

# CyberSource Payer Authentication

## RuPay Integration Guide

March 2019



## CyberSource Contact Information

For general information about our company, products, and services, go to <http://www.cybersource.com>.

For sales questions about any CyberSource Service, email [sales@cybersource.com](mailto:sales@cybersource.com) or call 650-432-7350 or 888-330-2300 (toll free in the United States).

For support information about any CyberSource Service, visit the Support Center: <http://www.cybersource.com/support>

## Copyright

© 2019 CyberSource Corporation. All rights reserved. CyberSource Corporation ("CyberSource") furnishes this document and the software described in this document under the applicable agreement between the reader of this document ("You") and CyberSource ("Agreement"). You may use this document and/or software only in accordance with the terms of the Agreement. Except as expressly set forth in the Agreement, the information contained in this document is subject to change without notice and therefore should not be interpreted in any way as a guarantee or warranty by CyberSource. CyberSource assumes no responsibility or liability for any errors that may appear in this document. The copyrighted software that accompanies this document is licensed to You for use only in strict accordance with the Agreement. You should read the Agreement carefully before using the software. Except as permitted by the Agreement, You may not reproduce any part of this document, store this document in a retrieval system, or transmit this document, in any form or by any means, electronic, mechanical, recording, or otherwise, without the prior written consent of CyberSource.

## Restricted Rights Legends

**For Government or defense agencies.** Use, duplication, or disclosure by the Government or defense agencies is subject to restrictions as set forth the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013 and in similar clauses in the FAR and NASA FAR Supplement.

**For civilian agencies.** Use, reproduction, or disclosure is subject to restrictions set forth in subparagraphs (a) through (d) of the Commercial Computer Software Restricted Rights clause at 52.227-19 and the limitations set forth in CyberSource Corporation's standard commercial agreement for this software. Unpublished rights reserved under the copyright laws of the United States.

## Trademarks

Authorize.Net, eCheck.Net, and The Power of Payment are registered trademarks of CyberSource Corporation.

CyberSource, CyberSource Payment Manager, CyberSource Risk Manager, CyberSource Decision Manager, and CyberSource Connect are trademarks and/or service marks of CyberSource Corporation.

All other brands and product names are trademarks or registered trademarks of their respective owners.

# Contents

<b>About This Guide</b>	<b>4</b>
Recent Revisions	4
Audience and Purpose	4
Conventions	4
Related Documentation	5
<b>Implementing RuPay Payer Authentication</b>	<b>6</b>
Introduction	6
Overview of RuPay Payer Authentication	6
Required Merchant Information	7
Step 1: Checking Enrollment	7
A. Requesting the Check Enrollment Service	7
B. Interpreting the Check Enrollment Service Reply	9
Step 2: Authenticating Enrolled Cards	10
A. Creating the HTTP POST Form	10
B. Creating the HTML Frame for Authentication	11
C. Receiving the PAREs Message from the RuPay Card-Issuing Bank	11
Step 3: Validating Authentication	12
A. Requesting the Validation Service	12
B. Interpreting the Validation Service Reply	14
C. Redirecting Customers to Pass or Fail Message Page	14
Request and Reply Examples	15
Enrollment Service	15
Validate Authentication Service	19
Authorization Service	23
Handling Authorization Timeouts Using the Check Status Service	26

# About This Guide

## Recent Revisions

---

Release	Changes
March 2019	Initial release.

---

## Audience and Purpose

---

This guide is written for application developers who want to use the CyberSource Simple Order API or SCMP API to integrate Payer Authentication services into their order management system to process RuPay payments. It describes the tasks you must perform in order to complete this integration.

Implementing CyberSource Payer Authentication services requires software development skills. You must write code that uses the API request and reply fields to integrate payer authentication services into your existing order management system.

## Conventions

---



### Note

A *Note* contains helpful suggestions or references to material not contained in the document.

---



### Important

An *Important* statement contains information essential to successfully completing a task or learning a concept.

---

## Related Documentation

---

- *Getting Started with CyberSource Advanced for the Simple Order API* ([PDF](#) | [HTML](#))
- *Getting Started with CyberSource Advanced for the SCMP API* ([PDF](#) | [HTML](#))
- *Payer Authentication Using the Simple Order API* ([PDF](#) | [HTML](#)) describes how to integrate CyberSource card authentication services with your web store, including Visa Verified by Visa, Mastercard and Maestro SecureCode (UK domestic and international), American Express SafeKey, and JCB J/Secure.
- *Payer Authentication Using the SCMP API* ([PDF](#) | [HTML](#)) describes how to integrate CyberSource card authentication services with your web store, including Visa Verified by Visa, Mastercard and Maestro SecureCode (UK domestic and international), American Express SafeKey, and JCB J/Secure.
- *Decision Manager Developer Guide Using the Simple Order API* describes how to integrate Decision Manager, a fraud detection service, with your order management system. ([PDF](#) | [HTML](#))
- *Decision Manager Developer Guide Using the SCMP API* describes how to integrate Decision Manager, a fraud detection service, with your order management system. ([PDF](#) | [HTML](#))
- *Credit Card Services Using the Simple Order API* describes how to integrate CyberSource payment processing services into your business. ([PDF](#) | [HTML](#))
- *Credit Card Services Using the SCMP API* describes how to integrate CyberSource payment processing services into your business. ([PDF](#) | [HTML](#))
- *Secure Acceptance Hosted Checkout Integration Guide* describes how to create Secure Acceptance profiles, which enable you to integrate your order management system with the Secure Acceptance web/mobile checkout. ([PDF](#) | [HTML](#))
- *Secure Acceptance Checkout API Integration Guide* describes how to create Secure Acceptance profiles, which enable you to integrate your order management system with a web site to process transactions. ([PDF](#) | [HTML](#))
- The [CyberSource API Versions page](#) provides information about the CyberSource API versions.

