cities have you lived in?

☑ Chicago, Illinois

□ Portland, Oregon

☑ Dallas, Texas

□ St Louis, Missouri

NEXT

Learn more about authentication ↓

Figure 4.14 depicts a sample Browser Lightbox processing screen.

Figure 4.14: Sample Browser Lightbox Processing Screen

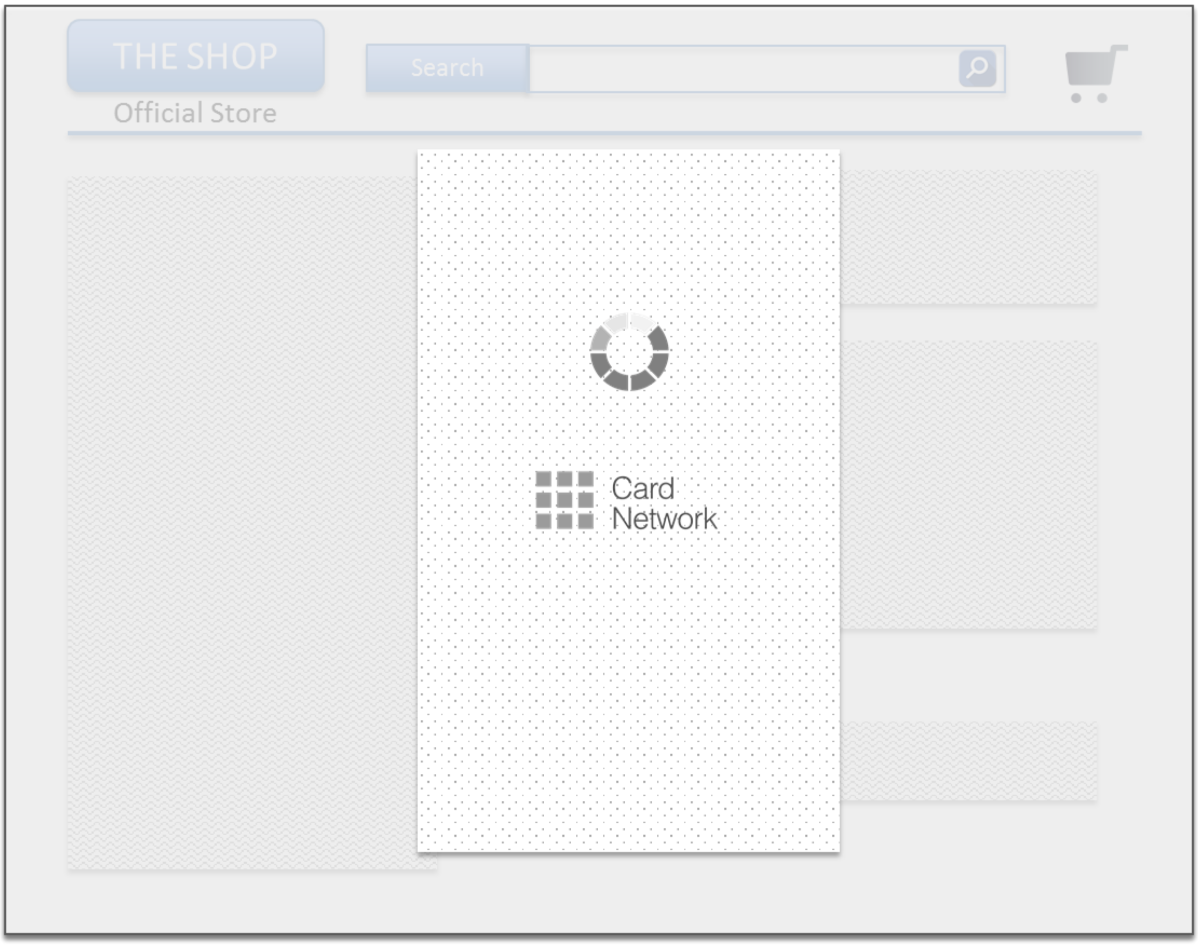


Figure 4.15 depicts a sample Inline Browser Processing screen.

Figure 4.15: Sample Inline Browser Processing Screen

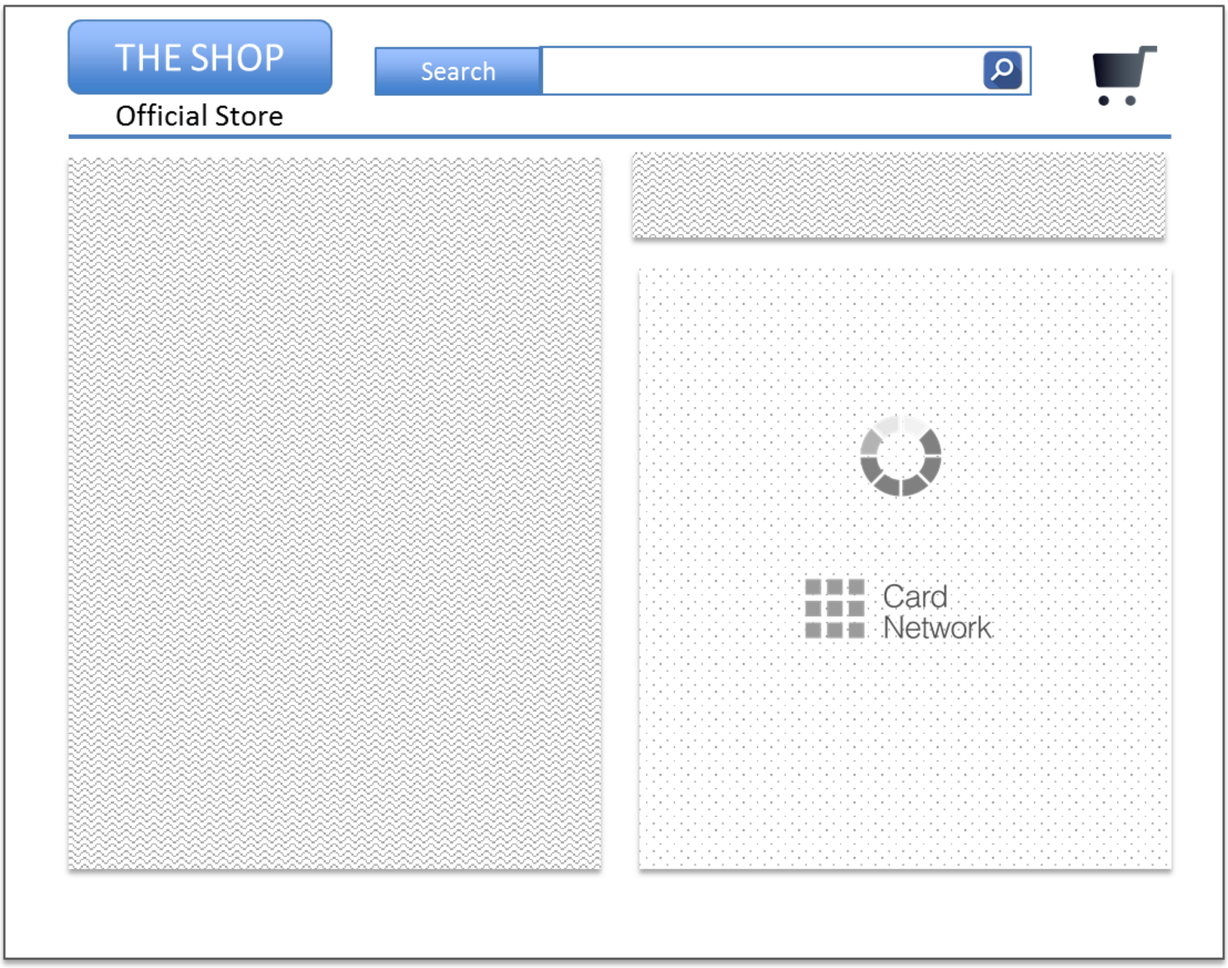
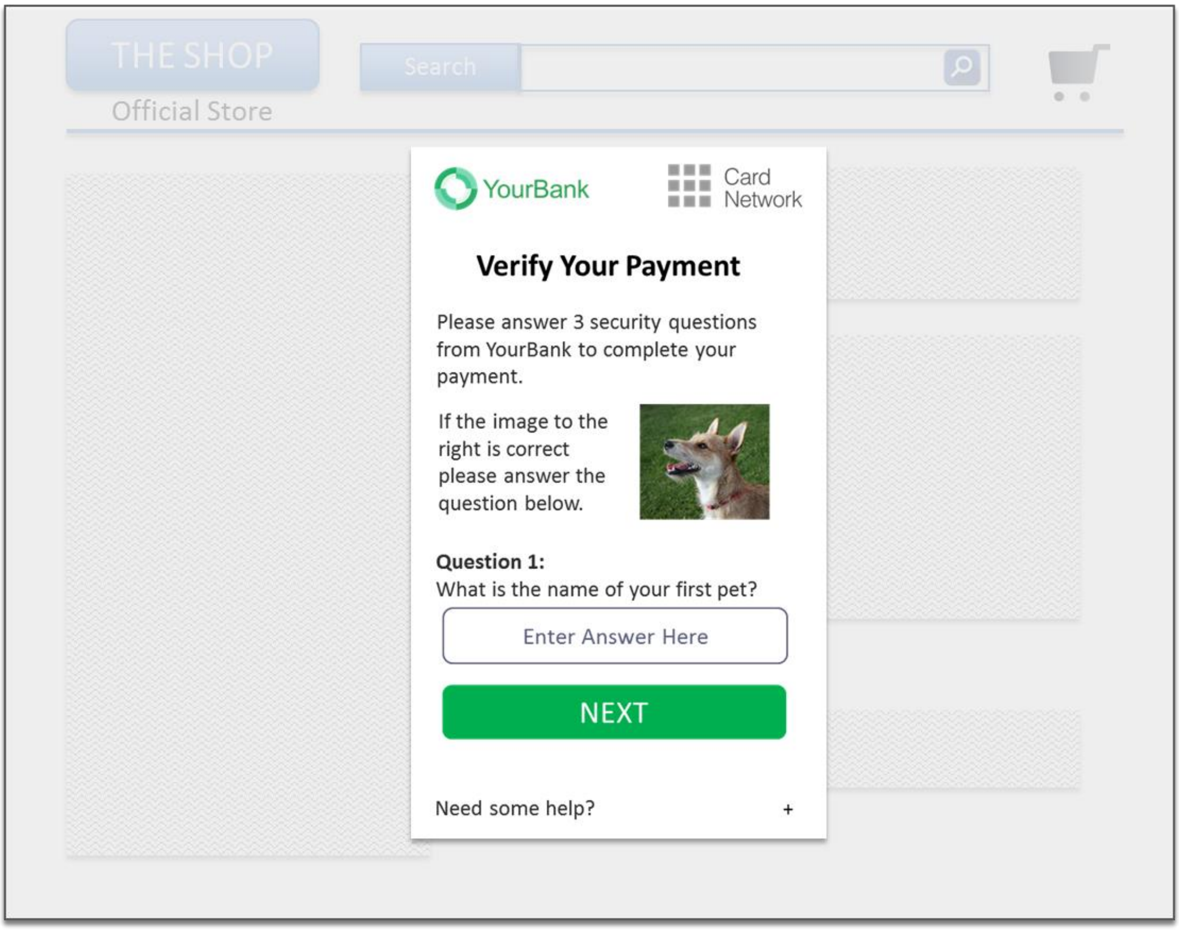


Figure 4.16 depicts a sample Browser with Lightbox UI.

Figure 4.16: Sample Browser with Lightbox UI—Payment Authentication



Verify Your Payment

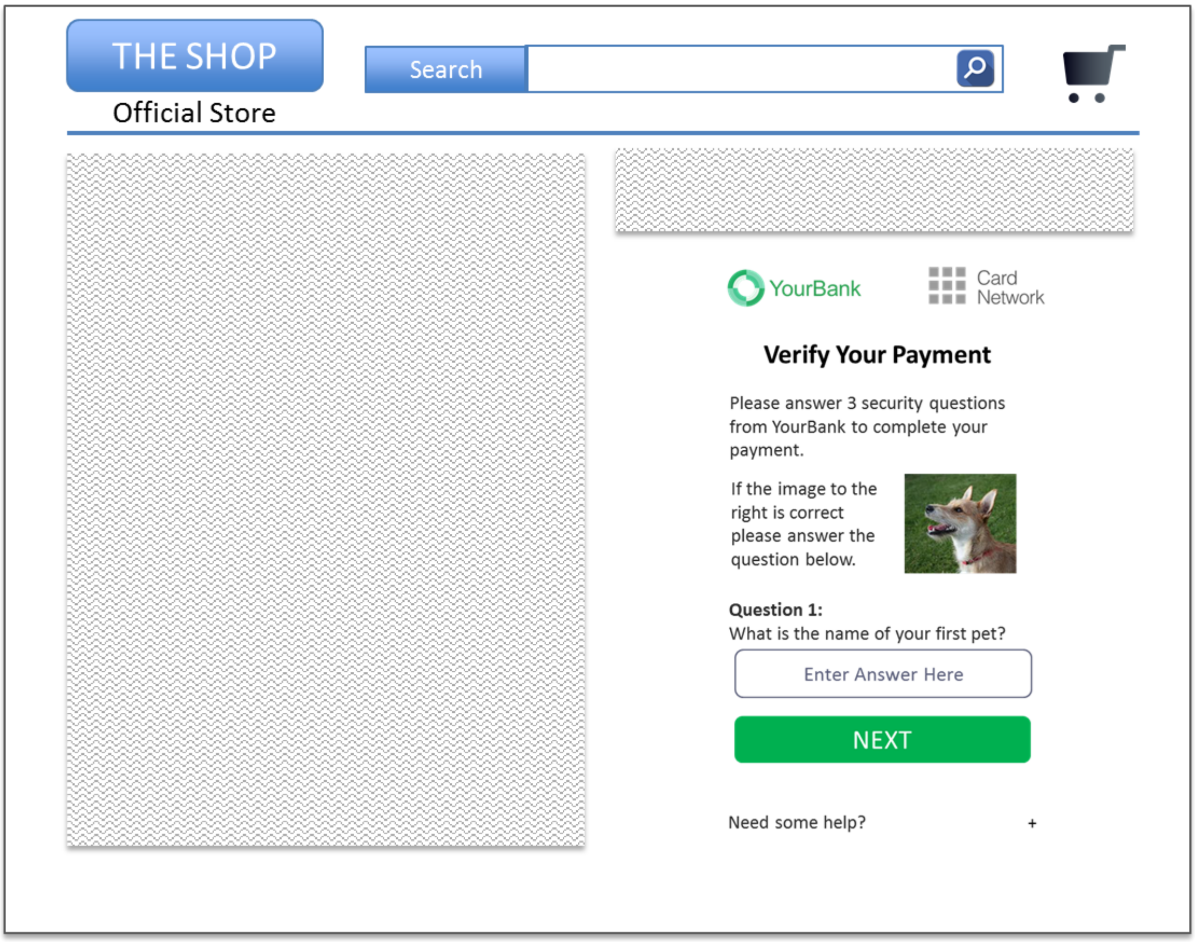
Please answer 3 security questions from YourBank to complete your payment.

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

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

Figure 4.17 depicts a sample Browser with Inline UI.

Figure 4.17: Sample Browser with Inline UI—Payment Authentication



Verify Your Payment

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

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

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

Need some help? +

This page intentionally left blank.

5 EMV 3-D Secure Message Handling Requirements

This chapter describes in more technical detail the functions of the 3-D Secure message handling including connection establishment, message parsing and validation, as well as message and error handling. The 3-D Secure messages across all flows utilise open Internet Standards to improve interoperability across domains, and those Standards are referenced in this section of the specification.

5.1 General Message Handling

5.1.1 HTTP POST

To ensure interoperability, all 3DS components shall follow these methods for HTTP POST:

Req. 5.1 All 3-D Secure message requests shall be sent via HTTP POST as defined in RFC 7231 over a secured connection as defined in Section 6.1.

Req. 5.2 Messages exchanged between 3-D Secure components shall be in the JSON data interchange format as defined in RFC 7159, or JWE object format as defined in RFC 7516.

Req. 5.3 The body of the HTTP message shall contain the JSON message properly formatted utilising the JSON required UTF-8 character set as defined in RFC 7159, or JWE object format as defined in RFC 7516.

Req. 5.4 Hypertext Transfer Protocol—HTTP/1.1 or greater shall be utilised for connectivity. Refer to RFC 2616 for detail on HTTP/1.1.

Req. 5.5 If chunked transfer coding is not used, the Content-Length: header shall be present (and set to the length of the message body). Refer to RFC 2616 for detail on HTTP/1.1 chunked transfer coding.

Req. 5.6 Responses shall be formatted as defined in Section 5.1.2 (including the formatting and Content-Type header) and sent in the reply to the HTTP POST.

5.1.2 HTTP Header—Content-Type

The HTTP Headers in the 3-D Secure messaging convey the content of the message body for receiving components to properly process the content.

3-D Secure utilises the HTTP headers, and specifically, the Content Type Header field defined in RFC 2616:

Req. 5.7 The HTTP headers shall contain the Content-Type field and have the value: Content-Type: application/json for the following messages:

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

Req. 5.8 The HTTP headers shall contain the Content-Type field and have the value: Content-Type: application/jose for the following messages:

- CReq/CRes

5.1.3 Base64 Encoding

Base64 encoding is used throughout the 3-D Secure specification to provide consistency in formatting the content of certain message elements as defined in Annex A.

Req. 5.9 The method of Base64 encoding shall follow Section 6.8 of the IETF RFC 2045.

Req. 5.10 Base64 decoding software shall ignore any white space (such as carriage returns or line ends) within Base64 encoded data, and shall not treat the presence of such characters as an error.

5.1.4 Message Protocol Version Numbers

Req. 5.11 A 3-D Secure Message Version Number shall be in the format major.minor[.update] for messaging (for example, 2.0.0).

Req. 5.12 Any Message Version Number field value less than 2.0.0 shall be returned as an error. The 3-D Secure component shall return an Error Message with an Error Code = 06.

5.1.5 Message Parsing

The recipient of a 3-D Secure message validates the message to ensure that it can be correctly processed.

When receiving a 3-D Secure message, the recipient shall validate that:

Req. 5.13 The message meets applicable technical and security requirements as defined in Annex A and Chapter 6.

Req. 5.14 The context of the transaction is valid for the receiving component based on the Message Type field.

Req. 5.15 The 3-D Secure message is properly formatted as a JSON message as defined in RFC 7159 or JWE object format as defined in RFC 7516.

Req. 5.16 The Message Type field is valid for the receiving component.

The receiving component receives the following message types:

Req. 5.17 The 3DS SDK shall only accept the following messages: CRes or Error Message. Any other message type shall be treated as an error.

Req. 5.18 The 3DS Server shall only accept the following messages: ARes, RReq, PRes, or Error Message. Any other message types shall be treated as an error.

Req. 5.19 The DS shall only accept the following messages: AReq, ARes, RReq, RRes, PReq, or Error Message. Any other message types shall be treated as an error.

Req. 5.20 The ACS shall only accept the following messages: AReq, CReq, RRes, or Error Message. Any other message types shall be treated as an error.

5.1.6 Message Content Validation

During the validation of the message, the following requirements apply:

Req. 5.21 All Required fields for the message type received shall be present as defined in Table A.1.

Req. 5.22 All Conditional fields for the message type received shall be present when the source component determines the conditions for presence are met as defined in Table A.1.

Req. 5.23 Required, Conditional, and Optional fields contained in the message shall meet formatting and length criteria as specified in Table A.1.

The message validation criteria is based on the Message Type field and applies as follows:

Req. 5.24 To support future versions of the protocol, implementations shall not use (or configure) JSON content parsers that validate strictly. If the message is syntactically correct and the validation criteria for the message type are met as defined in Table A.1, then the message shall be considered valid.

Req. 5.25 If the content validation for the Message Type received does not pass validation criteria as defined in Table A.1, then the message shall be treated as an error.

Req. 5.26 If there are additional data elements received that are not specified for the Message Type and Message Version Number, but the message otherwise passes validation, the message shall be considered valid. However the additional elements (with the exception of data extensions) shall be ignored and shall not be sent to the next 3DS component in the flow.

For example, the DS receives an AReq message from the 3DS Server with additional data elements that are not specified in Table A.1 for the AReq message type, and the DS validates the AReq content, and drops the additional elements when sending the AReq to the ACS. If the AReq does not pass validation criteria, it responds with an error.

Req. 5.27 All 3-D Secure components shall silently ignore unrecognised non-critical extension name/value pairs (that is, any extension that does not have a criticality attribute with a value of “true”) and pass them.

Req. 5.28 If the DS receives unrecognised non-critical message extensions, they shall be passed to the ACS in the AReq message or to the 3DS Server in the ARes or RReq messages.

