EMV®
Level 3 (L3) Testing Framework

Common Syntax for Online Message Responses

Version 1.0
August 2023

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.

Revision Log – Version 1.0

Version	Date	Description
V1.0	August 2023	First version of the document

Contents

Tables

1 Executive Summary

This document, the EMV® Level 3 (L3) Testing Framework - Common Syntax for Online Message Responses defines a common syntax standard for online message responses. Network simulators using this standard will be able to recognize characteristics of incoming messages and generate appropriate outgoing messages based on the requirements of a particular Participant System.

1.1 Scope

This document defines both the XML file syntax and a set of defined pseudo functions. It is a companion document to the EMVCo Level 3 Testing Framework – Implementation Guidelines [L3FIG].

In scope:

- XML File Syntax that can be used by Participant Systems to provide specific online message response requirements to Network Simulators.

Not in scope:

- EMVCo L3 Test Tool qualification.
- Default standard online message response requirements following Participant System specifications.
- Default standard Network Simulator verifications defined by a Participant System - e.g., a participant system may or may not require ARQC verification and/or ARPC generation by default, etc.

1.2 Common Syntax for Online Message Responses File Format

This section describes the details of the .xml file format and requirements.

General requirements:

- The file extension and file name shall not be case sensitive.
- EMVCo reserves the right to add xml tags. To support backwards compatibility, the test tool shall be able to import an xml file which may include additional data and not reject the file.
- As per the XML syntax described in the w3schools, XML tags are case sensitive. The tag <Letter> is different from the tag <letter>. Opening and closing tags shall be written with the same case (e.g., <message>This is correct</message>).
- An XML Schema Definition (XSD) file is available from EMVCo for the xml file defined below. It is only used to help implement the format. Please contact EMVCo via the query system to request the files.
- All attributes and defined values used in the files are case-sensitive unless otherwise stated in this document or in the XSD.
- The order of the occurrence of the XML tags shall not be modified unless otherwise stated in this document or in the XSD (i.e., <xs:all>).
- Leading and trailing whitespace for a tag's value must be ignored during processing.
- The value may be empty as defined in the XSD.

For presence data, the following notation is used:

- M: Mandatory – shall always be present.
- C: Conditional – shall be present unless the condition is not met.
- O: Optional – may or may not be present.

For occurrence data, the following notation is used:

- 1..1: Element is mandatory and can only occur once.
- 0..1: Element is optional and if present, can only occur once.
- 1..n: Element is mandatory and can occur more than once.
- 0..n: Element is optional and can occur more than once.

Note: The occurrence rule is bound by its parent element. If the parent element is not present or empty, the occurrence rule for the child elements will not apply.

1.2.1 File processing requirements

The Network Simulator will process the Response Profiles in the order that they occur in the file. The Simulator will attempt to match the characteristics of the incoming message (for example, an authorization request) to the Matching Criteria in each profile. When it finds a match, it will use the matching profile's list of actions to generate the required response message. This may involve setting an individual field to a particular value, responding without that field, or using a pseudo-function to generate field data (for example, Issuer Authentication Data or an Issuer Script). There may be more than one action for a given profile. If no match is found a standard response is returned as defined by the Payment System.

1.2.2 Machine Readable Online Message Response File Format

Table 1: Machine Readable Online Message Response File Format

Field	Block/Tag/Attribute	M/O/C	Occurrence	Description
<?xml version="1.0" encoding="utf-8"?>	Tag	M	1..1	xml version and encoding - e.g., <?xml version="1.0" encoding="utf-8" standalone="no"?>
<OnlineMessageResponse>	Block-start	M	1..1	XML root node.
<Header>	Block-start	M	1..1	
PSI	Tag	M	1..1	Participant System Identifier - e.g., 00.
SpecVersion	Tag	M	1..1	Version of this document.
FileVersion	Tag	M	1..1	The <FileVersion> tag represents the version assigned by the PSI in order to track the changes based on the following format: N.N where N is a numeric value – e.g., 1.0.
Date-Time	Tag	M	1..1	Standard UTC timestamp in ISO-8601 format. Example "2022-01- 31T08:06:18Z".
Description	Tag	O	0..1	Free text.
</Header>	Block-end	M		
<ResponseProfilesList>	Block-start	M	1..1	List of all profiles needed by a participant system to support one or more L3 test plans.
<Profile>	Block-start	M	1..n	Structure of one profile.
Name	Attribute	O	0..1	Name of the profile.

