cardlink

Cardlink Payment Gateway

Merchant integration
XML 4.1/2.1

17. Jun. 2019 (revised)

Contents

1. Overview

Modirum VPOS is a payment application that is designed for processing merchant payments in ecommerce environment. The inputs to VPOS are requests from merchant shopping solution and from t here the payment process is controlled by VPOS until the payment has completed successfully or failed a nd the information will be sent back to merchant shopping solution.

The payment methods available will depend on area application will be used and which are necessary fo r the client business model. It could have enabled credit and debit card payments that are also integrate d with Modirum 3D Secure merchant plugin technology or external payment methods like net payments in shopper local banks and so on. Exact payment methods available should be specified by client.

Modirum VPOS core design enables multiple types of merchant interfaces to be implemented and also t he easy to implement default interface and MPI integrated version is provided for reference.

Merchants can easily attach their look and feel to payment pages by supplying their own custom CSS styl esheet.

This document describes newest versions (4.1 and 2.1) of interfaces to date based on RSA SHA256 signature security (4.1) and shared secret based SHA2-256 digest (2.1).

2. XML API Interface

The XML API interface plugin makes possible that merchants with their own payment pages hosted in their system to use e-commerce services provided by VPOS using XML messaging.
XML Messaging is using request real time and response messages in the same request/response cycle. In request message merchant provides payment and order info and in response messages VPOS indicates the result of the action performed. By default the merchant should receive the response message within 30 seconds maximum.
Root element of request and response messages is *VPOS*
Current version of XML API is 4.1 and 2.1 that is copy of 4.1 only difference is that message security is in 2.1 ensured by a Digest element computed from canonicalized Message element appended with shared sercret.
The request message general structure:

```
<VPOS>
        <Message version="4.1" messageId="12345" timeStamp="" lang="en">
                <xxxxxRequest>
                        <Authentication> ...
                        </Authentication>
                        <OrderInfo> ..
                        </OrderInfo>
                        <PaymentInfo>
                                ..<ThreeDSecure>...</ThreeDSecure>
                        </PaymentInfo>
                </xxxxxRequest>
        </Message>
        <Signature>...</Signature>
</Merchant-VPOS>
```

The response message general structure:

```
<VPOS>
         <Message version="4.1" messageId="12345">
                <xxxxxResponse>
                         <OderId></OrderId/>
                          <OrderAmount><OrderAmount/>
                          <PaymentTotal></PaymentTotal>
                        <Currency></Currency>
                         <Status></Status>
                        <TxId></TxId>
                        <Sequence></Sequence>
                        <SeqTxId></SeqTxId>
                        <PaymentRef></PaymentRef>
                        <RiskScore></PaymentRef>
                         <ErrorCode></ErrorCode>
                        <Description></Description>
                </xxxxxResponse>
           </Message>
           <Signature>..</Signature>
</VPOS>
```

The general error message structure (returned in case request: message was unparseable or unvalidatable)

```
<VPOS>
  <Message version="1.0" messageId="12345">
        <ErrorMessage>
        <ErrorCode></ErrorCode>
        <Description></Description>
        <OriginalXML></OriginalXML>
        </ErrorMessage>
  </Message>
</VPOS>
```

The exact xml bindings are defined in xsd schema.
https://cardlink.test.modirum.com/vpos/xsd/VPOS41.xsd

Description *of request and response message elements and fields and their usage:*

Field/request	***Type***	***Description***
Request		
VPOS	element	XML root element
Message	element type Message	Message contents element
version	attribute, xsi:string	Message version default value "4.1" Required or 2.1
messageId	attribute, xsi:ID	Message unique identifier (values in request and reply messages this must match, also used for lookup signature reference object when validating signature) (“M1234567”)
lang	attribute, xsi:string(2)	Message attribute to specify context language (Optional) (ISO 639-1 language code en, fi, sv, el, etc..)
timeStamp	Attribute xsi:dateTime	Approximate time when message was created (optiuonal for now but recommended)
Digest (v2.1 only)	element xsi:string	**Required if version = 2.1.** The digest of message element if used instead of password to be calulated Base64(SHA2-256((utf8bytes(canonicalize(Message))+utf8bytes(sharedSecret))
Signature	element ds:SignatureType	**Required if version = 4.1** The xml signature as defined https://www.w3.org/TR/xmldsig-core/ Canonicalization http://www.w3.org/TR/2001/REC-xml-c14n-20010315 SignatureMethod Algorithm="http://www.w3.org/2001/04/xmldsig-more#rsa-sha256” Digest Method

		Algorithm="http://www.w3.org/2001/04/xmlenc#sha256" **Requests are signed by merchant private key and validated with merchant Certificate (merchant certificate generation is referred to section 5 page 30)**
SaleRequest, AuthorisationRequest, CaptureRequest, OriginalCreditRequest RefundRequest, CancelRequest RecurringOperationRequest, StatusRequest, TokenizationRequest	element	Request Message element depending on request type
Authentication	element	Authentication element of request Message
Mid	xsi:string (N1..30)	Merchant number/identification in VPOS
OrderInfo		Orderinfo element of request Message
DeviceCategory	xsi:string (1)	Optional
OrderId	xsi:string AN1..50	Merchant defined unique order id
OrderDesc	xsi:string AN1..128	Order description defined by Merchant
OrderAmount	xsi:decimal (max 9+3 or 10+2)	Order amount (decimal number >0.0 and max 12 digits + decimal point)
Currency	xsi:string A3	ISO4217 alphabetic currency code (USD, EUR)
PayerEmail	xsi:string AN1..64	Order payer email address (string..64)
PayerPhone	xsi:string N1..30	Order payer phone number, optional but strongly recommended (string..30)
AddFraudScore	xsi:integer	Incoming starting risk score (integer)
BlockScore	xsi:integer	Optional block score parameter that will be used to block the transaction if transaction riskScore reaches this value or above. (Postive Integer number)
Elements Var1.Var9 Var1, Var2, Var3, Var4, Var5, Var6, Var7, Var8, Var9	xsi:string AN1..1024	Free variable defined by merchant.
MOTO	xsi:integer N1	Indicating whether it is a MOTO transaction (1 indicates MOTO)
Weight	xsi:decimal	Order shipping weight (kg) if item is shippable and shipping needs to be calculated by VPOS (decimal number >0) and it is supported
Dimensions	xsi:string AN1..25	Order shipping dimensions (mm) in format width: height: depth for example a box 200:200:200 (string..25) can be used for shipping calculation if implemented so

BillingAddress	**element address**	Element of OrderInfo
country	xsi:string AN2	Billing address country code (string 2 ISO 3166-1-alpha-2 code (US, FI, GB))
state	xsi:string AN1..50	Billing address state (string.50)
zip	xsi:string AN1..16	Billing address zip code (string..16)
city	xsi:string AN1..64	Billing address city (string..64)
address	xsi:string AN1..100	Billing address street (string..100)
ShippingAddress	element:address	Element of OrderInfo
country	xsi:string AN2	Shipping address country code (string 2 ISO 3166-1-alpha-2 code (US, FI, GB)) Optional, required when parameter weight or dimensions are present.
state	xsi:string AN1..50	Shipping address state (string..50) Optional, required when parameter weight or dimensions are present.
zip	xsi:string AN1..16	Shipping address zip code (string..16) Optional, required when parameter weight or dimensions are present. Optional, required when parameter weight or dimensions are present.
city	xsi:string AN1..64	Shipping address city (string..64) Optional, required when parameter weight or dimensions are present.
address	xsi:string AN1..100	Shipping address street (string..100) Optional, required when parameter weight or dimensions are present.
PaymentInfo		Payment info element of request
PayMethod	xsi:string AN1..20	valid values: visa for VISA cards, mastercard for MasterCard, maestro for Maestro, amex for American Express, diners for Diners, discover for Discover
CardPan	xsi:string N11..19	Card number
CardExpDate	xsi:string N4	Card expiration date in format YYMM
CardCvv2	xsi:string N3..4	CVV2/CVC2 security code from card.
CardHolderName	xsi:string AN1..24	Card holder name
CardEncData	Xsi:string ..2048	In case on merchant merchant site user browser RSA card data encryption is used this field contains encrypted card data in form of **Base64**(**RSA**(**UTF8Bytes**("pn={pan}&ey={expyear}&em={expmonth}&c2={cvv2}&cn={cardholdername}")) Values are urlencoded and with utf-8 char encoding (with javascript encodeURIComponent). This all is handled by server supplied component,