Refer to the Support Center for complete CyberSource technical documentation:

[http://www.cybersource.com/support\\_center/support\\_documentation](http://www.cybersource.com/support_center/support_documentation)

# Implementing RuPay Payer Authentication

## Introduction

---

This guide is for merchants who want to connect to CyberSource endpoints for processing RuPay payments. It includes information about authenticating, authorizing, and capturing RuPay payments.

## Overview of RuPay Payer Authentication

---

CyberSource payer authentication services enable you to add support to your web store for card authentication services for RuPay cards.

Payer authentication for RuPay uses the same API services provided by CyberSource for other card brands. If you are currently using CyberSource payer authentication services for other card brands, you can enhance your existing integration to send the additional fields in the request that are required for RuPay cards.

Payer authentication provides the following services:

- **Check Enrollment:** determines whether the customer is enrolled in one of the card authentication programs.
- **Validate Authentication:** ensures that the authentication that you receive from the issuing bank is valid.



Unlike Visa and Mastercard cards, authentication is mandatory for RuPay cards. Without authentication, authorization cannot be performed and will be declined by the RuPay network.

---

## Required Merchant Information

Before using CyberSource payer authentication services for RuPay cards in production, you must contact Customer Support to provide information about your company and your acquiring bank so that CyberSource can configure your account to implement these services.

The process to integrate with CyberSource for RuPay processing is similar to the process to integrate for Visa and Mastercard 3D Secure.

The following sections describe the RuPay authentication flow and integration steps.

## Step 1: Checking Enrollment

When the customer places an order on your web site, your order management system processes the purchase information from the POST of the final page of the order. The goal is to verify that the card is enrolled and to authenticate the results if it is. To do so, you request the Enrollment Check service (VEReq) and then proceed as follows:

- If the card is enrolled, the VERes reply field indicates enrollment. The reply also contains the URL of the Access Control Server and the PAREq.
- If the card is not enrolled, decline the payment and ask the customer for another form of payment.

### A. Requesting the Check Enrollment Service

Use the Check Enrollment service to verify that the card is enrolled in a card authentication program. For a list of the fields used when requesting the service, see Appendix A, “API Fields,” in *Payer Authentication Using the Simple Order API* or *Payer Authentication Using the SCMP API*.

For RuPay, you can use the same request fields in the Check Enrollment service that you currently use for Visa and Mastercard. However, four additional fields are required:

#### 1 billto\_ipAddress (Simple Order API)

customer\_ipaddress (SCMP API)

The IP address must be the IP address of the customer who is making the purchase on your web site. It must not be hard-coded or contain the address of the merchant’s servers. RuPay requires that the correct IP address must be sent because it is used to manage disputes.

**2** payerAuthEnrollService\_httpAccept (Simple Order API)

pa\_http\_accept (SCMP API)

The http accept field must contain the value of the `Accept` header sent by the customer's web browser.

**3** payerAuthEnrollService\_httpUserAgent (Simple Order API)

pa\_http\_user\_agent (SCMP API)

The http user agent field must contain the value of the `User-Agent` header sent by the customer's web browser.

**4** card\_cvNumber (Simple Order API)

customer\_cc\_cv\_number (SCMP API)

In addition, you can send the following optional fields in the request. If they are not sent in the request, the values that have been configured for CyberSource during on boarding are used. CyberSource recommends that you use the configured values to minimize errors.

- invoiceHeader\_merchantDescriptor (name of the merchant as configured in CyberSource, 1-23 alphanumeric characters)
- invoiceHeader\_merchantDescriptorContact (telephone no. of the merchant)
- invoiceHeader\_merchantDescriptorStreet (street name of the merchant)
- invoiceHeader\_merchantDescriptorCity (city)
- invoiceHeader\_merchantDescriptorState (state, must use Indian state codes)
- invoiceHeader\_merchantDescriptorPostalCode (maximum of 9 alphanumeric characters, must be valid postal code)
- invoiceHeader\_merchantDescriptorCountry (must be 'IN' ONLY)



You can send the required and optional fields listed above for other card brands to keep your integration consistent.

---



## B. Interpreting the Check Enrollment Service Reply

The replies are similar for all card types.

### *Enrolled Cards*

You receive reason code 475 (Simple Order API) or reply flag DAUTHENTICATE (SCMP API) if the customer's card is enrolled in a payer authentication program.

When you receive this reply, you can proceed to ["Step 2: Authenticating Enrolled Cards."](#)

### *Cards Not Enrolled*

You receive reason code 100 or reply flag SOK if the account number is not enrolled in RuPay's payer authentication program. The other services in your request are declined.

If you receive this reply, you cannot proceed to validate authentication or to card authorization.