Content Validation Rules:

Req. 5.29 3-D Secure components shall ensure response message data elements as defined in Table A.1, including Transaction IDs and Reference Numbers match that of the corresponding request message.

Req. 5.30 Upon finding any message data elements that do not pass format validation, the validating component shall generate a response message having the Invalid Request Detail or Error Description include the data element name(s) of the incorrect elements populated. Refer to Table A.4 or Table A.5 for detailed information.

The AReq message will have additional message content validation requirements based on the content of the data elements: Device Channel and Message Category.

Req. 5.31 The AReq message validation shall be based on the two data elements: Device Channel and Message Category and the presence of a data element in the AReq message shall be validated based on the content of these two data elements as defined in Table A.1.

The AReq message matrix can be viewed as:

Table 5.1: AReq Message Criteria

		Device Channel	
		01 = App	**02 = Browser**
Message Category	01 = Payment Authentication	01, 01	01, 02
	02 = Non-Payment Authentication	02, 01	02, 02

5.2 Partial System Outages

A 3-D Secure component may experience system failures that effectively render the component inoperable.

Req. 5.32 When system failures are detected, the system shall stop accepting requests until the failure has been corrected and an Error Message as defined in Section A.5.5 with an Error Code = 98 or 99 shall be sent for any outstanding requests.

5.3 3-D Secure Component Availability

All 3-D Secure components are recommended to be architected with high availability as a key factor in the software, system and infrastructure design to maintain the integrity of the entire 3-D Secure ecosystem. Additional recommendations are:

- The availability of any one 3-D Secure component should not rely on any other component to handle message routing or load balancing.
- While 3-D Secure components may optionally store multiple URLs for routing purposes, they should not be leveraged as the only high availability solution.

Note: Details about high availability best practices are outside the scope of EMV 3-D Secure.

5.4 Error Codes and Invalid Request Codes (IReq)

Req. 5.33 All 3-D Secure components shall accept any value in the Error Code field and the Invalid Request Code field.

Req. 5.34 If the list of Error Code values in Table A.4, or the list of IReq Code values in Table A.5 does not contain an entry that matches a condition detected by a 3-D Secure component, a reasonably close match shall be used.

5.5 Timeouts

5.5.1 Transaction Timeouts

Transaction Timeouts provide the maximum amount of time a 3-D Secure transaction should be considered valid. 3-D Secure components will be responsible for maintaining transaction timeout values as defined by this specification.

The ACS shall:

Req. 5.35 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.

Req. 5.36 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.

Req. 5.37 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 and Transaction Status Reason = 13 (Challenge Transaction Timed Out) and clear the ephemeral key generated and stored for use in the CReq/CRes message exchange for the current transaction.

Req. 5.38 Upon receiving a CReq message for a transaction that has timed-out, send a CRes message with IReq Code = 13 (Transaction Timed Out) to the 3DS SDK or 3DS Requestor.

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 shall:

Req. 5.39 Set a timeout value of 10 min (or 600 sec) after successfully sending each CRes message to the 3DS SDK.

Req. 5.40 If the timeout expires before receiving the next CReq message from the 3DS SDK, send an RReq message to the DS to be passed to the 3DS Server with Transaction Status = N, Transaction Status Reason = 13 (Challenge Transaction Timed Out), and Challenge Cancelation Indicator = 4 and then clear any ephemeral key generated and stored for use in the CReq/CRes message exchange for this transaction.

Req. 5.41 Upon receiving the CReq message for a transaction that has timed out, send a CRes message with IReq Code 13 (Transaction Timed-Out) to the 3DS SDK.

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 shall:

Req. 5.42 Set a timeout value of 10 min (or 600 sec) after successfully sending each challenge interface to the challenge window.

Req. 5.43 If the timeout expires before cardholder authentication can complete, 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 to the Notification URL received in the initial CReq message for the Transaction Status = N.

5.5.2 Read Timeouts

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

> **Note: The read timeout values will be defined by each DS implementation and may vary per DS.**

5.5.2.1 AReq/ARes Message Timeouts

Read timeouts in the AReq/ARes messages are handled as follows:

Req. 5.44 The 3DS Server and ACS shall set appropriate AReq message timeout values, as set by DS requirements when communicating with each DS separately.

The 3DS Server:

Req. 5.45 Any failure to complete the initial TCP/IP connection and TLS handshake to the DS shall result in an immediate retry.

Req. 5.46 Upon the second failure to complete the TCP/IP connection and TLS handshake to a DS, the 3DS Server shall try an alternate DS (if available), or stop attempting connections for 30 seconds, and then retry the connection

Req. 5.47 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.

Req. 5.48 If the DS has not responded with the ARes message before the 3DS Server read time expiry, the 3DS Server shall close the connection and result in a failed 3-D Secure transaction.

The DS:

Req. 5.49 Any failure to complete the initial TCP/IP connection and TLS handshake to the ACS shall result in an immediate retry. Upon second failure, the DS shall send an ARes message with IReq Code = 13 to the 3DS Server to complete the transaction.

Req. 5.50 The DS shall set the ACS timeout value (as defined in the relevant DS requirements) from the time the TLS handshake has completed and the full AReq message is sent for processing to the ACS.

Req. 5.51 If the DS has not received the ARes message from the ACS before the read timeout expiry, the DS shall send an ARes message with IReq Code = 13 (Transaction Timed Out) to the 3DS Server to complete the transaction.

5.5.2.2 CReq/CRes Message Timeouts

Read timeouts for the CReq/CRes messages are handled as follows:

The 3DS SDK:

Req. 5.52 Any failure to complete the initial connection and TLS handshake to the ACS Server 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.

Req. 5.53 The 3DS SDK shall set a 5 second timeout value from the time the TLS handshake has completed and the full CReq message is sent for processing to the ACS.

Req. 5.54 If the ACS does not respond with the CRes message before the 3DS SDK 5-second read timeout expiry limit, the 3DS SDK shall retry sending the CReq message with appropriate UI messaging to the user (for example, display the processing screen).

Req. 5.55 After the retry, if the ACS does not respond with the CRes message before the second 3DS SDK 5-second read timeout expiry, the 3DS SDK **ends 3-D Secure processing**, and passes an error message to the calling app, ending the 3-D Secure transaction.

5.5.2.3 RReq/RRes Message Timeouts

Read timeouts for the RReq/RRes messages are handled as follows:

The ACS:

Req. 5.56 Any 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 to the DS.

Req. 5.57 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.

Req. 5.58 If the DS has not responded with the RRes message before the 5-second read timeout expiry, the ACS shall close and re-establish a new connection.

The DS:

Req. 5.59 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 RRes message with IReq Code = 13 to the ACS to complete the transaction.

Req. 5.60 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.

Req. 5.61 If the 3DS Server has not sent the RRes message before the read timeout expiry, the DS shall **end processing**.

5.6 PReq/PRes Message Handling Requirements

The PReq/PRes messages are utilised by the 3DS Server to cache information about the version supported by available ACSs 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.

Req. 5.62 3DS Servers shall make a call to each DS with which they have registered every 24 hours at a minimum, and once per hour at a maximum to refresh their cache.

The 3DS Server shall:

Req. 5.63 Send the PReq message via a secure link with the DS established as defined in Section 6.1.2.1.

Req. 5.64 Immediately retry a connection upon any failure to complete the initial TCP/IP connection and TLS handshake to the DS.

Req. 5.65 Upon the second failure to complete the TCP/IP connection and TLS handshake to the DS, end the transaction, and periodically within the 24-hour window at 60-second intervals until the transaction completes successfully.

The DS shall:

Req. 5.66 Send the PRes message containing the DS card range information as defined in the Card Data Range data element defined in Table A.1.

Req. 5.67 Send the PRes message containing only information about card account ranges that are participating in 3-D Secure 2.0 or greater and are registered with the DS that is responding to the request.

5.7 App-based Message Handling

The App-based flows have specific requirements for security and functionality that differ from the Browser-based flows. The 3DS SDK is the main component of the 3-D Secure integration into a 3DS Requestor App.

Req. 5.68 The 3DS SDK shall be developed adhering to the *EMV 3-D Secure SDK Specification* requirements and APIs.

The 3DS SDK has two key functions:

- Provide all data as specified in the *EMV 3-D Secure SDK Specification* to be sent through the 3DS Requestor Environment to the 3DS Server and on to the DS and ACS.
- Provide the user interface (UI) and Challenge Request and Response message handling for the Challenge Flow.

5.7.1 App-based CReq/CRes Message Handling

The CReq/CRes messages are used during the Cardholder authentication and is a direct communication between the Cardholder’s device (via the 3DS Client) and the ACS.

The 3DS SDK—initialised for the Challenge Flows by the 3DS Requestor App as defined in the *EMV 3-D Secure SDK Specification*—generates the CReq message using ARes message data received from the 3DS Server through the 3DS Requestor Environment.

Upon receipt of the ARes message data from the 3DS Requestor App, the 3DS SDK validates the JSON Web Signature (JWS) used to sign the ephemeral keys and the ACS URL, generates the JSON Web Encryption (JWE) object containing the CReq message, and sends it to the ACS URL received from the 3DS Server:

Req. 5.69 The 3DS SDK shall verify the JWS signature found in the ACS Signed Content of the ARes message received from the 3DS Server as defined in 6.2.3.2 using the pre-installed public key of the DS that was called for the transaction.

Req. 5.70 The 3DS SDK shall generate JWE objects for the CReq messaging to the ACS utilising the keys derived from the 3DS SDK's own ephemeral key and the ACS Ephemeral key received from the 3DS Server.

Refer to Section 6.2.4 for details of the JWS and JWE objects.

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

Note: Several message interactions may occur as the Challenge is completed.

5.8 Browser-based Message Handling

5.8.1 3DS Method Handling

The 3DS Method allows for additional browser information to be gathered by an ACS prior to receipt of the AReq message to help facilitate the transaction risk assessment. The use of the 3DS Method by an ACS is optional.

The inclusion of 3DS Method URL and account ranges in a DS will be optional for an ACS.

Req. 5.71 The 3DS Requestor and 3DS Server shall utilise the cached PRes message data to determine the ACS's 3DS Method URL via the Card Range Data data element.

Note: Caching methods are implementation-specific and are outside of the scope of the 3DS specifications.

Req. 5.72 For the account ranges that contain a 3DS Method URL in the cached PRes message, the 3DS Requestor shall invoke the 3DS Method.

Req. 5.73 The 3DS Method call shall occur in advance of the AReq message for the same transaction being sent to the ACS.

Note: The 3DS Requestor determines the timing of the 3DS Method call to optimise the user experience.

The 3DS Requestor shall:

Req. 5.74 Obtain from the 3DS Server or load from local cache, the 3DS Method URL if it exists for the card range.

Note: The 3DS Server Transaction ID is included in both the 3DS Method and the subsequent AReq message for the same transaction. Refer to Req. 3.82 and Req. 3.84.

Req. 5.75 The 3DS Method Data shall contain the following data as specified in Table A.2 and Table A.1:

- 3DS Server Transaction ID (same as sent in the AReq message)
- 3DS Method Notification URL

Req. 5.76 Create a JSON object with the 3DS Method Data elements, then Base64 encode the JSON object.

Req. 5.77 Render a hidden HTML iframe in the Cardholder browser and send a form containing the JSON Object via HTTP POST to the ACS's 3DS Method URL obtained from the PRes message cache data.

The ACS shall:

Req. 5.78 Interact with the Cardholder browser via the HTML iframe and then store the applicable values with the 3DS Server Transaction ID for use when the AReq message is received containing the same 3DS Server Transaction ID.

Req. 5.79 Complete the required action, recall the 3DS Server Transaction ID that was received in the initial 3DS Method POST and send via a form in the Cardholder browser HTML iframe using an HTTP POST method to the 3DS Method Notification URL. Refer to Table A.2 for detailed information about 3DS Method Data.

Note: Upon notification that the 3DS Method has completed, the 3DS Server can submit the AReq message.

Req. 5.80 The ACS should ensure that any action during the execution of the 3DS Method does not impact the user experience.

5.8.2 Browser Challenge Window Requirements

The Browser challenge will occur within the Cardholder browser and the ACS will provide a formatted challenge user interface to the Cardholder within the browser challenge window.

The 3DS Requestor shall:

Req. 5.81 Select the size of the HTML iframe to be generated by the 3DS Requestor from one of the window sizes specified in the Challenge Window Size field.

Req. 5.82 Utilise a server authenticated TLS session as defined in Section 6.1.4.2.

Req. 5.83 Create a 3-D Secure challenge window by generating a CReq message, creating an HTML iframe in the Cardholder browser, and generating an HTTP POST through the iframe to the ACS URL that was received in the ARes message.

Req. 5.84 Post the CReq message containing the selected size in the Challenge Window Size field to the ACS as defined in Table A.1.

The ACS shall:

Req. 5.85 Receive the CReq message, and respond with the code to render the challenge user interface within the HTML iframe.

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

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

The 3DS Requestor shall:

Req. 5.86 Close the challenge window upon receiving the CRes message by refreshing the parent page, and removing the HTML iframe.

5.9 Message Error Handling

This section outlines the detailed error handling performed by a 3-D Secure component when an invalid message or Error Message is received.

The 3-D Secure component will receive and validate the message and in some instances will verify and decrypt the message before performing further processing.

The validation process will check the message against the requirements for presence and format of each data element in the message as defined in Table A.1 and detailed outlined in Section 5.1.6.

The verification and decryption process will perform the cryptographic process against the message as defined in the relevant sub-section of Section 6.2.

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
- The 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. The section follows the 3-D Secure transaction flow and provides a section for each combination of a 3-D Secure component receiving a specific message.

Valid message handling and exceptions based on business logic are described in Chapter 3.

5.9.1 DS AReq Message Error Handling

The DS processes the validation of the AReq message as follows:

- For a message that cannot be recognised, the DS returns to the 3DS Server an Error Message (as defined in Section A.5.5) with Error Code = 02.
- For an AReq message, the DS Validates the AReq message as defined in Table B.1 and Section 5.1.6 according to the content of the Device Channel data element:
    - If any data element present fails Validation, the DS: returns to the 3DS Server an ARes message (as defined in Table B.2) with Transaction Status = U and IReq Code = 05 with the name(s) of the data element(s) with format problems in Error Description.
    - If any required data elements are missing, the DS returns to the 3DS Server an ARes message (as defined in Table B.2) with Transaction Status = U and IReq Code = 03 with the name(s) of the missing data elements in the Error Description.
    - For Device Channel = 01-App, decrypts the SDK Encrypted Data data element of the AReq message as defined in Section 6.2.2.2:

    - If decryption fails, the DS returns to the 3DS Server an ARes message (as defined in Table B.2) with Transaction Status = U and IReq Code = 12.
  - (Otherwise, the message is not in error).

5.9.2 ACS AReq Message Error Handling

The ACS processes the validation of the AReq message as follows:

- For a message that cannot be recognised, the ACS returns to the DS an Error Message (as defined in Section A.5.5) with Error Code = 02.
- For an AReq message, the ACS Validates the AReq message as defined in Table B.1 and Section 5.1.6 according to the content of the Device Channel data element:
  - If any data element present fails validation, the ACS returns to the DS an ARes message (as defined in Table B.2) with Transaction Status = U and IReq Code = 05 with the name(s) of the data element(s) with format problems in the Invalid Request Detail.
  - If any required data elements are missing, the ACS returns to the DS an ARes message (as defined in Section B.2) with Transaction Status = U and IReq Code = 03 with the name(s) of the missing data element(s) in the Invalid Request Detail.
  - (Otherwise, the message is not in error).

5.9.3 DS ARes Message Error Handling

The DS processes the validation of the ARes message or Error Message as follows:

- For a message that cannot be recognised, the DS:
  - Returns to the ACS an Error Message (as defined in Section A.5.5) with Error Code = 02.
  - Sends to the 3DS Server a message as defined in Section 5.9.3.1.
- For an ARes message, the DS Validates the ARes message as defined in Section B.2 and Section 5.1.6:
  - If any data element present fails validation, the DS:
    - Returns to the ACS an Error Message (as defined in Section A.5.5) with Error Code = 05 with the name(s) of the data element(s) with format problems in Error Description.
    - Sends to the 3DS Server a message as defined in Section 5.9.3.1.
  - If any required data elements are missing, the DS:
    - Returns to the ACS an Error Message (as defined in Section A.5.5) with Error Code = 03 with the name(s) of the missing data element(s) in the Error Description.
    - Sends to the 3DS Server a message as defined in Section 5.9.3.1.
  - If an IReq Code is present in the ARes message, the DS sends to the 3DS Server a message as defined in Section 5.9.3.2.
  - (Otherwise, the message is not in error).

- For an Error Message, the DS sends to the 3DS Server a message as defined in Section 5.9.3.3.

To finalise, the DS logs transaction information as required by DS rules.

5.9.3.1 Message in Error

The DS:

If a specific transaction can be identified, sends to the 3DS Server using the secure link established in Req. 3.12 EITHER:

- An Error Message (as defined in Section A.5.5) with the appropriate Error Code, OR
- An ARes message (as defined in Table B.2) with Transaction Status and optionally IReq Code set to the appropriate value as defined by the specific DS.

5.9.3.2 IReq Code Received

The DS:

Sends to the 3DS Server using the secure link established in Req. 3.12 EITHER:

- The ARes message as received from the ACS, OR
- An ARes message (as defined in Table B.2) with Transaction Status and optionally IReq Code set to the appropriate value as defined by the specific DS.

5.9.3.3 Error Message Received

Sends to the 3DS Server using the secure link established in Req. 3.12 EITHER:

- The Error Message as received from the ACS, OR
- An ARes message (as defined in Table B.2) with Transaction Status and optionally IReq Code set to the appropriate value as defined by the specific DS.

5.9.4 3DS Server ARes Message Error Handling

The 3DS Server processes the validation of the ARes message or Error Message as follows:

- For a message that cannot be recognised, the 3DS Server:
  - Returns to the DS an Error Message (as defined in Section A.5.5) with Error Code = 02.
  - If a specific transaction cannot be identified, informs the 3DS Requestor Environment that an error occurred.
- For an ARes message, the 3DS Server Validates the ARes message as defined in Table B.2 and Section 5.1.6:
  - If any data element present fails validation, the 3DS Server:
    - Returns to the DS an Error Message (as defined in Section A.5.5) with Error Code = 05 with the name(s) of the data element(s) with format problems in Error Description.
    - Informs the 3DS Requestor Environment that an error occurred.

    - If any required data elements are missing, the 3DS Server:
        - Returns to the DS an Error Message (as defined in Section A.5.5) with Error Code = 03 with the name(s) of the missing data element(s) in the Error Description.
        - Informs the 3DS Requestor Environment that an error occurred.
    - If an IReq Code is present in the ARes message, the 3DS Server informs the 3DS Requestor Environment that an error occurred.
    - (Otherwise, the message is not in error).
- For an Error Message, the 3DS Server informs the 3DS Requestor Environment that an error occurred.

Note: For App-based implementations, the 3DS Requestor Environment is expected to inform the 3DS SDK about the error.

5.9.5 ACS CReq Message Error Handling—01-APP

The ACS processes the validation of the CReq message or Error Message as follows:

- For a message, which the ACS cannot verify or decrypt correctly as defined in Section 6.2.4.3 or which the ACS cannot recognise, the ACS:
    - Returns to the 3DS SDK an Error Message (as defined in Section A.5.5) with Error Code = 08 (not recognised) or 12 (Verification or decryption failure).
    - If a specific transaction cannot be identified, sends to the DS an RReq message as defined in Section 5.9.5.1.
- For a correctly verified, decrypted, and recognised CReq message, the ACS Validates the CReq message as defined in Table B.3 and Section 5.1.6 according to the Device Channel = 01-APP:
    - If any data element present fails validation, the ACS:
        - Returns to the 3DS SDK a CRes message (as defined in Table B.3) with Transaction Status = U and IReq Code = 05 with the name(s) of the data element(s) with format problems in the Invalid Request Detail and protected as defined in Section 6.2.4.4.
        - Sends to the DS an RReq message as defined in Section 5.9.5.1.
    - If any required data elements are missing, the ACS:
        - Returns to the 3DS SDK a CRes message (as defined in Table B.4) with Transaction Status = U and IReq Code = 03 with the name(s) of the missing data element(s) in the Invalid Request Detail and protected as defined in Section 6.2.4.4.
        - Sends to the DS an RReq message as defined in Section 5.9.5.1.
    - (Otherwise, the message is not in error).
- For an Error Message, the ACS sends to the DS an RReq message as defined in Section 5.9.5.1.

5.9.5.1 Message in Error

The ACS:

- Establishes a secure link with the DS as defined in Section 6.1.3.2.
- Sends to the DS an RReq message (as defined in Table B.8) with Transaction Status = U and Challenge Cancelation Indicator = 6 using the secure link.