		merchant just need to forward value as returned to this field content. If this field is present then fields CardPan, CardExpDate, CardHolderName, CardCvv2 must not be present
RecurringIndicator	xsi:string AN1	Value "R" indicates recurring payment
RecurringParameters	element	Recurring parameters element
ExtRecurringfrequency	xsi:string N1..3	A value indicating the number of days between the recurring payments. 28 is a special value indicating a month.
ExtRecurringenddate	xsi:string N8	Recurring end date Format yyyymmdd
InstallmentParameters	element	Installments parameters element
ExtInstallmentoffset	xsi:integer N1..2	Defines the number of months between the entering of the transaction, n case installment payment
ExtInstallmentperiod	xsi:integer N1..2	Defines the number of monthly payments in case installment payment. Valid value must be >1
ThreeDSecure	element	Element to support ThreeDSecure in XML api
EnrollmentStatus	xsi:string AN1	**In case of merchant is processing 3D secure prior to sending this xml message this field should contain 3D secure enrollment status (Y, N, U)**
AuthenticationStatus	xsi:string AN1	**In case of merchant is processing 3D secure prior to sending this xml message this field should contain 3D secure authentication status (Y, N, U, A)**
CAVV	elem xsi:string AN 28	**In case of merchant is processing 3D secure prior to sending this xml message this field should contain 3D secure CAVV if authenticated. Base64 encoded value (28 chars) of CAVV of value of 20 bytes**
XID	elem xsi:string AN 28	**In case of merchant is processing 3D secure prior to sending this xml message this field should contain 3D secure XID if authenticated. base64 encoded 28 char value of 20 byte XID**
ECI	elem xsi:string N2	**In case of merchant is processing 3D secure prior to sending this xml message this field can optionally contain ECI value**
Protocol	elem xsi:string	**Required if not 3DS1, value from MPI responses** **values 3DS1.0.2, 3DS2.1.0**
Attribute	elem AttributeType 0..n counts	**Extra attributes for 3DS2** add all attibutes with names TDS2.transStatus TDS2.transStatusReason

		TDS2.threeDSServerTransID **TDS2.dsTransID** TDS2.acsTransID TDS2.authenticationType TDS2.challengeCancel depending if available in MPI response. **Attribute named TDS2.dsTransID is currently required if successful 3DS2 authentication, ot hers currently recommended.**
ExtXOrderId	xsi:string AN1..50	Optional merchant and acquirer agreed extens ion for recognizing returning customers with submitting previous successful order id of the merchant recognized customer. If functionality is not enabled for merchant this parameter is silently ignored. And if in such case CardPan is missing or is not valid error condition will be generated. Also used in original credit to locate original payment.
ExtTokenOptions	Xsi:string N1	Optional for merchant and acquirer agreed tok en extension Value 1 if request tokenization and PAN is supplied.
ExtToken	Xsi:string N12..19	Optional merchant and acquirer agreed token extension for recognizing payment tokens from previous successful payments.
TransactionInfo	element	Transaction info element (used in recurring ca ncel operation present in RecurringOperationR equest only)
OrderId	xsi:string AN1..50	Merchant defined unique order id (of original payment)
TxId	Xsi:long	TxId applicable in StatusRequest messsgae onl y
Operation	xsi:string AN1..25	Predefined String value, Currently supported o peration: Cancel (to cancel recurring occurring)
MasterPassInfo	element	A masterpass extension element if merchant inititated the xml api payment with MasterPass Wallet.
Attribute	element, attr nam e="status"	Element value MasterPass session result status: success, cancel or error
Attribute	element attr name ="txId"	Element value Required if status was success, the masterpass tx id, from masterpass checkout data TransactionId
Attribute	element attr name ="walletId"	Element value Required if status was success, the masterpass wallet id, from masterpass checkout data walletID

Attribute	element attr name ="authMethod"	Element value Required if status was success and masterpass returned authenticated options in chackout data
Responses/ Notification		
VPOS	element	XML root element
Message	element type Message	Message contents element
version	attribute, xsi:string	Message version default value "1.0" Required
messageId	attribute, **xsi:ID**	Message unique identifier (values in request and reply messages this must match, no other purpose)
lang	attribute, xsi:string (2)	Message attribute to specify context language (Optional) (ISO 639-1 language code en, fi, sv, el, etc..)
timeStamp	Attribute xsi:dateTime	Message timestamp when approximate time of when message was created. Example 2015-04-30T12:21:02.402+03:00
Digest (v2.1 only)	element xsi:string	The digest of message element if used instead of password to be calulated Base64(SHA2-256((utf8bytes(canonicalize(Message))+utf8bytes(sharedSecret))
Signature	element ds:SignatureType	The xml signature as defined https://www.w3.org/TR/xmldsig-core/ Canonicalization http://www.w3.org/TR/2001/REC-xml-c14n-20010315 SignatureMethod Algorithm="http://www.w3.org/2001/04/xmldsig-more#rsa-sha256" DigestMethod Algorithm=http://www.w3.org/2001/04/xmlenc#sha256 **Responses are signed by processor private key and validated with Processor certificate (processor certificate is referred to Section 6. page 31)**
Response	element	Element of response type and named as AuthorisationResponse, CaptureResponse, OriginalCreditResponse, RefundResponse, CancelResponse, RecurringOperationResponse
OrderId	xsi:string	Same value as in request message OnrderInfo
OrderAmount	xsi:decimal	Same value as in request message OnrderInfo
Currency	xsi:string	Same value as in request message OnrderInfo
PaymentTotal	xsi:decimal	Actual payment amount normally equals orderAmount or orderAmount + any fees if applicable.

Status	xsi:string	Transaction status in response or notficiation messages AUTHORIZED, CAPTURED - payment was successful (accept order) REFUSED - payment failed, payment was denied for card or by bank (deny order) REFUSEDRISK - payment failed, payment was denied for card by risk score (deny order) CANCELED - only in requrring operation response if supsequent requrrings are set to be canceled ERROR - input, sysrtem or network error (deny order)
TxId	Xsi:long	Server supplied transaction id
Sequence	Xsi:integer	Used with recurrings
PaymentRef	Xsi:string	Remote payment reference like issue approval code.
RiskScore	xsi:integer	Optional risk score calculated by risk scroring subsystem if available
ExtToken	Xsi:string	Optional payment token if tokenization was requested and performed
ExtTokenPanEnd	Xsi:string	Optional payment token related PAN ending 4 numbers
ExtTokenExp	Xsi:date	Optional payment token expiration. (YYYY-MM-DDZ) example 2018-02-01+02:00
ErrorCode	Xsi:string	Error code
Description	Xsi:string	Error or result description text
ReqcurringNotification		
Authentication	element	Authentication element of request Message
Mid	xsi:string (N1..8)	Merchant number/identification in VPOS
OrderId	xsi:string	Same value as in request message OnrderInfo
OrderAmount	xsi:decimal	Same value as in request message OnrderInfo
Currency	xsi:string	Same value as in request message OnrderInfo
PaymentTotal	xsi:decimal	Actual payment amount normally equals orderAmount or orderAmount + any fees if applicable.
Status	xsi:string	Transaction status in response or notficiation messages AUTHORIZED, CAPTURED - payment was successful (accept order) REFUSED - payment failed, payment was denied for card or by bank (deny order) CANCELED - only in requrring operation response if supsequent requrrings are set to be canceled