Field	Block/Tag/Attribute	M/O/C	Occurrence	Description
<MatchingCriteriaList>	Block-start	M	1..1	List of the Criteria to select the profile.
Criteria	Tag	M	1..n	Definition of one criterion.
DataItem	Attribute	M	1..1	Protocol DataItem, this element shall follow the syntax defined by the EMV L3 Testing Framework Implementation Guidelines v1.2 in Annex B Tool Pass/Fail Automation Criteria - DataItem - e.g., NET.01?0.DE.022.
Operator	Attribute	M	1..1	Operator of the evaluation - e.g., equals, like, exist, not exist, etc. These Operators shall follow the syntax defined by the EMV L3 Testing Framework Implementation Guidelines v1.2 – in section 4.3 Tool Pass/Fail Automation Criteria. An additional Operator: InList may be used to define a Criteria. The usage is defined in the value attribute.
Value	Attribute	C	0..1	Expected value (the element depends on the Operator). The value shall follow the syntax defined by the EMV L3 Testing Framework Implementation Guidelines v1.2 in section 4.3 Tool Pass/Fail Automation Criteria. **Note:** if using the STRING format, then the double quotes must not be included – e.g., STRING(EMVCo). If using the InList Operator, then the Criteria would be met if the DataItem is contained in the list of values separated by commas. The InList operator is restricted to the NUMBER() format. For example, if NET.01?0.DE.004 InList [NUMBER (000000000125),NUMBER(000000000123)], then: - NET.01?0.DE.004 = 000000000123 would be true - NET.01?0.DE.004 = 000000000125 would be true - NET.01?0.DE.004 = 000200000000 would be false
</MatchingCriteriaList>	Block-end	M		
<ResponseList>	Block-start	M	1..1	List of the responses which shall be executed by the simulator, if and only if all the Criteria in the <MatchingCriteria> are true.
Response	Tag	M	1..n	Definition of one response.

Field	Block/Tag/Attribute	M/O/C	Occurrence	Description
Action	Attribute	M	1..1	Define the action to execute during the profile: - set: set a value in a DataItem - remove: remove a DataItem in a response - Pseudo-function with parameters which define the action
DataItem	Attribute	C	0..1	If the action is “set” or “remove”, this element specifies which DataItem is impacted.
Value	Attribute	C	0..1	If the action is “set”, this element describes the value be pushed in the DataItem.
</ResponseList>	Block-end	M		
</Profile>	Block-end	M		
</ResponseProfilesList>	Block-end	M		
<SymmetricKeysList>	Block-start	C	0..1	List of the Card Symmetric Keys value used in the list of profiles, this element is mandatory if any profile in the list of profiles contains some crypto Validation/Generation.
<SymmetricKey>	Block-start	M	1..n	Definition of a Symmetric Key value.
KeySetName	Attribute	M	1..1	Name of the KeySet.
Key	Tag	M	1..n	Key value(s).
KeyType	Attribute	M	1..1	- MDK is a Master Derivation Key - UDK is a Unique Derivation Key
KeyName	Attribute	M	1..1	Key name.
</SymmetricKey>	Block-end	M		
</SymmetricKeysList>	Block-end	C		
</OnlineMessageResponse>	Block-end	M		

2 Pseudo-function definitions

Table 2: Pseudo-function definitions

Pseudo-function	Description	Parameters
emvsim.pin_validation(*pin*)	Method to validate the PIN value sent by the system under test to the Network simulator. **Note:** The PIN key and format used for this operation is system dependent and it should be set in the Network Simulator engine context.	- pin (mandatory) Value for the expected value of the PIN.
emvsim.arqc_validation(*cvn*, *key*)	Method to validate the ARQC value sent by the system under test to the Network simulator.	- cvn (mandatory) CVN (1 Hex byte) to use to validate the ARQC. **Note:** The value can be set to AUTO, in this case, the simulator shall use the CVN present in the network message. - key (mandatory) KeyName used by the card to generate the ARQC & ARPC. The list of keys available is define in SymmetricKeysList section of the Profiles files (XML). **Note:** The Key can either be a UDK or MDK.

