



***RuPay - PaySecure
Acquirer Integration Guide
Version 1.5 –
14 May 2018***

Contents

A.	Objective of the document.....	6
B.	RuPay eCommerce solution branding / RuPay Mark / PaySecure Mark.....	6
C.	References and publications.....	6
1.	Introduction	7
1.1	NPCI eCommerce Solution: PaySecure	7
1.2	Participants in the RuPay eCommerce transaction cycle	8
1.3	Defining roles of Entities	9
1.4	Architecture for authentication of RuPay online transactions	11
1.5	High-Level Steps for the Acquirer:.....	12
2.	Transaction Flow.....	13
2.1	Iframe Flow.....	14
2.2	Redirection Flow.....	16
3.	Transaction Processing Details	18
3.1	Merchant Ecommerce Site	20
3.2	Checkbin – API call	20
3.3	Checkbin2 – API call.....	20
3.4	Initiate – API call	21
3.5	Initiate2 – API call	21
3.6	Iframe Flow – Browser/App Interaction	21
3.7	Redirection Flow – Browser/App Interaction	26
3.8	Authorize – API call.....	32
4.	Transaction type	32
5.	Transaction-Status	32
6.	Refund Processing.....	33
7.	Clearing and Settlement Process	34
8.	Dispute Resolution and Chargeback Process	34
9.	Transaction ID.....	34
10.	Time out scenarios and handling.....	35
10.1	Acquirer-to-PaySecure (API Communication).....	35
10.2	Acquirer-to-PaySecure (during cardholder authentication process)	35
10.3	Cardholder-to-Issuer (during cardholder authentication)	35
10.4	PaySecure-to-Issuer (during cardholder authentication)	35
10.5	Cardholder-to-PaySecure (PIN capture)	36
10.6	Transaction Initialization to IFrame Initialize	36
10.7	PIN Capture to Authorization Request	36
10.8	Time Out Details	36
11.	System Security Features.....	37
11.1	Merchant/Aggregator Authentication Methods	37
11.2	Acquirer Authentication Methods	37
11.3	Cardholder/ Shopper Browser Authentication Methods.....	37
11.4	Separation of Card details/PAN and OTP authentication mechanism.....	38

11.5	SSL Connection	38
12.	Integration Requirements.....	38
12.1	Software Requirements	38
12.2	Connectivity to NPCI's PaySecure product	38
13.	Implementation Considerations.....	40
14.	Web-Services Requirements	40
15.	Web Service API Calls	41
15.1	CheckBIN	41
15.2	CheckBIN2.....	43
15.3	Initiate.....	44
15.4	Initiate2.....	46
15.5	Authorize	49
15.6	Transaction Status	50
16.	Best Practices.....	51
16.1	Transaction Review and Filtering.....	51
16.2	Network and Infra Security	52
16.3	Adhere to the PCI Data Security Standard Requirements	52
16.4	Suspicious Transactions Monitoring:.....	52
16.5	NPCI recommendations	53
16.6	Card Not Present (CNP) transactions acquisition	53
16.7	Merchant Training	54
16.8	Documents to be submitted	56
Annexures A.....		57
Annexures B.....		60
1.	CheckBIN	60
2.	CheckBIN2.....	62
3.	Initiate	64
4.	Initiate2	71
5.	Redirection Request Parameter	77
6.	Redirection Response Parameter	78
7.	Authorize	78
8.	Transaction Status	80
Annexures C.....		83
Annexures D		89
Annexures E.....		89
Annexures F.....		90

Document History

Document Title		RuPay Acquirer Integration Document	Business/Functional Requirements Document
Document Owner		RuPay eCommerce Team	
Version	Release Date	Prepared/Review by	Comments
1.0	14/08/12	NPCI eCommerce team	Initial Draft Version
1.1	14/02/13	NPCI eCommerce team	Revised Version
1.2	01/08/13	NPCI eCommerce team	Version V 1.2
1.3	30/01/18	NPCI eCommerce team	Version V 1.3
1.4	08/02/18	NPCI eCommerce team	Version V 1.4
1.5	11/05/18	NPCI eCommerce team	Version V 1.5

Document Change Control

Date of Change	Version Number	Section/Reference Number	Reason for Change	Summary of Change	CCR Number
01-08-2013	V1.2	Added 2.4.High value transaction flow, page:20			
01-08-2013	V1.2	Removed 16. Invalid-PIN Logic, page; 29			
01-08-2013	V1.2	Added 11.Hivalue transaction, Page:31			
01-08-2013	V1.2	Added 15.1 time out details in Page: 29			
01-08-2013	V1.2	Added Content regarding XML header in Web services requirements section 20, page: 37			
01-08-2013	V1.2	Removed ACCU700			

		page: 73			
01-08-2013	V1.2	Added MCC details in Annexure F page:75			
21-09-2017	V 1.3	Added Redirection Flow	To make it frictionless transaction flow	Enablement of Iframe and Redirection flow at acquirer	
30-01-2018	V 1.3	Added response code and made text correction	To make it frictionless transaction flow	Enablement of Iframe and Redirection flow at acquirer	
08-02-2018	V 1.4	Updated the description for Custom 4 field in Initiate/Initiate2 Request	for Small Scale Merchant Identifier	Description and possible values has been updated.	
11-05-2018	V1.5	Updated Multiple section and pages of the document Introduced hashing mechanism for MIMA	Yearly review	Re-direction flow I-frame Flow error codes and their descriptions	

A. Objective of the document

The RuPay Acquirer Integration Guide is intended for Acquirer banks and their service providers that are evaluating or have decided to implement the RuPay eCommerce solution. This guide explains the RuPay eCommerce solution and its benefits, transaction flows, and implementation planning and considerations. The objective is to help the Acquirer plan the development, testing, certification, and production setup of the RuPay eCommerce solution.

This guide is targeted at the Project Manager, Functional and Technical integration resources tasked with implementing PaySecure Internet PIN Debit/OTP Validation mechanism. The focus is online checkout interaction.

B. RuPay eCommerce solution branding / RuPay Mark / PaySecure Mark

Refer RuPay Card Marks and Specifications for this section.

C. References and publications

This document must be read in conjunction with the following documents:

- ▶ RuPay Card Marks and Specifications
- ▶ RuPay IIN Maintenance Manual
- ▶ RuPay Bylaws
- ▶ RuPay Implementation Guidebook
- ▶ RuPay Online Member Manual
- ▶ RuPay Dispute Management Rules and Regulations
- ▶ RuPay Global Clearing and Settlement Manual
- ▶ RuPay Product Manual
- ▶ RuPay eCommerce White Paper
- ▶ RuPay Fraud Risk Management
- ▶ Member Certification Guidebook

1. Introduction

1.1 NPCI eCommerce Solution: PaySecure

RuPay, the card scheme launched by the National Payments Corporation of India (NPCI), has been conceived to fulfil RBI's vision to offer a domestic, open-loop, multilateral system which will allow all Indian banks and financial institutions in India to participate in the electronic payments market. RuPay's strategic objectives include assisting in digitization of cash payments, creating a common platform for all banks & payment forms/channels, becoming a 'top of the wallet card' for all Indian consumers across segments, providing a viable domestic option to Indian banks and acting as a nodal body for the electronics payment industry.

RuPay, being the first domestic card scheme, is in a unique position to work together with banks and other entities including government, public sector and private sector entities to increase the cards spends as a percentage of personal consumption expenditure. The RuPay card aims to deliver to the stakeholders a convenient and easy e-commerce experience without compromising on the security and risk. The online PaySecure module hosted by NPCI for RuPay cards would:

- Reduce customers' effort
- Require minimal changes to stake holders system
- Quick on-boarding process for merchants
- No compromise on security and risk
- The solution offers enhanced security measures and is compliant to the RBI mandated 2-Factor authentication
- User friendly and smooth adaptability
- Simplified architecture & transaction flow reduces transaction time, resulting in faster transaction processing and reduction in drop-outs
- Customer Experience: During the online payment the cardholder's authentication data is collected in a secured manner

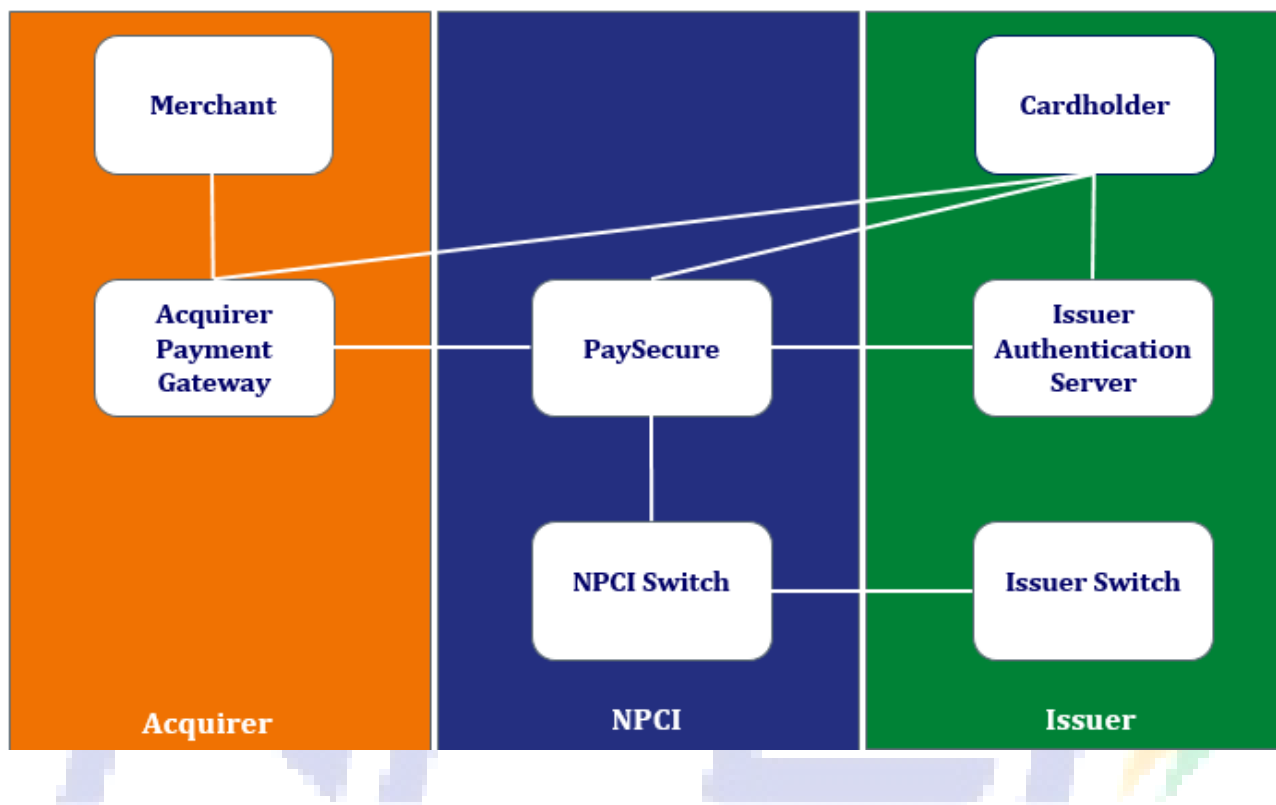
The solution will involve use of dynamic One-time-Password/Static PIN as an additional factor of authentication for e-commerce transactions on internet using RuPay Debit cards and Credit Cards.