		ERROR - input, sysrtem or network error (deny order)
TxId	Xsi:long	Server supplied transaction id of recurring master that started requiring sequence
Sequence	Xsi:integer	Recurring sequnece number
SeqTxId	Xsi:long	The recurring seequence transaction server supplied id
PaymentRef	Xsi:string	Remote payment reference like issue approval code.
ErrorCode	Xsi:string	Error code
Description	Xsi:string	Error or result description text
Attribute	Complex element many	
StatusRequest		Query for transaction status
Authentication	element	Authentication element of request Message
Mid	xsi:string	Merchant number/identification in VPOS
TransactionInfo	element	
OrderId	Element Xsi:string	Use either order id or txid to query results if order id used then all transactions referenced are included such as captures, refunds associated
TxId	Element Xsi:long	Use txId to query by txId, only single transaction data is returned
StatusResponse		Response of transaction status containing one or many TransactionDetails
TransactionDetails	element	One or many
OrderId	element	
OrderAmount	Element xs:decimal	Merchant submitted order amount
Currency	Element xs:string	Order currency
PaymentTotal	Element xs:decimal	Final payment amount (order +/- adjustments, fees etc)
Status	Element xs:string	Payment status
TxId	Element xs:long	Transaction identifier
Sequence	Element xs:integer	In case of recurring
PaymentRef	Element xs:string	Payment reference or approval code if available
RiskScore	Element xs:integer	Risk score if available
ErrorCode	Element xs:string	Not used
Description	Element xs:string	Status description
Attribute	Complex element many	Many, rest of the transaction data. As <Attribute name="MERCHANT NO">0000001</Attribute> <Attribute name="USER IP">195.222.10.3</Attribute>

		<Attribute name="CHANNEL">Redirection</Attribute> <Attribute name="3D STATUS">1 - Fully authenticated</Attribute> <Attribute name="SETTLEMENT STATUS">NA</Attribute> <Attribute name="BATCH NO">28</Attribute> <Attribute name="ISO response code">15</Attribute> <Attribute name="ORDER DESCRIPTION" /> <Attribute name="CARD MASK PAN">4016#####0002</Attribute> <Attribute name="ECOM-FLG">5</Attribute> <Attribute name="ECI">05</Attribute> <Attribute name="PAYEREMAIL">demo@modirum.com</Attribute> <Attribute name="PAYERPHONE">+372 123 1234</Attribute> <Attribute name="BILLCOUNTRY">FI</Attribute> <Attribute name="BILLSTATE">Harjumaa</Attribute> <Attribute name="BILLZIP">76543</Attribute> <Attribute name="BILLADDRESS">Billto tn 6-9</Attribute> <Attribute name="SHIPCOUNTRY">FI</Attribute> <Attribute name="SHIPSTATE">Harjumaa</Attribute> <Attribute name="SHIPZIP">12345</Attribute> <Attribute name="SHIPADDRESS">Viru tn 6-9</Attribute> <Attribute name="EXTACQUIRERID">026</Attribute>
TxType	Element xs:string	Transaction type
TxDate	Element xs:dateTime	Transaction execution timestamp
TxStarted	Element xs:dateTime	Transaction started timestamp
TxCompleted	Element xs:dateTime	Transaction completed timestamp
PaymentMethod	Element xs:string	Payment method used.
ErrorMessage	element	Response type of ErroMessage, normally given if request message validation failed or system error.

ErrorCode	Xsi:string	Error code
Description	Xsi:string	Error description text
OriginalXML	Xsi:string	Encoded original XML received in case the error was in XML parsed

Table of field requirements depending on messages:

R - required, O-optional, C-conditional

Field element/ requests	Sale/ AuthorizationRequest	CaptureRequest	OriginalCreditRequest	RefundRequest	CancelRequest	RecurringOperationRequest	SaleResponse	AuthorisationResponse	CaptureResponse	OriginalCreditResponse	RefundResponse	CancelResponse	RecurringOperationResponse	RecurringNotification	Description
Message															
version	R	R	R	R	R	R	R	R	R	R	R	R	R	R	**4.1** or **2.1**
messageId	R	R	R	R	R	R	R	R	R	R	R	R	R	R	Unique value of numbers and or chars **xsi:ID** and matching in request, response messages. max length 128
lang	O	O	O	O	O	O	O	O	O	O	O	O	O	O	Optional iso language code as el, en, ru, fi, et, sv. This is used to set context language in case emails or any other type actions are triggered with this request.
timeStamp	R	R	R	R	R	R	R	R	R	R	R	R	R	R	Required
Authentication															
Mid	R	R	R	R	R	R								R	
OrderInfo	R	R	R	R	R										
DeviceCategory															
OrderId	R	R	R	R	R										
OrderDesc	O		O												
OrderAmount	R	R	R	R	R										
Currency	R	R	R	R	R										
PayerEmail	O														
PayerPhone	O														

AddFraudScore	O													
BlockScore	O													
Var1	O		O											
Var2	O		O											
Var3	O		O											
Var4	O		O											
Var5	O		O											
Var6	O		O											
Var7	O		O											
Var8	O		O											
Var9	O		O											
MOTO	O													
Weight	O													
Dimensions	O													
BillingAddress	R													**Required for 3DSv2 transactions.** Billing address element and sub elements
ShippingAddress	O													optional shipping address element and sub element
PaymentInfo	R	O[1]		O[1]	O[1]									
PayMethod	R	O[1]		O[1]	O[1]									
CardPan	R[2]	O[1]		O[1]	O[1]									Not present if CardEncData present.
CardExpDate	R													Not present if CardEncData present.
CardCvv2	C													Required if not MOTO and required for card type/brand, Not present if CardEncData present.
CardHolderName	O													Optional but highly recommended. Not present if CardEncData present.
CardEncData	C													Used if RSA card encryption then CardPan, CardExpDate, CardHolderName and

															CardCcc2 shal not be present
RecurringIndicator	C														Required for recurring payment
RecurringParameters	C														Required for recurring payment
ExtRecurringfrequency	C														Required for recurring payment
ExtRecurringenddate	C														Required for recurring payment
InstallmentParameters	C														Required for installment payment
ExtInstallmentoffset	C														Required for installment payment
ExtInstallmentperiod	C														Required for installment payment
ThreeDSecure	C														Required for 3D transactions
EnrollmentStatus	C														Required for 3D transactions
AuthenticationStatus	C														Required for 3D transactions
CAVV	C														Required for 3D transactions
XID	C														Required for 3D transactions
ECI	C														Required for 3D transactions
Protocol	C														Required for 3DSv2 transactions
Attribute	C														**TDS2.dsTransID** attribute is required for 3DSv2 transactions
ExtXOrderId	O2		R												O2 – may be present instead of CardPan. Required for original credit to lookup source payment.
ExtTokenOptions	O														
ExtToken	O														
TransactionInfo						R									
OrderId						R									
Operation						R									
Signature	R	R	R	R	R	R	R	R	R	R	R	R	R	R	Required for all (**v4.1**)
Digest	R	R	R	R	R	R	R	R	R	R	R	R	R	R	Required for all (**v2.1**)

Responses/Notification														
OderId							R	R	R	R	R	R	R	R	Order Id supplied by merchant originally
OrderAmount							R	R	R	R	R	R		R	
PaymentTotal							R	R	R	R	R	R		R	
Currency							R	R	R	R	R	R		R	
Status							R	R	R	R	R	R	R	R	Status
TxId							C	C	C	C	C	C		R	In case of transaction processing has started (no rejection due invalid input request), In case of recurring Notificatuion this is master recurring transaction id
Sequence														R	Sequence of recurring in notification
SeqTxId														R	The executed recurring sequence transaction id
PaymentRef							C	C	C	C	C	C		C	Payment reference such as approval code if available
RiskScore							O	O							Optional risk score calculated by risk scoring subsystem if available
ExtToken							O	O							
ExtTokenPanEnd							O	O							
ExtTokenExp							O	O							
ErrorCode							C	C	C	C	C	C	C	C	Error code in case of Status=ERROR
Description							O	O	O	O	O	O	O	O	Optional error description
Attribute							O	O	O	O	O	O	O	O	Optional attributes, may be custom per implementation.
OriginalXML															In general error message only to copy back the error as content received for merchant debugging.
Signature							R	R	R	R	R	R	R	R	Required for all (**v4.1**)
Digest	R	R	R	R	R	R	R	R	R	R	R	R	R	R	Required for all (**v2.1**)

