# Reference manual for MERCHANT SYSTEM API integration

Release n. 2.3.2
**June 2020**

# 1. Summary

## 1.1 Summary

## 1.2 Table of schemes

## 1.3 Table of pictures

## 1.4 Revisions

| Date | Changes | Version |
|---|---|---|
| 2014-05-27 | First draft of the English version. | 1.2.4 |
| 2015-07-09 | Second draft of the English version. | 1.3.5 |
| 2017-07-12 | Update to version 1.6.4 and integrations | 1.6.5 |
| 2018-03-23 | Restyling. Missing Scheme added. RBA Messages. Minor fixes. | 1.7.0 |
| 2018-05-10 | Restyling. Added option and exponent on every message. Added name, surname, antifraud and product reference on authorization messages. Added description for some optional fields in the responses. | 1.7.1 |
| 2018-06-01 | API: added cardtype in the auth element; added taxid in auth msgs; fixed description for result code 38.<br>Redirect: added taxid and antifraud. In response added acquirerbin and merchantid (like api; with option) and cardtype (with service).<br>General: added iban network and token pan alias. | 1.8.0 |
| 2018-06-15 | API: fixed description for SENDMAIL on CREATELINK message. | 1.8.1 |
| 2018-12-17 | API: request of order status – Added pan alias section in response if the shop is enabled | 1.8.2 |
| 2019-02-08 | API: new 3DS 2.x API messages | 1.9.0 |
| 2019-05-27 | 5.2.1 Redirect: added 3DSDATA field<br>6.2.1 MAC: added 3DSDATA field<br>6.5 3DSData:<br>- added inclusion column for Redirect messages.<br>- added threeDSRequestorChallengeInd field | 1.9.1 |
| 2019-05-31 | Rebranding | 1.9.2 |
| 2019-06-07 | - IBANAuthorization and IbanCode<br>- Renaming of 3DS elements in ThreeDS and further modifications in 3DS2.x operations.<br>- URLMERCHANT to MERCHANTURL | 1.9.3 |
| 2019-09-30 | Guide separation in redirect and api.<br>Removed the following obsolete API:<br>DEFERREDAUTHORIZATION<br>RBAVERIFY<br>RBADECODEPARES<br>RBAAUTHORIZATION<br>DEFERREDREQUEST<br>DEFERREDCLOSING<br>SPLIT<br>VERIFY<br>Added new types in result lists.<br>Changed MAC for Authorizations (optionals)<br>New fields for 3ds2.0 authorizations | 2.0.0 |
| 2019-12-11 | 3.6.1, 3.6.2, 3.6.3:<br>- Added field CardholderInfo to response<br>- Added further details on 3DS2.x message exchange<br>4.2.9: Changed MAC for Authorization: CardholderInfo added<br>4.2.30: added new section<br>4.2.31: added new section | 2.0.1 |

| | | |
|---|---|---|
| 2020-01-07 | MAC minor fixes:<br>- MAC for ACCOUNTING: removed "if present" for the amount field;<br>- MAC for LISTAUTHORIZATION: removed opdescr field;<br>- MAC for AUTHORIZATION3DSSTEP1: added “if present” for the cvv2 field;<br>- MAC for THREEDSAUTHORIZATION0: added “if present” for the cvv2 field;<br>- MAC for CREATELINK: added “if present” for the linkemail field.<br><br>Updated the correct limits 8-18 for the OPERATORID field in every message. | 2.0.2 |
| 2020-02-26 | 3.6.1<br>- 3DS 2.x Authorization Request Step 0: if the 3DS Method completes within 10 seconds, the Requestor must be set to “Y”.<br>- Updated field NAMECH to optional<br>4.4 Removed mandatory and conditional fields within ThreeDSData object | 2.1.0 |
| 2020-03-31 | 3.3.1: updated result code table<br>3.6.1: updated returned result codes<br>3.6.2: updated returned result codes<br>3.11.1: update field LINKSURNAME length | 2.1.1 |
| 2020-04-10 | 3.11.1: added THREEDSDATA field to CREATELINK operation<br>3.4.1: added optional field InstallmentsNumber to AUTHORIZATION operation<br>3.5.1: added optional field InstallmentsNumber to AUTHORIZATION3DSSTEP1 operation<br>3.6.1:<br>- added optional field InstallmentsNumber to THREEDSAUTHORIZATION0 operation<br>- updated cross-reference to chapter 4.4<br>4.2.10: MAC for AUTHORIZATION: added “if present” for the field INSTALLMENTSNUMBER<br>4.2.11: MAC for AUTHORIZATION3DSSTEP1: added “if present” for the field INSTALLMENTSNUMBER<br>4.2.13: MAC for THREEDSAUTHORIZATION0: added “if present” for the field INSTALLMENTSNUMBER<br>4.2.23: MAC for CREATELINK: added “if present” for the fields THREEDSDATA | 2.2.0 |
| 2020-04-30 | New URLS for UTF-8 calls<br>New PanAliasData unified section<br>Update simple references with cross references<br>Satispay network<br>Recurring payments<br>General revision<br>MerchantID with SVAH | 2.3.0 |
| 2020-05-31 | Updated description for AMOUNT and ACCOUNTINGMODE fields and revisited paragraph 4.3.2 for ASI card verification transactions.<br>PayByLink with recurring payments. | 2.3.2 |

## 1.5 Glossary

| | |
|---|---|
| Back office | Used for making reference to the management functions of a store: statements, lists, queries, instructions, etc. |
| CC | Credit Card |
| Booking | Transaction creating the accounting effects of a previously authorized transaction |
| Credit | Accounting transaction for the repayment of a monetary sum to a customer |
| GET | HTTP protocol communication transaction |
| Hash | All the N bits (i.e. 128, 160) obtained from a string through a mathematical process in a way that a different result is invariably obtained from a different string |
| HTTP | Application protocol used for transmitting web pages. Standard RFC 2068 |
| MAC | Message authentication code |
| MD5 | Algorithm for generating a unique 16 byte message identifier. Defined in RFC 1321 |
| Merchant system | Virtual store management software system. Virtual store |
| SIA | FrontEnd Processor: SIA Spa |
| POST | HTTP protocol communication transaction |
| SHA-1 | Secure Hash Algorithm. Algorithm for generating hashes. Standard NIST FIPS 180-1 |
| Split | Transaction for subdividing/reducing a payment already effected. |
| SSL | Secure Socket Layer standard transport protocol created by Netscape Communication |
| Reversal | Transaction for the cancellation of a granted authorization with repayment of the sum and/or limit of expenditure to the card holder |
| URL | Universal resource locator |
| VBV | Verified By Visa, Visa security system for authenticating credit card holders during their purchases online |
| SecureCode | Security system for authenticating Mastercard and Maestro credit card holders during their purchases online (equivalent to VBV) |
| SafeKey | Security system for authenticating AMEX credit card holders during their purchases online (equivalent to VBV) |

# 2 Introduction

This document contains important technical information for virtual store designers who wish to integrate their website with the SIA VPOS service. This manual, therefore, is addressed strictly to technical personnel.  It does not contain an actual description of the SIA VPOS service, which, on the other hand, is provided by the appropriate documents.

This document provides a description of the **API Internet interface** of the SIA VPOS system and of the related integration with the order management systems on the merchant side.

SIA VPOS is an Internet virtual POS provided directly by SIA to sellers. It enables merchants to carry out transactions online with their credit card using a PC and an Internet connection. The system can be used both to substitute the physical "box" of the traditional POS and as a customizable gateway for credit card transactions. For a general description of its functionalities, see the related document.
The SIA VPOS service is complemented by the functionalities of a back office graphic interface.

As regards the security of the Internet communication route, the degree of reliability offered is equivalent to that of the TLS 1.2 protocol with 256-bit encryption.

For backoffice and redirect integration see "Merchant Integration VPOS REDIRECT".

# 3 API SIA VPOS Integration

## 3.1 Integration

Integration shall refer to the use by a software application of the functionalities offered by the SIA VPOS system in the form of APIs.

**URLs of the web APIs for the ISO-8859-1 name-value pair requests:**

TEST environment: https://atpostest.ssb.it/atpos/apibo/apiBO.app
PRODUCTION environment: https://atpos.ssb.it/atpos/apibo/apiBO.app

**URLs of the web APIs for the ISO-8859-1 XML requests:**

TEST environment: https://atpostest.ssb.it/atpos/apibo/apiBOXML.app
PRODUCTION environment: https://atpos.ssb.it/atpos/apibo/apiBOXML.app

**URLs of the web APIs for the UTF-8 name-value pair requests:**

TEST environment: https://virtualpostest.sia.eu/vpos/apibo/apiBO-UTF8.app
PRODUCTION environment: https://virtualpos.sia.eu/vpos/apibo/apiBO-UTF8.app

**URLs of the web APIs for the UTF-8 XML requests:**

TEST environment: https://virtualpostest.sia.eu/vpos/apibo/apiBOXML-UTF8.app
PRODUCTION environment: https://virtualpos.sia.eu/vpos/apibo/apiBOXML-UTF8.app

## 3.2 API SIA VPOS

This chapter provides a description of the procedures for integrating an application with the API system of the SIA VPOS payment service. The use of the API SIA VPOS is entirely optional.

**API Authorization:** After having authenticated himself through userid and password, the back office administrator may request authorization to use the SIA VPOS APIs by accessing the store profile, in particular, the store details.

The API is provided in the form of a web application accepting calls in POST HTTP format generated by a merchant application. Using this mechanism, the following transactions can be carried out:

- authorization request,
- reversal of payment,
- booking of authorized transaction,
- check status of transaction,
- questioning of transactions carried out by a merchant in a given period,
- etc.

Version no 1.4.0 of the SIA VPOS API also permits to send requests in XML format. These must be sent via the following means:
POST with a parameter called `data` filled in with the XML message in urlencoded format.

The *fields of the request messages* and the corresponding *XML routes* for making requests in XML format are set out in the following pages. The tag values indicated in said routes are provided strictly by way of example, except for the tag value <Operation> which must be identical to that indicated.
Regardless of which of the two formats is used for the request, the *reply will invariably be in XML*.
As regards the security of the Internet communication route, the degree of reliability offered is equivalent to that of the TLS 1.2 protocol with 256-bit encryption, which is considered a "*strong encryption*".
The communication protocol to be used by merchant applications for interfacing with the system is illustrated below. In particular, the series of steps that need to be taken for integration purposes is outlined below, including a list of the messages that are exchanged between the SIA VPOS system and the merchant systems.
Integration is understood as a mechanism enabling an individual's application to use the web API made available by SIA. It is probable that such application will be able to dialogue with the order management system and to collect data from the latter for the purpose of carrying out the transactions, as well as to update it with the results received online.

The functionalities made available to the merchant systems are the following:

| Function | Description |
|---|---|
| Online authorization request | It permits to forward authorizations to the authorization circuits. |
| Request of payment reversal | The reversal request is applied by the SIA VPOS system to a payment (authorization), regardless of its state. |
| Booking request | It permits to forward requests to SIA VPOS for the booking of a credit card authorization previously granted through a deferred booking. |
| Cancellation of booking request | It cancels a booking request and enables the credit card authorization to be booked again. |
| List of accounting transactions | It obtains the list of accounting transactions.<br>It contains those requested and those already sent to the acquirers, differentiated by state. |
| List of authorization requests | The following authorization requests forwarded to the system are displayed:<br>1. those with positive outcome<br>2. those with negative outcome<br>3. reversed authorizations<br>4. all of them |
| Order status request | It returns the current status of an order including all the authorization operations connected thereto. |

**The process** to be followed for **a request transaction** is schematically explained below:

1. the merchant system retrieves from its database all the information necessary to perform the transaction, for example: transaction ID, amount, etc.
2. the merchant system formats an http message containing all the fields specified as compulsory for the desired transaction, and sends it via GET or POST to SIA. The message also contains the authentication information;
3. the SIA SSL server processes the request data, transmits them to the legacy systems and replies with an xml document;
4. the merchant system processes the outcome message and, if necessary, updates its database.

The full lists of the fields of the various messages are provided in the related dedicated paragraphs.

The reply messages to the requests are formatted in XML. These contain all the request and reply data.

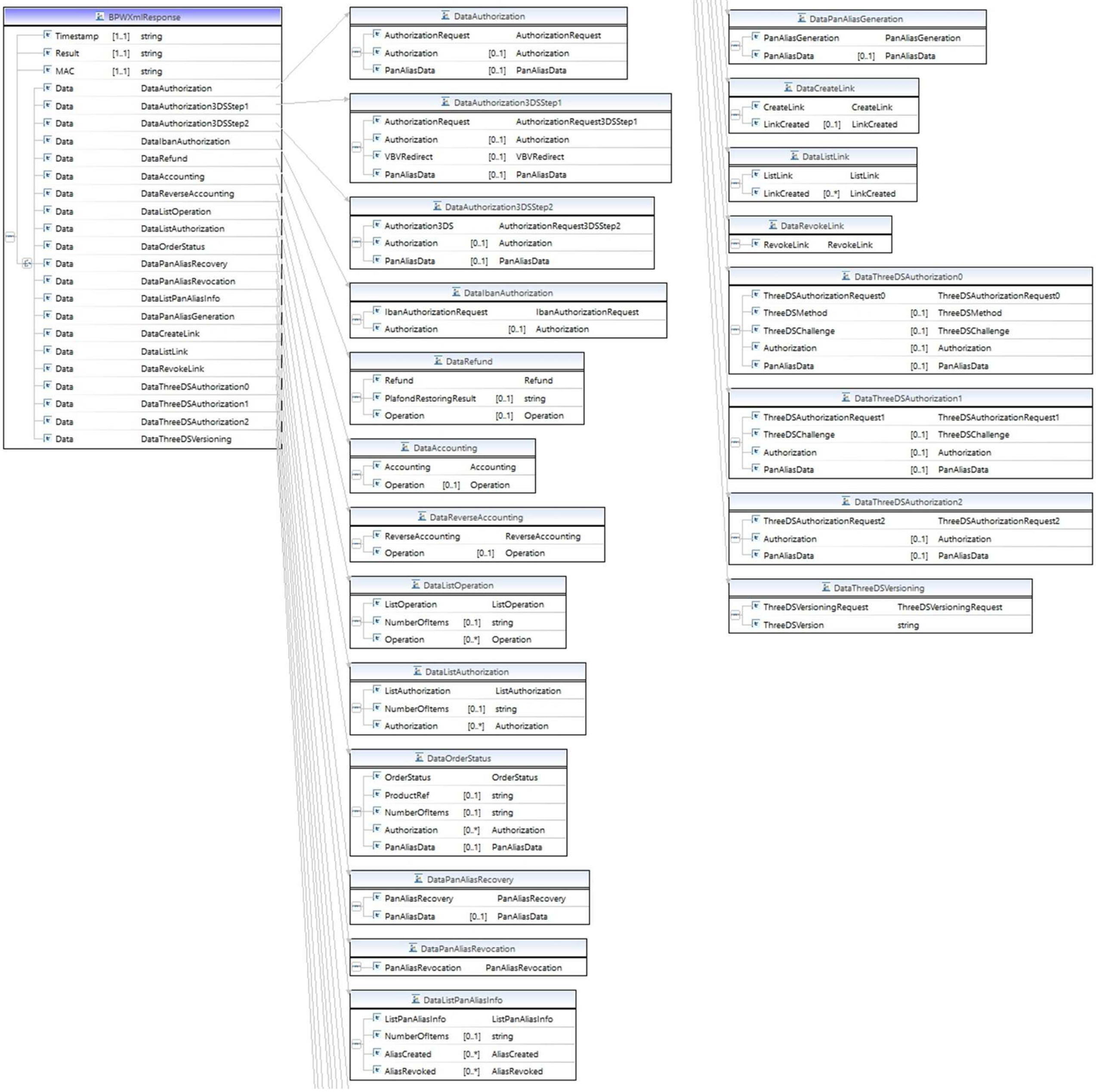
The next chapter contains a description of the format used for replies; the chapters following that provide a detailed description of the various functions available, grouped as follows:

- Simple authorizations, 3ds1.0 authorizations; 3ds2.0 authrizations.
- Operations on authorizations: capture, refund.
- Consultations

- Operations on pan aliases
- Transactions by link.

## 3.3 Response messages in XML

This chapter provides a general description of the XML format used for sending reply messages as well as a detailed explanation of the common elements that most of these messages share.
An overview of the XML format referred to above is provided in the following diagram.



**BPWXmlResponse**

| | | |
|---|---|---|
| Timestamp | [1..1] | string |
| Result | [1..1] | string |
| MAC | [1..1] | string |
| Data | | DataAuthorization |
| Data | | DataAuthorization3DSStep1 |
| Data | | DataAuthorization3DSStep2 |
| Data | | DataIbanAuthorization |
| Data | | DataRefund |
| Data | | DataAccounting |
| Data | | DataReverseAccounting |
| Data | | DataListOperation |
| Data | | DataListAuthorization |
| Data | | DataOrderStatus |
| Data | | DataPanAliasRecovery |
| Data | | DataPanAliasRevocation |
| Data | | DataListPanAliasInfo |
| Data | | DataPanAliasGeneration |
| Data | | DataCreateLink |
| Data | | DataListLink |
| Data | | DataRevokeLink |
| Data | | DataThreeDSAuthorization0 |
| Data | | DataThreeDSAuthorization1 |
| Data | | DataThreeDSAuthorization2 |
| Data | | DataThreeDSVersioning |

**DataAuthorization**

| | | |
|---|---|---|
| AuthorizationRequest | | AuthorizationRequest |
| Authorization | [0..1] | Authorization |
| PanAliasData | [0..1] | PanAliasData |

**DataAuthorization3DSStep1**

| | | |
|---|---|---|
| AuthorizationRequest | | AuthorizationRequest3DSStep1 |
| Authorization | [0..1] | Authorization |
| VBVRedirect | [0..1] | VBVRedirect |
| PanAliasData | [0..1] | PanAliasData |

**DataAuthorization3DSStep2**

| | | |
|---|---|---|
| Authorization3DS | | AuthorizationRequest3DSStep2 |
| Authorization | [0..1] | Authorization |
| PanAliasData | [0..1] | PanAliasData |

**DataIbanAuthorization**

| | | |
|---|---|---|
| IbanAuthorizationRequest | | IbanAuthorizationRequest |
| Authorization | [0..1] | Authorization |

**DataRefund**

| | | |
|---|---|---|
| Refund | | Refund |
| PlafondRestoringResult | [0..1] | string |
| Operation | [0..1] | Operation |

**DataAccounting**

| | | |
|---|---|---|
| Accounting | | Accounting |
| Operation | [0..1] | Operation |

**DataReverseAccounting**

| | | |
|---|---|---|
| ReverseAccounting | | ReverseAccounting |
| Operation | [0..1] | Operation |

**DataListOperation**

| | | |
|---|---|---|
| ListOperation | | ListOperation |
| NumberOfItems | [0..1] | string |
| Operation | [0..*] | Operation |

**DataListAuthorization**

| | | |
|---|---|---|
| ListAuthorization | | ListAuthorization |
| NumberOfItems | [0..1] | string |
| Authorization | [0..*] | Authorization |

**DataOrderStatus**

| | | |
|---|---|---|
| OrderStatus | | OrderStatus |
| ProductRef | [0..1] | string |
| NumberOfItems | [0..1] | string |
| Authorization | [0..*] | Authorization |
| PanAliasData | [0..1] | PanAliasData |

**DataPanAliasRecovery**

| | | |
|---|---|---|
| PanAliasRecovery | | PanAliasRecovery |
| PanAliasData | [0..1] | PanAliasData |

**DataPanAliasRevocation**

| | | |
|---|---|---|
| PanAliasRevocation | | PanAliasRevocation |

**DataListPanAliasInfo**

| | | |
|---|---|---|
| ListPanAliasInfo | | ListPanAliasInfo |
| NumberOfItems | [0..1] | string |
| AliasCreated | [0..*] | AliasCreated |
| AliasRevoked | [0..*] | AliasRevoked |

**DataPanAliasGeneration**

| | | |
|---|---|---|
| PanAliasGeneration | | PanAliasGeneration |
| PanAliasData | [0..1] | PanAliasData |

**DataCreateLink**

| | | |
|---|---|---|
| CreateLink | | CreateLink |
| LinkCreated | [0..1] | LinkCreated |

**DataListLink**

| | | |
|---|---|---|
| ListLink | | ListLink |
| LinkCreated | [0..*] | LinkCreated |

**DataRevokeLink**

| | | |
|---|---|---|
| RevokeLink | | RevokeLink |

**DataThreeDSAuthorization0**

| | | |
|---|---|---|
| ThreeDSAuthorizationRequest0 | | ThreeDSAuthorizationRequest0 |
| ThreeDSMethod | [0..1] | ThreeDSMethod |
| ThreeDSChallenge | [0..1] | ThreeDSChallenge |
| Authorization | [0..1] | Authorization |
| PanAliasData | [0..1] | PanAliasData |

**DataThreeDSAuthorization1**

| | | |
|---|---|---|
| ThreeDSAuthorizationRequest1 | | ThreeDSAuthorizationRequest1 |
| ThreeDSChallenge | [0..1] | ThreeDSChallenge |
| Authorization | [0..1] | Authorization |
| PanAliasData | [0..1] | PanAliasData |

**DataThreeDSAuthorization2**

| | | |
|---|---|---|
| ThreeDSAuthorizationRequest2 | | ThreeDSAuthorizationRequest2 |
| Authorization | [0..1] | Authorization |
| PanAliasData | [0..1] | PanAliasData |

**DataThreeDSVersioning**

| | | |
|---|---|---|
| ThreeDSVersioningRequest | | ThreeDSVersioningRequest |
| ThreeDSVersion | | string |

*Scheme 1 – BPWXmlResponse*

As can be seen, all messages have a single root element: BPWXmlResponse. Each message contains the most relevant data of the request and the data provided in the reply. The reply elements are present only if no errors occur.

**The effected parsing of XML replies must not be validating: thanks to the latest system developments, in the future it will be possible to add further elements to the messages. The applications must ignore any unknown elements without causing malfunctioning.**

As already said, many reply messages share common elements. In particular, the elements <BPWXmlResponse>, <Header>, <Authorization> and <Operation> are illustrated below

## 3.3.1 Element <BPWXmlResponse>

This is the root element of all the reply documents, there is a single element of this type in the message: here follows an example in which the part relating to the data element has been eliminated.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<BPWXmlResponse>
   <Timestamp>2015-07-04T12:02:55</Timestamp>
   <Result>00</Result>
   <!-- This MAC signs timestamp and result -->
   <MAC>ffb3553a4ab34c3a4ab34c3a4ab34c3a</MAC>
   <Data>
      ......................
   </Data>
</BPWXmlResponse>
```

**<BPWXmlResponse>**

- **<Timestamp>** date and time of reply message in yyyy-MM-ddTHH:mm:ss format
- **<Result>** outcome of request transaction

| Code | Description |
|---|---|
| 00 | Success |
| 01 | Order or ReqRefNum not found |
| 02 | ReqRefNum duplicated or not valid |
| 03 | Incorrect message format, missing or incorrect field |
| 04 | Incorrect API authentication, incorrect MAC or timestamp exceeding the limit range |
| 05 | Incorrect date, or period indicated is empty |
| 06 | Unforeseen error in the circuit during processing of request |
| 07 | TransactionID not found |
| 08 | Operator indicated not found |
| 09 | TRANSACTIONID indicated does not make reference to the entered ORDERID |
| 10 | Amount indicated exceeds maximum amount permitted |
| 11 | Incorrect status. Transaction not possible in the current status |
| 12 | Circuit disabled |
| 13 | Duplicated order |
| 16 | Currency not supported or not available for the merchant |
| 17 | Exponent not supported for the chosen currency |
| 20 | The card is VBV/SecureCode/SafeKey-enabled; the reply contains the data for redirection to ACS website |
| 21 | Maximum time-limit for forwarding VBV request step 2 expired |
| 25 | A call to *3DS method* must be performed by the Requestor |
| 26 | A *challenge flow* must be initiated by the Requestor |

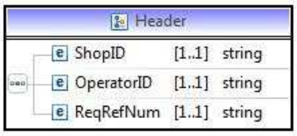
| | |
|---|---|
| 35 | No payment instrument is acceptable |
| 37 | Missing CVV2: this is compulsory for the circuit selected |
| 38 | Pan alias not found or revoked |
| 40 | Empty Xml or missing ‘data’ parameter |
| 41 | Xml not parsable |
| 98 | Application error |
| 99 | Transaction failed, see specific outcome attached to the element <Data> of the reply. |

- **`<MAC>`** signature of timestamp and of outcome (see appendix 4.2.7)
- **`<Data>`** data relating to authorization request and to reply message

In case of unexpected application crash (outcome 98), the tag **`<Data>`** will not be present and the value of MAC will be NULL:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<BPWXmlResponse>
   <Timestamp>2015-07-04T12:02:55</Timestamp>
   <Result>98</Result>
   <MAC>NULL</MAC>
</BPWXmlResponse>
```

## 3.3.2 Element <Header>



***Scheme 2 - Header***

This element describes a general header.

Here below is an example in XML of said element

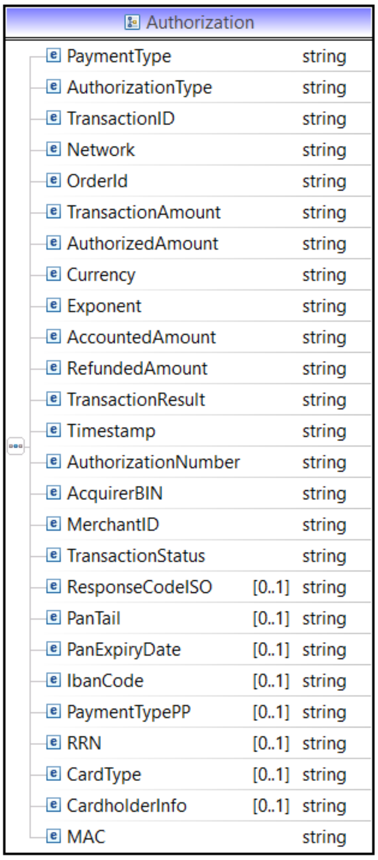
```
      <Header>
         <ShopID>0000000000000003</ShopID>
         <OperatorID> AD456123</OperatorID>
         <ReqRefNum>20150501901234567890123452289000</ ReqRefNum >
      </Header>
```

Here below are the sub-elements with their respective meanings:

**`<Header>`**

- **`<ShopID>`** store identifier (MID)
- **`<OperatorID>`** operator identifier (User ID)
- **`<ReqRefNum>`** unique request identifier managed by merchant

## 3.3.3 Element <Authorization>



| Authorization | | |
|---|---|---|
| PaymentType | | string |
| AuthorizationType | | string |
| TransactionID | | string |
| Network | | string |
| OrderId | | string |
| TransactionAmount | | string |
| AuthorizedAmount | | string |
| Currency | | string |
| Exponent | | string |
| AccountedAmount | | string |
| RefundedAmount | | string |
| TransactionResult | | string |
| Timestamp | | string |
| AuthorizationNumber | | string |
| AcquirerBIN | | string |
| MerchantID | | string |
| TransactionStatus | | string |
| ResponseCodeISO | [0..1] | string |
| PanTail | [0..1] | string |
| PanExpiryDate | [0..1] | string |
| IbanCode | [0..1] | string |
| PaymentTypePP | [0..1] | string |
| RRN | [0..1] | string |
| CardType | [0..1] | string |
| CardholderInfo | [0..1] | string |
| MAC | | string |

***Scheme 3 - Authorization***

This element describes a general authorization, whether immediate or deferred.

Here below is an example in XML of said element

```
<Authorization>
   <PaymentType>03</PaymentType>
   <AuthorizationType>I</AuthorizationType>
   <TransactionID> C355645658457564564565636</TransactionID>
   <Network>01</Network>
   <OrderID>A398459</OrderID>
   <TransactionAmount>10000</TransactionAmount>
   <AuthorizedAmount>10000</AuthorizedAmount>
   <Currency>978</Currency>
   <Exponent>2</Exponent>            → present with OPTION "x" or currency not euro
   <AccountedAmount>10000</AccountedAmount>
   <RefundedAmount>100</RefundedAmount>        → present if RELEASE=02 in the request
   <TransactionResult>00</TransactionResult>
   <Timestamp>2015-07-09T21:05:44</Timestamp>
   <AuthorizationNumber>A93485</AuthorizationNumber>
   <AcquirerBIN>123450943</AcquirerBIN>
   <MerchantID>0983473569324509</MerchantID>
   <TransactionStatus>01</TransactionStatus>
   <ResponseCodeISO>00</ResponseCodeISO>       → present if the store is SV53 enabled
   <PanTail>2025</PanTail>                     → present if the store is SV64 enabled
   <PanExpiryDate>2408</PanExpiryDate>         → present if the store is SV64 enabled
   <IbanCode>IT37Z0760101600000028426203</IbanCode>     → only for iban transactions
   <PaymentTypePP>0</PaymentTypePP>            → present if the store is SV73 enabled
```

```
        <RRN>161512000003</RRN>                     → present if the store is SV71 enabled
        <CardType>C</CardType>                      → present if the store is SV82 enabled
        <CardHolderInfo>A message from card issuer</CardHolderInfo>
        <!-- This MAC signs the authorization  -->
        <MAC>3204989a63de6ae849c930kd834oes83</MAC>
     </Authorization>
```

Here below are the sub-elements with their respective meanings:

**<Authorization>**

- **<PaymentType>** type of payment allowed

| Code | Description |
|---|---|
| 03 | SSL |
| 04 | VBV : merchant and consumer adhering to VBV |
| 05 | SecureCode : merchant and consumer adhering to SecureCode |
| 06 | Merchant VBV: merchant adhering to VBV and consumer not adhering |
| 07 | Merchant SecureCode: merchant adhering to SecureCode and consumer not adhering |
| 08 | VBV Owner not authenticated: merchant adhering to VBV; the consumer has not authenticated himself correctly |
| 09 | Mail order/Telephone order |
| 13 | SafeKey: merchant and consumer adhering to SafeKey |
| 14 | SafeKey Merchant: merchant adhering to SafeKey and consumer not adhering |
| 15 | SafeKey Owner not authenticated: merchant adhering to SafeKey; the consumer has not authenticated himself correctly |
| 16 | ProtectBuy: merchant and consumer adhering to ProtectBuy |
| 17 | ProtectBuy Merchant: merchant adhering to ProtectBuy and consumer not adhering |
| 18 | ProtectBuy Owner not authenticated: merchant adhering to ProtectBuy; the consumer has not authenticated himself correctly |

- **<AuthorizationType>** type of authorization granted: D=Deferred, I=Immediate
- **<TransactionID>** the transaction identifier assigned by SIA VPOS
- **<Network>** network code:

| Code | Description |
|---|---|
| 01 | Visa |
| 02 | Mastercard |
| 04 | Maestro |
| 06 | American Express |
| 07 | Diners |
| 08 | JCB |
| 80 | IBAN |
| 81 | AmazonPay |
| 82 | EnelX |
| 84 | Satispay |
| 91 | BancomatPay (Jiffy) |
| 94 | Postepay (deprecated) |
| 96 | MyBank |
| 97 | PayPal |

Pan Aliases and special codes (only for requests)

| | |
|---|---|
| 83 | COF Pan Alias |
| 88 | Token Pan Alias |
| 89 | GTW Pan Alias |
| 93 | Unknown Credit Card (the system will try to decode the network from the pan) |
| 98 | Default Pan Alias |

- `<OrderID>` order code
- `<TransactionAmount>` amount of transaction in Eur cents. For ASI card verification transactions the amount is usually 0 but it will be increased to 10 cent if the chosen network does not support a 0 cent transaction.
- `<AuthorizedAmount>` authorized amount in Eur cents. If authorization is denied, the amount is equal to zero
- `<Currency>` currency ISO code: 978=Eur
- `<Exponent>` number of decimals for the currency (present only if option=x or currency not euro)
- `<AccountedAmount>` the accounted amount in Eur cents.
- `<RefundedAmount>` the refunfed amount in Eur cents *(introduced by Release 02. It is present only if the **RELEASE=02** parameter is specified in the request)*
- `<TransactionResult>` outcome of transaction

| **Code** | **Description** |
|---|---|
| 00 | Success |
| 01 | Denied due to problems in the request message |
| 02 | Denied due to problems in the store registry |
| 03 | Denied due to communication problems with the authorization circuits |
| 04 | Denied by card issuer |
| 05 | Denied due to incorrect card number |
| 06 | Unforeseen error during processing of request |
| 45 | Denied authorization due to foreign card filter. |
| 99 | Authorization underway with MyBank or BancomatPay |

If the store is SV67 service enabled (explicit outcomes of antifraud filters from API) instead of the generic transaction 45 outcome, the following outcomes will be returned, for transactions carried out with active explicit outcome service (SV67 from API or SV54 from Redirect).

| **Code** | **Description** |
|---|---|
| 60 | Denied due to failed antifraud check Riskshield |
| 61 | Denied due to failed antifraud check AmexPan |
| 62 | Denied due to failed antifraud check AmexPanIP |
| 63 | Denied due to failed antifraud check H3GPan |
| 64 | Denied due to failed antifraud check ItaPanCountry |
| 65 | Denied due to failed antifraud check PaypalCountry |
| 66 | Denied due to failed antifraud check CardEnrolledAuthenticate |
| 67 | Denied due to failed antifraud check PanBlackList |
| 68 | Denied due to failed antifraud check CountryPan |
| 69 | Denied due to failed antifraud check PrepaidPan |
| 70 | Denied due to failed antifraud check DebitPan |
| 71 | Denied due to failed antifraud check VirtualPan |
| 72 | Denied due to failed antifraud check ThresholdAmount |
| 73 | Denied due to failed antifraud check H3GPanLit |
| 74 | Denied due to failed antifraud check AcqrBinTab |
| 75 | Denied due to failed antifraud check CountryWL |
| 76 | Denied due to failed antifraud check PrepgWLPan |
| 77 | Denied due to failed antifraud check IllimitPan |

- **<Timestamp>** date and time of transaction in yyyy-mm-ggTHH:mm:ss format
- **<AuthorizationNumber>** authorization code (filled in in case of positive outcome). This is a string having a maximum length of 6 characters for all circuits excluding MyBank; the latter, on the other hand, has a fixed length of 35 characters and contains the transaction identifier assigned by the Validation Service. It is not relevant if the transaction is performed using the Paypal circuit
- **<AcquirerBIN>** acquirer bin. Acquirer's International identification code.
- **<MerchantID>** vpos merchant code or acquirer merchant code if the store is SVAH enabled.
- **<TransactionStatus>** current status of the authorization

It can assume different meanings for immediate or deferred authorizations:

**Immediate**

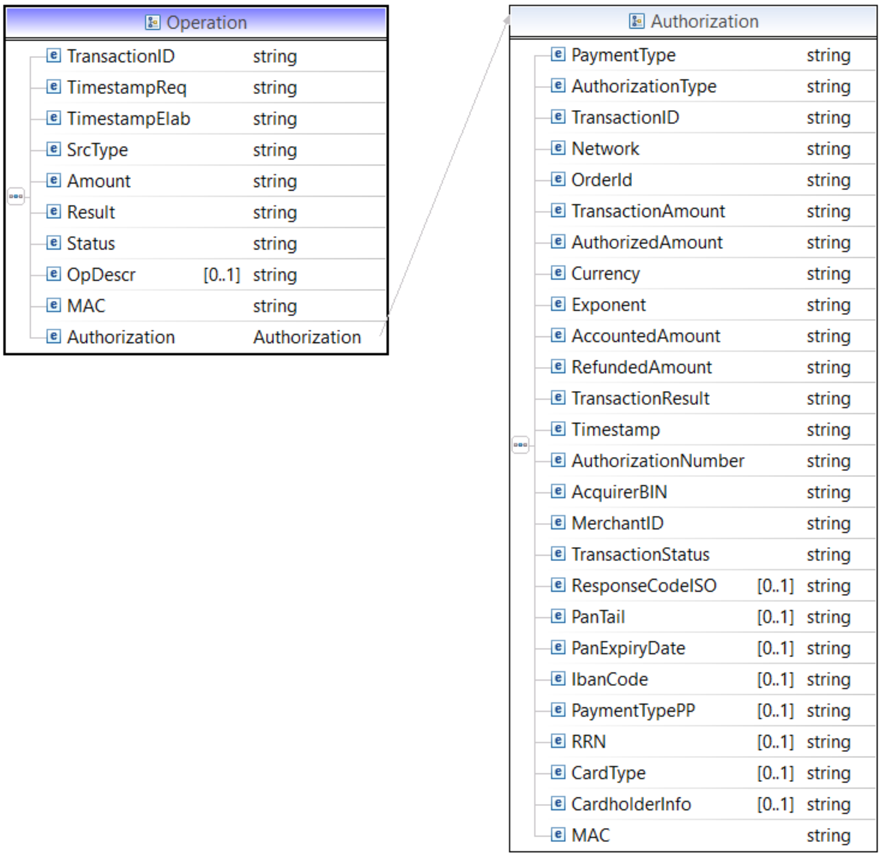
| **Code** | **Description** |
|---|---|
| 00 | Authorization granted, bookable |
| 01 | Authorization denied |
| 02 | Booked authorization to be processed |
| 03 | Booked authorization processed by clearing |
| 04 | Reversed authorization |
| 21 | Authorization to be reversed due to transaction error |
| 99 | Authorization underway with MyBank or BancomatPay |

**Deferred**

| **Code** | **Description** |
|---|---|
| 10 | Deferred authorization open |
| 11 | Deferred authorization closed |

- **<ResponseCodeISO>** Information present only for the sections Authorization in response to an authorization message (Online, Deferred or VBV Authorization) and for stores adhering to the "SV53 - Provide ResponseCodeISO in authorizations with APIs" service. It contains the outcome code received from the relevant circuit.
- **<PanTail>** last four digit of the card number (only if the shop is SV64 enabled)
- **<PanExpiryDate>** expiry date of the card (only if the shop is SV64 enabled)
- **<IbanCode>** iban (for iban transactions)
- **<PaymentTypePP>** "2" for PagaConPostepay transactions by Wallet; "4" for PagaConPostepay transactions by Card; "0" otherwise (only if the shop is SV73 enabled).
- **<RRN>** unique identifier of the transaction in the iso authorization message (only if the shop is SV71 enabled).
- **<CardType>** C for Credit; D for Debit; P for prepaid (only if the shop is SV82 enabled and the information is available).
- **<CardHolderInfo>** This field can contain an occasional message from the card holder issuer to be provided to the card holder
- **<MAC>** Message authentication code: signature of authorization (see appendix 4.2.9)

## 3.3.4 Element <Operation>



| Operation | | |
|---|---|---|
| TransactionID | | string |
| TimestampReq | | string |
| TimestampElab | | string |
| SrcType | | string |
| Amount | | string |
| Result | | string |
| Status | | string |
| OpDescr | [0..1] | string |
| MAC | | string |
| Authorization | | Authorization |

| Authorization | | |
|---|---|---|
| PaymentType | | string |
| AuthorizationType | | string |
| TransactionID | | string |
| Network | | string |
| OrderId | | string |
| TransactionAmount | | string |
| AuthorizedAmount | | string |
| Currency | | string |
| Exponent | | string |
| AccountedAmount | | string |
| RefundedAmount | | string |
| TransactionResult | | string |
| Timestamp | | string |
| AuthorizationNumber | | string |
| AcquirerBIN | | string |
| MerchantID | | string |
| TransactionStatus | | string |
| ResponseCodeISO | [0..1] | string |
| PanTail | [0..1] | string |
| PanExpiryDate | [0..1] | string |
| IbanCode | [0..1] | string |
| PaymentTypePP | [0..1] | string |
| RRN | [0..1] | string |
| CardType | [0..1] | string |
| CardholderInfo | [0..1] | string |
| MAC | | string |

*Scheme 4 - Operation*

This element represents a generic accounting transaction

```
<Operation>
   <TransactionID>C5555358794</TransactionID>
   <TimestampReq>2015-07-04T22:02:55</TimestampReq>
   <TimestampElab>NULL</TimestampElab>
   <SrcType>01</SrcType>
   <Amount>10000</Amount>
   <Result>00</Result>
   <Status>00</Status>
   <OpDescr>OrderRefundA398459Attempt1</OpDescr>
   <!-- This MAC signs operation data  -->
   <MAC>12334c3a4ab34c3a4ab34c3a4ab3ffa1</MAC>
   <Authorization>
      …………………………….
   </Authorization>
</Operation>
```

**<Operation>**

The element contains all the data relating to the accounting transaction consisting of the following elements:

- **`<TransactionID>`** identifier of the transaction of the bookable transaction
- **`<TimestampReq>`** date and time of request in yyyy-MM-ddTHH:mm:ss format
- **`<TimestampElab>`** date and time of processing in yyyy-MM-ddTHH:mm:ss format
- **`<SrcType>`** type of bookable transaction

| Code | Description |
|---|---|
| 01 | Reversal of authorization |
| 02 | Credit transaction |
| 03 | Cancellation of booking |
| 04 | Booking transaction |

- **`<Amount>`** amount of transaction in Eur cents
- **`<Result>`** outcome of transaction

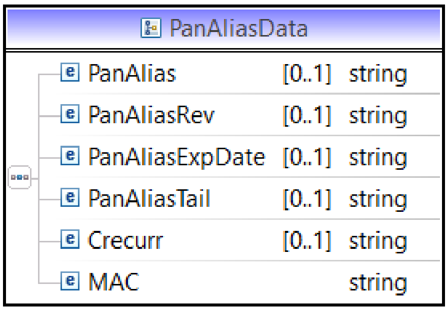
| Code | Description |
|---|---|
| 00 | Success |
| 01 | Expired time-limits |
| 02 | Denied due to problems with store registered data |
| 03 | Denied due to communication problems with authorization circuits |
| 04 | Denied by card issuer |
| 05 | Ceiling not restored |
| 06 | Unexpected error during processing of request |

- **`<Status>`** status of transaction

| Code | Description |
|---|---|
| 00 | Completed successfully |
| 01 | Failed |

- **`<OpDescr>`** optional description that could be associated to the accounting operation
- **`<MAC>`** message authentication code: signature of accounting transaction (see appendix 4.2.8)
- **`<Authorization>`** data relating to the authorization for which the accounting transaction was performed

## 3.3.5 Element <PanAliasData>



```
PanAliasData
PanAlias        [0..1]  string
PanAliasRev     [0..1]  string
PanAliasExpDate [0..1]  string
PanAliasTail    [0..1]  string
Crecurr         [0..1]  string
MAC                     string
```

*Scheme 5 - PanAliasData*

This element contains pan alias data for a granted transaction

```
<PanAliasData>
       <PanAlias>0000197412081271677</PanAlias>
       <PanAliasRev></PanAliasRev>
       <PanAliasExpDate>2911</PanAliasExpDate>
       <PanAliasTail>0003</PanAliasTail>
       <Crecurr>PST581426946</Crecurr>
       <MAC>E61612E0C0F71A2FE838BC0736B396E6</MAC>
</PanAliasData>
```

**<PanAliasData>**

Card number or billing agreement alias data

- **<PanAlias>** Pan Alias of the card number or billing agrrement id
- **<PanAliasRev>** Revoked Pan Alias, if any, based on the alias pan service configured for the store
- **<PanAliasExpDate>** Card expiry date (with service SV45 or SV88 only)
- **<PanAliasTail>** Last four digit of the card number (with service SV45 or SV88 only)
- **<CRecurr>** Recurring Code or Transaction ID to be used for following recurring transactions
- **<MAC>** Message authentication code: signature of timestamp and result (see appendix 4.2.19)

The **PanAliasRev** field is always present and shows a value only if the alias created replaces one for the same pan (typically with service SV34 – single alias pan).
The fields **PanAliasExpDate** and **PanAliasTail** are present only if the store has subscribed to the service SV45 or the service SV88.
The **CRecurr** field is present only for recurring authorizations or authorizations with CREATEPANALIAS = “S” (if SVA4 is not active).

# 3.4 *Authorization Request*

## 3.4.1 Online authorization request

The online authorization request message permits to forward authorization requests to the circuits.
The fields to be specified in the HTTP request message are the following:

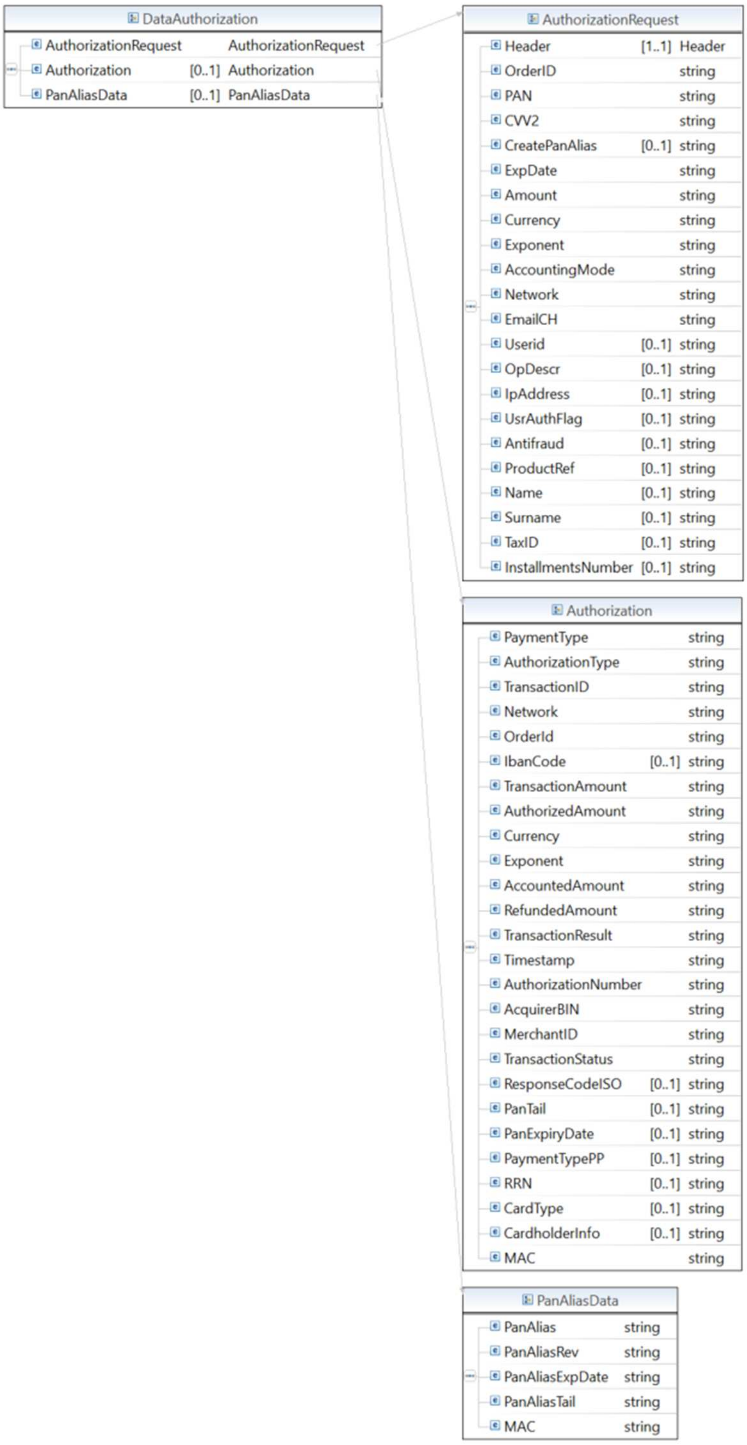
| Field | Compulsory | Size | Type | Description |
|---|---|---|---|---|
| | | | | |
| OPERATION | Y | | A | Requested transaction: to be filled in with "AUTHORIZATION" |
| TIMESTAMP | Y | 23 | AN | Local timestamp in yyyy-MM-ggTHH:mm:ss.SSS format |
| SHOPID | Y | 15 | AN | Identifier of merchant’s store assigned by SIA (Merchant ID) |
| ORDERID | Y | Min. 1<br>Max.50 | AN | Unique order identifier |
| OPERATORID | Y | Min. 8<br>Max.18 | AN | It indicates the person who has requested the transaction. |
| REQREFNUM | Y | 32 | N | Unique request identifier managed by the merchant. It can be used to retrieve information on the request made also in the case of no response. The first 8 digits must be in the yyyyMMdd format, including the request date. |
| PAN | Y | Min. 10<br>Max.19 | AN | Card number |
| CVV2 | N | Min. 3<br>Max.4 | N | Control code associated to the card number (optional) |
| CREATEPANALIAS | N | 0..1 | A | Create Pan Alias: S=DO, null=DON'T |
| EXPDATE | Y | 4 | N | Card expiry date – yyMM- |
| AMOUNT | Y | Min. 1<br>Max. 8 | N | Amount expressed in the smallest currency unit (EUR cents).<br>The amount must not be preceded by zeros.<br>Minimum length 1 maximum length 8.<br><br>For ASI card verification transactions the amount MUST be set to 0. For real transactions the amount has to be at least the minimum supported by the merchant’s acquirer (usually 10 cent for euro). |
| CURRENCY | Y | 3 | N | Currency: ISO code (EUR = 978). |
| ACCOUNTINGMODE | Y | 1 | AN | Type of booking to be used for this order:<br>• D Deferred<br>• I Immediate<br>For ASI card verification transactions the accounting mode MUST be set to D. |
| EXPONENT | N | 1 | N | Exponent of the chosen Currency<br>(Recommended if the currency is different from Euro) |
| NETWORK | Y | 2 | N | Card authorization circuit (e.g.: 01 for VISA). If the merchant is not in possession of the circuit code, it is possible to delegate the ATPOS system to calculate the latter by setting as NETWORK the value of 93. |
| EMAILCH | N | Min. 7<br>Max. 50 | AN | E-mail of card holder (optional) |
| USERID | N | Min. 1<br>Max. 30 | AN | Identifier of card holder |
| ACQUIRER | N | 5 | N | Code of acquirer in the transaction to be carried out |
| IPADDRESS | N | Min. 7<br>Max. 15 | AN | IP address associated to the request |

| | | | | |
|---|---|---|---|---|
| OPDESCR | N | 100 | AN | Additional description of the operation at merchant's discretion, in case of authorization with immediate booking. If the booking is deferred, this field is skipped. |
| USRAUTHFLAG | N | 1 | AN | "0" for occasional user; "1" for registered user; "2" for unrecognized user" |
| RELEASE | Y | 2 | AN | Release of APIs: to be filled in with "02" |
| OPTIONS | N | Min 0<br>Max 26 | AN | The field OPTIONS permits to activate various additional options for the ongoing message. |
| ANTIFRAUD | N | na | AN | Antifraud data payload containing additional information used for antifraud checks. Field is mandatory if SV69 is active. |
| PRODUCTREF | N | 15 | AN | Sale identifier |
| NAME | N | 40 | AN | Cardholder's first name |
| SURNAME | N | 40 | AN | Cardholder's surname |
| TAXID | N | 16 | AN | Cardholder's tax id |
| TRECURR | N | 1 | AN | Type of recurring payment.<br>Mandatory for a recurring payment or with CREATEPANALIAS = 'S' (if SVA4 is not active).<br>The admitted values are:<br>R – Scheduled **R**ecurring transaction<br>U – **U**nscheduled recurring transaction<br>C – **C**ard stored on file (pan alias/token) notification<br>For granted authorization, in the pan alias section of the result message CRECURR will be sent back to the merchant to be used for the following recurring payments. |
| CRECURR | N | Max 50 | AN | For a following recurring transaction must contain the CRECURR value received in the response of the **first** recurring transaction. |
| INSTALLMENTSNUMBER | N | Min 0<br>Max 2 | N | Number of installment. Value from 0 to 99. |
| MAC | Y | 32, 40, 64 | AN | Message Authentication Code. For the related calculation, see appendix 4.2.10 |