RuPay has adopted the PaySecure solution for enabling a second factor authentication in the web based channels. The objective is to improve both Cardholder and Merchant confidence in Internet purchases and to reduce disputes and fraudulent activity related to the use of Debit cards and Credit Cards online.

PaySecure can be used at various internet accessible devices like Personal or shared computers, Mobiles and Tables. PaySecure is functional for RuPay cardholders of any issuing bank and at any merchant website which are integrated to RuPay eCommerce solution.

1.2 Participants in the RuPay eCommerce transaction cycle

The RuPay e-Commerce architecture involves the following constituents:



1.2.1 Cardholder:

Card holder means any customer in possession of a payment card (RuPay Debit cards and Credit Cards)

1.2.2 Merchant:

The merchant website which has online shopping feature enabled i.e. the website permit customers to purchase goods/products/services online and accepts payments in an electronic manner using debit cards, credit cards, net banking etc.,

1.2.3 Acquirer Bank Payment Gateway:

A payment gateway facilitates the transfer of information between a payment portal (such as a website, mobile phone or IVR service) and the Front End Processor or acquiring bank. Payment gateways protect card details by encrypting sensitive information, such as card numbers etc., to ensure that information is passed securely between the customer and the merchant and also between merchant and the payment gateway.

1.2.4 Issuer Authentication Server:

The IAS(Issuer Authentication Server) will be responsible to confirm the card holder authenticity. For the e-Commerce solution, the customer is re-directed to Issuer bank's IAS(Issuer Authentication Server) module maintained and managed completely by Issuing bank for the authentication purpose. Issuing bank will use any authentication method defined as per bank policy. NPCI recommends to use dynamic OTP to authenticate the cardholder. The issuer bank would be responsible for properly authenticating the identity of the cardholder and confirming the correct status of authentication back to NPCI.

Note: *"It is advisable that Mobile number details should be maintained on IAS and updated daily so that Issuer switch won't be queried each time for every transaction to get the mobile number details."*

1.2.5 Issuer Switch:

Customer's information along with a tag to indicate successful authentication/PIN Captured will be shared with the Issuing bank to be routed to the bank's switch wherein the bank uses this information to authorize/decline the transaction according to pre-defined rules.

1.2.6 NPCI PaySecure System:

This forms the core of the whole NPCI e-Commerce solution. This module is responsible for activities such as receiving card information from payment gateway, providing the mechanism for re-directing customer to issuer page for authentication etc.

1.2.7 NPCI Switch:

The switch is maintained and operated by NPCI for all electronic transactions ATM, POS etc. For e-Commerce purposes, switch would be required for routing information from NPCI to Issuing Banks.

1.3 Defining roles of Entities

1.3.1 Role of Merchant

- Perform integration with the acquirer using acquirer's API.
- Send the purchase and card related information to the acquirer.
- Redirect/Transfer browser control to acquirer to complete authentication process
- Receive authorization response information from Acquirer
- Present the receipt page to the customer and deliver the goods / service upon confirmation of payment from acquirer.

1.3.2 Role of Acquirer

- Acquirer need to integrate their system to PaySecure System using NPCI's API. SOAP (Simple Object Access Protocol) web services will be used for messaging between Acquirer and PaySecure system.

- Acquirer to integrate merchant to acquirer's system using Acquirer's API.
- To perform merchant authentication before sending the data to NPCI.
- Guide merchants on the best practices that need to be adopted.
- Settlement and reporting with the merchants.

1.3.3 Role of Issuer

- Issuer verifies the card/cardholder as per the current authentication and authorization process.
- Decrypts & Parse the ISO block received in the ISO 8583 message from NPCI to check the presence of authentication tag and other data elements
- Issuer to authorize or decline the transaction.
- Park the funds related to the authorized transactions and service fees/charges, if any, in the settlement account.
- Liability of all e-Commerce fraud transactions lies with the issuer.
- Issuer authenticates the cardholder
- It is recommended to decline/drop the financial transaction if customer clicks back button or reload (F5) during authentication process (i.e. IAS Page).

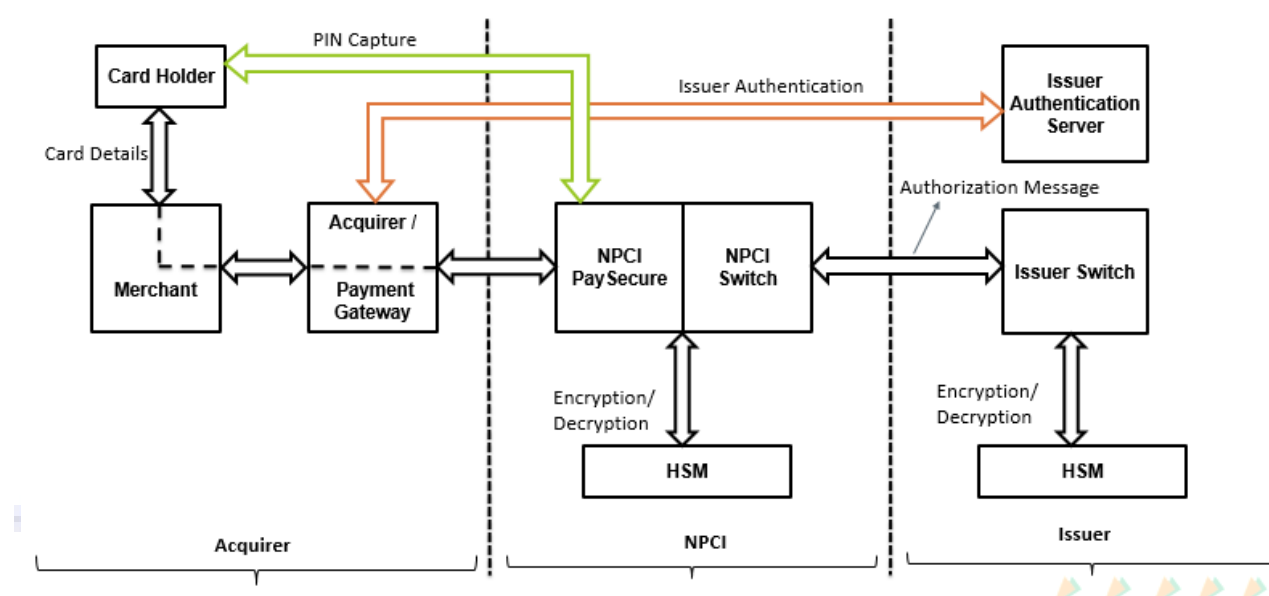
1.3.4 Role of the Cardholder

- To identify and select the goods or service on the merchant website
- To fill in the purchase form (customer name, phone number, email id, delivery address etc.)
- To select the payment option
- Upon prompt, enter the card number, expiry date and cvd2 on the payment page.
- Card holder to Authenticate with Issuer
- Enter the credentials (OTP/PIN) which he/she will prompt by Issuer/PaySecure page on authentication page.

1.3.5 Role of NPCI

- NPCI will certify acquiring and issuing banks system for exchanging data between the bank and NPCI over web services calls.
- NPCI will establish secure communication link between acquirers and issuers for processing the eCommerce transaction.
- Form ISO 8583 message packet post receiving Authorize API call along-with tag element (indicating successful authentication)
- To route authorization request message with Tag elements to the issuer
- Guide acquiring banks on merchant certification process, merchant authentication best practices.
- Guide acquiring banks on the best practices that the merchant can follow.
- To perform Geo location on the Cardholder's IP address.
- Conducts Clearing & Settlement amongst various stakeholder
- Coordinates for disputes for transactions processed using NPCI system.

1.4 Architecture for authentication of RuPay online transactions



Key Features

- Acquirer Bank (Payment Gateway) & Issuer Authentication Server (IAS) are integrated with PaySecure using Web Services API Calls (SOAP) and it doesn't require any separate software/plugin.
- Friction less payment transaction and integration process
- In addition to authentication process, PaySecure handles facilitates authorization leg by connecting to NPCI Switch which in turn is connected to respective Issuer switch and completes authorization process by generating ISO 8583 message and receiving the response from Issuer switch.
- For acceptance of RuPay eCom transaction, acquirer bank required to make changes only at Payment Gateway.

- Depending upon Issuer Bank requirement/policies, authentication flow can be customized.
 1. Authentication with Issuer Bank (Authentication details captured and validated by Issuing bank like OTP, Net banking credentials etc.)
 2. Authentication with PaySecure (Authentication details captured by NPCI and validated by issuing bank like ATM PIN.)
- No registration process.