StatusRequest/StaturResponse

Field element/ requests	StatusRequest	TokenizationRequest				StatusResponse	TokenizationResponse					Description
StausRequest												
Authentication												
Mid	R	R										
TransactionInfo	R											
OrderId	C											Either OrderId or TxId is required
TxId	C											Either OrderId or TxId is required
StatusResponse						R						
TransactionDatails						R						
OrderId						R						
OrderAmount						R						
Currency						R						
PaymentTotal						R						
Status						R						
TxId						R						
Sequence						O						
PaymentRef						O						
RiskScore						O						
Description						O						
Attribute						O						List of attributes depending on what information is available. Attribute name can be one of the following: MERCHANT NO - merchant number, REFUNDED AMOUNT - amount refunded if available,

											USER IP - use ip if available, CHANNEL - channle originated 3D STATUS - status CAPTURED AMOUNT - captured amt SETTLEMENT FILE - settl file name BATCH NO - batch number ISO response code - iso response if available ExtData – additional data from external payment systems if available ORDER DESCRIPTION - order descr CARD MASK PAN - masked pan 5+3 or 4+4 or 6+2 INSTALLMENT SEQUENCE, INSTALLMENT PERIOD, INSTALLMENT OFFSET, RECURRING SEQUENCE, RECURRING END DATE, RECURRING FREQUENCY, ECOM-FLG - ecom flag in auth message, ECI - eci from mpi, VAR1..VAR9, PAYEREMAIL, PAYERPHONE, BILLCOUNTRY, BILLSTATE, BILLZIP, BILLADDRESS SHIPCOUNTRY, CancelRequest RecurringOperationRequest, StatusRequest, TokenizationRequest,

												SHIPSTATE SHIPZIP SHIPADDRESS BONUS PARTICIPATION *, BONUS REF* BONUS ADJUSTMENT* BONUS STATUS* BONUS DETAILS* RETURNING USED** RETURNING ORDER ID** * - Only possible if with special bonus loyalty extension. ** - Only possible if with returning customer extesnion.
TxType						R						
TxDate						R						Transaction exec date
TxStarted						R						Transaction started
TxCompleted						O						May bemissing if transaction did not complete due errors.
PaymentMethod						O						
CardInfo	R	R										CardInfo type ref required, if CSE then encData reuired else pan,exp reuired
Attribute:ref		O										unique number request id within merchant scope
Attribute: pan		O										Card pan
Attribute: exp		O										Card expiry date
Attribute: chn		O										Cardholder name
Attribute: encData		O										CSE
TokenInfo							R					Token Information element (one or more) included in TokenizationResponse
Attribute: ref							R					unique number request id within merchant scope used only in

												return file to match request
Attribute: status							R					Status of tokenization (OK if no errors occured)
Attribute: tokenValue							O					Value of token
Attribute: panEnd							O					Last 4 digits of pan
Attribute: exp							O					Expiration Date

O^1 - if supported feature then fields may not need to be present if not supported then the fields are required. Availability of this option shall confirm with system administrator (Your customer support). If values not sent, then whole PaymentInfo element shall be excluded from message.

R^2 and O^2 - If system supports and merchant is set tp participate in returning customer recognition extension then if merchant already has a successful order with a card with this customer and the card is still valid and customer chooses to make this next order with same card and the days and amounts between orders are in certain limits then merchant may send ExtXOrderId instead of CardPan. In such case if validations are passed system automatically uses pan from previous specified order. Recommended maximum period between previous order and next returning customer extension order could be 6 months (180 days).

Currently supported operations:

AuthorisationRequest-make a pre-authorization
CaptureRequest- capture a pre-authorization
RefundRequest- make refund
SaleRequest- make a payment
CancelRequest- make reversal for an unsettled transaction
RecurringOperationRequest- with operation Cancel, cancel recurring master scheduling
RecurringNotification – Optional message posted to merchant if a recurring child is executed on server, merchant does not need to send response XML to this on accept merchant server should respond with http status code 200/OK or in case merchant does not recognize the transaction 406/Not Acceptable or 400/Bad Request if the message format is invalid. Server just acknowledges the response code and performs no additional actions based on merchant response code.
StatusRequest- query transaction status
TokenizationRequest- tokenize a card to token

Error code values:
Filled in case status is ERROR with following values
M1 – Invalid merchant id
M2 – Authentication failed (wrong password or digest or signature)
SE – System error (message contains error id, system or configuration error to be investigated)
XE – Invalid XML request not parseable or does not validate
I0 – Invalid or unsupported request
I1 – Message contains invalid data item or missing required item
I2 – Message contains invalid installment parameters
I3 – Message contains invalid recurring parameters

I4 - Message contains invalid or mismatching card data
I5 - Message contains invalid expiration date card data
I6 – Selected payment method does is not supported or not matching the payment card
O1 – Operation is not allowed because logic is violated or wrong amounts
O2 – Original transaction is not found to perform operation.
May be also filled in case of status is REFUSED with acquirer network supplied ISO response code

3. Digest calculation with XML API 2.1

At VPOS side there are both validations implemented if the Digest values is present then VPOS validates the authentication
of message using the digest and merchant shared secret.

Version 2.1
Base64(SHA256((utf8bytes(canonicalize(Message))+utf8bytes(sharedSecret)),
to be used only if the XML password is not used.
The canonicalization method to be used is
http://www.w3.org/TR/2001/REC-xml-c14n-20010315

Note that the XML documents should be handled with namespace aware xml libraries (parser/serializer).
When the Message element is serialized and canonicalized it should contain xmlns namespace attribute.
See from next section XML message with digest example.

Note for XML API with Three D Secure:
This is 2 step processing at first step merchant should implement MPI plugin session as decribed in Modirum MPI manual and obtain the Three D Secure authentication results from there and then next step is to fill the corresponding values to XML API ThreeDSecure element and proceed with XML api request to VPOS.

XML API plugin example message and digest

Secret=SecRetDigest1

<?xml version="1.0" encoding="UTF-8"
standalone="yes"?><VPOS xmlns="http://www.modirum.com/schemas/vposxmlapi41"
xmlns:ns2="http://www.w3.org/2000/09/xmldsig#"><**Message version="2.1" messageId="M1560776758348" timeStamp="2019-06-17T16:05:58.348+03:00"><SaleRequest><Authentication><Mid>0000001</Mid></Authentication><OrderInfo><OrderId>1560776271083</OrderId><OrderDesc>Test</OrderDesc><OrderAmount>1.25</OrderAmount><Currency>EUR</Currency><PayerEmail></PayerEmail></OrderInfo><PaymentInfo><PayMethod>visa</PayMethod><CardPan>4016000000002</CardPan><CardExpDate>2206</CardExpDate><CardCvv2>756</CardCvv2><CardHolderName>John Smith</CardHolderName></PaymentInfo></SaleRequest></Message>**<Digest>xmSXBhrE99FqiP2b73S0cS+oLrIi8+lng9IS9KmoWpM=</Digest></VPOS>

Message part canonicalized note xmlns added:

<Message **xmlns="http://www.modirum.com/schemas/vposxmlapi41" xmlns:ns2="http://www.w3.org/2000/09/xmldsig#"** messageId="M1560776758348" timeStamp="2019-06-17T16:05:58.348+03:00" version="2.1"><SaleRequest><Authentication><Mid>0000001</Mid></Authentication><OrderInfo><OrderId>1560776271083</OrderId><OrderDesc>Test</OrderDesc><OrderAmount>1.25</OrderAmount><Currency>EUR</Currency><PayerEmail></PayerEmail></OrderInfo><PaymentInfo><PayMethod>visa</PayMethod><CardPan>4016000000002</CardPan><CardExpDate>2206</CardExpDate><CardCvv2>756</CardCvv2><CardHolderName>John Smith</CardHolderName></PaymentInfo></SaleRequest></Message>SecRetDigest1

Then append SecRetDigest1 and apply sha2-256 function.
You will get digest
<Digest>xmSXBhrE99FqiP2b73S0cS+oLrIi8+lng9IS9KmoWpM=</Digest>