## Request of online authorization in XML format

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<BPWXmlRequest>
<Release>02</Release>
<Request>
<Operation>AUTHORIZATION</Operation>
<Timestamp>2015-02-08T12:02:00.000</Timestamp>
<MAC>115025d5a5b65df687790867bdece136</MAC>
</Request>
   <Data>
      <AuthorizationRequest>
         <Header>
            <ShopID>000000000000003</ShopID>
            <OperatorID>oper0001</OperatorID>
            <ReqRefNum>20150501901234567890123452289000</ReqRefNum>
         </Header>
         <OrderID>1234567890</OrderID>
         <Pan>9998500000000015</Pan>
         <CVV2>123</CVV2>
         <ExpDate>0409</ExpDate>
         <Amount>4450</Amount>
         <Currency>978</Currency>
         <Exponent>2</Exponent>
         <AccountingMode>I</AccountingMode>
         <Network>01</Network>
         <EmailCH>address@yourcompany.com</EmailCH>
         <Userid>user1</Userid>
         <OpDescr>CallCenterRequest1037</OpDescr>
         <ProductRef>12345678</ProductRef>
         <Name>Jon</Name>
         <Surname>Snow</Surname>
         <TaxID>SNWJNO96A01F205L</TaxID>
         <TRecurr>U</TRecurr>
         <CRecurr>PST581426946</CRecurr>
         <InstallmentsNumber>3</InstallmentsNumber>
      </AuthorizationRequest>
   </Data>
</BPWXmlRequest>
```

The response message to the authorization request is formatted in XML. The data section is described on the following scheme.



| DataAuthorization | | |
|---|---|---|
| AuthorizationRequest | | AuthorizationRequest |
| Authorization | [0..1] | Authorization |
| PanAliasData | [0..1] | PanAliasData |

| AuthorizationRequest | | |
|---|---|---|
| Header | [1..1] | Header |
| OrderID | | string |
| PAN | | string |
| CVV2 | | string |
| CreatePanAlias | [0..1] | string |
| ExpDate | | string |
| Amount | | string |
| Currency | | string |
| Exponent | | string |
| AccountingMode | | string |
| Network | | string |
| EmailCH | | string |
| Userid | [0..1] | string |
| OpDescr | [0..1] | string |
| IpAddress | [0..1] | string |
| UsrAuthFlag | [0..1] | string |
| Antifraud | [0..1] | string |
| ProductRef | [0..1] | string |
| Name | [0..1] | string |
| Surname | [0..1] | string |
| TaxID | [0..1] | string |
| InstallmentsNumber | [0..1] | string |

| Authorization | | |
|---|---|---|
| PaymentType | | string |
| AuthorizationType | | string |
| TransactionID | | string |
| Network | | string |
| OrderId | | string |
| IbanCode | [0..1] | string |
| TransactionAmount | | string |
| AuthorizedAmount | | string |
| Currency | | string |
| Exponent | | string |
| AccountedAmount | | string |
| RefundedAmount | | string |
| TransactionResult | | string |
| Timestamp | | string |
| AuthorizationNumber | | string |
| AcquirerBIN | | string |
| MerchantID | | string |
| TransactionStatus | | string |
| ResponseCodeISO | [0..1] | string |
| PanTail | [0..1] | string |
| PanExpiryDate | [0..1] | string |
| PaymentTypePP | [0..1] | string |
| RRN | [0..1] | string |
| CardType | [0..1] | string |
| CardholderInfo | [0..1] | string |
| MAC | | string |

| PanAliasData | |
|---|---|
| PanAlias | string |
| PanAliasRev | string |
| PanAliasExpDate | string |
| PanAliasTail | string |
| MAC | string |

*Scheme 6 - Data section for AUTHORIZATION response*

As can be noted, the response to an authorization request is essentially composed of an Authorization-type element. In the case where an authentication error occurs or there is a formal error in the request, the Authorization element will not be created.

Here below is an example of a file generated by the response to the online authorization request:

```
<?xml version="1.0" encoding="ISO88591" ?>
<BPWXmlResponse>
```

```
   <Timestamp>20150409T12:02:38</Timestamp>
   <Result>00</Result>
   <! This MAC signs timestamp and result >
   <MAC>8A74330BA1A1A085581EAA2409D8DC68FCC4395E</MAC>
   <Data>
      <! This element contains request data >
      <AuthorizationRequest>
         <Header>
            <ShopID>000000000000003</ShopID>
            <OperatorID>AD456123</OperatorID>
            <ReqRefNum>20150501901234567890123452289000</ReqRefNum>
         </Header>
         <OrderID>p91</OrderID>
         <Pan>999850xxxxxx0015</Pan>
         <CVV2>000</CVV2>
         <CreatePanAlias>S</CreatePanAlias>
         <ExpDate>0409</ExpDate>
         <Amount>4450</Amount>
         <Currency>978</Currency>
         <Exponent>2</Exponent>
         <AccountingMode>I</AccountingMode>
         <Network>01</Network>
         <EmailCH>info@titolare.it</EmailCH>
         <Userid>user1</Userid>
         <OpDescr>CallCenterRequest1037</OpDescr>
         <ProductRef>12345678</ProductRef>
         <Name>Jon</Name>
         <Surname>Snow</Surname>
         <TaxID>SNWJNO96A01F205L</TaxID>
         <TRecurr>U</TRecurr>
         <CRecurr>PST581426946</CRecurr>
         <InstallmentsNumber>3</InstallmentsNumber>
      </AuthorizationRequest>
      <Authorization>
         <PaymentType>03</PaymentType>
         <AuthorizationType>I</AuthorizationType>
         <TransactionID>8032180310AB0E30917930112</TransactionID>
         <Network>01</Network>
         <OrderID>pos91</OrderID>
         <TransactionAmount>4450</TransactionAmount>
         <AuthorizedAmount>4450</AuthorizedAmount>
         <Currency>978</Currency>
         <Exponent>2</Exponent>
         <AccountedAmount>0</AccountedAmount>
         <RefundedAmount>100</RefundedAmount>
         <TransactionResult>00</TransactionResult>
         <Timestamp>20150409T12:02:38</Timestamp>
         <AuthorizationNumber>622851</AuthorizationNumber>
         <AcquirerBIN>453997</AcquirerBIN>
         <MerchantID>000000000000476</MerchantID>
         <TransactionStatus>02</TransactionStatus>
         <ResponseCodeISO>00</ResponseCodeISO>
          <! This MAC signs authorization data >
         <MAC>0EA6645D79E9752BE05800BE9CFE623CE3973395</MAC>
      </Authorization>
      <PanAliasData>
         <PanAlias> 0000197412081271677</PanAlias>
         <PanAliasRev></PanAliasRev>
         <PanAliasExpDate>2911</PanAliasExpDate>
         <PanAliasTail>0003</PanAliasTail>
         <CRecurr>PST581426946</CRecurr>
         <MAC>E61612E0C0F71A2FE838BC0736B396E6</MAC>
      </PanAliasData>
   </Data>
</BPWXmlResponse>
```

The meaning of the elements is the following:

**<BPWXmlResponse>**

This is the root element of the document, there is only one element of this type in the message, which consists of the following elements:

- **<Timestamp>** date and time of the response message
- **<Result>** outcome of the requested transaction. Possible outcomes:

| Code | Description |
|---|---|
| 00 | Success |
| 02 | ReqRefNum duplicated or incorrect |
| 03 | Incorrect message format, missing or incorrect field |
| 04 | Incorrect API authentication, incorrect MAC |
| 06 | Unforeseen error during processing of request |
| 37 | Missing CVV2 |
| 40 | Empty Xml or missing 'data' parameter |
| 41 | Xml not parsable |
| 98 | Application error |
| 99 | Transaction failed, see specific outcome enclosed in the element <Data> of the response. |

- **<MAC>** message authentication code: signature of timestamp and result (see appendix 4.2.10)
- **<Data>** data relating to the authorization request and response message

**<Data>**

There is only one element of this type in the message, containing all the data relating to the authorization request and response message and consisting of the following elements:

- **<AuthorizationRequest>** data relating to the authorization request
- **<Authorization>** data relating to the response message
- **<PanAliasData>** data relating to the pan alias or recurring codes (if any)

**<AuthorizationRequest>**

There is only one element of this type in the message, containing all the data relating to the authorization request and consisting of the following elements:

- **<Header>** data relating to the request sent
- **<OrderID>** order identifier
- **<Pan>** hidden card number (showing only the first six and the last four digits)
- **<CVV2>** hidden additional card number (sequence of zeros having a length equal to the length of the field in the request)
- <**CreatePanAlias**> Create Pan Alias: S=DO, null=DON'T
- **<ExpDate>** card expiry date
- **<Amount>** amount of requested authorization in Euro cents
- **<Currency>** currency ISO code: 978=Eur
- **<Exponent>** number of decimals for the currency
- **<AccountingMode>** type of booking to be used: D=Deferred, I=Immediate

| | |
|---|---|
| **<Network>** | card authorization circuit |
| **<EmailCH>** | e-mail of card holder |
| **<Userid>** | identifier of card holder |
| **<OpDescr>** | optional description that could be associated to the operation |
| **<IpAddress>** | ip address associated to the request |
| **<UsrAuthFlag>** | flag indicating the type of userid sent (for Riskshield) |
| **<Antifraud>** | antifraud data payload containing additional information used for antifraud checks |
| **<ProductRef>** | sale identifier |
| **<Name>** | cardholder's first name |
| **<Surname>** | cardholder's surname |
| **<TaxID>** | cardholder's tax ID |
| **<TRecurr>** | type of recurring payment |
| **<CRecurr>** | recurring code |
| **<InstallmentsNumber>** | number of installment |

### <Header>

There is only one element of this type in the message, which includes the date relating to the request sent. For the related description see paragraph 3.3.2 in the chapter "Response messages in XML".

### <Authorization>

There is only one element of this type in the message, which includes the authorization data. For the related description see paragraph 3.3.3 in the chapter "Response messages in XML"

### <PanAliasData>

There is only one element of this type in the message, which includes the card alias data. For the related description see paragraph 3.3.5 in the chapter "Response messages in XML"

## 3.4.2 IBAN authorization request

The IBAN authorization request message permits to do IBAN checks with order registration.
The fields to be specified in the HTTP request message are the following:

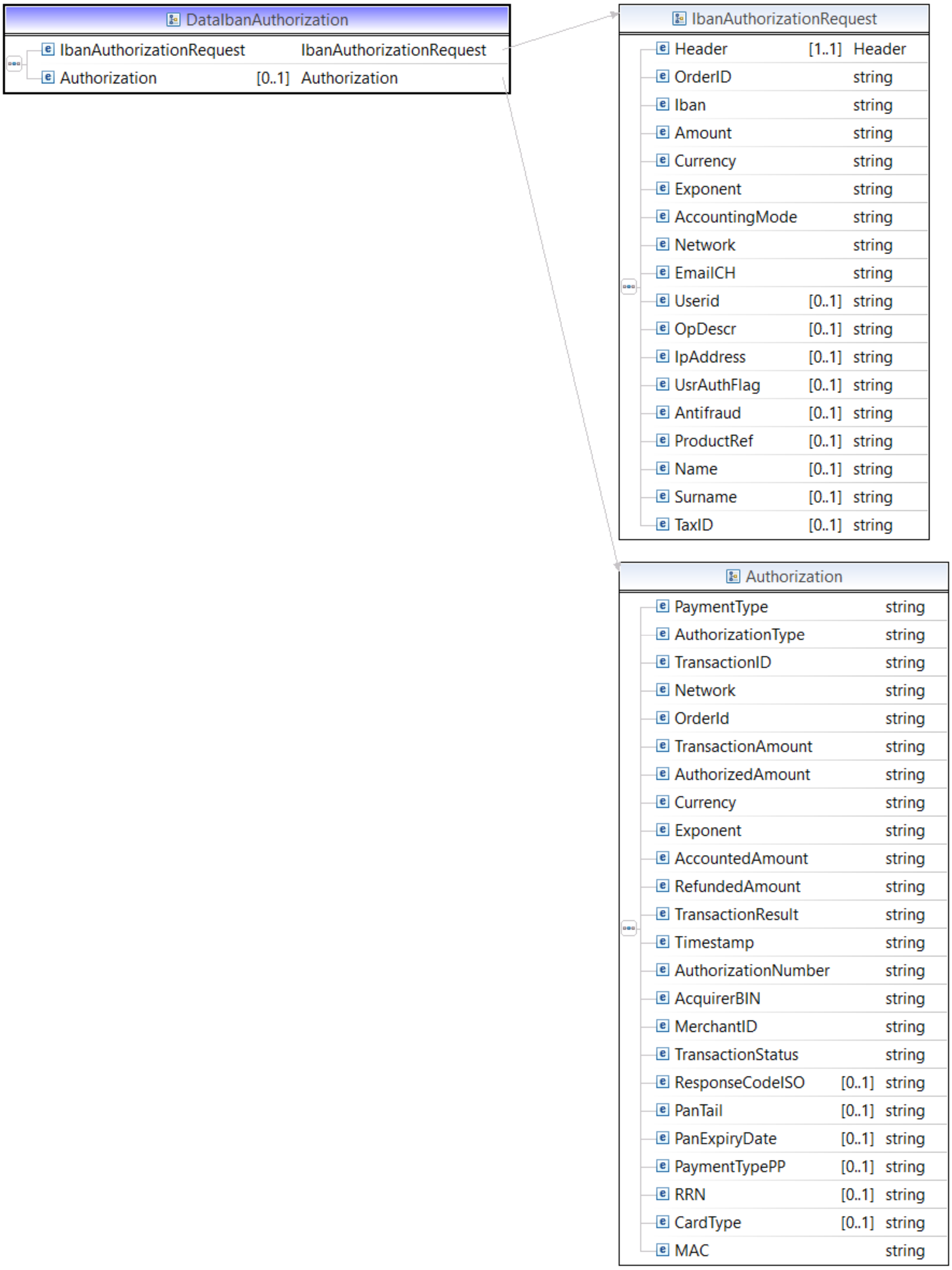
| Field | Compulsory | Size | Type | Description |
|---|---|---|---|---|
| | | | | |
| OPERATION | Y | | A | Requested transaction: to be filled in with "IBANAUTHORIZATION" |
| TIMESTAMP | Y | 23 | AN | Local timestamp in yyyy-MM-ggTHH:mm:ss.SSS format |
| SHOPID | Y | 15 | AN | Identifier of merchant's store assigned by SIA (Merchant ID) |
| ORDERID | Y | Min. 1<br>Max.50 | AN | Unique order identifier |
| OPERATORID | Y | Min. 8<br>Max.18 | AN | It indicates the person who has requested the transaction. |
| REQREFNUM | Y | 32 | N | Unique request identifier managed by the merchant. It can be used to retrieve information on the request made also in the case of no response. The first 8 digits must be in the yyyyMMdd format, including the request date. |
| IBAN | Y | 27 | AN | IBAN |
| AMOUNT | Y | Min. 2<br>Max. 8 | N | Amount expressed in the smallest currency unit (EUR cents).<br>The amount must not be preceded by zeros.<br>Minimum length 1 maximum length 8.<br><br>For ASI card verification transactions the amount MUST be set to 0. For real transactions the amount has to be at least the minimum supported by the merchant's acquirer (usually 10 cent for euro). |
| CURRENCY | Y | 3 | N | Currency: ISO code (EUR = 978). |
| EXPONENT | N | 1 | N | Exponent of the chosen Currency<br>(Recommended if the currency is different from Euro) |
| ACCOUNTINGMODE | Y | 1 | AN | Type of booking to be used for this order:<br>• D Deferred<br>• I Immediate<br>For ASI card verification transactions the accounting mode MUST be set to D. |
| NETWORK | Y | 2 | N | Fixed value "80". |
| EMAILCH | N | Min. 7<br>Max. 50 | AN | E-mail of card holder (optional) |
| USERID | N | Min. 1<br>Max. 30 | AN | Identifier of card holder |
| IPADDRESS | N | Min. 7<br>Max. 15 | AN | IP address associated to the request |
| OPDESCR | N | 100 | AN | Additional description of the operation at merchant's discretion, in case of authorization with immediate booking. If the booking is deferred, this field is skipped. |
| RELEASE | Y | 2 | AN | Release of APIs: to be filled in with "02" |
| OPTIONS | N | Min 0<br>Max 26 | AN | The field OPTIONS permits to activate various additional options for the ongoing message. |
| ANTIFRAUD | N | na | AN | Antifraud data payload containing additional information used for antifraud checks. Field is mandatory if SV69 is active. |
| PRODUCTREF | N | 15 | AN | Sale identifier |
| NAME | N | 40 | AN | Cardholder's first name |

|  |  |  |  |  |
|---|---|---|---|---|
| SURNAME | N | 40 | AN | Cardholder’s surname |
| TAXID | N | 16 | AN | Cardholder’s tax id |
| MAC | Y | 32, 40, 64 | AN | Message Authentication Code. For the related calculation, see appendix 4.2.29 |

## Request of IBAN authorization in XML format

```xml
<?xml version="1.0" encoding="ISO-8859-1"?>
<BPWXmlRequest>
<Release>02</Release>
<Request>
<Operation>IBANAUTHORIZATION</Operation>
<Timestamp>2019-02-08T12:02:00.000</Timestamp>
<MAC>115025d5a5b65df687790867bdece136</MAC>
</Request>
   <Data>
      <IbanAuthorizationRequest>
         <Header>
            <ShopID>000000000000003</ShopID>
            <OperatorID>oper0001</OperatorID>
            <ReqRefNum>20190501901234567890123452289000</ReqRefNum>
         </Header>
         <OrderID>1234567890</OrderID>
         <Iban>IT37Z0760101600000028426203</Iban>
         <Amount>4450</Amount>
         <Currency>978</Currency>
         <Exponent>2</Exponent>
         <AccountingMode>I</AccountingMode>
         <Network>01</Network>
         <EmailCH>address@yourcompany.com</EmailCH>
         <Userid>user1</Userid>
         <OpDescr>CallCenterRequest1037</OpDescr>
         <ProductRef>12345678</ProductRef>
         <Name>Jon</Name>
         <Surname>Snow</Surname>
         <TaxID>SNWJNO96A01F205L</TaxID>
      </IbanAuthorizationRequest>
   </Data>
</BPWXmlRequest>
```

The response message to the iban authorization request is formatted in XML. The data section is described on the following scheme.



| DataIbanAuthorization | | |
|---|---|---|
| IbanAuthorizationRequest | | IbanAuthorizationRequest |
| Authorization | [0..1] | Authorization |

| IbanAuthorizationRequest | | |
|---|---|---|
| Header | [1..1] | Header |
| OrderID | | string |
| Iban | | string |
| Amount | | string |
| Currency | | string |
| Exponent | | string |
| AccountingMode | | string |
| Network | | string |
| EmailCH | | string |
| Userid | [0..1] | string |
| OpDescr | [0..1] | string |
| IpAddress | [0..1] | string |
| UsrAuthFlag | [0..1] | string |
| Antifraud | [0..1] | string |
| ProductRef | [0..1] | string |
| Name | [0..1] | string |
| Surname | [0..1] | string |
| TaxID | [0..1] | string |

| Authorization | | |
|---|---|---|
| PaymentType | | string |
| AuthorizationType | | string |
| TransactionID | | string |
| Network | | string |
| OrderId | | string |
| TransactionAmount | | string |
| AuthorizedAmount | | string |
| Currency | | string |
| Exponent | | string |
| AccountedAmount | | string |
| RefundedAmount | | string |
| TransactionResult | | string |
| Timestamp | | string |
| AuthorizationNumber | | string |
| AcquirerBIN | | string |
| MerchantID | | string |
| TransactionStatus | | string |
| ResponseCodeISO | [0..1] | string |
| PanTail | [0..1] | string |
| PanExpiryDate | [0..1] | string |
| PaymentTypePP | [0..1] | string |
| RRN | [0..1] | string |
| CardType | [0..1] | string |
| MAC | | string |

*Scheme 7 - Data section for IBANAUTHORIZATION response*

As can be noted, the response to an authorization request is essentially composed of an Authorization-type element. In the case where an authentication error occurs or there is a formal error in the request, the Authorization element will not be created.

Here below is an example of a file generated by the response to the iban authorization request:

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<BPWXmlResponse>
   <Timestamp>2019-04-09T12:02:38</Timestamp>
   <Result>00</Result>
   <!-- This MAC signs timestamp and result -->
   <MAC>8A74330BA1A1A085581EAA2409D8DC68FCC4395E</MAC>
   <Data>
      <!-- This element contains request data -->
      <IbanAuthorizationRequest>
         <Header>
            <ShopID>000000000000003</ShopID>
            <OperatorID>AD456123</OperatorID>
            <ReqRefNum>20150501901234567890123452289000</ReqRefNum>
         </Header>
         <OrderID>p91</OrderID>
         <IBAN>IT37Z0760101600000028426203</IBAN>
         <Amount>4450</Amount>
         <Currency>978</Currency>
         <Exponent>2</Exponent>
         <AccountingMode>I</AccountingMode>
         <Network>01</Network>
         <EmailCH>info@titolare.it</EmailCH>
         <Userid>user1</Userid>
         <OpDescr>CallCenterRequest1037</OpDescr>
         <ProductRef>12345678</ProductRef>
         <Name>Jon</Name>
         <Surname>Snow</Surname>
         <TaxID>SNWJNO96A01F205L</TaxID>
      </IbanAuthorizationRequest>
      <Authorization>
         <PaymentType>03</PaymentType>
         <AuthorizationType>I</AuthorizationType>
         <TransactionID>8032180310AB0E30917930112</TransactionID>
         <Network>01</Network>
         <OrderID>pos91</OrderID>
         <IbanCode>IT37Z0760101600000028426203</IbanCode>
         <TransactionAmount>4450</TransactionAmount>
         <AuthorizedAmount>4450</AuthorizedAmount>
         <Currency>978</Currency>
         <Exponent>2</Exponent>
         <AccountedAmount>0</AccountedAmount>
         <RefundedAmount>100</RefundedAmount>
         <TransactionResult>00</TransactionResult>
         <Timestamp>2019-04-09T12:02:38</Timestamp>
         <AuthorizationNumber>622851</AuthorizationNumber>
         <AcquirerBIN>453997</AcquirerBIN>
         <MerchantID>000000000000476</MerchantID>
         <TransactionStatus>02</TransactionStatus>
         <ResponseCodeISO>00</ResponseCodeISO>
         <!-- This MAC signs authorization data -->
         <MAC>0EA6645D79E9752BE05800BE9CFE623CE3973395</MAC>
      </Authorization>
   </Data>
</BPWXmlResponse>
```

The meaning of the elements is the following:

**<BPWXmlResponse>**

This is the root element of the document, there is only one element of this type in the message, which consists of the following elements:

| | |
|---|---|
| **<Timestamp>** | date and time of the response message |
| **<Result>** | outcome of the requested transaction. Possible outcomes: |

| Code | Description |
|---|---|
| 00 | Success |
| 02 | ReqRefNum duplicated or incorrect |
| 03 | Incorrect message format, missing or incorrect field |
| 04 | Incorrect API authentication, incorrect MAC |
| 06 | Unforeseen error during processing of request |
| 40 | Empty Xml or missing 'data' parameter |
| 41 | Xml not parsable |
| 98 | Application error |
| 99 | Transaction failed, see specific outcome enclosed in the element <Data> of the response. |

| | |
|---|---|
| **<MAC>** | message authentication code: signature of timestamp and result (see appendix 4.2.10) |
| **<Data>** | data relating to the authorization request and response message |

### <Data>

There is only one element of this type in the message, containing all the data relating to the iban authorization request and response message and consisting of the following elements:

| | |
|---|---|
| **<IbanAuthorizationRequest>** | data relating to the authorization request |
| **<Authorization>** | data relating to the response message |

### <IbanAuthorizationRequest>

There is only one element of this type in the message, containing all the data relating to the iban authorization request and consisting of the following elements:

| | |
|---|---|
| **<Header>** | data relating to the request sent |
| **<OrderID>** | order identifier |
| **<IBAN>** | iban |
| **<Amount>** | amount of requested authorization in Euro cents |
| **<Currency>** | currency ISO code: 978=Eur |
| **<Exponent>** | number of decimals for the currency |
| **<AccountingMode>** | type of booking to be used: D=Deferred, I=Immediate |
| **<Network>** | card authorization circuit |
| **<EmailCH>** | e-mail of card holder |
| **<Userid>** | identifier of card holder |
| **<OpDescr>** | optional description that could be associated to the operation |
| **<IpAddress>** | ip address associated to the request |
| **<UsrAuthFlag>** | flag indicating the type of userid sent (for Riskshield) |
| **<Antifraud>** | antifraud data payload containing additional information used for antifraud checks |
| **<ProductRef>** | sale identifier |
| **<Name>** | cardholder's first name |
| **<Surname>** | cardholder's surname |
| **<TaxID>** | cardholder's tax ID |

### <Header>

There is only one element of this type in the message, which includes the date relating to the request sent. For the related description see paragraph 3.3.2 in the chapter  "Response messages in XML".

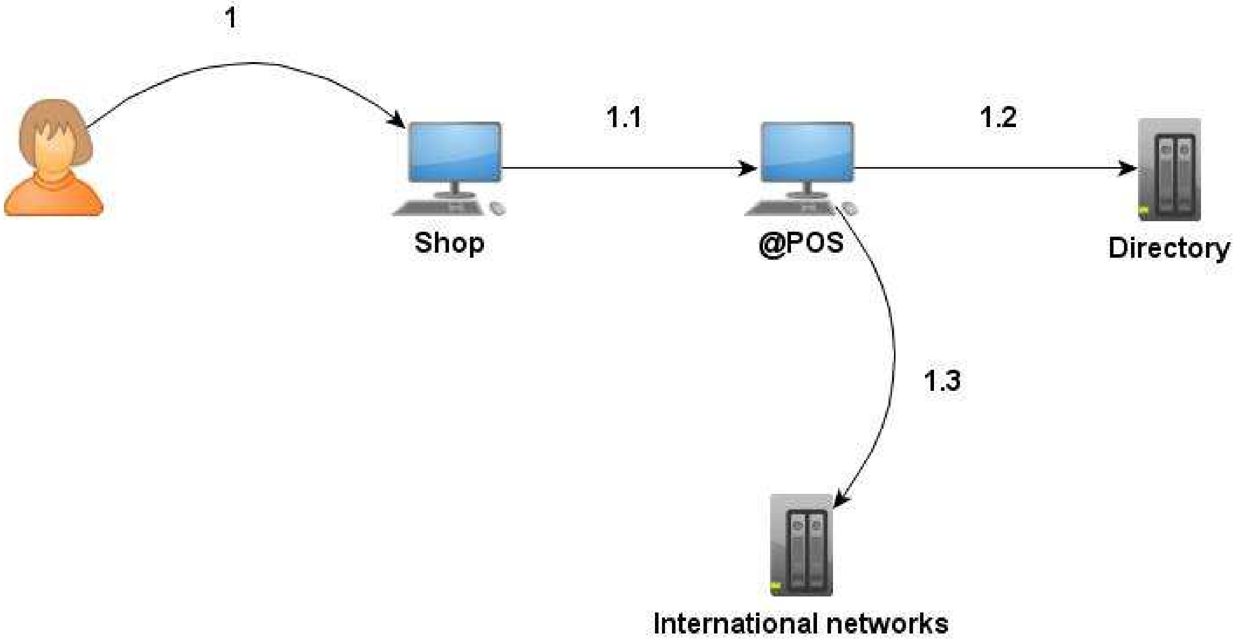
**<Authorization>**

There is only one element of this type in the message, which includes the authorization data. For the related description see the appropriate paragraph in the chapter  “Response messages in XML”

## 3.5 3DS 1.x Authorizations (Verified By Visa, SecureCode and SafeKey extension)

The Verified By Visa authorizations for Visa cards, SecureCode for Mastercard/Maestro cards and SafeKey for Amex cards (hereinafter VBV) apply only to Visa, MasterCard/Maestro and Amex cards that are enabled to such security system. Notwithstanding the foregoing, the SIA VPOS integration offers a single interface for dealing with all types of credit cards; clearly, for cards which are VBV-enabled, the system's behaviour is inevitably different.

Here below is a diagram showing the operation of the system in the two possible scenarios.



*Picture 1 - Standard Authorization*

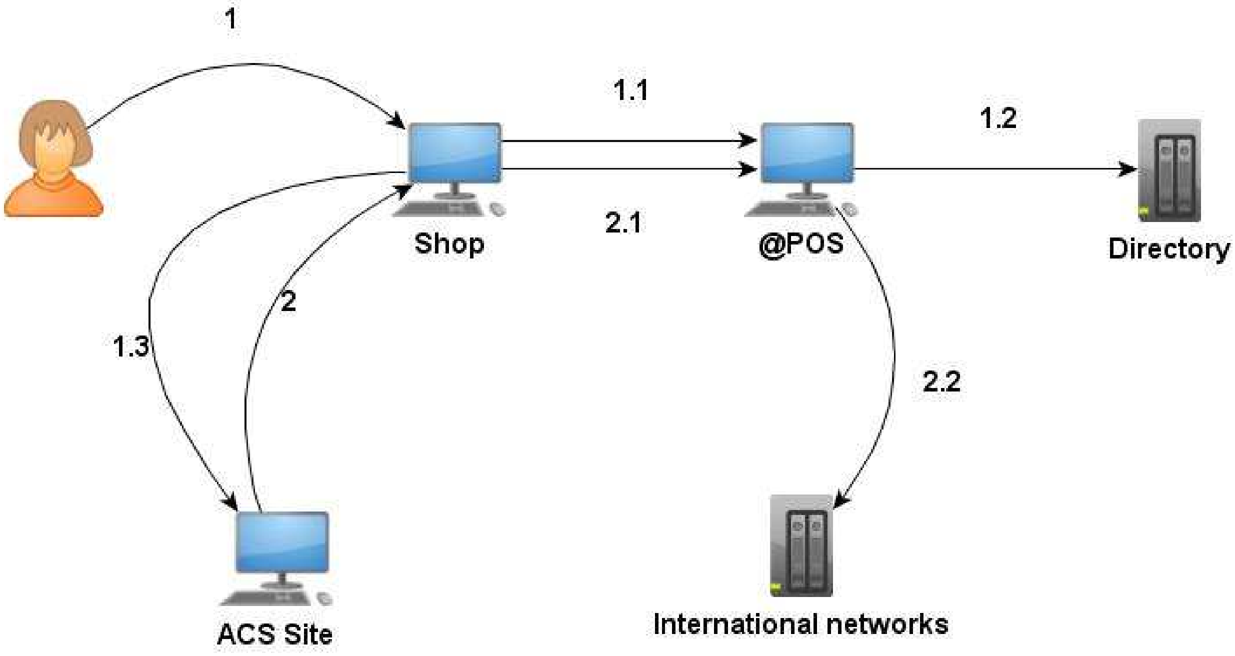
1. The user is connected to the store website and enters his credit card data.

    1.1. The store initiates the authorization request process, regardless of the card, simply by forwarding a AUTHORIZATION3DSSTEP1 message to the SIA VPOS system

    1.2. If it is indicated that the card is a Visa/MasterCard/Maestro/Amex, SIA VPOS connects to the appropriate directories and verifies whether the card is VBV enabled

    1.3. If the card is not VBV enabled (or if it is not a Visa/MasterCard/Maestro/Amex VBV or the request is of the MASTERPASS type) SIA VPOS sends an authorization message directly to the international circuits. The response given to the store website contains the transaction outcome. This completes the scenario.

If, during step 1.2 the SIA VPOS system detects that the card is VBV enabled, the following VBV scenario will be triggered:



*Picture 2 - 3DS Authorization*

1.3. If the card is VBV enabled, the response to the message of 1.1 will not contain the transaction results, but rather, the data for redirecting the user to the ACS website of the card issuer. The store creates the **MD** (merchant data) and **TermURL** (return URL) parameters and redirects the user

2. The user, while connected to the ACS website, enters his password and is redirected again to the store website bringing with him the **MD** and **PaRes** parameters.

2.1. Once the data are received from ACS, the store fills in a message of the AUTHORIZATION3DSSTEP2 type and sends it to the SIA VPOS system. **The message 2.1 must be received within 8 minutes from message 1.1**

2.2. SIA VPOS decodes the ACS data, sends the authorization request to the international circuits, and then provides the transaction outcome as a response to the AUTHORIZATION3DSSTEP2 message

In essence, the store must forward the authorization requests with the AUTHORIZATION3DSSTEP1 messages and verify the outcome of the response. If the response is of the “VBV enabled card” type, it must undertake the VBV workflow and redirect the user to the ACS website.

**The redirection in question must be performed via POST with the appropriate parameters.**
A simple HTML/Javascript example is provided in the following page.

```
<html>
<head>
<script language="Javascript">
function OnLoadEvent()
{
document.downloadForm.submit();
}
</script>
</head>
<body OnLoad="OnLoadEvent();">
<form name="downloadForm" action="${URLACS}" method="POST">
       <input type="hidden" name="PaReq" value="${PAREQ}">
       <input type="hidden" name="TermUrl" value="http://www.site.com/shop/fromacs">
       <input type="hidden" name="MD" value="base64">
</form>
</body>
</html>
```

The names **PaReq, TermUrl** and **MD** are part of the VBV standard and must be specified as such. The values included between ${} are the values transmitted by the SIA VPOS system.

**TermUrl** is the URL to which the user is redirected after VBV authentication is performed by the card issuer's ACS website. It cannot contain parameters or attributes.
**MD** stands for Merchant Data. The merchant can fill it in at will, ACS will return whatever it has received. The field is considered in Base64, hence only the characters which are part of said code are accepted.

The detailed specifications of the SIA VPOS messages and of the related responses are set out in the following paragraphs.

The full VBV process requires that the credit card holder is present online; consequently, it is not possible to grant authorizations of this type in automated "batches". Notwithstanding the foregoing, the SIA VPOS interface permits to indicate that the user is unavailable. In that case, the transaction will be completed only if the card is not VBV/SecureCode/SafeKey enabled or if the card is not a Visa/MasterCard/Maestro/Amex VBV.

## 3.5.1 3DS authorization request

The online authorization request message permits to send authorization requests to the circuits. In the case of Visa, Masterdard/Maestro or Amex VBV cards, the request can be of the VBV type.
The fields to be specified in the HTTP request message are the following:

| Field | Compulsory | Size | Type | Description |
|---|---|---|---|---|
| | | | | |
| OPERATION | Y | | A | Transaction requested: filled in with "AUTHORIZATION3DSSTEP1" |
| TIMESTAMP | Y | 23 | AN | Local timestamp of the yyyy-MM-ggTHH:mm:ss.SSS type |
| SHOPID | Y | 15 | AN | Identifier of merchant's store assigned by SIA (Merchant ID) |
| ORDERID | Y | Min. 1<br>Max.50 | AN | Unique order identifier |
| OPERATORID | Y | Min. 8<br>Max.18 | AN | It indicates the person who has requested the transaction. |
| REQREFNUM | Y | 32 | N | Unique request identifier managed by the merchant. It can be used to retrieve information on the request made also in the case of no response. The first 8 digits must be in the yyyyMMdd format, including the request date. |
| PAN | Y | Min. 10<br>Max.19 | AN | Card number |
| CVV2 | N | Min. 3<br>Max.4 | N | Control code associated to the card number (optional) |
| EXPDATE | Y | 4 | N | Card expiry date – yyMM- |
| AMOUNT | Y | Min. 2<br>Max. 8 | N | Amount expressed in the smallest currency unit (EUR cents).<br>The amount must not be preceded by zeros.<br>Minimum length 1 maximum length 8.<br><br>For ASI card verification transactions the amount MUST be set to 0. For real transactions the amount has to be at least the minimum supported by the merchant's acquirer (usually 10 cent for euro). |
| CURRENCY | Y | 3 | N | Currency: ISO code (EUR = 978). |
| EXPONENT | N | 1 | N | Exponent of the chosen Currency<br>(Recommended if the currency is different from Euro) |
| ACCOUNTINGMODE | Y | 1 | AN | Type of booking to be used for this order:<br>• D Deferred<br>• I Immediate<br>For ASI card verification transactions the accounting mode MUST be set to D. |
| NETWORK | Y | 2 | N | Card authorization circuit (e.g.: 01 for VISA). If the merchant is not in possession of the circuit code, it is possible to delegate the ATPOS system to calculate the latter, simply by setting the value of 93 as the NETWORK. |
| EMAILCH | N | Min. 7<br>Max. 50 | AN | E-mail of card holder (optional) |

| | | | | |
|---|---|---|---|---|
| USERID | N | Min. 1<br>Max. 30 | AN | Identifier of card holder |
| ACQUIRER | N | 5 | N | Code of acquirer in the transaction to be carried out |
| IPADDRESS | N | Min. 7<br>Max. 15 | AN | IP address associated to the request |
| OPDESCR | N | 100 | AN | Additional description of the operation at merchant's discretion, in case of authorization with immediate booking. If the booking is deferred, this field is skipped. |
| USRAUTHFLAG | N | 1 | AN | "0" for occasional user; "1" for registered user; "2" for unrecognized user" |
| RELEASE | Y | 2 | AN | Release of APIs: to be filled in with "02" |
| OPTIONS | N | Min 0<br>Max 26 | AN | The field OPTIONS permits to activate various additional options for the ongoing message. |
| ANTIFRAUD | N | na | AN | Antifraud data payload containing additional information used for antifraud checks. Field is mandatory if SV69 is active. |
| PRODUCTREF | N | 15 | AN | Sale identifier |
| NAME | N | 40 | AN | Cardholder's first name |
| SURNAME | N | 40 | AN | Cardholder's surname |
| TAXID | N | 16 | AN | Cardholder's tax id |
| CREATEPANALIAS | N | 1 | A | "S" to create a pan alias, *null* otherwise. |
| INPERSON | N | 1 | A | Indicates whether or not the user holder of the card is present and if the latter can therefore be involved in the VBV authentication. Possible values: S (Yes) or N (No). Default at N. |
| MERCHANTURL | N | 255 | AN | URL of the store website. This is compulsory if INPERSON S and NETWORK 01 or 02. The accepted characters are those permitted by the following regula expression: [A-Za-z0-9_\-/:. ] |
| SERVICE | N | 4 | AN | Compulsory field if we are to specify that it is an authorization request with MASTERPASS service. In that case the value of SV47 must be indicated |
| XID | N | 40 | AN | Unique identifier of transaction 3D. Format BASE 64. |
| CAVV | N | 40 | AN | Cardholder Authentication Verification Value: value generated by the issuer and present following 3D authentication performed successfully by the card holder. Format BASE 64 |
| ECI | N | 2 | N | Electronic Commerce Indicator: Present if the field PARESSTATUS has the value of "Y" or "A".<br>Possible values for Visa are:<br>1) 05 and 06 (Card Issuer Liability)<br>2) 07 (Merchant Liability)<br>Possible values for MasterCard are:<br>1) 01 (Merchant Liability)<br>2) 02 (Card Issuer Liability) |
| PP_AUTHENTICATEMETHOD | N | Min 3<br>Max 20 | AN | Authentication method used by the card holder. Possible values are:<br>1) MERCHANT ONLY: Transaction not authenticated with additional methods<br>2) 3DS: Transaction authenticated with 3DS<br>3) NO AUTHENTICATION: Non 3DS transaction |

| | | | | |
|---|---|---|---|---|
| PP_CARDENROLLMETHOD | N | Min 6<br>Max 20 | AN | Method used to validate the card and the card holder at the time the card is added to the MASTERPASS wallet. Possible values are:<br>1) Manual: Card added manually in the MASTERPASS wallet<br>2) Direct Provisioned: Card added to the MASTERPASS wallet through an Issuer<br>3) 3DS Manual: Card added manually to the MASTERPASS wallet and 3DS verified at the time it is added.<br>4) NFC Tap: MASTERPASS Card added to the wallet through an NFC-enabled Ultrabook's NFC reader. |
| PARESSTATUS | N | 1 | AN | Code of outcome of card holder authentication. Possible values are:<br>1) Y: Card holder authenticated successfully to 3D<br>2) N: Card holder authentication failed<br>3) A: Transaction with Attempt<br>4) U: Authentication results not available |
| SCENROLLSTATUS | N | 1 | AN | Indicates whether the card issuer supports 3D authentication. Possible values are:<br>1) Y: card enabled to 3D transactions<br>2) N: card not enabled to 3D transactions<br>3) U: enabling not available or not applicable to the type of card |
| SIGNATUREVERIFICATION | N | 1 | AN | Outcome of PARes signature verification. Possible values are:<br>1) Y: the PARes signature has been validated successfully<br>2) N: the PARes signature has not been validated successfully (e.g.: expired certificate) |
| TRECURR | N | 1 | AN | Type of recurring payment.<br>Mandatory for a recurring payment or with CREATEPANALIAS = 'S' (if SVA4 is not active).<br>The admitted values are:<br>R – First of a scheduled **R**ecurring transaction<br>U – First of an **U**nscheduled recurring transaction<br>C – **C**ard stored on file (pan alias/token) notification (one shot)<br>For granted authorization, in the pan alias section of the result message CRECURR will be sent back to the merchant to be used for the following recurring payments. |
| CRECURR | N | Max 50 | AN | For TRECURR=C may contain the previously received CRECURR. |
| INSTALLMENTSNUMBER | N | Min 0<br>Max 2 | N | Number of installment. Value from 0 to 99. |
| MAC | Y | 32, 40, 64 | AN | Message Authentication Code. For the related calculation see appendix 4.2.11 |

## Request of 3DS authorization in XML format

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<BPWXmlRequest>
   <Release>02</Release>
   <Request>
      <Operation>AUTHORIZATION3DSSTEP1</Operation>
      <Timestamp>2015-02-08T12:02:00.000</Timestamp>
      <MAC>115025d5a5b65df687790867bdece136</MAC>
   </Request>
   <Data>
      <AuthorizationRequest>
         <Header>
            <ShopID>000000000000003</ShopID>
            <OperatorID>oper0001</OperatorID>
            <ReqRefNum>12345678901234567890123452289000</ReqRefNum>
         </Header>
         <OrderID>1234567890</OrderID>
         <Pan>9998500000000015</Pan>
         <CVV2>123</CVV2>
         <ExpDate>0409</ExpDate>
         <Amount>4450</Amount>
         <Currency>978</Currency>
         <Exponent>2</Exponent>
         <AccountingMode>I</AccountingMode>
         <Network>01</Network>
         <EmailCH>address@yourcompany.com</EmailCH>
         <Userid>user1</Userid>
         <OpDescr>CallCenterRequest1037</OpDescr>
         <InPerson>S</InPerson>
         <MerchantURL>http://www.site.com</MerchantURL>
         <ProductRef>12345678</ProductRef>
         <Name>Jon</Name>
         <Surname>Snow</Surname>
         <TaxID>SNWJNO96A01F205L</TaxID>
         <TRecurr>C</TRecurr>
         <CRecurr>PST581426946</CRecurr>
         <InstallmentsNumber>3</InstallmentsNumber>
      </AuthorizationRequest>
   </Data>
</BPWXmlRequest>
```

In the case where the card indicated in the request is not VBV enabled (or if it is not a Visa/MasterCard/Maestro/Amex VBV or in the case of MASTERPASS requests), the system immediately forwards the transaction to the circuits. In that case the response will contain the actual results of the authorization request.

If the card is VBV enabled, the response of the system will consist of an element containing all the data necessary to the merchant's website for redirecting the acquirer to the related card issuer's ACS website for authentication.

Here below is an example of a file generated by the response to an online authorization request for a transaction carried out on a card that is not VBV enabled or which is not a Visa/Mastercard/Maestro/Amex VBV:

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<BPWXmlResponse>
   <Timestamp>2015-04-09T12:02:38</Timestamp>
   <Result>00</Result>
   <!- This MAC signs timestamp and result   -->
   <MAC>8A74330BA1A1A085581EAA2409D8DC68FCC4395E</MAC>
   <Data>
   <!- This element contains request data   -->
      <AuthorizationRequest>
         <Header>
            <ShopID>000000000000003</ShopID>
            <OperatorID>AD456123</OperatorID>
            <ReqRefNum>20150501901234567890123452289000</ReqRefNum>
         </Header>
         <OrderId>p91</OrderId>
         <Pan>999850xxxxxx0015</Pan>
         <CVV2>000</CVV2>
         <ExpDate>0409</ExpDate>
         <Amount>4450</Amount>
         <Currency>978</Currency>
         <Exponent>2</Exponent>
         <AccountingMode>I</AccountingMode>
         <Network>01</Network>
         <EmailCH>info@titolare.it</EmailCH>
         <Userid>user1</Userid>
         <OpDescr>CallCenterRequest1037</OpDescr>
         <InPerson>S</InPerson>
         <MerchantURL>http://www.sito.com</MerchantURL>
         <ProductRef>12345678</ProductRef>
         <Name>Jon</Name>
         <Surname>Snow</Surname>
         <TaxID>SNWJNO96A01F205L</TaxID>
         <TRecurr>U</TRecurr>
         <CRecurr>PST581426946</CRecurr>
         <InstallmentsNumber>3</InstallmentsNumber>
      </AuthorizationRequest>
      <Authorization>
         <PaymentType>03</PaymentType>
         <AuthorizationType>I</AuthorizationType>
         <TransactionID>8032180310AB0E30917930112</TransactionID>
         <Network>01</Network>
         <OrderID>pos91</OrderID>
         <TransactionAmount>4450</TransactionAmount>
         <AuthorizedAmount>4450</AuthorizedAmount>
         <Currency>978</Currency>
         <Exponent>2</Exponent>
         <AccountedAmount>0</AccountedAmount>
         <RefundedAmount>0</RefundedAmount>
         <TransactionResult>00</TransactionResult>
         <Timestamp>2015-04-09T12:02:38</Timestamp>
         <AuthorizationNumber>622851</AuthorizationNumber>
         <AcquirerBIN>453997</AcquirerBIN>
         <MerchantID>000000000000476</MerchantID>
         <TransactionStatus>02</TransactionStatus>
         <ResponseCodeISO>00</ResponseCodeISO>
         <!- This MAC signs authorization data -->
         <MAC>0EA6645D79E9752BE05800BE9CFE623CE3973395</MAC>
      </Authorization>
   </Data>
</BPWXmlResponse>
```

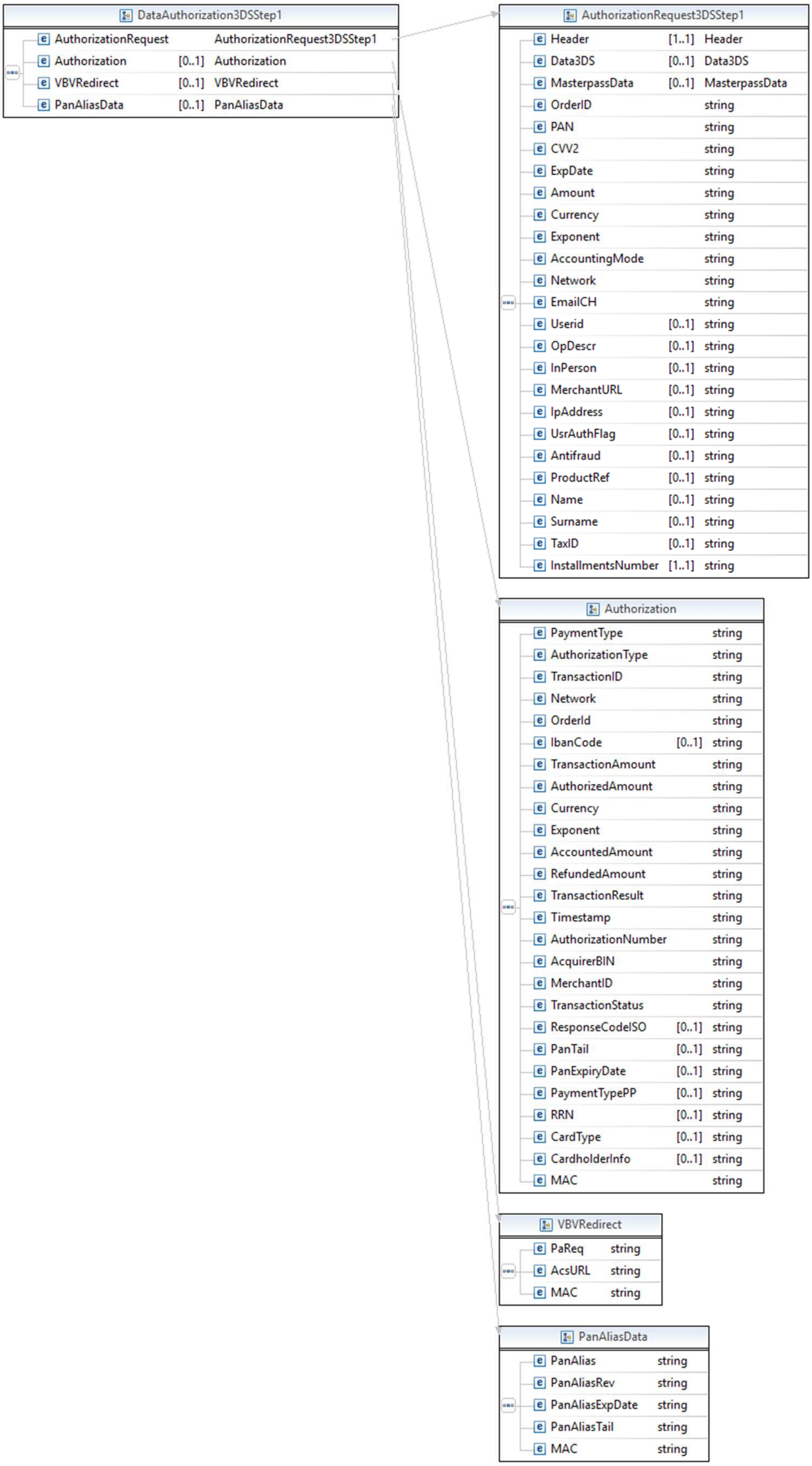
As can be noted, the response to an authorization request is essentially composed of  an Authorization-type element.