5.9.6 ACS CReq Message Error Handling—02-BRW

The ACS processes the validation of the CReq message as follows:

- For a message that cannot be recognised, the ACS:
  - Returns to the 3DS Server (via the 3DS Client Browser) an Error Message (as defined in Section A.5.5) with Error Code = 08.
  - If a specific transaction cannot be identified, sends to the DS an RReq message as defined in Section 5.9.5.1.
- For a CReq message, the ACS Validates the CReq message as defined in Table B.3 and Section 5.1.6 according to the Device Channel = 02-BRW:
  - If any data element present fails validation, the ACS:
    - Returns to the 3DS Server (via the 3DS Client Browser) a CRes message (as defined in Table B.4) with Transaction Status = U and IReq Code = 05 with the name(s) of the data element(s) with format problems in the Invalid Request Detail.
    - Sends to the DS an RReq message as defined in Section 5.9.5.1.
  - If any required data elements are missing, the ACS:
    - Returns to the 3DS Server (via the 3DS Client Browser) a CRes message (as defined in Table B.4) with Transaction Status = U and IReq Code = 03 with the name(s) of the missing data element(s) in the Invalid Request Detail.
    - Sends to the DS an RReq message as defined in Section 5.9.5.1.
  - (Otherwise, the message is not in error).

5.9.7 3DS SDK CRes Message Error Handling

The 3DS SDK processes the validation of the CRes message or Error Message as follows:

- For a message that the 3DS SDK cannot verify or decrypt correctly as defined in Section 6.2.4.2, or cannot recognise, the 3DS SDK returns to the ACS an Error Message (as defined in Section A.5.5) with Error Code = 08 (not recognised) or 12 (verification or decryption failure).
- For a correctly verified, decrypted and recognised CRes message, the 3DS SDK Validates the CRes message as defined in Table B.4 and Section 5.1.6:
  - If any data element present fails validation, the 3DS SDK returns to the ACS an Error Message (as defined in Section A.5.5) with Error Code = 05 with the name(s) of the data element(s) with format problems in Error Description.

  - If any required data elements are missing, the 3DS SDK returns to the ACS an Error Message (as defined in Section A.5.5) with Error Code = 03 with the name(s) of the missing data element(s) in Error Description.
  - (Otherwise, the message is not in error).
- For an Error Message, the 3DS SDK informs the 3Ds Requestor Environment that an error occurred.

5.9.8 DS RReq Message Error Handling

The DS processes the validation of the RReq message as follows:

- For a message that cannot be recognised, the DS:
  - Returns to the ACS an Error Message (as defined in Section A.5.5) with Error Code = 02.
  - If a specific transaction can be identified, sends to the 3DS Server an Error Message as defined in Section 5.9.8.1.
- For an RReq message, the DS Validates the RReq message as defined in Table B.8 and Section 5.1.6:
  - If any data element present fails validation, the DS:
    - Returns to the ACS an RRes message (as defined in Table B.9) with Transaction Status = U and IReq Code = 05 with the name(s) of the data element(s) with format problems in the Invalid Request Detail.
    - Sends to the 3DS Server an Error Message as defined in Section 5.9.8.1.
  - If any required data elements are missing, the DS:
    - Returns to the ACS an RRes message (as defined in Table B.9) with Transaction Status = U and IReq Code = 03 with the name(s) of the missing data element(s) in the Invalid Request Detail.
    - Sends to the 3DS Server an Error Message as defined in Section 5.9.8.1.
  - (Otherwise, the message is not in error).

5.9.8.1 Message in Error

The DS:

- Establish a secure link with the 3DS Server as defined in Section 6.1.2.2 using the 3DS Server URL extracted from the AReq message and stored in Req. 3.22.
- Sends to the 3DS Server an Error Message (as defined in Section A.5.5) with the appropriate Error Code using the secure link.

5.9.9 3DS Server RReq Message Error Handling

The 3DS Server processes the validation of the RReq message or Error Message as follows:

- For a message that cannot be recognised, the 3DS Server:
  - Returns to the DS an Error Message (as defined in Section A.5.5) with Error Code = 02.

    - If a specific transaction cannot be identified, informs the 3DS Requestor Environment that an error occurred.
- For an RReq message, the 3DS Server Validates the RReq message as defined in Section B.8 and Section 5.1.6:
    - If any data element present fails validation, the 3DS Server:
        - Returns to the DS an RRes message (as defined in Table B.9) with Transaction Status = U and IReq Code = 05 with the name(s) of the data element(s) with format problems in the Invalid Request Detail.
        - Informs the 3DS Requestor Environment that an error occurred.
    - If any required data elements are missing, the 3DS Server:
        - Returns to the DS an RRes message (as defined in Table B.9) with Transaction Status = U and IReq Code = 03 with the name(s) of the missing data element(s) in the Invalid Request Detail.
        - Informs the 3DS Requestor Environment that an error occurred.
    - (Otherwise, the message is not in error).
- For an Error Message, the 3DS Server informs the 3DS Requestor Environment that an error occurred.

5.9.10 DS RRes Message Error Handling

The DS processes the validation of the RRes message or Error Message as follows:

- For a message that cannot be recognised, the DS:
    - Returns to the 3DS Server an Error Message (as defined in Section A.5.5) with Error Code = 02.
    - If a specific transaction cannot be identified, sends to the ACS an RRes message (as defined in Table B.9) with Transaction Status = U and IReq Code = 08 using the secure link established in Req. 3.63.
- For an RRes message, the DS Validates the RRes message as defined in Table B.9 and Section 5.1.6:
    - If any data element present fails validation, the DS:
        - Returns to the 3DS Server an Error message (as defined in Section A.5.5) with Error Code = 05 with the name(s) of the data element(s) with format problems in the Error Description.
        - Sends to the ACS an RRes message (as defined in Table B.9) with Transaction Status = U and IReq Code = 05 using the secure link established in Req. 3.63.
    - If any required data elements are missing, the DS:
        - Returns to the 3DS Server an Error message (as defined in Section A.5.5) with Error Code = 03 with the name(s) of the missing data element(s) in the Error Description.
        - sends to the ACS an RRes message (as defined in Table B.9) with Transaction Status = U and IReq Code = 03 using the secure link established in Req. 3.63

  - If an IReq Code is present in the RRes message, the DS sends to the ACS the RRes message as received from the 3DS Server using the secure link established in Req. 3.63.
  - (Otherwise, the message is not in error).
- For an Error message, the DS sends to the ACS the Error Message as received from the 3DS Server using the secure link established in Req. 3.63.

To finalise, the DS logs transaction information as required by DS rules.

5.9.11 ACS RRes Message Error Handling—01-APP

The ACS processes the validation of the RRes message or Error Message as follows:

- For a message that cannot be recognised, the ACS:
  - Returns to the DS an Error Message (as defined in Section A.5.5) with Error Code = 02.
  - If a specific transaction can be identified, sends to the 3DS SDK a CRes message with Transaction Status = U and IReq Code = 09 protected as defined in Section 6.2.4.4 using the secure link established in Req. 3.56.
- For an RRes message, the ACS Validates the RRes message as defined in Table B.9 and Section 5.1.6:
  - If any data element present fails validation, the ACS:
    - Returns to the DS an Error message (as defined in Section A.5.5) with Error Code = 05 with the name(s) of the data element(s) with format problems in the Error Description.
    - Sends to the 3DS SDK a CRes message with Transaction Status = U and IReq Code = 05 protected as defined in Section 6.2.4.4 using the secure link established in Req. 3.56.
  - If any required data elements are missing, the ACS:
    - Returns to the DS an Error message (as defined in Section A.5.5) with Error Code = 03 with the name(s) of the missing data element(s) in the Error Description.
    - Sends to the 3DS SDK a CRes message with Transaction Status = U and IReq Code = 03 protected as defined in Section 6.2.4.4 using the secure link established in Req. 3.56.
  - If an IReq Code is present in the RRes message, the ACS sends to the 3DS SDK a CRes message with Transaction Status = U and IReq Code = 98 protected as defined in Section 6.2.4.4 using the secure link established in Req. 3.56.
  - (Otherwise, the RRes message is not in error. Note: Transaction Status in the CRes message was then determined by the ACS in Step 17).
- For an Error message, the ACS sends to the 3DS SDK a CRes message with Transaction Status = U and IReq Code = 98 protected as defined in Section 6.2.4.4 using the secure link established in Req. 3.56.

5.9.12 ACS RRes Message Error Handling—02-BRW

The ACS processes the validation of the RRes message or Error Message as follows:

- For a message that cannot be recognised, the ACS:
    - Returns to the DS an Error Message (as defined in Section A.5.5) with Error Code = 02.
    - If a specific transaction can be identified, the ACS sends to the 3DS Server (via Browser) a CReq message.
- For an RRes message, the ACS Validates the RRes message as defined in Table B.9 and Section 5.1.6:
    - If any data element present fails validation, the ACS:
        - Returns to the DS an Error message (as defined in Section A.5.5) with Error Code = 05 with the name(s) of the data element(s) with format problems in the Error Description.
        - Sends to the 3DS Server (via Browser) a CRes message.
    - If any required data elements are missing, the ACS:
        - Returns to the DS an Error message (as defined in Section A.5.5) with Error Code = 03 with the name(s) of the missing data element(s) in the Error Description.
        - Sends to the 3DS Server (via Browser) a CRes message.
    - If an IReq Code is present in the RRes message, the ACS sends to the 3DS Server (via Browser) a CRes message.
    - (Otherwise, the message is not in error. Note: Transaction Status in the CRes message was then determined by the ACS in Step 17).
- For an Error message, the ACS sends to the 3DS Server (via Browser) a CRes message.

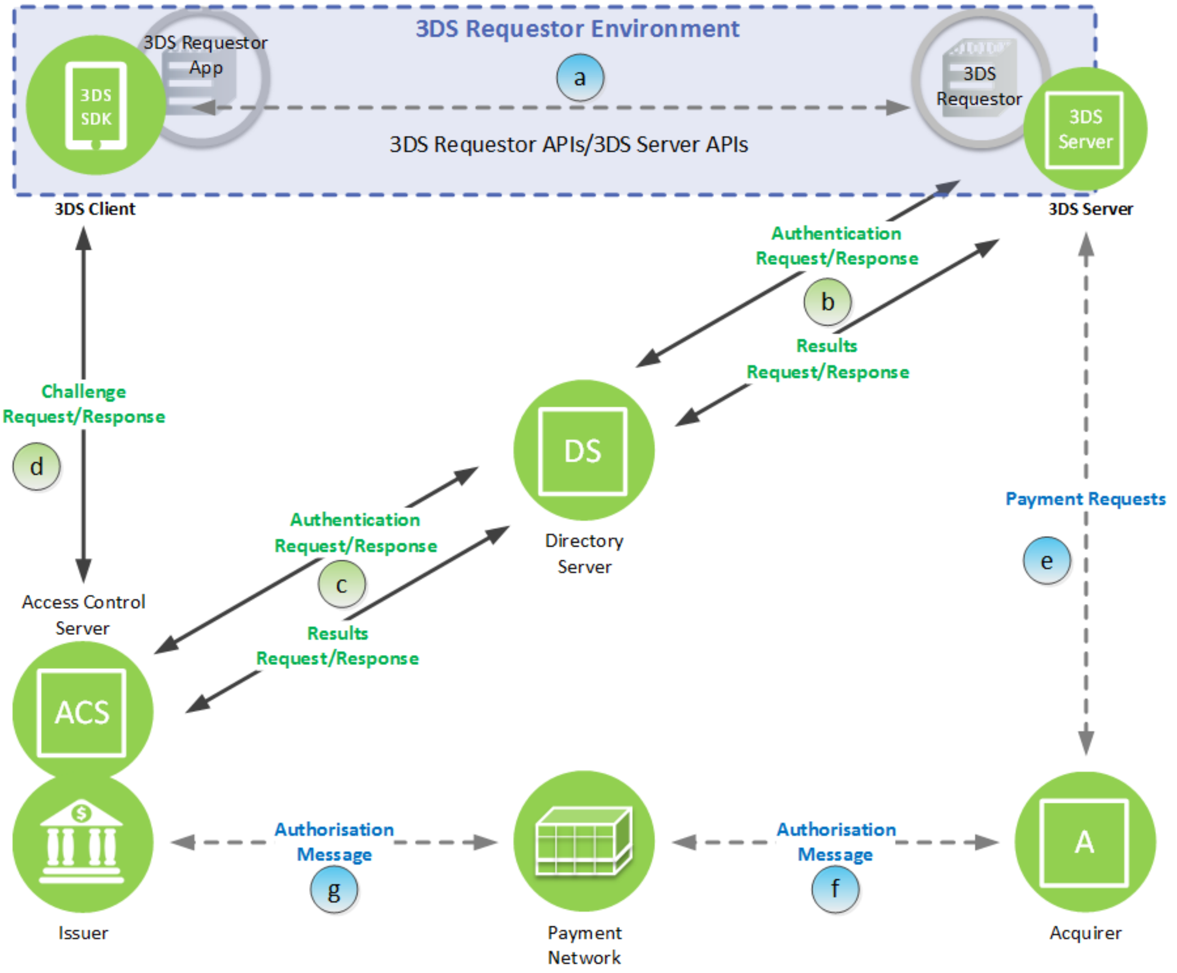
This page intentionally left blank.

6 EMV 3-D Secure Security Requirements

This chapter provides security detail for the EMV 3-D Secure security requirements referenced within this specification. Specifically, it describes the expected features of the links between the 3DS components and the 3DS security functions.

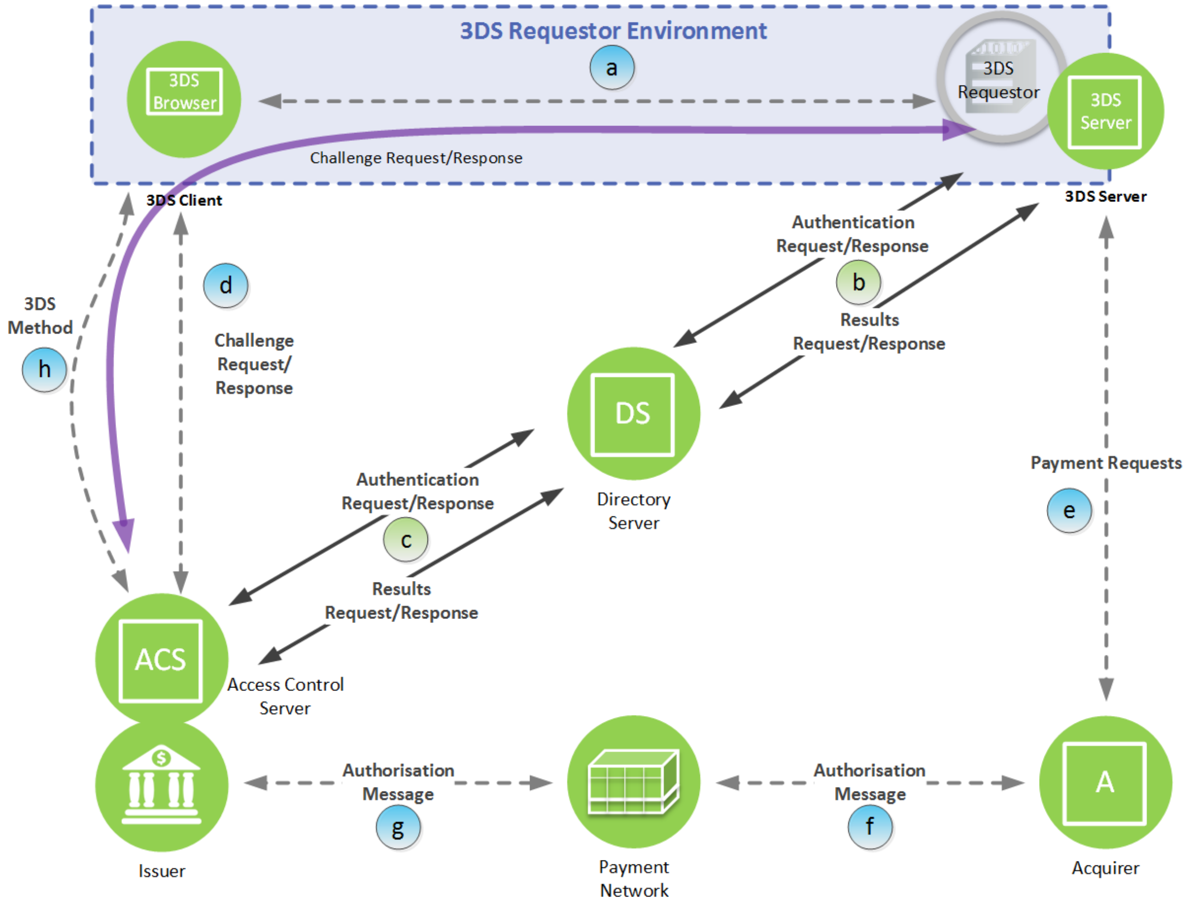
The characters in both Figure 6.1 and Figure 6.2 refer to the specific link as defined in the following sections.

Figure 6.1: Security Flow—App-based



Note: Dashed arrows and 3DS Requestor are not part of 3DS specification and are shown for clarity only

Figure 6.2: Security Flow—Browser-based



Note: Dashed arrows and 3DS Requestor are not part of 3DS specification and are shown for clarity only

6.1 Links

For each link, the following expectations apply:

6.1.1 Link a: Consumer Device—3DS Requestor

The Consumer Device to 3DS Requestor link is within the 3DS Requestor Environment between the 3DS Requestor App, and the 3DS Requestor. The 3DS Requestor link established between the 3DS Requestor and the Consumer device uses a standard Internet protocol as part of the interaction between the 3DS Requestor App or the browser on the Consumer device and the 3DS Requestor. Further links may be required between the browser and the 3DS Requestor Environment if the URLs for return of the 3DS Method Notification or the CRes are different than the initiator endpoints.

When the Cardholder interaction with the 3DS Requestor moves to 3-D Secure protocol-specific actions, the links will need to be in a secured state. This will be 3DS Requestor-specific, with the expectation that it satisfies PCI DSS with at least a standard TLS protocol with 3DS Requestor (server) authentication by the 3DS Requestor App or the Browser.

If the 3DS Requestor and 3DS Server are separate components, data transferred between them needs to be protected at a level that satisfies PCI DSS with mutual authentication of both servers.

In some implementations the customer, through the 3DS Requestor App, or the Browser will have logged on to the 3DS Requestor to start interaction, and the link may have met the 3DS Requestor's security requirements from the start. The level of customer authentication delivered by such a process is carried in the 3DS Requestor Authentication Information data element.

6.1.2 Link b: 3DS Server—DS

6.1.2.1 For AReq/ARes

The 3DS Server to DS link for the AReq/ARes messages is established using a standard 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.

- Protocol—TLS using cipher suite as described in Annex D
- 3DS Server public key Pb_R
  - Client Certificate format: X.509
- DS public key: Pb_{DS}
  - Server Certificate format: X.509
- CA signing 3DS Server key—DS CA
- CA signing DS key—DS CA

6.1.2.2 For RReq/RRes

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

- Protocol—TLS using cipher suite as described in Annex D
- DS public key: Pb_{DS}
- Client Certificate format: X.509
- 3DS Server public key Pb_R
- Server Certificate format: X.509
- CA signing DS key—DS CA
- CA signing 3DS Server key—DS CA

6.1.3 Link c: DS—ACS

6.1.3.1 For AReq/ARes

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

- Protocol—TLS using cipher suite as described in Annex D

- DS public key Pb_{DS}
- Client Certificate format: X.509
- ACS public key Pb_{ACS}
- Server Certificate format: X.509
- CA signing DS key—DS CA
- CA signing ACS key—DS CA

6.1.3.2 For RReq/RRes

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

- Protocol—TLS using cipher suite as described in Annex D
- ACS public key Pb_{ACS}
- Client Certificate format: X.509
- DS public key Pb_{DS}
- Server Certificate format: X.509
- CA signing ACS key—DS CA
- CA signing DS key—DS CA

6.1.4 Link d: 3DS SDK—ACS

6.1.4.1 For App-based CReq/CRes

For the App-based protocol, the direct link between the 3DS SDK and the ACS is only established if the transaction requires a challenge. It is initiated by the SDK using the URL provided to it in the ARes and established using a standard TLS protocol with ACS (server) authentication by the 3DS SDK. The public key certificate for the ACS is signed by a commercial CA.

- Protocol—standard Internet
- ACS public key—commercial
  - Certificate format: commercial
- CA signing ACS key—commercial CA

The challenge and Cardholder response data is encrypted and MACed using the session keys previously established between the ACS and the 3DS SDK.