1.5 High-Level Steps for the Acquirer:

1. Card Credentials details like Card No, Expiry Date, and Card Verification Data 2 (CVD2) shall be captured either at Merchant/Aggregator or Payment Gateway, it depends on the integration between Merchant/Aggregator and Acquirer Bank Payment Gateway. Considering that Card Credentials captured at Merchant end and submit to Payment Gateway along with purchase information, transaction flow will be as below:
2. On receipt of merchant payment request, Acquirer to check the first nine digits of the card number, called a BIN, to determine if the BIN is enrolled for RuPay eCommerce transactions by Issuing bank and also return the transaction flow indicator.
3. Acquirer submits a SOAP web-service call to initiate a transaction.
4. Based on Issuer Bank flow, acquirer bank follow either Iframe based communication or simple redirection approach
 - a) Redirection Approach
 - i. Complete browser re-direction to Issuer/PaySecure using POST method.
 - ii. Issuer completes authentication and return control back to acquirer again via complete browser re-direction, to submit for authorization.
 - b) Iframe Approach
 - i. Via Java Script function call, it display a modal Iframe which is hosted by NPCI PaySecure system.
 - ii. PaySecure gives control to Issuing bank who loads the IAS page, sends the OTP, if OTP is authentication method and authenticates the customer. The result is shared with NPCI.
 - iii. Acquirer is supplied a response from modal dialog as it closes indicating that OTP was authenticated successfully and transaction is ready for authorization
5. Acquirer submits authorization request to PaySecure via a SOAP based web service call that creates the ISO message, adds the successful authentication tags and sends

to the issuer switch for authorization. Response is routed back to merchant through the acquirer.

6. Merchant present the receipt page and confirm the payment/order status to the customer.

2. Transaction Flow

In PaySecure system, transactions can be processed in two different approach which is purely dependent on Issuer Bank's requirement and policies.

High Level Transaction Flow

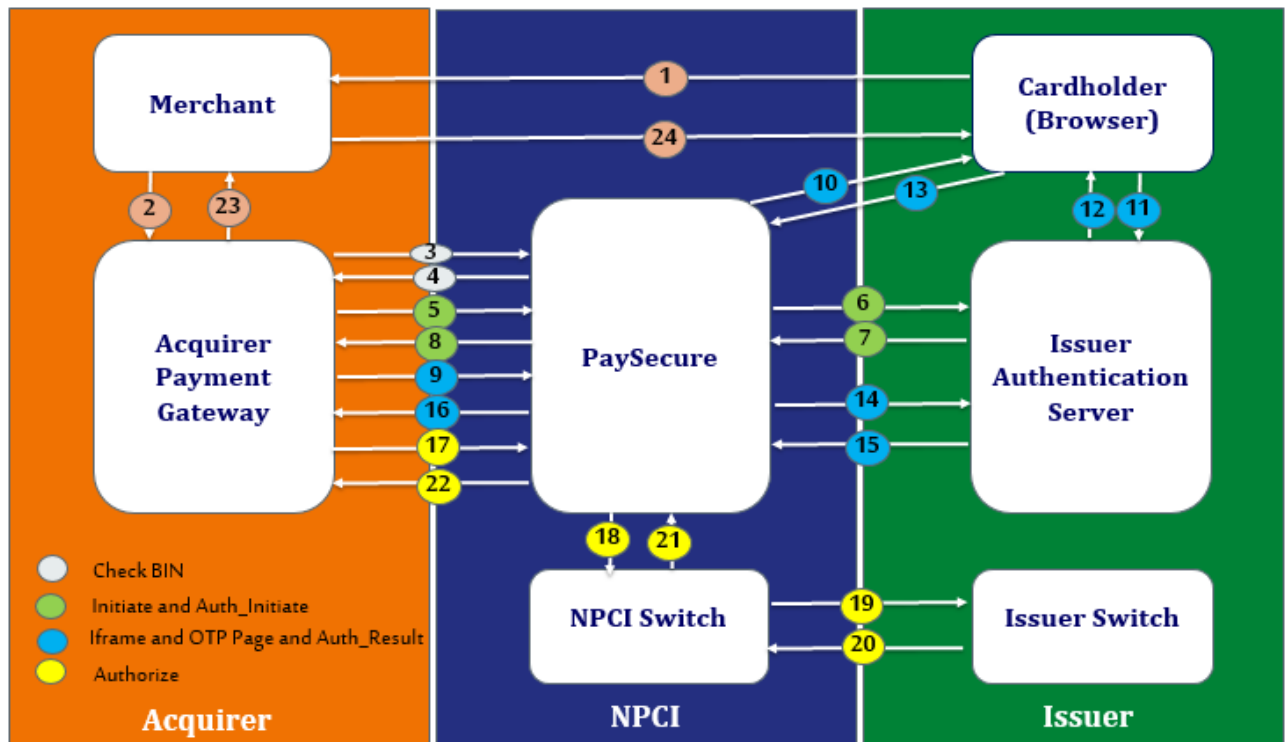
a) Iframe Approach

- i. PaySecure accepts "CheckBIN2" API call, to determine if the BIN is enrolled for eCommerce transactions by Issuing bank.
- ii. PaySecure accepts "Initiate" API Call to submit complete transaction request.
- iii. Via Java Script function call, it display a modal Iframe which is hosted by NPCI PaySecure system.
- iv. PaySecure gives control to Issuing bank who loads the IAS page, send the OTP, if OTP is authentication method and authenticates the customer. The result is shared with NPCI
- v. Acquirer is supplied a response from modal dialog as it closes indicating that OTP was authenticated successfully and transaction is ready for authorization
- vi. PaySecure accepts "Authorize" API call for authorization of transaction by Issuing bank.

b) Redirection Approach

- i. PaySecure accepts "CheckBIN2" API call, to determine if the BIN is enrolled for eCommerce transactions by Issuing bank.
- ii. PaySecure accepts "Initiate2" API Call to submit complete transaction request.
- iii. Complete browser redirection to Issuer/PaySecure using POST method.
- iv. Complete authentication and return control back to acquirer, to submit for authorization.
- v. Acquirer submit authorization request to PaySecure via a SOAP based web service call that creates the ISO message, adds the successful authentication tags and sends to the issuer for authorization. Response is routed back to merchant through the acquirer.

2.1 Iframe Flow

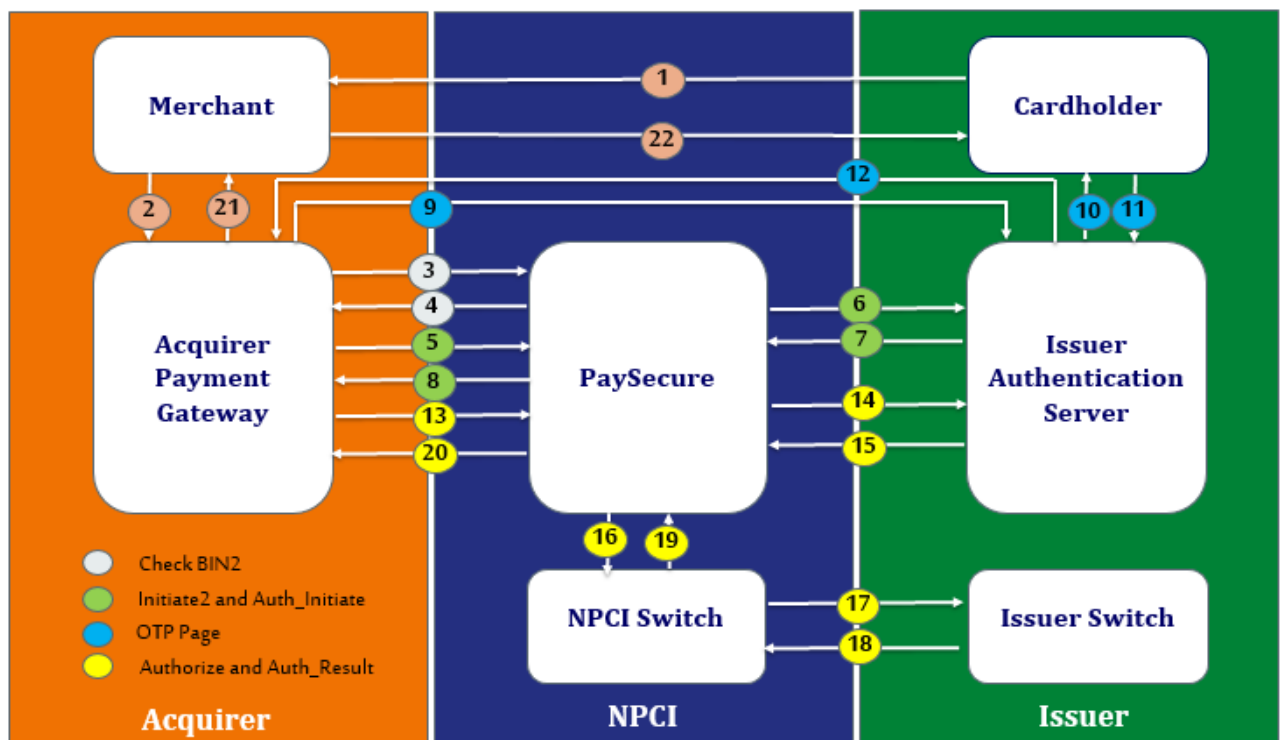


Step 1	<p>Cardholder accesses/log on to the merchant website, selects the goods/services that he/she intends to purchase, he/she adds the goods/services to the shopping cart.</p> <p>Cardholder now moves to the checkout process.</p> <p>On the checkout page (payment page) merchant website allows the cardholder to enter his card information on the merchant website.</p> <p>Cardholder selects the card type from the options provided by the merchant on the website. (card type example: RuPay)</p> <p>Now option to enter following will be provided by the merchant (acquirer / aggregator), it depends on acquirer integration process.</p> <p>✓ Card Type</p>
--------	--