## Request of 3DS authorization in XML format with MASTERPASS service

Here below is an example of an online authorization request for a transaction with MASTERPASS service:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<BPWXmlRequest>
   <Release>02</Release>
   <Request>
      <Operation>AUTHORIZATION3DSSTEP1</Operation>
      <Timestamp>2015-02-08T12:02:00.000</Timestamp>
      <MAC>115025d5a5b65df687790867bdece136</MAC>
   </Request>
   <Data>
      <AuthorizationRequest>
         <Header>
            <ShopID>000000000000003</ShopID>
            <OperatorID>oper0001</OperatorID>
            <ReqRefNum>12345678901234567890123452289000</ReqRefNum>
         </Header>
         <Data3DS>
            <Service>SV47</Service>
            <Eci>05</Eci>
            <Xid>B8B7CBEF5344497639ECB60679A239C7507C8DE0</Xid>
            <CAVV>1700010106612300000003292061231000000000</CAVV>
            <ParesStatus>Y</ParesStatus>
            <ScEnrollStatus>Y</ScEnrollStatus>
            <SignatureVerifytion>Y</SignatureVerifytion>
         </Data3DS>
         <MasterpassData>
            <PP_AuthenticateMethod>3DS</PP_AuthenticateMethod>
            <PP_CardEnrollMethod>Direct Provisioned</PP_CardEnrollMethod>
         </MasterpassData>
         <OrderID>1234567890</OrderID>
         <Pan>9998500000000015</Pan>
         <CVV2>123</CVV2>
         <ExpDate>0409</ExpDate>
         <Amount>4450</Amount>
         <Currency>978</Currency>
         <Exponent>2</Exponent>
         <AccountingMode>I</AccountingMode>
         <Network>01</Network>
         <EmailCH>address@yourcompany.com</EmailCH>
         <Userid>user1</Userid>
         <ProductRef>12345678</ProductRef>
         <Name>Jon</Name>
         <Surname>Snow</Surname>
      </AuthorizationRequest>
   </Data>
</BPWXmlRequest>
```

The response message to the authorization request is formatted in XML. The data section is described on the following scheme.



DataAuthorization3DSStep1

| | | |
|---|---|---|
| AuthorizationRequest | | AuthorizationRequest3DSStep1 |
| Authorization | [0..1] | Authorization |
| VBVRedirect | [0..1] | VBVRedirect |
| PanAliasData | [0..1] | PanAliasData |

AuthorizationRequest3DSStep1

| | | |
|---|---|---|
| Header | [1..1] | Header |
| Data3DS | [0..1] | Data3DS |
| MasterpassData | [0..1] | MasterpassData |
| OrderID | | string |
| PAN | | string |
| CVV2 | | string |
| ExpDate | | string |
| Amount | | string |
| Currency | | string |
| Exponent | | string |
| AccountingMode | | string |
| Network | | string |
| EmailCH | | string |
| Userid | [0..1] | string |
| OpDescr | [0..1] | string |
| InPerson | [0..1] | string |
| MerchantURL | [0..1] | string |
| IpAddress | [0..1] | string |
| UsrAuthFlag | [0..1] | string |
| Antifraud | [0..1] | string |
| ProductRef | [0..1] | string |
| Name | [0..1] | string |
| Surname | [0..1] | string |
| TaxID | [0..1] | string |
| InstallmentsNumber | [1..1] | string |

Authorization

| | | |
|---|---|---|
| PaymentType | | string |
| AuthorizationType | | string |
| TransactionID | | string |
| Network | | string |
| OrderId | | string |
| IbanCode | [0..1] | string |
| TransactionAmount | | string |
| AuthorizedAmount | | string |
| Currency | | string |
| Exponent | | string |
| AccountedAmount | | string |
| RefundedAmount | | string |
| TransactionResult | | string |
| Timestamp | | string |
| AuthorizationNumber | | string |
| AcquirerBIN | | string |
| MerchantID | | string |
| TransactionStatus | | string |
| ResponseCodeISO | [0..1] | string |
| PanTail | [0..1] | string |
| PanExpiryDate | [0..1] | string |
| PaymentTypePP | [0..1] | string |
| RRN | [0..1] | string |
| CardType | [0..1] | string |
| CardholderInfo | [0..1] | string |
| MAC | | string |

VBVRedirect

| | |
|---|---|
| PaReq | string |
| AcsURL | string |
| MAC | string |

PanAliasData

| | |
|---|---|
| PanAlias | string |
| PanAliasRev | string |
| PanAliasExpDate | string |
| PanAliasTail | string |
| MAC | string |

*Scheme 8 - Data section for 3DS AUTHORIZATION STEP1response*

Here below is an example of a file generated by the response to an online authorization request for a transaction with MASTERPASS service:

```xml
<?xml version="1.0" encoding="ISO-8859-1" ?>
<BPWXmlResponse>
   <Timestamp>2015-04-09T12:02:38</Timestamp>
   <Result>00</Result>
   <!- This MAC signs timestamp and result -->
   <MAC>8A74330BA1A1A085581EAA2409D8DC68FCC4395E</MAC>
   <Data>
   <!- This element contains request data -->
      <AuthorizationRequest>
         <Header>
            <ShopID>000000000000003</ShopID>
            <OperatorID>AD456123</OperatorID>
            <ReqRefNum>20150501901234567890123452289000</ReqRefNum>
         </Header>
         <Data3DS>
            <Service>SV47</Service>
            <Eci>05</Eci>
            <Xid>B8B7CBEF5344497639ECB60679A239C7507C8DE0</Xid>
            <CAVV>1700010106612300000003292061231000000000</CAVV>
            <ParesStatus>Y</ParesStatus>
            <ScEnrollStatus>Y</ScEnrollStatus>
            <SignatureVerifytion>Y</SignatureVerifytion>
         </Data3DS>
         <MasterpassData>
            <PP_AuthenticateMethod>3DS</PP_AuthenticateMethod>
            <PP_CardEnrollMethod>Direct Provisioned
            </PP_CardEnrollMethod>
         </MasterpassData>
         <OrderID>p91</OrderID>
         <Pan>999850xxxxxx0015</Pan>
         <CVV2>000</CVV2>
         <ExpDate>0409</ExpDate>
         <Amount>4450</Amount>
         <Currency>978</Currency>
         <Exponent>2</Exponent>
         <AccountingMode>I</AccountingMode>
         <Network>01</Network>
         <EmailCH>info@titolare.it</EmailCH>
         <Userid>user1</Userid>
         <InPerson>S</InPerson>
         <MerchantURL>http://www.site.com</MerchantURL>
         <ProductRef>12345678</ProductRef>
         <Name>Jon</Name>
         <Surname>Snow</Surname>
         <TaxID>SNWJNO96A01F205L</TaxID>
         <TRecurr>U</TRecurr>
         <CRecurr>PST581426946</CRecurr>
         <InstallmentsNumber>3</InstallmentsNumber>
      </AuthorizationRequest>
      <Authorization>
         <PaymentType>03</PaymentType>
         <AuthorizationType>I</AuthorizationType>
         <TransactionID>8032180310AB0E30917930112</TransactionID>
         <Network>01</Network>
         <OrderID>pos91</OrderID>
         <TransactionAmount>4450</TransactionAmount>
         <AuthorizedAmount>4450</AuthorizedAmount>
         <Currency>978</Currency>
         <Exponent>2</Exponent>
         <AccountedAmount>0</AccountedAmount>
         <RefundedAmount>0</RefundedAmount>
         <TransactionResult>00</TransactionResult>
         <Timestamp>2015-04-09T12:02:38</Timestamp>
         <AuthorizationNumber>622851</AuthorizationNumber>
         <AcquirerBIN>453997</AcquirerBIN>
         <MerchantID>000000000000476</MerchantID>
         <TransactionStatus>02</TransactionStatus>
         <ResponseCodeISO>00</ResponseCodeISO>
         <!- This MAC signs authorization data   -->
         <MAC>0EA6645D79E9752BE05800BE9CFE623CE3973395</MAC>
      </Authorization>
      <PanAliasData>
         <PanAlias> 0000197412081271677</PanAlias>
         <PanAliasRev></PanAliasRev>
         <PanAliasExpDate>2911</PanAliasExpDate>
         <PanAliasTail>0003</PanAliasTail>
         <CRecurr>PST581426946</CRecurr>
         <MAC>E61612E0C0F71A2FE838BC0736B396E6</MAC>
      </PanAliasData>
   </Data>
</BPWXmlResponse>
```

The meaning of the elements is the following:

### <BPWXmlResponse>

This is the root element of the document, there is only one element of this type in the message, consisting of the following elements:

- **<Timestamp>** date and time of response message
- **<Result>** outcome of the requested transaction. Possible outcomes:

| Code | Description |
|---|---|
| 00 | Success |
| 02 | ReqRefNum duplicated or incorrect |
| 03 | Incorrect message format, missing or incorrect field |
| 04 | Incorrect API authentication, incorrect MAC |
| 06 | Unforeseen error during processing of request |
| 20 | The card is VBV enabled; the response contains the data for redirecting to the ACS website |
| 40 | Empty xml or missing ‘data’ parameter |
| 41 | Xml not parsable |
| 99 | Transaction failed, see specific outcome enclosed in the element <Data> of the response. |

- **<MAC>** message authentication code: signature of timestamp and result (see appendix 4.2.7)
- **<Data>** data relating to the authorization request and response message

### <Data>

There is only one element of this type in the message, containing all the data relating to the authorization request and consisting of the following elements:

- **<AuthorizationRequest3DSStep1>** data relating to the authorization request
- **<Authorization>** data relating to the response message
- **<VBVredirect>** data relating to the redirection to the issuer’s ACS
- **<PanAliasData>** data relating to the pan alias or recurring codes (if any)

### <AuthorizationRequest>

There is only one element of this type in the message, containing all the data relating to the authorization request and consisting of the following elements:

| | |
|---|---|
| **<Header>** | data relating to the request sent |
| **<OrderID>** | order identifier |
| **<Pan>** | hidden credit card number (showing only the first six and the last four digits) |
| **<CVV2>** | hidden additional card number (a sequence of zeros having a length equal to that of the field in the request) |
| **<ExpDate>** | card expiry date |
| **<Amount>** | amount of the requested authorization in Eur cents |
| **<Currency>** | currency ISO code: 978=EUR |
| **<Exponent>** | number of decimals for the currency |
| **<AccountingMode>** | type of booking to be used: D=Deferred, I=Immediate |
| **<Network>** | card authorization circuit |
| **<EmailCH>** | e-mail of card holder |
| **<Userid>** | identifier of card holder if present in the request |
| **<OpDescr>** | optional description that could be associated to the operation |
| **<InPerson>** | whether or not user is present if present in the request |
| **<MerchantURL>** | Url merchant if present in the request |
| **<IpAddress>** | ip address associated to the request |
| **<UsrAuthFlag>** | flag indicating the type of userid sent (for Riskshield) |
| **<Antifraud>** | antifraud data payload containing additional information used for antifraud checks |
| **<ProductRef>** | sale identifier |
| **<Name>** | cardholder’s first name |
| **<Surname>** | cardholder’s surname |
| **<TaxID>** | cardholder’s tax ID |
| **<TRecurr>** | type of recurring payment |
| **<CRecurr>** | recurring code |
| **<InstallmentsNumber>** | number of installments |

### <Header>

There is only one element of this type in the message, which includes the date relating to the request sent. For the related description see paragraph 3.3.2 in the chapter "Response messages in XML".

### <Authorization>

There is only one element of this type in the message, containing all the authorization data. For the related description see the appropriate paragraph in the chapter “Response messages in XML”

As set out above, the response in the case where the card entered is VBV enabled contains all the data for redirecting to the card issuer’s ACS website. More specifically:

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<BPWXmlResponse>
   <Timestamp>2015-04-09T12:02:38</Timestamp>
   <Result>20</Result>
   <!- This MAC signs timestamp and result   -->
   <MAC>8A74330BA1A1A085581EAA2409D8DC68FCC4395E</MAC>
   <Data>
      <!- This element contains request data   -->
      <AuthorizationRequest>
         <Header>
            <ShopID>000000000000003</ShopID>
            <OperatorID>AD456123</OperatorID>
            <ReqRefNum>20150501901234567890123452289000</ReqRefNum>
         </Header>
         <OrderID>
            p91
         </OrderId>
            <Pan>999850xxxxxx0015</Pan>
            <CVV2>000</CVV2>
            <ExpDate>0409</ExpDate>
            <Amount>4450</Amount>
            <Currency>978</Currency>
            <Exponent>2</Exponent>
            <AccountingMode>I</AccountingMode>
            <Network>01</Network>
            <EmailCH>info@titolare.it</EmailCH>
            <Userid>user1</Userid>
            <InPerson>S</InPerson>
            <MerchantURL>http://www.sito.com</MerchantURL>
            <ProductRef>12345678</ProductRef>
            <Name>Jon</Name>
            <Surname>Snow</Surname>
      </AuthorizationRequest>
      <!- This element contains base64 PaReq data and ACS URL -->
      <VBVRedirect>
         <PaReq>agd83kjdhs899lijsfnsky33kslmdfhkaanqcpt03hsxmcnduhasncy
            Agqposcnha830fkvkfsky33kslmdhdhgfhsdfas3hsxmcnduhdfgcy
         </PaReq>
         <AcsURL>http://www.issuer.com/acs</AcsURL>
         <!- This MAC signs VBVRedirect data -->
         <MAC>0EA6645D79E9752BE05800BE9CFE623CE3973395</MAC>
      </VBVRedirect>
   </Data>
</BPWXmlResponse>
```

The meaning of the VBVRedirect elements is set out below:

**<VBVRedirect>**

This element contains all the data relating to the VBV redirection:

- **<PaReq>** These are the VBV data in Base64 to be sent via **POST** to the card issuer's ACS website
- **<AcsURL>** this is the URL of the card issuer's ACS website. The user must be redirected to this URL according to the procedures set out above.
- **<MAC>** message authentication code: signature of the VBVRedirect element (see appendix 4.2.17)

**<PanAliasData>**

There is only one element of this type in the message, which includes the card alias data. For the related description see paragraph 3.3.5 in the chapter "Response messages in XML"

## 3.5.2 3DS authorization request step 2

The VBV online authorization request message step 2 permits to forward VBV authorization requests to the circuits once the card issuer's ACS website has obtained the user authentication. **The message AUTHORIZATION3DSSTEP2 must arrive within 8 minutes from the time the original message AUTHORIZATION3DSSTEP1 is sent.**

The fields to be specified in the HTTP request message are the following:

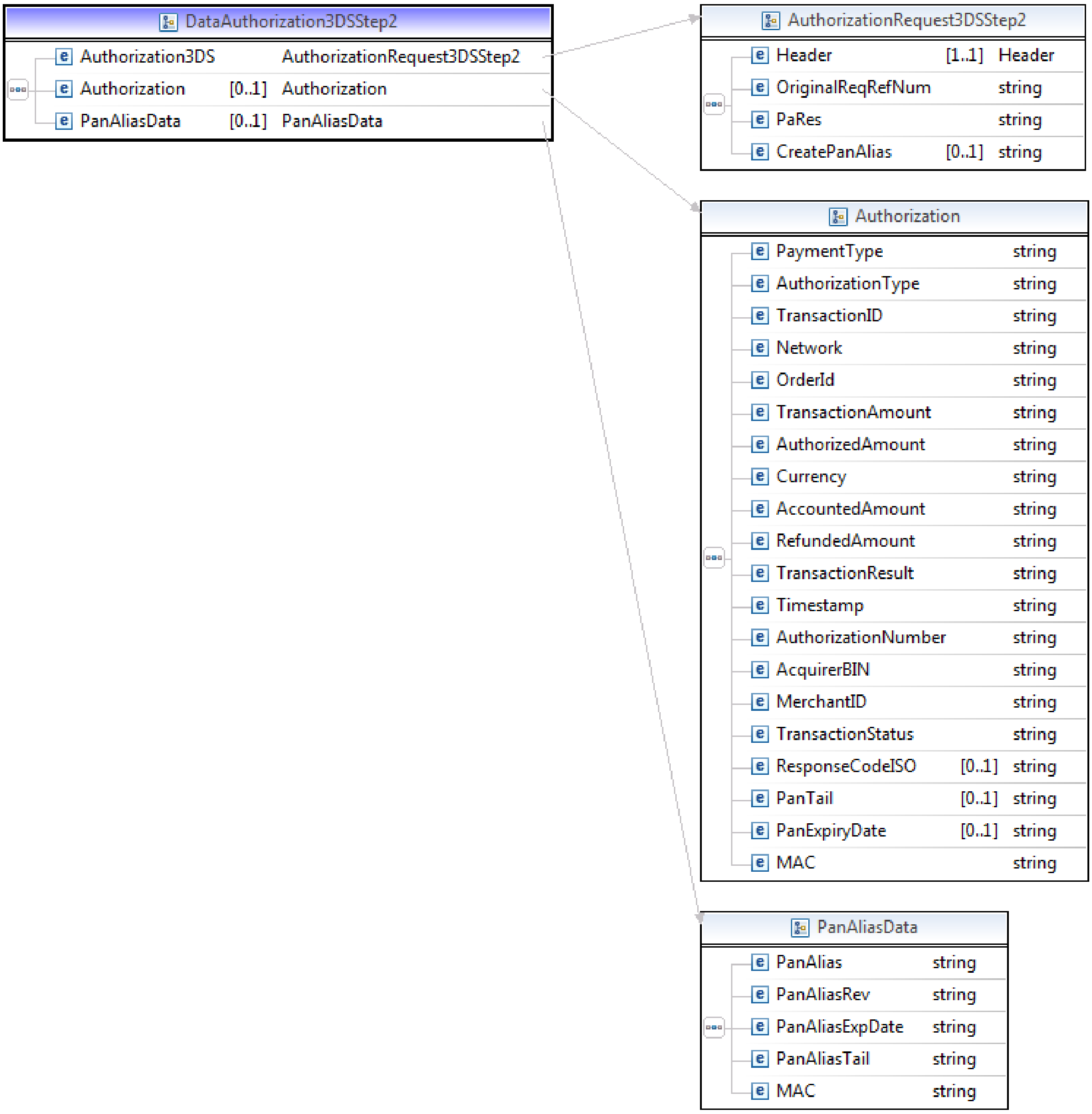
| Field | Compulsory | Size | Type | Description |
|---|---|---|---|---|
| | | | | |
| OPERATION | Y | | A | Transaction requested: filled in with "AUTHORIZATION3DSSTEP2" |
| TIMESTAMP | Y | 23 | AN | Local timestamp of the yyyy-MM-ggTHH:mm:ss.SSS type |
| SHOPID | Y | 15 | AN | Identifier of merchant's store assigned by SIA (Merchant ID) |
| OPERATORID | Y | Min. 8<br>Max.18 | AN | It indicates the person who has requested the transaction. |
| REQREFNUM | Y | 32 | N | Unique identifier of the request managed by the merchant. It can be used to retrieve information on the request made also in the case of no response. The first 8 digits must be in the yyyyMMdd format including the request date. |
| ORIGINALREQREFNUM | Y | 32 | N | Unique identifier of the original request that has produced the request for redirection to ACS. |
| PARES | Y | | | This is the PARES returned by the card issuer's ACS website after user authentication. |
| ACQUIRER | N | 5 | N | Code of acquirer in the transaction to be carried out |
| RELEASE | Y | 2 | AN | Release of APIs: to be filled in with "02" |
| OPTIONS | N | Min 0<br>Max 26 | AN | The field OPTIONS permits to activate various additional options for the ongoing message. |
| MAC | Y | 32, 40, 64 | AN | Message Authentication Code. For the related calculation see appendix 4.2.12 |

### Request of 3DS authorization step2 in XML format

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<BPWXmlRequest>
   <Release>02</Release>
   <Request>
      <Operation>AUTHORIZATION3DSSTEP2</Operation>
      <Timestamp>2015-02-08T12:02:00.000</Timestamp>
      <MAC>115025d5a5b65df687790867bdece136</MAC>
   </Request>
   <Data>
      <Authorization3DS>
         <Header>
            <ShopID>000000000000003</ShopID>
            <OperatorID>oper0001</OperatorID>
            <ReqRefNum>12345678901234567890123452289000</ReqRefNum>
         </Header>
         <OriginalReqRefNum>12345678901234567890123452289000</OriginalReqRefNum>

   <PaRes>Ssdfljlkj45098asdkgr09adsflkj9v26sfaheu73tags52gq7asgdhsdhvadghasags</PaRes>
      </Authorization3DS>
   </Data>
</BPWXmlRequest>
```

The response message to the authorization request is formatted in XML. The data section is described on the following scheme.



| DataAuthorization3DSStep2 | | |
|---|---|---|
| Authorization3DS | | AuthorizationRequest3DSStep2 |
| Authorization | [0..1] | Authorization |
| PanAliasData | [0..1] | PanAliasData |

| AuthorizationRequest3DSStep2 | | |
|---|---|---|
| Header | [1..1] | Header |
| OriginalReqRefNum | | string |
| PaRes | | string |
| CreatePanAlias | [0..1] | string |

| Authorization | | |
|---|---|---|
| PaymentType | | string |
| AuthorizationType | | string |
| TransactionID | | string |
| Network | | string |
| OrderId | | string |
| TransactionAmount | | string |
| AuthorizedAmount | | string |
| Currency | | string |
| AccountedAmount | | string |
| RefundedAmount | | string |
| TransactionResult | | string |
| Timestamp | | string |
| AuthorizationNumber | | string |
| AcquirerBIN | | string |
| MerchantID | | string |
| TransactionStatus | | string |
| ResponseCodeISO | [0..1] | string |
| PanTail | [0..1] | string |
| PanExpiryDate | [0..1] | string |
| MAC | | string |

| PanAliasData | |
|---|---|
| PanAlias | string |
| PanAliasRev | string |
| PanAliasExpDate | string |
| PanAliasTail | string |
| MAC | string |

*Scheme 9 - Data section for 3DS AUTHORIZATION STEP2 response*

The response to the message VBV step 2 is identical to the response that would be obtained from an online authorization request message for which an authorization request message has been forwarded to the circuits.
Here below is an example of a file generated by the response to the VBV online authorization request step 2:

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<BPWXmlResponse>
   <Timestamp>2015-04-09T12:02:38</Timestamp>
   <Result>00</Result>
   <!- This MAC signs timestamp and result   -->
   <MAC>8A74330BA1A1A085581EAA2409D8DC68FCC4395E</MAC>
   <Data>
      <!- This element contains request data -->
      <Authorization3DS>
         <Header>
```

```
            <ShopID>000000000000003</ShopID>
            <OperatorID>oper0001</OperatorID>
            <ReqRefNum>12345678901234567890123452289000</ReqRefNum>
         </Header>
         <OriginalReqRefNum>12345678901234567890123452289000
         </OriginalReqRefNum>
         <PaRes>Ssdfljlkj45098aadsflkj9v26sfaheu73tags52gq7asgdhsdhvadghasags
         </PaRes>
         <CreatePanAlias>S</CreatePanAlias>
      </Authorization3DS>
      <Authorization>
         <PaymentType>06</PaymentType>
         <AuthorizationType>I</AuthorizationType>
         <TransactionID>8032180310AB0E30917930112</TransactionID>
         <Network>01</Network>
         <OrderID>pos91</OrderID>
         <TransactionAmount>4450</TransactionAmount>
         <AuthorizedAmount>4450</AuthorizedAmount>
         <Currency>978</Currency>
         <Exponent>2</Exponent>
         <AccountedAmount>0</AccountedAmount>
         <RefundedAmount>100</RefundedAmount>
         <TransactionResult>00</TransactionResult>
         <Timestamp>2015-04-09T12:02:38</Timestamp>
         <AuthorizationNumber>622851</AuthorizationNumber>
         <AcquirerBIN>453997</AcquirerBIN>
         <MerchantID>000000000000476</MerchantID>
         <TransactionStatus>02</TransactionStatus>
         <ResponseCodeISO>00</ResponseCodeISO>
         <!- This MAC signs authorization data   -->
         <MAC>0EA6645D79E9752BE05800BE9CFE623CE3973395</MAC>
      </Authorization>
      <PanAliasData>
         <PanAlias> 0000197412081271677</PanAlias>
         <PanAliasRev></PanAliasRev>
         <PanAliasExpDate>2911</PanAliasExpDate>
         <PanAliasTail>0003</PanAliasTail>
         <CRecurr>PST581426946</CRecurr>
         <MAC>E61612E0C0F71A2FE838BC0736B396E6</MAC>
      </PanAliasData>
   </Data>
</BPWXmlResponse>
```

The value of 07 is added for the element Result of BPWXmlResponse, so as to obtain the following final result.

- **`<Result>`** outcome of the requested transaction. Possible outcomes:

| Code | Description |
|---|---|
| 00 | Success |
| 02 | ReqRefNum duplicated or incorrect |
| 03 | Incorrect message format, missing or incorrect field |
| 04 | Incorrect API authentication, incorrect MAC |
| 06 | Unforeseen error during processing of request |
| 07 | OriginalReqRefNum not found: it does not make reference to a VBV authorization request or too much time has passed from the original request |
| 21 | Maximum time-limit for forwarding the VBV step 2 request expired |
| 40 | Empty xml or missing ‘data’ parameter |
| 41 | Xml not parsable |

| | |
|---|---|
| 99 | Transaction failed, see specific outcome enclosed in the element <Data> of the response. |

For a description of **`<Authorization>`** see the appropriate paragraph in the chapter "Response messages in XML".

# 3.6 3DS 2.x Authorizations

Since 2019 circuits have completely rewritten the 3D Secure flow for ecommerce. Consequently a new set of four APIs has been introduced in the SIA VPOS.

1) THREEDSAUTHORIZATION0 → this is the first step of 3DS authorization.

The input of this step is the whole set of data required for a 3DS2.0 authorization.

The output of this step can be:

a) A 3DS method URL – The user has to be redirected to a silent method url provided by the customer's bank.
b) A Challenge URL – The user has to be redirected to a challenge page url provided by the customer's bank.
c) An authorization – No further steps are necessary. The authorization has been executed and its outcome is in the response.

In the first scenario the user has to be redirected to the url provided. Then the flow follows to the step 2.
In the second scenario the user has to be redirected to the url provided. The the flow follows to the step 3.

2) THREEDSAUTHORIZATION1 → this is the second step of 3DS authorization.

The input of this step is the 3ds transaction id and the output of the method url redirect.

The output of this step can be:

a) A Challenge URL – The user has to be redirected to a challenge page url provided by the customer's bank
b) An authorization – No further steps are necessary. The authorization has been executed and its outcome is in the response.

In the first scenario the user have to be redirected to the url provided.

3) THREEDSAUTHORIZATION2 → this is the third step of 3DS authorization.

The input of this step is the 3ds transaction id.

The output of this step is the authorization.

4) THREEDSVERSIONING

This is an optional API the merchant can use to know which 3DS flow is available for the customer's credit card.

## 3.6.1 3DS 2.x Authorization Request Step 0

The 3DS 2.x authorization request message permits to send authorization requests to the circuits. In the case of Visa, Mastercard/Maestro or Amex VBV cards, the request can be of the VBV type.
The fields to be specified in the HTTP request message are the following:

| Field | Compulsory | Size | Type | Description |
|---|---|---|---|---|
| | | | | |
| OPERATION | Y | | A | Transaction requested: filled in with "THREEDSAUTHORIZATION0" |
| TIMESTAMP | Y | 23 | AN | Local timestamp of the yyyy-MM-ggTHH:mm:ss.SSS type |

| | | | | |
|---|---|---|---|---|
| SHOPID | Y | 15 | AN | Identifier of merchant's store assigned by SIA (Merchant ID) |
| ORDERID | Y | Min. 1<br>Max.50 | AN | Unique order identifier |
| OPERATORID | Y | Min. 8<br>Max.18 | AN | It indicates the person who has requested the transaction. |
| REQREFNUM | Y | 32 | N | Unique request identifier managed by the merchant. It can be used to retrieve information on the request made also in the case of no response. The first 8 digits must be in the yyyyMMdd format, including the request date. |
| PAN | Y | Min. 10<br>Max.19 | AN | Card number |
| CVV2 | N | Min. 3<br>Max.4 | N | Control code associated to the card number (optional) |
| EXPDATE | Y | 4 | N | Card expiry date – yyMM- |
| AMOUNT | Y | Min. 2<br>Max. 8 | N | Amount expressed in the smallest currency unit (EUR cents).<br>The amount must not be preceded by zeros.<br>Minimum length 1 maximum length 8.<br><br>For ASI card verification transactions the amount MUST be set to 0. For real transactions the amount has to be at least the minimum supported by the merchant's acquirer (usually 10 cent for euro). |
| CURRENCY | Y | 3 | N | Currency: ISO code (EUR = 978). |
| EXPONENT | N | 1 | N | Exponent of the chosen Currency<br>(Recommended if the currency is different from Euro) |
| ACCOUNTINGMODE | Y | 1 | AN | Type of booking to be used for this order:<br>• D Deferred<br>• I Immediate<br>For ASI card verification transactions the accounting mode MUST be set to D. |
| NETWORK | Y | 2 | N | Card authorization circuit (e.g.: 01 for VISA). If the merchant is not in possession of the circuit code, it is possible to delegate the ATPOS system to calculate the latter, simply by setting the value of 93 as the NETWORK. |
| EMAILCH | Y | Min. 7<br>Max. 50 | AN | Cardholder e-mail |
| NAMECH | N | Min 2<br>Max 45 | AN | Cardholder name |
| USERID | N | Min. 1<br>Max. 30 | AN | Identifier of card holder |
| ACQUIRER | N | 5 | N | Code of acquirer in the transaction to be carried out |
| IPADDRESS | N | Min. 7<br>Max. 15 | AN | IP address associated to the request |
| USRAUTHFLAG | N | 1 | AN | "0" for occasional user; "1" for registered user; "2" for unrecognized user" |
| OPDESCR | N | 100 | AN | Additional description of the operation at merchant's discretion, in case of authorization with immediate booking. If the booking is deferred, this field is skipped. |
| RELEASE | Y | 2 | AN | Release of APIs: to be filled in with "02" |
| OPTIONS | N | Min 0<br>Max 26 | AN | The field OPTIONS permits to activate various additional options for the ongoing message. |

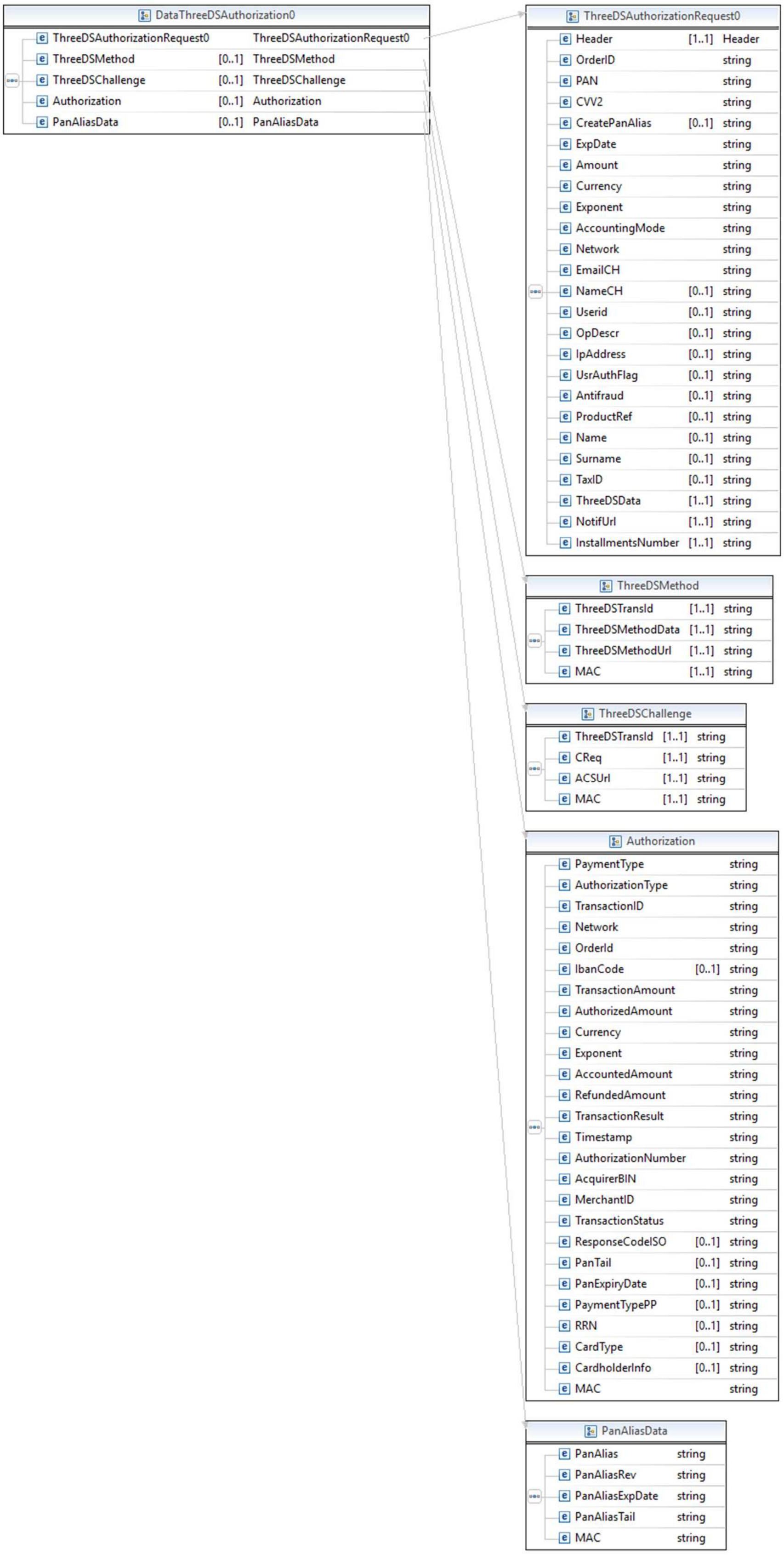
| | | | | |
|---|---|---|---|---|
| ANTIFRAUD | N | na | AN | Antifraud data payload containing additional information used for antifraud checks. Field is mandatory if SV69 is active. |
| PRODUCTREF | N | 15 | AN | Sale identifier |
| NAME | N | Min 0<br>Max 40 | AN | Optional customer first name |
| SURNAME | N | Min 0<br>Max 40 | AN | Optional customer surname |
| TAXID | N | 16 | AN | Cardholder’s tax id |
| CREATEPANALIAS | N | 1 | A | “S” to create a pan alias, *null* otherwise. |
| THREEDSDATA | Y | na | AN | 3DS data payload |
| NOTIFURL | Y | 254 | AN | URL that the final CRes (challenge response) must be sent to, via an HTTP POST |
| CPROF | N | 2 | AN | Channel for PROF Antifraud System |
| THREEDSMTDNOTIFURL | N | 256 | AN | 3DS Method notification URL |
| CHALLENGEWINSIZE | N | 2 | AN | Dimensions of the challenge window that has been displayed to the Cardholder. The ACS will reply with content that is formatted to appropriately render in this window to provide the best possible user experience.<br>Length: 2 characters<br>Value accepted:<br>01 = 250 x 400<br>02 = 390 x 400<br>03 = 500 x 600<br>04 = 600 x 400<br>05 = Full screen |
| TRECURR | N | 1 | AN | Type of recurring payment.<br>Mandatory for a recurring payment or with CREATEPANALIAS = ‘S’ (if SVA4 is not active).<br>The admitted values are:<br>R – First of a scheduled **R**ecurring transaction<br>U – First of an **U**nscheduled recurring transaction<br>C – **C**ard stored on file (pan alias/token) notification (one shot)<br>For granted authorization, in the pan alias section of the result message CRECURR will be sent back to the merchant to be used for the following recurring payments. |
| CRECURR | N | Max 50 | AN | For TRECURR=C may contain the previously received CRECURR. |
| INSTALLMENTSNUMBER | N | Min 0<br>Max 2 | N | Number of installment. Value from 0 to 99. |
| MAC | Y | 32, 40, 64 | AN | Message Authentication Code. For the related calculation see appendix 4.2.13. |

Please refer to chapter 4.4**Errore. L'origine riferimento non è stata trovata.** for more details about ThreeDSDATA field.

## Request of 3DS 2.x authorization step 0 in XML format

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<BPWXmlRequest>
   <Release>02</Release>
   <Request>
      <Operation>THREEDSAUTHORIZATION0</Operation>
      <Timestamp>2019-01-01T00:00:00.000</Timestamp>
      <MAC>115025d5a5b65df687790867bdece136</MAC>
   </Request>
   <Data>
      <ThreeDSAuthorizationRequest0>
         <Header>
            <ShopID>000000000000003</ShopID>
            <OperatorID>oper0001</OperatorID>
            <ReqRefNum>12345678901234567890123452289000</ReqRefNum>
         </Header>
         <OrderID>1234567890</OrderID>
         <Pan>9998500000000015</Pan>
         <CVV2>123</CVV2>
         <ExpDate>0409</ExpDate>
         <Amount>4450</Amount>
         <Currency>978</Currency>
         <Exponent>2</Exponent>
         <AccountingMode>I</AccountingMode>
         <Network>01</Network>
         <EmailCH>jon.snow@email.com</EmailCH>
         <NameCH>Jon Snow</NameCH>
         <Userid>user1</Userid>
         <OpDescr>CallCenterRequest1037</OpDescr>
         <ProductRef>12345678</ProductRef>
         <Name>Jon</Name>
         <Surname>Snow</Surname>
         <TaxID>SNWJNO96A01F205L</TaxID>
         <!-- base 64 encoded 3DS data -->
         <ThreeDSData>BASE64ENCODED3DSDATA=</ThreeDSData>
         <NotifUrl>https://mydomain.com/challengeNotification</NotifUrl>
         <ThreeDSMtdNotifUrl></ThreeDSMtdNotifUrl>
         <ChallengeWinSize></ChallengeWinSize>
         <TRecurr>C</TRecurr>
         <CRecurr>PST581426946</CRecurr>
         <InstallmentsNumber></InstallmentsNumber>
      </ThreeDSAuthorizationRequest0>
   </Data>
</BPWXmlRequest>
```

The response message to the authorization request is formatted in XML. The data section is described on the following scheme.



DataThreeDSAuthorization0

| | | |
|---|---|---|
| ThreeDSAuthorizationRequest0 | | ThreeDSAuthorizationRequest0 |
| ThreeDSMethod | [0..1] | ThreeDSMethod |
| ThreeDSChallenge | [0..1] | ThreeDSChallenge |
| Authorization | [0..1] | Authorization |
| PanAliasData | [0..1] | PanAliasData |

ThreeDSAuthorizationRequest0

| | | |
|---|---|---|
| Header | [1..1] | Header |
| OrderID | | string |
| PAN | | string |
| CVV2 | | string |
| CreatePanAlias | [0..1] | string |
| ExpDate | | string |
| Amount | | string |
| Currency | | string |
| Exponent | | string |
| AccountingMode | | string |
| Network | | string |
| EmailCH | | string |
| NameCH | [0..1] | string |
| Userid | [0..1] | string |
| OpDescr | [0..1] | string |
| IpAddress | [0..1] | string |
| UsrAuthFlag | [0..1] | string |
| Antifraud | [0..1] | string |
| ProductRef | [0..1] | string |
| Name | [0..1] | string |
| Surname | [0..1] | string |
| TaxID | [0..1] | string |
| ThreeDSData | [1..1] | string |
| NotifUrl | [1..1] | string |
| InstallmentsNumber | [1..1] | string |

ThreeDSMethod

| | | |
|---|---|---|
| ThreeDSTransId | [1..1] | string |
| ThreeDSMethodData | [1..1] | string |
| ThreeDSMethodUrl | [1..1] | string |
| MAC | [1..1] | string |

ThreeDSChallenge

| | | |
|---|---|---|
| ThreeDSTransId | [1..1] | string |
| CReq | [1..1] | string |
| ACSUrl | [1..1] | string |
| MAC | [1..1] | string |

Authorization

| | | |
|---|---|---|
| PaymentType | | string |
| AuthorizationType | | string |
| TransactionID | | string |
| Network | | string |
| OrderId | | string |
| IbanCode | [0..1] | string |
| TransactionAmount | | string |
| AuthorizedAmount | | string |
| Currency | | string |
| Exponent | | string |
| AccountedAmount | | string |
| RefundedAmount | | string |
| TransactionResult | | string |
| Timestamp | | string |
| AuthorizationNumber | | string |
| AcquirerBIN | | string |
| MerchantID | | string |
| TransactionStatus | | string |
| ResponseCodeISO | [0..1] | string |
| PanTail | [0..1] | string |
| PanExpiryDate | [0..1] | string |
| PaymentTypePP | [0..1] | string |
| RRN | [0..1] | string |
| CardType | [0..1] | string |
| CardholderInfo | [0..1] | string |
| MAC | | string |

PanAliasData

| | |
|---|---|
| PanAlias | string |
| PanAliasRev | string |
| PanAliasExpDate | string |
| PanAliasTail | string |
| MAC | string |

*Scheme 10 - Data section for 3DS 2.x AUTHORIZATION STEP 0 response*

The response to the 3DS 2.x authorization message step 0 can lead to three different scenarios:

- A call to the ACS 3DS method
- A challenge request
- A 3DS 2.x authorized transaction

Here below is an example of a file generated by the response to the 3DS 2.x authorization request step 0 for all the tree cases:

**3DS Method Call**

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<BPWXmlResponse>
    <Timestamp>2015-04-09T12:02:38</Timestamp>
    <Result>25</Result>
    <!-- This MAC signs timestamp and result -->
    <MAC>8A74330BA1A1A085581EAA2409D8DC68FCC4395E</MAC>
    <Data>
        <!-- This element contains request data -->
        <ThreeDSAuthorizationRequest0>
            <Header>
                <ShopID>000000000000003</ShopID>
                <OperatorID>AD456123</OperatorID>
                <ReqRefNum>20150501901234567890123452289000</ReqRefNum>
            </Header>
            <OrderID>p91</OrderID>
            <Pan>999850xxxxxx0015</Pan>
            <CVV2>000</CVV2>
            <CreatePanAlias>S</CreatePanAlias>
            <ExpDate>0409</ExpDate>
            <Amount>4450</Amount>
            <Currency>978</Currency>
             <Exponent>2</Exponent>
            <AccountingMode>I</AccountingMode>
            <Network>01</Network>
            <Userid>user1</Userid>
            <OpDescr>CallCenterRequest1037</OpDescr>
            <ProductRef>12345678</ProductRef>
            <Name>Jon</Name>
            <Surname>Snow</Surname>
            <TaxID>SNWJNO96A01F205L</TaxID>
            <ThreeDSData>BASE64ENCODED3DSDATA=</ThreeDSData>
            <NotifUrl>https://mydomain.com/challengeNotification</NotifUrl>
            <EmailCH>will.smith@email.com</EmailCH>
            <NameCH>Smith Will</NameCH>
            <ThreeDSMtdNotifUrl></ThreeDSMtdNotifUrl>
            <ChallengeWinSize></ChallengeWinSize>
            <TRecurr>C</TRecurr>
            <CRecurr>PST581426946</CRecurr>
            <InstallmentsNumber></InstallmentsNumber>
        </ThreeDSAuthorizationRequest0>
        <ThreeDSMethod>
            <!-- 3DS Server Transaction ID -->
            <ThreeDSTransId>df4b3490-db44-4a88-9619-ab173ff76fbe</ThreeDSTransId>
            <!-- base 64 encoded 3DS method data -->
            <ThreeDSMethodData>BASE64ENCODED3DSMETHODDATA=</ThreeDSMethodData>
            <ThreeDSMethodUrl>http://acsdomain.com/3dsmethod</ThreeDSMethodUrl>
            <MAC>115025d5a5b65df687790867bdece136</MAC>
        </ThreeDSMethod>
    </Data>
</BPWXmlResponse>
```

In this case, the Requestor is supposed to submit a parameter named `threeDSMethodData`, via a form in a HTML iframe within the Cardholder browser, using an HTTP POST method to the 3DS Method URL received in the `ThreeDSAuthorizationRequest0` message. If the 3DS Method does not complete within 10 seconds, merchant should set the `THREEDSMTDCOMPLIND` field to “N” in the next call to the `ThreeDSAuthorizationRequest1` message. Otherwise, if the 3DS Method does complete within 10 seconds, the Requestor shoud set `THREEDSMTDCOMPLIND` to “Y” in the next call to the `ThreeDSAuthorizationRequest1` message.

A lightweight JavaScript library (**nca-3ds-web-sdk.js**) allows Requestors to easily invoke 3DS Method messages for browser-based transactions. **nca-3ds-web-sdk.js** library can be found at the following path under API address: /editor/nca-3ds-web-sdk.js

### Challenge Requested

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<BPWXmlResponse>
    <Timestamp>2015-04-09T12:02:38</Timestamp>
    <Result>26</Result>
    <!-- This MAC signs timestamp and result -->
    <MAC>8A74330BA1A1A085581EAA2409D8DC68FCC4395E</MAC>
    <Data>
        <!-- This element contains request data -->
        <ThreeDSAuthorizationRequest0>
            <Header>
                <ShopID>000000000000003</ShopID>
                <OperatorID>AD456123</OperatorID>
                <ReqRefNum>20150501901234567890123452289000</ReqRefNum>
            </Header>
            <OrderID>p91</OrderID>
            <Pan>999850xxxxxx0015</Pan>
            <CVV2>000</CVV2>
            <CreatePanAlias>S</CreatePanAlias>
            <ExpDate>0409</ExpDate>
            <Amount>4450</Amount>
            <Currency>978</Currency>
            <Exponent>2</Exponent>
            <AccountingMode>I</AccountingMode>
            <Network>01</Network>
            <EmailCH>info@titolare.it</EmailCH>
            <Userid>user1</Userid>
            <OpDescr>CallCenterRequest1037</OpDescr>
            <ProductRef>12345678</ProductRef>
            <Name>Jon</Name>
            <Surname>Snow</Surname>
            <TaxID>SNWJNO96A01F205L</TaxID>
            <ThreeDSData>BASE64ENCODED3DSDATA=</ThreeDSData>
            <NotifUrl>https://mydomain.com/challengeNotification</NotifUrl>
            <ThreeDSMtdNotifUrl></ThreeDSMtdNotifUrl>
            <ChallengeWinSize></ChallengeWinSize>
            <TRecurr>C</TRecurr>
            <CRecurr>PST581426946</CRecurr>
            <InstallmentsNumber></InstallmentsNumber>
        </ThreeDSAuthorizationRequest0>
        <ThreeDSChallenge>
            <!-- 3DS Server Transaction ID -->
            <ThreeDSTransId>df4b3490-db44-4a88-9619-ab173ff76fbe</ThreeDSTransId>
            <!-- base 64 encoded challenge request -->
            <CReq>BASE64ENCODEDCHALLENGEREQUEST=</CReq>
            <!-- ACS challenge url -->
            <ACSUrl>http://acsdomain.com/challenge</ACSUrl>
            <MAC>115025d5a5b65df687790867bdece136</MAC>
        </ThreeDSChallenge>
    </Data>
</BPWXmlResponse>
```

In this case, the Requestor is supposed to submit a parameter named `creq`, via a form in a HTML iframe within the Cardholder browser, using an HTTP POST method to the ACS URL that was received in the response of the `ThreeDSAuthorizationRequest0` message.

At the end of challenge process, the url submitted in the `NotifUrl` field of the `ThreeDSAuthorizationRequest0` message will be called through an HTTP POST: the Requestor is supposed to invoke the final `ThreeDSAuthorizationRequest2` message.

A lightweight JavaScript library (nca-3ds-web-sdk.js) allows Requestors to easily invoke Challenge Request messages for browser-based transactions. **nca-3ds-web-sdk.js** library can be found at the following path under API address: /editor/nca-3ds-web-sdk.js

**Authorized**

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<BPWXmlResponse>
    <Timestamp>2015-04-09T12:02:38</Timestamp>
    <Result>00</Result>
    <!-- This MAC signs timestamp and result -->
    <MAC>8A74330BA1A1A085581EAA2409D8DC68FCC4395E</MAC>
    <Data>
        <!-- This element contains request data -->
        <ThreeDSAuthorizationRequest0>
            <Header>
                <ShopID>000000000000003</ShopID>
                <OperatorID>AD456123</OperatorID>
                <ReqRefNum>20150501901234567890123452289000</ReqRefNum>
            </Header>
            <OrderID>p91</OrderID>
            <Pan>999850xxxxxx0015</Pan>
            <CVV2>000</CVV2>
            <CreatePanAlias>S</CreatePanAlias>
            <ExpDate>0409</ExpDate>
            <Amount>4450</Amount>
            <Currency>978</Currency>
             <Exponent>2</Exponent>
            <AccountingMode>I</AccountingMode>
            <Network>01</Network>
            <EmailCH>info@titolare.it</EmailCH>
            <Userid>user1</Userid>
            <OpDescr>CallCenterRequest1037</OpDescr>
            <ProductRef>12345678</ProductRef>
            <Name>Jon</Name>
            <Surname>Snow</Surname>
            <TaxID>SNWJNO96A01F205L</TaxID>
            <ThreeDSData>BASE64ENCODED3DSDATA=</ThreeDSData>
            <NotifUrl>https://mydomain.com/challengeNotification</NotifUrl>
            <ThreeDSMtdNotifUrl></ThreeDSMtdNotifUrl>
            <ChallengeWinSize></ChallengeWinSize>
            <TRecurr>C</TRecurr>
            <CRecurr>PST581426946</CRecurr>
            <InstallmentsNumber></InstallmentsNumber>
        </ThreeDSAuthorizationRequest0>
        <Authorization>
            <PaymentType>03</PaymentType>
            <AuthorizationType>I</AuthorizationType>
            <TransactionID>8032180310AB0E30917930112</TransactionID>
            <Network>01</Network>
            <OrderID>pos91</OrderID>
            <TransactionAmount>4450</TransactionAmount>
            <AuthorizedAmount>4450</AuthorizedAmount>
            <Currency>978</Currency>
            <Exponent>2</Exponent>
            <AccountedAmount>0</AccountedAmount>
            <RefundedAmount>0</RefundedAmount>
            <TransactionResult>00</TransactionResult>
            <Timestamp>2015-04-09T12:02:38</Timestamp>
            <AuthorizationNumber>622851</AuthorizationNumber>
            <AcquirerBIN>453997</AcquirerBIN>
            <MerchantID>000000000000476</MerchantID>
            <TransactionStatus>02</TransactionStatus>
            <ResponseCodeISO>00</ResponseCodeISO>
            <CardholderInfo>Additional authentication is needed…</CardholderInfo>
            <!-- This MAC signs authorization data -->
            <MAC>0EA6645D79E9752BE05800BE9CFE623CE3973395</MAC>
        </Authorization>
        <PanAliasData>
            <PanAlias> 000197412081271677</PanAlias>
            <PanAliasRev></PanAliasRev>
            <PanAliasExpDate>2911</PanAliasExpDate>
            <PanAliasTail>0003</PanAliasTail>
            <CRecurr>PST581426946</CRecurr>
            <MAC>E61612E0C0F71A2FE838BC0736B396E6</MAC>
        </PanAliasData>
   </Data>
</BPWXmlResponse>
```

The meaning of the elements is the following:

### <BPWXmlResponse>

This is the root element of the document, there is only one element of this type in the message, consisting of the following elements:

- **<Timestamp>** date and time of response message
- **<Result>** outcome of the requested transaction. Possible outcomes:

| Code | Description |
|---|---|
| 00 | Success |
| 02 | ReqRefNum duplicated or incorrect |
| 03 | Incorrect message format, missing or incorrect field |
| 04 | Incorrect API authentication, incorrect MAC |
| 06 | Unforeseen error during processing of request |
| 20 | The card is VBV enabled; the response contains the data for redirecting to the ACS website |
| 25 | A call to *3DS method* must be performed by the Requestor |
| 26 | A *challenge flow* must be initiated by the Requestor |
| 40 | Empty xml or missing ‘data’ parameter |
| 41 | Xml not parsable |
| 99 | Transaction failed, see specific outcome enclosed in the element <Data> of the response. |

- **<MAC>** message authentication code: signature of timestamp and result (see appendix 4.2.7)
- **<Data>** data relating to the authorization request and response message

### <Data>

There is only one element of this type in the message, containing all the data relating to the authorization request and consisting of the following elements:

- **<ThreeDSAuthorizationRequest0>** data relating to the authorization request
- **<ThreeDSMethod>** data relating to the 3DS redirection (if any)
- **<ThreeDSChallenge>** data relating to the 3DS challenge (if any)
- **<Authorization>** data relating to the authorization (if any)
- **<PanAliasData>** data relating to the pan alias or recurring codes (if any)

### <AuthorizationRequest>

There is only one element of this type in the message, containing all the data relating to the authorization request and consisting of the following elements:

| | |
|---|---|
| **<Header>** | data relating to the request sent |
| **<OrderID>** | order identifier |
| **<Pan>** | hidden credit card number (showing only the first six and the last four digits) |
| **<CVV2>** | hidden additional card number (a sequence of zeros having a length equal to that of the field in the request) |
| **<ExpDate>** | card expiry date |
| **<Amount>** | amount of the requested authorization in Eur cents |
| **<Currency>** | currency ISO code: 978=EUR |
| **<Exponent>** | number of decimals for the currency |
| **<AccountingMode>** | type of booking to be used: D=Deferred, I=Immediate |
| **<Network>** | card authorization circuit |
| **<EmailCH>** | e-mail of card holder |
| **<NameCH>** | Name of card holder |
| **<Userid>** | identifier of card holder if present in the request |
| **<OpDescr>** | optional description that could be associated to the operation |
| **<InPerson>** | whether or not user is present if present in the request |
| **<MerchantURL>** | Url merchant if present in the request |
| **<IpAddress>** | ip address associated to the request |
| **<UsrAuthFlag>** | flag indicating the type of userid sent (for Riskshield) |
| **<Antifraud>** | antifraud data payload containing additional information used for antifraud checks |
| **<ProductRef>** | sale identifier |
| **<Name>** | cardholder’s first name |
| **<Surname>** | cardholder’s surname |
| **<TaxID>** | cardholder’s tax ID |
| **<CProf>** | Channel for PROF Antifraud System |
| **<ThreeDSMtdNotifUrl>** | Notification URL for the 3DS Method call |
| **<ChallengeWinSize>** | Code identifying the requested window size of the challenge page |
| **<TRecurr>** | type of recurring payment |
| **<CRecurr>** | recurring code |
| **<InstallmentsNumber>** | Number of installments |

### <Header>

There is only one element of this type in the message, which includes the date relating to the request sent. For the related description see paragraph 3.3.2 in the chapter "Response messages in XML".