- Protocol—secure channel as described in Section 6.2.4 using the previously established session keys.

6.1.4.2 For Browser-based CReq/CRes

For the Browser-based protocol, the direct link between the Browser and the ACS is only established if the transaction requires a challenge. It is initiated by the Browser from an iframe using the URL provided to it in the ARes and established using a standard TLS protocol with ACS (server) authentication by the Browser. The public key certificate for the ACS is signed by a commercial CA.

- Protocol—standard Internet
- ACS public key—commercial
    - Certificate format: commercial
- CA signing ACS key—commercial CA

6.1.5 Link e: 3DS Integrator—Acquirer (Payment Authentication only)

The 3DS Integrator to Acquirer link is the regular payment link for the Merchant. No additional security requirements are set by 3-D Secure.

6.1.6 Link f: Acquirer—Payment System (Payment Authentication only)

The Acquirer to Payment System link is across the regular Payment System network for the Payment System involved. No additional security requirements are set by 3-D Secure.

6.1.7 Link g: Payment System—Issuer (Payment Authentication only)

The Payment System to Issuer link is across the regular Payment System network for the Payment System involved. No additional security requirements are set by 3-D Secure.

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

The link between the Browser and the ACS for the 3DS Method is opened from a hidden iframe loaded by the 3DS Server as part of the check-out page. It is used for the ACS to load JavaScript which gathers device information to be returned to the ACS. This includes the 3DS Server Transaction ID which enables the ACS to marry the information to the correct transaction.

Note that the URL of the ACS used for the 3DS Method is regarded as different from the URL of the ACS for the Challenge Flow and separate TLS links will be established for the 3DS Method and for any Challenge Flow.

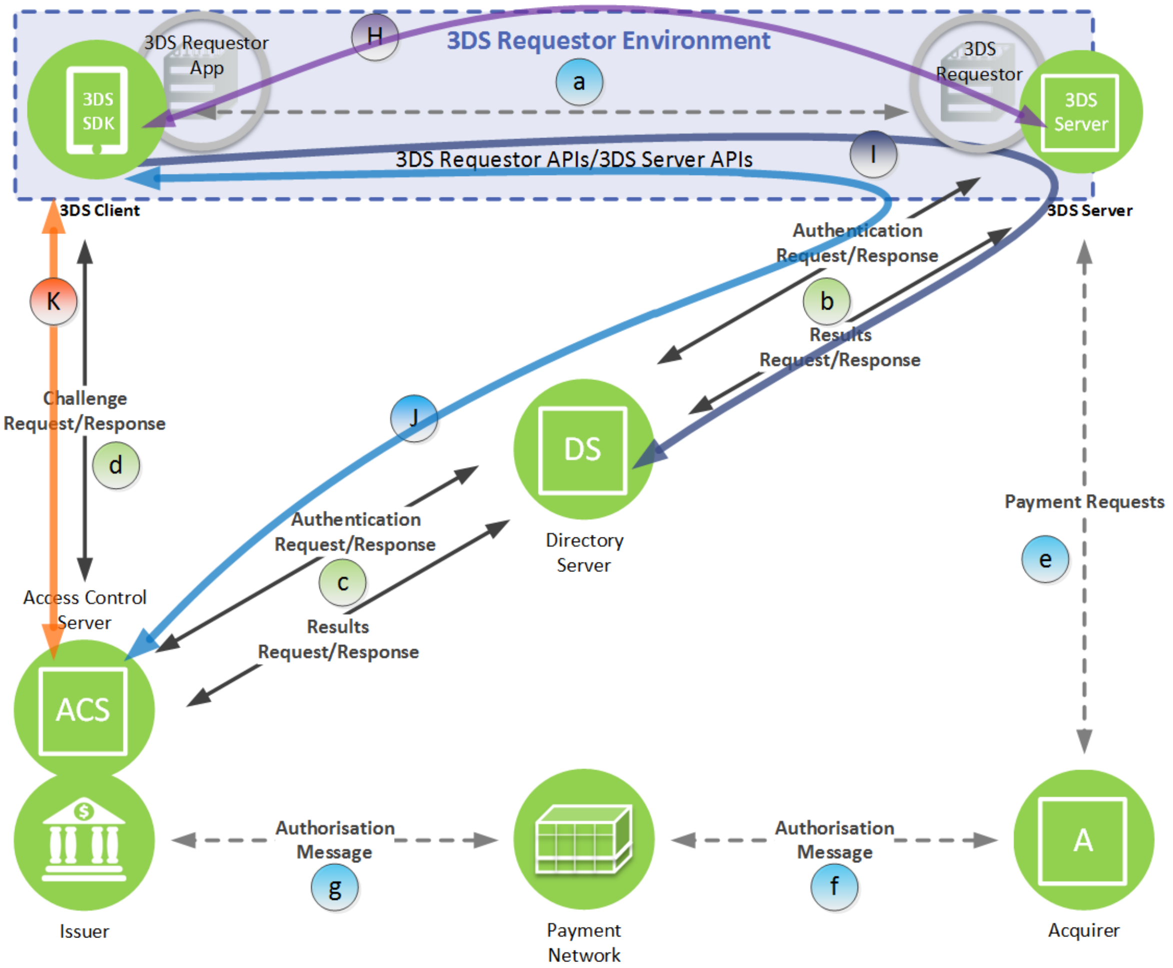
The link is established using a standard TLS protocol with server authentication of the ACS by the Browser. The public key certificate for the ACS is signed by a commercial CA.

- Protocol—standard Internet
- ACS public key—commercial
    - Certificate format: commercial
- CA signing ACS key—commercial CA

6.2 Security Functions

In addition to the internet based links there are a number of security functions that are specific to the App-based flow.

Figure 6.3: Security Functions



Note: Dashed arrows and 3DS Requestor are not part of 3DS specification and are shown for clarity only

6.2.1 Function H: Authenticity of the 3DS SDK

3DS Requestors deploy an EMVCo approved 3DS SDK embedded in their App and are required to have a mechanism to authenticate the 3DS Requestor App to the 3DS Requestor, including confirmation that the embedded 3DS SDK has not been changed.

6.2.2 Function I: 3DS SDK Encryption to DS

This 3DS SDK to DS function occurs via the 3DS Server. The purpose is to allow the 3DS SDK to encrypt data (e.g. Device Information) destined for the ACS. The decryption occurs at the DS that is trusted by the ACS, which consequently means the data from the 3DS SDK is securely delivered to the ACS.

6.2.2.1 3DS SDK Encryption

The 3DS SDK:

- Identifies the DS public key $P_{DS,}$ associated attributes, and encryption function (relating to the BIN and optionally other information) from information provided by the 3DS Requestor Environment. If the public key cannot be identified, ceases processing and report error.
- Creates a JSON Object of Device Information.
- If $P_{DS,}$ is an RSA key:
    - Encrypts the JSON object according to JWE (RFC 7516) using JWE Compact Serialization. The parameter values supported in this version of the specification are:
        - "alg": RSA-OAEP
        - "enc": A128CBC-HS256 or A128GCM
        - All other parameters: not present
- Else if P_{DS} is an EC key:
    - Encrypts the JSON object as follows:
    - Generates a fresh ephemeral key pair ($Q_{SDK,}$ d_{SDK}) as described in Annex C.
    - Conducts a Diffie-Hellman key exchange process according to JWA (RFC 7518) in Direct Key Agreement mode using curve P-256, d_{SDK} and P_{DS} to produce a CEK. The parameter values supported in this version of the specification are:
        - "alg":ECDH-ES
        - "apv":DS Reference Number
        - "epk": P_{DS}, in JSON Web Key (JWK) format
          {"kty":"EC"
          "crv":"P-256"}
        - All other parameters: not present
    - CEK: "kty":oct - 256 bits
    - Generates 128-bit random data as IV
    - Encrypt the JSON object according to JWE (RFC 7516) using the CEK and JWE Compact Serialization. The parameter values supported in this version of the specification are:
        - "alg":dir
        - "epk": Q_{SDK}, in JSON Web Key (JWK) format
          {"kty": "EC",
          "crv": "P-256"}
        - "enc":either
      - A128CBC-HS256 using the full CEK or
      - A128GCM using the leftmost 128 bits of CEK and the IV
        - "kid":ACS Transaction ID

    - All other parameters: not present

- Deletes the ephemeral key pair (*Q_{SDK}*, *d_{SDK}*)
- Makes the resulting JWE object available to the 3DS Server as SDK Encrypted Data.

6.2.2.2 DS Decryption

The DS:

- If the JWE in the SDK Encrypted Data field indicates that a RSA key was used for encryption:
  - Decrypts the SDK Encrypted Data field from the AReq message according to JWE (RFC 7516).
- Else, if the JWE in the SDK Encrypted Data field indicates that an EC key was used for encryption:
  - Decrypts the SDK Encrypted Data field as follows:
  - Conducts a Diffie-Hellman key exchange process according to JWA (RFC 7518) in Direct Key Agreement mode using curve P-256, *$Q_{SDK,}$* and *d_{DS}* to produce a CEK. The parameter values supported in this version of the specification are:
    - "alg":ECDH-ES
    - "apv": DS Reference Number
    - "epk": *Q_{SDK}*, in JSON Web Key (JWK) format
      {"kty":"EC"
      "crv":"P-256"}
    - All *other* parameters: not present
  - CEK: "kty":oct - 256 bits
  - Decrypt the JWE in the SDK Encrypted Data field according to JWE (RFC 7516) using the CEK. If the algorithm is A128GCM the leftmost 128bits of CEK is used with the received IV. If decryption fails, ceases processing and reports error.
- Insert the result into Device Information for the AReq message to the ACS.

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

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

6.2.3.1 3DS SDK Preparation for Secure Channel

The 3DS SDK:

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

6.2.3.2 ACS Secure Channel Set-Up

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

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

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

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

The ACS:

- Generates a fresh ephemeral key pair (Q_T, d_T) as described in Annex C and makes Q_T available for inclusion in the ARes message.
- Checks that Q_C is a point on the curve P-256.
- Completes the Diffie-Hellman key exchange process as a local mechanism according to JWA (RFC 7518) in Direct Key Agreement mode using curve P-256, d_T and Q_C to produce a pair of CEKs (one for each direction) which are identified by the ACS Transaction ID. The parameter values supported in this version of the specification are:
    - "alg": ECDH-ES
    - "apv": SDK Reference Number
    - "epk": Q_C, in JSON Web Key (JWK) format
    - {"kty":"EC"
      "crv":"P-256"}
    - All other parameters: not present
    - CEK: "kty":oct - 256 bits extracted as:
        - CEK_{A-S}: 256 bits
        - CEK_{S-A}: 256 bits
- Creates a JSON object of the following data as the JWS payload to be signed:
    - {"ACSPublicKey": "Q_T", "threeDSSDKPublicKey":" Q_C", "ACSURL":"ACS URL" }
- Generates a digital signature of the full JSON object according to JWS (RFC 7515) using JWS Compact Serialization. The parameter values supported in this version of the specification are:
    - "alg": PS256[5] or ES256
    - "x5c": X.5C v3: Cert(Pb_{ACS}) plus optionally chaining certificates
- All other parameters: not present
- Includes the resulting JWS in the ARes message as ACS Signed Data
- Deletes the ephemeral key pair (Q_T, d_T)

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

- Zeros the channel counters ACSCounterAtoS (:oct – 8 bits) and ACSCounterStoA (:oct – 8 bits)

6.2.3.3 3DS SDK Secure Channel Set-Up

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

The 3DS SDK:

- Using the CA public key of the DS CA identified from information provided by the 3DS Server, Validate the JWS from the ACS according to JWS (RFC7515). The 3DS SDK is required to support both “alg” parameters PS256 and ES256. If validation fails, ceases processing and report error.
- Completes the Diffie-Hellman key exchange process according to JWA (RFC 7518) in Direct Key Agreement mode using curve P-256, d_C and Q_T to produce a pair of CEKs (one for each direction), which are identified to the ACS Transaction ID received in the ARes. The parameter values supported in this version of the specification are:
  - “alg”: ECDH-ES
  - “apv”: SDK Reference Number
  - “epk”: Q_T, in JSON Web Key (JWK) format
  - {“kty”:”EC”
    “crv”:“P-256”}
  - All other parameters: not present
  - CEK: “kty”:oct - 256 bits extracted as:
    - CEK_{A-S}: 256 bits
    - CEK_{S-A}: 256 bits
- Deletes the ephemeral key pair (Q_C, d_C)
- Zeros the channel counters SDKCounterAtoS (:oct – 8 bits) and SDKCounterStoA (:oct – 8 bits)

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

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

6.2.4.1 3DS SDK—CReq

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

- Creates a JSON object of the data elements identified in the CReq message defined in Section B.3 appended with SDKCounterStoA.
- Encrypts the JSON object according to JWE (RFC 7516) using the CEK_{S-A} obtained in Section 6.2.3.3 and JWE Compact Serialization. The parameter values supported in this version of the specification are:
  - “alg”: dir
  - “enc”: either

    - A128CBC-HS256 using the full CEK$_{S-A}$ and a fresh 128-bit random data as IV
    - A128GCM using the leftmost 128 bits of CEK$_{S-A}$ with SDKCounterStoA (padded to the left with ‘00’ bytes) as the IV
  - “kid”:ACS Transaction ID
  - All other parameters: not present
- Sends the resulting JWE to the ACS as the encrypted CReq message.
- Increments SDKCounterStoA. If SDKCounterStoA = zero, ceases processing and reports error.

6.2.4.2 3DS SDK—CRes

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

- Decrypts the message according to JWE (RFC 7516) using the CEK$_{A-S}$ obtained in Section 6.2.3.3 identified by “kid” If the algorithm is A128GCM the rightmost 128 bits of CEK$_{A-S}$ is used with SDKCounterAtoS (padded to the left with ‘FF’ bytes) as the IV. If decryption fails, ceases processing and reports error.
- Checks that ACSCounterAtoS in the decrypted message equals SDKCounterAtoS. If not ceases processing and reports error.
- Increments SDKCounterAtoS. If SDKCounterAtoS = zero, ceases processing and reports error.

6.2.4.3 ACS—CReq

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

- Decrypts the message according to JWE (RFC 7516) using the CEK$_{S-A}$ obtained in Section 6.2.3.2 identified by “kid”. If the algorithm is A128GCM the leftmost 128 bits of CEK$_{S-A}$ is used with ACSCounterStoA (padded to the left with ‘00’ bytes) as the IV. If decryption fails, ceases processing and reports error.
- Checks that SDKCounterStoA in the decrypted message equals ACSCounterStoA. If not ceases processing and reports error.
- Increments ACSCounterStoA. If ACSCounterStoA = zero, ceases processing and reports error.

6.2.4.4 ACS—CRes

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

- Creates a JSON object of the data elements identified in the CRes message defined in Section B.4 appended with ACSCounterAtoS.
- Encrypts the JSON object according to JWE (RFC 7516) using the same “enc” algorithm used by the 3DS SDK for the CReq message, the CEK$_{A-S}$ obtained in Section 6.2.3.2 identified by “kid” and JWE Compact Serialization. The parameter values supported in this version of the specification are:
  - “alg”: dir
  - “enc”: either
    - A128CBC-HS256 using the full CEK$_{A-S}$ and a fresh 128-bit random data as IV

        - A128GCM using the rightmost 128 bits of CEK_{A-S} with ACSCounterAtoS (padded to the left with ‘FF’ bytes) as the IV
    - “kid”: ACS Transaction ID
    - All other parameters: not present
- Sends the resulting JWE to the 3DS SDK as the encrypted CRes message.
- Increments ACSCounterAtoS. If ACSCounterAtoS = zero, ceases processing, and reports error.

Annex A 3-D Secure Data Elements

This annex contains an alphabetical listing of all EMV 3DS Data Elements. Data Element information 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
- **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. Unless explicitly noted otherwise in Table A.1, if a field is required for the Device Channel and Message Category of a specific transaction, the value must be present and not be empty or null.

The following Standards are used to identify Inclusion values:

- **R** = Required—Sender shall include the Data Element in the identified Message Type, Device Channel, and Message Category; Recipient shall check for Data Element Presence and Validate Data Element contents.
- **C** = Conditional—Sender shall include the Data Element in the identified Message Type **if** the Conditional Inclusion requirements are met; Recipient shall check for Data Element Presence and Validate Data Element contents. When no data is to be sent for an Optional element (including a Conditional element that is not required based on the contents of the message), the element should be absent.
- **O** = Optional—Sender may include the Data Element in the identified Message Type; Recipient shall Validate the Data Element contents when present. When no data is to be sent for an Optional element (including a Conditional element that is not required based on the contents of the message), the element should be absent.

- **Conditional Inclusion**—Identifies the applicable conditions for including the Data Element in the applicable Message Type with the responsibility of the Source component to meet the conditions.

Example:

Data Element/ Field Name	Description	Source	Length/Format/ Values	Device Channel	Message Category	Message Inclusion	Conditional Inclusion
3DS Requestor URL Field Name: 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. For example: http://server.domainname.com	3DS Server	Length: Variable, maximum 2048 characters Format: String Must be any fully qualified URL.	01-APP 02-BRW	01-PA 02-NPA	01-PA: AReq = R 02-NPA: AReq = R	

In the example above, the 3DS Requestor URL is provided by the 3DS Server and is used for both app-based and Browser-based Authentication in the Authentication Request (AReq) message. The Data Element is Required for a Payment Authentication, but is Conditional for a Non-Payment Authentication.

A.1 Missing Required Fields

A data field is missing if either the:

- name/value pair is absent, or
- field name is present but the value is empty

Unless explicitly noted, if a required field is missing, the receiving component must return an Error Message as defined in Section A.5.5 with Error Code = 03. This applies whether the field is always Required or Conditionally required.

A.2 Field Edit Criteria

Only the specified validations are to be performed. Do not reject a message based on any validation that is not listed in the tables in this annex.

If a field is present, but its value does not conform to the edit criteria specified in Table A.1, the receiving component must return an Error Message as defined in Table B.10 with Error Code = 05.

A.3 Encryption of AReq Data

The AReq message contains fields that are included without encryption (these are protected in transit by the secure links defined in Section 6.1). Additionally, the Device Information data element is encrypted by the 3DS SDK and passed to the 3DS Server before inclusion in the message to the DS. All data that is to be encrypted is sent as one block of encrypted data in the 3DS SDK Encrypted Data field as a JWE object. Only the 3DS SDK and DS process the JWE object before being broken down into its constituent fields. The resulting decrypted data will be placed into the AReq message to the ACS unencrypted in the Device Information data element.

The requirements for encryption of the Device Information data element is also indicated in the first column of message data elements in Table A.1.