Response example:

<?xml version="1.0" encoding="UTF-8" standalone="yes"?><VPOS xmlns="http://www.modirum.com/schemas/vposxmlapi41" xmlns:ns2="http://www.w3.org/2000/09/xmldsig#"><Message version="2.1" messageId="M1560776758348" timeStamp="2019-06-17T16:05:58.517+03:00"><SaleResponse><OrderId>1560776271083</OrderId><OrderAmount>1.25</OrderAmount><Currency>EUR</Currency><PaymentTotal>1.25</PaymentTotal><Status>CAPTURED</Status><TxId>927703881</TxId><PaymentRef>104040</PaymentRef><RiskScore>10</RiskScore><Description>OK, CAPTURED response code 00</Description><Attribute name="EXTACQUIRERID">014</Attribute></SaleResponse></Message><Digest>oavTfZECv1L8hKcjw0mV+bOvIjSdq+UNSNU7/xRvnAA=</Digest></VPOS>

4. Signature calculation with XML API V4.1

Signatures shall be calculated and verified according to documentation
https://www.w3.org/TR/xmldsig-core/
Canonicalization method to be used is http://www.w3.org/TR/2001/REC-xml-c14n-20010315
SignatureMethod Algorithm="http://www.w3.org/2001/04/xmldsig-more#rsa-sha256”
DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256"
The signed element is Message element referenced with its **ID** attribute named **messageId**.
ID attribute is an attribute which type in schema is defined as xsd:ID.

Messages sent by merchant are signed by merchant private key and verified with merchant certificate.
Messages sent by VPOS service are signed by service provider private key and validated with service provider provided certificate.

XML API plugin example message and signature calculation

Here is an example request message to VPOS and how the signature is calculated.
(used apache santuario)

Merchant Private key PKCS8:
-----BEGIN PRIVATE KEY-----
MIIEvgIBADANBgkqhkiG9w0BAQEFAASCBKgwggSkAgEAAoIBAQDaX7Jd16os2Mti
cXHXGjanQ3fDSwwoRRhVWi12+SiFDMVBpBwZEGdmHopO5cpSGptFxeau7HqGfSaq
5NoI01pbf/OPFpstO4mSlIBj2OO9wzcW2yNeAQjzycEQmgNr1UQACUmXsNzBZZ2m
rcddkdRpxfHPaZx+GIYMdemFY7G0yBXsG0Dq+3hi9kqyGYlAN3PFsqCEdwD3H8qd
5UKz4wKEYhuqhKBZZoGBBUQZt7X9plwdMoZhtqbJIJTpda5Og/yNxkSjiTQrOMnt
vSl5dAQ8dGxoFaKAdvaE09eqt0F6RI76qyUU3B0PKBVB/kIYhvFSvJtef6a8fF4S
y56VOMptAgMBAAECggEBAM9tj1Qsg21OEQNVlzknoTqIj75mDwpBd7e7jOwyCBc5
5jVP2ZDFUDJkWCRRijkrJMrGDTWjU09kmdJCyAkSGgZIJ+aHJqd0ol0lyj8NymZ6
hF2lkpa8jPBleIp4gT9wuMMAD3OTgF4EVBf7giCTYR2H9QV74Da2vL4hUsxtwmNg
2jQjjHTsVA/ESjiyGveh1X6+GV6CsTZsoAWLlOhuDHiOMuOXDBmn9JjArFsl2W4X
yrtrDx68nVdPdIH2LzIrBzqRG6tB9RpNQNWGs/IxuEUG07fLMGzQiureOTUm/ybt
ZrO9Ab59tzWXCFXHljsGJu9SnZuPNOT0L8PuJIxKOIECgYEA9w6hdFaVr0HMnQtX
ndtZQfiqNnQMymV0mR9gtyw20/krOW5yt7WqhrzzTB72m4bsm27Yz3Dn0jfhQ1h5
zyihrT+FGeF6jS6+Hr3FXFyMizxH9AZPl13UmZo1fKxeoL+sE5PppFE9Qlsz0TBp
2phlVjzLI7i3KOu8Hyzt/rafZDkCgYEA4kdFMSHTQGLounpPauKaVi8v9TjyFdST
qSuQ0pMG4R9xuZ0x52L081goYmxo4jDo7P+m3iHDFdJqg+D7aAVay4Hv0PGKIq8G
vOAXm6mnXBaIMDVMnTRtqRynDoo2qKp9UU2Sv4D0L6Zbm9axDxMvqXCa8Lz5Kbnh
zJufUAwzn9UCgYEAkboGkDn2Zv8X81ZaYxmcZ6aGuEHxvXzkruFsSf+Bg71IusKk
ViqJIJrZo//rlMecTv6uUoYVp9EgRXott30PCMMb/q0afaahrD5h6N4KZKK1CoKi
dfV5zvTAMf72fjkxBgdMXIky6i4jvXOiLLeRprGLXVG6cB/EwlrdM06DbDkCgYBc
TdJt3mx8gVyKZUZsRY/LxGf90oL+YL7zbXAgVhWiU99iZjtrNjTR545hx/NpAaai
tw7s4jzgc/s7XNVxc228Qn7/buh4iYloFsnKmARLTm2zrKpaHn71U1jaV4tAdnu0
ZL6OHB6AKY6JHaUQjzUMG4E43v2NBeSUQI9WagPNGQKBgDj5qk4Jauy8zg/IBkXD
eJsgwGrMH7o1vj2Uhcd2K2NrxO3qRaJitNXH+cso836/Ez///kdepX3hQ3gKZS7i
aGhDFF3r0LU2OmskhoDSyhzVlCgsXbW1skFwL3Y161uYHwgpkFqrAODONXLu3PBd
S8jJbKkA3lQnmCCbET3NLfIV
-----END PRIVATE KEY-----

Merchant certificate X509:
-----BEGIN CERTIFICATE-----
MIIDuzCCAqOgAwIBAgIJANh5ptk5BWu5MA0GCSqGSIb3DQEBCwUAMHQxCzAJBgNV
BAYTAkVFMREwDwYDVQQIDAhNeSBTdGF0ZTEQMA4GA1UEBwwHbXkgQ2l0eTEVMBMG
A1UECgwMQ29tcGFueSBOYW1lMRAwDgYDVQQLDAc3NzExMjIzMRcwFQYDVQQDDA53
d3cubXlzaXRlLmNvbTAeFw0xNzAzMjkxNzM3MDFaFw0yMTAzMjgxNzM3MDFaMHQx
CzAJBgNVBAYTAkVFMREwDwYDVQQIDAhNeSBTdGF0ZTEQMA4GA1UEBwwHbXkgQ2l0
eTEVMBMGA1UECgwMQ29tcGFueSBOYW1lMRAwDgYDVQQLDAc3NzExMjIzMRcwFQYD
VQQDDA53d3cubXlzaXRlLmNvbTCCASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoC
ggEBANpfsl3XqizYy2JxcdcaNqdDd8NLDChFGFVaLXb5KIUMxUGkHBkQZ2Yeik7l
ylIam0XF5q7seoZ9Jqrk2gjTWlt/848Wmy07iZKUgGPY473DNxbbI14BCPPJwRCa
A2vVRAAJSZew3MFlnaatx12R1GnF8c9pnH4Yhgx16YVjsbTIFewbQOr7eGL2SrIZ
iUA3c8WyoIR3APcfyp3lQrPjAoRiG6qEoFlmgYEFRBm3tf2mXB0yhmG2pskglOl1
rk6D/I3GRKOJNCs4ye29KXl0BDx0bGgVooB29oTT16q3QXpEjvqrJRTcHQ8oFUH+
QhiG8VK8m15/prx8XhLLnpU4ym0CAwEAAaNQME4wHQYDVR0OBBYEFJaXNDk3UIJT
7bjuedk13vmz62RjMB8GA1UdIwQYMBaAFJaXNDk3UIJT7bjuedk13vmz62RjMAwG
A1UdEwQFMAMBAf8wDQYJKoZIhvcNAQELBQADggEBAJx7UBdBddBbJ8sz/Fa3YvDl
VR/GNTLp/haKC6G+FA97H5u2S7OGgXUnIX2T3M94QllhTykkzfr1zJeDZD+YrYyh
Ayp/ykHL0gk0tumHw8DN1BRmglRMc4QEXXHsx1HnMlcS0uE622M2+IQeDzDtLYpf
XL36Dqoik0hIuNSjlxqlIX4kBweA83Xx9IGyhsMhXHSS0BcPVmup97PTAs81YGOu
7vVgzyLBTHjabRktd0hVdm9+EJ/RMMFTW4XM+Ue2ekFx3uEX2B53ND6Mx5mtP/pi
bQ7/860FXUNdrHbcQCFufqhk7Ikr3+kv+Rqmh5DmrUbblpmXFvm6iLc6uYZqIvE=
-----END CERTIFICATE-----