## Step 2: Authenticating Enrolled Cards

When you have verified that a customer's card is enrolled in a card authentication program, you must redirect the customer to the URL of the card-issuing bank's Access Control Server (ACS URL) by using an HTTP POST request web form that contains the PaReq data, the Termination URL (TermURL), and merchant data (MD).



### Note

The MD value must be posted for RuPay. If needed, you can include it for other card brands as well.

### A. Creating the HTTP POST Form

#### Example POST Form

```
if card is enrolled == TRUE Then
    variable acsURL = <acsURL reply field>
    variable paReq = <paReq reply field>

    <body onload="document.PAEnrollForm.submit ();">
    <form id="PAEnrollForm" name="PAEnrollForm" action="acsURL value"
    method="post" target="paInlineFrame">
        <input type="hidden" name="PaReq" value="paReq value"
        <input type="hidden" name="TermUrl" value="http://
        myPAValidationPage.ext" /
        <input type="hidden" name="MD" value="<xid value>" />
    </form>
else
```

The page typically includes JavaScript that automatically posts the form. This code provides the following:

- A page that receives the reply fields for the enrollment check service.
- A form that contains the required data for the card-issuing bank.

## B. Creating the HTML Frame for Authentication

When your customers are redirected to the ACS URL, their browsers display the frame that contains the card-issuing bank's password authentication dialog box or the option to sign up for the program (activation form).

On the page that contains the in-line frame for the ACS URL, add the following:

- HTML frame large enough to accommodate the card-issuer's authentication form or the activation form, and text that describes the process to customers.
- Outside the HTML frame, you must provide a brief message that guides customers through the process. For example, *"Please wait while we process your request. Do not click the Back button or refresh the page. Otherwise, this transaction may be interrupted."*

## C. Receiving the PAREs Message from the RuPay Card-Issuing Bank

The card-issuing bank sends a PAREs message to your TermURL in response to the PAREq data that you sent with the web form. The PAREs message is sent by using an HTTP POST request and contains the result of the authentication that you requested.

The signed **PaRes** field contains a base64-encoded string that contains the following information:

- **PAREs**  
Digitally signed payer authentication response message that contains the authentication result.



The field name has a lowercase "a" (**PaRes**), but the message name has an uppercase "A" (PAREs).

---

- **MD**  
Merchant data, which must be submitted for RuPay.

## Step 3: Validating Authentication

For enrolled cards, the next step is to request the validation service to verify the authentication message (PAREs) returned by the card-issuing bank.

### A. Requesting the Validation Service

When you make the validation request, you must:

- Extract the PAREs message from the form received from the card-issuing bank.
- Remove all spaces created by tabs, spaces, or line breaks from the **PaRes** field. Do not modify any other part of the **PaRes** field.



If you use the Simple Order API 1.128 or later, CyberSource will remove all space characters during the validation service. If you use an earlier API version, you must remove all space characters or the validation service request will fail.

If you use the SCMP API, you must remove all space characters or the validation service request will fail.

---

- Send the PAREs to CyberSource in the signed **PaRes** field of the validation service. The reply that you receive contains the validation result.


You can use the validation and card authorization services in the same request or in separate requests:

- *Same request:* CyberSource automatically attempts to authorize your customer's card if validation succeeds. The values of the required fields are added automatically to the authorization service. If you use this method, do not pass into your request any fields that CyberSource derives from the PAREs because that data could be overwritten.

- *Separate requests*: you must manually include the validation result values (Payer Authentication Reply Fields) in the authorization service request (Card Authorization Request Fields), which are listed in the following table:

**Table 1 Authorization Check and Reply Fields**

Identifier	Payer Authentication Reply Field	Card Authorization Request Field
XID		
Simple Order API	payerAuthValidateReply_xid	ccAuthService_xid
SCMP API	pa_validate_xid	xid
E-commerce indicator		
Simple Order API	payerAuthValidateReply_commerceIndicator	ccAuthService_commerceIndicator
SCMP API	pa_validate_e_commerce_indicator	e_commerce_indicator
CAVV		
Simple Order API	PayerAuthValidateReply_cavv	ccAuthService_cavv
SCMP API	pa_validate_cavv	cavv

 **Important** If you are currently passing additional card-specific values in the Payer Authentication Validate Reply for Visa and Mastercard, you can continue to pass them for RuPay as well.

## B. Interpreting the Validation Service Reply

Proceed with the order according to the validation response that you receive. The replies are similar for all card types:

- *Success:*

You receive the reason code 100 or reply flag SOK, and other service requests, including authorization, are processed normally.

- *Failure:*

You receive reason code 476 indicating that the authentication failed or the reply flag DAUTHENTICATIONFAILED, so the other services in your request are not processed.

- *Error:*

If you receive an error from the payment card company, process the order according to your business rules. If the error occurs frequently, report it to Customer Support. If you receive a CyberSource system error, determine the cause, and proceed with card authorization only if appropriate.

To verify that the enrollment and validation checks are for the same transaction, ensure that the XID in the enrollment check and validation replies are identical.