A.4 EMV 3-D Secure Data Elements

Table A.1: EMV 3-D Secure Data Elements

Data Element/ Field Name	Description	Source	Length/Format/ Values	Device Channel	Message Category	Message Inclusion	Conditional Inclusion
3DS Requestor Authentication Information Field Name: threeDSRequestorAuthenticationInfo	Information about how the 3DS Requestor authenticated the cardholder before or during the transaction. Refer to Table A.11 for data elements to include. Data will be formatted into a JSON object prior to being placed into the 3DS requestor Authentication Information field of the message.	3DS Server	Length: Variable Format: String JSON Object	01-APP 02-BRW	01-PA 02-NPA	AReq = O	Optional, recommended to include

Data Element/ Field Name	Description	Source	Length/Format/ Values	Device Channel	Message Category	Message Inclusion	Conditional Inclusion
3DS Requestor Challenge Indicator Field Name: 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 Format: Number Values (one of the following): 01 = No preference 02 = No challenge requested 03 = Challenge requested: 3DS Requestor Preference 04 = Challenge requested: Mandate	01-APP 02-BRW	01-PA 02-NPA	AReq = O	

Data Element/ Field Name	Description	Source	Length/Format/ Values	Device Channel	Message Category	Message Inclusion	Conditional Inclusion
3DS Requestor ID Field Name: threeDSRequestorID	DS assigned 3DS Requestor identifier. Each DS will provide a unique ID to each 3DS Requestor on an individual basis. Note: Any individual DS may impose specific formatting and character requirements on the contents of this field.	3DS Server	Length: Variable, maximum 35 characters Format: String	01-APP 02-BRW	01-PA 02-NPA	AReq = R	
3DS Requestor Name Field Name: 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 Format: String	01-APP 02-BRW	01-PA 02-NPA	AReq = R	

Data Element/ Field Name	Description	Source	Length/Format/ Values	Device Channel	Message Category	Message Inclusion	Conditional Inclusion
3DS Requestor Non-Payment Authentication Indicator Field Name: threeDSRequestorNPAInd	Indicates the type of Non-Payment Authentication request. This data element provides additional information to the ACS to determine the best approach for handing a Non-Payment Authentication request.	3DS Server	Length: 2 characters Format: Number Values (one of the following): 01 = Add card 02 = Maintain card information 03 = Cardholder verification as part of EMV token ID&V 04–80 = Reserved for EMVCo future use 80–99 = Reserved for DS use	01-APP 02-BRW	02-NPA	AReq = C	Conditional to Non-Payment Required when Message Category = 02

Data Element/ Field Name	Description	Source	Length/Format/ Values	Device Channel	Message Category	Message Inclusion	Conditional Inclusion
3DS Requestor URL Field Name: 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. For example: http://server.domainname.com	3DS Server	Length: Variable, maximum 2048 characters Format: String Must be any fully qualified URL.	01-APP 02-BRW	01-PA 02-NPA	01-PA: AReq = R 02-NPA: AReq = R	
3DS Server Reference Number Field Name: threeDSServerRefNumber	Unique identifier assigned by the EMVCo secretariat upon testing and approval.	3DS Server	Length: Variable, maximum 32 characters Format: String	01-APP 02-BRW	01-PA 02-NPA	AReq = R PReq = R	

Data Element/ Field Name	Description	Source	Length/Format/ Values	Device Channel	Message Category	Message Inclusion	Conditional Inclusion
3DS Server Transaction ID Field Name: threeDSServerTransID	Universally unique transaction identifier assigned by the 3DS Server to identify a single transaction. Must be in the canonical format as defined in IETF RFC 4122. May utilise any of the specified versions if the output meets specified requirements.	3DS Server	Length: 36 characters Format: String Value: UUID	01-APP 02-BRW	01-PA 02-NPA	AReq = R ARes = R CReq = R CRes = R PReq = R PRes = R RReq = R RRes = R Erro = R	
3DS Server URL Field Name: threeDSServerURL	Fully qualified URL of the 3DS Server to which the DS will send the RReq after the challenge has completed. Incorrect formatting will result in a failure to deliver the transaction results via the RReq message. Example value: https://server.adomainname.net	3DS Server ACS	Length: Variable, maximum 2048 characters Format: String Value: Fully qualified URL	01-APP 02-BRW	01-PA 02-NPA	AReq = R	

Data Element/ Field Name	Description	Source	Length/Format/ Values	Device Channel	Message Category	Message Inclusion	Conditional Inclusion
Account Type Field Name: acctType	Indicates the type of account. For example, for a multi-account card product.	3DS Server	Length: 2 characters Format: Number Values (one of the following): 01 = Not Applicable 02 = Credit 03 = Debit 03–29 = DS or Payment System-specific 30–99 = Reserved for EMVCo future use	01-APP 02-BRW	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.
Acquirer BIN Field Name: acquirerBIN	Acquiring institution identification code as assigned by the DS receiving the AReq message. This may be the same value that is used in authorisation requests sent on behalf of the 3DS Requestor, with the format represented in ISO 8583.	3DS Server	Length: 11 characters Format: Number	01-APP 02-BRW	01-PA 02-NPA	01-PA: AReq = R 02-NPA: AReq = O	

Data Element/ Field Name	Description	Source	Length/Format/ Values	Device Channel	Message Category	Message Inclusion	Conditional Inclusion
Acquirer Merchant ID Field Name: 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. Note: Individual Directory Servers may impose specific format and character requirements on the contents of this field.	3DS Server	Length: Variable, maximum 35 characters Format: String	01-APP 02-BRW	01-PA 02-NPA	01-PA: AReq = R 02-NPA AReq = O	
ACS Transaction ID Field Name: acsTransID	Universally Unique transaction identifier assigned by the ACS to identify a single transaction. Must be in the canonical format as defined in IETF RFC 4122. May utilise any of the specified versions if the output meets specified requirements.	ACS	Length: 36 characters Format: String	01-APP 02-BRW	01-PA 02-NPA	ARes = R CReq = R CRes = R RReq = R RRes = R	

Data Element/ Field Name	Description	Source	Length/Format/ Values	Device Channel	Message Category	Message Inclusion	Conditional Inclusion
ACS Ephemeral Public Key (Q_T) Field Name: acsEphemPubKey (ACS Signed Content)	Public key component of the ephemeral key pair (d_T, Q_T) generated by the ACS and used to establish session keys between the 3DS SDK and the ACS. Value must be generated in accordance with FIPS 186-4. Note: This will be contained within the ACS Signed Content JWS Object and will not be populated in the ARes in its own field. Refer to Annex C for additional detail.	ACS	Length: Variable, maximum 128 characters Format: String Value: Binary data Base64 Encoded	01-APP	01-PA 02-NPA	ARes = C	Required if Transaction Status = C This data element will be contained within the ACS Signed Content field, and will not be a unique field.
ACS HTML Field Name: acsHTML	HTML provided by the ACS in the CRes message. Utilised in the HTML UI type during the Cardholder challenge. This value will be Base64 Encoded prior to being placed into the CRes message.	ACS	Length: Variable, maximum 100KB Format: String Value: Base64 Encoded HTML	01-APP	01-PA 02-NPA	CRes = C	Conditional if ACS upon selection of the HTML UI Type is= 5 (HTML) by the ACS.

Data Element/ Field Name	Description	Source	Length/Format/ Values	Device Channel	Message Category	Message Inclusion	Conditional Inclusion
ACS Reference Number Field Name: acsReferenceNumber	Unique identifier for the ACS that is assigned by a DS when the ACS is registered.	ACS	Length: Variable, maximum 32 characters Format: String	01-APP 02-BRW	01-PA 02-NPA	ARes = R	
ACS Rendering Type Field Name: acsRenderingType	Identifies the UI type that will be utilised by the ACS to complete the Cardholder challenge. Provides additional information about the challenge to the 3DS Requestor. This field will contain a JSON array with two values, ordered.	ACS	Length: 2 character Format: Array Values: Interface (Position 1): 01 = Native UI 02 = HTML UI UI Type (Position 2): 01 = Text 02 = Single Select 03 = Multi Select 04 = OOB 05 = HTML Other	01-APP	01-PA 02-NPA	ARes = C RReq = C	Required if Transaction Status = C; Otherwise omitted.

Data Element/ Field Name	Description	Source	Length/Format/ Values	Device Channel	Message Category	Message Inclusion	Conditional Inclusion
ACS Signed Content Field Name: acsSignedContent (Signed Content)	Contains the JWS object created by the ACS for the ARes message. The body of JWS object will contain the following Data Elements as defined in Table A.1: • ACS URL • ACS Ephemeral Public Key (Q$_{T)}$ SDK Ephemeral Public Key (Q$_{C)}$	ACS	Length: Variable Content: String Value: JWS Object	01-APP	01-PA 02-NPA	ARes = C	Conditionally required if the Transaction Status=C
ACS UI Type Field Name: uiType	User interface type that the 3DS SDK will render, which includes the specific data mapping and requirements.	ACS	Length: 1 character Format: Number Values (one of the following): 1 = Text 2 = Single Select 3 = Multi Select 4 = OOB 5 = HTML	01-APP	01-PA 02-NPA	CRes = R	

Data Element/ Field Name	Description	Source	Length/Format/ Values	Device Channel	Message Category	Message Inclusion	Conditional Inclusion
ACS URL Field Name: acsURL (ACS Signed Content)	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. Note: For App-based, this will be contained within the ACS Signed Content JWS Object and will not be populated in the ARes message in a unique field.	ACS	Length: Variable, maximum 2048 characters Format: String Value: Fully qualified URL. For example: https://server.acsdomainname.com	01-APP 02-BRW	01-PA 02-NPA	ARes = C	Required if Transaction Status = C.

Data Element/ Field Name	Description	Source	Length/Format/ Values	Device Channel	Message Category	Message Inclusion	Conditional Inclusion
Address Match Indicator Field Name: addrMatch	Indicates whether the Cardholder Shipping Address and Cardholder Billing Address are the same.	3DS Server	Length: 1 character Format: String Values (one of the following): Y = Shipping Address matches Billing Address. N = Shipping Address does not match Billing Address.	01-APP 02-BRW	01-PA 02-NPA	AReq = C	Required in 01-PA if 3DS Requestor has Billing and Shipping Address information.

Data Element/ Field Name	Description	Source	Length/Format/ Values	Device Channel	Message Category	Message Inclusion	Conditional Inclusion
Authentication Method Field Name: 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 Format: Number Values (one of the following): 01 = Static Passcode 02 = SMS OTP 03 = Keyfob or EMV cardreader OTP 04 = App OTP 05 = OTP Other 06 = KBA 07 = OOB Biometrics 08 = OOB Login 09 = OOB Other 10 = Other 11–80 = Future EMVCo Use 80–99 = DS Future 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 from the DS to the 3DS Server URL.

Data Element/ Field Name	Description	Source	Length/Format/ Values	Device Channel	Message Category	Message Inclusion	Conditional Inclusion
Authentication Type Field Name: 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 character Format: Number Values: 01 = Static 02 = Dynamic 03 = OOB	01-APP 02-BRW	01-PA 02-NPA	ARes = C RReq = C	Required in the ARes if the Transaction Status = C in the ARes. Required in the RReq if the Transaction Status = Y or N in the RReq.
Authentication Value Field Name: 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. Contains a 20-byte value that has been Base64 encoded, giving a 28 byte result.	ACS	Length: 28 characters Format: String	01-APP 02-BRW	01-PA 02-NPA	01-PA: ARes = C RReq = C 02-NPA: ARes = C RReq = C	01-PA: Required if Transaction Status = Y or A. Omitted from the RReq when sent as an abandonment notification. 02-NPA: Conditional based on DS rules.

Data Element/ Field Name	Description	Source	Length/Format/ Values	Device Channel	Message Category	Message Inclusion	Conditional Inclusion
Browser Accept Headers Field Name: browserAcceptHeader	Exact content of the HTTP accept headers as sent to the 3DS Requestor from the Cardholder’s browser. If the total length of the accept header sent by the browser exceeds 2048 characters, the 3DS Server truncates the excess portion. Refer to Section A.5.2 for additional detail.	3DS Server	Length: Variable, maximum 2048 characters Format: String	02-BRW	01-PA 02-NPA	AReq = R	All transactions originating 02-BRW shall include this field.

Data Element/ Field Name	Description	Source	Length/Format/ Values	Device Channel	Message Category	Message Inclusion	Conditional Inclusion
Browser IP Address Field Name: browserIP	IP address of the browser as returned by the HTTP headers to the 3DS Requestor 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 octets. The groups are separated by colons (:). Example IPv6 address:2011:0db8:85a3:0101:0101:8a2e:0370:7334 Refer to Section A.5.2 for additional detail.	3DS Server	Length: Variable, maximum 45 characters Format: String	02-BRW	01-PA 02-NPA	AReq = C	All transactions originating from 02-BRW shall include this field. Shall include this field where regionally acceptable.

Data Element/ Field Name	Description	Source	Length/Format/ Values	Device Channel	Message Category	Message Inclusion	Conditional Inclusion
Browser Java Enabled Field Name: browserJavaEnabled	Boolean that represents the ability of the cardholder browser to execute JavaScript. Value is returned from the navigator.javaEnabled property. Refer to Section A.5.2 for additional detail.	3DS Server	Length: Variable, 4–5 characters Format: Boolean Values (one of the following): true false	02-BRW	01-PA 02-NPA	AReq = R	All transactions originating from 02-BRW shall include this field.
Browser Language Field Name: browserLanguage	Value representing the browser language as defined in IETF BCP47. Returned from navigator.language property. Refer to Section A.5.2 for additional detail.	3DS Server	Length: Variable, 1–8 characters Format: String	02-BRW	01-PA 02-NPA	AReq = R	All transactions originating from 02-BRW shall include this field.

Data Element/ Field Name	Description	Source	Length/Format/ Values	Device Channel	Message Category	Message Inclusion	Conditional Inclusion
Browser Screen Color Depth Field Name: 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 Section A.5.2 for additional detail.	3DS Server	Length: 1–2 characters Format: Number: Values: 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	All transactions originating from 02-BRW shall include this field.
Browser Screen Height Field Name: browserScreenHeight	Total height of the Cardholder’s screen in pixels. Value is returned from the screen.height property. Refer to Section A.5.2 for additional detail.	3DS Server	Length: Variable, 1–6 characters Format: Number	02-BRW	01-PA 02-NPA	AReq = R	All transactions originating from 02-BRW shall include this field.

Data Element/ Field Name	Description	Source	Length/Format/ Values	Device Channel	Message Category	Message Inclusion	Conditional Inclusion
Browser Screen Width Field Name: browserScreenWidth	Total width of the cardholder's screen in pixels. Value is returned from the screen.width property. Refer to Section A.5.2 for additional detail.	3DS Server	Length: Variable, 1–6 characters Format: Number	02-BRW	01-PA 02-NPA	AReq = R	All transactions originating from 02-BRW shall include this field.
Browser Time Zone Field Name: browserTZ	Time difference between UTC time and the Cardholder browser local time, in minutes. Value is returned from the getTimezoneOffset() method. Refer to Section A.5.2 for additional detail.	3DS Server	Length: 1–5 characters Format: Number	02-BRW	01-PA 02-NPA	AReq = R	All transactions originating from 02-BRW shall include this field.

Data Element/ Field Name	Description	Source	Length/Format/ Values	Device Channel	Message Category	Message Inclusion	Conditional Inclusion
Browser User-Agent Field Name: browserUserAgent	Exact content of the HTTP user-agent header. 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 Section A.5.2 for additional detail.	3DS Server	Length: Variable, maximum 2048 characters Format: String	02-BRW	01-PA 02-NPA	AReq = R	All transactions originating from 02-BRW shall include this field.

Card Range Data Field Name: cardRangeData	Card range data from the DS indicating the most recent EMV 3-D Secure version supported by the ACS 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. See Table A.7 for additional information. Note: The 3DSMethodURL may be omitted if not required. Example: cardRangeData: [{“startRange”: “1000000000000000” , “endRange”: “1000000000005000”, “protocolVersion”: “2.0.0, “3DSMethodURL”: “http://www.acs.com/script”} , {“startRange”: “1000000000007000” , “endRange”: “1000000000009000”, “protocolVersion”: “2.0.1”}]	DS	Length: Variable startRange: 13–19 characters endRange: 13–19 characters protocolVersion: 5 characters 3DSMethodURL: Variable, maximum 256 characters Format: JSON Array Values: cardRangeData: [{“startRange”: “nnnnnnnnnnnnnnnn” , “endRange”: “nnnnnnnnnnnnnnnn”, “protocolVersion”: “n.n.n, “3DSMethodURL”: “URL”}	N/A	01-PA 02-NPA	PRes = R	

Data Element/ Field Name	Description	Source	Length/Format/ Values	Device Channel	Message Category	Message Inclusion	Conditional Inclusion
Card/Token Expiry Date Field Name: cardExpiryDate	Expiry Date of the PAN or token supplied to the 3DS Requestor by the Cardholder.	3DS Server	Length: 4 characters Format: Number Value: Expiry date in YYMM format	01-APP 02-BRW	01-PA 02-NPA	AReq = R	
Cardholder Account Information Field Name: acctInfo	Additional information about the Cardholder’s account provided by the 3DS Requestor. The data will be formatted into a JSON object Refer to Table A.8 for Cardholder Account Information data elements.	3DS Server	Length: Variable Format: JSON object	01-APP 02-BRW	01-PA 02-NPA	AReq = O	Optional, but strongly recommended to include
Cardholder Account Number Field Name: acctNumber	Account number that will be used in the authorisation request. May be represented by PAN, token. Format represented ISO 7813.	3DS Server	Length: Variable, 13–19 characters Format: Number	01-APP 02-BRW	01-PA 02-NPA	AReq = R	

Data Element/ Field Name	Description	Source	Length/Format/ Values	Device Channel	Message Category	Message Inclusion	Conditional Inclusion
Cardholder Account Identifier Field Name: acctID	Additional information about the account optionally provided by the 3DS Requestor.	3DS Server	Length: Variable, maximum 64 characters Format: String	01-APP 02-BRW	01-PA 02-NPA	AReq = O	
Cardholder Billing Address City Field Name: billAddrCity	The city of the Cardholder billing address associated with the card used for this purchase.	3DS Server	Length: Variable, maximum 50 characters Format: String	01-APP 02-BRW	01-PA 02-NPA	AReq = C	Required unless there is a market or regional mandate to restrict sending of this information.
Cardholder Billing Address Country Field Name: billAddrCountry	The country of the Cardholder billing address associated with the card used for this purchase. Shall be the ISO 3166-1 numeric three-digit country code, other than exceptions listed in Table A.6.	3DS Server	Length: 3 characters Format: Number	01-APP 02-BRW	01-PA 02-NPA	AReq = C	Required unless there is a market or regional mandate to restrict sending of this information.
Cardholder Billing Address Line 1 Field Name: 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 Format: String	01-APP 02-BRW	01-PA 02-NPA	AReq = C	Required unless there is a market or regional mandate to restrict sending of this information.

Data Element/ Field Name	Description	Source	Length/Format/ Values	Device Channel	Message Category	Message Inclusion	Conditional Inclusion
Cardholder Billing Address Line 2 Field Name: 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 Format: String	01-APP 02-BRW	01-PA 02-NPA	AReq = C	Required if available unless there is a market or regional mandate to restrict sending of this information.
Cardholder Billing Address Postal Code Field Name: 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 Format: String	01-APP 02-BRW	01-PA 02-NPA	AReq = C	Required unless there is a market or regional mandate to restrict sending of this information.
Cardholder Billing Address State Field Name: billAddrState	The state or province of the Cardholder billing address associated with the card used for this purchase. Value as defined in ISO 3166-2 country subdivision code.	3DS Server	Length: 3 characters Format: String	01-APP 02-BRW	01-PA 02-NPA	AReq = C	Required unless there is a market or regional mandate to restrict sending of this information.
Cardholder Email Address Field Name: email	The email address associated with the account that is either entered by the Cardholder, or is on file with the 3DS Requestor. Shall meet requirements of Section 3.4 of IETF RFC 5322.	3DS Server	Length: Variable, maximum 254 characters Format: String	01-APP 02-BRW	01-PA 02-NPA	AReq = C	Required unless there is a market or regional mandate to restrict sending of this information.

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

Cardholder Home Phone Number
Field Name: homePhone
	The home phone number provided by the Cardholder.
	A JSON object with the Country Code (cc) 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”
	}
		3DS Server	Length: Variable
			cc: 1–3 characters
			subscriber: variable, maximum 15 characters
			Format: JSON Object; strings
			Values:
			The following values in NVP format:
			• cc
			• subscriber
				01-APP
				02-BRW
					01-PA
					02-NPA
						AReq = C	Required unless there is a market or regional mandate to restrict sending of this information.

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

Cardholder Mobile Phone
Field Name: mobilePhone
	The mobile phone number provided by the Cardholder.
	A JSON object with the 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”
	}
		3DS Server	Length: Variable
			cc: 1–3 characters
			subscriber: variable, maximum 15 characters
			Format: JSON Object; strings
			Values:
			The following values in NVP format:
			• cc
			• subscriber
				01-APP
				02-BRW
					01-PA
					02-NPA
						AReq = C	Required unless there is a market or regional mandate to restrict sending of this information.

Cardholder Name Field Name: cardholderName	Name of the Cardholder. Note: Alphanumeric special characters, listed in *EMV Book 4*, “Appendix B”.	3DS Server	Length: Variable, 2–45 characters Format: String	01-APP 02-BRW	01-PA 02-NPA	AReq = C	Required unless there is a market or regional mandate to restrict sending of this information.

Data Element/ Field Name	Description	Source	Length/Format/ Values	Device Channel	Message Category	Message Inclusion	Conditional Inclusion
Cardholder Shipping Address City Field Name: shipAddrCity	City portion of the shipping address requested by the Cardholder.	3DS Server	Length: Variable, maximum 50 characters Format: String	01-APP 02-BRW	01-PA 02-NPA	AReq = C	Required unless there is a market or regional mandate to restrict sending of this information.
Cardholder Shipping Address Country Field Name: shipAddrCountry	Country of the shipping address requested by the Cardholder. Must be the ISO 3166-1 three-digit country code, other than those listed in Table A.6.	3DS Server	Length: 3 characters Format: Number	01-APP 02-BRW	01-PA 02-NPA	AReq = C	Required unless there is a market or regional mandate to restrict sending of this information.
Cardholder Shipping Address Line 1 Field Name: 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 Format: String	01-APP 02-BRW	01-PA 02-NPA	AReq = C	Required unless there is a market or regional mandate to restrict sending of this information.
Cardholder Shipping Address Line 2 Field Name: 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	01-APP 02-BRW	01-PA 02-NPA	AReq = C	Required unless there is a market or regional mandate to restrict sending of this information.