### <Authorization>

There is only one element of this type in the message, containing all the authorization data. For the related description see the appropriate paragraph in the chapter “Response messages in XML”

Please note that, in this case, a further field named `<CardholderInfo>` could be present in the `<Authorization>` element. `<CardholderInfo>` is a text provided by the ACS/Issuer to Cardholder during a Frictionless or Decoupled transaction. The Issuer can provide information to Cardholder. For example, “Additional authentication is needed for this transaction, please contact (Issuer Name) at xxx-xxx-xxxx.”

| Field | Compulsory | Size | Type | Description |
|---|---|---|---|---|
| CardholderInfo | N | 128 chars | AN | URL encoded cardholder info |

**<ThreeDSMethod>**

There is only one element of this type in the message, containing all the 3DS method data.

| Field | Compulsory | Size | Type | Description |
|---|---|---|---|---|
| ThreeDSTransId | Y | 36 chars | UUID | New 3DS server transaction ID |
| ThreeDSMethodData | Y | NA | AN | Base 64 encoded 3DS method data to be sent to ACS 3DS method |
| ThreeDSMethodUrl | Y | NA | AN | ACS 3DS method URL |
| MAC | Y | NA | AN | Message authentication code: signature of the 3DSMethod element |

**<ThreeDSChallenge>**

There is only one element of this type in the message, containing all the challenge data.

| Field | Compulsory | Size | Type | Description |
|---|---|---|---|---|
| ThreeDSTransId | Y | 36 chars | UUID | New 3DS server transaction ID |
| CReq | Y | NA | AN | Base 64 encoded challenge request to be sent to ACS challenge URL |
| ACSUrl | Y | NA | AN | ACS 3DS challenge URL |
| MAC | Y | NA | AN | Message authentication code: signature of the 3DSChallenge element |

**<PanAliasData>**

There is only one element of this type in the message, which includes the card alias data. For the related description see paragraph 3.3.5 in the chapter "Response messages in XML"

## 3.6.2 3DS 2.x Authorization Request Step 1

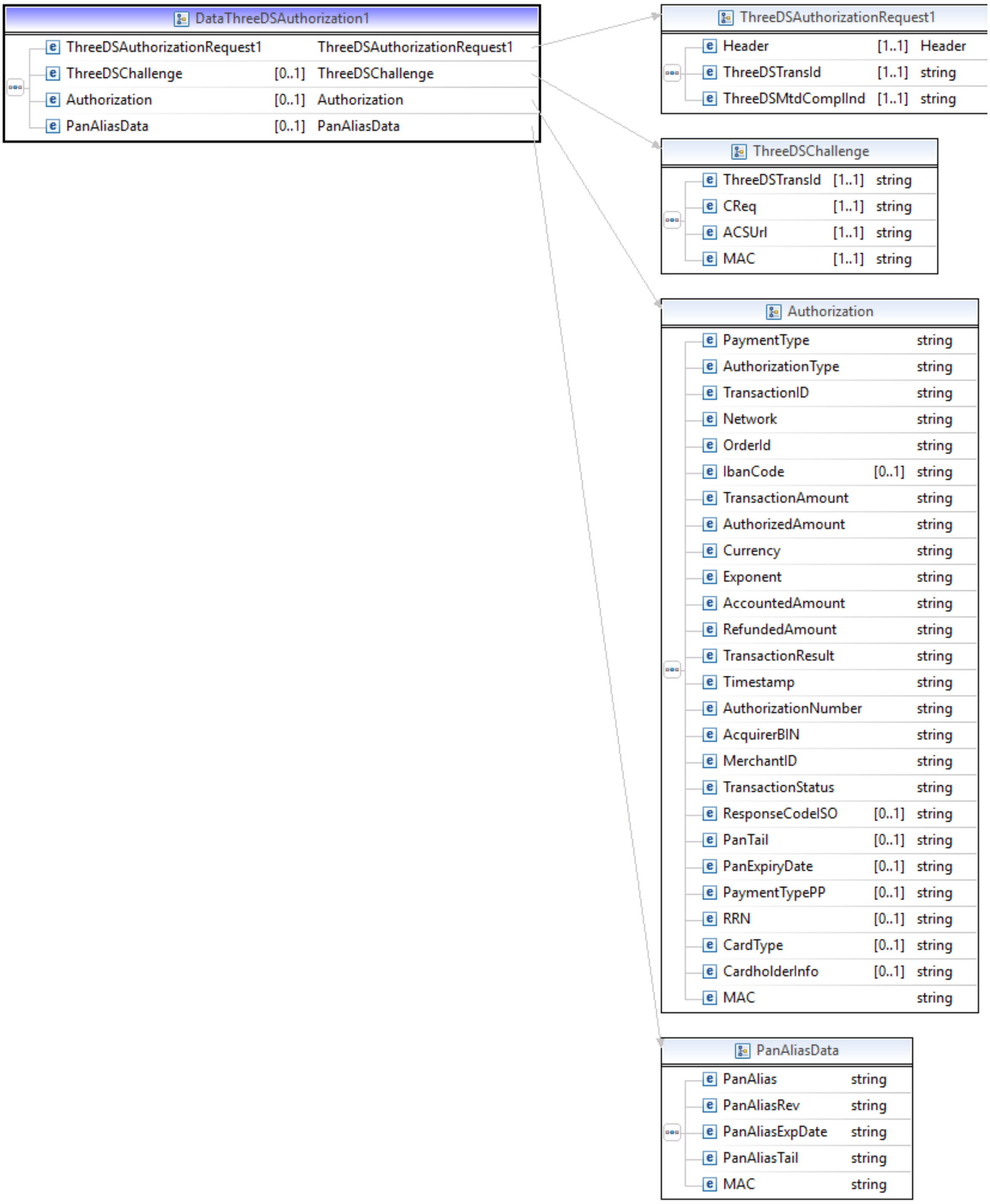
The 3DS 2.x authorization request message step 1 permits to forward authentication requests to the circuits once a call to the ACS 3DS method has been performed. **The message THREEDSAUTHORIZATION1 must arrive within 8 minutes from the time the original message THREEDSAUTHORIZATION0 is sent.**
The fields to be specified in the HTTP request message are the following:

| Field | Compulsory | Size | Type | Description |
|---|---|---|---|---|
| | | | | |
| OPERATION | Y | na | A | Transaction requested: filled in with "THREEDSAUTHORIZATION1" |
| TIMESTAMP | Y | 23 | AN | Local timestamp of the yyyy-MM-ggTHH:mm:ss.SSS type |
| SHOPID | Y | 15 | AN | Identifier of merchant's store assigned by SIA (Merchant ID) |
| OPERATORID | Y | Min. 8<br>Max.18 | AN | It indicates the person who has requested the transaction. |
| REQREFNUM | Y | | | Unique request identifier managed by the merchant. It can be used to retrieve information on the request made also in the case of no response. The first 8 digits must be in the yyyyMMdd format, including the request date. |
| RELEASE | Y | 2 | AN | Release of APIs: to be filled in with "02" |
| THREEDSTRANSID | Y | | | 3DS Server Transaction ID, obtained within the response of THREEDSAUTHORIZATION0 message |
| THREEDSMTDCOMPLIND | Y | | | 3DS Method Completion Indicator. Indicates whether the 3DS Method successfully completed:<br>• Y → Successfully completed<br>• N → Did not successfully complete |
| MAC | Y | 32, 40, 64 | AN | Message Authentication Code. For the related calculation see appendix 4.2.14 |

### Request of 3DS 2.x authorization step 1 in XML format

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<BPWXmlRequest>
   <Release>02</Release>
   <Request>
      <Operation>THREEDSAUTHORIZATION1</Operation>
      <Timestamp>2019-01-01T00:00:00.000</Timestamp>
      <MAC>115025d5a5b65df687790867bdece136</MAC>
   </Request>
   <Data>
      <ThreeDSAuthorizationRequest1>
         <Header>
            <ShopID>000000000000003</ShopID>
            <OperatorID>oper0001</OperatorID>
            <ReqRefNum>2015050190123456789012345228900</ReqRefNum>
         </Header>
         <ThreeDSTransID>df4b3490-db44-4a88-9619-ab173ff76fbe
         </ThreeDSTransID>
         <ThreeDSMtdComplInd>Y</ThreeDSMtdComplInd>
      </ThreeDSAuthorizationRequest1>
   </Data>
</BPWXmlRequest>
```

The response message to the authorization request is formatted in XML. The data section is described on the following scheme.



| DataThreeDSAuthorization1 | | |
|---|---|---|
| ThreeDSAuthorizationRequest1 | | ThreeDSAuthorizationRequest1 |
| ThreeDSChallenge | [0..1] | ThreeDSChallenge |
| Authorization | [0..1] | Authorization |
| PanAliasData | [0..1] | PanAliasData |

| ThreeDSAuthorizationRequest1 | | |
|---|---|---|
| Header | [1..1] | Header |
| ThreeDSTransId | [1..1] | string |
| ThreeDSMtdComplInd | [1..1] | string |

| ThreeDSChallenge | | |
|---|---|---|
| ThreeDSTransId | [1..1] | string |
| CReq | [1..1] | string |
| ACSUrl | [1..1] | string |
| MAC | [1..1] | string |

| Authorization | | |
|---|---|---|
| PaymentType | | string |
| AuthorizationType | | string |
| TransactionID | | string |
| Network | | string |
| OrderId | | string |
| IbanCode | [0..1] | string |
| TransactionAmount | | string |
| AuthorizedAmount | | string |
| Currency | | string |
| Exponent | | string |
| AccountedAmount | | string |
| RefundedAmount | | string |
| TransactionResult | | string |
| Timestamp | | string |
| AuthorizationNumber | | string |
| AcquirerBIN | | string |
| MerchantID | | string |
| TransactionStatus | | string |
| ResponseCodeISO | [0..1] | string |
| PanTail | [0..1] | string |
| PanExpiryDate | [0..1] | string |
| PaymentTypePP | [0..1] | string |
| RRN | [0..1] | string |
| CardType | [0..1] | string |
| CardholderInfo | [0..1] | string |
| MAC | | string |

| PanAliasData | |
|---|---|
| PanAlias | string |
| PanAliasRev | string |
| PanAliasExpDate | string |
| PanAliasTail | string |
| MAC | string |

***Scheme 11 - Data section for 3DS 2.x AUTHORIZATION STEP 1 response***

The response to the 3DS 2.x authorization message step 1 can lead to two different scenarios:

- A challenge request
- A 3DS 2.x authorized transaction

Here below is an example of a file generated by the response to the 3DS 2.x authorization request step 1 for both the cases:

**Challenge Requested**

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<BPWXmlResponse>
   <Timestamp>2015-04-09T12:02:38</Timestamp>
   <Result>26</Result>
   <!-- This MAC signs timestamp and result -->
   <MAC>8A74330BA1A1A085581EAA2409D8DC68FCC4395E</MAC>
   <Data>
      <!-- This element contains request data -->
      <ThreeDSAuthorizationRequest1>
         <Header>
            <ShopID>000000000000003</ShopID>
            <OperatorID>AD456123</OperatorID>
            <ReqRefNum>20150501901234567890123452289000</ReqRefNum>
         </Header>
         <ThreeDSTransID>df4b3490-db44-4a88-9619-ab173ff76fbe
         </ThreeDSTransID>
         <ThreeDSMtdComplInd>Y</ThreeDSMtdComplInd>
      </ThreeDSAuthorizationRequest1>
      <ThreeDSChallenge>
         <ThreeDSTransId>df4b3490-db44-4a88-9619-ab173ff76fbe
         </ThreeDSTransId>
         <!-- base 64 encoded challenge request -->
         <CReq>BASE64ENCODEDCHALLENGEREQUEST=</CReq>
         <URLAcs>http://acsdomain.com/challenge</URLAcs>
      </ThreeDSChallenge>
   </Data>
</BPWXmlResponse>
```

In this case, the Requestor is supposed to submit a parameter named `creq`, via a form in a HTML iframe within the Cardholder browser, using an HTTP POST method to the ACS URL that was received in the response of the `ThreeDSAuthorizationRequest0` message.
At the end of challenge process, the url submitted in the `NotifUrl` field of the `ThreeDSAuthorizationRequest0` message will be called through an HTTP POST: the Requestor is supposed to invoke the final `ThreeDSAuthorizationRequest2` message.

A lightweight JavaScript library (nca-3ds-web-sdk.js) allows Requestors to easily invoke Challenge Request messages for browser-based transactions. **nca-3ds-web-sdk.js** library can be found at the following path under API address: /editor/nca-3ds-web-sdk.js

**Authorized**

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<BPWXmlResponse>
    <Timestamp>2015-04-09T12:02:38</Timestamp>
    <Result>00</Result>
    <!-- This MAC signs timestamp and result -->
    <MAC>8A74330BA1A1A085581EAA2409D8DC68FCC4395E</MAC>
    <Data>
        <!-- This element contains request data -->
        <ThreeDSAuthorizationRequest1>
            <Header>
                <ShopID>000000000000003</ShopID>
                <OperatorID>AD456123</OperatorID>
                <ReqRefNum>20150501901234567890123452289000</ReqRefNum>
            </Header>
            <ThreeDSTransID>df4b3490-db44-4a88-9619-ab173ff76fbe</ThreeDSTransID>
            <ThreeDSMtdComplInd>Y</ThreeDSMtdComplInd>
        </ThreeDSAuthorizationRequest1>
        <Authorization>
            <PaymentType>03</PaymentType>
            <AuthorizationType>I</AuthorizationType>
            <TransactionID>8032180310AB0E30917930112</TransactionID>
            <Network>01</Network>
            <OrderID>pos91</OrderID>
            <TransactionAmount>4450</TransactionAmount>
            <AuthorizedAmount>4450</AuthorizedAmount>
            <Currency>978</Currency>
            <Exponent>2</Exponent>
            <AccountedAmount>0</AccountedAmount>
            <RefundedAmount>0</RefundedAmount>
            <TransactionResult>00</TransactionResult>
            <Timestamp>2015-04-09T12:02:38</Timestamp>
            <AuthorizationNumber>622851</AuthorizationNumber>
            <AcquirerBIN>453997</AcquirerBIN>
            <MerchantID>000000000000476</MerchantID>
            <TransactionStatus>02</TransactionStatus>
            <ResponseCodeISO>00</ResponseCodeISO>
            <CardholderInfo>Additional authentication is needed…</CardholderInfo>
            <!-- This MAC signs authorization data -->
            <MAC>0EA6645D79E9752BE05800BE9CFE623CE3973395</MAC>
        </Authorization>
        <PanAliasData>
            <PanAlias> 000197412081271677</PanAlias>
            <PanAliasRev></PanAliasRev>
            <PanAliasExpDate>2911</PanAliasExpDate>
            <PanAliasTail>0003</PanAliasTail>
            <CRecurr>PST581426946</CRecurr>
          <MAC>E61612E0C0F71A2FE838BC0736B396E6</MAC>
        </PanAliasData>
    </Data>
</BPWXmlResponse>
```

Please refer to chapter “Request of 3DS 2.x authorization step 0 in XML format” for more details about <ThreeDSChallenge> and <Authorization> elements.

The value of 07 is added for the element Result of BPWXmlResponse, so as to obtain the following final result.

- **<Result>** outcome of the requested transaction. Possible outcomes:

| Code | Description |
|---|---|
| 00 | Success |
| 02 | ReqRefNum duplicated or incorrect |
| 03 | Incorrect message format, missing or incorrect field |
| 04 | Incorrect API authentication, incorrect MAC |
| 06 | Unforeseen error during processing of request |
| 07 | ThreeDSTransID not found: it does not make reference to a VBV authorization request or too much time has passed from the original request |
| 21 | Maximum time-limit for forwarding the VBV step 1 request expired |
| 26 | A *challenge flow* must be initiated by the Requestor |
| 40 | Empty xml or missing ‘data’ parameter |
| 41 | Xml not parsable |
| 99 | Transaction failed, see specific outcome enclosed in the element <Data> of the response. |

## 3.6.3 3DS 2.x Authorization Request Step 2

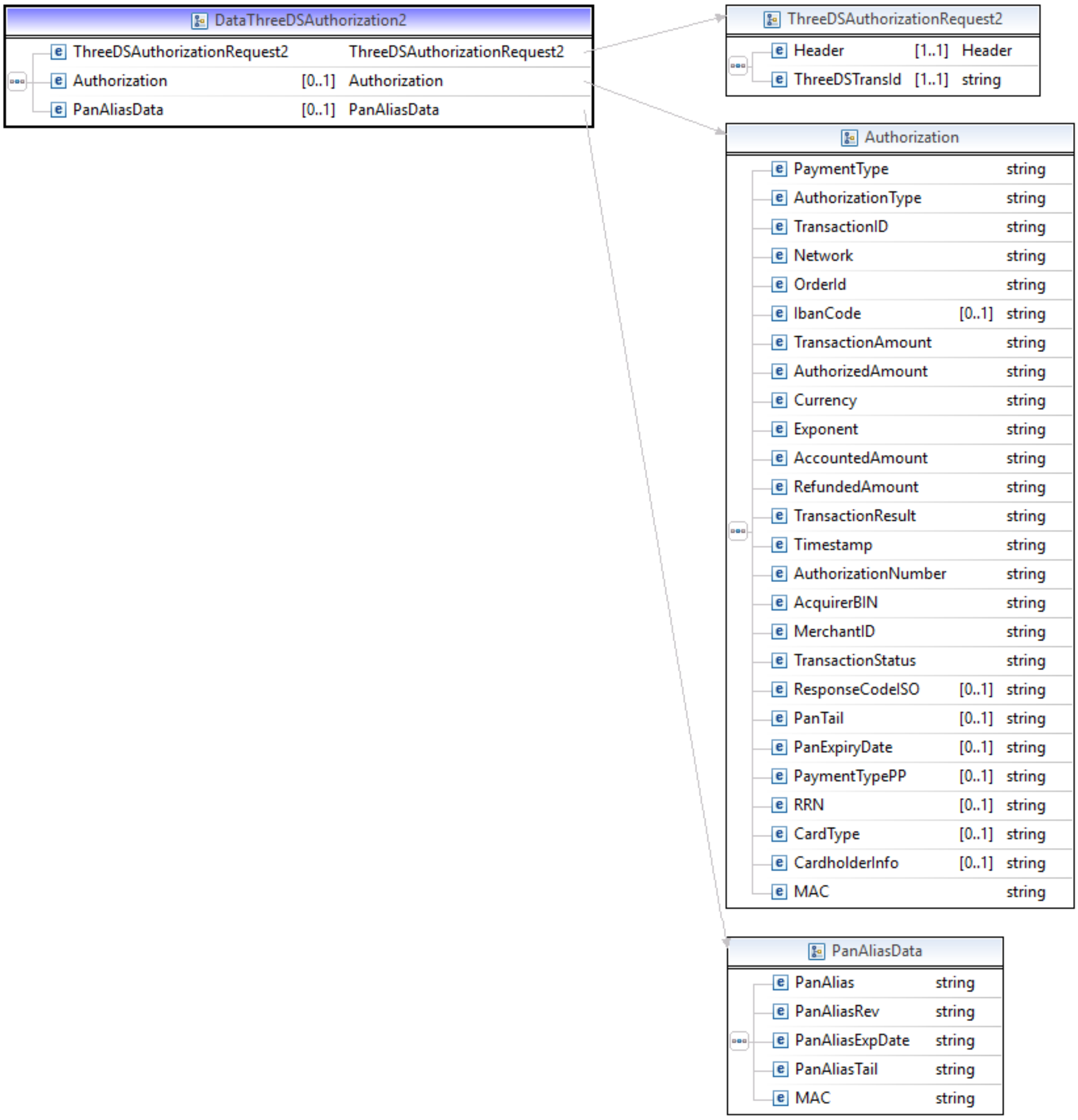
The 3DS 2.x authorization request message step 2 permits to forward authentication requests to the circuits once an user authentication challenge has been performed. **The message THREEDSAUTHORIZATION2 must arrive within 8 minutes from the time the original message THREEDSAUTHORIZATION0 or THREEDSAUTHORIZATION1 are sent.**
The fields to be specified in the HTTP request message are the following:

| Field | Compulsory | Size | Type | Description |
|---|---|---|---|---|
| | | | | |
| OPERATION | Y | na | A | Transaction requested: filled in with "THREEDSAUTHORIZATION2" |
| TIMESTAMP | Y | 23 | AN | Local timestamp of the yyyy-MM-ggTHH:mm:ss.SSS type |
| SHOPID | Y | 15 | AN | Identifier of merchant's store assigned by SIA (Merchant ID) |
| OPERATORID | Y | Min. 8 Max.18 | AN | It indicates the person who has requested the transaction. |
| REQREFNUM | Y | | | Unique request identifier managed by the merchant. It can be used to retrieve information on the request made also in the case of no response. The first 8 digits must be in the yyyyMMdd format, including the request date. |
| RELEASE | Y | 2 | AN | Release of APIs: to be filled in with "02" |
| THREEDSTRANSID | Y | | | 3DS Server Transaction ID, obtained within the response of THREEDSAUTHORIZATION0 message |
| MAC | Y | 32, 40, 64 | AN | Message Authentication Code. For the related calculation see appendix 4.2.15 |

### Request of 3DS 2.x authorization step 2 in XML format

```xml
<?xml version="1.0" encoding="ISO-8859-1"?>
<BPWXmlRequest>
   <Release>02</Release>
   <Request>
      <Operation>THREEDSAUTHORIZATION2</Operation>
      <Timestamp>2019-01-01T00:00:00.000</Timestamp>
      <MAC>115025d5a5b65df687790867bdece136</MAC>
   </Request>
   <Data>
      <ThreeDSAuthorizationRequest2>
         <Header>
            <ShopID>000000000000003</ShopID>
            <OperatorID>oper0001</OperatorID>
            <ReqRefNum>20150501901234567890123452289000</ReqRefNum>
         </Header>
         <ThreeDSTransID>df4b3490-db44-4a88-9619-ab173ff76fbe
         </ThreeDSTransID>
      </ThreeDSAuthorizationRequest2>
   </Data>
</BPWXmlRequest>
```

The response message to the authorization request is formatted in XML. The data section is described on the following scheme.



DataThreeDSAuthorization2

| Element | Occurs | Type |
|---|---|---|
| ThreeDSAuthorizationRequest2 | | ThreeDSAuthorizationRequest2 |
| Authorization | [0..1] | Authorization |
| PanAliasData | [0..1] | PanAliasData |

ThreeDSAuthorizationRequest2

| Element | Occurs | Type |
|---|---|---|
| Header | [1..1] | Header |
| ThreeDSTransId | [1..1] | string |

Authorization

| Element | Occurs | Type |
|---|---|---|
| PaymentType | | string |
| AuthorizationType | | string |
| TransactionID | | string |
| Network | | string |
| OrderId | | string |
| IbanCode | [0..1] | string |
| TransactionAmount | | string |
| AuthorizedAmount | | string |
| Currency | | string |
| Exponent | | string |
| AccountedAmount | | string |
| RefundedAmount | | string |
| TransactionResult | | string |
| Timestamp | | string |
| AuthorizationNumber | | string |
| AcquirerBIN | | string |
| MerchantID | | string |
| TransactionStatus | | string |
| ResponseCodeISO | [0..1] | string |
| PanTail | [0..1] | string |
| PanExpiryDate | [0..1] | string |
| PaymentTypePP | [0..1] | string |
| RRN | [0..1] | string |
| CardType | [0..1] | string |
| CardholderInfo | [0..1] | string |
| MAC | | string |

PanAliasData

| Element | Type |
|---|---|
| PanAlias | string |
| PanAliasRev | string |
| PanAliasExpDate | string |
| PanAliasTail | string |
| MAC | string |

*Scheme 12 - Data section for 3DS 2.x AUTHORIZATION STEP 2 response*

The response to the message is identical to the response that would be obtained from an online authorization request message for which an authorization request message has been forwarded to the circuits.

Here below is an example of a file generated by the response to the 3DS 2.x authorization request step 2:

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<BPWXmlResponse>
   <Timestamp>2015-04-09T12:02:38</Timestamp>
   <Result>00</Result>
   <!-- This MAC signs timestamp and result -->
   <MAC>8A74330BA1A1A085581EAA2409D8DC68FCC4395E</MAC>
   <Data>
      <!-- This element contains request data -->
      <ThreeDSAuthorizationRequest2>
         <Header>
            <ShopID>000000000000003</ShopID>
            <OperatorID>AD456123</OperatorID>
            <ReqRefNum>20150501901234567890123452289000</ReqRefNum>
         </Header>
         <ThreeDSTransID>df4b3490-db44-4a88-9619-ab173ff76fbe
         </ThreeDSTransID>
      </ThreeDSAuthorizationRequest2>
      <Authorization>
         <PaymentType>03</PaymentType>
         <AuthorizationType>I</AuthorizationType>
         <TransactionID>8032180310AB0E30917930112</TransactionID>
         <Network>01</Network>
         <OrderID>pos91</OrderID>
         <TransactionAmount>4450</TransactionAmount>
         <AuthorizedAmount>4450</AuthorizedAmount>
         <Currency>978</Currency>
         <Exponent>2</Exponent>
         <AccountedAmount>0</AccountedAmount>
         <RefundedAmount>0</RefundedAmount>
         <TransactionResult>00</TransactionResult>
         <Timestamp>2015-04-09T12:02:38</Timestamp>
         <AuthorizationNumber>622851</AuthorizationNumber>
         <AcquirerBIN>453997</AcquirerBIN>
         <MerchantID>000000000000476</MerchantID>
         <TransactionStatus>02</TransactionStatus>
         <ResponseCodeISO>00</ResponseCodeISO>
         <CardholderInfo>Additional authentication is needed…</CardholderInfo>
         <!-- This MAC signs authorization data -->
         <MAC>0EA6645D79E9752BE05800BE9CFE623CE3973395</MAC>
      </Authorization>
      <PanAliasData>
         <PanAlias> 000197412081271677</PanAlias>
         <PanAliasRev></PanAliasRev>
         <PanAliasExpDate>2911</PanAliasExpDate>
         <PanAliasTail>0003</PanAliasTail>
         <CRecurr>PST581426946</CRecurr>
         <MAC>E61612E0C0F71A2FE838BC0736B396E6</MAC>
      </PanAliasData>
   </Data>
</BPWXmlResponse>
```

Please refer to chapter “Request of 3DS 2.x authorization step 0 in XML format” for more details about <Authorization> element.

The value of 07 is added for the element Result of BPWXmlResponse, so as to obtain the following final result.

- **<Result>** outcome of the requested transaction. Possible outcomes:

| Code | Description |
|---|---|
| 00 | Success |
| 02 | ReqRefNum duplicated or incorrect |
| 03 | Incorrect message format, missing or incorrect field |
| 04 | Incorrect API authentication, incorrect MAC |
| 06 | Unforeseen error during processing of request |
| 07 | ThreeDSTransID not found: it does not make reference to a VBV authorization request or too much time has passed from the original request |
| 21 | Maximum time-limit for forwarding the VBV step 2 request expired |
| 40 | Empty xml or missing ‘data’ parameter |
| 41 | Xml not parsable |
| 99 | Transaction failed, see specific outcome enclosed in the element <Data> of the response. |

# 3.7 3DS Versioning request

## 3.7.1 3DS Versioning request

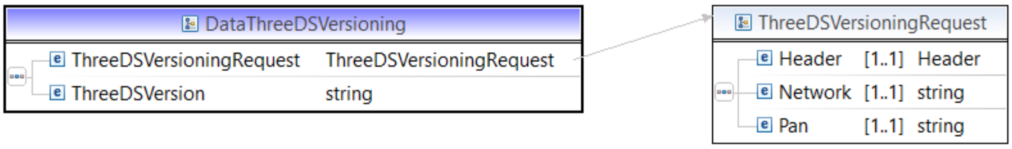
The 3DS versioning request message permits to retieve the supported protocol (1, 2, none) for a specific card number. The fields to be specified in the HTTP request message are the following:

| Field | Compulsory | Size | Type | Description |
|---|---|---|---|---|
| | | | | |
| OPERATION | Y | na | A | Transaction requested: filled in with "THREEDSVERSIONING" |
| TIMESTAMP | Y | 23 | AN | Local timestamp of the yyyy-MM-ggTHH:mm:ss.SSS type |
| SHOPID | Y | 15 | AN | Identifier of merchant’s store assigned by SIA (Merchant ID) |
| OPERATORID | Y | Min. 8 Max.18 | AN | It indicates the person who has requested the transaction. |
| REQREFNUM | Y | | | Unique request identifier managed by the merchant. It can be used to retrieve information on the request made also in the case of no response. The first 8 digits must be in the yyyyMMdd format, including the request date. |
| RELEASE | Y | 2 | AN | Release of APIs: to be filled in with “02” |
| NETWORK | Y | 2 | N | Card authorization circuit (e.g.: 01 for VISA). If the merchant is not in possession of the circuit code, it is possible to delegate the VPOS system to calculate the latter by setting as NETWORK the value of 93. |
| PAN | Y | Min. 10 Max. 19 | AN | Card number |
| MAC | Y | 32, 40, 64 | AN | Message Authentication Code. For the related calculation see appendix 4.2.16 |

### Request of 3DS versioning in XML format

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<BPWXmlRequest>
   <Release>02</Release>
   <Request>
      <Operation>THREEDSVERSIONING</Operation>
      <Timestamp>2019-01-01T00:00:00.000</Timestamp>
      <MAC>115025d5a5b65df687790867bdece136</MAC>
   </Request>
   <Data>
      <ThreeDSVersioningRequest>
         <Header>
            <ShopID>000000000000003</ShopID>
            <OperatorID>oper0001</OperatorID>
            <ReqRefNum>20150501901234567890123452289000</ReqRefNum>
         </Header>
         <Netowork>01</Netowork>
         <Pan>9998500000000015</Pan>
      </ThreeDSVersioningRequest>
   </Data>
</BPWXmlRequest>
```

The response message to the versioning request is formatted in XML. The data section is described on the following scheme.



*Scheme 13 - Data section for 3DS VERSIONING response*

Here below is an example of a file generated by the response to the 3DS versioning request:

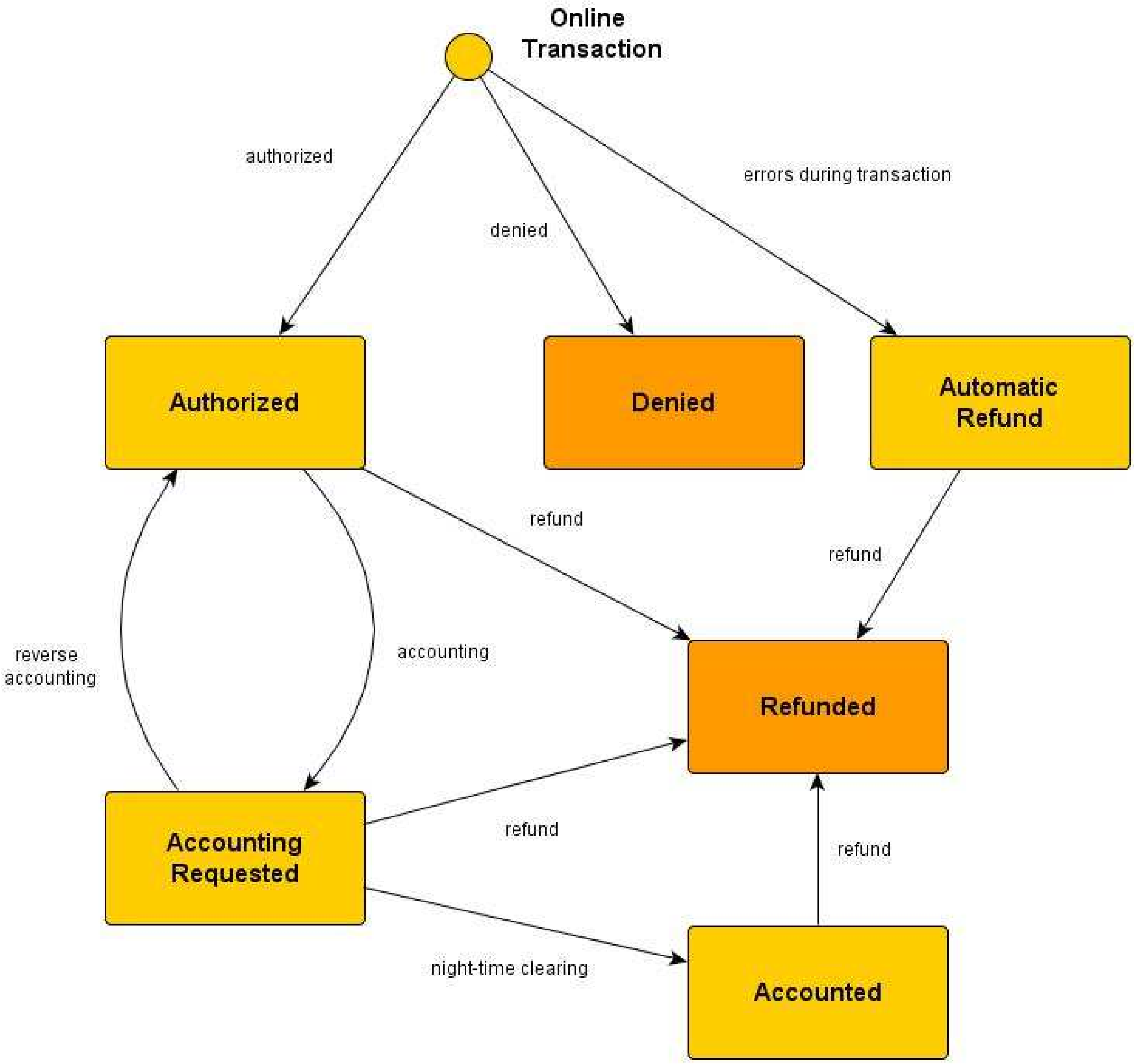
```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<BPWXmlResponse>
   <Timestamp>2015-04-09T12:02:38</Timestamp>
   <Result>00</Result>
   <!-- This MAC signs timestamp and result -->
   <MAC>8A74330BA1A1A085581EAA2409D8DC68FCC4395E</MAC>
   <Data>
      <!-- This element contains request data -->
      <ThreeDSVersioningRequest>
         <Header>
            <ShopID>000000000000003</ShopID>
            <OperatorID>AD456123</OperatorID>
            <ReqRefNum>20150501901234567890123452289000</ReqRefNum>
         </Header>
         <Netowork>01</Netowork>
         <Pan>9998500000000015</Pan>
      </ThreeDSVersioningRequest>
      <ThreeDSVersion>2</ThreeDSVersion>
   </Data>
</BPWXmlResponse>
```

The following table describes the new <ThreeDSVersion> XML element:

| Field | Compulsory | Size | Type | Description |
|---|---|---|---|---|
| ThreeDSVersion | Y | 1 char | AN | Supported protocol version. May be:<br>• 0: 3DS not supported<br>• 1: 3DS 1.x supported<br>• 2: 3DS 2.x supported |

## 3.8 Transactions on immediate authorizations

This paragraph illustrates the possible transactions on authentic authorizations (immediate authorizations). A status diagram is set out below.



***Picture 3 - Online transactions***

The online transaction is that carried out online by the merchant to the authorization circuits, or the transaction triggered by an authorization request message confirming a deferred authorization.

The possible transactions are:

- booking request
- booking cancellation request
- authorization reversal request

## 3.8.1 Booking request

A booking request transaction permits the SIA VPOS system to forward to the competent acquirer the request for the booking of an authorization previously granted with a deferred booking. Booking requests are sent to the acquirers in a batch during night processing. Booking requests for the current day can be forwarded until 12:00 p.m.. Booking requests relate to payments made via credit card.

For ASI card verification transactions the booking request cannot be submitted.

The fields to be specified in the HTTP request message are the following:

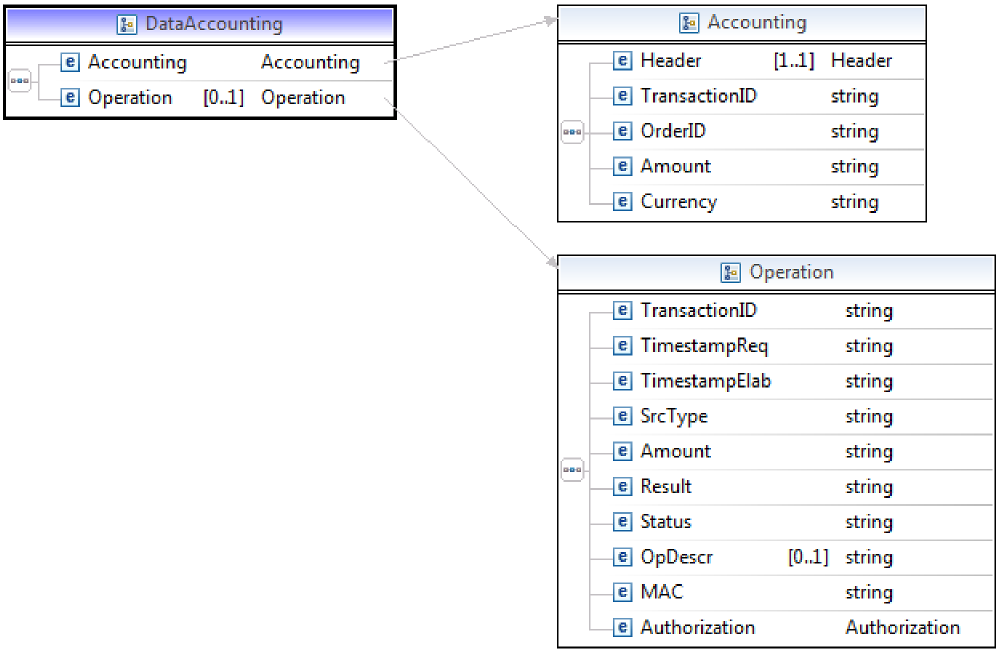
| Field | Compulsory | Size | Type | Description |
|---|---|---|---|---|
| | | | | |
| OPERATION | Y | | A | Transaction requested: filled in with "ACCOUNTING" |
| TIMESTAMP | Y | 23 | AN | Local timestamp of the yyyy-MM-ddTHH:mm:ss.SSS type |
| SHOPID | Y | 15 | AN | Identifier of the merchant's store assigned by SIA, Merchant ID (MID) |
| OPERATORID | Y | Min. 8<br>Max.18 | AN | It indicates the person who has requested the transaction. |
| REQREFNUM | Y | 32 | N | Unique identifier of the request managed by the merchant. It can be used to retrieve information on the request made also in the case of no response. The first 8 digits must be in the yyyyMMdd format including the request date. |
| TRANSACTION ID | Y | 25 | AN | Identifier of the authorization transaction carried out by the customer |
| ORDERID | Y | Min.1<br>Max.50 | AN | Unique order identifier corresponding to the TRANSACTIONID entered |
| AMOUNT | Y | Min.2<br>Max.8 | N | Amount expressed in the smallest currency unit<br>(EUR cents) |
| CURRENCY | Y | 3 | N | Currency: ISO code (EUR = 978) |
| EXPONENT | N | 1 | N | Exponent of the chosen Currency<br>(Recommended if the currency is different from Euro) |
| OPDESCR | N | 100 | AN | Optional description of the Operation at merchant's discretion |
| RELEASE | Y | 2 | AN | Release of APIs: to be filled in with "02" |
| OPTIONS | N | Min 0<br>Max 26 | AN | The field OPTIONS permits to activate various additional options for the ongoing message. |
| MAC | Y | 32, 40, 64 | AN | Message Authentication Code. For the related calculation see appendix 4.2.2 |

### Request of accounting in XML format

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<BPWXmlRequest>
   <Release>02</Release>
   <Request>
      <Operation>ACCOUNTING</Operation>
      <Timestamp>2015-03-04T11:20:00.000</Timestamp>
      <MAC>115025d5a5b65df687790867bdece136</MAC>
   </Request>
   <Data>
      <Accounting>
```

```
         <Header>
            <ShopID>000000000000003</ShopID>
            <OperatorID>oper0001</OperatorID>
            <ReqRefNum>201512121234567890123467879000000</ReqRefNum>
         </Header>
         <TransactionID>1234567890</TransactionID>
         <OrderID>9998500000000015</OrderID>
         <Amount>7700</Amount>
         <Currency>978</Currency>
         <Exponent>2</Exponent>
         <OpDescr>CallCenterRequest1037</OpDescr>
      </Accounting>
   </Data>
</BPWXmlRequest>
```

The response message to the booking request is formatted in XML. The data section is described on the following scheme.



***Scheme 14 - Data section for ACCOUNTING response***

The response to a booking request consists of an Operation-type element containing the data relating to the effected transaction.
If the TRANSACTIONID of the original transaction does not exist, or if an authentication error occurs, the element Operation will not be created.

Here below is an example of a file generated by the response to a booking request:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<BPWXmlResponse>
   <Timestamp>2015-07-04T12:02:55</Timestamp>
   <Result>00</Result>
   <!-- This MAC signs timestamp and result -->
   <MAC>ffb3553a4ab34c3a4ab34c3a4ab34c3a</MAC>
   <Data>
      <!-- This element contains request data -->
      <Accounting>
         <Header>
            <ShopID>23486788</ShopID>
            <OperatorID>A4348B</OperatorID>
            <ReqRefNum>201505014962046909345843058345 64</ReqRefNum>
         </Header>
         <TransactionID>C3956432584575645645656366</TransactionID>
         <OrderID>A398459</OrderID>
         <Amount>7000</Amount>
         <Currency>978</Currency>
         <Exponent>2</Exponent>
      </Accounting>
      <Operation>
         <TransactionID>C9435879294</TransactionID>
         <TimestampReq>2015-07-04T12:02:55</TimestampReq>
         <TimestampElab>NULL</TimestampElab>
         <SrcType>20</SrcType>
         <Amount>7000</Amount>
         <Result>00</Result>
         <Status>03</Status>
         <OpDescr>CallCenterRequest1037</OpDescr>
         <!-- This MAC signs operation data  -->
         <MAC>12334c3a4ab34c3a4ab34c3a4ab3ffa1</MAC>
         <Authorization>
            <PaymentType>03</PaymentType>
            <AuthorizationType>I</AuthorizationType>
            <TransactionID> C3956456584575645645656366</TransactionID>
            <Network>01</Network>
            <OrderID>A398459</OrderID>
            <TransactionAmount>10000</TransactionAmount>
            <AuthorizedAmount>10000</AuthorizedAmount>
            <Currency>978</Currency>
            <Exponent>2</Exponent>
            <AccountedAmount>7000</AccountedAmount>
            <RefundedAmount>100</RefundedAmount>
            <TransactionResult>00</TransactionResult>
            <Timestamp>2015-07-09T21:05:44</Timestamp>
            <AuthorizationNumber>A93485</AuthorizationNumber>
            <AcquirerBIN>123450943</AcquirerBIN>
            <MerchantID>09834509</MerchantID>
            <TransactionStatus>01</TransactionStatus>
            <ResponseCodeISO>00</ResponseCodeISO>
            <!-- This MAC signs the authorization  -->
            <MAC>4ab34c3a4ab34c3a4ab34c3a4ab34c3a</MAC>
         </Authorization>
      </Operation>
   </Data>
</BPWXmlResponse>
```

The meaning of the elements is the following:

**<BPWXmlResponse>**

This is the root element of the document, there is only one element of this type in the message, which consists of the following elements:

- **<Timestamp>** date and time of response message
- **<Result>** outcome of transaction requested

| Code | Description |
|---|---|
| 00 | Success |
| 02 | ReqRefNum duplicated or incorrect |
| 03 | Incorrect message format, missing or incorrect field |
| 04 | Incorrect API authentication, incorrect MAC |
| 06 | Unforeseen error during processing of request |
| 07 | TransactionID not found |
| 40 | Empty xml or missing ‘data’ parameter |
| 41 | Xml not parsable |

| | |
|---|---|
| 99 | Transaction failed, see specific outcome enclosed in the element <Data> of the response. |

| | |
|---|---|
| • **<MAC>** | message authentication code: signature of timestamp and result. See appendix 4.2.7 |
| • **<Data>** | data relating to the authorization request and response message |

**<Data>**

There is only one element of this type in the message, containing all the data relating to the booking request and to the response message and consisting of the following elements:

| | |
|---|---|
| • **<Accounting>** | data relating to the booking request |
| • **<Operation>** | data relating to the accounting transaction |

**<Accounting>**

There is only one element of this type in the message, containing all the data relating to the booking request and consisting of the following elements:

| | |
|---|---|
| • **<Header>** | data relating to the request sent |
| • **<TransactionID>** | identifier of the booking request transaction |
| • **<OrderID>** | order code |
| • **<Amount>** | amount of the requested authorization in Eur cents |
| • **<Currency>** | currency ISO code: 978=EUR |
| • **<Exponent>** | number of decimals for the currency |

**<Header>**

There is only one element of this type in the message, which includes the date relating to the request sent. For the related description see paragraph 3.3.2 in the chapter "Response messages in XML".

**<Operation>**

This element contains all the data relating to the accounting transaction carried out. For a detailed description see the chapter “Response messages in XML”

## 3.8.2 Cancellation of booking request

Booking request cancellation transactions may occur within 12:00 p.m. of the day on which the related request has been forwarded. This transaction cancels the booking request and makes the authorization again bookable. Booking cancellation requests relate to payments made using a credit card.

For ASI card verification transactions the request of cancellation of booking cannot be submitted.

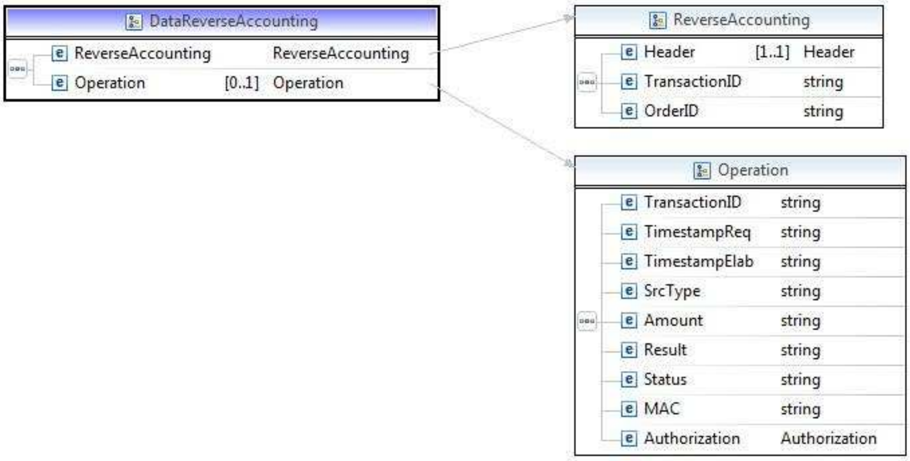
The fields to be specified in the HTTP request message are the following:

| Field | Compulsory | Size | Type | Description |
|---|---|---|---|---|
| | | | | |
| OPERATION | Y | | | Transaction requested: filled in with "REVERSEACCOUNTING" |
| TIMESTAMP | Y | 23 | AN | Local timestamp of the yyyy-MM-ddTHH:mm:ss.SSS type |
| SHOPID | Y | 15 | AN | Identifier of the merchant’s store assigned by SIA, Merchant ID (MID) |
| OPERATORID | Y | Min. 8<br>Max.18 | AN | It indicates the person who has requested the transaction. |
| REQREFNUM | Y | 32 | N | Unique identifier of the request managed by the merchant. It can be used to retrieve information on the request made also in the case of no response. The first 8 digits must be in the yyyyMMdd format including the request date. |
| TRANSACTION ID | Y | 11 | AN | Identifier of the booking transaction carried out by the customer |
| ORDERID | Y | Min.1<br>Max.50 | AN | Unique order identifier corresponding to the TRANSACTIONID entered |
| RELEASE | Y | 2 | AN | Release of APIs: to be filled in with “02” |
| OPTIONS | N | Min 0<br>Max 26 | AN | The field OPTIONS permits to activate various additional options for the ongoing message. |
| MAC | Y | 32, 40,<br>64 | AN | Message Authentication Code. For the related calculation see appendix 4.2.3 |

### Request of cancellation of a booking request in XML format

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<BPWXmlRequest>
   <Release>02</Release>
   <Request>
      <Operation>REVERSEACCOUNTING</Operation>
      <Timestamp>2015-03-04T11:20:00.000</Timestamp>
      <MAC>115025d5a5b65df687790867bdece136</MAC>
   </Request>
   <Data>
      <ReverseAccounting>
         <Header>
            <ShopID>000000000000003</ShopID>
            <OperatorID> oper0001</OperatorID>
            <ReqRefNum>12345678901234567890123456789000</ReqRefNum>
         </Header>
         <TransactionID>1234567890</TransactionID>
         <OrderID>9998500000000015</OrderID>
      </ReverseAccounting>
   </Data>
</BPWXmlRequest>
```

The response message to the booking cancellation request is formatted in XML. The data section is described on the following scheme.



*Scheme 15 - Data section for REVERSE ACCOUNTING response*

The response to a booking request consists of a Operation element, containing the data of the transaction carried out.
If the TRANSACTIONID of the original transaction does not exist, or if an authentication error occurs, the Operation element will not be created.