## C. Redirecting Customers to Pass or Fail Message Page

After authentication is complete, redirect the customer to a page containing a success or failure message. You must ensure that all messages that display to customers are accurate, complete, and that they address all possible scenarios for enrolled and non-enrolled cards. For example, if the authentication fails, a message such as the following should be displayed to the customer:

```
Authentication Failed
```

```
Your card issuer cannot authenticate this card. Please select another card or form of payment to complete your purchase.
```

# Request and Reply Examples

---

## Enrollment Service

### Example 1 Enrollment Request (SCMP API)

---

```
bill_address1=123 Main St.
bill_city=Ann Arbor
bill_country=US
bill_state=MI
bill_zip=48104-2201
card_type=061
client_lib_version=Oracle Corporation/1.8.0_172/Windows Server 2008 R2/
6.1/-/Java/5.2.1
currency=INR
customer_cc_cv_number=123
customer_cc_expmo=12
customer_cc_expyr=2031
customer_cc_number=5082302886091
customer_email=null@cybersource.com
customer_firstname=John
customer_ipaddress=10.0.0.1
customer_lastname=Smith
customer_phone=999-999-9999
grand_total_amount=100
ics_applications=ics_pa_enroll
merchant_id=npr_rupay
merchant_ref_number=S208441-1
pa_http_accept=text/html,application/xhtml+xml,application/
xml;q=0.9,image/webp,image/apng,*/*;q=0.8
pa_http_user_agent=** Mozilla/5.0 (Linux; Android 6.0; Nexus 5 Build/
MRA58N) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/69.0.3497.92
Mobile Safari/537.36
request_id=5378691389053406795329
sender_id=npr_rupay
```

---

**Example 2 Enrollment Reply (SCMP API)**


---

```

ics_decision_reason_code=475
ics_rcode=0
ics_return_code=1000000
ics_rflag=DAUTHENTICATE
ics_rmsg=The cardholder is enrolled in Payer Authentication. Please
authenticate before proceeding with authorization.
merchant_ref_number=S208441-1
pa_enroll_reason_code=475
pa_enroll_acs_url=https://s173flxapq001.visa.com:8443/cybersource/
payerAuthentication/paySecure/initiate?binType=D
pa_enroll_authentication_path=ENROLLED
pa_enroll_authentication_transaction_id=ODI4MzE1MjYwNjAz
pa_enroll_
pareq=e0NCQ2l2fTdIUWRIVmh5dDNpBxJRVGdzCUNWMUFvbiszK3dlckQxbmNHNUTCTnNGS
jBCaWprR3AyL0JGMXhpN1I2WnNleDloUmVmNW95cnIyQk1rRVZkVW1ZMjFWRk05Q1J5SFMr
Tk45WWUxa1c2MDJqazNITWViafZpY240Z1R5V3d0RGFjM3NrRHBudGdCaUU0a jdmTdxQTJ
QZ1VwQURFd0M3R2phZEhCeFluaUJGNHJ3WThWYXd5NGpDZUtFemZ2YTJPeHoyeFNNT2lvcG
dQR3pSN29iy21jMzRuTUFnRG1BMy83MS9MT09VaUczTUFTU3ZxL2RDQnBiWUpldlIxdUY3R
VU0cUFTQ3QwbFV3ZjBrY2QzR0szcVRnbzVCSGNLRGROaERjWGZnK3JkeTJFeEg3b0MxcDdt
bUI2TXZNRmd4UnJLaWhZMnNHAEJSOFVjSDEvb2dzM3U2NWpaODZsK1JJULhkWm1WTFFLT3M
5YnAZmWs1QXRnZjlza2R2dEtRLzhpZndyOW1LUW9NdEwrWjF1VG1CS21xS0JIR0I2bFRRUF
dod1J3UUyYwktWTG1ZZjk4V0hleUhrNXBmZjVkeXlHSWZJMC8vY1BmMzB2V0grREZxREN1S
1AyQmxyUEpmSVRkR2d1NVBkVlM2WXEwNXJOCVVwRTNITUNQTm5McHFNa3NrbmtTOWdpcDlt
YkVEZlFqTmRZaVZXWnVsb1BuUm8vekZmanQwemp6U0hwSGhTQ0NRRzZLbG1RdlZMYzdGZZp
NdGo4MX1aVhd2c2NYaXFKU1NmekZNSWVTcFFqS1JwUmlrYlFqQ0JwelExenM9
pa_enroll_rcode=0
pa_enroll_return_code=1052000
pa_enroll_rflag=DAUTHENTICATE
pa_enroll_rmsg=The cardholder is enrolled in Payer Authentication.
Please authenticate before proceeding with authorization.
pa_enroll_specification_version=rupay
pa_enroll_veres_enrolled=Y
pa_enroll_xid=ODI4MzE1MjYwNjAz
request_id=5391652683756017801540
request_token=AhjzBWSTJEg2Pa0G991E3j1P90YqAeFkHTiEC2u3t/9i4nmGhsBoAAAA/
AEj