Service provider certificate:
-----BEGIN CERTIFICATE-----
MIID5TCCAo0CBFjeXq8wDQYJKoZIhvcNAQELBQAwdzEoMCYGA1UEAxMfVlBPUyBERU1PIHZwb3Nh
ZG1pbi5tb2RpcnVtLmNvbTENMAsGA1UECxMEVlBPUzEQMA4GA1UEChMHTW9kaXJ1bTEQMA4GA1UE
BxMHVGFsbGlubjELMAkGA1UECBMCSE0xCzAJBgNVBAYTAkVFMB4XDTE3MDMzMTEzNTAzOVoXDTIy
MDkyMTEzNTAzOVowdzEoMCYGA1UEAxMfVlBPUyBERU1PIHZwb3NhZG1pbi5tb2RpcnVtLmNvbTEN
MAsGA1UECxMEVlBPUzEQMA4GA1UEChMHTW9kaXJ1bTEQMA4GA1UEBxMHVGFsbGlubjELMAkGA1UE
CBMCSE0xCzAJBgNVBAYTAkVFMIIBYjANBgkqhkiG9w0BAQEFAAOCAU8AMIIBSgKCAUEAyhFCdFGD
pchDXC7ryDUiMOlRHjce4N9e4hNUZ6+hTshRBTNeHqcTfhxKuiReaC6AVbQEbBYBGCUs8EQAWppK
RIB+ZnTytY8bhJqQ1YuiWvAN5cTBLoS2jE5vxf/Xx/+G+UhjfmK6XM0UKnQ4mR+MKM5/iSgV/Un7
ysHoLLepwefEUBQEODqAIsc6N5pMeeShT/66WEtxEkiXQPn48PXDRLLzSBzB247w03r+92WWrlVe
IMgTQc0kgx2gsgMziiqiUDSB69Bm/ugT81wDcUNklmbo8r3IsxtjOT+/HQ8Qbo4vQpJI7yzIcnvt
6U8Ub5TLjz4UmIBg8y6lY/kbJoxA/4n/M+1MwZqgM7cKGi5lG429A3h/1g2zhQ8bZBexnY5FLW1G
PTCS4ahE67ZYl8CWXjoDAzFtVcdpMDFnvZ6noMkCAwEAATANBgkqhkiG9w0BAQsFAAOCAUEAp0mN
/2Ml6tVC8Zi0bkXJ8j+bUxaxCUU1nV7htzWOqlAsQn1mVb7lbkLZgOc7RfD5CxdLspAVIVU1Gekp
/tSLjbdA3obSlBFmIm5yU4PGN9YjLRi5jbAAJNhJYThFB0YJu4M6tqX0nbxX6GphPeh2ruQ6WzeS
KwUf62gqd96WZeIwAKLoAZng4G9LZNITL7jUgl4OWq9OzZ+JYpe/rSz1tKWAg9r5U/AEkoZasfPo
3MLQlNCTh/WQm8jmtsyglct4k5SNI3ABhFcPfcR0PIhCjTVd7vlY8NcdaxSYYRzQgKZ7N8pdhvi3
NyPZmbu4OJXkc4Fupuyp2YxhGh0AtLKvdPRmybNZCmTRejgGbJeE6LjkcJ2zcunb+LxbyoxJ1DdU
K1tddzVPdH+QK8q3EKBNt0H3KwbRPk9qRmH4xuoX4XA=
-----END CERTIFICATE-----

Example code:

```
import javax.xml.transform.Transformer;
import javax.xml.transform.TransformerFactory;
import javax.xml.transform.dom.DOMSource;
import javax.xml.transform.stream.StreamResult;
import javax.xml.transform.stream.StreamSource;

import org.apache.xml.security.keys.KeyInfo;
import org.apache.xml.security.keys.content.X509Data;
import org.apache.xml.security.keys.content.x509.XMLX509Certificate;
import org.apache.xml.security.signature.XMLSignature;

public class Signer
{
	public byte[] sign(VPOS root, PrivateKey prik, java.security.cert.X509Certificate[] crts) throws
Exception
	{
		org.w3c.dom.Document dom = apis.marschalToDOM(root);
		// apis.normalizeDOM(dom); dom nomralization is very slow using instead
		// msg.setIdAttribute("messageId", true);
		Element vpos = dom.getDocumentElement();
		XMLSignature xmlsigAp = new XMLSignature(dom, null,
				"http://www.w3.org/2001/04/xmldsig-more#rsa-sha256",
				"http://www.w3.org/TR/2001/REC-xml-c14n-20010315");

		Element sigel = xmlsigAp.getElement();
		vpos.appendChild(sigel);

		Element msg = (Element)vpos.getFirstChild();
		// setting id attribute instead of dom normalization
		msg.setIdAttribute("messageId", true);
		xmlsigAp.addDocument("#" + msg.getAttribute("messageId"), null,
			"http://www.w3.org/2001/04/xmlenc#sha256", null, null);

		for (int i = 0; crts != null && i < crts.length; i++)
		{
			xmlsigAp.addKeyInfo(crts[i]);
		}
		xmlsigAp.sign(prik);
		ByteArrayOutputStream bos = new ByteArrayOutputStream(4096);
		TransformerFactory transfac = TransformerFactory.newInstance();
		Transformer trans = transfac.newTransformer();
		trans.setOutputProperty(OutputKeys.OMIT_XML_DECLARATION, "no");
		trans.setOutputProperty(OutputKeys.INDENT, "no");
		trans.setOutputProperty(OutputKeys.ENCODING, "utf-8");

		DOMSource source = new DOMSource(dom);
```

```
                trans.transform(source, new StreamResult(bos));
                return bos.toByteArray();
        }
}
```

Example sale request (assume there is no line breaks until end of <Message> part)

<VPOS xmlns="http://www.modirum.com/schemas/vposxmlapi41"
xmlns:ns2="http://www.w3.org/2000/09/xmldsig#">**<Message messageId="M1560776270228" timeSt
amp="2019-06-17T15:57:50.228+03:00"
version="4.1"><SaleRequest><Authentication><Mid>0000001</Mid></Authentication><OrderInfo><
OrderId>1560776235400</OrderId><OrderDesc>Test</OrderDesc><OrderAmount>1.25</OrderAmou
nt><Currency>EUR</Currency><PayerEmail/></OrderInfo><PaymentInfo><PayMethod>visa</PayMet
hod><CardPan>4016000000002</CardPan><CardExpDate>2206</CardExpDate><CardCvv2>756</Card
Cvv2><CardHolderName>John
Smith</CardHolderName></PaymentInfo></SaleRequest></Message>**<ds:Signature xmlns:ds="http://
www.w3.org/2000/09/xmldsig#">
<ds:SignedInfo>
<ds:CanonicalizationMethod Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315"/>
<ds:SignatureMethod Algorithm="http://www.w3.org/2001/04/xmldsig-more#rsa-sha256"/>
<ds:Reference URI="#M1560776270228">
<ds:DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256"/>
<ds:DigestValue>82t/HCbRKUrAKVsA1tOpU8zXi3wIupTUeBndZ90VALM=</ds:DigestValue>
</ds:Reference>
</ds:SignedInfo>
<ds:SignatureValue>
DhADR21OEzIikjwgZh61pibBULtI0iRbkSEt6z2mdVGpQRgI3UFIepkYvTeNZv84cF2jM6JCrFbx
dXMIRQ643rFXwOAnstv0QyRFPD4XCQDltSfoqDNfjAQE2wXmYWgHGJdI/0Vu12TJ64XzdEhb4E6t
8yGfyYL6DdXZk4oBRZxBRqGBA6zxyDRdRvLq9V+LGIwZk4J7p6M+wZWDTb50/pOSU2wlP/s4IPtQ
vZQYWct9Huq/sFI+qwAG7na0L25zE9cB467lcaKmgGGLXFrRwDX6xAmoZOwFIW5x0CXbtM2X2j8v
H53/Hfh1rdsWRxbOs7+ObLYvct/BA6KRbMxBPA==
</ds:SignatureValue>
<ds:KeyInfo>
<ds:X509Data>
<ds:X509Certificate>
MIIDuzCCAqOgAwIBAgIJANh5ptk5BWu5MA0GCSqGSIb3DQEBCwUAMHQxCzAJBgNVBAYTAkVFMREw
DwYDVQQIDAhNeSBTdGF0ZTEQMA4GA1UEBwwHbXkgQ2l0eTEVMBMGA1UECgwMQ29tcGFueSBOYW1l