Here below is an example of a file generated by the response to a booking cancellation request:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<BPWXmlResponse>
   <Timestamp>2015-07-04T12:02:55</Timestamp>
   <Result>00</Result>
   <!-- This MAC signs timestamp and result -->
   <MAC>ffb3553a4ab34c3a4ab34c3a4ab34c3a</MAC>
   <Data>
      <!-- This element contains request data -->
      <ReverseAccounting>
         <Header>
            <ShopID>23486788</ShopID>
            <OperatorID>A4348B</OperatorID>
            <ReqRefNum>20150501496204690934584305834564</ReqRefNum>
         </Header>
         <TransactionID> C9435879294</TransactionID>
         <OrderID>A398459</OrderID>
      </ReverseAccounting>
      <Operation>
         <TransactionID>C5555358792</TransactionID>
         <TimestampReq>2015-07-04T22:02:55</TimestampReq>
         <TimestampElab>NULL</TimestampElab>
         <SrcType>40</SrcType>
         <Amount>7000</Amount>
         <Result>00</Result>
         <Status>SGN03</Status>
         <!-- This MAC signs operation data  -->
         <MAC>12334c3a4ab34c3a4ab34c3a4ab3ffa1</MAC>
         <Authorization>
            <PaymentType>03</PaymentType>
```

```
           <AuthorizationType>I</AuthorizationType>
           <TransactionID> C395645658457564564565636</TransactionID>
           <Network>01</Network>
           <OrderID>A398459</OrderID>
           <TransactionAmount>10000</TransactionAmount>
           <AuthorizedAmount>10000</AuthorizedAmount>
           <Currency>978</Currency>
           <Exponent>2</Exponent>
           <AccountedAmount>0</AccountedAmount>
           <RefundedAmount>0</RefundedAmount>
           <TransactionResult>00</TransactionResult>
           <Timestamp>2015-07-09T21:05:44</Timestamp>
           <AuthorizationNumber>A93485</AuthorizationNumber>
           <AcquirerBIN>123450943</AcquirerBIN>
           <MerchantID>09834509</MerchantID>
           <TransactionStatus>01</TransactionStatus>
           <ResponseCodeISO>00</ResponseCodeISO>
           <!-- This MAC signs the authorization  -->
           <MAC>4ab34c3a4ab34c3a4ab34c3a4ab34c3a</MAC>
        </Authorization>
      </Operation>
   </Data>
</BPWXmlResponse>
```

The meaning of the elements is the following:

**<BPWXmlResponse>**

This is the root element of the document, there is only one element of this type in the message, which consists of the following elements:

- **<Timestamp>** date and time of response message
- **<Result>** outcome of transaction requested

| Code | Description |
|---|---|
| 00 | Success |
| 02 | ReqRefNum duplicated or incorrect |
| 03 | Incorrect message format, missing or incorrect field |
| 04 | Incorrect API authentication, incorrect MAC |
| 05 | Incorrect date or indicated period is empty |
| 06 | Unforeseen error during processing of request |
| 07 | TransactionID not found |
| 40 | Empty xml or missing ‘data’ parameter |
| 41 | Xml not parsable |
| 99 | Transaction failed, see specific outcome enclosed in the element <Data> of the response. |

- **<MAC>** message authentication code: signature of timestamp and result. See appendix 4.2.7
- **<Data>** data relating to authorization request and response message

**<Data>**

There is only one element of this type in the message, containing all the data relating to the authorization request and response message and consisting of the following elements:

- **<ReverseAccounting>** data relating to the booking cancellation request
- **<Operation>** data relating to the booking transaction

**<ReverseAccounting>**

There is only one element of this type in the message, containing all the data of the booking cancellation request and consisting of the following elements:

| | |
|---|---|
| • **`<Header>`** | data relating to the request sent |
| • **`<TransactionID>`** | identifier of the booking cancellation request transaction |
| • **`<OrderID>`** | order code |

### <Header>

There is only one element of this type in the message, which includes the date relating to the request sent. For the related description see paragraph 3.3.2 in the chapter "Response messages in XML".

### <Operation>

This element includes all the data relating to the booking transaction carried out. For a detailed description see the chapter “Response messages in XML”

## 3.8.3 Payment reversal request

Payment reversal requests are applied by the SIA VPOS system to granted authorizations. The transactions behind  this transaction differ depending on the status of the authorization being processed. If the authorization has not yet been booked, a ceiling restoration transaction will take place; if the authorization has been booked on the current day, and hence, has not yet been sent to the acquirer, the ceiling restoration and booking cancellation transaction will take place. If the authorization has already been booked by the acquirer, the ceiling restoration and card holder crediting transactions will take place.
After a partial payment reversal, it will be possible to perform only as many partial reversals up to the maximum reversable amount established. In that case, multiple reversals will occur. Multiple reversals are not permitted on debit circuits (Pagobancomat).

For ASI card verification transactions the refund request only updates the internal state of the transaction on the virtual pos. No other actions will be taken.

The fields to be specified in the HTTP request message are the following:

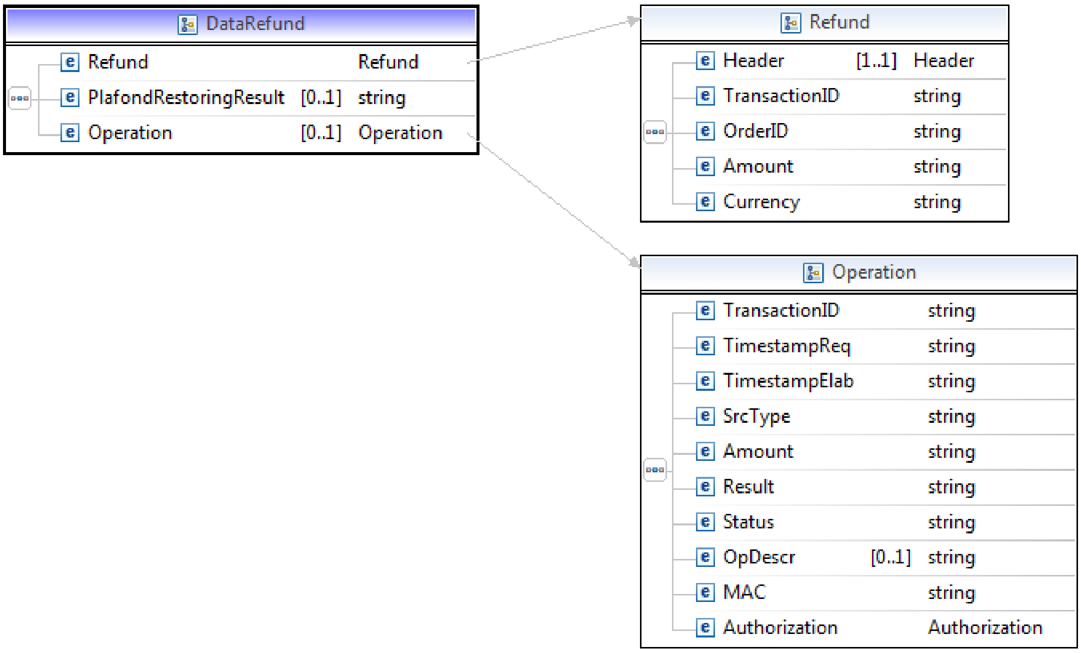
| Field | Compulsory | Size | Type | Description |
|---|---|---|---|---|
| | | | | |
| OPERATION | Y | | A | Transaction requested: filled in with "REFUND" |
| TIMESTAMP | Y | 23 | AN | Local timestamp of the yyyy-MM-ddTHH:mm:ss.SSS type |
| SHOPID | Y | 15 | AN | Identifier of the merchant's store assigned by SIA, Merchant ID (MID) |
| OPERATORID | Y | Min. 8<br>Max.18 | AN | It indicates the person who has requested the transaction. |
| REQREFNUM | Y | 32 | N | Unique identifier of the request managed by the merchant. It can be used to retrieve information on the request made also in the case of no response. The first 8 digits must be in the yyyyMMdd format including the request date. |
| TRANSACTION ID | Y | 25 | AN | Identifier of the authorization transaction on which the reversal is to be made |
| ORDERID | Y | Min.1<br>Max.50 | AN | Unique order identifier corresponding to the TRANSACTIONID entered |
| AMOUNT | Y | Min.2<br>Max.8 | N | Amount to be reversed expressed in the smallest currency unit (EUR cents) |
| CURRENCY | Y | 3 | N | Currency: ISO code (EUR = 978) |
| EXPONENT | N | 1 | N | Exponent of the chosen Currency<br>(Recommended if the currency is different from Euro) |
| OPDESCR | N | 100 | AN | Optional description of the Operation at merchant's discretion |
| RELEASE | Y | 2 | AN | Release of APIs: to be filled in with "02" |
| OPTIONS | N | Min 0<br>Max 26 | AN | The field OPTIONS permits to activate various additional options for the ongoing message. |
| MAC | Y | 32, 40, 64 | AN | Message Authentication Code. For the related calculation see appendix 4.2.1 |

### Request of refund in XML format

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<BPWXmlRequest>
   <Release>02</Release>
   <Request>
```

```
      <Operation>REFUND</Operation>
      <Timestamp>2015-03-04T11:20:00.000</Timestamp>
      <MAC>115025d5a5b65df687790867bdece136</MAC>
   </Request>
   <Data>
      <Refund>
         <Header>
            <ShopID>000000000000003</ShopID>
            <OperatorID>oper0001</OperatorID>
            <ReqRefNum>12345678901234567890123456789000</ReqRefNum>
         </Header>
         <TransactionID>1234567890</TransactionID>
         <OrderID>9998500000000015</OrderID>
         <Amount>7700</Amount>
         <Currency>978</Currency>
         <Exponent>2</Exponent>
         <OpDescr>CallCenterRequest1038</OpDescr>
      </Refund>
   </Data>
</BPWXmlRequest>
```

The response message to the payment reversal request is formatted in XML. The data section is described on the following scheme.



***Scheme 16 - Data section for REFUND response***

As can be noted the response to a payment reversal request consists of two elements: the outcome of the ceiling restoration transaction and any accounting transaction carried out in order to repay the amount to the card holder.
If the TRANSACTIONID of the original transaction does not exist, or if an authentication error occurs, the response elements contained in Data will not be created.

Here below is an example of a file generated by the response to a previously booked authorization reversal request:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<BPWXmlResponse>
   <Timestamp>2015-07-04T12:02:55</Timestamp>
   <Result>00</Result>
   <!-- This MAC signs timestamp and result -->
   <MAC>ffb3553a4ab34c3a4ab34c3a4ab34c3a</MAC>
   <Data>
      <!-- This element contains request data -->
      <Refund>
         <Header>
            <ShopID>23486788</ShopID>
            <OperatorID>A4348B</OperatorID>
            <ReqRefNum>20150501496204690934584305834564</ReqRefNum>
         </Header>
         <TransactionID>C355645658457564564565636</TransactionID>
         <OrderID>A398459</OrderID>
         <Amount>10000</Amount>
         <Currency>978</Currency>
         <Exponent>2</Exponent>
      </Refund>
      <PlafondRestoringResult>00</PlafondRestoringResult>
      <Operation>
         <TransactionID>C5555358793</TransactionID>
         <TimestampReq>2015-07-04T22:02:55</TimestampReq>
         <TimestampElab>NULL</TimestampElab>
         <SrcType>01</SrcType>
         <Amount>10000</Amount>
         <Result>00</Result>
         <Status>00</Status>
         <OpDescr>CallCenterRequest1038</OpDescr>
         <!-- This MAC signs operation data  -->
         <MAC>12334c3a4ab34c3a4ab34c3a4ab3ffa1</MAC>
         <Authorization>
            <PaymentType>03</PaymentType>
            <AuthorizationType>I</AuthorizationType>
            <TransactionID> C395645658457564564565636</TransactionID>
            <Network>01</Network>
            <OrderID>A398459</OrderID>
            <TransactionAmount>10000</TransactionAmount>
            <AuthorizedAmount>10000</AuthorizedAmount>
            <Currency>978</Currency>
            <Exponent>2</Exponent>
            <AccountedAmount>0</AccountedAmount>
            <RefundedAmount>0</RefundedAmount>
            <TransactionResult>00</TransactionResult>
            <Timestamp>2015-07-09T21:05:44</Timestamp>
            <AuthorizationNumber>A93485</AuthorizationNumber>
            <AcquirerBIN>123450943</AcquirerBIN>
            <MerchantID>09834509</MerchantID>
            <TransactionStatus>01</TransactionStatus>
            <ResponseCodeISO>00</ResponseCodeISO>
            <!-- This MAC signs the authorization  -->
            <MAC>4ab34c3a4ab34c3a4ab34c3a4ab34c3a</MAC>
         </Authorization>
      </Operation>
   </Data>
</BPWXmlResponse>
```

The meaning of the elements is the following:

**<BPWXmlResponse>**

This is the root element of the document, there is only one element of this type in the message, which consists of the following elements:

| | |
|---|---|
| • **<Timestamp>** | date and time of response message |
| • **<Result>** | outcome of transaction requested |

| Code | Description |
|---|---|

| | |
|---|---|
| 00 | Success |
| 02 | ReqRefNum duplicated or incorrect |
| 03 | Incorrect message format, missing or incorrect field |
| 04 | Incorrect API authentication, incorrect MAC |
| 06 | Unforeseen error during processing of request |
| 07 | TransactionID not found |
| 40 | Empty xml or missing 'data' parameter |
| 41 | Xml not parsable |
| 98 | Application error |

In case of multiple reversals (that is, reversal requests on already partially reversed authorizations) special emphasis should be placed on the following circumstances:

- reversal with unspecified amount: the result will be 03 (that is, full reversals after partial reversals are not accepted).
- Reversal on an already fully reversed authorization: the result will be 00.

| | |
|---|---|
| • **<MAC>** | message authentication code: signature of timestamp and result. See appendix 4.2.7 |
| • **<Data>** | data relating to the reversal request and response message |

**<Data>**

There is only one element of this type in the message, containing all the data of the reversal request and of the response message and consisting of the following elements:

| | |
|---|---|
| • **<Refund>** | data relating to the authorization reversal request |
| • **<PlafondRestoringResult>** | outcome of the ceiling restoration |
| • **<Operation>** | data relating to the accounting transaction |

**<Refund>**

There is only one element of this type in the message, containing all the data of the reversal request and of the response message and consisting of the following elements:

| | |
|---|---|
| • **<Header>** | data relating to the request sent |
| • **<TransactionID>** | identifier of the reversal request transaction |
| • **<OrderID>** | order code |
| • **<Amount>** | amount of the requested authorization in Eur cents |
| • **<Currency>** | currency ISO code: 978=EUR |

**<Header>**

There is only one element of this type in the message, which includes the date relating to the request sent. For the related description see paragraph 3.3.2 in the chapter "Response messages in XML".

**<Operation>**

This element is optional: it is present only if, to perform the reversal, an accounting transaction is required. If it is present, it will contain all the data relating to the accounting transaction performed. For a detailed description see the chapter "Response messages in XML"

# 3.9 Consultation operations

## 3.9.1 List of operations on transactions

This message permits to obtain a list of all the operations on transactions. It refers to the booking and credit requests that have been forwarded to the system.
Here below is a list of both the requests that have already been sent to the acquirers and of those that have not yet been sent. The latter are distinguished by the fact that the processing date is not filled in.

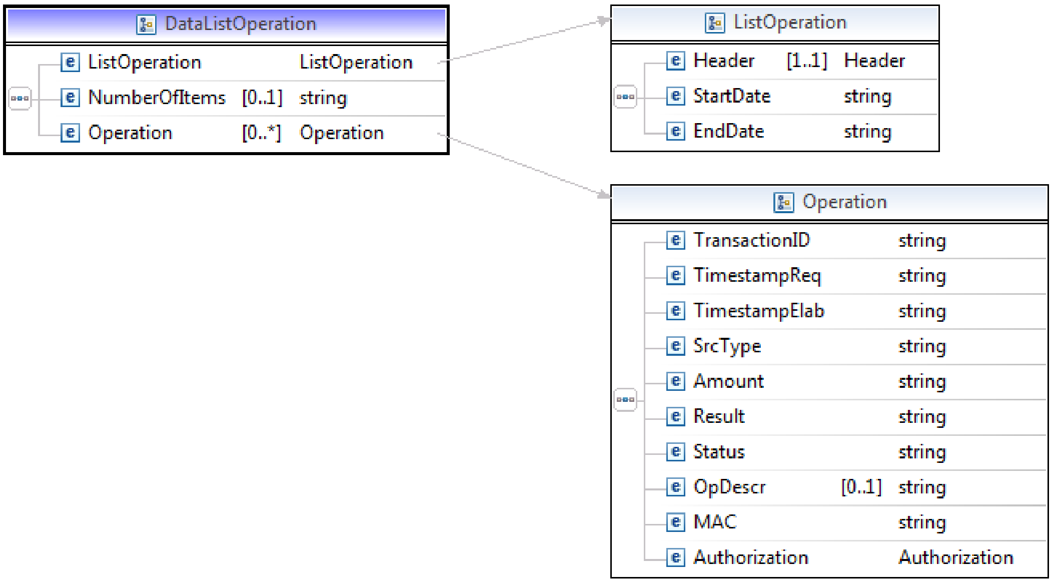
The fields to be specified in the HTTP request message are the following:

| Field | Compulsory | Size | Type | Description |
|---|---|---|---|---|
| | | | | |
| OPERATION | Y | | A | Transaction requested: filled in with "LISTOPERATION" |
| TIMESTAMP | Y | 23 | AN | Local timestamp of the yyyy-MM-ddTHH:mm:ss.SSS type |
| SHOPID | Y | 15 | AN | Identifier of the merchant's store assigned by SIA, Merchant ID (MID) |
| OPERATORID | Y | Min. 8<br>Max.18 | AN | It indicates the person who has requested the transaction. |
| REQREFNUM | Y | 32 | N | Unique identifier of the request managed by the merchant. It can be used to retrieve information on the request made also in the case of no response. The first 8 digits must be in the yyyyMMdd format including the request date. |
| STARTDATE | Y | 10 | D | Date of start of period, in yyyy-MM-dd format |
| ENDDATE | Y | 10 | D | Date of end of period, in yyyy-MM-dd format |
| RELEASE | Y | 2 | AN | Release of APIs: to be filled in with "02" |
| OPTIONS | N | Min 0<br>Max 26 | AN | The field OPTIONS permits to activate various additional options for the ongoing message. |
| SRCTYPE | N | 2 | AN | Type of transaction to be extracted. The possible values are: 01 02 03 04, which make reference to the field <SrcType> of the element <Operation>. See the chapter "Response messages in XML" |
| OPDESCR | N | 100 | AN | The resulting list includes only the operations with the optional description (see refund message) |
| MAC | Y | 32, 40, 64 | AN | Message Authentication Code. For the related calculation see appendix 4.2.4 |

### Request of list of operations on transactions in XML format

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<BPWXmlRequest>
   <Release>02</Release>
   <Request>
      <Operation>LISTOPERATION</Operation>
      <Timestamp>2015-03-04T11:20:00.000</Timestamp>
      <MAC>115025d5a5b65df687790867bdece136</MAC>
   </Request>
   <Data>
      <ListOperation>
         <Header>
            <ShopID>000000000000003</ShopID>
            <OperatorID> oper0001</OperatorID>
            <ReqRefNum>12345678901234567890123456789000</ReqRefNum>
         </Header>
         <StartDate>2014-12-01</StartDate>
         <EndDate>2014-12-31</EndDate>
         <SrcType>03</SrcType>
         <OpDescr>CallCenterRequest1038</OpDescr>
      </ListOperation>
   </Data>
</BPWXmlRequest>
```

The response message to the request of list of accounting transactions is formatted in XML. The data section is described on the following scheme.



*Scheme 17 - Data section for LISTOPERATION response*

The response to a request of accounting list consists of a series of elements of the Operation type.
If an error occurs, the element OperationList will not be created.

Here below is an example of a file generated by the response to a request of list of accounting transactions:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<BPWXmlResponse>
   <Timestamp>2015-07-04T12:02:55</Timestamp>
   <Result>00</Result>
   <!-- This MAC signs timestamp and result -->
   <MAC>ffb3553a4ab34c3a4ab34c3a4ab34c3a</MAC>
   <Data>
      <!-- This element contains request data -->
      <ListOperation>
         <Header>
            <ShopID>23486788</ShopID>
            <OperatorID>A4348B</OperatorID>
            <ReqRefNum>20150501496204690934584305834564</ReqRefNum>
         </Header>
         <StartDate>2015-01-01</StartDate>
         <EndDate>2015-07-10</EndDate>
      </ListOperation>
            <NumberOfItems>2</NumberOfItems>
      <Operation>
         <TransactionID>C9435879295</TransactionID>
         <TimestampReq>2015-07-04T12:02:55</TimestampReq>
         <TimestampElab>2015-07-04T23:02:55</TimestampElab>
         <SrcType>21</SrcType>
         <Amount>10000</Amount>
         <Result>00</Result>
         <Status>03</Status>
         <OpDescr>CallCenterRequest1038</OpDescr>
         <!-- This MAC signs operation data  -->
         <MAC>12dd4c3a4ab34c3a4abc4c3a4ab3ffa1</MAC>
         <Authorization>
            <PaymentType>03</PaymentType>
            <AuthorizationType>I</AuthorizationType>
            <TransactionID> C3956456584575645645656 36</TransactionID>
            <Network>01</Network>
            <OrderID>A398459</OrderID>
            <TransactionAmount>10000</TransactionAmount>
            <AuthorizedAmount>10000</AuthorizedAmount>
            <Currency>978</Currency>
            <Exponent>2</Exponent>
            <AccountedAmount>8000</AccountedAmount>
            <RefundedAmount>100</RefundedAmount>
            <TransactionResult>00</TransactionResult>
            <Timestamp>2015-07-09T21:05:44</Timestamp>
            <AuthorizationNumber>A93485</AuthorizationNumber>
            <AcquirerBIN>123450943</AcquirerBIN>
            <MerchantID>09834509</MerchantID>
            <TransactionStatus>01</TransactionStatus>
            <ResponseCodeISO>00</ResponseCodeISO>
            <!-- This MAC signs the authorization  -->
            <MAC>4ab34c3a4ab34c3a4ab34c3a4ab34c3a</MAC>
         </Authorization>
      </Operation>
      <Operation>
         <TransactionID>C9435879384</TransactionID>
         <TimestampReq>2015-17-04T12:02:55</TimestampReq>
         <TimestampElab>2015-17-04T23:02:55</TimestampElab>
         <SrcType>20</SrcType>
         <Amount>2000</Amount>
         <Result>00</Result>
         <Status>00</Status>
         <OpDescr>CallCenterRequest1038</OpDescr>
         <!-- This MAC signs operation data  -->
         <MAC>aa334c3a4ab34c3a4ab34c3a4ab3ffa1</MAC>
         <Authorization>
            <PaymentType>03</PaymentType>
            <AuthorizationType>I</AuthorizationType>
            <TransactionID> C3956456584575645645656 36</TransactionID>
            <Network>01</Network>
            <OrderID>A398459</OrderID>
            <TransactionAmount>10000</TransactionAmount>
            <AuthorizedAmount>10000</AuthorizedAmount>
            <Currency>978</Currency>
            <Exponent>2</Exponent>
            <AccountedAmount>8000</AccountedAmount>
            <RefundedAmount>100</RefundedAmount>
            <TransactionResult>00</TransactionResult>
            <Timestamp>2015-07-09T21:05:44</Timestamp>
            <AuthorizationNumber>A93485</AuthorizationNumber>
            <AcquirerBIN>123450943</AcquirerBIN>
            <MerchantID>09834509</MerchantID>
            <TransactionStatus>01</TransactionStatus>
            <ResponseCodeISO>00</ResponseCodeISO>
            <!-- This MAC signs the authorization  -->
            <MAC>4ab34c3a4ab34c3a4ab34c3a4ab34c3a</MAC>
         </Authorization>
      </Operation>
   </Data>
</BPWXmlResponse>
```

The meaning of the elements is the following:

**<BPWXmlResponse>**

This is the root element of the document, there is only one element of this type in the message, which is composed of the following elements:

- **<Timestamp>** date and time of response message
- **<Result>** outcome of transaction requested "00" list performed

| Code | Description |
|---|---|
| 00 | Success |
| 02 | ReqRefNum duplicated or incorrect |
| 03 | Incorrect message format, missing or incorrect field |
| 04 | Incorrect API authentication, incorrect MAC |
| 05 | Incorrect date, or period indicated empty |
| 06 | Unforeseen error during processing of request |
| 07 | TransactionID not found |
| 40 | Empty xml or missing 'data' parameter |
| 41 | Xml not parsable |

- **<MAC>** message authentication code: signature of timestamp and result. See appendix 4.2.7
- **<Data>** data relating to the request of list of accounting transactions and to the response message

**<Data>**

There is only one element of this type in the message, containing all the data relating to the request of list of accounting transactions and of the response message and consisting of the following elements:

- **<ListOperation>** data relating to the request of list of accounting transactions
- **<Operation>** data relating to the accounting transactions

**<ListOperation>**

There is only one element of this type in the message, containing all the data relating to the request of list of accounting transactions and consisting of the following elements:

- **<Header>** data relating to the request sent
- **<StartDate>** date of start of period of the list
- **<EndDate>** date of end of period of the list

**<Header>**

There is only one element of this type in the message, which includes the date relating to the request sent. For the related description see paragraph 3.3.2 in the chapter "Response messages in XML".

**<NumberOfItems>**

This element contains the number of elements constituting the requested list

**<Operation>**

There are as many occurrences of this element as the accounting transactions making up the list produced. For a detailed description see the chapter “Response messages in XML”

## 3.9.2 List of authorizations

This transaction permits to obtain a list of the authorization requests forwarded by the SIA VPOS system to the international or national payment circuits within a given time period.
It is possible to indicate whether all authorizations, only those for which authorization was granted, only those that have been denied, or only those that have been reversed are to be obtained.

The fields to be specified in the HTTP request message are the following:

| Field | Compulsory | Size | Type | Description |
|---|---|---|---|---|
| | | | | |
| OPERATION | Y | | A | Transaction requested: filled in with "LISTAUTHORIZATION" |
| TIMESTAMP | Y | 23 | AN | Local timestamp of the yyyy-MM-ddTHH:mm:ss.SSS type |
| SHOPID | Y | 15 | AN | Identifier of the merchant's store assigned by SIA, Merchant ID (MID) |
| OPERATORID | Y | Min. 8<br>Max.18 | AN | It indicates the person who has requested the transaction. |
| REQREFNUM | Y | 32 | N | Unique identifier of the request managed by the merchant. It can be used to retrieve information on the request made also in the case of no response. The first 8 digits must be in the yyyyMMdd format including the request date. |
| STARTDATE | N | 10 | D | Date of start of period, yyyy-MM-dd format |
| ENDDATE | N | 10 | D | Date of end of period, yyyy-MM-dd format |
| FILTER | Y | 1 | N | Type of list requested:<br>1. Only those with POSITIVE outcome<br>2. Only those with negative outcome<br>3. Only those reversed<br>4. All of them |
| TRANSACTION ID | N | 25 | AN | Unique identifier of the transaction. If present, the system will ignore any filter, date and time fields in order to recover the transaction indicated. |
| STARTTIME | N | 5 | D | Time of start of period, HH.mm format |
| ENDTIME | N | 5 | D | Time of end of period, HH.mm format |
| RELEASE | Y | 2 | AN | Release of APIs: to be filled in with "02" |
| OPTIONS | N | Min 0<br>Max 26 | AN | The field OPTIONS permits to activate various additional options for the ongoing message. |
| MAC | Y | 32, 40,<br>64 | AN | Message Authentication Code. For the related calculation see appendix 4.2.5 |

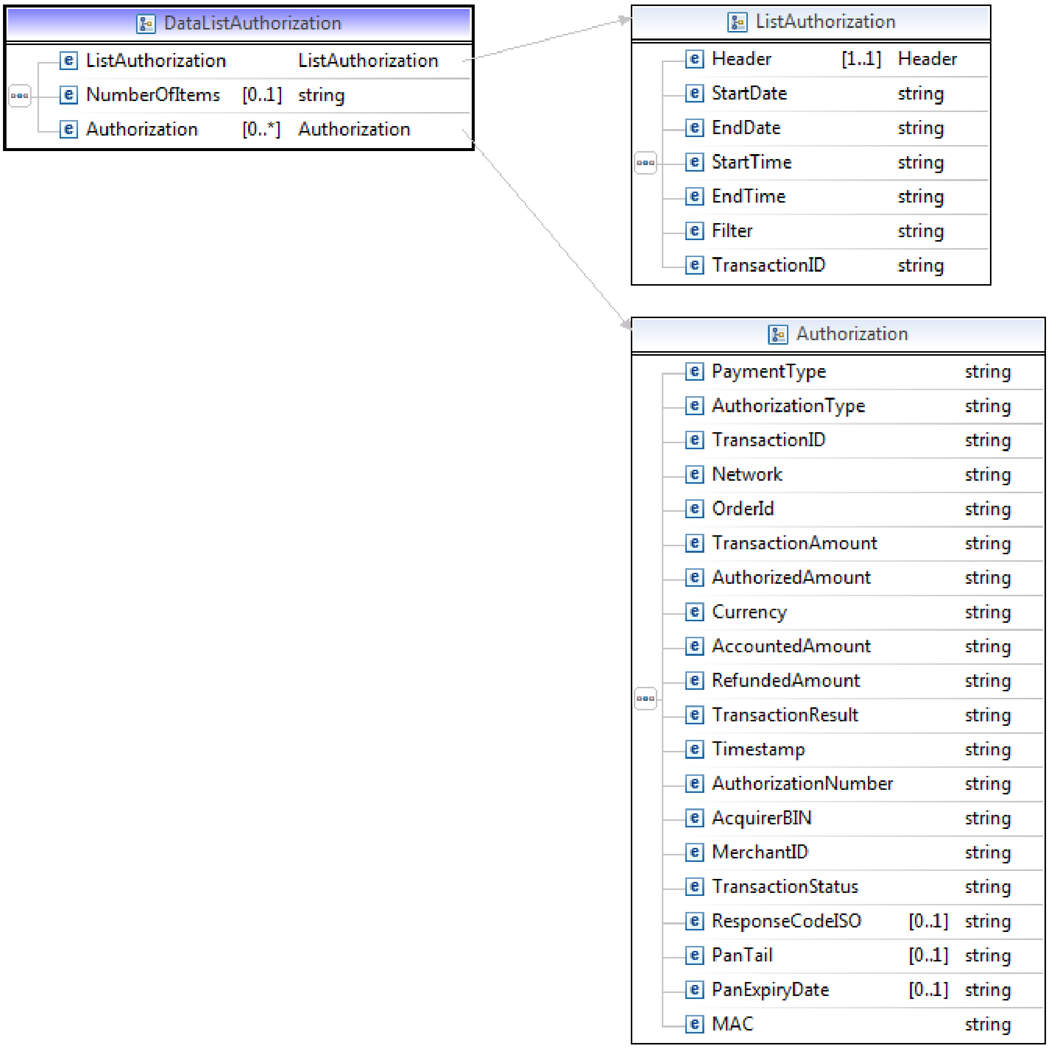
The search can be performed with one of the following alternatives:

1) Field TRANSACTIONID specified: the search will be performed taking only said field as a discriminating parameter (ignoring any filter, date and time fields)
2) Field TRANSACTIONID not specified: the search will be performed taking into account the parameters FILTER, STARTDATE, ENDDATE, STARTTIME, ENDTIME. The fields FILTER, STARTDATE, ENDDATE are in this case compulsory, whilst STARTTIME and ENDTIME can be omitted.

## Request of list of authorizations in XML format

```
<?xml version="1.0" encoding="ISO-8859-1"?>
   <BPWXmlRequest>
      <Release>02</Release>
      <Request>
         <Operation>LISTAUTHORIZATION</Operation>
         <Timestamp>2015-03-04T11:20:00.000</Timestamp>
         <MAC>115025d5a5b65df687790867bdece136</MAC>
      </Request>
      <Data>
         <ListAuthorization>
            <Header>
               <ShopID>000000000000003</ShopID>
               <OperatorID>oper0001</OperatorID>
               <ReqRefNum>12345678901234567890123452289000</ReqRefNum>
            </Header>
            <StartDate>2014-12-01</StartDate>
            <EndDate>2014-12-31</EndDate>
            <Filter>1</Filter>
            <TransactionID/>
            <StartTime>00.00</StartTime>
            <EndTime>18.25</EndTime>
         </ListAuthorization>
      </Data>
   </BPWXmlRequest>
```

The response message to the request of list of authorizations is formatted in XML. The data section is described on the following scheme.



***Scheme 18 - Data section for LISTAUTHORIZATION response***

The response to a request of list of authorizations consists of a series of elements of the Authorization type. If an error occurs the element Authorization will not be created.

Here below is an example of a file generated by the response to a request of list of authorizations:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<BPWXmlResponse>
   <Timestamp>2015-07-04T12:02:55</Timestamp>
   <Result>00</Result>
   <!-- This MAC signs timestamp and result -->
   <MAC>ffb3553a4ab34c3a4ab34c3a4ab34c3a</MAC>
   <Data>
      <!-- This element contains request data -->
      <ListAuthorization>
         <Header>
            <ShopID>23486788</ShopID>
            <OperatorID>A4348B</OperatorID>
            <ReqRefNum>20150501496204690934584305834564</ReqRefNum>
         </Header>
         <StartDate>2015-01-01</StartDate>
         <EndDate>2015-07-10</EndDate>
         <StartTime>10.00</StartTime>
         <EndTime>18.30</EndTime>
         <Filter>1</Filter>
         <TransactionID> C3956456584575645645565636</TransactionID>
      </ListAuthorization>
      <NumberOfItems>2</NumberOfItems>
      <Authorization>
         <PaymentType>03</PaymentType>
         <AuthorizationType>I</AuthorizationType>
         <TransactionID> C3956456584575645645565636</TransactionID>
         <Network>01</Network>
         <OrderID>A398459</OrderID>
         <TransactionAmount>10000</TransactionAmount>
         <AuthorizedAmount>10000</AuthorizedAmount>
         <Currency>978</Currency>
         <Exponent>2</Exponent>
         <AccountedAmount>10000</AccountedAmount>
         <RefundedAmount>100</RefundedAmount>
         <TransactionResult>00</TransactionResult>
         <Timestamp>2015-07-09T21:05:44</Timestamp>
         <AuthorizationNumber>A93485</AuthorizationNumber>
         <AcquirerBIN>123450943</AcquirerBIN>
         <MerchantID>09834509</MerchantID>
         <TransactionStatus>01</TransactionStatus>
         <ResponseCodeISO>00</ResponseCodeISO>
         <!-- This MAC signs the authorization  -->
         <MAC>4ab34c3a4ab34c3a4ab34c3a4ab34c3a</MAC>
      </Authorization>
      <Authorization>
         <AuthorizationType>D</AuthorizationType>
         <TransactionID> C3956456584575645645565636</TransactionID>
         <Network>01</Network>
         <OrderID>A398459</OrderID>
         <TransactionAmount>10000</TransactionAmount>
         <AuthorizedAmount>5000</AuthorizedAmount>
         <Currency>978</Currency>
         <Exponent>2</Exponent>
         <AccountedAmount>5000</AccountedAmount>
         <RefundedAmount>100</RefundedAmount>
         <TransactionResult>00</TransactionResult>
         <Timestamp>2015-07-09T21:05:44</Timestamp>
         <AuthorizationNumber>A93485</AuthorizationNumber>
         <AcquirerBIN>123450943</AcquirerBIN>
         <MerchantID>09834509</MerchantID>
         <TransactionStatus>03</TransactionStatus>
         <!-- This MAC signs the authorization  -->
         <MAC>aab3553a4ab34c3a4ab34c3a4ab34c3a</MAC>
      </Authorization>
   </Data>
</BPWXmlResponse>
```

The meaning of the elements is the following:

**<BPWXmlResponse>**

This is the root element of the document, there is only one element of this type in the message, which is composed of the following elements:

- **<Timestamp>** date and time of the response message
- **<Result>** outcome of transaction requested

| Code | Description |
|---|---|
| 00 | Success |

| | |
|---|---|
| 02 | ReqRefNum duplicated or incorrect |
| 03 | Incorrect message format, missing or incorrect field |
| 04 | Incorrect API authentication, incorrect MAC |
| 05 | Incorrect date, or period indicated empty |
| 06 | Unforeseen error during processing of request |
| 07 | TransactionID not found |
| 40 | Empty xml or missing 'data' parameter |
| 41 | Xml not parsable |

- **<MAC>** message authentication code: signature of timestamp and result. See appendix 4.2.7
- **<Data>** data relating to the request of list of authorizations and to the response message

**<Data>**

There is only one element of this type in the message, containing all the data relating to the request of list of authorizations and to the response message and consisting of the following elements:

- **<ListAuthorization>** data relating to the request of list of authorizations
- **<NumberOfItems>** data relating to the list of authorizations

**<ListAuthorization>**

There is only one element of this type in the message, containing all the data relating to the request of list of authorizations and consisting of the following elements:

- **<Header>** data relating to the request sent
- **<StartDate>** date of start of period of the list
- **<EndDate>** date of end of period of the list
- **<StartTime>** time of start of period of the list, if indicated in the request
- **<EndTime>** time of end of period of the list, if indicated in the request
- **<Filter>** type of list requested:
  1 – Authorizations with positive outcome ( Statuses : 00 – 02 – 03 – 10 )
  2 – Denied authorizations ( Statuses : 01 – 21 )
  3 – Reversed authorizations ( Statuses : 04 – 05 – 20 )
  4 – All authorizations
- **<TransactionID>** ID of the transaction to be searched

**<Header>**

There is only one element of this type in the message, which includes the date relating to the request sent. For the related description see paragraph 3.3.2 in the chapter "Response messages in XML".

**<NumberOfItems>**

This element, if present, contains the number of authorizations set out in the attribute NumberOfItems.

**<Authorization>**

There are N occurrences of this element. Each one of them contains all the data relating to an authorization of the list. For a detailed description see chapter "Response messages in XML"

If the store is pan tail return service (SV64) enabled, the authorization will contain also the <PanTail> and <PanExpiryDate> elements.

## 3.9.3 Request of order status

This transaction returns the current status of an order, including all the related authorization transactions. The main purpose of this message is to enable the merchant systems to verify the status of an order that is still "pending" during payment.

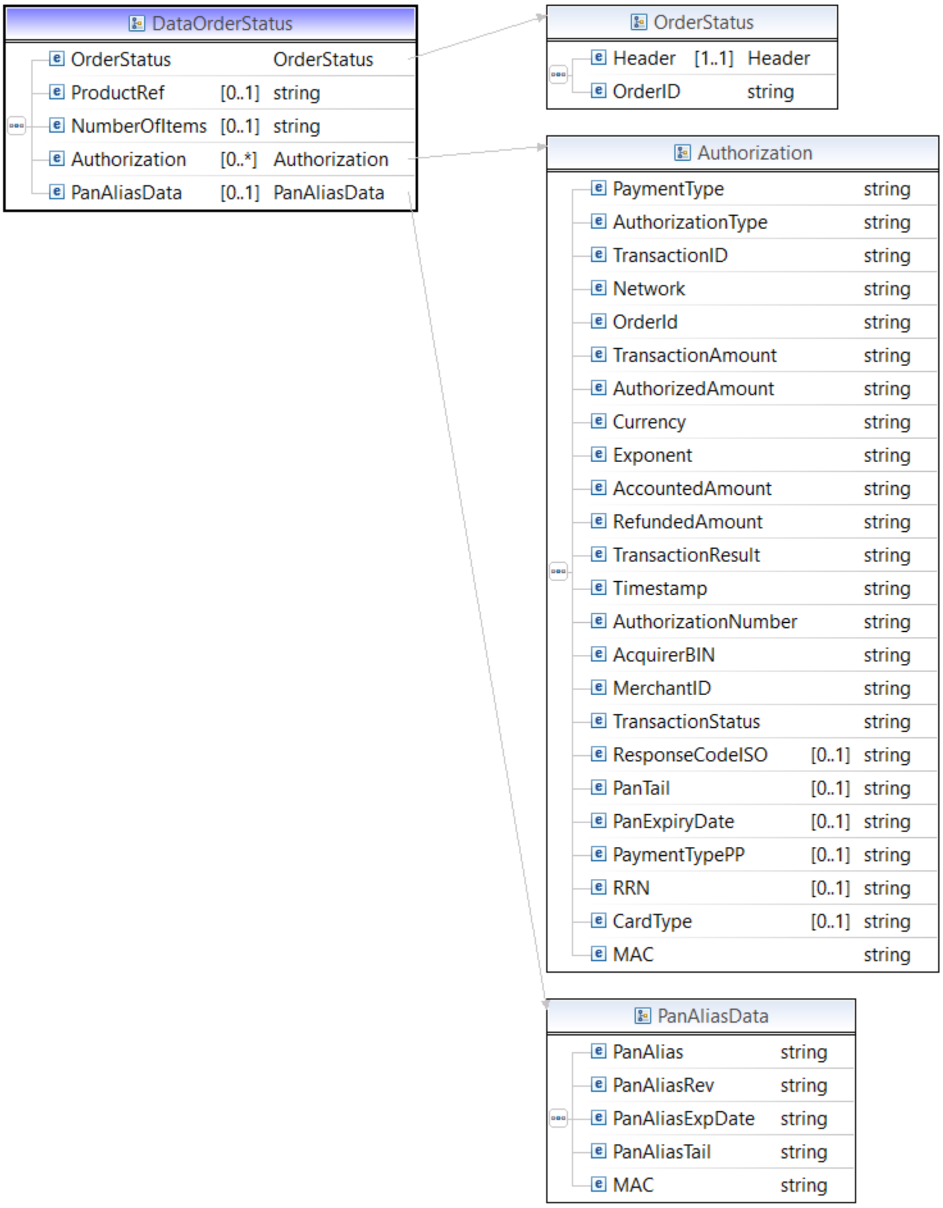
The fields to be specified in the HTTP request message are the following:

| Field | Compulsory | Size | Type | Description |
|---|---|---|---|---|
| | | | | |
| OPERATION | Y | | A | Transaction requested: filled in with "ORDERSTATUS" |
| TIMESTAMP | Y | 23 | AN | Local timestamp of the yyyy-MM-ddTHH:mm:ss.SSS type |
| SHOPID | Y | 15 | AN | Identifier of the merchant's store assigned by SIA, Merchant ID (MID) |
| OPERATORID | Y | Min. 8<br>Max.18 | AN | It indicates the person who has requested the transaction. |
| REQREFNUM | Y | 32 | N | Unique identifier of the request managed by the merchant. It can be used to retrieve information on the request made also in the case of no response. The first 8 digits must be in the yyyyMMdd format including the request date. |
| ORDERID | Y | Min.1<br>Max.50 | AN | Unique order identifier. |
| PRODUCTREF | N | 15 | AN | Sale identifier |
| RELEASE | Y | 2 | AN | Release of APIs: to be filled in with "02" |
| OPTIONS | N | Min 0<br>Max 26 | AN | The field OPTIONS permits to activate various additional options for the ongoing message. |
| MAC | Y | 32, 40,<br>64 | AN | Message Authentication Code. For the related calculation see appendix 4.2.6 |

### Request of order status in XML format

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<BPWXmlRequest>
   <Release>02</Release>
   <Request>
      <Operation>ORDERSTATUS</Operation>
      <Timestamp>2015-03-04T11:20:00.000</Timestamp>
      <MAC>115025d5a5b65df687790867bdece136</MAC>
   </Request>
   <Data>
      <OrderStatus>
         <Header>
            <ShopID>000000000000003</ShopID>
            <OperatorID>oper0001</OperatorID>
            <ReqRefNum>12345678901234567890123452289000</ReqRefNum>
         </Header>
         <OrderID>9998500000000015</OrderID>
      </OrderStatus>
   </Data>
</BPWXmlRequest>
```

The response message to an order status request is formatted in XML. The data section is described on the following scheme.



***Scheme 19 - Data section for ORDERSTATUS response***

The response to an order status request consists of a series of elements of the Authorization type: these consist of the various authorizations connected to the indicated order number. If the order has been processed with immediate authorization, only one authorization will appear.
If an error occurs, no Authorization element will appear.

Here below is an example of a file generated by the response to an order status request:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<BPWXmlResponse>
   <Timestamp>2015-07-04T12:02:55</Timestamp>
   <Result>00</Result>
   <MAC>ffb3553a4ab34c3a4ab34c3a4ab34c3a</MAC>
   <Data>
```

```
    <OrderStatus>
      <Header>
        <ShopID>23486788</ShopID>
        <OperatorID>A4348B</OperatorID>
        <ReqRefNum>20150501496204690934584305834564</ReqRefNum>
      </Header>
      <OrderID>12348A33</OrderID>
    </OrderStatus>
    <ProductRef>XYZABC</ProductRef>
    <NumberOfItems>2</NumberOfItems>
    <Authorization>
    <PaymentType>03</PaymentType>
      <AuthorizationType>I</AuthorizationType>
      <TransactionID> C355645658457564564565636</TransactionID>
      <Network>01</Network>
      <OrderID>A398459</OrderID>
      <TransactionAmount>10000</TransactionAmount>
      <AuthorizedAmount>10000</AuthorizedAmount>
      <Currency>978</Currency>
      <Exponent>2</Exponent>
      <AccountedAmount>10000</AccountedAmount>
      <RefundedAmount>100</RefundedAmount>
      <TransactionResult>00</TransactionResult>
      <Timestamp>2015-07-09T21:05:44</Timestamp>
      <AuthorizationNumber>A93485</AuthorizationNumber>
      <AcquirerBIN>123450943</AcquirerBIN>
      <MerchantID>09834509</MerchantID>
      <TransactionStatus>01</TransactionStatus>
      <ResponseCodeISO>00</ResponseCodeISO>
      <MAC>12334c3a4ab34c3a4ab34c3a4ab3ffa1</MAC>
    </Authorization>
    <Authorization>
      <PaymentType>03</PaymentType>
      <AuthorizationType>D</AuthorizationType>
      <TransactionID>C355645658457564564565636</TransactionID>
      <Network>01</Network>
      <OrderID>A398459</OrderID>
      <TransactionAmount>10000</TransactionAmount>
      <AuthorizedAmount>5000</AuthorizedAmount>
      <Currency>978</Currency>
      <Exponent>2</Exponent>
      <AccountedAmount>5000</AccountedAmount>
      <RefundedAmount>100</RefundedAmount>
      <TransactionResult>00</TransactionResult>
      <Timestamp>2015-07-02T21:05:44</Timestamp>
      <AuthorizationNumber>A93485</AuthorizationNumber>
      <AcquirerBIN>123450943</AcquirerBIN>
      <MerchantID>09834509</MerchantID>
      <TransactionStatus>03</TransactionStatus>
      <ResponseCodeISO>00</ResponseCodeISO>
      <MAC>bbb34c3a4ab34c3a4ab34c3a4ab3ffa1</MAC>
    </Authorization>
    <PanAliasData>
      <PanAlias>0000197412081271677</PanAlias>
      <PanAliasRev></PanAliasRev>
      <PanAliasExpDate>2911</PanAliasExpDate>
      <PanAliasTail>0003</PanAliasTail>
      <MAC>E61612E0C0F71A2FE838BC0736B396E6</MAC>
    </PanAliasData>
  </Data>
</BPWXmlResponse>
```

The meaning of the elements is the following:

**<BPWXmlResponse>**

- This is the root element of the document, there is only one element of this type in the message, which is composed of the following elements:

- **`<Timestamp>`** date and time of response message
- **`<Result>`** outcome of transaction requested

| Code | Description |
|---|---|
| 00 | Success |
| 01 | Order or ReqRefNum not found |
| 02 | ReqRefNum duplicated or incorrect |
| 03 | Incorrect message format, missing or incorrect field |
| 04 | Incorrect API authentication, incorrect MAC |
| 06 | Unforeseen error during processing of request |
| 07 | TransactionID not found |
| 40 | Empty xml or missing 'data' parameter |
| 41 | Xml not parsable |
| 99 | Transaction failed, see specific outcome enclosed in the element <Data> of the response. |

- **`<MAC>`** message authentication code: signature of timestamp and result. See appendix 4.2.7
- **`<Data>`** data relating to the order status request and to the response message

**`<Data>`**

There is only one element of this type in the message, containing all the data relating to the order status request and response message and consisting of the following elements:

- **`<OrderStatus>`** data relating to the order status request
- **`<NumberOfItems>`** number of authorizations found
- **`<Authorization>`** authorization associated to the order

**`<OrderStatus`**

There is only one element of this type in the message, containing all the data relating to the order status request and consisting of the following elements:

- **`<Header>`** data relating to the request sent
- **`<OrderID>`** number of the order for which the status was requested

**`<Header>`**

There is only one element of this type in the message, which includes the date relating to the request sent. For the related description see paragraph 3.3.2 in the chapter "Response messages in XML".

**`<Authorization>`**

There may be more than one element of this type. Each element consists of an authorization connected to the selected order.
For a detailed description see the chapter "Response messages in XML"

**<PanAliasData>**

There is only one element of this type in the message, which includes the card alias data. For the related description see paragraph 3.3.5 in the chapter  "Response messages in XML"

# 3.10 Operations on Pan Alias

## 3.10.1 Pan alias recovery request

This transaction returns the pan alias generated by the store's authorization of a given order.

The fields to be specified in the HTTP request message are the following:

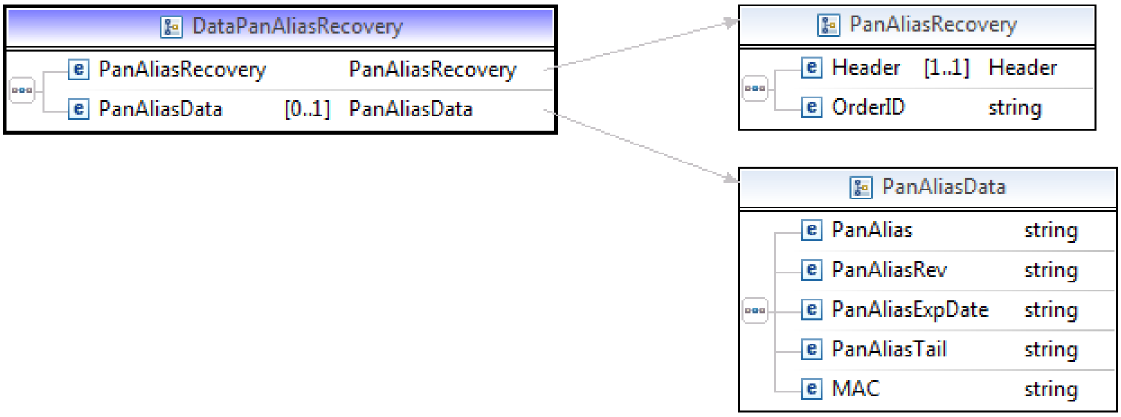
| Field | Compulsory | Size | Type | Description |
|---|---|---|---|---|
| | | | | |
| OPERATION | Y | | A | Transaction requested: filled in with "PANALIASRECOVERY" |
| TIMESTAMP | Y | 23 | AN | Local timestamp of the yyyy-MM-ddTHH:mm:ss.SSS type |
| SHOPID | Y | 15 | AN | Identifier of the merchant's store assigned by SIA [MID] |
| OPERATORID | Y | Min. 8<br>Max.18 | AN | It indicates the person who has requested the transaction. |
| REQREFNUM | Y | 32 | N | Unique identifier of the request managed by the merchant. The first 8 digits must be in the yyyyMMdd format including the request date. |
| ORDERID | Y | Max.50 | AN | Unique identifier of the order entered for the search. |
| RELEASE | Y | 2 | AN | Release of APIs: to be filled in with "02" |
| OPTIONS | N | Min 0<br>Max 26 | AN | The field OPTIONS permits to activate various additional options for the ongoing message. |
| MAC | Y | 32, 40, 64 | AN | Message Authentication Code. For the related calculation see appendix 4.2.18 |

### Request of pan alias recovery in XML format

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<BPWXmlRequest>
   <Release>02</Release>
   <Request>
      <Operation>PANALIASRECOVERY</Operation>
      <Timestamp>2015-03-04T11:20:00.000</Timestamp>
      <MAC>115025d5a5b65df687790867bdece136</MAC>
   </Request>
   <Data>
      <PanAliasRecovery>
         <Header>
            <ShopID>000000000000003</ShopID>
            <OperatorID>oper0001</OperatorID>
            <ReqRefNum>12345678901234567890123452289000</ReqRefNum>
         </Header>
         <OrderID>9998500000000015</OrderID>
      </PanAliasRecovery>
   </Data>
</BPWXmlRequest>
```

The response, which is in XML format, will include the element **<PanAliasData>** (present if and only if the authorization request and the creation of the alias pan have a positive outcome).

The data section of the response is described on the following scheme.



***Scheme 20 - Data section for PANALIASRECOVERY response***

Example of API XML response in which the pan alias code is returned:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<BPWXmlResponse>
   <Timestamp>2015-07-04T12:02:55</Timestamp>
   <Result>00</Result>
   <MAC>ffb3553a4ab34c3a4ab34c3a4ab34c3a</MAC>
   <Data>
      <PanAliasRecovery>
         <Header>
            <ShopID>23486788</ShopID>
            <OperatorID>A4348B</OperatorID>
            <ReqRefNum>20150501496204690934584305834564</ReqRefNum>
         </Header>
         <OrderID>12348A33</OrderID>
      </PanAliasRecovery>
      <PanAliasData>
              <PanAlias>0000197412081271677</PanAlias>
         <PanAliasRev></PanAliasRev>
         <PanAliasExpDate>2911</PanAliasExpDate>
         <PanAliasTail>0003</PanAliasTail>
         <MAC>E61612E0C0F71A2FE838BC0736B396E6</MAC>
      </PanAliasData>
   </Data>
</BPWXmlResponse>
```

The meaning of the elements is the following:

**<BPWXmlResponse>**

This is the root element of the document, there is only one element of this type in the message, which is composed of the following elements:

- **<Timestamp>** date and time of response message
- **<Result>** outcome of transaction requested

| Code | Description |
|---|---|
| 00 | Success |
| 02 | ReqRefNum duplicated or incorrect |
| 03 | Incorrect message format, missing or incorrect field |
| 04 | Incorrect API authentication, incorrect MAC |
| 06 | Unforeseen error during processing of request |
| 38 | Order not found or alias revoked |
| 40 | Empty xml or missing ‘date’ parameter |

| | |
|---|---|
| 41 | Xml not parsable |

- **<MAC>** signature of timestamp and result. See appendix 4.2.7
- **<Data>** data relating to the alias pan recovery and to the response message

**<Data>**

There is only one element of this type in the message, containing all the data relating to the alias recovery request and to the response message and consisting of the following elements:

- **< PanAliasRecoveryRequest>** data relating to the alias recovery request
- **< PanAliasData>** data relating to the recovered alias

**< PanAliasRecoveryRequest>**

There is only one element of this type in the message, containing all the data relating to the alias pan recovery request and consisting of the following elements:

- **<Header>** data relating to the request sent
- **<OrderId>** identifier of the alias-generating order

**<Header>**

There is only one element of this type in the message, which includes the date relating to the request sent. For the related description see paragraph 3.3.2 in the chapter "Response messages in XML".