```

---



**Example 3 Enrollment Request (Simple Order API)**


---

```

<soapenv:Envelope
  xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:urn="urn:schemas-cybersource-com:transaction-data-1.145">
  <soapenv:Header/>
  <soapenv:Body>
    <requestMessage xmlns="urn:schemas-cybersource-com:transaction-
data-1.145">
      <merchantID>tpm_rupay</merchantID>
      <merchantReferenceCode>Merchant_Ref</merchantReferenceCode>
      <billTo>
        <firstName>John</firstName>
        <lastName>Doe</lastName>
        <street1>1st street</street1>
        <city>Bangalore</city>
        <state>KA</state>
        <postalCode>94043</postalCode>
        <country>IN</country>
        <email>john.doe@example.com</email>
        <ipAddress>1.1.1.1</ipAddress>
      </billTo>
      <purchaseTotals>
        <currency>INR</currency>
        <grandTotalAmount>120</grandTotalAmount>
      </purchaseTotals>
      <card>
        <accountNumber>6074819900004939</accountNumber>
        <expirationMonth>12</expirationMonth>
        <expirationYear>20</expirationYear>
        <cvNumber>123</cvNumber>
      </card>
      <payerAuthEnrollService run="true"> <httpAccept>text/
html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/
png,*/*;q=0.8</httpAccept>
      <httpUserAgent>Mozilla/5.0 (Linux; Android 6.0; Nexus 5 Build/
MRA58N) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/69.0.3497.92
Mobile Safari/537.36</httpUserAgent>
    </payerAuthEnrollService>
  </requestMessage>
</soapenv:Body>
</soapenv:Envelope>

```

---

**Important**

In the enroll reply, the URL returned by CyberSource provides the type of card. For instance, in [Example 4](#) in the URL (<https://pnrstage.ic3.com:9448/cybersource/payerAuthentication/paySecure/initiate?binType=S>), the card/BIN type of the RuPay card is S, which is a single message BIN. If this type of BIN is returned by CyberSource, then you must send the authorization service and capture service API requests together to CyberSource at the same time.

---

**Example 4 Enrollment Reply (Simple Order API)**


---

```

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <wsse:Security xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd">
      <wsu:Timestamp xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd" wsu:Id="Timestamp-1343922504">
        <wsu:Created>2018-10-10T09:07:30.755Z</wsu:Created>
      </wsu:Timestamp>
    </wsse:Security>
  </soap:Header>
  <soap:Body>
    <c:replyMessage xmlns:c="urn:schemas-cybersource-com:transaction-data-1.145">
      <c:merchantReferenceCode>Merchant_Ref</c:merchantReferenceCode>
      <c:requestID>5391624464076018403031</c:requestID>
      <c:decision>REJECT</c:decision>
      <c:reasonCode>475</c:reasonCode>
      <c:requestToken>AhjzbwSTJefR/AbE7cbX3z1P90c3QRFkHXcEC2u3t/9i4nmpQYCKAAAA9Abp</c:requestToken>
      <c:payerAuthEnrollReply>
        <c:reasonCode>475</c:reasonCode>
        <c:acsURL>https://pnrstage.ic3.com:9448/cybersource/payerAuthentication/paySecure/initiate?binType=S</c:acsURL>
        <c:paReq>e0NCQ2l2fUJlMHM5eFowTct4SFNTTzBvNmZJT1N1aU1SUFZyRTN5RmxPNHQxV2U5d3J2bHQvdVQ2SWhBWS9sTnc3em1NcWhPNTQxT01Dc2RIc01ZMU51VzRIWlHMTBSU0pacTRYOXNuMGFBUC9HL2YxYlpySS9ldit6YmpsS2MrcEJFUWIXb095aVFDaWZraXhHSmxSRlcrdTzkT2lVeFlQNS9aWFJEMXRrakQwWfVJdXfMY0RIR2JVWFlowe0zL1dhTzJQdER5Y1VoQ2hzSmlpbzZSNXpueDIxNTF1blU0RmMvY0RKRGmzVFE4SDVXVW1QNmlPVjZXWnIwZUx2MFFCSWJodUkvZHJEd0Y2d2dNSGpRRHNnd0olbUtPUXdmTmxHdlFTT0dGcUdaLzVGd0JjTFpQNWxIZ3FoRnhHT1FOeGc0UmJteXROMVdDQVBualdoSUJlcGYvaVF3dUx1NGRqWHRLe1hkdmTzZmJlCdEJhSjBnaDgxVk5mMitEdHpNVmpPLORhUEZ2N1p5dkJDMFltQjJTMnJSmlVnNVlONmQrcExXczhNalVEKzNjYzV0Nm13WVNmZmIwUUNPbmNjLzRrP3JXaXBYVDFNWjBoedNqKzBEaitRZE52TDZ0YVZjSDZoNUFHd0U5RGpDRWpiVzZEZ243bGY0cjdTMVEzYz1DL3RzTldBcXErZ012WmFXU0czRUPOdloyNmJHeVExNG9POTJkRG5jMWZ1elQ3ci84dEV6RjlvRUJSchdRSFk3NTN1ODFuWWVBQWJJOu5Zbi9RblhkdzdJc1R6d2pMaW5SdmNnOXFCTzBHNUxtKORlMVE9</c:paReq>
        <c:xid>ODI4MzE0MjU3NjI4</c:xid>
        <c:veresEnrolled>Y</c:veresEnrolled>
        <c:authenticationPath>ENROLLED</c:authenticationPath>
        <c:specificationVersion>rupay</c:specificationVersion>
        <c:authenticationTransactionID>ODI4MzE0MjU3NjI4</c:authenticationTransactionID>
      </c:payerAuthEnrollReply>
    </c:replyMessage>
  </soap:Body>
</soap:Envelope>