	<ul style="list-style-type: none"> ✓ Card number ✓ Expiry date ✓ Card Verification Data 2 (CVD2) <p>Current functionality supports -13-19 card number length.</p> <p>Cardholder now clicks on the submit button.</p> <p><i>Merchant.js is downloaded from PaySecure application on cardholder browser during this process.</i></p>
Step 2	Merchant sends the complete payment data to Acquirer.
Step 3	Acquirer sends First 9 digits of card-number to PaySecure in Check Bin API call to determine if the card is eligible for eCommerce transaction. Upon a successful BIN check, the Acquirer sends in a request to Initiate the transaction to PaySecure.
Step 4	Acquirer system sends Initiate API call, which contains the payment data, to PaySecure system.
Step 5	Based on BIN PaySecure identifies transaction flow.
Step 6	PaySecure sends, Auth_Initiate API call, the request for authentication to the issuer authentication server for the verification of card number & mobile number availability.
Step 7	PaySecure receives the response from issuer authentication server.
Step 8	PaySecure responds to Initiate API call with transaction details allowing the Acquirer to continue with the payment process.
Step 9	Acquirer substantiates the PaySecure iFrame overtop of their payment page for Consumer interaction with PaySecure. PaySecure now has direct communications and control of the cardholder browser.
Step 10	PaySecure system redirects to issuer OTP Page. The Issuer now has direct communications and control of the consumer browser.
Step 11	<p>The Issuer provides cardholder with available authentication options.</p> <p>For example:</p> <ul style="list-style-type: none"> • Enter OTP
Step 12	<ul style="list-style-type: none"> • Cardholder enters OTP on issuer OTP page.
Step 13	Issuer validates OTP as entered by cardholder on OTP page.
Step 14	Issuer passes the control of iFrame and cardholder back to the PaySecure system. PaySecure server query's IAS server via Auth_Result API call to

	securely confirm the result and method of cardholder authentication.
Step 15	iFrame is closed, Acquirer is notified by PaySecure through browser only if OTP authentication status is successful.
Step 16	Acquirer completes any pre-authorization steps and send Authorization _API request to PaySecure.
Step 17	PaySecure will create the ISO message, as per Rupay specifications with Mandatory Tag elements and will send the authorization message to the NPCI switch.
Step 18	NPCI switch sends the authorization message to the issuer switch for authorization.
Step 19	Issuer switch will validate the ISO Message and will approve/decline the transaction. Issuer will send the response back to the NPCI Switch.
Step 20	NPCI Switch will send the response to the PaySecure system.
Step 21	PaySecure system sends the response back to the acquirer.
Step 22	Acquirer sends the response to the merchant website.
Step 23	Merchant displays appropriate message to cardholder.

2.2 Redirection Flow



Step 1	<p>Cardholder comes onto the merchant website, selects the goods/services that he/she intends to purchase, he/she adds the goods to the shopping cart.</p> <p>Cardholder now moves to the checkout process.</p> <p>On the checkout page (payment page) merchant website allows the cardholder to enter his/her card information on the merchant website.</p> <p>Cardholder selects the card type from the options provided by the merchant on the website. (card type example: RuPay)</p> <p>Now option to enter following will be provided by the merchant (acquirer / aggregator), it depends on acquirer integration process.</p> <ul style="list-style-type: none"> ✓ Card Type ✓ Card number ✓ Expiry date ✓ Card Verification Data 2 (CVD2) <p>Current functionality supports -13-19 card number length.</p> <p>Cardholder now clicks on the submit button.</p>
Step 2	Merchant sends the complete payment data to Acquirer.
Step 3	Acquirer sends First 9 digits of card-number to PaySecure in Check Bin2 API call to determine if the card is eligible for eCommerce transaction.
Step 4	Based on BIN, PaySecure identifies transaction flow. If BIN not enabled for redirection flow then same will be mentioned in CheckBin2 Response , then Acquirer needs to follow Iframe Flow . Upon Successful Bin Check PaySecure Responds to Acquirer.
Step 5	Acquirer system sends Initiate2_API call to transmit the payment data to PaySecure system.
Step 6	PaySecure sends Auth_Initiate API call to transmit the request for authentication to the issuer authentication server (IAS).
Step 7	PaySecure receives the response from issuer authentication server.
Step 8	PaySecure responds to Initiate2_API with transaction details along with Issuer OTP Page URL and Cardholder ID allowing the Acquirer to continue with the payment process.
Step 9	Acquirer redirects to the Issuer OTP Page URL received in Initiate2 API response for cardholder interaction with IAS. IAS System now has direct communication and control of the cardholder browser.
Step 10	The Issuer provides cardholder with available authentication options to the

	cardholder. For example: <ul style="list-style-type: none"> • Enter OTP
Step 11	Cardholder enters OTP on issuer OTP page.
Step 12	Issuer validates OTP as entered by cardholder on OTP page.
Step 13	**Issuer redirects cardholder browser back to Acquire system along with authentication status of cardholder.
Step 14	Acquirer completes any pre-authorization steps and sends Authorization_API request to PaySecure.
Step 15	PaySecure server queries IAS via Auth_Result API call to securely confirm the result of cardholder authentication.
Step 16	On Successful receipt of Auth_Result API response from IAS, PaySecure will create the ISO message with Mandatory Tag elements and will send the authorization message to the NPCI switch.
Step 17	NPCI switch sends the authorization message to the issuer switch for authorization.
Step 18	Issuer switch will validate the ISO Message and will approve/decline the transaction. Issuer will send the response back to the NPCI Switch.
Step 19	NPCI Switch will send the response to the PaySecure system.
Step 20	PaySecure system sends the response back to the acquirer.
Step 21	Acquirer sends the response to the merchant website.
Step 22	Merchant displays appropriate message to cardholder.

3. Transaction Processing Details

In PaySecure, there are two type of transaction flows. Both transaction flow needs to be implemented & supported at acquirer.

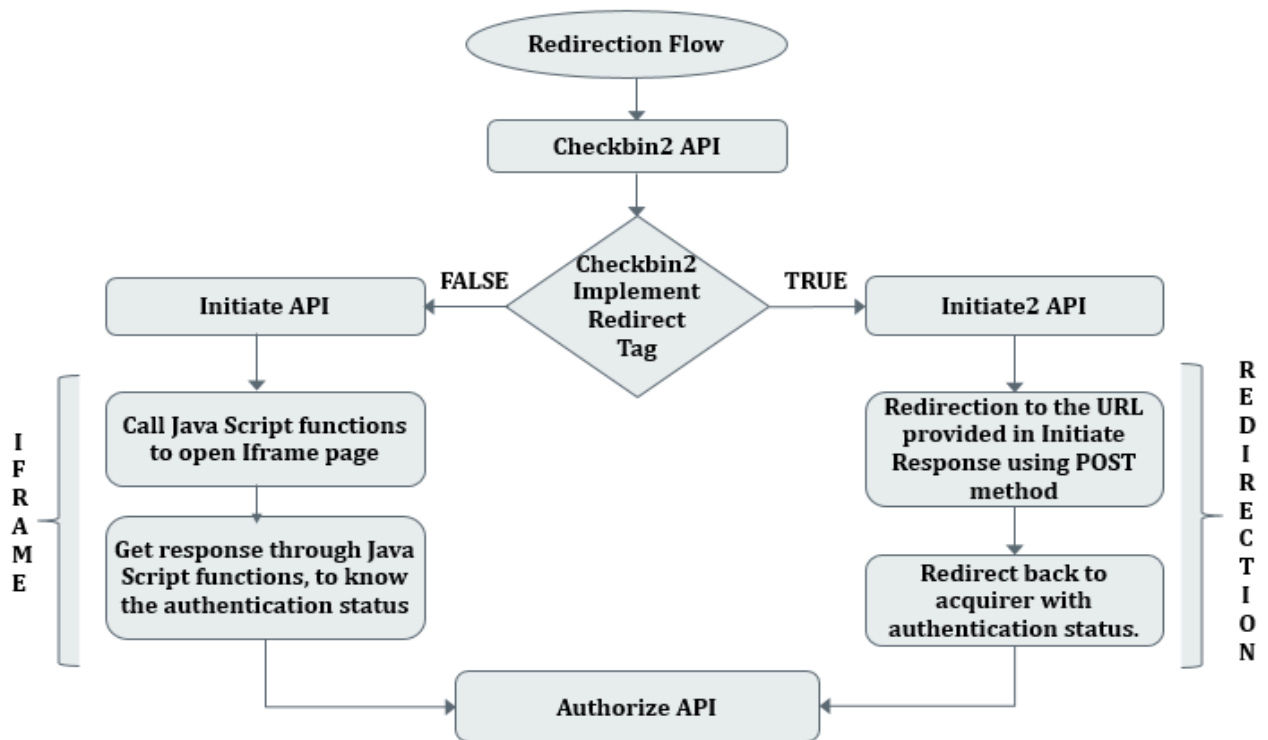
1. Iframe Flow
2. Redirection Flow

The acquiring banks which are already live on “Iframe Flow” is required to undergo certification process to enable the Redirection flow and acquiring bank which will be enabling RuPay eCommerce transactions via PaySecure for first time, have to be certified for both type of transaction flows i.e. Iframe and Redirection.