**<PanAliasData>**

There is only one element of this type in the message, which includes the card alias data. For the related description see paragraph 3.3.5 in the chapter “Response messages in XML”

## 3.10.2 Pan alias revocation

The message for request of revocation of a pan alias allows to mark a given alias as no longer active.

The following cases may arise for the indicated alias pan:

1) No record found: an attempt is being made to revoke a non-existent alias → the revocation request is denied by the system with **outcome 38 (Alias Pan non-existent or not active).**
2) Record found but with alias not active → this means an attempt is being made to revoke a previously revoked alias → the system replies with outcome 00.
3) Found a record precisely with active alias → the system revokes the alias and replies with outcome 00.

For **API SIA VPOS** a **PANALIASREVOCATION** transaction request must be made.

The fields to be specified in the HTTP request message are the following:

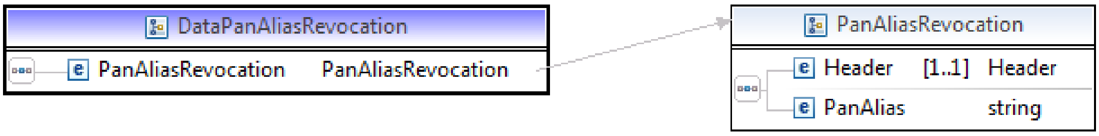
| Field | Compulsory | Size | Type | Description |
|---|---|---|---|---|
| | | | | |
| OPERATION | Y | | A | Transaction requested: filled in with "PANALIASREVOCATION" |
| TIMESTAMP | Y | 23 | AN | Local timestamp of the yyyy-MM-ddTHH:mm:ss.SSS type |
| SHOPID | Y | 15 | AN | Identifier of the merchant's store assigned by SIA, Merchant ID |
| OPERATORID | Y | Min. 8<br>Max.18 | AN | It indicates the person who has requested the transaction. |
| REQREFNUM | Y | 32 | N | Unique identifier of the request managed by the merchant. It can be used to retrieve information on the request made also in the case of no response. The first 8 digits must be in the yyyyMMdd format including the request date. |
| PANALIAS | Y | 19 | N | Pan alias |
| RELEASE | Y | 2 | AN | Release of APIs: to be filled in with "02" |
| OPTIONS | N | Min 0<br>Max 26 | AN | The field OPTIONS permits to activate various additional options for the ongoing message. |
| MAC | Y | 32, 40, 64 | AN | Message Authentication Code. For the related calculation see appendix 4.2.20 |

For **API SIA VPOS XML** a message must be generated specifying the PanAliasRevocation element.

## Request of pan alias revocation in XML format

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<BPWXmlRequest>
   <Release>02</Release>
   <Request>
      <Operation>PANALIASREVOCATION</Operation>
      <Timestamp>2015-03-04T11:20:00.000</Timestamp>
      <MAC>115025d5a5b65df687790867bdece136</MAC>
   </Request>
   <Data>
      <PanAliasRevocation>
         <Header>
            <ShopID>000000000000003</ShopID>
            <OperatorID>oper0001</OperatorID>
            <ReqRefNum>12345678901234567890123452289000</ReqRefNum>
         </Header>
         <PanAlias>0000123456789012345345</PanAlias>
      </PanAliasRevocation>
   </Data>
</BPWXmlRequest>
```

The data section of the response is described on the following scheme.



*Scheme 21 - Data section for PANALIASREVOCATION response*

Example of alias pan revocation response with API XML:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<BPWXmlResponse>
   <Timestamp>2015-07-04T12:02:55</Timestamp>
   <Result>00</Result>
   <MAC>ffb3553a4ab34c3a4ab34c3a4ab34c3a</MAC>
   <Data>
      <PanAliasRevocation>
         <Header>
            <ShopID>23486788</ShopID>
            <OperatorID>A4348B</OperatorID>
            <ReqRefNum>20150501496204690934584305834564</ReqRefNum>
         </Header>
         <PanAlias>0000123456789012345345</PanAlias>
      </PanAliasRevocation>
   </Data>
</BPWXmlResponse>
```

The meaning of the elements is the following:

**<BPWXmlResponse>**

This is the root element of the document, there is only one element of this type in the message, which is composed of the following elements:

- **<Timestamp>** date and time of response message
- **<Result>** outcome of transaction requested. Possible outcomes:

| **Code** | **Description** |
|---|---|
| 00 | Success |
| 02 | ReqRefNum duplicated or incorrect |
| 03 | Incorrect message format, missing or incorrect field |
| 04 | Incorrect API authentication, incorrect MAC |
| 06 | Unforeseen error during processing of request |
| **38** | **Alias Pan non-existent or not active** |
| 40 | Empty xml or missing 'date' parameter |
| 41 | Xml not parsable |
| 98 | Application error |
| 99 | Transaction failed, see specific outcome enclosed in the element <Data> of the response. |

- **<MAC>** signature of timestamp and result
- **<Data>** data relating to the authorization request and to the response message

**<Data>**

There is only one element of this type in the message, containing all the data relating to the revocation request and response message and consisting of the following elements:

- **<PanAliasRevocation>** data relating to the revocation request

**< PanAliasRevocation>**

There is only one element of this type in the message, containing all the data relating to the revocation request and consisting of the following elements:

- **< Header>**
- < **PanAlias**>

**<Header>**

There is only one element of this type in the message, which includes the date relating to the request sent. For the related description see paragraph 3.3.2 in the chapter "Response messages in XML".

<**PanAlias**> **Alias to be revoked**

### 3.10.3 List of  information on PAN ALIAS operations

The fields to be specified in the HTTP request message are the following:

| Field | Compulsory | Size | Type | Description |
|---|---|---|---|---|
| | | | | |
| OPERATION | Y | | A | Transaction requested: to be filled in with "LISTPANALIASINFO" |
| TIMESTAMP | Y | 23 | AN | Local timestamp of the yyyy-MM-ddTHH:mm:ss.SSS type |
| SHOPID | Y | 15 | AN | Identifier of the merchant's store assigned by SIA [MID] |
| OPERATORID | Y | Min. 8<br>Max.18 | AN | It indicates the person who has requested the transaction. |
| REQREFNUM | Y | 32 | N | Unique identifier of the request managed by the merchant. The first 8 digits must be in the yyyyMMdd format including the request date. |
| STARTDATE | N | 10 | D | Date of start of period, yyyy-MM-dd format |
| ENDDATE | N | 10 | D | Date of end of period, yyyy-MM-dd format |
| STARTTIME | N | 5 | D | Time of start of period, HH.mm format |
| ENDTIME | N | 5 | D | Time of end of period, HH.mm format |
| RELEASE | Y | 2 | AN | Release of APIs: to be filled in with "02" |
| OPTIONS | N | Min 0<br>Max 26 | AN | The field OPTIONS permits to activate various additional options for the ongoing message. |
| MAC | Y | 32, 40, 64 | AN | Message Authentication Code. For the related calculation see appendix 4.2.22 |

The API XML LISTPANALIASINFO extracts, within a period of time, the data concerning the two possible actions that may concern an alias: create or revoke.

The format used for presenting the information on one create alias action is the following:

```
<AliasCreated>
   <PanAliasRevHash />
   <PanAliasHash />
   <Timestamp />
   <OperatorId />
   <MAC />
</AliasCreated>
```

The action for the revocation of an alias can be the result of a revocation request or a creation request (automatic revocation of previous alias). In case of automatic revocation, the details of the revoked element will in any case be included in the list, given that the revocation timestamp of the revoked element matches the creation timestamp of the created element.

The format used for presenting the information on one revoke alias action is the following:

```
<AliasRevoked>
   <PanAliasHash />
   <Timestamp />
   <OperatorId />
   <TimestampRev />
   <OperatorIdRev />
   <MAC />
</AliasRevoked>
```

For calculating the MAC in the request message see appendix 4.2.22.

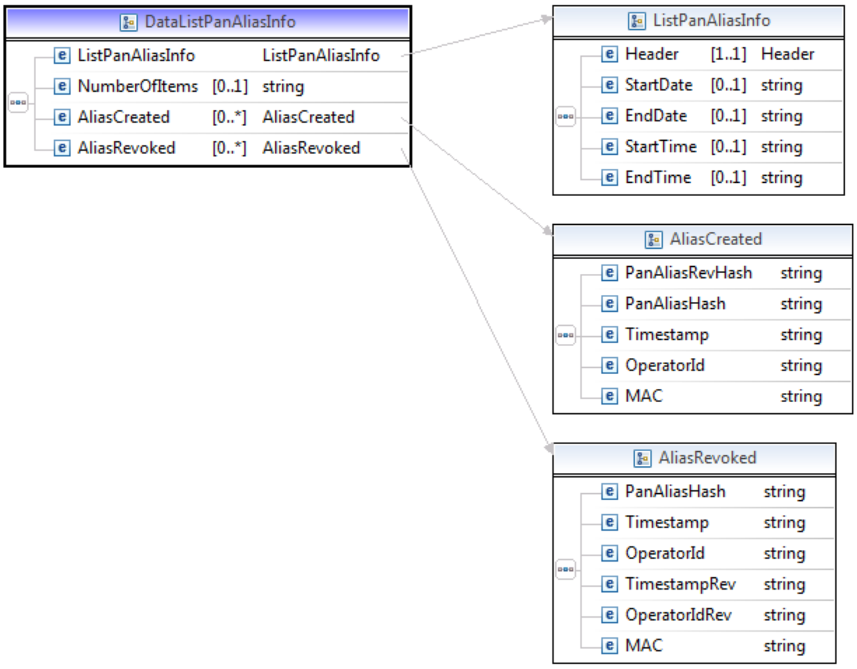
For calculating the MAC of the AliasCreated and AliasRevoked elements, see appendix 4.2.21.

## Request of list pan alias info in XML format

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<BPWXmlRequest>
   <Release>02</Release>
   <Request>
      <Operation>LISTPANALIASINFO</Operation>
      <Timestamp>2015-03-04T11:20:00.000</Timestamp>
      <MAC>115025d5a5b65df687790867bdece136</MAC>
   </Request>
   <Data>
      <ListPanAliasInfo>
         <Header>
            <ShopID>000000000000003</ShopID>
            <OperatorID>oper0001</OperatorID>
            <ReqRefNum>12345678901234567890123452289000</ReqRefNum>
         </Header>
         <StartDate>2014-12-01</StartDate>
         <EndDate>2014-12-31</EndDate>
         <StartTime>00.00</StartTime>
         <EndTime>18.25</EndTime>
      </ ListPanAliasInfo >
   </Data>
</BPWXmlRequest>
```

If an error occurs, the ListPanAliasInfo element will not be created.

The data section of the response is described on the following scheme.



*Scheme 22 - Data section for LISTPANALIASINFO response*

Below is an example of a file generated by the response to the request of LISTPANALIASINFO:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
```

```
<BPWXmlResponse>
   <Timestamp>2015-07-04T12:02:55</Timestamp>
   <Result>00</Result>
   <MAC>ffb3553a4ab34c3a4ab34c3a4ab34c3a</MAC>
   <Data>
      <ListPanAliasInfo>
         <Header>
            <ShopID>23486788</ShopID>
            <OperatorID>A4348B</OperatorID>
            <ReqRefNum>20150501496204690934584305834564</ReqRefNum>
         </Header>
         <StartDate>2014-12-01</StartDate>
         <EndDate>2014-12-31</EndDate>
         <StartTime>00.00</StartTime>
         <EndTime>18.25</EndTime>
      </ListPanAliasInfo>
      <NumberOfItems>2</NumberOfItems>
      <AliasCreated>
         <PanAliasRevHash>00.00</PanAliasRevHash>
         <PanAliasHash>00.00</PanAliasHash>
         <Timestamp>2015-07-04T12:02:55</Timestamp>
         <OperatorId>A4348B</OperatorId>
         <MAC>ffb3553a4ab34c3a4ab34c3a4ab34c3a</MAC>
      </AliasCreated>
      <AliasCreated>
         <PanAliasRevHash />
         <PanAliasHash>AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA</PanAliasHash>
         <Timestamp>2015-07-04T12:02:55</Timestamp>
         <OperatorId>A4348B</OperatorId>
         <MAC>ffb3553a4ab34c3a4ab34c3a4ab34c3a</MAC>
      </AliasCreated>
   </Data>
</BPWXmlResponse>
```

The meaning of the elements is the following:

**<BPWXmlResponse>**

This is the root element of the document, there is only one element of this type in the message, which is composed of the following elements:

- **<Timestamp>** date and time of response message
- **<Result>** outcome of transaction requested ("00" = list performed)

| Code | Description |
|---|---|
| 00 | Success |
| 02 | ReqRefNum duplicated or incorrect |
| 03 | Incorrect message format, missing or incorrect field |
| 04 | Incorrect API authentication, incorrect MAC |
| 05 | Incorrect date, or period indicated is empty |
| 06 | Unforeseen error during processing of request |
| 40 | Empty xml or missing 'date' parameter |
| 41 | Xml not parsable |
| 98 | Application error |
| 99 | Transaction failed, see specific outcome enclosed in the element <Data> of the response. |

- **<MAC>** signature of timestamp and result.
- **<Data>** data relating to the request

### <Data>

There is only one element of this type in the message, containing all the data relating to the request of a list and the response message and consisting of the following elements:

| | |
|---|---|
| **<ListPanAliasInfoRequest>** | data relating to the request of an alias pan list |
| **<ListPanAliasInfo>** | data relating to the alias pan list |

### <ListPanAliasInfoRequest>

There is only one element of this type in the message and it includes the data concerning the request of a list of alias pan represented by the following elements:

| | |
|---|---|
| **<Header>** | data concerning the submitted request |
| **<StartDate>** | start date of period of list |
| **<EndDate>** | end date of period of list |
| **<StartTime>** | start time of period of list, if indicated in the request |
| **<EndTime>** | end time of period of list, if indicated in the request |

### <Header>

There is only one element of this type in the message, which includes the date relating to the request sent. For the related description see paragraph 3.3.2 in the chapter  "Response messages in XML".

### < ListPanAliasInfo>

| | |
|---|---|
| **<AliasCreated>** | Data concerning alias created: N occurrences in the **ListPanAliasInfo** |
| **<AliasRevoked>** | Data concerning alias revoked: N occurrences in the **ListPanAliasInfo** |

### < AliasCreated>

| | |
|---|---|
| **<PanAliasRevHash>** | Hash of alias revoked, if any (SHA-1 of alias) |
| **<PanAliasHash>** | Hash of alias created |
| **<Timestamp>** | Timestamp of creation |
| **<OperatorId>** | Creation operator |
| **<MAC>** | Mac (see appendix 4.2.21) |

### < AliasRevoked>

| | |
|---|---|
| **<PanAliasHash>** | Hash of alias revoked, if any (SHA-1 of alias) |
| **<Timestamp>** | Timestamp of creation |
| **<OperatorId>** | Creation operator |
| **<TimestampRev>** | Revocation timestamp |
| **<OperatorIdRev>** | Revocation operator |
| **<MAC>** | Mac (see appendix 4.2.21) |

## 3.10.4 Request to generate PAN ALIAS

This transaction generates the pan alias for the pan or the billing agreement id specified in the message.
To be able to use this API, one of the Pan Alias services must be active and a specific enablement by SIA to the "SV68 - create PanAlias Service" is required.

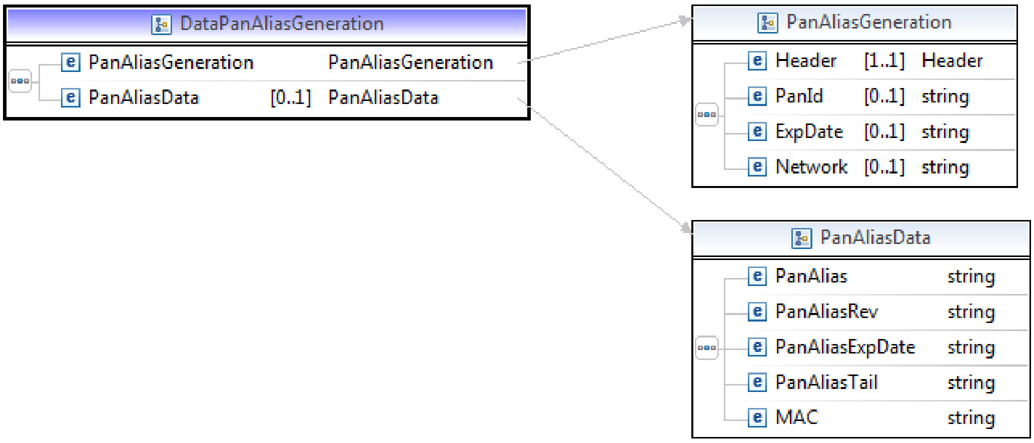
The fields to be specified in the request message are the following:

| Field | Compulsory | Size | Type | Description |
|---|---|---|---|---|
| | | | | |
| OPERATION | Y | | A | Transaction requested: to be filled in with "PANALIASGENERATION" |
| TIMESTAMP | Y | 23 | AN | Local timestamp of the yyyy-MM-ddTHH:mm:ss.SSS type |
| SHOPID | Y | 15 | AN | Identifier of the merchant’s store assigned by SIA [MID] |
| OPERATORID | Y | Min. 8<br>Max.18 | AN | It indicates the person who has requested the transaction. |
| REQREFNUM | Y | 32 | N | Unique identifier of the request managed by the merchant. The first 8 digits must be in the yyyyMMdd format including the request date. |
| PANID | Y | 16 | N | Card PAN, used to create the PAN alias (or billing agreement ID for Paypal) |
| EXPDATE | N | 4 | N | Expiry date of the card specified in the “PANID” field, in yyMM format |
| NETWORK | Y | 2 | N | Numerical code representing the network associated with the card for which the PAN alias is requested; it represents:<br>• 01 → Visa<br>• 02 → Mastercard<br>• 04 → Maestro<br>• 06 → Amex<br>• 97 → Paypal |
| RELEASE | Y | 2 | AN | Release of APIs: to be filled in with “02” |
| OPTIONS | N | Min 0<br>Max 26 | AN | The field OPTIONS permits to activate various additional options for the ongoing message. |
| MAC | Y | 32, 40 o 64 | AN | Message Authentication Code. For the related calculation see appendix 4.2.27 |

### Request of pan alias generation in XML format

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<BPWXmlRequest>
   <Release>02</Release>
   <Request>
      <Operation>PANALIASGENERATION</Operation>
      <Timestamp>2015-03-04T11:20:00.000</Timestamp>
      <MAC>115025d5a5b65df687790867bdece136</MAC>
   </Request>
   <Data>
      <PanAliasGeneration>
         <Header>
            <ShopID>000000000000003</ShopID>
            <OperatorID>oper0001</OperatorID>
            <ReqRefNum>12345678901234567890123452289000</ReqRefNum>
         </Header>
         <PanId>0000123456789012345345</PanId>
   <ExpDate>2101</ExpDate>
   <Network>02</Network>
      </PanAliasGeneration>
   </Data>
</BPWXmlRequest>
```

The response message to a pan alias generation request is formatted in XML. The data section is described on the following scheme.



***Scheme 23 - Data section for PANALIASGENERATION response***

Here below is an example of a file generated by the response to a pan alias generation request:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<BPWXmlResponse>
   <Timestamp>2015-07-04T12:02:55</Timestamp>
   <Result>00</Result>
   <MAC>ffb3553a4ab34c3a4ab34c3a4ab34c3a</MAC>
   <Data>
      <PanAliasGeneration>
         <Header>
            <ShopID>23486788</ShopID>
            <OperatorID>A4348B</OperatorID>
            <ReqRefNum>20150501496204690934584305834564</ReqRefNum>
         </Header>
         <PanId>0000123456789012345345</ PanId>
         <ExpDate>2101</ExpDate>
         <Network>02</Network>
      </PanAliasGeneration>
      <PanAliasData>
         <PanAlias>0000197412081271677</PanAlias>
         <PanAliasRev></PanAliasRev>
         <PanAliasExpDate>2911</PanAliasExpDate>
         <PanAliasTail>0003</PanAliasTail>
         <MAC>E61612E0C0F71A2FE838BC0736B396E6</MAC>
   </PanAliasData>
   </Data>
</BPWXmlResponse>
```

The meaning of the elements is the following:

**<BPWXmlResponse>**

This is the root element of the document, there is only one element of this type in the message, which is composed of the following elements:

- **<Timestamp>** date and time of response message

- **<Result>** outcome of transaction requested ("00" = alias pan created)

| Code | Description |
|---|---|
| 00 | Success |
| 02 | ReqRefNum duplicated or incorrect |
| 03 | Incorrect message format, missing or incorrect field |
| 04 | Incorrect API authentication, incorrect MAC |
| 05 | Incorrect date, or period indicated is empty |
| 06 | Unforeseen error during processing of request |
| 40 | Empty xml or missing 'date' parameter |
| 41 | Xml not parsable |
| 98 | Application error |
| 99 | Transaction failed, see specific outcome enclosed in the element <Data> of the response. |

- **<MAC>** signature of timestamp and result (see appendix 4.2.27).
- **<Data>** data relating to the request

**<Data>**

There is only one element of this type in the message, containing all the data relating to the creation request and the response message and consisting of the following elements:

- **<PanAliasGenerationRequest>** data related to request to create alias pan
- **<PanAliasData>** data related to results of alias pan creation

**< PanAliasGeneration>**

There is only one element of this type in the message, containing all the data relating to the request to create alias pan and consisting of the following elements:

- **<Header>** data related to request submitted
- **<PanID>** pan of card for which the creation of the alias pan was requested
- **<ExpDate>** expiry date of the card for which the creation of the pan alias was requested
- **<Network>** the network of the card for which the creation of the pan alias was requested

**<Header>**

There is only one element of this type in the message, which includes the date relating to the request sent. For the related description see paragraph 3.3.2 in the chapter "Response messages in XML".

**< PanAliasData>**

There is only one element of this type in the message, which includes the card alias data. For the related description see paragraph 3.3.5 in the chapter "Response messages in XML"

# 3.11 Pay-By-Link Transactions

## 3.11.1 Request to create link

The message requesting the creation of the link allows to create a payment link.
The fields to be specified in the HTTP request message are the following:

| Field | Compulsory | Size | Type | Description |
|---|---|---|---|---|
| | | | | |
| OPERATION | Y | | A | Transaction requested: to be filled in with "CREATELINK" |
| TIMESTAMP | Y | 23 | AN | Local timestamp of the yyyy-MM-ddTHH:mm:ss.SSS type |
| REQREFNUM | Y | 32 | N | Unique identifier of the request managed by the merchant. It can be used to retrieve information on the request made also in the case of no response. The first 8 digits must be in the yyyyMMdd format including the request date. |
| SHOPID | Y | 15 | AN | Identifier of the merchant's store assigned by SIA [MID] |
| OPERATORID | Y | Min 8 Max 18 | AN | It indicates the person who has requested the transaction. |
| SENDMAIL | Y | 1 | AN | It can have the following values: Y (yes) or N (no). If Y, the system will send the email and create the link at the same time. |
| RELEASE | Y | 2 | AN | Release of APIs: to be filled in with "02" |
| OPTIONS | N | Min 0 Max 26 | AN | The field OPTIONS permits to activate various additional options for the ongoing message. |
| MAC | Y | 32, 40 o 64 | AN | Request signature field: it makes the data sent unchangeable by third parties. For the calculation see appendix 4.2.23. |
| LINKEXPIRATION DATE | N | 23 | AN | Indication of the date of the end of validity of the link in the format yyyy-MM-ddTHH:mm:ss.SSS, if it has no value, the validity indicated in the store registered data will be used. |
| LINKAMOUNT | Y | Min 2 Max 8 | N | Amount expressed in the smallest currency unit (EUR cents).<br>The amount must not be preceded by zeros.<br>Minimum length 1 maximum length 8.<br>For ASI card verification transactions the amount MUST be set to 0. For real transactions the amount has to be at least the minimum supported by the merchant's acquirer (usually 10 cent for euro). |
| LINKCURRENCY | Y | 3 | N | Currency: ISO code (EUR = 978). |
| LINKEXPONENT | N | 1 | N | Exponent of the chosen Currency<br>(Recommended if the currency is different from Euro) |
| LINKORDERID | Y | 50 | AN | Single identifier of the order. **Its uniqueness must be guaranteed for at least 5 years**. Admitted characters include letters, figures, "-" and "_". The regular expression [a-zA-Z0-9\-_] is applied. It is not possible to request that a link be created for an order number already used for a link than can still be used. |
| LINKURLDONE | N* | 254 | AN | Complete URL to which the client's browser is to be redirected upon successful outcome of the transaction (it may include all of the parameters to be passed). The system will hang the outcome parameters on it. Maximum length 254 characters. In case of absence of the parameter the outcome page will not show the related button.<br>* the field becomes compulsory if options G or N are specified |
| LINKURLMS | Y | 254 | AN | URL of the merchant system toward which SIA carries out the GET or POST confirming the effected payment (it may contain any |

| | | | | |
|---|---|---|---|---|
| | | | | parameters set by the store). The system hangs the outcome parameters on it. In case of absence of the parameter the system will not send the payment outcome to the operator via server-to-server communication. |
| LINKACCOUNTINGMODE | Y | 1 | A | Type of accounting to be used for this order:<br>• D deferred<br>• I immediate |
| LINKAUTHORMODE | Y | 1 | A | Immediate authorization only is available. The field must have the value of I. |
| LINKLANG | Y | 3 | A | Language in which the messages of interaction with the final user must be shown. The field is optional; the default language is Italian. The following are currently available:<br>ITA Italian<br>EN English |
| LINKSHOPEMAIL | N | 50 | AN | It contains the email address to which the transaction outcome is to be sent. If it is not present, the one available in the store data registered at SIA will be used. Minimum length 7 alphanumerical characters maximum length 50 |
| LINKOPTIONS | N | 10 | A | It contains the indicators of the additional options that are to be activated for the payment under way. The order in which the options appear is irrelevant. The content of the field is not case sensitive. See corresponding paragraph for further details.<br>The value of W cannot be used via PBL.<br>If options G and N are requested, the URLDONE field will become compulsory.<br>The system assumes that option B is always present. |
| LINKCOMMIS | N | Min 1<br>Max 8 | N | Amount of commission on the service expressed in the miminum unit of the value (EURO cents). Bear in mind that the AMOUNT parameter includes the commission.<br>NOTE: the COMMIS parameter, if present, is relevant if and only if the additional option OPTIONS = F has also been set (payment with commission on service).<br>Minimum length 1 maximum length 8. |
| LINKEMAIL | N | Min 7<br>Max 50 | AN | Email address of client addressee of the link.<br>If the field is not present, it will be requested from the user together with the credit card data.<br>Compulsory if the field SENDMAIL is "Y".<br>Minimum length 7 alphanumerical characters maximum length 50. |
| LINKNAME | Y | 40 | A | Name of holder of payment instrument |
| LINKSURNAME | Y | 40 | A | Last name of holder of payment instrument |
| LINKORDDESCR | N | 140 | AN | Description of order (see OPTIONS O and V).<br>Maximum length 140. |
| LINKOPDESCR | N | 100 | AN | Additional description of accounting transaction, at operator's discretion (for immediate accounting only).<br>Maximum length 100. |

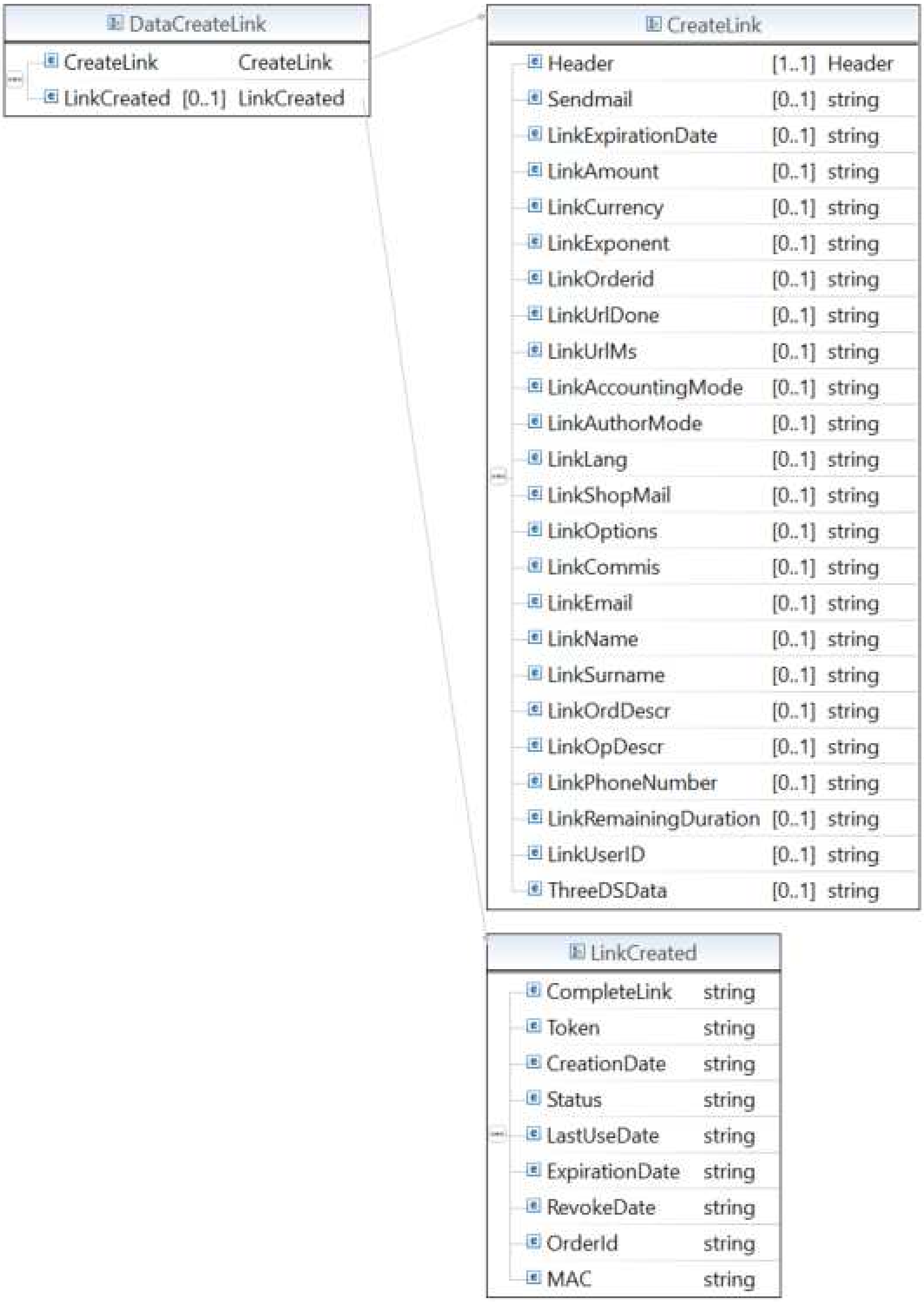
| | | | | |
|---|---|---|---|---|
| LINKPHONENUMBER | N | Min 10<br>Max 14 | N | Telephone number to be assigned beforehand for transactions carried out with the Jiffy circuit |
| LINKREMAININGDURATION | N | Num<br>0-60 | N | Minimum number of months of residual duration of the card used (associated with option D). Number from 0 to 60. |
| LINKUSERID | N | 255 | AN | User identifier generated by merchant associated with payment |
| LINKPRODUCTREF | N | Max 15 | AN | Sale identifier |
| LINKTRECURR | N | 1 | AN | Type of recurring payment.<br>Mandatory for a recurring payment or with OPTION M (if SVA4 is not active).<br>The admitted values are:<br>R – First of a scheduled **R**ecurring transaction<br>U – First of an **U**nscheduled recurring transaction<br>C – **C**ard stored on file (pan alias/token) notification (one shot)<br>For granted authorization a CRECURR will be sent back to the merchant to be used for the following recurring payments. |
| LINKCRECURR | N | Max 50 | AN | For LINKTRECURR=C may contain the previously received CRECURR. |
| THREEDSDATA | N | 5000 | AN | 3DS data payload containing additional information used for the 3D secure 2.0. |

The result of the request to create a link is a complete link in the following form:
https://atpos.ssb.it/atpos/pagamenti/main?PAGE=PBL&TOKEN=1qwpt99pslotul1budkx3f712

## Request of a link creation in XML format

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<bpwxmlrequest>
   <release>02</release>
   <request>
      <operation>CREATELINK</operation>
      <timestamp>2017-07-07T15:48:00.690</timestamp>
      <mac>a307e7321f5ca31f6cc755a10e478b6af598e6b3650e15831487541572b4e563</mac>
      <release>02</release>
   </request>
   <data>
      <createlinkrequest>
         <header>
            <shopid>129280505999998</shopid>
            <operatorid>AF06TSTAPI1</operatorid>
            <reqrefnum>20170707154800000000000000000000</reqrefnum>
         </header>
         <sendmail>N</sendmail>
         <linkexpirationdate>2017-07-12T15:48:06.000</linkexpirationdate>
         <linkamount>9900</linkamount>
         <linkcurrency>978</linkcurrency>
         <linkexponent>2</linkexponent>
         <linkorderid>API20170707154800</linkorderid>
         <linkurldone></linkurldone>
         <linkurlms>https://atpostest.ssb.it/atpos/pagamenti/main?PAGE</linkurlms>
         <linkaccountingmode>I</linkaccountingmode>
         <linkauthormode>I</linkauthormode>
         <linklang>EN</linklang>
         <linkshopemail>OperatorMail@mail.com</linkshopemail>
         <linkoptions></linkoptions>
         <linkcommis></linkcommis>
         <linkemail>CardholderMail@mail.com</linkemail>
         <linkname>Cardholder Name</linkname>
         <linksurname>Cardholder Surname</linksurname>
         <linkorddescr></linkorddescr>
         <linkopdescr></linkopdescr>
         <linkphonenumber></linkphonenumber>
         <linkremainingduration></linkremainingduration>
         <linkuserid>jonsnow72</linkuserid>
         <linktrecurr>C</linktrecurr>
         <linkcrecurr>PST581426946</linkcrecurr>
         <threedsdata>kBzKJjxOXJjtQkH4AVArkjjiZ...</threedsdata>
      </createlinkrequest>
   </data>
</bpwxmlrequest>
```

The data section of the response is described on the following scheme.



| DataCreateLink | | |
|---|---|---|
| CreateLink | | CreateLink |
| LinkCreated | [0..1] | LinkCreated |

| CreateLink | | |
|---|---|---|
| Header | [1..1] | Header |
| Sendmail | [0..1] | string |
| LinkExpirationDate | [0..1] | string |
| LinkAmount | [0..1] | string |
| LinkCurrency | [0..1] | string |
| LinkExponent | [0..1] | string |
| LinkOrderid | [0..1] | string |
| LinkUrlDone | [0..1] | string |
| LinkUrlMs | [0..1] | string |
| LinkAccountingMode | [0..1] | string |
| LinkAuthorMode | [0..1] | string |
| LinkLang | [0..1] | string |
| LinkShopMail | [0..1] | string |
| LinkOptions | [0..1] | string |
| LinkCommis | [0..1] | string |
| LinkEmail | [0..1] | string |
| LinkName | [0..1] | string |
| LinkSurname | [0..1] | string |
| LinkOrdDescr | [0..1] | string |
| LinkOpDescr | [0..1] | string |
| LinkPhoneNumber | [0..1] | string |
| LinkRemainingDuration | [0..1] | string |
| LinkUserID | [0..1] | string |
| ThreeDSData | [0..1] | string |

| LinkCreated | |
|---|---|
| CompleteLink | string |
| Token | string |
| CreationDate | string |
| Status | string |
| LastUseDate | string |
| ExpirationDate | string |
| RevokeDate | string |
| OrderId | string |
| MAC | string |

*Scheme 24 - Data section for CREATELINK response*

Example of response to CREATELINK:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<BPWXmlResponse>
   <Timestamp>2017-07-07T15:48:06</Timestamp>
   <Result>00</Result>
   <!-- This MAC signs timestamp and result -->
   <MAC>0fd153689a096b438e4b78b4ce0745cea9e14502b2ef4390f214ba50f7580671
   </MAC>
   <Data>
      <!-- This element contains request data -->
      <CreateLink>
         <Header>
            <Shopid>129280505999998</Shopid>
            <Operatorid>AF06TSTAPI1</Operatorid>
            <Reqrefnum>20170707154800000000000000000000</Reqrefnum>
         </Header>
         <Sendmail>N</Sendmail>
         <LinkExpirationDate>2017-07-12T15:48:06.000</LinkExpirationDate>
         <LinkAmount>9900</LinkAmount>
         <LinkCurrency>978</LinkCurrency>
         <LinkExponent>2</LinkExponent>
         <LinkOrderid>API20170707154800</LinkOrderid>
         <LinkUrlDone></LinkUrlDone>
         <LinkUrlMs>https://atpostest.ssb.it/atpos/pagamenti/main?PAGE
         </LinkUrlMs>
         <LinkAccountingMode>I</LinkAccountingMode>
         <LinkAuthorMode>I</LinkAuthorMode>
         <LinkLang>EN</LinkLang>
         <LinkShopMail>OperatorMail@mail.com</LinkShopMail>
         <LinkOptions></LinkOptions>
         <LinkCommis></LinkCommis>
         <LinkEmail>CardholderMail@mail.com</LinkEmail>
         <LinkName>Cardholder Name</LinkName>
         <LinkSurname>Cardholder Surname</LinkSurname>
         <LinkOrdDescr></LinkOrdDescr>
         <LinkOpDescr></LinkOpDescr>
         <LinkPhoneNumber></LinkPhoneNumber>
         <LinkRemainingDuration></LinkRemainingDuration>
         <LinkUserID>jonsnow</LinkUserID>
         <LinkTRecurr>C</LinkTRecurr>
         <LinkCRecurr>PST581426946</LinkCRecurr>
      </CreateLink>
      <LinkCreated>
         <CompleteLink>https://atpostest.ssb.it/atpos/pagamenti/main?
            PAGE=PBL&amp;TOKEN=scxfuyegan23510hi68vm7s12</CompleteLink>
         <Token>scxfuyegan23510hi68vm7s12</Token>
         <CreationDate>2017-07-07T15:48:00.690</CreationDate>
         <Status>01</Status>
         <LastUseDate></LastUseDate>
         <ExpirationDate>2017-07-12T15:48:06.000848</ExpirationDate>
         <RevokeDate></RevokeDate>
         <OrderId>API20170707154800</OrderId>
         <MAC>e48af2e91f0c4c0e79617ed07ac2b3ba445a940d946e83e390a08023c92fee09
         </MAC>
      </LinkCreated>
   </Data>
</BPWXmlResponse>
```

The meaning of the elements is the following:

**<BPWXmlResponse>**

This is the root element of the document, there is only one element of this type in the message, which is composed of the following elements:

- **<Timestamp>** date and time of response message
- **<Result>** outcome of transaction requested

| Code | Description |
|---|---|

| | |
|---|---|
| 00 | Success |
| 02 | ReqRefNum duplicated or incorrect |
| 03 | Incorrect message format, missing or incorrect field |
| 04 | Incorrect API authentication, incorrect MAC |
| 06 | Unforeseen error during processing of request |
| 13 | Duplicated order number |
| 40 | Empty xml or missing ‘date’ parameter |
| 41 | Xml not parsable |
| 98 | Application error |
| 99 | Transaction failed, see specific outcome enclosed in the element <Data> of the response. |

- **<MAC>** signature of timestamp and result (see appendix 4.2.7)
- **<Data>** data relating to request and response

### <Data>

There is only one element of this type in the message, containing all the data relating to the request of a list and the response message and consisting of the following elements:

- **< CreateLink>** data relating to the request to create a link
- **<LinkCreated>** data relating to the link created

### <CreateLink>

There is only one element of this type in the message and it includes the data related to the request to create a link represented by the following elements:

- **<Header>** data related to the request sent
- **<Sendmail>** request to send payment email with link
- **<LinkExpirationDate>** end of validity of link created
- **<LinkAmount>** amount of payment created with link
- **<LinkCurrency>** currency of payment created with link
- **<LinkExponent>** number of decimals for the currency of payment created with link
- **<LinkOrderid>** order number of payment created with link
- **<LinkUrlDone>** url of redirect at the end of the payment created with link
- **<LinkUrlMs>** url of server notifying outcome of the payment created wih the link
- **<LinkAccountingMode>** type of accounting of the payment created with link
- **<LinkAuthorMode>** type of authorization of the payment created with link
- **<LinkLang>** language of the payment created with link
- **<LinkShopMail>** email operator of the payment created with link
- **<LinkOptions>** options of the payment created with link
- **<LinkCommis>** commissions of the payment created with link
- **<LinkEmail>** email holder of the payment created with link
- **<LinkName>** name of holder of the payment created with link
- **<LinkSurname>** last name of holder of the payment created with link
- **<LinkOrdDescr>** description of the order of payment created with link
- **<LinkOpDescr>** description of the accounting transaction of the payment created with link
- **<LinkPhoneNumber>** telephone number of the payment created with link (for Jiffy)
- **<LinkRemainingDuration>** minimum duration of validity of a card of the payment created with link
- **<LinkUserID>** userid of the payment created with link
- **<LinkTRecurr>** type of recurring payment of the payment created with link

| | |
|---|---|
| <LinkTRecurr> | recurring code of the payment created with link |

For the details for all the "Link" fields see the integration specifications of the Redirect mode.

**<Header>**

There is only one element of this type in the message, which includes the date relating to the request sent. For the related description see paragraph 3.3.2 in the chapter "Response messages in XML".

**<LinkCreated>**

| | |
|---|---|
| <CompleteLink> | Complete link to start payment |
| <Token> | Identification token of the payment from link |
| <CreationDate> | Date of creation of link |
| <Status> | Link status |
| <LastUseDate> | Date of last use of link |
| <ExpirationDate> | Expiry date of link |
| <RevokeDate> | Date of revocation of link |
| <OrderId> | Order number of the payment |
| <MAC> | signature of data present in the link created (see appendix 4.2.26) |

The element <Status> can have the following values and meanings:

| Status | Description | Meaning |
|---|---|---|
| 00 | Created | The link has been created correctly (technical status, not visible from the outside) |
| 01 | Returned | The link has been communicated to the operator |
| 02 | Sent | The link has been sent to the holder |
| 03 | Used | The link has been used at least once |
| 04 | Paid | The payment tied to the link has been completed |
| 05 | Revoked | The link has been revoked |

## 3.11.2 Request of list of links created

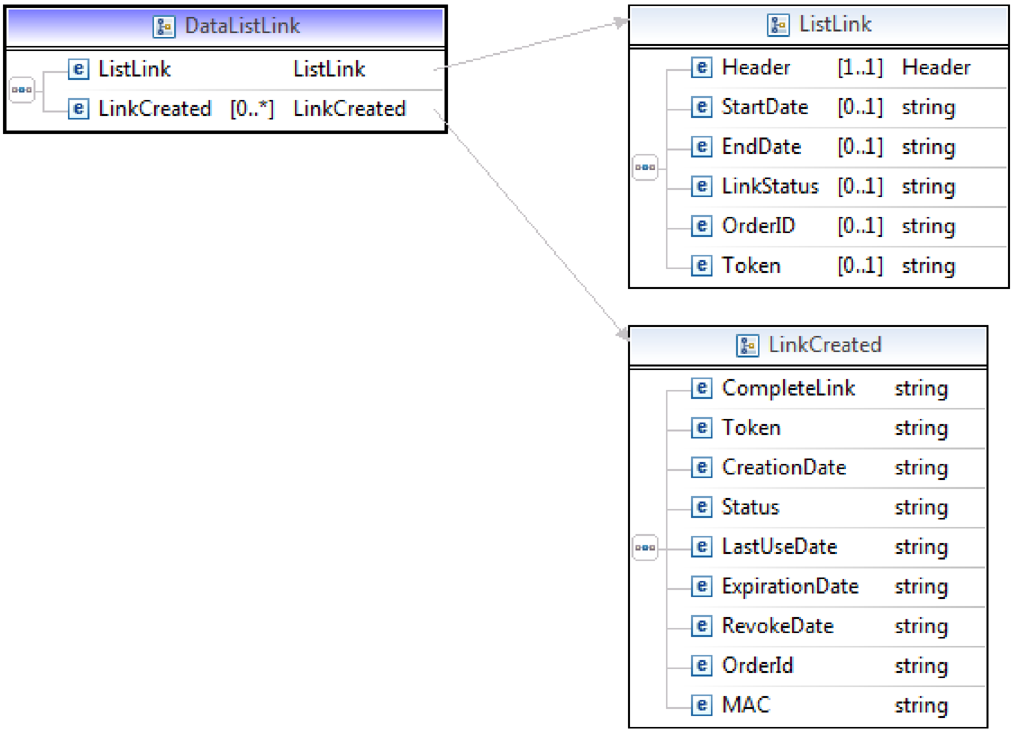
The message requesting a list of links allow to search the payment links created.
The fields to be specified in the request message are the following:

| Field | Compulsory | Size | Type | Description |
|---|---|---|---|---|
| | | | | |
| OPERATION | Y | | A | Transaction requested: filled in with "LISTLINK" |
| TIMESTAMP | Y | 23 | AN | Local timestamp of the yyyy-MM-ddTHH:mm:ss.SSS type |
| REQREFNUM | Y | 32 | N | Unique identifier of the request managed by the merchant. It can be used to retrieve information on the request made also in the case of no response. The first 8 digits must be in the yyyyMMdd format including the request date. |
| SHOPID | Y | 15 | AN | Identifier of the merchant's store assigned by SIA [MID] |
| OPERATORID | Y | Min. 8<br>Max. 18 | AN | It indicates the person who has requested the transaction. |
| RELEASE | Y | 2 | AN | Release of APIs: to be filled in with "02" |
| OPTIONS | N | Min 0<br>Max 26 | AN | The field OPTIONS permits to activate various additional options for the ongoing message. |
| MAC | Y | 32, 40 o 64 | AN | Request signature field: it makes the data sent unchangeable by third parties. For the calculation see appendix 4.2.24. |
| STARTDATE | Y | 10 | AN | Search start date, yyyy-MM-dd format |
| ENDDATE | Y | 10 | AN | Search end date, yyyy-MM-dd format |
| LINKSTATUS | N | 2 | N | Link status |
| ORDERID | N | 50 | AN | ID of the order associated with the link |
| TOKEN | N | 25 | AN | Token associated with the payment transaction stored during the creation stage |

### Request of a list of links in XML format

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<bpwxmlrequest>
   <release>02</release>
   <request>
      <operation>LISTLINK</operation>
      <timestamp>2017-07-07T16:31:55.916</timestamp>
      <mac>e3e25cc0886624eadbd4d0370ad8471d6b2c12268f5b4bbf398f7f5809a1b831</mac>
   </request>
   <data>
      <listlinkrequest>
         <header>
            <shopid>129280505999998</shopid>
            <operatorid>Operator123</operatorid>
            <reqrefnum>20170707163155000000000000000000</reqrefnum>
         </header>
         <token></token>
         <startdate>2017-07-01</startdate>
         <enddate>2017-07-31</enddate>
         <linkstatus></linkstatus>
         <orderid></orderid>
      </listlinkrequest>
   </data>
</bpwxmlrequest>
```

The data section of the response is described on the following scheme.



*Scheme 25 - Data section for LISTLINK response*

Example of response to LISTLINK:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<BPWXmlResponse>
   <Timestamp>2017-07-07T16:32:23</Timestamp>
   <Result>00</Result>
   <!-- This MAC signs timestamp and result -->
   <MAC>4fa442eff00a888fd602951f2cfef268f66bd28dca4bbf0b954cd7978cfcd378</MAC>
   <Data>
      <!-- This element contains request data -->
      <ListLink>
         <Header>
            <ShopId>129280505999998</ShopId>
            <Operatorid>Operator123</Operatorid>
            <Reqrefnum>20170707163155000000000000000000</Reqrefnum>
         </Header>
         <StartDate>2017-07-01</StartDate>
         <EndDate>2017-07-31</EndDate>
         <LinkStatus></LinkStatus>
         <OrderId></OrderId>
         <Token></Token>
      </ListLink>
      <LinkCreated>
         <CompleteLink>https://atpostest.ssb.it/atpos/pagamenti/main?
            PAGE=PBL&amp;TOKEN=scxfuyegan235l0hi68vm7s12</CompleteLink>
         <Token>scxfuyegan235l0hi68vm7s12</Token>
         <CreationDate>2017-07-07T15:48:00.690</CreationDate>
         <Status>01</Status>
         <LastUseDate></LastUseDate>
         <ExpirationDate>2017-07-12T15:48:06.000848</ExpirationDate>
         <RevokeDate></RevokeDate>
         <OrderId>API20170707154800</OrderId>
         <MAC>1f6fc3fb7a878488f769d81ace4240ad1a933e8314d556f085d05d058bacef4d</MAC>
      </LinkCreated>
```

```
    </Data>
</BPWXmlResponse>
```

The meaning of the elements is the following:

**<BPWXmlResponse>**

This is the root element of the document, there is only one element of this type in the message, which is composed of the following elements:

- **<Timestamp>** date and time of response message
- **<Result>** outcome of transaction requested ("00" = list performed)

| Code | Description |
|---|---|
| 00 | Success |
| 02 | ReqRefNum duplicated or incorrect |
| 03 | Incorrect message format, missing or incorrect field |
| 04 | Incorrect API authentication, incorrect MAC |
| 06 | Unforeseen error during processing of request |
| 40 | Empty xml or missing 'date' parameter |
| 41 | Xml not parsable |
| 52 | No link found with the preset search criteria |
| 98 | Application error |
| 99 | Transaction failed, see specific outcome enclosed in the element <Data> of the response. |

- **<MAC>** signature of timestamp and result (see appendix 4.2.7)
- **<Data>** data relating to request and response

**<Data>**

There is only one element of this type in the message, containing all the data relating to the request of a list and the response message and consisting of the following elements:

- **<ListLink>** data relating to the request of a list of links
- **<LinkCreated>** data relating to the created link meeting search criteria

**<ListLink>**

There is only one element of this type in the message and it includes the data concerning the request of a list of links represented by the following elements:

- **<Header>** data concerning the submitted request
- **<StartDate>** start date of search of links
- **<EndDate>** end date of search of links
- **<LinkStatus>** search only links in a given status
- **<OrderId>** search only links associated with a given order id
- **<Token>** search only link associated with a given token

**<Header>**

There is only one element of this type in the message, which includes the date relating to the request sent. For the related description see paragraph 3.3.2 in the chapter "Response messages in XML".