MRAwDgYDVQQLDAc3NzExMjIzMRcwFQYDVQQDDA53d3cubXlzaXRlLmNvbTAeFw0xNzAzMjkxNzM3
MDFaFw0yMTAzMjgxNzM3MDFaMHQxCzAJBgNVBAYTAkVFMREwDwYDVQQIDAhNeSBTdGF0ZTEQMA4
G
A1UEBwwHbXkgQ2l0eTEVMBMGA1UECgwMQ29tcGFueSBOYW1lMRAwDgYDVQQLDAc3NzExMjIzMRcw

FQYDVQQDDA53d3cubXlzaXRlLmNvbTCCASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBANpf
sl3XqizYy2JxcdcaNqdDd8NLDChFGFVaLXb5KIUMxUGkHBkQZ2Yeik7lylIam0XF5q7seoZ9Jqrk
2gjTWlt/848Wmy07iZKUgGPY473DNxbbI14BCPPJwRCaA2vVRAAJSZew3MFlnaatx12R1GnF8c9p
nH4Yhgx16YVjsbTIFewbQOr7eGL2SrIZiUA3c8WyoIR3APcfyp3lQrPjAoRiG6qEoFlmgYEFRBm3

tf2mXB0yhmG2pskglOl1rk6D/I3GRKOJNCs4ye29KXl0BDx0bGgVooB29oTT16q3QXpEjvqrJRTc
HQ8oFUH+QhiG8VK8m15/prx8XhLLnpU4ym0CAwEAAaNQME4wHQYDVR0OBBYEFJaXNDk3UIJT7bju
edk13vmz62RjMB8GA1UdIwQYMBaAFJaXNDk3UIJT7bjuedk13vmz62RjMAwGA1UdEwQFMAMBAf8w
DQYJKoZIhvcNAQELBQADggEBAJx7UBdBddBbJ8sz/Fa3YvDlVR/GNTLp/haKC6G+FA97H5u2S7OG
gXUnIX2T3M94QllhTykkzfr1zJeDZD+YrYyhAyp/ykHL0gk0tumHw8DN1BRmglRMc4QEXXHsx1Hn
MlcS0uE622M2+IQeDzDtLYpfXL36Dqoik0hIuNSjlxqlIX4kBweA83Xx9IGyhsMhXHSS0BcPVmup
97PTAs81YGOu7vVgzyLBTHjabRktd0hVdm9+EJ/RMMFTW4XM+Ue2ekFx3uEX2B53ND6Mx5mtP/pi
bQ7/860FXUNdrHbcQCFufqhk7Ikr3+kv+Rqmh5DmrUbblpmXFvm6iLc6uYZqIvE=
</ds:X509Certificate>
</ds:X509Data>
</ds:KeyInfo>
</ds:Signature></VPOS

Response signed by service provider (assume no line breaks until end of <Message>:

<VPOS xmlns="http://www.modirum.com/schemas/vposxmlapi41"
xmlns:ns2="http://www.w3.org/2000/09/xmldsig#"><Message messageId="M1560776270228" timeSta
mp="2019-06-17T15:57:50.502+03:00"
version="4.1"><SaleResponse><OrderId>1560776235400</OrderId><OrderAmount>1.25</OrderAmou
nt><Currency>EUR</Currency><PaymentTotal>1.25</PaymentTotal><Status>CAPTURED</Status><TxId
>927703821</TxId><PaymentRef>104037</PaymentRef><RiskScore>10</RiskScore><Description>OK,
CAPTURED response code 00</Description><Attribute
name="EXTACQUIRERID">014</Attribute></SaleResponse></Message><ds:Signature xmlns:ds="http://
www.w3.org/2000/09/xmldsig#">
<ds:SignedInfo>
<ds:CanonicalizationMethod Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315"/>
<ds:SignatureMethod Algorithm="http://www.w3.org/2001/04/xmldsig-more#rsa-sha256"/>
<ds:Reference URI="#M1560776270228">
<ds:DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256"/>
<ds:DigestValue>6nt7AHK5fyrhVW/Mdp9Slx/NBHMfekjbfThFVBRKkt8=</ds:DigestValue>
</ds:Reference>
</ds:SignedInfo>
<ds:SignatureValue>
Wjb1yBQzPok9VKu9U37ua3i/OsqcMZQKAvyE6iOML43rteMgorpmwlOWQSQvLHqFQts4HVxvMkru
Dufn7wuRfqmjDWLzgUqHpFTz+heOGDXhc88ovCaE7vFeYDJg+/isHjaO29ETe6+NH8oDvq4/no00
mA/eHWqNB+vH51+jQCZfRI+tavz1iPAFLAF9Sl5IitaiuGXkEOoOxMbZ7FAb8GT++1MuZYDFgWLh
Z/skR57b/LobPY5n5+AkEdqc86Dyk8/zOJC6RRS9TuJWoAIgJOaVNulSB6X/lsmfu7+GDDEynqxo
bZ0djEMwXhfLSfNlNHHqkePKxEhlXMFkEL5B1jGTnHs26yymy9JYq6TtwUq9XjEn2XnYl0Oa9hwC
FIJ8a5p8u0nPJqtWNJKqDD1YH7FSEc7cBbM8SoTjXAyLZssZmBvJ+bba+FyIl5wTeD2RKtPenptu
3uoyyL60c+ZeGs9+N3sfWh2jpztcSAj4xLQEre59UvFE478Kw78MfF0k
</ds:SignatureValue>
<ds:KeyInfo>
<ds:X509Data>
<ds:X509Certificate>
MIIEXjCCAsYCAQEwDQYJKoZIhvcNAQELBQAwdTElMCMGA1UEAxMcQ2FyZGxpbmsgVUFUIFNpZ25p
bmcgYW5kIENTRTENMAsGA1UECxMERUNPTTERMA8GA1UEChMIQ2FyZGxpbmsxDzANBgNVBAcTBkF0