Pseudo-function	Description	Parameters
emvsim.arpcgeneration(*cvn, key, arc*)	Method to generate the Issuer Authentication Data (i.e., ARPC+CSU/ARC) value inside the Network simulator.	- cvn (mandatory) CVN to use to generate the ARPC. ***Note :*** The value can be set to AUTO, in this case, the simulator should use the CVN present in the network message. - key (mandatory) KeyName used by the card to generate the ARQC & ARPC. The list of keys available is define in SymmetricKeysList section of the Profiles files (XML). - arc (mandatory) ARC (2 Hex bytes) or CSU (4 Hex bytes) to use to generate the ARPC. **Note:** The value can be set to AUTO, in this case, the simulator shall generate default ARC/CSU to only indicate approval or decline of the transaction based on participant system requirements.
emvsim.issuerscript(*cvn, key, script*)	Method to generate the IssuerScript value inside the Network simulator.	- cvn (mandatory) CVN to use to generate the MAC. **Note:** The value can be set to AUTO, in this case, the simulator should use the CVN present in the network message. - key (mandatory) KeySetName used by the card to generate the MAC. The list of keys available is define in SymmetricKeysList section of the Profiles files (XML).The KeySetName needs to include the following KeyName: • AC_KEY to calculate/validate the ARQC & ARPC • MAC_KEY to calculate the MAC • ENC_KEY to encrypt blocks of data such as PIN Block for a PIN change script - script (mandatory) List of script commands in the order required. The format will be the same as the data which is expected to be generated, but with clear data for the encrypted data (e.g., PIN block) and a placeholder for the MACs consisting of MMs. It may include the Script ID (tag 9F18). The

Pseudo-function	Description	Parameters
		format of an Issuer Script is defined in EMV Book 3, "Issuer-to-Card Script Processing". Where a PIN block is required to be encrypted for a PIN Change command, it shall be represented as the unencrypted PIN block defined in EMV Book 3, Verify command. For example (PIN Change to “1234” with an 8-byte MAC): "71 1E 9F 18 04 41 42 43 44 86 15 84 24 00 02 10 24 12 34 FF FF FF FF FF MM MM MM MM MM MM MM MM", where MM must be replaced by the MAC value.
emvsim.do_not_respond()	Method to stop the process without response from the Network simulator.	
emvsim.delay_response(Time)	The method to delay the response from the network simulator.	Time - Indicates the number of seconds the response shall be delayed.

3 Terminology

Below is a working glossary of commonly used terms and acronyms:

A	
ARQC	Authorization Request Cryptogram.
ARPC	Authorization Response Cryptogram.
C	
CVN	Cryptogram Version Number.
E	
EMV	A trademark owned by EMVCo, referring to the technical specifications published by EMVCo.
EMVCo LLC (EMVCo)	The organization that manages the EMV Specifications and their related testing processes.
EMV Specifications	Technical specifications developed and maintained by EMVCo to facilitate worldwide interoperability and acceptance of secure payment transactions, including the requirements described in the bulletins available on the EMVCo website.
H	
Hex	Hexadecimal.
I	
ISO	International Organization for Standardization.
L	
L3TG	Level 3 Testing Group.
M	
MDK	Master Derivation Key.
P	
Participant System	An entity (e.g., domestic payment systems, global payment systems and other similar entities) that has been assigned a L3 Participant System Identifier (PSI) through EMV's registration service to enable the usage of the EMV L3 Testing Framework.
Pass Criteria	A Test Plan-defined description of an expected result for a successful outcome or conclusion of a test case.
PS	Participant System.
PSI	Participant System Identifier.
T	
Test Tool Qualification	The process undertaken by each Participant System to provide themselves, tool vendors and clients with a level of assurance that the tools being used by clients to execute terminal integration

	testing, will do so in compliance with Participant System requirements.
U	
UDK	Unique Derivation Key.
X	
XML	Extensible Mark-up Language (XML) is a markup language that defines a set of rules for encoding documents in a format that is both human-readable and machine-readable.
XSD	XML Schema Definition.

***** END OF DOCUMENT *****