**<LinkCreated>**

The response will contain the list of the elements <LinkCreated> corresponding to the specified search parameters. Each <LinkCreated> element will contain the related signature. If the indicated search parameters do not correspond to any request, there will be no <LinkCreated> element.

| | |
|---|---|
| • **<CompleteLink>** | Complete link to start payment |
| • **<Token>** | Identification token of the payment from link |
| • **<CreationDate>** | Date of creation of link |
| • **<Status>** | Link status |
| • **<LastUseDate>** | Date of last use of link |
| • **<ExpirationDate>** | Expiry date of link |
| • **<RevokeDate>** | Date of revocation of link |
| • **<OrderId>** | Order number of the payment |
| • **<MAC>** | signature of data present in the link created (see appendix 4.2.26) |

For the values and meanings assigned to the <Status> element, see the chapter on Request to create link.

### 3.11.3 Request to revoke a link

The ‘revoke link’ functionality via API allows the operator to request the revocation of a link.

The link revocation process brings the link indicated in the REVOKED status and marks the date on which the revocation has been carried out.

It is possible to request the revocation of the link only if in a status other than PAID and REVOKED. If a request is made to revoke an order in the PAID or already REVOKED status, the request will be denied with an error message.

The revocation process has no effect on the links for which the holder has started the payment process: if the link is valid when the holder is redirected to the payment interface, it will be possible to complete the payment

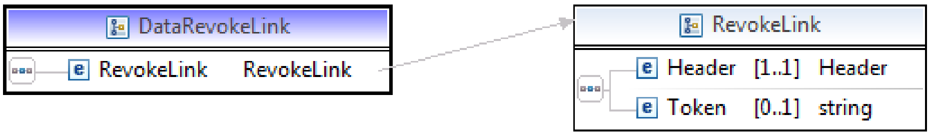
The fields to be specified in the request message are the following:

| Field | Compulsory | Size | Type | Description |
|---|---|---|---|---|
| OPERATION | Y | | A | Transaction requested: filled in with "REVOKELINK" |
| TIMESTAMP | Y | 23 | AN | Local timestamp of the yyyy-MM-ddTHH:mm:ss.SSS type |
| REQREFNUM | Y | 32 | N | Unique identifier of the request managed by the merchant. It can be used to retrieve information on the request made also in the case of no response. The first 8 digits must be in the yyyyMMdd format including the request date. |
| SHOPID | Y | 15 | AN | Identifier of the merchant’s store assigned by SIA [MID] |
| OPERATORID | Y | Min. 8<br>Max. 18 | AN | It indicates the person who has requested the transaction. |
| TOKEN | Y | 25 | AN | Token associated with the payment transaction stored during the creation stage |
| RELEASE | Y | 2 | AN | Release of APIs: to be filled in with “02” |
| OPTIONS | N | Min 0<br>Max 26 | AN | The field OPTIONS permits to activate various additional options for the ongoing message. |
| MAC | Y | 32, 40 o 64 | AN | Request signature field: it makes the data sent unchangeable by third parties. For the calculation see appendix 4.2.25. |

#### Request of a revoke of link in XML format

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<bpwxmlrequest>
   <release>02</release>
   <request>
      <operation>REVOKELINK</operation>
      <timestamp>2017-07-07T17:21:54.262</timestamp>
      <mac>2e0d575b0960b7650e9763558d0f9d1ed6874f321f81e456cb2fdd9ea70e08e1</mac>
   </request>
   <data>
      <revokelinkrequest>
         <header>
            <shopid>129280505999998</shopid>
            <operatorid>Operator123</operatorid>
            <reqrefnum>20170707172155000000000000000000</reqrefnum>
         </header>
         <token>scxfuyegan23510hi68vm7s12</token>
      </revokelinkrequest>
   </data>
</bpwxmlrequest>
```

The data section of the response is described on the following scheme.



*Scheme 26 - Data section for REVOKELINK response*

Example of response to REVOKELINK:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<BPWXmlResponse>
  <Timestamp>2017-07-07T17:22:01</Timestamp>
  <Result>00</Result>
  <!-- This MAC signs timestamp and result -->
  <MAC>1b85c672ff4a4e7c98fbfb1a11f58689ba940b64405e2e488033b4852e75ee1a</MAC>
  <Data>
    <!-- This element contains request data -->
        <RevokeLink>
           <Header>
              <ShopId>129280505999998</ShopId>
              <Operatorid>Operator123</Operatorid>
              <Reqrefnum>20170707172155000000000000000000</Reqrefnum>
           <Header>
           <Token>scxfuyegan23510hi68vm7s12</Token>
        </RevokeLink>
  </Data>
</BPWXmlResponse>
```

The meaning of the elements is the following:

**<BPWXmlResponse>**

This is the root element of the document, there is only one element of this type in the message, which is composed of the following elements:

- **<Timestamp>** date and time of response message
- **<Result>** outcome of transaction requested

| Code | Description |
|---|---|
| 00 | Success |
| 02 | ReqRefNum duplicated or incorrect |
| 03 | Incorrect message format, missing or incorrect field |
| 04 | Incorrect API authentication, incorrect MAC |
| 06 | Unforeseen error during processing of request |
| 40 | Empty xml or missing ‘date’ parameter |
| 41 | Xml not parsable |
| 50 | Token not found |
| 51 | Status of link not valid (link cannot be revoked) |
| 52 | No link found with the preset search criteria |
| 98 | Application error |
| 99 | Transaction failed, see specific outcome enclosed in the element <Data> of the response. |

- **<MAC>** signature of timestamp and result (see appendix 4.2.7)
- **<Data>** data relating to request and response

### <Data>

There is only one element of this type in the message, containing all the data relating to the request to revoke a link and consisting of the following elements:

| | |
|---|---|
| • **<RevokeLink>** | data relating to the request to revoke a link |

### <RevokeLink>

There is only one element of this type in the message, containing all the data relating to the request to revoke a link and consisting of the following elements:

| | |
|---|---|
| • **<Header>** | data relating to the request sent |
| • **<Token>** | search only link associated with a given token |

### <Header>

There is only one element of this type in the message, which includes the date relating to the request sent. For the related description see paragraph 3.3.2 in the chapter  "Response messages in XML".

# 4 SIA VPOS Appendices

## 4.1 References

Here below is a list of useful sources to which reference can be made for merchant system integration purposes.

SIA S.p.A. does not provide any type of warranty or support for the third party products set out below.

To obtain the hash MD5 or hash SHA-1 constituting the MAC, the SIA server makes use of the object Java MessageDigest of JDK Oracle. To calculate the HMAC-256, it makes use of the javax.crypto.Mac class with the HmacSHA256 algorithm, also provided by JDK.

A form for calculating hash MD5 in PERL can be obtained from the following URL:

http://www.perl.com/CPAN-local/modules/by-module/MD5/

A number of (business) forms for calculating the hash MD5 in visual basic can be obtained from the following addresses:

http://www.aspencrypt.com/index.html
http://www.hotscripts.com/ASP/Scripts_and_Components/Security_Systems/
http://www.anei.com/aneimd5.asp
http://www.aspin.com/func/search?qry=md5&cat=all&IMAGE1.x=25&IMAGE1.y=7

Function **md5** for calculating the hash on a string is available in the standard libraries of PHP3.

For a definition of the standard MD5 see:

http://www.columbia.edu/~ariel/ssleay/rfc1321.html

For a definition of the standard SHA-1 see:

http://csrc.nist.gov/cryptval/shs.html

For a definition of the HMAC-256 standard and examples of implementation in various languages, see:

https://en.wikipedia.org/wiki/Hash-based_message_authentication_code

https://www.supermind.org/blog/1102/generating-hmac-md5-sha1-sha256-etc-in-java

https://www.jokecamp.com/blog/examples-of-creating-base64-hashes-using-hmac-sha256-in-different-languages

# 4.2 Appendix D Generating MACs for API SIA VPOS

## 4.2.1 Generating the MAC  for the REFUND messages

The MAC to be sent as attachment to the REFUND messages can be obtained through the procedure described herein.

The operator may choose as it sees fit from among three standard algorythms: SHA-1 (also called SHA), MD5 and HMAC-256 (recommended). Given that the three algorythms produce a varying number of bits (160 the first, 128 the second, 256 the third) the system is capable of automatically recognising the type of function used for generating the MAC. The store site may vary the algorythm used at its will.

The merchant and SIA share a secret string of 50 or 100 characters. To produce the MAC of the data, a hash of the connection between text to be signed and secret string is performed (SHA-1 and MD5). Otherwise, an HMAC-256 is calculated using the secret string as key for the text to be signed.

**For the REFUND messages the text to be signed must contain the following fields:**

- OPERATION
- TIMESTAMP
- SHOPID
- OPERATORID
- REQREFNUM
- TRANSACTIONID
- ORDERID
- AMOUNT
- CURRENCY
- EXPONENT (if present)
- OPDESCR (if present)
- OPTIONS (if present)

**The MAC will be as follows:**

**MAC=Hash(OPERATION=REFUND&TIMESTAMP=<timestamp>&SHOPID=<shopid>&OPERATORID=<OperatorID>&REQREFNUM=<requestnumber>&TRANSACTIONID=<transactionID>&ORDERID=<orderId>&AMOUNT=<Amount>&CURRENCY=<Currency>&OPDESCR=<OpDescr>&<secret string>)**

NOTE: in case of signature with HMAC-256 the secret key must not be queued to the string, but rather, it must be used as a calculation key.

The wording in < > indicates the field values. The order in which the fields appear is clearly fundamental. The MAC is not case sensitive. Capital and lowercase letters can be used without distinction.

An example of such a string could be the following:

OPERATION=REFUND&TIMESTAMP=2015-04-08T13:04:21.852&SHOPID=12345678901&OPERATORID=KR839H&REQREFNUM=20150501496204690934584305834564&TRANSACTIONID=HK84HL2G&ORDERID=A4845b2&AMOUNT=100&CURRENCY=978&Absd830923fk32..

The MAC, which is the result of a hash, must be coded appropriately for it to be trasmitted in HTTP. To that end, an exadecimal conversion must be used. The result of said conversion is a 32-character string, if the hash function used is MD5. If, on the other hand, SHA-1 is used, the result will be a 40-character string. If HMAC-256 is used, the result will be a 64-character string.

## 4.2.2 Generating the MAC for the ACCOUNTING messages

The MAC to be sent as attachment to the ACCOUNTING messages can be obtained through the procedure described herein.

The operator may choose as it sees fit from among three standard algorythms: SHA-1 (also called SHA), MD5 and HMAC-256 (recommended). Given that the three algorythms produce a varying number of bits (160 the first, 128 the second, 256 the third) the system is capable of automatically recognising the type of function used for generating the MAC. The store site may vary the algorythm used at its will.

The merchant and SIA share a secret string of 50 or 100 characters. To produce the MAC of the data, a hash of the connection between text to be signed and secret string is performed (SHA-1 and MD5). Otherwise, an HMAC-256 is calculated using the secret string as key for the text to be signed.

**For the ACCOUNTING messages, the text to be signed must contain the following fields:**

- OPERATION
- TIMESTAMP
- SHOPID
- OPERATORID
- REQREFNUM
- TRANSACTIONID
- ORDERID
- AMOUNT
- CURRENCY
- EXPONENT (if present)
- OPDESCR (if present)
- OPTIONS (if present)

**The MAC will be as follows:**

**MAC=Hash(OPERATION=ACCOUNTING&TIMESTAMP=<timestamp>&SHOPID=<shopid>&OPERATORID=<OperatorID>&REQREFNUM=<requestnumber>&TRANSACTIONID=<transactionID>&ORDERID=<orderId>&AMOUNT=<Amount>&CURRENCY=<Currency>&OPDESCR=<OpDescr>&<secret string> )**

NOTE: in case of signature with HMAC-256 the secret key must not be queued to the string, but rather, it must be used as a calculation key.

The wording in < > indicates the field values. The order in which the fields appear is clearly fundamental. The MAC is not case sensitive. Capital and lowercase letters can be used without distinction.

An example of such a string could be the following:

OPERATION=ACCOUNTING&TIMESTAMP=2015-04-08T13:04:21.852&SHOPID=123456789012345&
OPERATORID=KR839H&REQREFNUM=20150501496204690934584305834564&TRANSACTIONID=HK84HL2
G&ORDERID=A4845b2&AMOUNT=100&CURRENCY=978&Absd830923fk32..

The MAC, which is the result of a hash, must be coded appropriately for it to be trasmitted in HTTP. To that end, an exadecimal conversion must be used. The result of said conversion is a 32-character string, if the hash function used is MD5. If, on the other hand, SHA-1 is used, the result will be a 40-character string. If HMAC-256 is used, the result will be a 64-character string.

### 4.2.3 Generating the MAC for the REVERSEACCOUNTING messages

The MAC to be sent as attachment to the REVERSEACCOUNTING messages can be obtained through the procedure described herein.

The operator may choose as it sees fit from among three standard algorythms: SHA-1 (also called SHA), MD5 and HMAC-256 (recommended). Given that the three algorythms produce a varying number of bits (160 the first, 128 the second, 256 the third) the system is capable of automatically recognising the type of function used for generating the MAC. The store site may vary the algorythm used at its will.

The merchant and SIA share a secret string of 50 or 100 characters. To produce the MAC of the data, a hash of the connection between text to be signed and secret string is performed (SHA-1 and MD5). Otherwise, an HMAC-256 is calculated using the secret string as key for the text to be signed.

**For the REVERSEACCOUNTING messages, the text to be signed must contain the following fields:**

- OPERATION
- TIMESTAMP
- SHOPID
- OPERATORID
- REQREFNUM
- TRANSACTIONID
- ORDERID
- OPTIONS (if present)

**The MAC will be as follows:**

**MAC=Hash(OPERATION=REVERSEACCOUNTING&TIMESTAMP=<timestamp>&SHOPID=<shopid>&OPERATORID=<OperatorID>&REQREFNUM=<requestnumber>&TRANSACTIONID=<transactionID>&ORDERID=<orderId>&<secret string> )**

NOTE: in case of signature with HMAC-256 the secret key must not be queued to the string, but rather, it must be used as a calculation key.

The wording in < > indicates the field values. The order in which the fields appear is clearly fundamental. The MAC is not case sensitive. Capital and lowercase letters can be used without distinction.

An example of such a string could be the following:

OPERATION=REVERSEACCOUNTING&TIMESTAMP=2015-04-8T13:04:21.852
&SHOPID=123456789012345&OPERATORID=KR839H&REQREFNUM=20150501496204690934584305834564&
TRANSACTIONID=HK84HL2G&ORDERID=A4845b2&Absd830923fk32..

The MAC, which is the result of a hash, must be coded appropriately for it to be trasmitted in HTTP. To that end, an exadecimal conversion must be used. The result of said conversion is a 32-character string, if the hash function used is MD5. If, on the other hand, SHA-1 is used, the result will be a 40-character string. If HMAC-256 is used, the result will be a 64-character string.

## 4.2.4 Generating the MAC  for the LISTOPERATION messages

The MAC to be sent as attachment to the LISTOPERATION messages can be obtained through the procedure described herein.

The operator may choose as it sees fit from among three standard algorythms: SHA-1 (also called SHA), MD5 and HMAC-256 (recommended). Given that the three algorythms produce a varying number of bits (160 the first, 128 the second, 256 the third) the system is capable of automatically recognising the type of function used for generating the MAC. The store site may vary the algorythm used at its will.

The merchant and SIA share a secret string of 50 or 100 characters. To produce the MAC of the data, a hash of the connection between text to be signed and secret string is performed (SHA-1 and MD5). Otherwise, an HMAC-256 is calculated using the secret string as key for the text to be signed.

**For the LISTOPERATION messages, the text to be signed must contain the following fields:**

- OPERATION
- TIMESTAMP
- SHOPID
- OPERATORID
- REQREFNUM
- STARTDATE
- ENDDATE
- OPDESCR (if present)
- OPTIONS (if present)

**The MAC will be as follows:**

**MAC=Hash(OPERATION=LISTOPERATION&TIMESTAMP=<timestamp>&SHOPID=<shopid>&OPERATORID=<OperatorID>&REQREFNUM=<requestnumber>&STARTDATE=<startdate>&ENDDATE=<enddate>&<secret string> )**

NOTE: in case of signature with HMAC-256 the secret key must not be queued to the string, but rather, it must be used as a calculation key.

The wording in  < > indicates the field values. The order in which the fields appear is clearly fundamental. The MAC is not case sensitive. Capital and lowercase letters can be used without distinction.

An example of such a string could be the following:

OPERATION=LISTOPERATION&TIMESTAMP=2015-04-08T13:04:21.852&SHOPID=123456789012345&OPERATORID=KR839H&REQREFNUM=20150501496204690934584305834564&STARTDATE=2015-04-04&ENDDATE=2015-04-04&Absd830923fk32..

The MAC, which is the result of a hash, must be coded appropriately for it to be trasmitted in HTTP. To that end, an exadecimal conversion must be used. The result of said conversion is a 32-character string, if the hash function used is MD5. If, on the other hand, SHA-1 is used, the result will be a 40-character string. If HMAC-256 is used, the result will be a 64-character string.

## 4.2.5 Generating the MAC for the LISTAUTHORIZATION messages

The MAC to be sent as attachment to the LISTAUTHORIZATION messages can be obtained through the procedure described herein.

The operator may choose as it sees fit from among three standard algorythms: SHA-1 (also called SHA), MD5 and HMAC-256 (recommended). Given that the three algorythms produce a varying number of bits (160 the first, 128 the second, 256 the third) the system is capable of automatically recognising the type of function used for generating the MAC. The store site may vary the algorythm used at its will.

The merchant and SIA share a secret string of 50 or 100 characters. To produce the MAC of the data, a hash of the connection between text to be signed and secret string is performed (SHA-1 and MD5). Otherwise, an HMAC-256 is calculated using the secret string as key for the text to be signed.

**For the LISTAUTHORIZATION messages, the text to be signed must contain the following fields:**

- OPERATION
- TIMESTAMP
- SHOPID
- OPERATORID
- REQREFNUM
- STARTDATE (if present)
- ENDDATE (if present)
- FILTER
- TRANSACTIONID (if present)
- STARTTIME (if present)
- ENDTIME (if present)
- OPTIONS (if present)

**The MAC will be as follows:**

**MAC=Hash(OPERATION=LISTAUTHORIZATION&TIMESTAMP=<timestamp>&SHOPID=<shopid>&OPERATORID=<OperatorID>&REQREFNUM=<requestnumber>&STARTDATE=<startdate>&ENDDATE=<enddate>&FILTER=<filter>&TRANSACTIONID=<transactionId>&STARTTIME=<starttime>&ENDTIME=<endTime>&<secret string> )**

NOTE: in case of signature with HMAC-256 the secret key must not be queued to the string, but rather, it must be used as a calculation key.

The wording in < > indicates the field values. The order in which the fields appear is clearly fundamental. The MAC is not case sensitive. Capital and lowercase letters can be used without distinction.

If the search is not performed on the basis of the TRANSACTIONID, said field shall in any case be taken into account in calculating the MAC as: TRANSACTIONID=. If the search is performed by date, but not by hour, the fields STARTTIME and ENDTIME must not be taken into account in calculating the MAC.

An example of such a string could be the following:

OPERATION=LISTAUTHORIZATION&TIMESTAMP=2015-04-08T13:04:21.852&SHOPID=123456789012345&OPERATORID=KR839H&REQREFNUM=20150501496204690934584305834564&STARTDATE=2015-04-04&ENDDATE=2015-04-04&FILTER=1&TRANSACTIONID=HK84HL2G&STARTTIME=10.00&ENDTIME=18.30&Absd830923fk32..

The MAC, which is the result of a hash, must be coded appropriately for it to be trasmitted in HTTP. To that end, an exadecimal conversion must be used. The result of said conversion is a 32-character string, if the hash function used is MD5. If, on the other hand, SHA-1 is used, the result will be a 40-character string. If HMAC-256 is used, the result will be a 64-character string.

# 4.2.6Generating the MAC  for the ORDERSTATUS messages

The MAC to be sent as attachment to the ORDERSTATUS messages can be obtained through the procedure described herein.

The operator may choose as it sees fit from among three standard algorythms: SHA-1 (also called SHA), MD5 and HMAC-256 (recommended). Given that the three algorythms produce a varying number of bits (160 the first, 128 the second, 256 the third) the system is capable of automatically recognising the type of function used for generating the MAC. The store site may vary the algorythm used at its will.

The merchant and SIA share a secret string of 50 or 100 characters. To produce the MAC of the data, a hash of the connection between text to be signed and secret string is performed (SHA-1 and MD5). Otherwise, an HMAC-256 is calculated using the secret string as key for the text to be signed.

**For the ORDERSTATUS messages, the text to be signed must contain the following fields:**

- OPERATION
- TIMESTAMP
- SHOPID
- OPERATORID
- REQREFNUM
- ORDERID
- OPTIONS (if present)
- PRODUCTREF (if present)

**The MAC will be as follows:**

**MAC=Hash(OPERATION=ORDERSTATUS&TIMESTAMP=<timestamp>&SHOPID=<shopid>&OPERATORID=<OperatorID>&REQREFNUM=<requestnumber>&ORDERID=<orderId>&<secret string>)**

NOTE: in case of signature with HMAC-256 the secret key must not be queued to the string, but rather, it must be used as a calculation key.

The wording in  < > indicates the field values. The order in which the fields appear is clearly fundamental. The MAC is not case sensitive. Capital and lowercase letters can be used without distinction.

An example of such a string could be the following:

OPERATION=ORDERSTATUS&TIMESTAMP=2015-04-08T13:04:21.852&SHOPID=123456789012345&
OPERATORID=KR839H&ORDERID=A4845b2&Absd830923fk32..

The MAC, which is the result of a hash, must be coded appropriately for it to be trasmitted in HTTP. To that end, an exadecimal conversion must be used. The result of said conversion is a 32-character string, if the hash function used is MD5. If, on the other hand, SHA-1 is used, the result will be a 40-character string..

## 4.2.7Generating the MAC for the XML <BPWXmlResponse> elements

The MAC attached by SIA VPOS to the XML BPWXmlResponse elements contained in the reply messages forwarded to the merchant system can be obtained through the procedure described herein.

The merchant and SIA share a secret string of 50 or 100 characters. To produce the MAC of the data, a hash of the connection between text to be signed and secret string is performed (SHA-1 and MD5). Otherwise, an HMAC-256 is calculated using the secret string as key for the text to be signed.

The hash function used by the system for generating the MAC is the same as that used by the merchant for generating the MAC of the request message. Given that the algorythms SHA1, MD5 and HMAC256 produce a varying number of bits (160 the first, 128 the second, 256 the third) the system is capable of automatically recognising the type of function used for generating the MAC of the request and, in turn, use the same algorithm for the reply.

In essence, if the MAC of the request message is calculated using MD5, the MAC of the reply will also be calculated using MD5. If the MAC of the request message is calculated using SHA-1, the MAC of the reply will also be calculated using SHA-1. If the request message is in HMAC256, so will be the reply message.

**For the BPWXmlResponse element, the signed text will contain the value of the following subelements:**

- Timestamp
- Result

**The MAC will be as follows:**

**MAC = Hash(<timestamp>&<Result>&<secret string>)**

NOTE: in case of signature with HMAC-256 the secret key must not be queued to the string, but rather, it must be used as a calculation key.

The wording in < > indicates the field values. The order in which the fields appear is clearly fundamental. The MAC is not case sensitive. Capital and lowercase letters can be used without distinction.

An example of such a string could be the following:

2015-07-04T12:02:55&00&Absd830923fk32h7de23r..

The MAC, which is the result of a hash, must be coded appropriately for it to be trasmitted in HTTP. To that end, an exadecimal conversion must be used. The result of said conversion is a 32-character string, if the hash function used is MD5. If, on the other hand, SHA-1 is used, the result will be a 40-character string. If HMAC-256 is used, the result will be a 64-character string.

The MAC is not case sensitive. The SIA server uses capital letters.

NOTE: The names of the XML elements are not used for calculating the MAC. Only the values are used.

NOTE: If the outcome of the request is an authentication error the MAC will not be calculated and it will have the value of “NULL”.

## 4.2.8Generating the MAC  for the XML <Operation> elements

The MAC to be sent as attachment to the XML <Operation> elements can be obtained through the procedure described herein.

The merchant and SIA share a secret string of 50 or 100 characters. To produce the MAC of the data, a hash of the connection between text to be signed and secret string is performed (SHA-1 and MD5). Otherwise, an HMAC-256 is calculated using the secret string as key for the text to be signed.

The hash function used by the system for generating the MAC is the same as that used by the merchant for generating the MAC of the request message. Given that the algorythms SHA1, MD5 and HMAC256 produce a varying number of bits (160 the first, 128 the second, 256 the third) the system is capable of automatically recognising the type of function used for generating the MAC of the request and, in turn, use the same algorithm for the reply.

In essence, if the MAC of the request message is calculated using MD5, the MAC of the reply will also be calculated using MD5. If the MAC of the request message is calculated using SHA-1, the MAC of the reply will also be calculated using SHA-1. If the request message is in HMAC256, so will be the reply message.

**For the XML <Operation> elements, the signed text will contain the values of the following subelements:**

- TransactionID
- TimestampReq
- TimestampElab
- SrcType
- Amount
- Result
- Status
- OpDescr (if present)

**The MAC will be as follows:**

**MAC=Hash(<TransactionID>&<TimestampReq>&<TimestampElab>&<SrcType>&<Amount> &<Result>&<Status>&<secret string> )**

NOTE: in case of signature with HMAC-256 the secret key must not be queued to the string, but rather, it must be used as a calculation key.

The wording in  < > indicates the field values. The order in which the fields appear is clearly fundamental. The MAC is not case sensitive. Capital and lowercase letters can be used without distinction.

An example of such a string could be the following:

CC8424&2015-07-04T12:02:54&2015-07-07T12:03:02&CTO05&100&00&SGN03&Absd830923fk32..

The MAC, which is the result of a hash, must be coded appropriately for it to be trasmitted in HTTP. To that end, an exadecimal conversion must be used. The result of said conversion is a 32-character string, if the hash function used is MD5. If, on the other hand, SHA-1 is used, the result will be a 40-character string. If HMAC-256 is used, the result will be a 64-character string.

NOTE: The names of the XML elements are not used for calculating the MAC. Only the values are used.

## 4.2.9 Generating a MAC for the XML <Authorization> elements

The MAC sent by SIA VPOS as attachment to the XML <Authorization> elements can be obtained through the procedure described herein.

The merchant and SIA share a secret string of 50 or 100 characters. To produce the MAC of the data, a hash of the connection between text to be signed and secret string is performed (SHA-1 and MD5). Otherwise, an HMAC-256 is calculated using the secret string as key for the text to be signed.

The hash function used by the system for generating the MAC is the same as that used by the merchant for generating the MAC of the request message. Given that the algorythms SHA1, MD5 and HMAC256 produce a varying number of bits (160 the first, 128 the second, 256 the third) the system is capable of automatically recognising the type of function used for generating the MAC of the request and, in turn, use the same algorithm for the reply.

In essence, if the MAC of the request message is calculated using MD5, the MAC of the reply will also be calculated using MD5. If the MAC of the request message is calculated using SHA-1, the MAC of the reply will also be calculated using SHA-1. If the request message is in HMAC256, so will be the reply message.

**For the XML elements <Authorization>, the signed text will contain the values of the following subelements:**

- AuthorizationType
- TransactionID
- Network
- OrderId
- TransactionAmount
- AuthorizedAmount
- Currency
- AccountedAmount
- RefundedAmount (*only if the parameter **RELEASE=02** is specified in the request)*
- TransactionResult
- Timestamp
- AuthorizationNumber
- AcquirerBIN
- MerchantID
- Status
- ResponseCodeISO (*if present*)
- PanTail (*if present*)
- PanExpiryDate (*if present*)
- PaymentTypePP (*if present*)
- RRN (*if present*)
- IbanCode (*only for iban transactions)*
- CardType (*if present*)
- CardholderInfo (*if present)*

**The MAC will be as follows:**

**MAC=Hash(<authorizationType>&<transactionID>&<Network>&<orderId>&<transactionAmount>&<authorizedAmount>&<currency>&<accountedAmount>&<refundedAmount>&<transactionResult>&<timestamp>&<authorizationNumber>&<aquirerBIN>&<merchantID>&<status>& <responsecodeiso>&<cardholderInfo>&<secret string> )**

NOTE: in case of signature with HMAC-256 the secret key must not be queued to the string, but rather, it must be used as a calculation key.

The wording in < > indicates the field values. The order in which the fields appear is clearly fundamental. The MAC is not case sensitive. Capital and lowercase letters can be used without distinction.

An example of such a string could be the following:

I&8032180310wieeuejjwerrrrr&01&order1&10000&10000&978&10000&0&00&2015-07-06T13:04:34&123456&234569&05423956754389&02&Absd830923fk32..

The MAC, which is the result of a hash, must be coded appropriately for it to be trasmitted in HTTP. To that end, an exadecimal conversion must be used by the SIA server.
The result of said conversion is a 32-character string, if the hash function used is MD5. If, on the other hand, SHA-1 is used, the result will be a 40-character string. If HMAC-256 is used, the result will be a 64-character string.

NOTE: The names of the XML elements are not used for calculating the MAC. Only the values are used.

### 4.2.10 Generating a MAC for the AUTHORIZATION message

The MAC to be sent as attachment to the AUTHORIZATION messages can be obtained through the procedure described herein.

The operator may choose as it sees fit from among three standard algorythms: SHA-1 (also called SHA), MD5 and HMAC-256 (recommended). Given that the three algorythms produce a varying number of bits (160 the first, 128 the second, 256 the third) the system is capable of automatically recognising the type of function used for generating the MAC. The store site may vary the algorythm used at its will.

The merchant and SIA share a secret string of 50 or 100 characters. To produce the MAC of the data, a hash of the connection between text to be signed and secret string is performed (SHA-1 and MD5). Otherwise, an HMAC-256 is calculated using the secret string as key for the text to be signed.

**For the AUTHORIZATION messages, the text to be signed must contain the following fields:**

- OPERATION
- TIMESTAMP
- SHOPID
- ORDERID
- OPERATORID
- REQREFNUM
- PAN
- CVV2 (if present)
- EXPDATE
- AMOUNT
- CURRENCY
- EXPONENT (if present)
- ACCOUNTINGMODE
- NETWORK
- EMAILCH (if present)
- USERID (if present)
- ACQUIRER (if present)
- IPADDRESS (if present)
- OPDESCR (if present)
- USRAUTHFLAG (if present)
- OPTIONS (if present)
- ANTIFRAUD (if present)
- PRODUCTREF (if present)
- NAME (if present)
- SURNAME (if present)
- TAXID (if present)
- TRECURR (if present)
- CRECURR (if present)
- INSTALLMENTSNUMBER (if present)

**The MAC will be as follows:**

**MAC=Hash(OPERATION=AUTHORIZATION&TIMESTAMP=<timestamp>&SHOPID=<shopid>&ORDERID=<OrderID>&OPERATORID=<OperatorID>&REQREFNUM=<requestnumber>&PAN=<pan>&CVV2=<cvv2>&EXPDATE=<expirydate>&AMOUNT=<Amount>&CURRENCY=<Currency>&ACCOUNTINGMODE=<accountingmode>&NETWORK=<network>&EMAILCH=<cardholderemail>&USERID=<userid>&ACQUIRER=<acquirer>&IPADDRESS=<ipaddress>&OPDESCR=<OpDescr>&USRAUTHFLAG=<UsrAuthFlag>&OPTIONS=<Options>&ANTIFRAUD=<AntiFraud>&PRODUCTREF=<ProductRef>&NAME=<Name>&SURNAME=<Surname>&TAXID=<TaxID>&TRECURR=<TRecurr>&CRECURR=<CRecurr>&INSTALLMENTNUMBER=<InstallmentsNumber>&<secret string>)**

NOTE: in case of signature with HMAC-256 the secret key must not be queued to the string, but rather, it must be used as a calculation key.

The wording in < > indicates the field values. The order in which the fields appear is clearly fundamental. The MAC is not case sensitive. Capital and lowercase letters can be used without distinction.

An example of such a string could be the following:

OPERATION=AUTHORIZATION&TIMESTAMP=2015-04-08T13:04:21.852&SHOPID=123456789012345&ORDERID=ord1&OPERATORID=KR839H&REQREFNUM=20150501496204690934584305834564&PAN=1234567812345678&EXPDATE=0317&AMOUNT=100&CURRENCY=978&ACCOUNTINGMODE=I&NETWORK=01&Absd830923fk32..

The MAC, which is the result of a hash, must be coded appropriately for it to be trasmitted in HTTP. To that end, an exadecimal conversion must be used. The result of said conversion is a 32-character string, if the hash function used is MD5. If, on the other hand, SHA-1 is used, the result will be a 40-character string. If HMAC-256 is used, the result will be a 64-character string.

### 4.2.11 Generating a MAC for the AUTHORIZATION3DSSTEP1 messages

The MAC to be sent as attachment to the AUTHORIZATION3DSSTEP1 messages can be obtained through the procedure described herein.

The operator may choose as it sees fit from among three standard algorythms: SHA-1 (also called SHA), MD5 and HMAC-256 (recommended). Given that the three algorythms produce a varying number of bits (160 the first, 128 the second, 256 the third) the system is capable of automatically recognising the type of function used for generating the MAC. The store site may vary the algorythm used at its will.

The merchant and SIA share a secret string of 50 or 100 characters. To produce the MAC of the data, a hash of the connection between text to be signed and secret string is performed (SHA-1 and MD5). Otherwise, an HMAC-256 is calculated using the secret string as key for the text to be signed.

**For the AUTHORIZATION3DSSTEP1 messages, the text to be signed must contain the following fields:**

- OPERATION
- TIMESTAMP
- SHOPID
- ORDERID
- OPERATORID
- REQREFNUM
- PAN
- CVV2 (if present)
- EXPDATE
- AMOUNT
- CURRENCY
- EXPONENT (if present)
- ACCOUNTINGMODE
- NETWORK
- EMAILCH (if present)
- USERID (if present)
- ACQUIRER (if present)
- IPADDRESS (if present)
- OPDESCR (if present)
- USRAUTHFLAG (if present)
- OPTIONS (if present)
- ANTIFRAUD (if present)
- PRODUCTREF (if present)
- NAME (if present)
- SURNAME (if present)
- TAXID (if present)
- TRECURR (if present)
- CRECURR (if present)
- INPERSON (if present)
- MERCHANTURL (if present)
- SERVICE (if present)
- XID (if present)
- CAVV (if present)
- ECI (if present)

- PP_AUTHENTICATEMETHOD (if present)
- PP_CARDENROLLMETHOD (if present)
- PARESSTATUS (if present)
- SCENROLLSTATUS (if present)
- SIGNATUREVERIFICATION (if present)
- INSTALLMENTSNUMBER (if present)

**The MAC will be as follows:**

```
MAC=Hash(OPERATION=AUTHORIZATION3DSSTEP1&TIMESTAMP=<timestamp>&SHOPID
=<shopid>&ORDERID=<OrderID>&OPERATORID=<OperatorID>&REQREFNUM=<requestnum
ber>&PAN=<pan>&CVV2=<cvv2>&EXPDATE=<expirydate>&AMOUNT=<Amount>&CURREN
CY=<Currency>&ACCOUNTINGMODE=<accountingmode>&NETWORK=<network>&EMAILC
H=<cardHolderEmail>USERID=<userid>&IPADDRESS=<ipaddress>&OPDESCR=<OpDescr>&U
SRAUTHFLAG=<UsrAuthFlag>&OPTIONS=<Options>&ANTIFRAUD=<AntiFraud>&PRODUCT
REF=<ProductRef>&NAME=<Name>&SURNAME=<Surname>&TAXID=<TaxID>&TRECURR=
<TRecurr>&CRECURR=<CRecurr>&INSTALLMENTNUMBER=<InstallmentsNumber>&INPERS
ON=<inperson>&MERCHANTURL=<url>&SERVICE=<service>&XID=<xid>&CAVV=<cavv>&
ECI=<>&PP_AUTHENTICATEMETHOD=<PP_AuthenticateMethod>&PP_CARDENROLLMETH
OD=<PP_CardEnrollMethod>&PARESSTATUS=<ParesStatus>&SCENROLLSTATUS=<ScenRoll
Status>&SIGNATUREVERIFICATION=<SignatureVerification>&<secret string>)
```

NOTE: in case of signature with HMAC-256 the secret key must not be queued to the string, but rather, it must be used as a calculation key.

The wording in < > indicates the field values. The order in which the fields appear is clearly fundamental. The MAC is not case sensitive. Capital and lowercase letters can be used without distinction.
Any optional fields that have not been filled in must be completely removed from the string determining the MAC.

The MAC, which is the result of a hash, must be coded appropriately for it to be trasmitted in HTTP. To that end, an exadecimal conversion must be used. The result of said conversion is a 32-character string, if the hash function used is MD5. If, on the other hand, SHA-1 is used, the result will be a 40-character string. If HMAC-256 is used, the result will be a 64-character string.

### 4.2.12 Generating a MAC for the AUTHORIZATION3DSSTEP2 message

The MAC to be sent as attachment to the AUTHORIZATION3DSSTEP2 messages can be obtained through the procedure described herein.

The operator may choose as it sees fit from among three standard algorythms: SHA-1 (also called SHA), MD5 and HMAC-256 (recommended). Given that the three algorythms produce a varying number of bits (160 the first, 128 the second, 256 the third) the system is capable of automatically recognising the type of function used for generating the MAC. The store site may vary the algorythm used at its will.

The merchant and SIA share a secret string of 50 or 100 characters. To produce the MAC of the data, a hash of the connection between text to be signed and secret string is performed (SHA-1 and MD5). Otherwise, an HMAC-256 is calculated using the secret string as key for the text to be signed.

**For the AUTHORIZATION3DSSTEP2 messages, the text to be signed must contain the following fields:**

- OPERATION
- TIMESTAMP
- SHOPID
- OPERATORID
- REQREFNUM
- ORIGINALREQREFNUM
- PARES
- ACQUIRER (if present)

**The MAC will be as follows:**

**MAC=Hash(OPERATION=AUTHORIZATION3DSSTEP2&TIMESTAMP=<timestamp>&SHOPID=<shopid>&OPERATORID=<OperatorID>&REQREFNUM=<requestnumber>& ORIGINALREQREFNUM=<originalreqrefnum>&PARES=<pares>&<secret string > )**

NOTE: in case of signature with HMAC-256 the secret key must not be queued to the string, but rather, it must be used as a calculation key.

The wording in < > indicates the field values. The order in which the fields appear is clearly fundamental. The MAC is not case sensitive. Capital and lowercase letters can be used without distinction.
Any optional fields that have not been filled in must be completely removed from the string determining the MAC.

The MAC, which is the result of a hash, must be coded appropriately for it to be trasmitted in HTTP. To that end, an exadecimal conversion must be used. The result of said conversion is a 32-character string, if the hash function used is MD5. If, on the other hand, SHA-1 is used, the result will be a 40-character string. If HMAC-256 is used, the result will be a 64-character string.

### 4.2.13 Generating a MAC for the THREEDSAUTHORIZATION0 messages

The MAC to be sent as attachment to the THREEDSAUTHORIZATION0 messages can be obtained through the procedure described herein.

The operator may choose as it sees fit from among three standard algorythms: SHA-1 (also called SHA), MD5 and HMAC-256 (recommended). Given that the three algorythms produce a varying number of bits (160 the first, 128 the second, 256 the third) the system is capable of automatically recognising the type of function used for generating the MAC. The store site may vary the algorythm used at its will.

The merchant and SIA share a secret string of 50 or 100 characters. To produce the MAC of the data, a hash of the connection between text to be signed and secret string is performed (SHA-1 and MD5). Otherwise, an HMAC-256 is calculated using the secret string as key for the text to be signed.

**For the THREEDSAUTHORIZATION0 messages, the text to be signed must contain the following fields:**

- OPERATION
- TIMESTAMP
- SHOPID
- ORDERID
- OPERATORID
- REQREFNUM
- PAN
- CVV2 (if present)
- EXPDATE
- AMOUNT
- CURRENCY
- EXPONENT (if present)
- ACCOUNTINGMODE
- NETWORK
- EMAILCH (if present)
- USERID (if present)
- ACQUIRER (if present)
- IPADDRESS (if present)
- USRAUTHFLAG (if present)
- OPDESCR (if present)
- OPTIONS (if present)
- ANTIFRAUD (if present)
- PRODUCTREF (if present)
- NAME (if present)
- SURNAME (if present)
- TAXID (if present)
- THREEDSDATA
- NAMECH
- NOTIFURL (if present)
- THREEDSMTDNOTIFURL (if present)
- CHALLENGEWINSIZE (if present)
- TRECURR (if present)
- CRECURR (if present)
- INSTALLMENTSNUMBER (if present)

**The MAC will be as follows:**

```
MAC=Hash(OPERATION=THREEDSAUTHORIZATION0&TIMESTAMP=<timestamp>&SHOPI
D=<shopid>&ORDERID=<OrderID>&OPERATORID=<OperatorID>&
REQREFNUM=<requestnumber>&PAN=<pan>&CVV2=<cvv2>&EXPDATE=<expiry
date>&AMOUNT=<Amount>&CURRENCY=<Currency>&ACCOUNTINGMODE=<accounting
mode>&NETWORK=<network>&EMAILCH=<card holder email>USERID=<userid>&
IPADDRESS=<ipaddress>&OPDESCR=<OpDescr>&THREEDSDATA=<threedsdata>&NOTIFUR
L=<notifurl>&THREEDSMTDNOTIFURL=<threedsmtdnotifurl>&CHALLENGEWINSIZE=<chall
engewinsize>&TRECURR=<TRecurr>&CRECURR=<CRecurr>&INSTALLMENTSNUMBER=<in
stallmentsnumber>&<secret string>)
```

NOTE: in case of signature with HMAC-256 the secret key must not be queued to the string, but rather, it must be used as a calculation key.

The wording in < > indicates the field values. The order in which the fields appear is clearly fundamental. The MAC is not case sensitive. Capital and lowercase letters can be used without distinction.
Any optional fields that have not been filled in must be completely removed from the string determining the MAC.

The MAC, which is the result of a hash, must be coded appropriately for it to be trasmitted in HTTP. To that end, an exadecimal conversion must be used. The result of said conversion is a 32-character string, if the hash function used is MD5. If, on the other hand, SHA-1 is used, the result will be a 40-character string. If HMAC-256 is used, the result will be a 64-character string.

### 4.2.14 Generating a MAC for the THREEDSAUTHORIZATION1 messages

The MAC to be sent as attachment to the THREEDSAUTHORIZATION1 messages can be obtained through the procedure described herein.

The operator may choose as it sees fit from among three standard algorythms: SHA-1 (also called SHA), MD5 and HMAC-256 (recommended). Given that the three algorythms produce a varying number of bits (160 the first, 128 the second, 256 the third) the system is capable of automatically recognising the type of function used for generating the MAC. The store site may vary the algorythm used at its will.

The merchant and SIA share a secret string of 50 or 100 characters. To produce the MAC of the data, a hash of the connection between text to be signed and secret string is performed (SHA-1 and MD5). Otherwise, an HMAC-256 is calculated using the secret string as key for the text to be signed.

**For the THREEDSAUTHORIZATION1 messages, the text to be signed must contain the following fields:**

- OPERATION
- TIMESTAMP
- SHOPID
- OPERATORID
- REQREFNUM
- THREEDSTRANSID
- THREEDSMTDCOMPLIND

**The MAC will be as follows:**

**MAC=Hash(OPERATION=THREEDSAUTHORIZATION1&TIMESTAMP=<timestamp>&SHOPID=<shopid>&OPERATORID=<OperatorID>& REQREFNUM=<requestnumber>&THREEDSTRANSID=<threedstransid>&THREEDSMTDCOMPLIND=<threedsmtdcomplind>&<secret string>)**

NOTE: in case of signature with HMAC-256 the secret key must not be queued to the string, but rather, it must be used as a calculation key.

The wording in < > indicates the field values. The order in which the fields appear is clearly fundamental. The MAC is not case sensitive. Capital and lowercase letters can be used without distinction.
Any optional fields that have not been filled in must be completely removed from the string determining the MAC.

The MAC, which is the result of a hash, must be coded appropriately for it to be trasmitted in HTTP. To that end, an exadecimal conversion must be used. The result of said conversion is a 32-character string, if the hash function used is MD5. If, on the other hand, SHA-1 is used, the result will be a 40-character string. If HMAC-256 is used, the result will be a 64-character string.

## 4.2.15 Generating a MAC for the THREEDSAUTHORIZATION2 messages

The MAC to be sent as attachment to the THREEDSAUTHORIZATION2 messages can be obtained through the procedure described herein.

The operator may choose as it sees fit from among three standard algorythms: SHA-1 (also called SHA), MD5 and HMAC-256 (recommended). Given that the three algorythms produce a varying number of bits (160 the first, 128 the second, 256 the third) the system is capable of automatically recognising the type of function used for generating the MAC. The store site may vary the algorythm used at its will.

The merchant and SIA share a secret string of 50 or 100 characters. To produce the MAC of the data, a hash of the connection between text to be signed and secret string is performed (SHA-1 and MD5). Otherwise, an HMAC-256 is calculated using the secret string as key for the text to be signed.

**For the THREEDSAUTHORIZATION2 messages, the text to be signed must contain the following fields:**

- OPERATION
- TIMESTAMP
- SHOPID
- OPERATORID
- REQREFNUM
- THREEDSTRANSID

**The MAC will be as follows:**

**MAC=Hash(OPERATION=3DSAUTHORIZATION2&TIMESTAMP=<timestamp>&SHOPID =<shopid>&OPERATORID=<OperatorID>& REQREFNUM=<requestnumber>&THREEDSTRANSID=<threedstransid>&<secret string>)**

NOTE: in case of signature with HMAC-256 the secret key must not be queued to the string, but rather, it must be used as a calculation key.

The wording in < > indicates the field values. The order in which the fields appear is clearly fundamental. The MAC is not case sensitive. Capital and lowercase letters can be used without distinction.
Any optional fields that have not been filled in must be completely removed from the string determining the MAC.

The MAC, which is the result of a hash, must be coded appropriately for it to be trasmitted in HTTP. To that end, an exadecimal conversion must be used. The result of said conversion is a 32-character string, if the hash function used is MD5. If, on the other hand, SHA-1 is used, the result will be a 40-character string. If HMAC-256 is used, the result will be a 64-character string.

## 4.2.16 Generating a MAC for the THREEDSVERSIONING messages

The MAC to be sent as attachment to the THREEDSVERSIONING messages can be obtained through the procedure described herein.

The operator may choose as it sees fit from among three standard algorythms: SHA-1 (also called SHA), MD5 and HMAC-256 (recommended). Given that the three algorythms produce a varying number of bits (160 the first, 128 the second, 256 the third) the system is capable of automatically recognising the type of function used for generating the MAC. The store site may vary the algorythm used at its will.

The merchant and SIA share a secret string of 50 or 100 characters. To produce the MAC of the data, a hash of the connection between text to be signed and secret string is performed (SHA-1 and MD5). Otherwise, an HMAC-256 is calculated using the secret string as key for the text to be signed.

**For the THREEDSVERSIONING messages, the text to be signed must contain the following fields:**

- OPERATION
- TIMESTAMP
- SHOPID
- OPERATORID
- REQREFNUM
- NETWORK
- PAN

**The MAC will be as follows:**

**MAC=Hash(OPERATION=THREEDSVERSIONING&TIMESTAMP=<timestamp>&SHOPID=<shopid>&OPERATORID=<OperatorID>&REQREFNUM=<requestnumber>&NETWORK=<network>&PAN=<pan>&<secret string>)**

NOTE: in case of signature with HMAC-256 the secret key must not be queued to the string, but rather, it must be used as a calculation key.

The wording in < > indicates the field values. The order in which the fields appear is clearly fundamental. The MAC is not case sensitive. Capital and lowercase letters can be used without distinction.
Any optional fields that have not been filled in must be completely removed from the string determining the MAC.

The MAC, which is the result of a hash, must be coded appropriately for it to be trasmitted in HTTP. To that end, an exadecimal conversion must be used. The result of said conversion is a 32-character string, if the hash function used is MD5. If, on the other hand, SHA-1 is used, the result will be a 40-character string. If HMAC-256 is used, the result will be a 64-character string.

### 4.2.17 Generating a MAC for the XML <VBVRedirect> element

The MAC attached by SIA VPOS to the XML <VBVRedirect> elements forwarded to the merchant system can be obtained through the procedure described herein.

The merchant and SIA share a secret string of 50 or 100 characters. To produce the MAC of the data, a hash of the connection between text to be signed and secret string is performed (SHA-1 and MD5). Otherwise, an HMAC-256 is calculated using the secret string as key for the text to be signed.

The hash function used by the system for generating the MAC is the same as that used by the merchant for generating the MAC of the request message. Given that the algorythms SHA1, MD5 and HMAC256 produce a varying number of bits (160 the first, 128 the second, 256 the third) the system is capable of automatically recognising the type of function used for generating the MAC of the request and, in turn, use the same algorithm for the reply.

In essence, if the MAC of the request message is calculated using MD5, the MAC of the reply will also be calculated using MD5. If the MAC of the request message is calculated using SHA-1, the MAC of the reply will also be calculated using SHA-1. If the request message is in HMAC256, so will be the reply message.

**For the XML elements <VBVRedirect>, the signed text will contain the values of the following subelements:**

- PaReq
- URLAcs

**The MAC will be as follows:**

**MAC=Hash(<PaReq>&<URLAcs>&<secret string>)**

NOTE: in case of signature with HMAC-256 the secret key must not be queued to the string, but rather, it must be used as a calculation key.

The wording in < > indicates the field values. The order in which the fields appear is clearly fundamental. The MAC is not case sensitive. Capital and lowercase letters can be used without distinction.
Any optional fields that have not been filled in must be completely removed from the string determining the MAC.

The MAC, which is the result of a hash, must be coded appropriately for it to be trasmitted in HTTP. To that end, an exadecimal conversion must be used. The result of said conversion is a 32-character string, if the hash function used is MD5. If, on the other hand, SHA-1 is used, the result will be a 40-character string. If HMAC-256 is used, the result will be a 64-character string.

NOTE: The names of the XML elements are not used for calculating the MAC. Only the values are used.

NOTE: If the outcome of the request is an authentication error, the MAC will not be calculated and its value will be “NULL”.

### 4.2.18 Generating the MAC for the PANALIASRECOVERY messages

The MAC to be sent as attachment to the PANALIASRECOVERY messages can be obtained through the following procedure.

The operator may choose as it sees fit from among three standard algorythms: SHA-1 (also called SHA), MD5 and HMAC-256 (recommended). Given that the three algorythms produce a varying number of bits (160 the first, 128 the second, 256 the third) the system is capable of automatically recognising the type of function used for generating the MAC. The store site may vary the algorythm used at its will.

The merchant and SIA share a secret string of 50 or 100 characters. To produce the MAC of the data, a hash of the connection between text to be signed and secret string is performed (SHA-1 and MD5). Otherwise, an HMAC-256 is calculated using the secret string as key for the text to be signed.

**For the PANALIASRECOVERY messages, the text to be signed must contain the following fields:**

- OPERATION
- TIMESTAMP
- SHOPID
- OPERATORID
- REQREFNUM
- ORDERID
- OPTIONS (if present)

**The MAC will be as follows:**

**MAC=Hash(OPERATION=PANALIASRECOVERY&TIMESTAMP=<timestamp>&SHOPID=<shopid>&OPERATORID=<OperatorID>&REQREFNUM=<requestnumber>&ORDERID=<orderid>&<secret string > )**

NOTE: in case of signature with HMAC-256 the secret key must not be queued to the string, but rather, it must be used as a calculation key.

The wording in < > indicates the field values. The order in which the fields appear is clearly fundamental. The MAC is not case sensitive. Capital and lowercase letters can be used without distinction.
Any optional fields that have not been filled in must be completely removed from the string determining the MAC.

An example of such a string could be the following:

OPERATION=PANALIASRECOVERY&TIMESTAMP=2015-04-08T13:04:21.852&SHOPID=123456789012345&OPERATORID=KR839H&REQREFNUM=20150501496204690934584305834564&ORDERID=ord1&Absd830923fk32..