```

---

## Validate Authentication Service

### Example 5 Validate Authentication Request (SCMP API)

---

```
customer_cc_number=5082302886091
pa_http_user_agent=user
customer_ipaddress=10.0.0.1
pa_http_accept=accept
currency=INR
customer_cc_cv_number=123
merchant_descriptor_name=name
merchant_descriptor_city=bangalore
merchant_descriptor_state=ka
merchant_descriptor_postal_code=1234
merchant_descriptor_contact=1234567890
ics_applications=ics_pa_validate
merchant_ref_number=S208441-1
customer_cc_expmo=12
customer_cc_expyr=2031
merchant_category_code=1234
grand_total_amount=100
pa_
signedpares=e0NCQ2l2fTlLTDJ2VzJFRG8vN0RZbDRXM28xY1VHZkxPdWF3MC9XQnViOUd
1bjQ3MkUvS2tZdlhoRmVqUzLzldldQRVZ3NDd3R1J2ZzNGUnlXSnRUVVBFES8xSkFLbnQ3d3
FpZUxRM29xb1ZKOFF5M2ltdUJuVWclTmZNV3V2T0pjSStEQ2hrVFEvNmJGdk9pNzVWdDdnb
FpKeVBEed3dMalltMG5tOGhxVlE5MwWxbEJieXA5cHhRVFhWQVRQMw1lVlRBbEtYLyTTnVo
dllKOUFqaWMyRU5JZ244OUpadzJmSENQRlZaUG1OSTNQBhHsQS9aSHlpNjdFalhnT1lFYjV
idktkRW1WLzY4ZUdyOWU2Q1B1VDU0bzdkc0Y0cWlFTnlpN2dzazdUa01seEhvQi9PdGZNOG
I4UnVCd0plTUh5aGNnbUt6bi9mMUxrM1NHVlQxQTJ6V3VuckZDRmN5RjdkZ0hQTVJ1SXFjb
2g4b1NQSHRqRlhEN3QzTi9ZZ1Jxa2dEOWsvceZUOE9kbnlYNwdQWUpDSGNwWGyRSlI4c25X
a3VscVVKWFBuazQ3aHVjYVYvUzJ5NGxqOUhVTnNmQTFrSGRmM0M5N1BaNFfw2R6eEVRUS9
YVWI2MWRzVXQvQld4cE9acFBZNYtNbn3QVB4a1Y4UUY1Z0NURWRjr3hfU2FFWJRUmVTME
tjY1Fjs1lqMGZlV3hHSHB6N2o0VG9HUCtxdzkzekYlWFBaQ2RCMUxoa2pETk91SmdIb2dlO
C9RcXFzQXd0TmNXOEo0SGM4RlhBa1dtNTBVSWG2TXFXc3BJU3VMZUNhazJyLzhUQU1pcXNT
aDRBTnhSS0VPTWxRK2ZNY2dkUG8yZ0w4eFlnc25tdmE3bG1iYVN3PT0mQ0FSRF9UWVBFPTA
2MQ==
merchant_id=npr_rupay_test
```

---

**Example 6    Validate Authentication Reply (SCMP API)**

---

```
ics_decision_reason_code=100
ics_rcode=1
ics_return_code=1000000
ics_rflag=SOK
ics_rmsg=Request was processed successfully.
merchant_ref_number=S208441-1
pa_validate.reason_code=100
pa_validate_authentication_result=0
pa_validate_authentication_status_msg=Success
pa_validate_cavv=NTAwMDAwMDAwMDAwMDAwMDAwMDAwMDAyMTkyNjAw
pa_validate_e_commerce_indicator=rpy
pa_validate_eci=05
pa_validate_eci_raw=05
pa_validate_pares_status=Y
pa_validate_rcode=1
pa_validate_return_code=1050001
pa_validate_rflag=SOK
pa_validate_rmsg=ics_pa_validate service was successful
pa_validate_specification_version=rupay
pa_validate_xid=ODI4MzE2MjU5MjE0
request_id=5391703149786000601540
request_token=AhjjbwSTJEjpiENFUTHE3k/3RzdB9WQdOIQLcMm3/2LiaAewGgA8QSs
```