Data Element/ Field Name	Description	Source	Length/Format/ Values	Device Channel	Message Category	Message Inclusion	Conditional Inclusion
Cardholder Shipping Address Postal Code Field Name: shipAddrPostCode	The ZIP or other postal code of the shipping address requested by the Cardholder.	3DS Server	Length: Variable, maximum 16 characters Format: String	01-APP 02-BRW	01-PA 02-NPA	AReq = C	Required unless there is a market or regional mandate to restrict sending of this information.
Cardholder Shipping Address State Field Name: shipAddrState	The state or province of the shipping address associated with the card being used for this purchase, as defined in ISO 3166-2 country subdivision code (using a 2 character alphabetic code).	3DS Server	Length: Variable: maximum 3 characters Format: String	01-APP 02-BRW	01-PA 02-NPA	AReq = C	Required unless there is a market or regional mandate to restrict sending of this information.

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

Cardholder Work Phone Number
Field Name: workPhone
	The work phone number provided by the Cardholder.
	A JSON object with the 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”
	}
		3DS Server	Length: Variable
			cc: 1–3 characters
			subscriber: variable, maximum 15 characters
			Format: JSON Object; strings
			Values:
			The following values in NVP format:
			• cc
			• subscriber
				01-APP
				02-BRW
					01-PA
					02-NPA
						AReq = C	Required if available, unless there is a market or regional mandate to restrict sending of this information.

Challenge Additional Information Text Field Name: challengeAddInfo	Additional text provided by the ACS/Issuer to Cardholder during the Challenge Message exchange that could not be accommodated in the Challenge Information Text field.	ACS	Length: Variable, maximum 256 characters Format: String	01-APP	01-PA 02-NPA	CRes = C	Required based upon the ACS UI format selected.

Challenge Cancelation Indicator Field Name: challengeCancel	Indicator informing the ACS and the DS that the authentication has been cancelled.	3DS SDK ACS	Length: 1 character Format: Number Value (one of the following): 1 = Cardholder selected “Choose another payment method.” CReq 01-PA only 2 = Cardholder selected “Cancel & Continue Shopping.” CReq 01-PA only 3 = 3DS Requestor cancelled Authentication. CReq 01-PA, 02-NPA 4 = Transaction Timed Out at ACS. RReq only, 01-PA, 02-NPA 6 = Transaction Error. CReq/RReq 01-PA, 02-NPA 7 = Unknown	01-APP	01-PA 02-NPA	CReq = C RReq = C	Required if the authentication transaction was cancelled by user interaction with the cancelation button in the user interface or for any other reason as indicated.

Data Element/ Field Name	Description	Source	Length/Format/ Values	Device Channel	Message Category	Message Inclusion	Conditional Inclusion
Challenge Completion Indicator Field Name: 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 Format: String Values: 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	
Challenge Data Entry Field Name: challengeDataEntry	Contains the data that the Cardholder entered into the Native UI text field.	3DS SDK	Length: Variable, maximum 45 characters Format: String	01-APP	01-PA 02-NPA	CReq = C	Required if the Cardholder entered information for the ACS UI to validate. Conditional for Native UI.

Data Element/ Field Name	Description	Source	Length/Format/ Values	Device Channel	Message Category	Message Inclusion	Conditional Inclusion
Challenge HTML Data Entry Field Name: challengeHTMLDataEntry	Data that the Cardholder entered into the HTML UI.	3DS SDK	Length: Variable, maximum 256 characters Format: String	01-APP	01-PA 02-NPA	CReq = C	Required when ACS UI Type = 5 and challenge data has been entered into the UI.
Challenge Information Header Field Name: challengeInfoHeader	Header text that for the challenge information screen that is being presented.	ACS	Length: Variable, maximum 45 characters Format: String	01-APP	01-PA 02-NPA	CRes = C	Required if ACS UI Type = 1–4.
Challenge Information Label Field Name: challengeInfoLabel	Label to modify the text provided by the Issuer to describe what is being requested from the Cardholder.	ACS	Length: Variable, maximum 45 characters Format: String	01-APP	01-PA 02-NPA	CRes = C	Required if ACS UI Type = 1–4.
Challenge Information Text Field Name: challengeInfoText	Text provided by the ACS/Issuer to Cardholder during the Challenge Message exchange.	ACS	Length: Variable, maximum 256 characters Format: String	01-APP	01-PA 02-NPA	CRes = C	Required if ACS UI Type = 1–4.

Data Element/ Field Name	Description	Source	Length/Format/ Values	Device Channel	Message Category	Message Inclusion	Conditional Inclusion
Challenge Information Text Colour Indicator Field Name: challengeInfoTextColour	Indicates when Challenge information text should be displayed in a different colour. Defined in w3c HTML data types. Example: #F53506	ACS	Length: 7 characters Format: String Value: Hex Color Code	01-APP	01-PA 02-NPA	CRes = C	Required if ACS UI Type = 1–4.
Challenge Mandated Indicator Field Name: challengeMandated	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 Format: Sting Values (one of the following): Y = A challenge will be required by the Issuer as a condition for approving the transaction. N = A challenge is not required by the Issuer as a condition for approving the transaction.	01-APP	01-PA 02-NPA	ARes = C AReq = O	Required if Transaction Status = C.

Data Element/ Field Name	Description	Source	Length/Format/ Values	Device Channel	Message Category	Message Inclusion	Conditional Inclusion
Challenge Selection Information Field Name: 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. Example: “Challenge Selection Information”: [{"mobile": "**** **** 123"}, {"email": " s******k**@g***.com"}]	ACS	Length: Variable, each name/value pair maximum 45 characters Format: Array	01-APP	01-PA 02-NPA	CRes = C	Required if the ACS UI Type = 2 or 3

Data Element/ Field Name	Description	Source	Length/Format/ Values	Device Channel	Message Category	Message Inclusion	Conditional Inclusion
Challenge Window Size Field Name: 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: 1 character Format: Number Value: (one of the following): 1 = 250 x 400 2 = 390 x 400 3 = 500 x 600 4 = 600 x 400	02-BRW	01-PA 02-NPA	CReq = R	
Device Channel Field Name: deviceChannel	Indicates the type of channel interface being used for shopping.	3DS Server	Length: 2 characters Format: Number Values (one of the following): 01 = App-based (APP) 02 = Browser (BRW)	01-APP 02-BRW	01-PA 02-NPA	AReq = R	

Data Element/ Field Name	Description	Source	Length/Format/ Values	Device Channel	Message Category	Message Inclusion	Conditional Inclusion
Device Information Field Name: deviceInfo (SDK Encrypted Data)	Device information gathered by the 3DS SDK from a Consumer Device. This is JSON name value pairs that as a whole is Base64 encoded. This will be populated by the DS as unencrypted data to the ACS Contained in SDK Encrypted Data as the body of the JWE Object. Refer to *EMV 3-D Secure SDK Specification* for values.	DS originating from the 3DS SDK to the ACS	Length: Variable, maximum 15360 characters Format: String Value: Base64 Encoded JSON Object	01-APP	01-PA 02-NPA	AReq = R	Required if there is no market or regional mandate to restrict sending of this information.

Data Element/ Field Name	Description	Source	Length/Format/ Values	Device Channel	Message Category	Message Inclusion	Conditional Inclusion
Device Rendering Options Supported Field Name: deviceRenderOptions	Defines the UI types that the device supports for displaying specific challenge user interfaces within the SDK. This field will contain a JSON array with two objects: interface = selected number value UI = JSON Array with all selected values. Example: “deviceRenderOptions” = [{ “interface” = 3 } , { “UI” = [1,2,3,4] }]	3DS Server	Length: Variable Format: JSON Object Value: interface (Position 1 value): 1 = Native 2 = HTML 3 = Both UI (Position 2, JSON array as value): 1 = Text 2 = Single Select 3 = Multi Select 4 = OOB 5 = HTML Other	01-APP	01-PA 02-NPA	AReq = R	
DS Reference Number Field Name: dsReferenceNumber	EMVCo-assigned unique identifier to track approved DS.	DS	Length: Variable, maximum 32 characters Format: String	01-APP 02-BRW	01-PA 02-NPA	AReq = C ARes = R	The DS will populate the AReq with this data element prior to passing to the ACS.

Data Element/ Field Name	Description	Source	Length/Format/ Values	Device Channel	Message Category	Message Inclusion	Conditional Inclusion
DS Transaction ID Field Name: dsTransID	Universally unique transaction identifier assigned by the DS to identify a single transaction. Must be in the canonical format as defined in IETF RFC 4122. May utilise any of the specified versions as long as the output meets specified requirements.	DS	Length: 36 characters Format: String	01-APP 02-BRW	01-PA 02-NPA	AReq = C ARes = R RReq = R RRes = R	The DS will populate the AReq with this data element prior to passing to the ACS.
DS URL Field Name: 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. Example: http://server.domainname.c om	DS	Length: Variable, maximum 2048 characters Format: String Value: Valid fully qualified URL.	01-APP 02-BRW	01-PA 02-NPA	AReq = R	

Data Element/ Field Name	Description	Source	Length/Format/ Values	Device Channel	Message Category	Message Inclusion	Conditional Inclusion
Electronic Commerce Indicator (ECI) Field Name: eci	Payment System-specific value provided by the ACS to indicate the results of the attempt to authenticate the Cardholder. Note: Values are DS-specific	ACS	Length: 2 characters Format: Number	01-APP 02-BRW	01-PA	ARes = C RReq = C	The requirements for the presence of this field are DS specific.
Error Code Field Name: errorCode	Code indicating the type of problem identified in the message. Must be one of the values listed in Table A.4. Note: Additional values may be defined at any time. All components must accept any value.		Length: 2 characters Format: Number.	N/A	N/A	Error = R	
Error Description Field Name: errorDescription	Text describing the problem identified in the message.		Length: Variable, maximum 2048 characters Format: String	N/A	N/A	Error = R	
Error Detail Field Name: errorDetail	Additional detail regarding the problem identified in the message.		Length: Variable, maximum 2048 characters Format: String	N/A	N/A	Error = R	

Data Element/ Field Name	Description	Source	Length/Format/ Values	Device Channel	Message Category	Message Inclusion	Conditional Inclusion
EMV Payment Token Indicator Field Name: payTokenInd	Indicates that the transaction was de-tokenised prior to sending the message. This data element will be populated by the system residing in the 3-D Secure domain where the de-tokenisation occurs. The Boolean value of true is the only valid response for this field when it is present.	3DS Server DS	Length: 4 characters Format: Boolean Values (This is the only valid value if the element is present): true	01-APP 02-BRW	01-PA 02-NPA	AReq = C	Required if there is a de-tokenisation of an account number.
Expandable Information Label 1 Field Name: expandInfoLabel1	Label displayed to the Cardholder for the content in Expandable Information Text 1.	ACS	Length: Variable, maximum 45 characters Format: String	01-APP	01-PA 02-NPA	CRes = C	Required based upon the ACS UI Type selected.
Expandable Information Label 2 Field Name: expandInfoLabel2	Label to be displayed to the Cardholder for the content in Expandable Information Text 2.	ACS	Length: Variable, maximum 45 characters Format: String	01-APP	01-PA 02-NPA	CRes = C	Required based upon the ACS UI Type selected.

Data Element/ Field Name	Description	Source	Length/Format/ Values	Device Channel	Message Category	Message Inclusion	Conditional Inclusion
Expandable Information Text 1 Field Name: expandInfoText1	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 Format: String	01-APP	01-PA 02-NPA	CRes = C	Required based upon the ACS UI Type selected.
Expandable Information Text 2 Field Name: expandInfoText2	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 Format: String	01-APP	01-PA 02-NPA	CRes = C	Required based upon the ACS UI Type selected.
Instalment Payment Data Field Name: purchaseInstalData	Indicates the maximum number of authorisations permitted for instalment payments. Value shall be greater than 1.	3DS Server	Length: Variable, maximum 3 characters Format: Number	01-APP 02-BRW	01-PA 02-NPA	AReq = C	Required if the Merchant and Cardholder have agreed to instalment payments. Omitted if instalment payment authentication.

Data Element/ Field Name	Description	Source	Length/Format/ Values	Device Channel	Message Category	Message Inclusion	Conditional Inclusion
Interaction Counter Field Name: interactionCounter	Indicates the number of authentication cycles attempted by the Cardholder. Value to be tracked by the ACS. Begin at 1 and increment by 1 until the exchange is complete.	ACS	Length: 2 characters Format: Number	01-APP 02-BRW	01-PA 02-NPA	RReq = R	
Invalid Request Code Field Name: ireqCode	Code indicating the problem identified in a 3-D Secure request message. Must be one of the values listed in Table A.5.	DS 3DS Server	Length: Variable, maximum 3 characters Format: Number	01-APP 02-BRW	01-PA 02-NPA	ARes = C CRes = C RRes = C PRes = C	Required if the AReq/CReq/RReq/ PReq is syntactically correct, but business processing cannot be performed for one of the reasons defined in Table A.5.

Data Element/ Field Name	Description	Source	Length/Format/ Values	Device Channel	Message Category	Message Inclusion	Conditional Inclusion
Invalid Request Detail Field Name: ireqDetail	Provides supporting detail, such as the specific data elements that caused the Invalid Request Code. Table A.5 defines standard contents.	ACS DS 3DS Server 3DS SDK	Length: Variable, maximum 2048 characters	01-APP 02-BRW	01-PA 02-NPA	ARes = C CRes = C RRes = C PRes = C	Required if the Invalid Request Code is present in the message and the Invalid Request Code alone does not provide sufficient detail to describe the error. Not present if there is no Invalid Request Detail associated with the Invalid Request Code in Table A.5.

Issuer Image Field name: 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. Three fully qualified URLs with small, medium and large images to be loaded and cached for use in the current challenge. SDK to select size appropriate for the current device screen resolution. Example: “issuerImage” = {“small”: “http://acs.com/small_image.jpg, ”medium”: “http://acs.com/medium_image.jpg”, “large”: http://acs.com/large_image.jpg } Option 2: May also be ‘none” if no image is to be displayed. Example: “issuerImage” = “none”	ACS	Length: Variable, maximum 2048 characters Format: JSON Object Value: Option 1: JSON Object, string, values as follows: “small”: Fully qualified URL of small image resource “medium” Fully qualified URL of medium image resource “large”– Fully qualified URL of large image resource Option 2: String containing the value “none”	01-APP	01-PA 02-NPA	CRes = C	Required on the initial CRes message from the ACS, omitted after. Conditional for Native UI.

Data Element/ Field Name	Description	Source	Length/Format/ Values	Device Channel	Message Category	Message Inclusion	Conditional Inclusion
Merchant Category Code Field Name: mcc	DS-specific code describing the Merchant's type of business product or service. The same value must be used in the authorisation request. Supported values are specified by each Payment System or DS.	3DS Server	Length: 4 characters Format: Number	01-APP 02-BRW	01-PA 02-NPA	01-PA: AReq = R 02-NPA: AReq = C	Required for 02-NPA if the merchant is also the 3DS Requestor.
Merchant Country Code Field Name: merchantCountryCode	Country Code of the Merchant. The same value must be used in the authorisation request. Must be the ISO 3166-1 numeric three-digit country code, other than those listed in Table A.6.	3DS Server	Length: 3 characters Format: Number	01-APP 02-BRW	01-PA 02-NPA	01-PA: AReq = R 02-NPA: AReq = C	Required for 02-NPA if the merchant is also the 3DS Requestor.
Merchant Name Field Name: merchantName	Merchant name assigned by the Acquirer or Payment System. Same name used in the authorisation message as defined in ISO 8583.	3DS Server	Length: Variable, maximum 40 characters Format: String.	01-APP 02-BRW	01-PA 02-NPA	01-PA: AReq = R 02-NPA: AReq = C	

Data Element/ Field Name	Description	Source	Length/Format/ Values	Device Channel	Message Category	Message Inclusion	Conditional Inclusion
Merchant Risk Indicator Field Name: merchantRiskIndicator	Merchant’s assessment of the level of fraud risk for the specific authentication for both the cardholder and the authentication being conducted. Refer to Table A.10 for data elements.	3DS Server	Length: Variable Format: JSON Object	01-APP 02-BRW	01-PA 02-NPA	AReq = O	Optional, but strongly recommended to include
Message Category Field Name: messageCategory	Identifies the category of the message for a specific use case.	3DS Server	Length: 2 characters Format: Number Values: 01 = PA 02 = NPA 80–99 = reserved for DS-specific use	01-APP 02-BRW	01-PA 02-NPA	AReq = R RReq = R	

Data Element/ Field Name	Description	Source	Length/Format/ Values	Device Channel	Message Category	Message Inclusion	Conditional Inclusion
Message Extension Field Name: messageExtension	Data necessary to support requirements not otherwise defined in the 3-D Secure message must be carried in a Message Extension.	3DS Server	Length: Variable, maximum 8192 bytes Format: JSON Array of objects	01-APP 02-BRW	01-PA 02-NPA	AReq = C ARes = C CReq = C CRes = C PReq = C PRes = C RReq = C RRes = C	Conditions to be set by each DS.

Data Element/ Field Name	Description	Source	Length/Format/ Values	Device Channel	Message Category	Message Inclusion	Conditional Inclusion
Message Type Field Name: messageType	Identifies the type of message that is being passed.	3DS Server 3DS SDK DS ACS	Length: 4 characters Format: String Values: AReq ARes CReq CRes PReq PRes RReq RRes Erro	01-APP 02-BRW	01-PA 02-NPA	AReq = R ARes = R CReq = R CRes = R PReq = R PRes = R RReq = R RRes = R Error = R	

Data Element/ Field Name	Description	Source	Length/Format/ Values	Device Channel	Message Category	Message Inclusion	Conditional Inclusion
Message Version Number Field Name: messageVersion	Specification version identifier This shall be the version number of the specification utilised by the system creating this message.	3DS Server DS ACS 3DS SDK	Length: 5 characters Format: String Value: n.n.n where: “n” represents a numeric digit that relates to the major and minor of the specification version number.	01-APP 02-BRW	01-PA 02-NPA	AReq = R ARes = R CReq = R CRes = R PReq = R PRes = R RReq = R RRes = R Error = R	
Notification URL Field Name: notificationURL	Fully qualified URL of the system that receives the CRes message. The CReq message is posted from the Cardholder browser at the end of the challenge and receipt of the RRes message.	ACS	Length: Variable, maximum 256 characters Format: String Value: Fully Qualified URL	02-BRW	01-PA 02-NPA	CReq = R	
OOB App URL Field Name: 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 Format: String Value: Fully Qualified URL	01-APP	01-PA 02-NPA	CRes = C	Required for ACS UI Type = 4 or 5

Data Element/ Field Name	Description	Source	Length/Format/ Values	Device Channel	Message Category	Message Inclusion	Conditional Inclusion
OOB App Label Field Name: 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 Format: String	01-APP	01-PA 02-NPA	CRes = C	
OOB Continuation Indicator Field Name: 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.	3DS SDK	Length: 1 character Format: Boolean Value: true	01-APP	01-PA 02-NPA	CReq = C	Required when ACS UI Type = 4 when the Cardholder has selected that option on the device.
OOB Continuation Label Field Name: 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 Format: String	01-APP	01-PA 02-NPA	CRes = C	Required when ACS UI Type = 4 in when the Cardholder has selected that option on the device.

Payment System Image Field Name: psImage	Sent in the initial CRes message from the ACS to the 3DS SDK to provide the URL(s) of the DS logo or image to be used in the Native UI. Option 1: Three fully qualified URLs with small, medium and large images to be loaded and cached for use in the current challenge. SDK to select size appropriate for the current device screen resolution. Example: “psImage” = {“small”: “http://ds.com/small_image.jpg, ”medium”: “http://ds.com/medium_image.jpg”, “large”: “http://ds.com/large_image.jpg” }	ACS	Length: Variable, maximum 2048 characters Format: JSON object Value: Option 1: JSON Object, string values as follows “small”: Fully qualified URL of small image resource “medium”: Fully qualified URL of medium image resource “large”: Fully qualified URL of large image resource Option 2: String containing the value “none”	01-APP	01-PA 02-NPA	CRes = C	Required on the intial CRes message from the ACS, omitted after. Conditional for ACS UI Type = 1–4.

Data Element/ Field Name	Description	Source	Length/Format/ Values	Device Channel	Message Category	Message Inclusion	Conditional Inclusion
	Option 2: May also be “none” if no image is to be displayed. Example: “psImage” = “none”						
Purchase Amount Field Name: 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. Example: If the purchase amount is USD 123.45, element will contain the value 12345.	3DS Server	Length: Variable, maximum 48 characters Format: Number	01-APP 02-BRW	01-PA 02-NPA	01-PA: AReq = R 02-NPA: AReq = C	Required for 02-NPA if Instalment, or Recurring transactions.
Purchase Currency Field Name: purchaseCurrency	Currency in which purchase amount is expressed. Must be an ISO 4217 three-digit currency code, other than those listed in Table A.6.	3DS Server	Length: 3 characters Format: Number	01-APP 02-BRW	01-PA 02-NPA	01-PA: AReq = R 02-NPA: AReq = C	Required for 02-NPA if Instalment, or Recurring transactions.