The MAC, which is the result of a hash, must be coded appropriately for it to be trasmitted in HTTP. To that end, an exadecimal conversion must be used. The result of said conversion is a 32-character string, if the hash function used is MD5. If, on the other hand, SHA-1 is used, the result will be a 40-character string. If HMAC-256 is used, the result will be a 64-character string.

## 4.2.19 Generating a MAC for the XML <PanAliasData> elements

The MAC to be sent as attachment to the XML <PanAliasData> elements can be obtained through the procedure described herein.

The merchant and SIA share a secret string of 50 or 100 characters. To produce the MAC of the data, a hash of the connection between text to be signed and secret string is performed (SHA-1 and MD5). Otherwise, an HMAC-256 is calculated using the secret string as key for the text to be signed.

The hash function used by the system for generating the MAC is the same as that used by the merchant for generating the MAC of the request message. Given that the algorythms SHA1, MD5 and HMAC256 produce a varying number of bits (160 the first, 128 the second, 256 the third) the system is capable of automatically recognising the type of function used for generating the MAC of the request and, in turn, use the same algorithm for the reply.

In essence, if the MAC of the request message is calculated using MD5, the MAC of the reply will also be calculated using MD5. If the MAC of the request message is calculated using SHA-1, the MAC of the reply will also be calculated using SHA-1. If the request message is in HMAC256, so will be the reply message.

**For the XML <PanAliasData> elements, the signed text will contain the values of the following subelements:** <PanAliasRev>, <PanAlias>, <PanAliasExpDate>, <PanAliasTail>, <CRecurr>

NOTES
- PanAliasExpDate and PanAliasTail are present only if the store subscribes to the service SV45 – Payments with Single Alias Pan for Card no automatic revocation and return or to the service SV88 – Tokenizer pan alias.
- CRecurr element is present in authorization panalias section only if the store substribes to the service SVA8 – Recurring payments and the request has CREATEPANALIAS = ‘S’ or TRecurr field is present.
- For recurring payments without the request of pan alias, only the CRecurr element will be present in pan alias data.

The MAC will be as follows:

With PanAliasRev
**MAC = Hash (<PanAliasRev>&<PanAlias>&<PanAliasExpDate>&<PanAliasTail>&<CRecurr>&<secret string>)**

Without PanAliasRev
**MAC = Hash (&<PanAlias>&<PanAliasExpDate>&<PanAliasTail>&<secret string>)**

For recurring payments only, without PanAlias
**MAC = Hash (<CRecurr>&<secret string>)**

NOTE: in case of signature with HMAC-256 the secret key must not be queued to the string, but rather, it must be used as a calculation key.

The wording in < > indicates the field values.

Note that the names of the XML elements are not used for calculating the MAC. Only the values are used.

An example of such a string could be the following:

&0000197412081271677&2911&0003&--8-rhXuVB56wckRc-EnCTRBa-n-UbJPD-74-mRPk-wPbqvXE4

The MAC, which is the result of a hash, must be coded appropriately for it to be trasmitted in HTTP. To that end, an exadecimal conversion must be used. The result of said conversion is a 32-character string, if the hash function used is MD5. If, on the other hand, SHA-1 is used, the result will be a 40-character string. If HMAC-256 is used, the result will be a 64-character string.
The MAC is not case sensitive. The SIA server uses capital letters.

NOTE:
If the authorization is denied, the MAC of the PanAliasData will not be calculated. All the elements of the PanAliasData section will be valued with “NULL”.
If there is an unexpected error, the MAC all the elements of the PanAliasData section will be valued with “ERROR”; the MAC will be calculated as usual.

### 4.2.20 Generating the MAC for the PANALIASREVOCATION messages

The MAC to be sent as attachment to the PANALIASREVOCATION messages can be obtained through the following procedure.

The operator may choose as it sees fit from among three standard algorythms: SHA-1 (also called SHA), MD5 and HMAC-256 (recommended). Given that the three algorythms produce a varying number of bits (160 the first, 128 the second, 256 the third) the system is capable of automatically recognising the type of function used for generating the MAC. The store site may vary the algorythm used at its will.

The merchant and SIA share a secret string of 50 or 100 characters. To produce the MAC of the data, a hash of the connection between text to be signed and secret string is performed (SHA-1 and MD5). Otherwise, an HMAC-256 is calculated using the secret string as key for the text to be signed.

**For the PANALIASREVOCATION messages, the text to be signed must contain the following fields:**

- OPERATION
- TIMESTAMP
- SHOPID
- OPERATORID
- REQREFNUM
- PANALIAS
- OPTIONS (if present)

**The MAC will be as follows:**

**MAC=Hash("OPERATION=PANALIASREVOCATION&TIMESTAMP=<timestamp>&SHOPID=<shopid>&OPERATORID=<operatorid>&REQREFNUM=<reqrefnum>&PANALIAS=<panalias>&<secret string>" )**

NOTE: in case of signature with HMAC-256 the secret key must not be queued to the string, but rather, it must be used as a calculation key.

The wording in < > indicates the field values. The order in which the fields appear is clearly fundamental. The MAC is not case sensitive. Capital and lowercase letters can be used without distinction.
Any optional fields that have not been filled in must be completely removed from the string determining the MAC.

An example of such a string could be the following:

OPERATION=PANALIASREVOCATION&TIMESTAMP=2009-02-11T15:32:58.484&SHOPID=negozio1&OPERATORID=operatore1&REQREFNUM=45434356788987661234343546547654&PANALIAS=0001110001112223322&1234567890123456789012345678901234567890

The MAC, which is the result of a hash, must be coded appropriately for it to be trasmitted in HTTP. To that end, an exadecimal conversion must be used. The result of said conversion is a 32-character string, if the hash function used is MD5. If, on the other hand, SHA-1 is used, the result will be a 40-character string. If HMAC-256 is used, the result will be a 64-character string.

### 4.2.21 Generating the MAC for the XML<AliasCreated> and <AliasRevoked> elements

The MAC which SIA VPOS encloses in the XML <AliasCreated> and <AliasRevoked> elements contained in the reply messages sent to the merchant system is obtained with the procedures described below.

The merchant and SIA share a secret string of 50 or 100 characters. To produce the MAC of the data, a hash of the connection between text to be signed and secret string is performed (SHA-1 and MD5). Otherwise, an HMAC-256 is calculated using the secret string as key for the text to be signed.

The hash function used by the system for generating the MAC is the same as that used by the merchant for generating the MAC of the request message. Given that the algorythms SHA1, MD5 and HMAC256 produce a varying number of bits (160 the first, 128 the second, 256 the third) the system is capable of automatically recognising the type of function used for generating the MAC of the request and, in turn, use the same algorithm for the reply.

In essence, if the MAC of the request message is calculated using MD5, the MAC of the reply will also be calculated using MD5. If the MAC of the request message is calculated using SHA-1, the MAC of the reply will also be calculated using SHA-1. If the request message is in HMAC256, so will be the reply message.

**For the AliasCreated element the signed text will contain the following values:**
**<PanAliasRevHash>,<PanAliasHash> ,<Timestamp> e <OperatorId>**

**The MAC will be calculated as follows:**
**MAC = Hash ( "<PanAliasRevHash>&<PanAliasHash>&<Timestamp>&<OperatorId>&**<secret string>**" )**

**In particular cases where the alias is created for the first time, the MAC will be:**
**MAC = Hash("&< PanAliasHash>&<Timestamp>&<OperatorId>&**<secret string>**")**

**For the AliasRevoked element the signed text will contain the following values:**
**<PanAliasHash> , <Timestamp>, <OperatorId> , <TimestampRev> e <OperatorRev>**

**The MAC will be calculated as follows:**
**MAC = Hash("<PanAliasHash>&<Timestamp>&<OperatorId>&<TimestampRev>&<OperatoreRev>&**<secret string>**" )**

**The wording in < > indicates the field values.**

NOTE: in case of signature with HMAC-256 the secret key must not be queued to the string, but rather, it must be used as a calculation key.

NOTE: Note that the names of the XML elements are not used for calculating the MAC. Only the values are used.

The MAC, which is the result of a hash, must be coded appropriately for it to be trasmitted in HTTP. To that end, an exadecimal conversion must be used. The result of said conversion is a 32-character string, if the hash function used is MD5. If, on the other hand, SHA-1 is used, the result will be a 40-character string. If HMAC-256 is used, the result will be a 64-character string.

The MAC is not case sensitive. The SIA server uses capital letters.

## 4.2.22 Generating the MAC for the LISTPANALIASINFO messages

The MAC to be sent as attachment to the LISTPANALIASINFO messages can be obtained through the following procedure.

The operator may choose as it sees fit from among three standard algorythms: SHA-1 (also called SHA), MD5 and HMAC-256 (recommended). Given that the three algorythms produce a varying number of bits (160 the first, 128 the second, 256 the third) the system is capable of automatically recognising the type of function used for generating the MAC. The store site may vary the algorythm used at its will.

The merchant and SIA share a secret string of 50 or 100 characters. To produce the MAC of the data, a hash of the connection between text to be signed and secret string is performed (SHA-1 and MD5). Otherwise, an HMAC-256 is calculated using the secret string as key for the text to be signed

**For the LISTALIASPANINFO messages, the text to be signed must contain the following fields:**

- OPERATION
- TIMESTAMP
- SHOPID
- OPERATORID
- REQREFNUM
- STARTDATE
- ENDDATE
- STARTTIME (if present)
- ENDTIME(if present)
- OPTIONS (if present)

**The MAC will be as follows:**

**MAC=Hash("OPERATION=LISTPANALIASINFO&TIMESTAMP=<timestamp>&SHOPID=<shopid>&OPERATORID=<operatorid>&REQREFNUM=<reqrefnum>&STARTDATE=<startdate>&ENDDATE=<enddate>&<secret string>" )**

NOTE: in case of signature with HMAC-256 the secret key must not be queued to the string, but rather, it must be used as a calculation key.

The wording in < > indicates the field values. The order in which the fields appear is clearly fundamental. The MAC is not case sensitive. Capital and lowercase letters can be used without distinction.
Any optional fields that have not been filled in must be completely removed from the string determining the MAC.

An example of such a string could be the following:

OPERATION=LISTPANALIASINFO&TIMESTAMP=2009-02-11T15:32:58.484&SHOPID=negozio1&OPERATORID=operatore1&REQREFNUM=45434356788987661234343546547654&STARTDATE=2009-04-10&ENDDATE=2009-04-10&1234567890123456789012345678901234567890

The MAC, which is the result of a hash, must be coded appropriately for it to be trasmitted in HTTP. To that end, an exadecimal conversion must be used. The result of said conversion is a 32-character string, if the hash function used is MD5. If, on the other hand, SHA-1 is used, the result will be a 40-character string. If HMAC-256 is used, the result will be a 64-character string.

## 4.2.23 Generating the MAC for the CREATELINK messages

The MAC to be sent as attachment to the CREATELINK messages can be obtained through the following procedure.

The operator may choose as it sees fit from among three standard algorythms: SHA-1 (also called SHA), MD5 and HMAC-256 (recommended). Given that the three algorythms produce a varying number of bits (160 the first, 128 the second, 256 the third) the system is capable of automatically recognising the type of function used for generating the MAC. The store site may vary the algorythm used at its will.

The merchant and SIA share a secret string of 50 or 100 characters. To produce the MAC of the data, a hash of the connection between text to be signed and secret string is performed (SHA-1 and MD5). Otherwise, an HMAC-256 is calculated using the secret string as key for the text to be signed.

**For the CREATELINK messages, the text to be signed must contain the following fields:**

- OPERATION
- TIMESTAMP
- REQREFNUM
- SHOPID
- OPERATORID
- SENDMAIL
- LINKEXPIRATIONDATE (if present)
- LINKAMOUNT
- LINKCURRENCY
- LINKEXPONENT (if present)
- LINKORDERID
- LINKURLDONE (if present)
- LINKURLMS
- LINKACCOUNTINGMODE
- LINKAUTHORMODE
- LINKLANG
- LINKSHOPEMAIL(if present)
- LINKOPTIONS (if present)
- LINKCOMMIS (if present)
- LINKEMAIL (if present)
- LINKNAME
- LINKSURNAME
- LINKORDDESCR (if present)
- LINKOPDESCR (if present)
- LINKPHONENUMBER (if present)
- LINKREMAININGDURATION (if present)
- LINKUSERID (if present)
- LINKPRODUCTREF (if present)
- LINKTRECURR (if present)
- LINKCRECURR (if present)
- THREEDSDATA (if present)
- OPTIONS (if present)

**The MAC will be as follows:**

**MAC=Hash(OPERATION=CREATELINK&TIMESTAMP=<timestamp>&REQREFNUM=<reqrefnum>&SHOPID=<shopid>&OPERATORID=<operatorid>&SENDMAIL=<sendmail>&LINKEXPIRATIONDATE=<linkexpirationdate>&LINKAMOUNT=<linkamount>&LINKCURRENCY=<linkcurrency>&LINKORDERID=<linkorderid>&LINKURLDONE=<linkurldone>&LINKURLMS=<linkurlms>&LINKACCOUNTINGMODE=<linkaccountingmode>&LINKAUTHORMODE=<linkauthormode>&LINKLANG=<linklang>&LINKEMAIL=<linkemail>&LINKNAME=<linkname>&LINKSURNAME=<linksurname>&LINKORDDESCR=<linkorddescr>& LINKOPDESCR=<linkopdescr>& LINKPHONENUMBER=<linkphonenumber>&LINKREMAININGDURATION=<linkremainingduration>&LINKUSERID=<linkuserid>&LINKPROUCTREF=<linkproductref>&LINKTRECURR=<linktrecurr>&LINKCRECURR=<linkcrecurr>&<THREEDSDATA=<threedsdata>&<secretstring>)**

NOTE: in case of signature with HMAC-256 the secret key must not be queued to the string, but rather, it must be used as a calculation key.

The wording in < > indicates the field values. The order in which the fields appear is clearly fundamental. The MAC is not case sensitive. Capital and lowercase letters can be used without distinction.
Any optional fields that have not been filled in must be completely removed from the string determining the MAC.

An example of such a string could be the following:

OPERATION=CREATELINK&TIMESTAMP=2017-04-07T13:26:14.639&REQREFNUM=20170407132523000000000000000000&SHOPID=129280509999998&OPERATORID=AF06TSTAPI1&SENDMAIL=Y&LINKEXPIRATIONDATE=2017-09-08T09:02:12.191&LINKAMOUNT=9900&LINKCURRENCY=978&LINKORDERID=PayBy01OrderIH&LINKURLDONE=http://linkdone&LINKURLMS=http://osv-sviluppo.office.corp.sia.it/atpos/pagamenti/main?PAGE&LINKACCOUNTINGMODE=I&LINKAUTHORMODE=I&LINKLANG=ITA&LINKSHOPEMAIL=sia@sia.eu&LINKOPTIONS=V&LINKCOMMIS=99&LINKEMAIL=user@mail.com&LINKNAME=Andrea&LINKSURNAME=Aiello&LINKORDDESCR=ordedescr&LINKOPDESCR=OpDesc&LINKPHONENUMBER=123456789123&LINKREMAININGDURATION=9&LINKUSERID=User

The MAC, which is the result of a hash, must be coded appropriately for it to be trasmitted in HTTP. To that end, an exadecimal conversion must be used. The result of said conversion is a 32-character string, if the hash function used is MD5. If, on the other hand, SHA-1 is used, the result will be a 40-character string. If HMAC-256 is used, the result will be a 64-character string.

## 4.2.24 Generating the MAC for the LISTLINK messages

The MAC to be sent as attachment to the LISTLINK messages can be obtained through the following procedure.

The operator may choose as it sees fit from among three standard algorythms: SHA-1 (also called SHA), MD5 and HMAC-256 (recommended). Given that the three algorythms produce a varying number of bits (160 the first, 128 the second, 256 the third) the system is capable of automatically recognising the type of function used for generating the MAC. The store site may vary the algorythm used at its will.

The merchant and SIA share a secret string of 50 or 100 characters. To produce the MAC of the data, a hash of the connection between text to be signed and secret string is performed (SHA-1 and MD5). Otherwise, an HMAC-256 is calculated using the secret string as key for the text to be signed

**For the LISTLINK messages, the text to be signed must contain the following fields:**

- OPERATION
- TIMESTAMP
- REQREFNUM
- SHOPID
- OPERATORID
- STARTDATE
- ENDDATE
- LINKSTATUS (if present)
- ORDERID (if present)
- TOKEN (if present)
- OPTIONS (if present)

**The MAC will be as follows:**

**MAC=Hash(OPERATION=LISTLINK&TIMESTAMP=<timestamp>&REQREFNUM=<reqrefnum>&SHOPID=<shopid>&OPERATORID=<operatorid>&STARTDATE=<startdate>&ENDDATE=<enddate>&TOKEN=<token> &<secretstring>)**

NOTE: in case of signature with HMAC-256 the secret key must not be queued to the string, but rather, it must be used as a calculation key.

The wording in < > indicates the field values. The order in which the fields appear is clearly fundamental. The MAC is not case sensitive. Capital and lowercase letters can be used without distinction.
Any optional fields that have not been filled in must be completely removed from the string determining the MAC.

An example of such a string could be the following:

OPERATION=LISTLINK&TIMESTAMP=2017-04-07T13:26:14.639&REQREFNUM=20170407132523000000000000000000&SHOPID=129280509999998&OPERATORID=AF06TSTAPI1&STARTDATE=2017-03-01&ENDDATE=2017-04-01&TOKEN=1qwpt99pslotul1budkx3f712

The MAC, which is the result of a hash, must be coded appropriately for it to be trasmitted in HTTP. To that end, an exadecimal conversion must be used. The result of said conversion is a 32-character string, if the hash function used is MD5. If, on the other hand, SHA-1 is used, the result will be a 40-character string. If HMAC-256 is used, the result will be a 64-character string.

## 4.2.25 Generating the MAC for the REVOKELINK messages

The MAC to be sent as attachment to the REVOKELINK messages can be obtained through the following procedure.

The operator may choose as it sees fit from among three standard algorythms: SHA-1 (also called SHA), MD5 and HMAC-256 (recommended). Given that the three algorythms produce a varying number of bits (160 the first, 128 the second, 256 the third) the system is capable of automatically recognising the type of function used for generating the MAC. The store site may vary the algorythm used at its will.

The merchant and SIA share a secret string of 50 or 100 characters. To produce the MAC of the data, a hash of the connection between text to be signed and secret string is performed (SHA-1 and MD5). Otherwise, an HMAC-256 is calculated using the secret string as key for the text to be signed

**For the REVOKELINK messages, the text to be signed must contain the following fields:**

- OPERATION
- TIMESTAMP
- REQREFNUM
- SHOPID
- OPERATORID
- TOKEN
- OPTIONS (if present)

**The MAC will be as follows:**

**MAC=Hash(OPERATION=REVOKELINK&TIMESTAMP=<timestamp>&REQREFNUM=<reqrefnum>&SHOPID=<shopid>&OPERATORID=<operatorid>&TOKEN=<token>&<secretstring>)**

NOTE: in case of signature with HMAC-256 the secret key must not be queued to the string, but rather, it must be used as a calculation key.

The wording in < > indicates the field values. The order in which the fields appear is clearly fundamental. The MAC is not case sensitive. Capital and lowercase letters can be used without distinction.
Any optional fields that have not been filled in must be completely removed from the string determining the MAC.

An example of such a string could be the following:

OPERATION=REVOKELINK&TIMESTAMP=2017-04-07T13:26:14.639&REQREFNUM=20170407132523000000000000000000&SHOPID=129280509999998&OPERATORID=AF06TSTAPI1&TOKEN=1qwpt99pslotul1budkx3f712

The MAC, which is the result of a hash, must be coded appropriately for it to be trasmitted in HTTP. To that end, an exadecimal conversion must be used. The result of said conversion is a 32-character string, if the hash function used is MD5. If, on the other hand, SHA-1 is used, the result will be a 40-character string. If HMAC-256 is used, the result will be a 64-character string.

### 4.2.26 Generating the MAC for the XML <LinkCreated> elements

The MAC which SIA VPOS encloses in the XML <LinkCreated> elements sent to the merchant system is obtained with the procedures described below.

The merchant and SIA share a secret string of 50 or 100 characters. To produce the MAC of the data, a hash of the connection between text to be signed and secret string is performed (SHA-1 and MD5). Otherwise, an HMAC-256 is calculated using the secret string as key for the text to be signed.

The hash function used by the system for generating the MAC is the same as that used by the merchant for generating the MAC of the request message. Given that the algorythms SHA1, MD5 and HMAC256 produce a varying number of bits (160 the first, 128 the second, 256 the third) the system is capable of automatically recognising the type of function used for generating the MAC of the request and, in turn, use the same algorithm for the reply.

In essence, if the MAC of the request message is calculated using MD5, the MAC of the reply will also be calculated using MD5. If the MAC of the request message is calculated using SHA-1, the MAC of the reply will also be calculated using SHA-1. If the request message is in HMAC256, so will be the reply message.

**For the XML *<LinkCreated>* elements, the text to be signed must contain the following fields:**

- CompleteLink
- Token
- CreationDate
- Status
- LastUseDate (if present)
- ExpirationDate
- RevokeDate (if present)
- OrderId

**The MAC will be as follows:**

**MAC=Hash("<CompleteLink>&<Token>&<CreationDate>&<Status>&<LastUseDate>&<ExpirationDate>&<RevokeDate>&<OrderId>&<secretstring>")**

NOTE: in case of signature with HMAC-256 the secret key must not be queued to the string, but rather, it must be used as a calculation key.

The wording in < > indicates the field values. The order in which the fields appear is clearly fundamental. The MAC is not case sensitive. Capital and lowercase letters can be used without distinction.
Any optional fields that have not been filled in must be completely removed from the string determining the MAC.

NOTE: Note that the names of the XML elements are not used for calculating the MAC. Only the values are used.

The MAC, which is the result of a hash, must be coded appropriately for it to be trasmitted in HTTP. To that end, an exadecimal conversion must be used. The result of said conversion is a 32-character string, if the hash function used is MD5. If, on the other hand, SHA-1 is used, the result will be a 40-character string. If HMAC-256 is used, the result will be a 64-character string.

The MAC is not case sensitive. The SIA server uses capital letters.

### 4.2.27 Generating the MAC for the PANALIASGENERATION messages

The MAC to be sent as attachment to the PANALIASGENERATION messages can be obtained through the following procedure.

The operator may choose as it sees fit from among three standard algorythms: SHA-1 (also called SHA), MD5 and HMAC-256 (recommended). Given that the three algorythms produce a varying number of bits (160 the first, 128 the second, 256 the third) the system is capable of automatically recognising the type of function used for generating the MAC. The store site may vary the algorythm used at its will.

The merchant and SIA share a secret string of 50 or 100 characters. To produce the MAC of the data, a hash of the connection between text to be signed and secret string is performed (SHA-1 and MD5). Otherwise, an HMAC-256 is calculated using the secret string as key for the text to be signed

**For the PANALIASGENERATION messages, the text to be signed must contain the following fields:**

- OPERATION
- TIMESTAMP
- SHOPID
- OPERATORID
- REQREFNUM
- PANID
- EXPDATE (if present)
- NETWORK
- OPTIONS (if present)

**The MAC will be as follows:**

**MAC=Hash(OPERATION=PANALIASGENERATION&TIMESTAMP=<timestamp>&SHOPID=<shopid>&OPERATORID=<operatorid>&REQREFNUM=<reqrefnum>&PANID=<panid>&EXPDATE=<expdate>&NETWORK=<network>&<secretscring>)**

NOTE: in case of signature with HMAC-256 the secret key must not be queued to the string, but rather, it must be used as a calculation key.

The wording in < > indicates the field values. The order in which the fields appear is clearly fundamental. The MAC is not case sensitive. Capital and lowercase letters can be used without distinction.
Any optional fields that have not been filled in must be completely removed from the string determining the MAC.

An example of such a string could be the following:

```
OPERATION=PANALIASGENERATION&TIMESTAMP=2017-04-
07T15:51:45.552&SHOPID=129280505050505&OPERATORID=000286AD&REQREFNUM=20170407155145000000
000000000000&PANID=B-7FB31251F28061234&EXPDATE=9912&NETWORK=97
```

The MAC, which is the result of a hash, must be coded appropriately for it to be trasmitted in HTTP. To that end, an exadecimal conversion must be used. The result of said conversion is a 32-character string, if the hash function used is MD5. If, on the other hand, SHA-1 is used, the result will be a 40-character string. If HMAC-256 is used, the result will be a 64-character string.

### 4.2.28 Generating a MAC for the XML <PaResData> element

The MAC attached by SIA VPOS to the XML <PaResData> elements forwarded to the merchant system can be obtained through the procedure described herein.

The merchant and SIA share a secret string of 50 or 100 characters. To produce the MAC of the data, a hash of the connection between text to be signed and secret string is performed (SHA-1 and MD5). Otherwise, an HMAC-256 is calculated using the secret string as key for the text to be signed.

The hash function used by the system for generating the MAC is the same as that used by the merchant for generating the MAC of the request message. Given that the algorythms SHA1, MD5 and HMAC256 produce a varying number of bits (160 the first, 128 the second, 256 the third) the system is capable of automatically recognising the type of function used for generating the MAC of the request and, in turn, use the same algorithm for the reply.

In essence, if the MAC of the request message is calculated using MD5, the MAC of the reply will also be calculated using MD5. If the MAC of the request message is calculated using SHA-1, the MAC of the reply will also be calculated using SHA-1. If the request message is in HMAC256, so will be the reply message.

**For the XML elements <PaResData>, the signed text will contain the values of the following subelements:**

- Xid
- Cavv
- Eci
- Status

**The MAC will be as follows:**

```
MAC=Hash(<Xid>&<Cavv>&<Eci>&<Status>&secret string>)
```

NOTE: in case of signature with HMAC-256 the secret key must not be queued to the string, but rather, it must be used as a calculation key.

The wording in < > indicates the field values. The order in which the fields appear is clearly fundamental. The MAC is not case sensitive. Capital and lowercase letters can be used without distinction.
Any optional fields that have not been filled in must be completely removed from the string determining the MAC.

The MAC, which is the result of a hash, must be coded appropriately for it to be trasmitted in HTTP. To that end, an exadecimal conversion must be used. The result of said conversion is a 32-character string, if the hash function used is MD5. If, on the other hand, SHA-1 is used, the result will be a 40-character string. If HMAC-256 is used, the result will be a 64-character string.

NOTE: The names of the XML elements are not used for calculating the MAC. Only the values are used.

NOTE: If the outcome of the request is an authentication error, the MAC will not be calculated and its value will be “NULL”.

### 4.2.29 Generating a MAC for the IBANAUTHORIZATION message

The MAC to be sent as attachment to the IBANAUTHORIZATION messages can be obtained through the procedure described herein.

The operator may choose as it sees fit from among three standard algorythms: SHA-1 (also called SHA), MD5 and HMAC-256 (recommended). Given that the three algorythms produce a varying number of bits (160 the first, 128 the second, 256 the third) the system is capable of automatically recognising the type of function used for generating the MAC. The store site may vary the algorythm used at its will.

The merchant and SIA share a secret string of 50 or 100 characters. To produce the MAC of the data, a hash of the connection between text to be signed and secret string is performed (SHA-1 and MD5). Otherwise, an HMAC-256 is calculated using the secret string as key for the text to be signed.

**For the IBANAUTHORIZATION messages, the text to be signed must contain the following fields:**

- OPERATION
- TIMESTAMP
- SHOPID
- ORDERID
- OPERATORID
- REQREFNUM
- IBAN
- AMOUNT
- CURRENCY
- EXPONENT (if present)
- ACCOUNTINGMODE
- NETWORK
- EMAILCH (if present)
- USERID (if present)
- IPADDRESS (if present)
- OPDESCR (if present)
- USRAUTHFLAG (if present)
- OPTIONS (if present)
- ANTIFRAUD (if present)
- PRODUCTREF (if present)
- NAME (if present)
- SURNAME (if present)
- TAXID (if present)

**The MAC will be as follows:**

**MAC=Hash(OPERATION=IBANAUTHORIZATION&TIMESTAMP=<timestamp>&SHOPID=<shopid>&ORDERID=<OrderID>&OPERATORID=<OperatorID>&REQREFNUM=<requestnumber>&PAN=<pan>&CVV2=<cvv2>&EXPDATE=<expirydate>&AMOUNT=<Amount>&CURRENCY=<Currency>&ACCOUNTINGMODE=<accountingmode>&NETWORK=<network>&EMAILCH=<cardholderemail>USERID=<userid>&IPADDRESS=<ipaddress>&OPDESCR=<OpDescr>&<secret string > )**

NOTE: in case of signature with HMAC-256 the secret key must not be queued to the string, but rather, it must be used as a calculation key.

The wording in < > indicates the field values. The order in which the fields appear is clearly fundamental. The MAC is not case sensitive. Capital and lowercase letters can be used without distinction.

An example of such a string could be the following:

OPERATION=IBANAUTHORIZATION&TIMESTAMP=2019-04-08T13:04:21.852&SHOPID=123456789012345&ORDERID=ord1&OPERATORID=KR839H&REQREFNUM=20190501496204690934584305834564&PAN=1234567812345678&EXPDATE=0317&AMOUNT=100&CURRENCY=978&ACCOUNTINGMODE=I&NETWORK=01&Absd830923fk32..

The MAC, which is the result of a hash, must be coded appropriately for it to be trasmitted in HTTP. To that end, an exadecimal conversion must be used. The result of said conversion is a 32-character string, if the hash function used is MD5. If, on the other hand, SHA-1 is used, the result will be a 40-character string. If HMAC-256 is used, the result will be a 64-character string.

### 4.2.30 Generating a MAC for the XML <ThreeDSMethod> elements

The MAC sent by SIA VPOS as attachment to the XML <ThreeDSMethod> elements can be obtained through the procedure described herein.

The merchant and SIA share a secret string of 50 or 100 characters. To produce the MAC of the data, a hash of the connection between text to be signed and secret string is performed (SHA-1 and MD5). Otherwise, an HMAC-256 is calculated using the secret string as key for the text to be signed.

The hash function used by the system for generating the MAC is the same as that used by the merchant for generating the MAC of the request message. Given that the algorythms SHA1, MD5 and HMAC256 produce a varying number of bits (160 the first, 128 the second, 256 the third) the system is capable of automatically recognising the type of function used for generating the MAC of the request and, in turn, use the same algorithm for the reply.

In essence, if the MAC of the request message is calculated using MD5, the MAC of the reply will also be calculated using MD5. If the MAC of the request message is calculated using SHA-1, the MAC of the reply will also be calculated using SHA-1. If the request message is in HMAC256, so will be the reply message.

**For the XML elements <ThreeDSMethod>, the signed text will contain the values of the following subelements:**

- ThreeDSTransId
- ThreeDSMethodData
- ThreeDSMethodUrl

**The MAC will be as follows:**

**MAC=Hash(<ThreeDSTransId>&<ThreeDSMethodData>&<ThreeDSMethodUrl>&<secret string> )**

NOTE: in case of signature with HMAC-256 the secret key must not be queued to the string, but rather, it must be used as a calculation key.

The wording in < > indicates the field values. The order in which the fields appear is clearly fundamental. The MAC is not case sensitive. Capital and lowercase letters can be used without distinction.

An example of such a string could be the following:

I&8032180310wieeuejjwerrrrr&01&order1&10000&10000&978&10000&0&00&2015-07-06T13:04:34&123456&234569&05423956754389&02&Absd830923fk32..

The MAC, which is the result of a hash, must be coded appropriately for it to be trasmitted in HTTP. To that end, an exadecimal conversion must be used by the SIA server.
The result of said conversion is a 32-character string, if the hash function used is MD5. If, on the other hand, SHA-1 is used, the result will be a 40-character string. If HMAC-256 is used, the result will be a 64-character string.

NOTE: The names of the XML elements are not used for calculating the MAC. Only the values are used.

### 4.2.31 Generating a MAC for the XML <ThreeDSChallenge> elements

The MAC sent by SIA VPOS as attachment to the XML <ThreeDSChallenge> elements can be obtained through the procedure described herein.

The merchant and SIA share a secret string of 50 or 100 characters. To produce the MAC of the data, a hash of the connection between text to be signed and secret string is performed (SHA-1 and MD5). Otherwise, an HMAC-256 is calculated using the secret string as key for the text to be signed.

The hash function used by the system for generating the MAC is the same as that used by the merchant for generating the MAC of the request message. Given that the algorythms SHA1, MD5 and HMAC256 produce a varying number of bits (160 the first, 128 the second, 256 the third) the system is capable of automatically recognising the type of function used for generating the MAC of the request and, in turn, use the same algorithm for the reply.

In essence, if the MAC of the request message is calculated using MD5, the MAC of the reply will also be calculated using MD5. If the MAC of the request message is calculated using SHA-1, the MAC of the reply will also be calculated using SHA-1. If the request message is in HMAC256, so will be the reply message.

**For the XML elements <ThreeDSChallenge>, the signed text will contain the values of the following subelements:**

- ThreeDSTransId
- CReq
- ACSUrl

**The MAC will be as follows:**

**MAC=Hash(<ThreeDSTransId>&<CReq>&<ACSUrl>&<secret string> )**

NOTE: in case of signature with HMAC-256 the secret key must not be queued to the string, but rather, it must be used as a calculation key.

The wording in < > indicates the field values. The order in which the fields appear is clearly fundamental. The MAC is not case sensitive. Capital and lowercase letters can be used without distinction.

An example of such a string could be the following:

I&8032180310wieeuejjwerrrrr&01&order1&10000&10000&978&10000&0&00&2015-07-06T13:04:34&123456&234569&05423956754389&02&Absd830923fk32..

The MAC, which is the result of a hash, must be coded appropriately for it to be trasmitted in HTTP. To that end, an exadecimal conversion must be used by the SIA server.
The result of said conversion is a 32-character string, if the hash function used is MD5. If, on the other hand, SHA-1 is used, the result will be a 40-character string. If HMAC-256 is used, the result will be a 64-character string.

NOTE: The names of the XML elements are not used for calculating the MAC. Only the values are used.

# 4.3 Parameters AUTHORMODE, ACCOUNTINGMODE and possible scenarios

Here follows a brief description of the meaning of the parameters TAUTOR and TCONTAB in connection with the various possible uses of the SIA VPOS system.

## 4.3.1 AUTHORMODE

### Immediate authorization I

The immediate authorization procedure provides that during the online payment phase the authorization request is sent immediately to the international circuits. Once the transaction has been successfully completed, the merchant is certain that what is owed by the customer has been "booked" from his ceiling.

### Deferred authorization D

The deferred authorization procedure provides that during the online payment phase the transactions are accepted but not forwarded to the circuits (the card's validity is in any case verified at the issuer's).
The merchant who follows this payment acceptance procedure will eventually be able to have the pending authorization requests processed. The PIB may receive deferred authorization requests for an amount lower than the original; the merchant may forward as many deferred authorizations up to the original total amount.

## 4.3.2 ACCOUNTINGMODE

### Immediate booking I

The immediate booking procedure permits the merchant to make any authorized transactions automatically bookable. Without merchant's intervention, the same evening of the day on which the transaction took place, the front end processor automatically performs a clearing of the transactions for the full authorized amount.
This procedure can be adopted, for example, in the case where the goods/services sold can be used immediately by the acquirer (software, music, online services, etc.).

For ASI card verification transactions the accounting mode must be set to D and this option is not available.

### Deferred booking D

The deferred booking procedure provides that authorized transactions are explicitly made bookable by the merchant. The merchant has a preset number of days from the time authorization is granted to book a transaction.

This procedure makes available to the merchant the following transactions:

- Overall booking: a transaction is made bookable for the full amount of the authorized sum.
- Partial booking: a transaction is made bookable for an amount which is lower than the authorized sum; a partial booking transaction may refer to an authorization for which a partial booking (split shipment) has already been requested, provided that the final booking term has not expired.
- Cancellation: a booking transaction carried out during the day is cancelled, the transaction can be booked again.

For ASI card verification transactions the accounting mode must be set to D and this option is the only one available.

## 4.3.3 Possible scenarios

Here below is a list of possible scenarios of the system operation

| Functionality | Mechanism that can be used |
|---|---|
| Payment of immaterial goods (download or services) | Payment initiation with immediate authorization AUTHORMODE=I, immediate or deferred booking |
| Payment of inseparable material goods always available | Initiation of payment with immediate authorization AUTHORMODE=I, immediate or deferred booking |
| Payment of material goods to be procured | Initiation of payment with deferred authorization (AUTHORMODE=D) and subsequent authorization request with immediate booking |
| | |
| | |
| Split shipment (division and/or reduction): delivery of goods at different stages<br>Assumption:<br>• The store knows beforehand that it will deliver in pieces<br>• The total amount is known in advance | Initiation of payment with deferred authorization (AUTHORMODE=D) and N subsequent authorization requests with immediate booking |
| Split shipment (division and/or reduction): delivery of goods at different stages<br>Assumption:<br>• The store did not know in advance that it was to deliver by instalments<br>• An online authorization has been sent for the full amount | In this circumstance, a split (division and/or reduction) of the authorization must be performed: this transaction will transform an online order into a deferred order.<br>After the split (division and/or reduction) transaction, N authorization requests can be sent as in the case of a normal deferred authorization |
| | |

## 4.4 ThreeDSData (API)

THREEDSDATA field must be obtained through AES encryption of the JSON representation of all the fields the merchant wants to send to the networks. The following table contains all the fields that can be include within 3DSDATA. Encryption algorithm must be AES/CBC/PKCS5Padding and must use as encrypting key the first 16 bytes of the API secret key. The initialization vector to be used for data encryption must be 16 bytes length equal to 0. Encrypted byte array must encoded to base64.

The following table lists all the fields that can be used in the JSON object for the 3DSDATA. The JSON object is a simple unordered set of name/value pairs. All strings are UTF-8 encoded.

Note that the fields descriptions and the related references reported in the table are directly extracted from the EMVco standard defining 3DS 2.

| Field Name | Short Description | Description | Values | API |
|---|---|---|---|---|
| browserAcceptHeader | Browser Accept Headers | Exact content of the HTTP accept headers as sent to the 3DS Requestor from the Cardholder's browser. | Length: Variable, maximum 2048 characters<br>JSON Data Type: String<br>Value accepted:<br>If the total length of the accept header sent by the browser exceeds 2048 characters, the 3DS Server truncates the excess portion.<br>Refer to Section A.5.2 for additional detail. | R |
| browserIP | Browser IP Address | IP address of the browser as returned by the HTTP headers to the 3DS Requestor. | Length: Variable, maximum 45 characters<br>JSON Data Type: String<br>Value accepted:<br>• 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<br>• 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<br>Refer to Section A.5.2 for additional detail. | R |
| browserJavaEnabled | Browser Java Enabled | Boolean that represents the ability of the cardholder browser to execute Java.<br>Value is returned from the navigator.javaEnabled property. | JSON Data Type: Boolean<br>Values accepted:<br>• true<br>• false | R |

|  |  |  |  |  |
|---|---|---|---|---|
|  |  | Refer to Section A.5.2 for additional detail. |  |  |
| browserLanguage | Browser Language | Value representing the browser language as defined in IETF BCP47.<br>Returned from navigator.language property.<br>Refer to Section A.5.2 for additional detail. | Length: Variable, 1–8 characters<br>JSON Data Type: String | R |
| browserColorDepth | Browser Screen Color | t depth of the color palette for displaying images, in bits per pixel.<br>Obtained from Cardholder browser using the screen.colorDepth property.<br>Refer to Section A.5.2 for additional detail. | Length: 1–2 characters<br>JSON Data Type: String<br>Values accepted:<br>• 1 = 1 bit<br>• 4 = 4 bits<br>• 8 = 8 bits<br>• 15 = 15 bits<br>• 16 = 16 bits<br>• 24 = 24 bits<br>• 32 = 32 bits<br>• 48 = 48 bits | R |
| browserScreenHeight | Browser Screen Height | Total height of the Cardholder's screen in pixels.<br>Value is returned from the screen.height property.<br>Refer to Section A.5.2 for additional detail. | Length: Variable, 1–6 characters<br>JSON Data Type: String | R |
| browserScreenWidth | Browser Screen Width | Total width of the cardholder's screen in pixels.<br>Value is returned from the screen.width property.<br>Refer to Section A.5.2 for additional detail. | Length: Variable, 1–6 characters<br>JSON Data Type: String | R |
| browserTZ | Browser Time Zone | Time difference between UTC time and the Cardholder browser local time, in minutes. | Length: 1–5 characters<br>JSON Data Type: String<br>Value accepted:<br>Value is returned from the getTimezoneOffset() method.<br>Refer to Section A.5.2 for additional detail. | R |
| browserUserAgent | Browser User-Agent | Exact content of the HTTP user-agent header. | Length: Variable, maximum 2048 characters<br>JSON Data Type: String<br>Value accepted:<br>Note: If the total length of the User-Agent sent by the browser exceeds 2048 characters, the 3DS Server truncates the excess portion. | R |

| | | | | |
|---|---|---|---|---|
| | | | Refer to Section A.5.2 for additional detail. | |
| threeDSRequestorChallengeInd | 3DS Requestor Challenge Indicator | Indicates whether a challenge is requested for this transaction.<br>For example:<br>For 01-PA, a 3DS Requestor may have concerns about the transaction, and request a challenge.<br>For 02-NPA, a challenge may be necessary when adding a new card to a wallet.<br>For local/regional mandates or other variables. | Length: 2 characters<br>JSON Data Type: String<br>Values accepted:<br>• 01 = No preference<br>• 02 = No challenge requested<br>• 03 = Challenge requested: 3DS Requestor Preference<br>• 04 = Challenge requested: Mandate<br>• 05–79 = Reserved for EMVCo future use (values invalid until defined by EMVCo)<br>• 80-99 = Reserved for DS use<br>Note: If the element is not provided, the expected action is that the ACS would interpret | O |
| addrMatch | Address Match Indicator | Indicates whether the Cardholder Shipping Address and Cardholder Billing Address are the same. | Y or N | Or |
| chAccAgeInd | Cardholder Account Age Indicator | Length of time that the cardholder has had the account with the 3DS Requestor. | • 01 = No account (guest check-out)<br>• 02 = Created during this transaction<br>• 03 = Less than 30 days<br>• 04 = 30-60 days<br>• 05 = More than 60 days | Or |
| chAccChange | Cardholder Account Change | 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. | Date format = YYYYMMDD | Or |
| chAccChangeInd | Cardholder Account Change Indicator | 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. | • 01 = Changed during this transaction<br>• 02 = Less than 30 days<br>• 03 = 30-60 days<br>• 04 = More than 60 days | O |
| chAccDate | Cardholder Account Date | Date that the cardholder opened the account with the 3DS Requestor. | Date format = YYYYMMDD | O |
| chAccPwChange | Cardholder Account | Date that cardholder's account with the 3DS | Date format = YYYYMMDD | O |

| | | | | |
|---|---|---|---|---|
| | Password Change | Requestor had a password change or account reset. | | |
| chAccPwChangeInd | Cardholder Account Password Change Indicator | Indicates the length of time since the cardholder’s account with the 3DS Requestor had a password change or account reset. | • 01 = No change<br>• 02 = Changed during this transaction<br>• 03 = Less than 30 days<br>• 04 = 30-60 days<br>• 05 = More than 60 days | O |
| nbPurchaseAccount | Cardholder Account Purchase Count | Number of purchases with this cardholder account during the previous six months. | String max 4 | O |
| txnActivityDay | Number of Transactions Day | Number of transactions (successful and abandoned) for this cardholder account with the 3DS Requestor across all payment accounts in the previous 24 hours. | String max 3 | O |
| txnActivityYear | Number of Transactions Year | Number of transactions (successful and abandoned) for this cardholder account with the 3DS Requestor across all payment accounts in the previous year. | String max 3 | O |
| shipAddressUsage | Shipping Address Usage | Date when the shipping address used for this transaction was first used with the 3DS Requestor. | Date format = YYYYMMDD | O |
| shipAddressUsageInd | Shipping Address Usage Indicator | Indicates when the shipping address used for this transaction was first used with the 3DS Requestor. | • 01 = This transaction<br>• 02 = Less than 30 days<br>• 03 = 30-60 days<br>• 04 = More than 60 days | O |
| shipNameIndicator | Shipping Name Indicator | Indicates if the Cardholder Name on the account is identical to the shipping Name used for this transaction. | • 01 = Account Name identical to shipping Name<br>• 02 = Account Name different than shipping Name | O |
| acctID | Cardholder Account Identifier | | String max 64 | O |
| billAddrCity | Cardholder Billing Address City | The city of the Cardholder billing address associated with the card used for this purchase. | String max 50 | Or |

| | | | | |
|---|---|---|---|---|
| billAddrCountry | Cardholder Billing Address Country | The country of the Cardholder billing address associated with the card used for this purchase. | ISO 3166-1 numeric three-digit country code, other than exceptions listed in Table A.5. | Or |
| billAddrLine1 | Cardholder Billing Address Line 1 | First line of the street address or equivalent local portion of the Cardholder billing address associated with the card used for this purchase. | String max 50 | Or |
| billAddrLine2 | Cardholder Billing Address Line 2 | Second line of the street address or equivalent local portion of the Cardholder billing address associated with the card used for this purchase. | String max 50 | O |
| billAddrLine3 | Cardholder Billing Address Line 3 | Third line of the street address or equivalent local portion of the Cardholder billing address associated with the card used for this purchase. | String max 50 | O |
| billAddrPostCode | Cardholder Billing Address Postal Code | ZIP or other postal code of the Cardholder billing address associated with the card used for this purchase | String max 16 | Or |
| billAddrState | Cardholder Billing Address State | The state or province of the Cardholder billing address associated with the card used for this purchase. | Variable, maximum 3 characters. Should be country subdivision code defined in ISO 3166-2 | Or |
| homePhone | Cardholder Home Phone Number | The home phone number provided by the Cardholder. | country code(1-3) - number (max 15) | Or |
| mobilePhone | Cardholder Mobile Phone Number | The mobile phone number provided by the Cardholder. | country code(1-3) - number (max 15) | Or |
| shipAddrCity | Cardholder Shipping Address City | City portion of the shipping address requested by the Cardholder. Required unless shipping information is the same as billing information (addrMatch = Y). | String max 50 | O |

|  |  |  |  |  |
|---|---|---|---|---|
| shipAddrCountry | Cardholder Shipping Address Country | Country of the shipping address requested by the Cardholder. Required unless shipping information is the same as billing information (addrMatch = Y). | ISO 3166-1 numeric three-digit country code, other than exceptions listed in Table A.5. | O |
| shipAddrLine1 | Cardholder Shipping Address Line 1 | First line of the street address or equivalent local portion of the shipping address requested by the Cardholder. Required unless shipping information is the same as billing information (addrMatch = Y). | String max 50 | O |
| shipAddrLine2 | Cardholder Shipping Address Line 2 | Second line of the street address or equivalent local portion of the shipping address requested by the Cardholder. | String max 50 | O |
| shipAddrLine3 | Cardholder Shipping Address Line 3 | Third line of the street address or equivalent local portion of the shipping address requested by the Cardholder. | String max 50 | O |
| shipAddrPostCode | Cardholder Shipping Address Postal Code | The ZIP or other postal code of the shipping address requested by the Cardholder. Required unless shipping information is the same as billing information (addrMatch = Y). | String max 16 | O |
| shipAddrState | Cardholder Shipping Address State | The state or province of the shipping address associated with the card being used for this purchase. Required unless shipping information is the same as billing information (addrMatch = Y). | Variable, maximum 3 characters. Should be country subdivision code defined in ISO 3166-2 | O |
| workPhone | Cardholder Work Phone Number | The work phone number provided by the Cardholder. | country code(1-3) - number (max 15) | O |

| | | | | |
|---|---|---|---|---|
| deliveryEmailAddress | Delivery Email Address | For Electronic delivery, the email address to which the merchandise was delivered. | String max 254 | Or |
| deliveryTimeframe | Delivery Timeframe | Indicates the merchandise delivery timeframe. | • 01 = Electronic Delivery<br>• 02 = Same day shipping<br>• 03 = Overnight shipping<br>• 04 = Two-day or more shipping | Or |
| preOrderDate | Pre-Order Date | For a pre-ordered purchase, the expected date that the merchandise will be available. | Date format = YYYYMMDD | Or |
| preOrderPurchaseInd | Pre-Order Purchase Indicator | Indicates whether Cardholder is placing an order for merchandise with a future availability or release date. | • 01 = Merchandise available<br>• 02 = Future availability | Or |
| reorderItemsInd | Reorder Items Indicator | Indicates whether the cardholder is reordering previously purchased merchandise. | Length: 2 characters<br>JSON Data Type: String<br>Values accepted:<br>• 01 = First time ordered<br>• 02 = Reordered | Or |
| shipIndicator | Shipping Indicator | Indicates shipping method chosen for the transaction.<br>Merchants must choose the Shipping Indicator code that most accurately describes the cardholder's specific transaction, not their general business.<br>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. | • 01 = Ship to cardholder's billing address<br>• 02 = Ship to another verified address on file with merchant<br>• 03 = Ship to address that is different than the cardholder's billing address<br>• 04 = "Ship to Store" / Pick-up at local store (Store address shall be populated in shipping address fields)<br>• 05 = Digital goods (includes online services, electronic gift cards and redemption codes)<br>• 06 = Travel and Event tickets, not shipped<br>• 07 = Other (for example, Gaming, digital services not shipped, emedia subscriptions, etc.) | Or |

Please note that:

- According to the EMVCo 3DS standards "3DS Requestor" stands for "Merchant".
- All strings must use UTF-8-character set.
- **Inclusion** column meaning:

- “R” required
- “Or” optional recommended
- “O” optional

| | | | | | |
|---|---|---|---|---|---|
| SP | 0 | @ | P | ` | p |
| ! | 1 | A | Q | a | q |
| “ | 2 | B | R | b | r |
| # | 3 | C | S | c | s |
| $ | 4 | D | T | d | t |
| % | 5 | E | U | e | u |
| & | 6 | F | V | f | v |
| ‘ | 7 | G | W | g | w |
| ( | 8 | H | X | h | x |
| ) | 9 | I | Y | i | y |
| * | : | J | Z | j | z |
| + | ; | K | [ | k | { |
| , | < | L | \ | l | \| |
| - | = | M | ] | m | } |
| . | > | N | ^ | n | ~ |
| / | ? | O | _ | o | |

## Code example

The following Java code is provided just as a mean to clarify the encryption process to be applied to produce the 3DSDATA field.

```
import java.security.InvalidAlgorithmParameterException;
import java.io.UnsupportedEncodingException;
import java.security.InvalidKeyException;
import java.security.NoSuchAlgorithmException;

import javax.crypto.BadPaddingException;
import javax.crypto.Cipher;
import javax.crypto.IllegalBlockSizeException;
import javax.crypto.NoSuchPaddingException;
import javax.crypto.spec.IvParameterSpec;
import javax.crypto.spec.SecretKeySpec;
import javax.xml.bind.DatatypeConverter;

public class Utility {

   public static String encode3DSdata(String APISecretMerchant, String
JSONobject) throws Throwable {

        // Initialization vector
        byte[] iv = { 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 };

        // AES Key from the API merchant key
        byte[] key = APISecretMerchant.substring(0, 16).getBytes();
        IvParameterSpec ivParameterSpec = new IvParameterSpec(iv);
        SecretKeySpec secretKeySpec = new SecretKeySpec(key, "AES");

        // What we should encrypt
        byte[] toEncrypt = JSONobject.getBytes("UTF-8");

        // Encrypt
        Cipher cipher = Cipher.getInstance("AES/CBC/PKCS5Padding");
        cipher.init(Cipher.ENCRYPT_MODE, secretKeySpec, ivParameterSpec);
        byte[] encrypted = cipher.doFinal(toEncrypt);

        // Convert to base64
        return DatatypeConverter.printBase64Binary(encrypted);

    }
}
```