---

**Example 7    Validate Authentication Request (Simple Order API)**


---

```

<soapenv:Envelope
  xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:urn="urn:schemas-cybersource-com:transaction-data-1.145">
  <soapenv:Header/>
  <soapenv:Body>
    <requestMessage xmlns="urn:schemas-cybersource-com:transaction-
data-1.145">
      <merchantID>tpm_rupay</merchantID>
      <merchantReferenceCode>Merchant_Ref</merchantReferenceCode>
      <purchaseTotals>
        <currency>INR</currency>
        <grandTotalAmount>120</grandTotalAmount>
      </purchaseTotals>
      <payerAuthValidateService run="true">
<signedPAREs>e0NCQ2l2fTlLTDJ2VzJFRG8vN0RZbDRXM28xY1VHZkxPdWF3MC9XQnViOU
d1bjQ3MkUvS2tZdlhoRmVqUz1zldldQRVZ3Ndd3R1J2ZzNGUnlXSnRUVVBFES8xSkFLbnQ3d
3FpZUxRM29xb1ZKOFF5M21tdUJuVwclTmZNV3V2T0pjsStEQ2hrVFEvNmJGdk9pNzVWdDdn
bFpKeVBEd3dMalltMG5tOGhxV1E5MWwxbEJieXA5cHhRVFhWQVRQMw1IV1RBbEtYLyTtTnV
od1lKOUFqaWMyRU5JZ2440UpadzJmSENQR1ZaUG1OSTNQbHhSQS9aSHlpNjdFalhnT1lFYj
VidktrRW1WLzY4ZUdyOWU2Q1B1VDU0bzdkc0Y0cW1FTnlpN2dzazdUa01seEhvQi9PdGZNO
GI4UnVCd0plTUh5aGNnbUt6bi9mMuxrM1NHV1QxQTJ6V3VuckZDRmN5RjdkZ0hQTVJ1SXFj
b2g4b1NQSHRqRlhEN3QzTi9ZZ1Jxa2dEOWsvceZUOE9kbn1YNwdQWUpDSGNwWGYrS1I4c25
Xa3VscVVKWFBUazQ3aHVjYVYvUzJ5NGxqOUhVTnNmQTFrSGRmM0M5N1BaNFfb2R6eEVRUS
9YVWI2MWRzVXQvQld4cE9acFBZNYtNbn3QVB4a1Y4UUY1Z0NURWRjr3hFU2FFWGJRUMVTM
EtjY1Fjs1lqMGZ1V3hHSHB6N2o0VG9HUCtxdzkzekY1WFBaQ2RCMUxoa2pETk91SmdIb2dl
OC9RcXFzQXd0TmNXOEo0SGM4RlhBaldtNTBVSWg2TXXFc3BJU3VMZUNhazJyLzhUQU1pcXN
taDRBTnhSS0VPTWxRK2ZNY2dkUG8yZ0w4eFlnc25tdmE3bG1iYVN3PT0mQ0FSRF9UWVBFPT
A2MQ==</signedPAREs>
      </payerAuthValidateService>
    </requestMessage>
  </soapenv:Body>
</soapenv:Envelope>

```

---

**Example 8    Validate Authentication Reply (Simple Order API)**


---

```

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <wsse:Security xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd">
      <wsu:Timestamp xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd" wsu:Id="Timestamp-1675446337">
        <wsu:Created>2018-10-10T11:21:30.926Z</wsu:Created>
      </wsu:Timestamp>
    </wsse:Security>
  </soap:Header>
  <soap:Body>
    <c:replyMessage xmlns:c="urn:schemas-cybersource-com:transaction-data-1.145">
      <c:merchantReferenceCode>Merchant_Ref</c:merchantReferenceCode>
      <c:requestID>5391704908026004903031</c:requestID>
      <c:decision>ACCEPT</c:decision>
      <c:reasonCode>100</c:reasonCode>
      <c:requestToken>AhjzLwSTJEjvx19sNBR33z1P90c3QflkHXcEC3DJt/9i4nmpQYAAvDmh</c:requestToken>
      <c:payerAuthValidateReply>
        <c:reasonCode>100</c:reasonCode>
        <c:authenticationResult>0</c:authenticationResult>
        <c:authenticationStatusMessage>Success</c:authenticationStatusMessage>
        <c:cavv>NTAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAyMTkyNjAw</c:cavv>
        <c:commerceIndicator>rpy</c:commerceIndicator>
        <c:eci>05</c:eci>
        <c:eciRaw>05</c:eciRaw>
        <c:xid>ODI4MzE2MjU5MjE0</c:xid>
        <c:paresStatus>Y</c:paresStatus>
        <c:specificationVersion>rupay</c:specificationVersion>
      </c:payerAuthValidateReply>
    </c:replyMessage>
  </soap:Body>
</soap:Envelope>

```

---

## Authorization Service

The authorization service format that you need to send for RuPay is the same as you use for other card types. You must send the CAVV and XID in the authorization service request along with the card details for CyberSource to process this request with the RuPay card network.

For RuPay, the e-commerce indicator returned in the validation service response must be set to `rupy` or the authorization results in an error.



### Note

For an SMS type of card, you must send the authorization service and capture service requests at the same time. Sending the authorization service request alone for an SMS type of card results in an error response.

### Example 9 Authorization Request (SCMP API)

```
bill_address1=123 Main St.
bill_city=Ann Arbor
bill_country=IN
bill_state=MI
bill_zip=48104-2201
card_type=061
cavv=MTAwMDAwMDAwMDAwMDAwMDAwMDAwMDI1MjM2
client_lib_version=Oracle Corporation/1.8.0_172/Windows Server 2008 R2/
6.1/-/Java/5.2.1
currency=INR
customer_cc_cv_number=123
customer_cc_expmo=12
customer_cc_expyr=2031
customer_cc_number=5082302886091
customer_email=null@cybersource.com
customer_firstname=John
customer_lastname=Smith
customer_phone=999-999-10048
e_commerce_indicator=rupy
grand_total_amount=100
ics_applications=ics_auth
merchant_id=npr_rupay_test
merchant_ref_number=S208441-50
request_id=5378689648713406795329
sender_id=npr_rupay
xid=ODI4MTEyMjU5ODAy
```

**Example 10 Authorization Reply (SCMP API)**

---

```
auth.reason_code=100
auth_auth_amount=100.00
auth_auth_avs=2
auth_auth_code=183217
auth_auth_time=2018-10-10T112701Z
auth_payment_service_data=320LTM06165658LTD041010RRN12828112259802
auth_rcode=1
auth_responsecode=100
auth_return_code=1010000
auth_rflag=SOK
auth_rmsg=Request was processed successfully.
auth_status=SUCCESS
currency=INR
ics_decision_reason_code=100
ics_rcode=1
ics_return_code=1000000
ics_rflag=SOK
ics_rmsg=Request was processed successfully.
merchant_ref_number=S208441-50
request_id=5391708182936018401540
request_token=Ahj77wSTJEj7aeLUOrkE3k/3RioCTPU/3RioCTWQdOIQLEcMm3/
2LleaAe0GkmSJH208WodXIIAAA8A2I
```