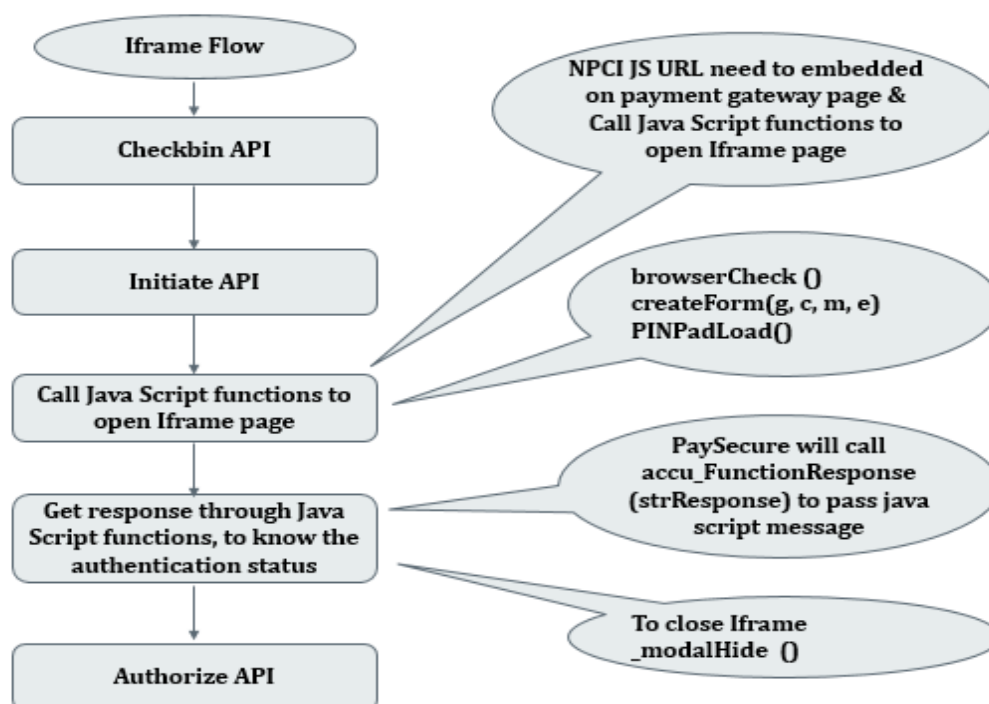
Both transactions flows (Iframe & Re-direction) will be continued to be supported at all Acquirers by the time all RuPay issuing banks migrate to redirection flow.

Below diagram illustrate integrated flow of Iframe and Redirection flow:

Integrated Iframe and Redirection Flow



Iframe Flow



3.1 Merchant Ecommerce Site

The cardholder initiates a payment process while shopping and by checking out on a Merchant's ecommerce site. The cardholder is asked to select a method of payment to initiate and complete a payment process for the goods or services to be purchased. One of the methods of payment is the RuPay Debit/Credit card.

3.2 Checkbin – API call

The CheckBin API call is used by the acquirer to determine whether Issuer Bank has enabled eCom service based on BIN. This web service is designed to contain minimal information and tuned for performance. The call must contain the first 9 digits of the card as some banks have subdivided 6 digit BINs into additional products out to the 9th digit. Refer Annexure B for message specification for CheckBin API call.

3.3 Checkbin2 – API call

The CheckBin API and Checkbin2 API are similar in terms of its core functionality of checking the eligibility of BIN for eCommerce transactions.

Only difference is that Checkbin2 API response contains "Redirect" Tag which confirms the typed of flow supported by respective Issuing bank and to be followed by acquirer to complete that particular transaction. If Redirect Tag value is "TRUE" then acquirer should flow Redirection Flow and if Redirect Tag is "FALSE" then acquirer should follow Iframe flow.

Refer Annexure B for message specification for CheckBin2 API call.

*Note: Once Acquirer bank is certified for re-direction flow, **CheckBIN API** call will be absolute for respective Acquirer and **CheckBIN2 API** will only be invoked for all transactions.*

3.4 Initiate – API call

The Initiate API call is used to securely exchange necessary information related to the card and transaction between the Acquirer and PaySecure. All the necessary Data elements required to create ISO message block is received from the acquirer bank payment gateway in this API call. Full card number is supplied to PaySecure, this card number will drive subsequent product functions. PaySecure will return a tran_ID (transaction ID) that is unique to the transaction and will be used throughout the lifecycle of the transaction including authentication, authorization and refunds. This transaction is also part of ISO message block and send to the issuer in the online message to the issuer i.e. Transaction ID is available to both acquirers and issuers. Additionally PaySecure will provide a GUID, Modulus, and Exponent that are related to the security of the iFrame and must be passed unaltered to the consumer's browser during the loading of the iFrame.

Refer Annexure B for message specification of Initiate API call.

3.5 Initiate2 – API call

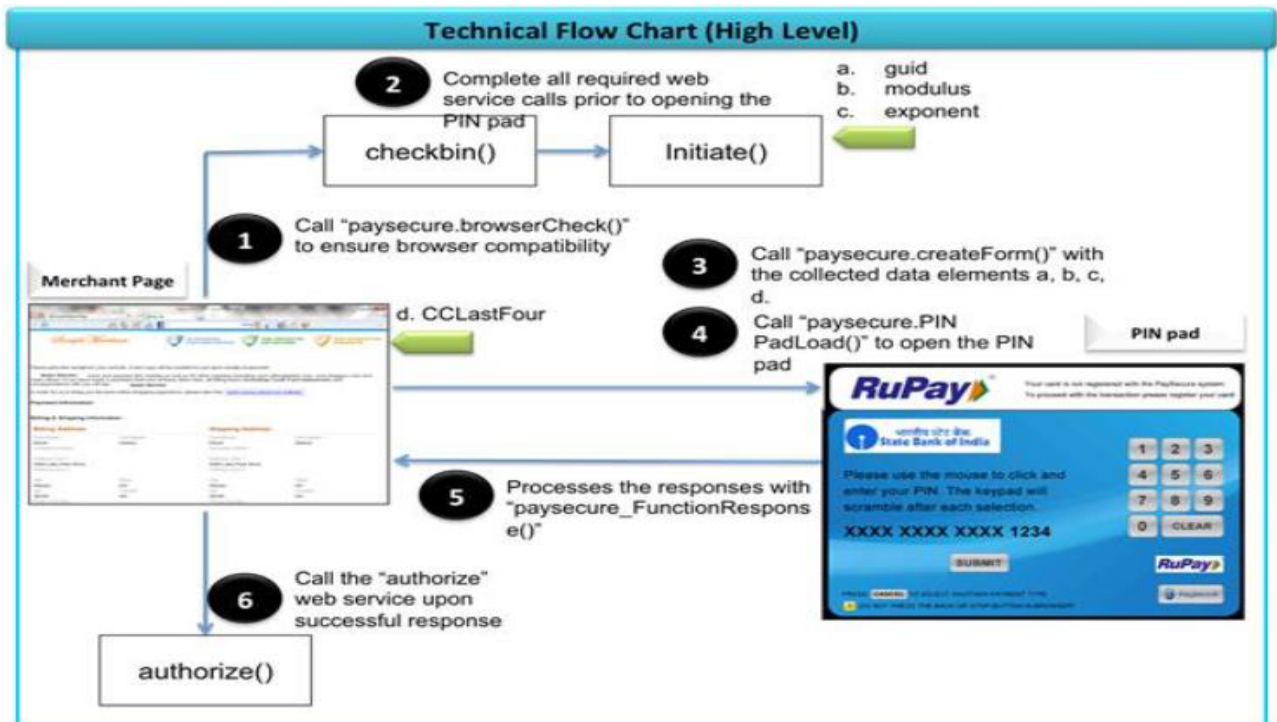
The Initiate2 API call is used to securely exchange necessary information related to the card and transaction between the Acquirer and PaySecure. All the necessary Data elements required to create ISO message block is received from the acquirer bank payment gateway in this API call. Full card number is supplied to PaySecure, this card number will drive subsequent product functions. PaySecure will return a tran_ID (transaction ID) that is unique to the transaction and will be used throughout the lifecycle of the transaction including authentication, authorization and refunds. This transaction is also part of ISO message block and send to the issuer in the online message to the issuer i.e. Transaction ID is available to both acquirers and issuers.

On comparing “Initiate” API call with “Initiate2”, 3 new tags has been added in request i.e. BrowserUserAgent, IP Address, and HTTPAccept. Additionally PaySecure will provide a GUID, hkey & RedirectURL in “Initiate2” response back to Acquirer, to which the URL of cardholder needs to be redirected to initiate authentication process.

Refer Annexure B for message specification of Initiate2 API call.

3.6 Iframe Flow – Browser/App Interaction

Integration of the PaySecure PINPad, requires initiating a session via the acquirer web services, opening an Ajax-based modal popup window (iFrame) hosted by PaySecure, receiving the response from the PaySecure PIN Pad and closing the session.



Technical Flow Chart

In the above graphic flow chart, step 1 and 5 utilizes the Web Service URL while step 2 and 3 utilizes the Merchant Scripts URL to directly communicate to/from the RuPay PaySecure PINPad. Step 4 will be a function that resides on the merchant's page that will process responses from the PIN Pad.

To have the PINPad displayed on the client's machine, these JavaScript commands must be made:

1. On acquirer PG page, embedded java script file in header section.

```
<script language="javascript" src="<Merchant Scripts URL>"
type="text/javascript"> </script>
```

2. Call "Acculynk.browserCheck()" and ensure a "true" is returned before proceeding
3. Call "Acculynk.createForm(g, c, m, e)" where

g = guid received from the initiate() call

c = last four digits of the card no, which is collected by the merchant

m = modulus received from the initiate() call

e = exponent received from the initiate() call

4. Call "Acculynk.PINPadLoad()"
5. The javascript function "accu_FunctionResponse(strResponse)" must be created by the acquirer in order to wait for a response by the PINPad.
6. Once final response received, call "_modalHide()" to close iframe interaction.