Data Element/ Field Name	Description	Source	Length/Format/ Values	Device Channel	Message Category	Message Inclusion	Conditional Inclusion
Purchase Currency Exponent Field Name: purchaseExponent	Minor units of currency as specified in the ISO 4217 currency exponent. Example: • USD = 2 • Yen = 0	3DS Server	Length: 1 character Format: Number	01-APP 02-BRW	01-PA 02-NPA	01-PA: AReq = R 02-NPA: AReq = C	Required for 02-NPA if Instalment, or Recurring transactions.
Purchase Date & Time Field Name: purchaseDate	Date and time of the purchase, expressed in GMT. For instalment and reoccurring purchases, this will be the time of the original cardholder purchase agreement.	3DS Server	Length: 17 characters Format: String Value: Must be in the format: YYYYMMDD: HH:MM:SS	01-APP 02-BRW	01-PA 02-NPA	01-PA: AReq = R 02-NPA: AReq = C	Required for 02-NPA if Instalment, or Recurring transactions
Recurring Expiry Field Name: recurringExpiry	Date after which no further authorisations shall be performed.	3DS Server	Length: 8 characters Format: Number Value: Must be in the format :YYYYMMDD	01-APP 02-BRW	01-PA 02-NPA	01-PA: AReq = C 02-NPA: AReq = C	Required if Instalment, or Recurring transactions

Data Element/ Field Name	Description	Source	Length/Format/ Values	Device Channel	Message Category	Message Inclusion	Conditional Inclusion
Recurring Frequency Field Name: recurringFrequency	Indicates the minimum number of days between authorisations.	3DS Server	Length: Variable, maximum 4 characters Format: Number	01-APP 02-BRW	01-PA 02-NPA	01-PA: AReq = C 02-NPA: AReq = C	Required for if Instalment, or Recurring transactions
Resend Challenge Information Code Field Name: resendChallenge	Indicator to the ACS to resend the challenge information code to the Cardholder.	3DS SDK	Length: 1 character Format: String Values: Y = Resend N = Do not Resend	01-APP	01-PA 02-NPA	CReq = C	Required if the Cardholder is requesting for the ACS to resend challenge information. Conditional for Native UI.
Resend Information Label Field Name: 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 Format: String	01-APP	01-PA 02-NPA	CRes = C	Required when the ACS UI Type = 1, 2, or 3

Data Element/ Field Name	Description	Source	Length/Format/ Values	Device Channel	Message Category	Message Inclusion	Conditional Inclusion
Results Message Status Field Name: 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: 1 character Format: Number Values: 0 = Results Request Received for further Processing 1 = Challenge Request not sent to ACS by 3DS Requestor (3DS Server or 3DS Requestor opted out of the challenge) 2 = ARes challenge data not passed delivered to the 3DS Requestor due to technical error	01-APP 02-BRW	01-PA 02-NPA	RRes = R	

Data Element/ Field Name	Description	Source	Length/Format/ Values	Device Channel	Message Category	Message Inclusion	Conditional Inclusion
SDK App ID Field Name: 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. Must be in the canonical format as defined in IETF RFC 4122. This may utilise any of the specified versions as long as the output meets specified requirements.	3DS SDK (sent via 3DS Server)	Length: 36 characters Format: String	01-APP	01-PA 02-NPA	AReq = R	
SDK Encrypted Data Field Name: sdkEncData	JWE Object as defined in Section 6.2.2.1 containing data encrypted by the SDK for the DS to decrypt. Note: Device Information is the only field encrypted in this version of the EMV 3-D Secure specification.	3DS SDK (sent via 3DS Server)	Length: Variable, maximum 15360 characters Format: String Value: JWE Object	01-APP	01-PA 02-NPA	AReq = C	Required 3DS Server to DS, but will not be present DS to ACS.

Data Element/ Field Name	Description	Source	Length/Format/ Values	Device Channel	Message Category	Message Inclusion	Conditional Inclusion
SDK Ephemeral Public Key (Qc) Field Name: sdkEphemPubKey (Signed Content)	Public key component of the ephemeral key pair (dc, Qc) generated by the 3DS SDK and used to establish session keys between the 3DS SDK and ACS. In cryptographic algorithms throughout this specification, this value is identified as “Qc”. Note: In the ARes, this will be contained within the ACS Signed Content JWS Object and will not be populated in the ARes in its own field. Refer to Annex C for additional detail.	3DS SDK (sent via 3DS Server)	Length: Variable, maximum 128 characters Format: String Value: Base64 Encoded binary key	01-APP	01-PA 02-NPA	AReq = R ARes = C	For the ARes, required if Transaction = C.
SDK Reference Number Field Name: 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 Format: String	01-APP	01-PA 02-NPA	AReq = R	

Data Element/ Field Name	Description	Source	Length/Format/ Values	Device Channel	Message Category	Message Inclusion	Conditional Inclusion
SDK Transaction ID Field Name: sdkTransID	Universally unique transaction identifier assigned by the 3DS SDK to identify a single transaction. Must be in the canonical format as defined in IETF RFC 4122. This may utilise any of the specified versions as long as the output meets specified requirements.	3DS SDK (sent via 3DS Server)	Length: 36 characters Format: String	01-APP	01-PA 02-NPA	AReq = R ARes = R CReq = R CRes = R	
Submit Authentication Label Field Name: submitAuthenticationLabel	Label to be used in the UI for the button that the user selects when they have completed the authentication. This is not used for OOB authentication.	ACS	Length: Variable maximum 45 characters Format: String	01-APP	01-PA 02-NPA	CRes = C	Required when the ACS UI Type = 1, 2, or 3

Transaction Status Field Name: transStatus	Indicates whether a transaction qualifies as an authenticated transaction. Note: The CRes message can contain only a value of Y or N. Note: If the IReq is included in the message, the Transaction Status must be U.	ACS	Length: 1 character Format: String Values: Y = Authentication Successful; All data needed for authorisation, including the Authentication Value, is included in the message for 01-PA N = Not Authenticated; Transaction denied U = Authentication Could Not Be Performed; Technical or other problem, as indicated in ARes or CRes A = Attempts Processing Performed; Not authenticated, but a proof of attempted authentication is provided. All data needed for authorisation	01-APP 02-BRW	01-PA 02-NPA	ARes = R RReq = R CRes = C	This data element only present in the final CRes message.

Data Element/ Field Name	Description	Source	Length/Format/ Values	Device Channel	Message Category	Message Inclusion	Conditional Inclusion
			including the Authentication Value is included in the message for 01-PA. C = Challenge Required; Additional authentication is required using the CReq/CRes. R = Authentication Rejected; Issuer is rejecting authentication and request that authorisation not be attempted.				

Transaction Status Reason Field Name: transStatusReason	Provides information on why the Transaction Status field has the specified value.	ACS	Length: 2 characters Format: Number Values (one of the following): 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	01-APP 02-BRW	01-PA 02-NPA	ARes = C RReq = C	For 01-PA, required if the Transaction Status field is set to either: N, U, or R. For 02-NPA, Conditional as defined by the DS.

Data Element/ Field Name	Description	Source	Length/Format/ Values	Device Channel	Message Category	Message Inclusion	Conditional Inclusion
			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 20–50 = EMVCo-defined values 51–99 = DS specific				

Data Element/ Field Name	Description	Source	Length/Format/ Values	Device Channel	Message Category	Message Inclusion	Conditional Inclusion
Transaction Type Field Name: transType	Identifies the type of transaction being authenticated. Values derived from the 8583 ISO Standard.	3DS Server	Length: 2 characters Format: Number Values: Values: 01 = Goods/ Service Purchase 03 = Check Acceptance 10 = Account Funding 11 = Quasi-Cash Transaction 28 = Prepaid Activation and Load	01-APP 02-BRW	01-PA	AReq = C	This field is required in some markets (e.g. for Merchants in Brazil). Otherwise, optional.
Why Information Label Field Name: whyInfoLabel1	Label to be displayed to the Cardholder for the "why" information section.	ACS	Length: Variable, maximum 45 characters Format: String	01-APP	01-PA 02-NPA	CRes = C	Required based upon the ACS UI format selected.

Data Element/ Field Name	Description	Source	Length/Format/ Values	Device Channel	Message Category	Message Inclusion	Conditional Inclusion
Why Information Text Field Name: whyInfoText1	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 Format: String	01-APP	01-PA 02-NPA	CRes = C	Required based upon the ACS UI format selected.

A.5 Detailed Field Values

The following sections provide additional details on the values for some data elements listed in Table A.1.

A.5.1 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* This data is placed into a JWE object that will encrypt the data using the DS public key.

A.5.2 Browser Information—02-BRW Only

Accurate browser information must be obtained in the AReq 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 obtained from the browser 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

Refer to Table A.1 for data element specifications.

A.5.3 3DS Method Data

The following table defines the data elements to be sent in the 3DS Method. The data is exchanged between the 3DS Requestor via the cardholder browser. 3DS Method Data applies to both 01-PA and 02-NPA.

Table A.2: 3DS Method Data

Data Element / Field Name	Description	Inclusion	Recipient
3DS Method Notification URL Field Name: threeDSMethodNotificationURL	The URL that will receive the notification of 3DS Method completion from the ACS. This is sent in the initial request to the ACS from the 3DS Requestor executing the 3DS Method.	R	ACS
3DS Server Transaction ID Field Name: threeDSServerTransId	A unique identifier for the transaction that will be the same as the 3DS Server Transaction ID in the AReq message, and will have the same format as specified in Table A.1. This will be sent to the ACS in the 3DS Method HTTP POST, and will be returned in the POST to the 3DS Method Notification URL.	R	ACS 3DS Requestor

A.5.4 Browser CReq and CRes POST

The following table defines the data elements to be sent in the Browser POST to the ACS for the CReq flow, and to the Notification URL in the CRes flow. A form is utilised within the cardholder browser and the data is sent via the cardholder browser in an HTTP POST.

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

Data Element / Field Name	Recipient	Description	Field Requirements	Inclusion
3DS Requestor Session Data Field Name: threeDSSessionData	ACS 3DS Requestor	3DS Requestor session data that must be returned to the 3DS Requestor by the ACS along with the CRes. Optionally used to accommodate the different ways 3DS Requestor systems handle session information. If the 3DS Requestor system can associate the final post with the original session without further assistance, the 3DS Requestor Session Data field may be missing. If the 3DS Requestor system does not maintain a session for a given authentication session, the 3DS Requestor Session Data field can carry any data the 3DS Requestor needs to continue the session. Because the content of this field varies by 3DS Requestor implementation, the ACS must preserve it unchanged and without assumptions about its content.	Length: Maximum 1024 Format: Alphanumeric The field must be Base64 encoded. The size of the field (after Base64 encoding, if applicable) is limited to 1024 bytes	O
CReq Field Name: creq	ACS	The entire CReq message as defined in Table B.3 that has been Base64 encoded.	Length: Variable length, Base64	R

Data Element / Field Name	Recipient	Description	Field Requirements	Inclusion
CRes Field name: cres	3DS Requestor	The entire CRes message as defined in Table B.4 that has been Base64 encoded.		R

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

Error messages are used when a system receives a message that cannot be processed to determine how to formulate a response, or when the error is required as part of receiving a response message from another system.

For example, a 3DS Server receives an ARes message from a DS that contains an error, and the 3DS Server responds with an Error message to the DS using the 3DS Server Transaction ID of the transaction that had an error.

The following table describes the defined values for Error Code and specifies the associated contents for Error Description and Error Detail, when applicable.

Note: Additional Error Code values may be defined at any time. All components must accept any value.

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

Value	Error Code	Error Description	Error Detail
02	Message element not a defined message	Message is not AReq, ARes, CReq, CRes, PReq, PRes, RReq, or RRes, OR Valid Message Type is sent to an inappropriate component (such as AReq message being sent to the 3DS Server).	Invalid Message OR Invalid Message Type

Value	Error Code	Error Description	Error Detail
03	Required element missing	A message element required as defined in Table A.1 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.
04	Data element not recognised	Message element not recognised.	Name of critical element(s) that was not recognised; if more than one element is detected, this is a comma delimited list.
05	Format of one or more elements is invalid according to the specification	Data element not in the required format. For example, not numeric or wrong length.	Name of invalid element(s); if more than one invalid element is detected, this is a comma delimited list.
06	Message version not supported	Message version received is not valid for the receiving component.	All supported versions in comma delimited list.
07	Duplicate Data Element	Valid data element present 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.
08	Transaction timed out	Timeout expiry reached for the transaction as defined in Section 5.5.	Message not received in allotted time.
50	Access denied, invalid endpoint	For example, Acquirer or Merchant not participating.	"Access denied, invalid endpoint".
98	Transient system failure	For example, a slowly processing back-end system.	Description of the failure.
99	Permanent system failure	For example, a critical database cannot be accessed.	Description of the failure.

A.5.6 Invalid Request Code, Invalid Request Detail

Invalid Request message elements are added to a response message when an issue arises from processing a request message and the receiving system is able to return the response. The following table describes the defined values for Invalid Request Code (IReqCode) and specifies the contents of Invalid Request Detail (IReqDetail), when applicable.

Table A.5: Invalid Request Detail Values

Value	IReq Code	IReq Description	IReq Detail
02	Message not defined	Message is not AReq, ARes, CReq, CRes, PReq, PRes, RReq, or RRes or, Error OR Valid Message Type is sent to an inappropriate component (such as AReq message being sent to the 3DS Server).	Invalid Message OR Invalid Message Type
03	Required element missing	A message element required as defined in Table A.1 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.
04	Data element not recognised	Message element not recognised	Name of critical element(s) that was not recognised; if more than one element is detected, this is a comma delimited list.
05	Format of one or more elements is invalid according to the specification	Data element not in the required format. For example, not numeric or wrong length.	Name of invalid element(s); if more than one element is detected, this is a comma delimited list.
06	Message version not supported	Message version received is not valid for the receiving component.	All supported versions in comma delimited list

Value	IReq Code	IReq Description	IReq Detail
07	Duplicate Data Element	Valid data element present 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.
08	Transaction ID Not Recognised	Transaction ID received is not valid for the receiving component.	The Transaction ID that was received and that is invalid
10	Unsupported device	Device type is not supported	Non Required
12	Data decryption failure	Data could not be decrypted by the DS due to technical or other reason	Description of the failure
13	Transaction Timed Out	Timeout expiry reached for the transaction as defined in Section 5.5.	"Message not received in allotted time"
50	Access denied, invalid endpoint	For example, Acquirer or Merchant not participating	"Access denied, invalid endpoint"
54	ISO code not valid	ISO code not valid per ISO tables (for either country or currency), or code is one of the excluded values listed in Table A.6.	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 IReqDetail.

Value	IReq Code	IReq Description	IReq Detail
56	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 CReq message was incorrectly sent: • CReq message was received by the wrong ACS, or • CReq message was not sent, based on the values in the ARes message, or • 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 comma-delimited list
59	Merchant Category Code (MCC) not valid for Payment System	Invalid MCC received in the AReq	None Required.
98	Transient system failure	For example, a slowly processing back-end system	Description of the failure
99	Permanent system failure	For example, a critical database cannot be accessed	Description of the failure

A.5.7 Excluded ISO Currency and Country Code Values

The following table lists exclusions from the ISO values for Currency Code (ISO 4217) and Country Code (ISO 3166).

Table A.6: Excluded Currency Code and Country Code Values

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

A.5.8 Card Range Data

The Card Range Data data element contains information returned to the 3DS Server from the DS in the PRes message 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.

Note: There may be as many JSON Objects as there are stored card ranges in the DS being called.

The detailed data elements are outlined in Table A.1.

Table A.7: Card Range Data

Data Element / Field Name	Description	Length/Format/Values	Inclusion
Start Card Range Field Name: startRange	Start of the card range.	Length: 13-19 characters Format: Number	R
End Card Range Field Name: endRange	End of the card range.	Length: 13-19 characters Format: Number	R
Protocol Version Field Name: protocolVersion	The most recent protocol version that is valid for the URL of the ACS that will be used by the 3DS Method.	Length: 5 characters Format: String Value: n.n.n where: “n” represents a numeric digits that relates to the major and minor digits of the Message Version Number	R.
3DS Method URL Field name: threeDSMethodURL	The URL of the ACS that will be used by the 3D Method. Note: The 3DSMethodURL data element may be omitted if not supported by the ACS this specific card range.	Length: Variable, Maximum 256 characters Format: String Value: Fully qualified URL	O

A.6 Message Extension Data

Message extensions are used to carry additional data that is not defined in the core specification of EMV 3-D Secure. The party defining the message extension shall define the format of the data. Examples of data to be sent via extensions:

- Data represented in JSON Objects
- Binary data
- Single data elements

Data shall be sent in the “Message Extension” field with the data populated within a JSON array. Multiple extensions represented as JSON Objects may be within the JSON array if required.

The specific elements that shall comprise the extension are:

- Extension name (name)
- Assigned extension group identifier (id)
- Criticality indicator (criticalityIndicator)
- Data (data)

All extensions carried in a 3-D Secure message shall be represented in the JSON format as follows:

```
“messageExtension”:
[
{“name”: “Extension Name”,
“id”: “Extension Identifier”,
“criticalityIndicator”: “Criticality Indicator”,
 “data”: “Extension Data Set”}
]
```

As an example (with multiple extensions defined):

```
“messageExtension”:
[
{“name”: “extensionField1”
“id”: “ID1”,
“criticalityIndicator”: “true”,
 “data”: “Value1”},
{“name”: “extensionField2”
“id”: “ID2”,
“criticalityIndicator”: “true”,
 “data”: “Value2”};
{“name”: “sharedData”
“id”: “ID3”,
“criticalityIndicator”: “false”,
 “data”: “IkpTT05EYXRhIjogew0KImRhdGExIjogInNvbWUgZGF0YSIsDQoiZGF0YTIiOiAic29tZSBvdGhlciBkYXRhIg0KfQ==”}
]
```

A.6.1 Message Extension Attributes

Table A.8: Message Extension Attributes

Attribute Name	Description	Length/Format	Inclusion
name	The name of the extension data set as defined by the extension owner.	Length: Variable, Maximum 64 characters Format: String	R
id	A unique identifier for the extension.	Length: Variable, Maximum 64 characters Format: String	R
criticalityIndicator	A Boolean value indicating whether the recipient must understand the contents of the extension to interpret the entire message Values are lowercase: • true • false To ensure interoperability, the sender of the message must include this attribute even when the value is ‘false’.	Length: 4-5 characters Format: Boolean	R
data	The data being carried in the extension.	Length: Variable, Maximum 8192 characters Format: String	R

A.6.2 Identification

Each message extension defined for use in 3-D Secure must have a unique identifier assigned. Examples of unique identifiers include:

- EMVCo-assigned IDs
- Object IDs (OID)
- Uniform Resource Identifiers (URI)
- DS-Assigned IDs

The party defining the message extension specifies the format of the identifier and the value.

A.6.3 Criticality

The data in a message extension may affect the meaning of the rest of the data such that the entire message can only be understood in the context of the extension data. When this occurs, the extension is deemed to be critical and the value of the criticality attribute must be ‘true’.

When an extension is critical, recipients of the message must recognise and be able to process the extension. If a 3-D Secure application other than the DS receives a message containing a critical extension that it does not recognise, it must treat the message as invalid.

Note: DS requirements for responding to an unrecognised critical Extension element are described in Table A.4.

When an extension is non-critical, recipients that cannot process the extension shall ignore the data and pass it to the destination system unaltered.

All Critical extensions shall be assigned by EMVCo.

A.7 3DS Requestor Risk Information

3DS Requestor Risk Information are specific data elements within the AReq message that the 3DS Requestor provides to the ACS in support of the ACS risk assessment. The data elements are optional in the AReq message, however the presence of the data elements in the AReq message will make the risk-based authentication more precise. By evaluating these data elements, the ACS has data available that can reduce the number of unnecessary challenges.

The data elements include the following types of information:

- **Cardholder Account—**Cardholders account at the 3DS Requestor (if cardholder is not a Guest)
- **Merchant Risk Indicator—**Purchase and its risk
- **3DS Requestor Authentication—**How the 3DS Requestor authenticated the cardholder

These data elements are included in Table A.1 as individual data elements with a JSON Object format. The following sections provide detailed information of each NVP data element.

A.7.1 Cardholder Account Information

Note: For Cardholder Account Information data elements used to define a time period, can be included as either the specific date, or an approximate indicator for when the action occurred. 3DS Requestors can use either format.