aGVuczEMMAoGA1UECBMDQVRIMQswCQYDVQQGEwJHUjAeFw0xODA2MjEyMTAwMDBaFw0yNTA2Mj
ly
MDU5NTlaMHUxJTAjBgNVBAMTHENhcmRsaW5rIFVBVCBTaWduaW5nIGFuZCBDU0UxDTALBgNVBAsT
BEVDT00xETAPBgNVBAoTCENhcmRsaW5rMQ8wDQYDVQQHEwZBdGhlbnMxDDAKBgNVBAgTA0FUSDEL
MAkGA1UEBhMCR1IwggGiMA0GCSqGSIb3DQEBAQUAA4IBjwAwggGKAoIBgQDlZIj4eMY2hU7ot4kk
gB1e7xJniAe07ntRVwPZdJ1cxevLvSoQMvgd8070RrT7cPDXp6iJll0RKBnCwZspwoO5evUngdfo
AleyLSVUXljKp2G/e6Kt22RMCLtYsqNv4qFW5nW8XwB88wvqziSMPu9Mo1gGhOxWpS4Viy3NvrtE
VOWXvssx+ZLPolb3AW93w7BOfzEpt7LM3GwrSYZuPoPHcwdkBs0nF+htIEOq/2T7GDcZPNIUmllu
4nQt6u7T1SJ0/TpdHta/p55xptE7QLZlNdphIxvu4Zc9U7mwvlCN8MqMNQnQSFlqnBdOgtQ5gxfE
8x/cSWOVLzTh6dWOc2o7aiAhk8sVopl7N4jeL4U4Nvp0GyDodoWgUJeweDookIb9DL2fgQeBLKn8
ZFDPOyoBQSNr8AAm3p0bgTDY4XkTuav919LGgCjR5k389CW256zXCgsj5Dnn8gcTrf0mwziUbjlG
t/UIy7CA7kmpELwna4NNo7Lt6laILqletJi1rlECAwEAATANBgkqhkiG9w0BAQsFAAOCAYEAVkOF
bVwxj/pbnTH8Z2y/17P1yzv4H6vKB2RdG60CMSou0X/WNybBgaMSf6qJJs3osUC68qx27Q3pYp4i
7onsTlNedhSsUVZVabRHXkjLxGLx9saZNiZ9turIyxzfC7VdeGaogvmcFPZAFgkGSFy4tAZz8fIk
L7XI9pp5NTrjP9AL1ETVgwoHFKoeEKU1ewgQGRXpsM2sQnanMrTOgfVWz+qmaMmCcgeuQnYDPkZX
X3jo456N0IDcGhJRmzkO8x0ge3DGyTc2mdS+38c61VEDd2TQHDHJuGsjCSVMjYh83JF7Ut3imFYh
v3jgmHNkEDsp7XU81UMaV1nD0WzwNTbuMlyuvUQltLtQ0lciDl+yT7zciHZr3JkL3am9lCtny/DR
Oyw7pZnDCbWHaUKl4pV5UtwCIT/o5v7yo3av1z5o6UfiaI+kemeyhcU7PtMXZ6mgW9Hcq4htX1BT
l/LsTN/42XxvrdzystkmvJeSlrNLPbeASi8MC3j/xQdUjc6mWQ/t
</ds:X509Certificate>
</ds:X509Data>
</ds:KeyInfo>
</ds:Signature></VPOS>

5. Examples how to generate merchant keys

With openssl

It's just possible to do all in one line:
openssl req -x509 -newkey rsa:2048 -sha256 -keyout merchantkey.pem -out merchantcert.pem -days 1460 -subj "/C=EE/ST=My State/L=my City/O=Company Name/OU=7711223/CN=www.mysite.com"

The output file **merchantcert.pem** need to be sent to service provider to install with Your merchant account so Your messages will be validated with public key in Your certificate.
C – is two letter country code
L – locality eg. city where you are located.
OU - is recommended to fill with Your merchant number with service provider.
O - shall be your company full or public name.
CN – is recommended (not required as with server certificates) to be your website name

rsa:keysize is recommended to be 2048 or 3072 bits for foreseeable future and validity days up to 1460 days (4 years), ask service provider if it has specific policy or requirements.
Use necessary measures to protect your private key in generated file merchantkey.pem.
Converting private key to PKCS8 format handleable by java:
openssl pkcs8 -topk8 -in merchantkey.pem -inform PEM -outform PEM -out merchantkey-p8.pem -nocrypt

With java keytool

With java keytool private key remains in keystore and cannot be extracted unless special software is used. So Your software shall operate directly with this keystore then.
keytool -genkey -keyalg **RSA** -sigalg **SHA256withRSA** -
dname "CN=www.mysite.com,OU=7711223,O=Company Name,L=my City,S=My State,C=EE" -
keysize 2048 -validity 1460 -alias mykey2017 -storetype **JCEKS** -keystore mykeystore.jceks -
keypass strongPassKey -keystore mycerts.jceks -storepass strongPass
Now export Your certificate to a file that can be sent to service provider:
keytool -exportcert -alias mykey2017 -file merchantcert.pem.cer -storetype JCEKS -
keystore mycerts.jceks -storepass strongPass -rfc

6. Processor Certificate

Processor certificate is used by merchant to calculate the signature value for the response messages.

For testing purposes, merchant can use the following processor certificate:

-----BEGIN CERTIFICATE-----
MIIEXjCCAsYCAQEwDQYJKoZIhvcNAQELBQAwdTElMCMGA1UEAxMcQ2FyZGxpbmsgVUFUIFNpZ25p
bmcgYW5kIENTRTENMAsGA1UECxMERUNPTTERMA8GA1UEChMIQ2FyZGxpbmsxDzANBgNVBAcTBkF0
aGVuczEMMAoGA1UECBMDQVRIMQswCQYDVQQGEwJHUjAeFw0xODA2MjEyMTAwMDBaFw0yNTA2MjIy
MDU5NTlaMHUxJTAjBgNVBAMTHENhcmRsaW5rIFVBVCBTaWduaW5nIGFuZCBDU0UxDTALBgNVBAsT
BEVDT00xETAPBgNVBAoTCENhcmRsaW5rMQ8wDQYDVQQHEwZBdGhlbnMxDDAKBgNVBAgTA0FUSDEL
MAkGA1UEBhMCR1IwggGiMA0GCSqGSIb3DQEBAQUAA4IBjwAwggGKAoIBgQDlZIj4eMY2hU7ot4kk
gB1e7xJniAe07ntRVwPZdJ1cxevLvSoQMvgd8070RrT7cPDXp6iJll0RKBnCwZspwoO5evUngdfo
AleyLSVUXljKp2G/e6Kt22RMCLtYsqNv4qFW5nW8XwB88wvqziSMPu9Mo1gGhOxWpS4Viy3NvrtE
VOWXvssx+ZLPolb3AW93w7BOfzEpt7LM3GwrSYZuPoPHcwdkBs0nF+htIEOq/2T7GDcZPNIUmllu
4nQt6u7T1SJ0/TpdHta/p55xptE7QLZlNdphIxvu4Zc9U7mwvlCN8MqMNQnQSFlqnBdOgtQ5gxfE
8x/cSWOVLzTh6dWOc2o7aiAhk8sVopl7N4jeL4U4Nvp0GyDodoWgUJeweDookIb9DL2fgQeBLKn8
ZFDPOyoBQSNr8AAm3p0bgTDY4XkTuav919LGgCjR5k389CW256zXCgsj5Dnn8gcTrf0mwziUbjlG
t/UIy7CA7kmpELwna4NNo7Lt6laILqletJi1rlECAwEAATANBgkqhkiG9w0BAQsFAAOCAYEAVkOF
bVwxj/pbnTH8Z2y/17P1yzv4H6vKB2RdG60CMSou0X/WNybBgaMSf6qJJs3osUC68qx27Q3pYp4i
7onsTlNedhSsUVZVabRHXkjLxGLx9saZNiZ9turIyxzfC7VdeGaogvmcFPZAFgkGSFy4tAZz8flk
L7XI9pp5NTrjP9AL1ETVgwoHFKoeEKU1ewgQGRXpsM2sQnanMrTOgfVWz+qmaMmCcgeuQnYDPkZX
X3jo456N0IDcGhJRmzkO8x0ge3DGyTc2mdS+38c61VEDd2TQHDHJuGsjCSVMjYh83JF7Ut3imFYh
v3jgmHNkEDsp7XU81UMaV1nD0WzwNTbuMlyuvUQltLtQ0lciDl+yT7zciHZr3JkL3am9lCtny/DR
Oyw7pZnDCbWHaUKl4pV5UtwCIT/o5v7yo3av1z5o6Ufial+kemeyhcU7PtMXZ6mgW9Hcq4htX1BT
l/LsTN/42XxvrdzystkmvJeSlrNLPbeASi8MC3j/xQdUjc6mWQ/t
-----END CERTIFICATE-----

For production, please contact via email at ecommerce_support@cardlink.gr