Java Script & HTML Changes

The example below shows the mandatory JavaScript and html elements that need to reside on the merchant page that will open/close and communicate with the Iframe:

1. `<script language="javascript" src="<Merchant Scripts url>" type="text/javascript">`
 //This script has to be written in the header part of the html code. It will be pulled from Paysecure to merchant browser
`</script>`

2. `<script language="javascript" type="text/javascript">`
 //reads the response back from PaySecure


```
function accu_FunctionResponse(strResponse){
alert("this is the response that was received " + strResponse);}
//Actual code has been given in the below table for Accu_FunctionResponse

//checks browser compatibility
Acculynk.browserCheck();

//preps the PIN Pad for opening
Acculynk.createForm("77AC...EF34", "9339", "32498CBC7E...ED78D", "010001");
//these argument values needs to be replaced with actual g,c,m,e

//opens the authentication and PIN Pad for consumer
Acculynk.PINPadLoad();

//closes the PIN Pad
Acculynk._modalHide();
</script>
```

3. include the below html tags

```
<center>
<div id="accu_screen" style="display: none;"></div>
<div id="accu_keypad" style="display: none;"></div>
<div id="accu_form" style="display: none;"></div>
<div id="accu_loading" style="display: none;"></div>
<div id="accu_issuer" style="display: none;"></div>
</center>
```

*replace **Merchant Scripts URL** with

["https://cert.mwsrec.npci.org.in/MWS/Scripts/MerchantScript_v1.0.js"](https://cert.mwsrec.npci.org.in/MWS/Scripts/MerchantScript_v1.0.js), for test setup.

["https://mwsrec.npci.org.in/MWS/Scripts/MerchantScript_v1.0.js"](https://mwsrec.npci.org.in/MWS/Scripts/MerchantScript_v1.0.js), for production setup.

Below is an example of the code that needs to be added to the merchant page.

JavaScript Code	<pre> /* Function to be written by merchant to process response */ function accu_FunctionResponse(strResponse){ switch (strResponse) { case 'ACCU000': //PIN was received so merchant can process the authorization Acculynk._modalHide(); break; case 'ACCU200': //user pressed 'cancel' button Acculynk._modalHide(); break; case 'ACCU400': //user was inactive Acculynk._modalHide(); break; case 'ACCU600': //invalid data was posted to PaySecure Acculynk._modalHide(); break; case 'ACCU700': //Card Issuer Error Acculynk._modalHide(); break; case 'ACCU800': //general catch all error Acculynk._modalHide(); break; case 'ACCU999': //modal popup was opened successfully //no action necessary, but open for merchant to use break; default: break; } } </pre>
HTML Code	<pre> <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd"> <html> <head> <script language="javascript" src="<Merchant Scripts url>" type="text/javascript"></script>etc. </head> <body> <!--The code in the '<center>' tag cannot be within a '<form>' tag --> <center> <div id="accu_screen" style="display: none;"></div> <div id="accu_keypad" style="display: none;"></div> <div id="accu_form" style="display: none;"></div> <div id="accu_loading" style="display: none;"></div> <div id="accu_issuer" style="display: none;"></div> </pre>

	<pre> </center> <form> <input type="button" value="Start PIN Pad" onclick="if(Acculynk.browserCheck()){ Acculynk.createForm('ABC', '4123', 'ACB', '010001'); Acculynk.PINPadLoad();} " /> </form> </body> </html> </pre>
--	--

The table below describes the JavaScript functions that the merchant will use to create, open, and then close the PIN Pad.

Function	Description
Acculynk.browserCheck();	Function that checks to see if the client browser is compatible with the PaySecure PIN Pad. The function returns true/false: true: compatible false: not compatible
Acculynk.createForm(g, c, m, e)	Function that will prepare the form that will be posted to PaySecure, where: g = guid returned from the initiate() web service c = last four digits of card, which is collected by the merchant m = modulus returned from the initiate() web service e = exponent returned from the initiate() web service
Acculynk.PINPadLoad()	Grays out the background of the screen and opens the PaySecure PINPad for the cardholder.
Acculynk._modalHide();	Hides the PaySecure PINPad and enabled the grayed background.

Below are the codes that are used that the merchant/acquirer will act upon. These codes will be passed back to the function named "accu_FunctionResponse(strResponse)," which is a function that the merchant/acquirer has to write.

Response Code	Description	Required Action
ACCU000	PIN collected / Authentication completed.	Move to Authorization (Authorize API Call)
ACCU100	Authentication Failed	Decline transaction
ACCU200	Cardholder pressed Cancel button	Decline transaction

ACCU400	Cardholder inactivity timeout	Decline transaction
ACCU600	Invalid data was posted to the PaySecure PIN Pad/Issuer	Decline transaction
ACCU800	General Error Encountered	Decline transaction
ACCU999	PINPad was successfully opened	Start the browser session of 7 Minutes.

Time out scenarios in Java Script Interaction

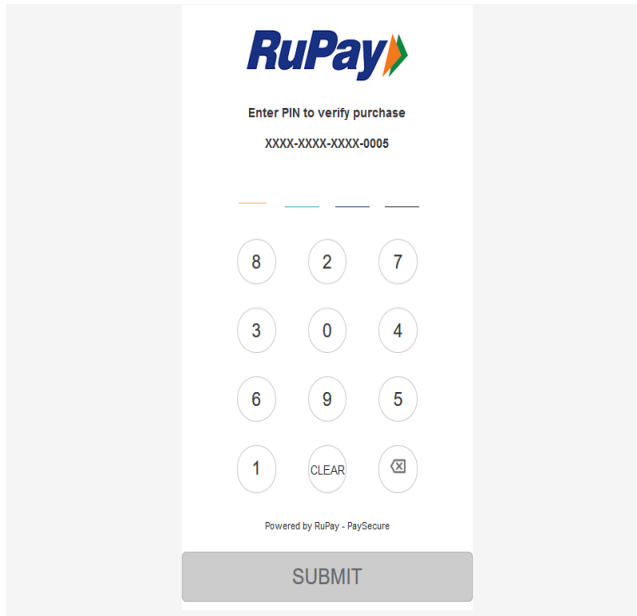
Merchant-side script must exist to wait for some period of time and close the iFrame if the timeout is reached. This timeout must be enough to account for cardholder authentication at the issuer and/or PIN capture at PaySecure.

3.7 Redirection Flow – Browser/App Interaction

To render the Issuer Authentication page or PaySecure PINpad, the acquirer will redirect the cardholder completely to the “RedirectURL” that was provided during the “initiate2” API call response. The redirection will be accomplished with POST method only and values are in hidden form variables.

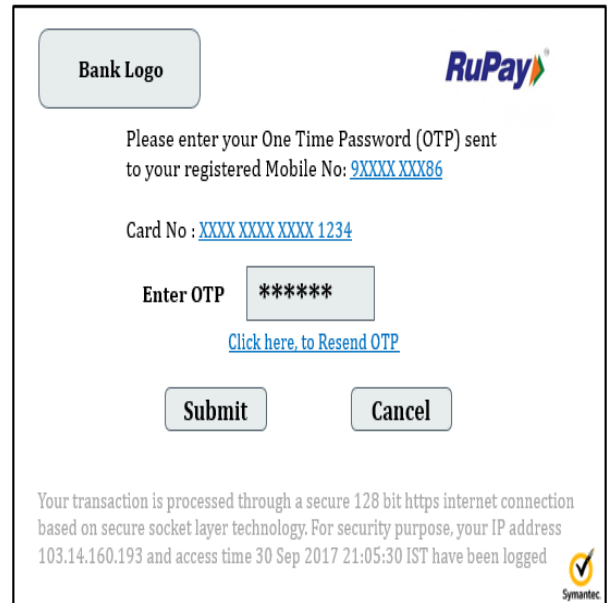
Once the cardholder has completed submission of their PIN or authentication details on issuer authentication page, the cardholder will be redirected back to the acquirer using the URL send in the AccuReturnURL field by Acquirer. PaySecure will determine the viewport (desktop vs mobile) and render the PINpad in an adaptive manner.

Please refer below example of the desktop version. Layout and design of authentication page may differ for different issuers keeping RuPay & Bank’s logo in standard format. Below is the sample of PIN entry screen and Issuer authentication page.



The image shows a RuPay PINPAD interface. At the top, it says "Enter PIN to verify purchase" followed by a masked PIN "XXXX-XXXX-XXXX-0005". Below this is a numeric keypad with digits 0-9, a CLEAR button, and a backspace button. At the bottom, there is a "SUBMIT" button. The text "Powered by RuPay - PaySecure" is visible above the submit button.

PaySecure – Scrambling PINPAD Page



The image shows an Issuer Authentication Page. It features the RuPay logo and a "Bank Logo" placeholder. The text asks the user to enter their One Time Password (OTP) sent to their registered Mobile No: 9XXXX XXX86. Below this, it shows the Card No: XXXX XXXX XXXX 1234. There is an "Enter OTP" field with a masked input "*****" and a "Click here, to Resend OTP" link. At the bottom, there are "Submit" and "Cancel" buttons. A footer message states: "Your transaction is processed through a secure 128 bit https internet connection based on secure socket layer technology. For security purpose, your IP address 103.14.160.193 and access time 30 Sep 2017 21:05:30 IST have been logged". A Symantec logo is also present.

Issuer Authentication Page

Based on Initiate2 API Response, acquirer PG is required to construct redirection request along with required parameters and post to the RedirectURL received.

Redirection Request Parameters:

Method: POST

Parameter Name & Type: Hidden

Parameter	Description
AccuCardholderId	Generated by Issuer/PaySecure and extract value from Initiate2 API Response in "RedirectURL" tag
AccuGuid	Generated by PaySecure and extract from Initiate2 API Response in "RedirectURL" tag.
AccuReturnURL	Fully qualified Acquirer PG's URL, Issuer will use to redirect the cardholder back, upon completion of the authentication.
session	To handle session at acquirer end, acquirer can populate any value and same value will be echo back in response. This value is required to be generated by Acquirer & should be unique for each transaction.
AccuRequestId	To avoid tampering of request data in transit, the acquirer is required to provide a hash code for the issuer to validate. This hash code can be generated by hashing Transaction ID, AccuCardholderId, AccuGuid, and session parameter value.