Table A.9: Cardholder Account Information

Data Element / Field Name	Description	Length/Format/Values
Cardholder Account Age Indicator Field Name: chAccAgeInd	Length of time that the cardholder has had the account with the 3DS Requestor.	Length: 1 character Format: Number Values: 0 = No account (guest check-out) 1 = Created during this transaction 2 = Less than 30 days 3 = 30–60 days 4 = More than 60 days

Data Element / Field Name	Description	Length/Format/Values
Cardholder Account Date Field name: chAccDate	Date that the cardholder opened the account with the 3DS Requestor.	Length: 8 characters Format: Number; YYYYMMDD
Cardholder Account Change Indicator Field Name: chAccChangeInd	Length of time since the cardholder's account information with the 3DS Requestor was last changed. Including Billing or Shipping address, new payment account, or new user(s) added.	Length: 1 character Format: Number Values: 1 = Changed during this transaction 2 = Less than 30 days 3 = 30–60 days 4 = More than 60 days
Cardholder Account Change Field Name: chAccChange	Date that the cardholder's account with the 3DS Requestor was last changed. Including Billing or Shipping address, new payment account, or new user(s) added.	Length: 8 characters Format: Number; YYYYMMDD
Cardholder Account Password Change Indicator Field Name: chAccPwChangeInd	Length of time since the cardholder's account with the 3DS Requestor had a password change or account reset.	Length: 1 character Format: Number Values: 0 = No change 1 = Changed during this transaction 2 = Less than 30 days 3 = 30–60 days 4 = More than 60 days

Data Element / Field Name	Description	Length/Format/Values
Cardholder Account Password Change Field Name: chAccPwChange	Date that cardholder’s account with the 3DS Requestor had a password change or account reset.	Length: 8 characters Format: Number; YYYYMMDD
Shipping Address Usage Indicator Field Name: shipAddressUsageInd	Indicates when the shipping address used for this transaction was first used with the 3DS Requestor.	Length: 1 character Format: Number Values: 1 = This transaction 2 = Less than 30 days 3 = 30–60 days 4 = More than 60 days
Shipping Address Usage Field Name: shipAddressUsage	Date when the shipping address used for this transaction was first used with the 3DS Requestor.	Length: 8 characters Format: Number; YYYYMMDD
Number of Transactions Day Field Name: txnActivityDay	Number of transactions (successful and abandoned) for this cardholder account with the 3DS Requestor across all payment accounts in the previous 24 hours.	Length: maximum 3 characters Format: Number
Number of Transactions Year Field Name: txnActivityYear	Number of transactions (successful and abandoned) for this cardholder account with the 3DS Requestor across all payment accounts in the previous year.	Length: maximum 3 characters Format: Number
Number of Provisioning Attempts Day Field Name: provisionAttemptsDay	Number of Add Card attempts in the last 24 hours.	Length: maximum 3 characters Format: Number

Data Element / Field Name	Description	Length/Format/Values
Cardholder Account Purchase Count Field Name: nbPurchaseAccount	Number of purchases with this cardholder account during the previous six months.	Length: maximum 4 characters Format: Number
Suspicious Account Activity Field Name: suspiciousAccActivity	Indicates whether the 3DS Requestor has experienced suspicious activity (including previous fraud) on the cardholder account.	Length: 1 character Format: Number Values: 0 = No suspicious activity has been observed 1 = Suspicious activity has been observed
Shipping Name Indicator Field Name: shipNameIndicator	Indicates if the Cardholder Name on the account is identical to the shipping Name used for this transaction.	Length: 1 character Format: Number Values: 0 = Account Name identical to shipping Name 1 = Account Name different than shipping Name
Payment Account Age Indicator Field Name: paymentAccInd	Indicates the length of time that the payment account was enrolled in the cardholder’s account with the 3DS Requestor.	Length: 1 character Format: Number Values: 0 = No account (guest check-out) 1 = During this transaction 2 = Less than 30 days 3 = 30–60 days 4 = More than 60 days

Data Element / Field Name	Description	Length/Format/Values
Payment Account Age Field Name: paymentAccAge	Date that the payment account was enrolled in the cardholder’s account with the 3DS Requestor.	Length: 8 characters Format: ; YYYYMMDD

A.7.2 Merchant Risk Indicator

The Merchant Risk Indicator contains information about the specific purchase by the Cardholder. The detailed data elements are outlined in Table A.10.

Table A.10: Merchant Risk Indicator

Data Element / Field Name	Description	Length/Format/Values
Shipping Indicator Field Name: shipIndicator	Indicates shipping method chosen for the transaction. Merchants must choose the Shipping Indicator code that most reasonably and fairly describes the cardholder's specific transaction, not their general business. If one or more items are included in the sale, use the Shipping Indicator code for the physical goods, or if all digital goods, use the Shipping Indicator code that describes the most expensive item.	Length: 2 characters Format: Number Values: 01 = Ship to cardholder's billing address (addrMatch = Y) 02 = Ship to another verified address on file with merchant (addrMatch = N) 03 = Ship to address that is different than the cardholder's billing address (addrMatch = N) 04 = "Ship to Store" / Pick-up at local store (Store address shall be populated in shipping address fields) 05 = Digital goods (includes online services, electronic gift cards and redemption codes) 06 = Travel and Event tickets, not shipped 07 = Other (for example, Gaming, digital services not shipped, emedia subscriptions, etc.)

Data Element / Field Name	Description	Length/Format/Values
Delivery Timeframe Field name: deliveryTimeframe	Indicates the merchandise delivery timeframe.	Length: 1 character Format: Number Values: 0 = Electronic Delivery 1 = Same day shipping 2 = Overnight shipping 3 = Two or more day shipping
Delivery Email Address Field Name: deliveryEmailAddress	For Electronic delivery, the email address to which the merchandise was delivered.	Length: maximum 254 characters Format: String
Reorder Items Indicator Field Name: reorderItemsInd	Indicates whether the cardholder is reordering previously purchased merchandise.	Length: 1 character Format: Number Values: 0 = First time ordered 1 = Reordered
Pre-Order Purchase Indicator Field Name: preOrderPurchaseInd	Indicates whether Cardholder is placing an order for merchandise with a future availability or release date.	Length: 1 character Format: Number Values: 0 = Merchandise available 1 = Future availability

Data Element / Field Name	Description	Length/Format/Values
Pre-Order Date Field Name: preOrderDate	For a pre-ordered purchase, the expected date that the merchandise will be available.	Length: 8 characters Format: Number; YYYYMMDD
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 Format: Number
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 Table A.6.	Length: 3 characters Format: Number
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 Format: Number

A.7.3 3DS Requestor Authentication Information

The 3DS Requestor Authentication Information contains information about how the cardholder authenticated themselves during login to their account with the 3DS Requestor. The detailed data elements are outlined in Table A.11.

Table A.11: 3DS Requestor Authentication Information

Data Element / Field Name	Description	Length / Format / Values
3DS Requestor Authentication Method Field Name: threeDSReqAuthMethod	Mechanism used by the Cardholder to authenticate to the 3DS Requestor.	Length: 2 character Format: Number Values: 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 FIDO Authenticator 05–80 = EMVCo defined values 80–99 = DS specific use
3DS Requestor Authentication Timestamp Field name: threeDSReqAuthTimestamp	Date and time in UTC of the cardholder authentication.	Length: 12 characters Format: Number; YYYYMMDDHHMM

Data Element / Field Name	Description	Length / Format / Values
3DS Requestor Authentication Data Field Name: threeDSReqAuthData	Data that documents and supports a specific authentication process.	Length: maximum 2048 bytes Format: Any 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 instance: For method 02, the field can carry generic 3DS Requestor authentication information For method 03, the data element can carry information about the provider of the federated ID and related information For method 04, the data element can carry the IDO attestation data (including the signature) In future versions of the specification, these details are expected to be included

Annex B Message Format

This annex provides the EMV 3-D Secure data elements and field names by Message Type. Refer to Table A.1 for Data Element specifications.

B.1 AReq Message Data Elements

Table A.1 outlines the default validation requirements for the AReq message. A specific DS may specify other DS validations or actions to meet requirements specific for that DS.

Table B.1: AReq Data Elements

Data Element	Field Name
3DS Requestor Authentication Information	threeDSRequestorAuthenticationInfo
3DS Requestor Challenge Indicator	threeDSRequestorChallengeInd
3DS Requestor ID	threeDSRequestorID
3DS Requestor Name	threeDSRequestorName
3DS Requestor Non-Payment Authentication Indicator	threeDSRequestorNPAInd
3DS Requestor URL	threeDSRequestorURL
3DS Server Reference Number	threeDSServerRefNumber
3DS Server Transaction ID	threeDSServerTransID
3DS Server URL	threeDSServerURL
Account Type	acctType
Acquirer BIN	acquirerBIN
Acquirer Merchant ID	acquirerMerchantID
Address Match Indicator	addrMatch
Browser Accept Headers	browserAcceptHeader
Browser IP Address	browserIP
Browser Java Enabled	browserJavaEnabled

Data Element	Field Name
Browser Language	browserLanguage
Browser Screen Color Depth	browserColorDepth
Browser Screen Height	browserScreenHeight
Browser Screen Width	browserScreenWidth
Browser Time Zone	browserTZ
Browser User-Agent	browserUserAgent
Card/Token Expiry Date	cardExpiryDate
Cardholder Account Information	acctInfo
Cardholder Account Number	acctNumber
Cardholder Account Identifier	acctID
Cardholder Billing Address City	billAddrCity
Cardholder Billing Address Country	billAddrCountry
Cardholder Billing Address Line 1	billAddrLine1
Cardholder Billing Address Line 2	billAddrLine2
Cardholder Billing Address Postal Code	billAddrPostCode
Cardholder Billing Address State	billAddrState
Cardholder Email Address	email
Cardholder Home Phone Number	homePhone
Cardholder Mobile Phone Number	mobilePhone
Cardholder Name	cardholderName
Cardholder Shipping Address City	shipAddrCity
Cardholder Shipping Address Country	shipAddrCountry
Cardholder Shipping Address Line 1	shipAddrLine1
Cardholder Shipping Address Line 2	shipAddrLine2
Cardholder Shipping Address Postal Code	shipAddrPostCode

Data Element	Field Name
Cardholder Shipping Address State	shipAddrState
Cardholder Work Phone Number	workPhone
Challenge Mandated Indicator	challengeMandated
Device Channel	deviceChannel
Device Information	deviceInfo
Device Rendering Options Supported	deviceRenderOptions
DS Reference Number	dsReferenceNumber
DS Transaction ID	dsTransID
DS URL	dsURL
EMV Payment Token Indicator	payTokenInd
Instalment Payment Data	purchaseInstalData
Merchant Category Code	mcc
Merchant Country Code	merchantCountryCode
Merchant Name	merchantName
Merchant Risk Indicator	merchantRiskIndicator
Message Category	messageCategory
Message Extension	messageExtension
Message Type	messageType
Message Version Number	messageVersion
Purchase Amount	purchaseAmount
Purchase Currency	purchaseCurrency
Purchase Currency Exponent	purchaseExponent
Purchase Date & Time	purchaseDate
Recurring Expiry	recurringExpiry
Recurring Frequency	recurringFrequency

Data Element	Field Name
SDK App ID	sdkAppID
SDK Encrypted Data	sdkEncData
SDK Ephemeral Public Key (Qc)	sdkEphemPubKey
SDK Reference Number	sdkReferenceNumber
SDK Transaction ID	sdkTransID
Transaction Type	transType

B.2 ARes Message Data Elements

Table A.1 outlines the default validation requirements for the ARes message. A specific DS may specify other DS validations or actions to meet requirements specific for that DS.

Table B.2: ARes Data Elements

Data Element	Field Name
3DS Server Transaction ID	threeDSServerTransID
ACS Ephemeral Public Key (QT)	acsEphemPubKey
ACS Transaction ID	acsTransID
ACS Reference Number	acsReferenceNumber
ACS Rendering Type	acsRenderingType
ACS Signed Content	acsSignedContent
ACS URL	acsURL
Authentication Type	authenticationType
Authentication Value	authenticationValue
Challenge Mandated Indicator	challengeMandated
DS Reference Number	dsReferenceNumber
DS Transaction ID	dsTransID
Electronic Commerce Indicator	eci

Data Element	Field Name
Invalid Request Code	ireqCode
Invalid Request Detail	ireqDetail
Message Extension	messageExtension
Message Type	messageType
Message Version Number	messageVersion
SDK Ephemeral Public Key (Qc)	sdkEphemPubKey
SDK Transaction ID	sdkTransID
Transaction Status	transStatus
Transaction Status Reason	transStatusReason

B.3 CReq Message Data Elements

Table A.1 outlines the default validation requirements for the CReq message.

Table B.3: CReq Data Elements

Data Element	Field Name
3DS Server Transaction ID	threeDSServerTransID
ACS Transaction ID	acsTransID
Challenge Cancelation Indicator	challengeCancel
Challenge Data Entry	challengeDataEntry
Challenge HTML Data Entry	challengeHTMLDataEntry
Challenge Window Size	challengeWindowSize
Message Extension	messageExtension
Message Type	messageType
Message Version Number	messageVersion
Notification URL	notificationURL

Data Element	Field Name
OOB Continuation Indicator	oobContinue
Resend Challenge Information Code	resendChallenge
SDK Transaction ID	sdkTransID

B.4 CRes Message Data Elements

Table A.1 outlines the default validation requirements for the CRes message.

Table B.4: CRes Data Elements

Data Element	Field Description
3DS Server Transaction ID	threeDSServerTransID
ACS Transaction ID	acsTransID
ACS HTML	acsHTML
ACS UI Type	uiType
Challenge Additional Information Text	challengeAddInfo
Challenge Completion Indicator	challengeCompletionInd
Challenge Information Header	challengeInfoHeader
Challenge Information Label	challengeInfoLabel
Challenge Information Text	challengeInfoText
Challenge Information Text Colour Indicator	challengeInfoTextColour
Challenge Selection Information	challengeSelectInfo
Expandable Information Label 1	expandInfoLabel1
Expandable Information Label 2	expandInfoLabel2
Expandable Information Text 1	expandInfoText1
Expandable Information Text 2	expandInfoText2
Invalid Request Code	ireqCode

Data Element	Field Description
Invalid Request Detail	ireqDetail
Issuer Image	issuerImage
Message Extension	messageExtension
Message Type	messageType
Message Version Number	messageVersion
OOB App URL	oobAppURL
OOB App Label	oobAppLabel
OOB Continuation Label	oobContinueLabel
Payment System Image	psImage
Resend Information Label	resendInformationLabel
SDK Transaction ID	sdkTransID
Submit Authentication Label	submitAuthenticationLabel
Transaction Status	transStatus
Why Information Label	whyInfoLabel1
Why Information Text	whyInfoText1

B.5 Final CRes Message Data Elements

Table B.5 provides the data elements for the final CRes message sent upon completion of the challenge from the ACS.

Table A.1 outlines the default validation requirements for the CRes message.

Table B.5: Final CRes Data Elements

Data Element	Field Description
3DS Server Transaction ID	threeDSServerTransID
ACS Transaction ID	acsTransID
Challenge Completion Indicator	challengeCompletionInd

Data Element	Field Description
Invalid Request Code	ireqCode
Invalid Request Detail	ireqDetail
Message Extension	messageExtension
Message Type	messageType
Message Version Number	messageVersion
SDK Transaction ID	sdkTransID
Transaction Status	transStatus

B.6 PReq Message Data Elements

Table A.1 outlines the default validation requirements for the PReq message.

Table B.6: PReq Data Elements

Data Element	Field Name
3DS Server Reference Number	threeDSServerRefNumber
3DS Server Transaction ID	threeDSServerTransID
Message Extension	messageExtension
Message Type	messageType
Message Version Number	messageVersion

B.7 PRes Message Data Elements

Table A.1 outlines the default validation requirements for the PRes message.

Table B.7: PRes Data Elements

Data Element	Field Name
3DS Server Transaction ID	threeDSServerTransID
Card Range Data	cardRangeData
Invalid Request Code	ireqCode
Invalid Request Detail	ireqDetail
Message Extension	messageExtension
Message Type	messageType
Message Version Number	messageVersion

B.8 RReq Message Data Elements

Table A.1 outlines the default validation requirements for the RReq message.

Table B.8: RReq Data Elements

Data Element	Field Name
3DS Server Transaction ID	threeDSServerTransID
ACS Transaction ID	acsTransID
ACS Rendering Type	acsRenderingType
Authentication Method	authenticationMethod
Authentication Type	authenticationType
Authentication Value	authenticationValue
Challenge Cancelation Indicator	challengeCancel
DS Transaction ID	dsTransID

Data Element	Field Name
Electronic Commerce Indicator	eci
Interaction Counter	interactionCounter
Message Category	messageCategory
Message Extension	messageExtension
Message Type	messageType
Message Version Number	messageVersion
Transaction Status	transStatus
Transaction Status Reason	transStatusReason

B.9 RRes Message Data Elements

Table A.1 outlines the default validation requirements for the RRes message.

Table B.9: RRes Data Elements

Data Element	Field Name
3DS Server Transaction ID	threeDSServerTransID
ACS Transaction ID	acsTransID
DS Transaction ID	dsTransID
Invalid Request Code	ireqCode
Invalid Request Detail	ireqDetail
Message Extension	messageExtension
Message Type	messageType
Message Version Number	messageVersion
Results Message Status	resultsStatus

B.10 Error Messages Data Elements

Table A.4 provides detailed information including Error Code values.

Table B.10: Error Message Data Elements

Data Element	Field Name
3DS Server Transaction ID	threeDSServerTransID
Error Code	errorCode
Error Description	errorDescription
Error Detail	errorDetail
Message Type	messageType
Message Version Number	messageVersion

Annex C Generate ECC Key Pair

The following method of elliptic curve key pair generation follows [ISO/IEC 15946-1].

In this method, a random number is obtained and tested to determine that it will produce a value of d (the private key) in the correct range ($1 < d < n$). If *d* is out-of-range, another random number is obtained (for example, the process is iterated until an acceptable value of *d* is obtained).

> **Note: The integers used in this function are large (32 or 66 bytes), which may mean they would be represented as strings of bytes (as in most cryptographic libraries) rather than built-in integers in an implementation.**

The following process or its equivalent may be used to generate an ECC key pair.

C.1 Input

Curve parameters (*p*, *a*, *b*, *G*, *n*, *h*):

C.2 Output

- status—Status returned from the key pair generation procedure. The status will indicate SUCCESS or an ERROR.
- (*d*, *Q*)—Generated private and public key
    - *d*, the generated private key, is an integer in the range [1, *n*–1]
    - *Q*, the generated public key, is the point on the specified curve

      If an error is encountered during the generation process, no values for *d* and *Q* should be returned

C.2.1 Process

1. *N* := len(*n*), the bit-length of *n*
2. *d* := StringToInteger(Random(*N*))
3. If ($d > n – 2$), then go to step 2
4. $d := d + 1$
5. *Q* := PointMultiply(*d*, *G*, *CurveID*)
6. If successful, return *SUCCESS*, *d*, and *Q*

   Else, return ERROR status.

Auxiliary functions used:

- StringToInteger(s) converts a string *s* of bits to a non negative integer
- Random(*c*) generates a string of *c* bits, where *c* is a positive integer

- PointMultiply() performs scalar multiplication

Annex D Approved Transport Layer Security Versions

For establishing the links secured by TLS, the version number shall be V1.2 or higher.

RSA keys shall be 2048 bits or longer.

ECC keys shall be 256 bits or longer.

Requirements and recommendations for the supported Cipher Suites are as follows:

D.1 TLS Cipher Suites 1.2

D.1.1 TLS Cipher Suites 1.2 Shall support:

- TLS_ECHDE_RSA_WITH_AES_128_GCM_SHA256
- TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256

D.1.2 TLS Cipher Suites 1.2 Should support:

- TLS_RSA_WITH_AES_128_GCM_SHA256
- TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256
- TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256

D.1.3 TLS Cipher Suites 1.2 May support:

- TLS_RSA_WITH_AES_256_GCM_SHA384
- TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384
- TLS_RSA_WITH_AES_256_CBC_SHA
- TLS_ECDHE_RSA_WITH_3DES_EDE_CBC_SHA
- TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA
- TLS_ECDHE_ECDSA_WITH_3DES_EDE_CBC_SHA
- TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA

D.1.4 Implementation Notes

RC4 is not an approved algorithm so any Cipher Suite incorporating RC4 shall be disabled.

Most implementations supporting TLS 1.1 also support TLS 1.2, thus only TLS 1.2 is approved.

The use of 3DES and SHA (SHA-1) are to be phased out. They are included here under “may be supported” for legacy reasons only and may be deprecated in future versions of this specification.

***** END OF DOCUMENT *****