---



**Example 11 Authorization Request (Simple Order API)**


---

```

<requestMessage xmlns="urn:schemas-cybersource-com:transaction-data-
1.145">
    <merchantID>tpm_rupay</merchantID>
    <merchantReferenceCode>Merchant_Ref</merchantReferenceCode>
    <billTo>
        <firstName>John</firstName>
        <lastName>Doe</lastName>
        <street1>1st street</street1>
        <city>Bangalore</city>
        <state>KA</state>
        <postalCode>94043</postalCode>
        <country>IN</country>
        <email>john.doe@example.com</email>
        <ipAddress>1.1.1.1</ipAddress>
    </billTo>
    <purchaseTotals>
        <currency>INR</currency>
        <grandTotalAmount>120</grandTotalAmount>
    </purchaseTotals>
    <card>
        <accountNumber>6074819900004939</accountNumber>
        <expirationMonth>12</expirationMonth>
        <expirationYear>20</expirationYear>
        <cvNumber>123</cvNumber>
    </card>
    <ccAuthService run="true">
        <cavv>PpmBUYXt2uytV6p12345KuImAb8XgnOk</cavv>
        <commerceIndicator>rpy</commerceIndicator>
        <xid>WhPlErd9WE1234562pblyFjFHlewUIQwQ</xid>
        <veresEnrolled>Y</veresEnrolled>
        <paresStatus>Y</paresStatus>
    </ccAuthService>
</requestMessage>

```

---

**Example 12 Authorization Reply (Simple Order API)**


---

```

<c:replyMessage xmlns:c="urn:schemas-cybersource-com:transaction-data-1.145">
  <c:merchantReferenceCode>0123456789</c:merchantReferenceCode>
  <c:requestID>1921312345620167904567</c:requestID>
  <c:decision>ACCEPT</c:decision>
  <c:reasonCode>100</c:reasonCode>
  <c:purchaseTotals>
    <c:currency>INR</c:currency>
  </c:purchaseTotals>
  <c:ccAuthReply>
    <c:reasonCode>100</c:reasonCode>
    <c:amount>120</c:amount>
    <c:authorizationCode>ABC12345</c:authorizationCode>
    <c:processorResponse>0</c:processorResponse>
    <c:reconciliationID>19119123438</c:reconciliationID>
  </c:ccAuthReply>
</c:replyMessage>

```

---

## Handling Authorization Timeouts Using the Check Status Service

Typically, when a timeout occurs during an authorization, CyberSource automatically performs an authorization reversal. However, RuPay does not support online authorization reversals. When a timeout occurs during an authorization for a RuPay transaction, CyberSource includes **ccAuthReply\_reasonCode** field set to 151 (Simple Order API) or an **auth\_rflag** field set to ETIMEOUT (SCMP API) in the authorization reply message. When you receive this value, check the status of the authorization by requesting the check status service.

The check status service includes a payment status field in the reply message. When the value of the payment status field is **AUTHORIZED**, proceed with the order by requesting the capture service. When the value of the payment status field is **DECLINED**, the authorization has been declined. You can request a different form of payment from the customer.

**Example 13 Check Status Request (SCMP API)**


---

```

merchant_id=npr_rupay_test
merchant_ref_number=S208441-50
ics_applications=ics_check_status
auth_request_id=5396859731856233201541
sender_id=npr_rupay_test

```

---

**Example 14 Check Status Reply (SCMP API)**


---

```

ics_rcode=1
ics_rflag=SOK
ics_rmsg=Request was processed successfully.
merchant_ref_number=S208441-1
request_id=5402857441596000201777
check_status_rcode=1
check_status_rflag=SOK
check_status_payment_status=AUTHORIZED
check_status_return_code=9999999
check_status.reason_code=100
request_token=AhjjbwSTJ001iZDlZGwx3lFqAaRm3rIOneIGiGTqzD7qym4r5QNAAQCo
ics_decision_reason_code=100

```

---

**Example 15 Check Status Request (Simple Order API)**


---

```

<requestMessage xmlns="urn:schemas-cybersource-com:transaction-data-1.145">
    <merchantID>tpm_rupay</merchantID>
    <merchantReferenceCode>0123456789</merchantReferenceCode>
    <ccCheckStatusService run="true">
    <authRequestID>545435534535343454334</authRequestID>
    </ccCheckStatusService>
</requestMessage>

```

---

**Example 16 Check Status Reply (Simple Order API)**


---

```

<c:replyMessage xmlns:c="urn:schemas-cybersource-com:transaction-data-1.145">
<c:merchantReferenceCode>0123456789</c:merchantReferenceCode>
<c:requestID>1921312345620167904567</c:requestID>
<c:decision>ACCEPT</c:decision>
<c:reasonCode>100</c:reasonCode>
<c:ccCheckStatusReply>
<paymentStatus>AUTHORIZED</paymentStatus>
</c: ccCheckStatusReply >
</c:replyMessage>

```

---