Redirection Response Parameters:

Method: POST

<u>Parameter</u>	<u>Description</u>
AccuResponseCode	Code that provides authentication status to drive business rules on how to process the cardholder transaction. This data will be POST to URL of the AccuReturnURL.
session	To handle session at Acquirer end, Issuer/PaySecure, same value which is received in redirection request, same value will be sent in response. This data will be POST to URL of the AccuReturnURL.
AccuGuid	Issuer will be echo back in response. This data will be POST to URL of the AccuReturnURL.
AccuRequestId	To avoid tampering of response data in transit, the issuer is required to provide a hash code for the acquirer to validate. This hash code can be generated by hashing Transaction ID, AccuGuid, session and AccuResponseCode parameter value.

**Please ensure that no custom variables are prefixed with "ACCU" as those are reserved for RuPay.*

1. Let's assume the response from the initiate2 for "RedirectURL" is below:

IssuerBankURL?ParameterName1=Value&ParameterName2=Value&ParameterName3=Value

https://issuerbank.com/redirect_ias/Home/IssuerReg?AccuCardholderId=89172389132&AccuGuid=6089d50e-e012-1160-8b3b-0ab8de556755&AccuHkey=5629y50g-e743-0022-5i2b-9aw8de632896

2. To construct redirection request, following 5 information will be extracted from Initiate2 API Response under RedirectURL tag
 - a) **Redirection Request URL** : https://issuerbank.com/redirect_ias/Home/IssuerReg
 - b) **AccuCardholderId**=89172389132
 - c) **AccuGuid**=6089d50e-e012-1160-8b3b-0ab8de556755
 - d) **AccuHkey**=5629y50g-e743-0022-5i2b-9aw8de632896
 - e) **TransactionId**=400000000000000000000318783342

Page 29 of 106

3. Finally, simply POST the data to the URL:

AccuCardholderId=89172389132

AccuGuid=6089d50e-e012-1160-8b3b-0ab8de556755

AccuReturnURL=https%3A%2F%2Facquirerbank.com%2Fsupport_redirect%2FCheckout%2FPinPadResult%2Fredirect

session= CwzmsrQN2f15faUUOmHIHkGefRcg8BgHPnvx9E3pW7MNkwC6GUmi!-2058637968!30723374!1114682342431

AccuRequestId=ZGU2NmExMzM3N2M0ZjkyYTM2YzQxMWMwYTdjY2Q5ZDRlYzM2Y2E0ZDdhNmMxZTY3OWU4YzBlOTdmMzlmNDlkYw==

Note: By default, forms data is generated with URL encoding by browser engine and there is no separate URL encoding is required by Acquirer.

Endpoint: https://issuerbank.com/redirect_ias/Home/IssuerReg

Method : POST

Request Body (url encoded):

AccuCardHolderId=89172389132&AccuGuid=6089d50e-e012-1160-8b3b-0ab8de556755&AccuReturnURL=https%3A%2F%2Facquirerbank.com%2Fsupport_redirect%2FCheckout%2FPinPadResult%2Fredirect&session=CwzmsrQN2f15faUUOmHIHkGefRcg8BgHPnvx9E3pW7MNkwC6GUmi!-2058637968!30723374!1114682342431&AccuRequestId=ZGU2NmExMzM3N2M0ZjkyYTM2YzQxMWMwYTdjY2Q5ZDRlYzM2Y2E0ZDdhNmMxZTY3OWU4YzBlOTdmMzlmNDlkYw==

4. Once authentication is completed, below is the data that will be post back to the acquirer.

AccuGuid=6089d50e-e012-1160-8b3b-0ab8de556755

session= CwzmsrQN2f15faUUOmHIHkGefRcg8BgHPnvx9E3pW7MNkwC6GUmi!-2058637968!30723374!1114682342431

AccuResponseCode=ACCU000

AccuRequestId - Hashing technique used will be HMACSHA256. Then Acquirer needs to match hash using the same values and Key (hkey). If the hash messages do not match, the acquirer is required to decline transaction and do not proceed further with authorization request to NPCI.

Endpoint: https://acquirerbank.com/support_redirect/Checkout/PinPadResult/redirect

Request Body (url encoded):

session= CwzmsrQN2f15faUUOmHIHkGefRcg8BgHPnvx9E3pW7MNkwC6GUmi!-2058637968!30723374!1114682342431&AccuResponseCode=ACCU000&AccuGuid=6089d50e-e012-1160-8b3b-0ab8de556755&AccuRequestId=ZjgyMGJwYzcxY2I2NWw4OTIzOWUyYTBhNzE3MDY2NzIzZmJlYjc0ZmVmOTc5NWJlMzA4ZjM5MjIjZGRlYjVhNA==

Issuer Hash Code Process for AccuRequestId in Re-direction Response

hkey : 5629y50g-e743-0022-5i2b-9aw8de632896

TransactionId : 400000000000000000000318783342

(TransactionId value to be extracted from "tran_id" parameter in Initiate2 response received by Acquirer)

AccuGuid : 6089d50e-e012-1160-8b3b-0ab8de556755

session : CwzmsrQN2f15faUUOmHIHkGefRcg8BgHPnvx9E3pW7MNkwC6GUmi!-2058637968!30723374!1114682342431

AccuResponseCode : ACCU000

- Concatenate values of the request; each field is separated by the "&" symbol.
40000000000000000000000318783342&6089d50e-e012-1160-8b3b-0ab8de556755&CwzmsrQN2f15faUUOmHIHkGefRcg8BgHPnvx9E3pW7MNkwC6GUmi!-2058637968!30723374!1114682342431&ACCU000
- Generate HMACSHA256 object using the HEKY as the pre-shared key.
- Compute the hash of the data using this hash algorithm.

HMAC_SHA256("key", "Message")

HMAC_SHA256("5629y50g-e743-0022-5i2b-9aw8de632896",
40000000000000000000000318783342&6089d50e-e012-1160-8b3b-0ab8de556755&CwzmsrQN2f15faUUOmHIHkGefRcg8BgHPnvx9E3pW7MNkwC6GUmi!-2058637968!30723374!1114682342431&ACCU000")

f820b0c71cb65c89239e2a0a717066723fbeb74fef9795be308f3929cddeb5d4

- Convert the binary data to base64 encoded string. For the above binary value, the final base64 encoded string should be:

ZjgyMGIwYzcxY2I2NWw4OTIzOWUyYTBhNzE3MDY2NzIzZmJlYjc0ZmVmOTc5NWJlMzA4ZjM5MjliZGRlYjVkdA==

Note:

- ** By default, forms data is generated with URL encoding by browser engine and there is no separate URL encoding is required by Issuer.
- ** Sequence of parameters in hashing input, will be same as given in example.
- ** Acquirer & Issuers are strongly recommended not to pass/use/communicate secrete Hash Key i.e. **hkey** & **TransactionID** in any way over browser communication or to any third party.
- ** Acquirer & Issuers are liable to manage the secrecy of Hash Key i.e. **hkey** & **TransactionID** at their respective environment/infrastructure.

Authentication Response Status Code

Resp Code	Description	Next Action
ACCU000	PIN collected / Authentication completed	Move to Authorization

ACCU200	Cardholder pressed Cancel button	Decline transaction
ACCU400	Cardholder inactivity timeout	Decline transaction
ACCU600	Invalid Data received posted	Decline transaction
ACCU700	Duplicate Data posted or Session already expired	Decline transaction
ACCU800	General Error Encountered	Decline transaction

3.8 Authorize – API call

Authorize API call is initiated by the Acquirer PG once PIN is captured by PaySecure or authentication is successfully completed & validated by the IAS and successful response for the same has been send back to Acquirer PG.

The authorize API call is used to initiate the authorization message. Authorize API call triggers the creation of ISO message block by the PaySecure system. Upon receipt of Authorize API call PaySecure collects all the transaction details along-with a tag element (indicating successful authentication validation confirmation by IAS) to create ISO message block and sends it to the NPCI switch which in turn forwards to the respective Issuer switch.

The “authorize” API call is used to authorize a transaction for which authentication was successfully validated. Only one authorization is allowed per initialized transaction.

1. Acquirer Payment gateway performs all internal business pre-authorization processes
2. Acquirer Payment gateway calls authorize API call passing in Transaction Id, PAN, Transaction Amount, card Expiry Date, CVD2, currency, and other custom fields such as order Id to the acquiring bank payment gateway.
3. PaySecure combines the authorization information with tag element and requests authorization confirmation from issuing bank via NPCI switch.
4. Issuing bank sends the response to NPCI switch; NPCI switch in turn sends the response back to PaySecure.
5. PaySecure sends successful response along with approval code to the acquiring bank payment gateway.
6. Merchant displays receipt page to cardholder.

Refer Annexure B for message specification of Authorize API call

4. Transaction type

RuPay eCommerce platform supports purchase transaction type both for SMS and DMS transactions. SMS/DMS is indicated by MTI field in BIN update file from RTGS.

5. Transaction-Status

The “transactionstatus” API call is used to request the status of a transaction from the PaySecure system. Acquirer PG is recommended to invoke this API call, for all transactions

which has timed out due to no response for Authorize request from PaySecure and within 24 hrs from the original transaction time.

6. Refund Processing

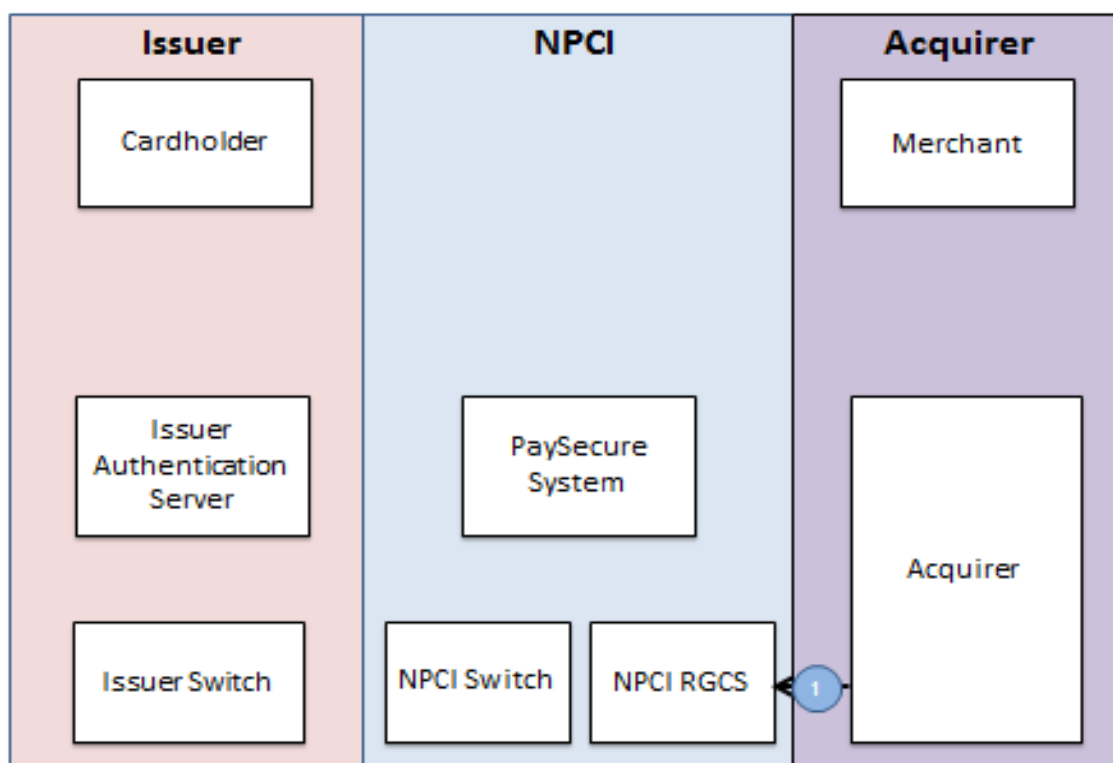
E-commerce Refund is a financial transaction originated at the merchant's website that instructs the issuer to credit the cardholders account for the return of goods, tickets etc. Refund amount should be less than or equal to the amount of original purchase.

E-commerce refund is carried out as follows:

1. E-commerce refund is carried out offline i.e. through RGCS system. This essentially means that refund transaction is to be processed only in clearing and settlement cycle.
2. While a customer is doing an E-Commerce purchase, a Transaction Id is generated from the PaySecure system which gets stored in DE 48 of ISO message. This transaction Id is for the particular transaction.
3. When a customer wants to do the Refund of the previous transaction, he needs to request/select for refund.
4. Once a customer initiates a refund, the merchant portal will provide the following details to the Acquirer payment GW:
 - Transaction ID (mandatory)
 - Original Transaction Date Time (Same as DE12 at acquirer end)
 - Refund Amount
5. Based on the above parameter acquirer will retrieve the original transaction and shall ensure that the refund amount is less than original purchase amount. After all these checks acquirer will generate a refund message for clearing cycle as described in NPCI Clearing and Settlement manual.
6. The issuer by seeing the presentment data will process the refund and credit the customer's account.

Please refer "RuPay Global Clearing and Settlement Technical Message Specification" for refund file format.

7. Clearing and Settlement Process



Step 1	At the End Of the Day process, acquiring bank will download the transaction file from the Payment Gateway and will present the DMS eCommerce transaction to NPCI through RGCS system.
--------	---

Acquiring bank will use RuPay Global Clearing and Settlement System (RGCS) for clearing and settlement process, as done today for POS transactions.

Note: Acquirer will know about SMS and DMS transaction by looking at the MTI Identifier field. Based on this MTI identifier information Acquirer will present only DMS transaction to NPCI for settlement.

Please refer RuPay Global Clearing and Settlement Manual for detailed clearing and settlement process.

8. Dispute Resolution and Chargeback Process

For dispute and chargeback process refer “RuPay Dispute Management Rules and Regulations”

9. Transaction ID

In response of Initiate/Initiate2 API call, PaySecure sends Transaction ID to Acquirer which is unique for each transaction and the same transaction id would also be supplied to the issuer bank switch in the Online ISO message in data element 48 tag 61.

10. Time out scenarios and handling

The various timeout scenarios fall into the following categories:

10.1 Acquirer-to-PaySecure (API Communication)

- a. During Checkbin/Checkbin2, Initiate/Initiate2 API calls
 - i. Time-out during these API calls would result in the decline of respective transaction at PG end and prompting the cardholder for an alternate payment method or try again.
 - ii. The acquirer should return a failure response to the merchant when a timeout to PaySecure occurs.
- b. During authorization
 - i. PaySecure provides a “transaction_status” API call which enables acquirer PG to request the status of a transaction in the event that the authorization request times out. This API call will allow for the graceful recovery and handling of the transaction. If the transaction has been approved and the merchant has already notified the cardholder of the failure, it is best practice to initiate for the refund of the transaction. The merchant/acquirer will need to process the refund.

10.2 Acquirer-to-PaySecure (during cardholder authentication process)

- a. The acquirer has given control to PaySecure via the modal iFrame.
- b. Acquirer-side script must exist to wait for some period of time and close the iFrame if the timeout is reached. This timeout must be enough to account for cardholder authentication at the issuer and revert from PaySecure.

10.3 Cardholder-to-Issuer (during cardholder authentication)

- a. PaySecure passes iFrame control to Issuer (IAS) for cardholder authentication.
- b. Issuer-side script must exist to wait for some period of time and return control via Issuer 400 – JavaScript trigger to PaySecure in the event that the cardholder authentication exceeds that time.
- c. PaySecure will treat this timeout as a negative-authentication and respond to the merchant with a JavaScript error code indicating that the authentication was not completed successfully.

10.4 PaySecure-to-Issuer (during cardholder authentication)

- a. Cardholder has authenticated with the issuer and control returned back to PaySecure. PaySecure requests “authentication result” from the Issuer. (Auth_Result API call)
- b. PaySecure will wait for 10 seconds for a response from the Issuer.
- c. PaySecure will treat this timeout as a negative-authentication and respond to the Acquirer PG with a JavaScript error code indicating that the authentication was not successful.

10.5 Cardholder-to-PaySecure (PIN capture)

- a. PaySecure has presents scrambling PINPad for PIN capture as part of authentication.
- d. PaySecure script will exist to close the scrambling PINPad page and respond to the Acquirer PG in the event that a timeout period is reached or if there is a period of no activity by the cardholder.
- e. PaySecure will treat this timeout as a negative-PIN capture and respond back to the Acquirer PG with an error code indicating that authentication was not successful.

10.6 Transaction Initialization to IFrame Initialize

- a. PaySecure allows a maximum of 15 minutes time between the initiate API request/response and the IFrame being rendered by the Acquirer PG. If an IFrame render is attempted by Acquirer PG after the 15 minutes, transaction will not be allowed to process further.
- b. PaySecure will treat this timeout as a negative scenario and will decline the transaction.

10.7 PIN Capture to Authorization Request

- a. PaySecure retains the transaction details in session for a short period of time. If an authorization request has not received within 15 minutes of transaction initiation, session is cleared from memory for the respective transaction.
- b. An authorization attempt after the session expired, is declined with error code "96" and error message "system error" response.

10.8 Time Out Details

Time Out Table				
Acquirer Payment Gateway to PaySecure				
Sr	Command	Entity	Max Response Time	Remarks
1	CheckBIN	Acquirer PG to PaySecure	10 seconds from PaySecure	Decline the transaction after 10 second timeout.
2	CheckBIN2	Acquirer PG to PaySecure	10 seconds from PaySecure	Decline the transaction after 10 second timeout.
3	Initiate	Acquirer PG to PaySecure	20 seconds from PaySecure	Decline the transaction after 20 second timeout.
4	Initiate	Acquirer PG to PaySecure	20 seconds from PaySecure	Decline the transaction after 20 second timeout.
5	Authorize	Acquirer PG to PaySecure	35 seconds from PaySecure	Acquirer can timeout after 35 seconds.
6	Transaction Status	Acquirer PG to PaySecure	10 seconds from PaySecure	Decline the transaction after 10 second timeout.
7	Acquirer Iframe idle time out	Acquirer PG to Issuer/PaySecure	420 seconds	Acquirer PG can timeout after 420 seconds, if acquirer doesn't receive the Javascript function call from PaySecure regarding completion of the authentication/PIN capture and

				mark the transaction as declined.
8	Acquirer Re-direction idle time out	Acquirer PG to Issuer/ PaySecure	360 seconds	Acquirer PG can timeout after 300 seconds, if acquirer doesn't receive the redirection back to its page with proper response and mark the transaction as declined.
9	Authentication Page time out	Issuer OTP page	300 seconds	Acquirer will receive ACCU400 from Issuer/PaySecure and mark the transaction as declined.

11. System Security Features

The following security measures are being considered for the OTP based authentication system:

11.1 Merchant/Aggregator Authentication Methods

The interaction between the Merchant/Aggregator and Acquirer is only possible if the Merchant authenticates correctly with the acquirer.

Additionally merchant is also authenticated by the PaySecure system. PaySecure system uses Merchant ID and Merchant Password for authentication. These values are assigned to the merchants by the acquiring bank. These values are configured at system of all three entities i.e. PaySecure, Acquirer & Merchant.

11.2 Acquirer Authentication Methods

The interaction between the Acquirer and PaySecure system is only possible if the acquirer authenticates correctly with the PaySecure system. Following checks are performed for every HTTPS request:

1. Token
2. CallerID
3. Version
4. UserID
5. Password
6. PartnerID
7. Source IP address

11.3 Cardholder/ Shopper Browser Authentication Methods

NPCI validates the cardholder browser using Global Unique ID, GUID and Unique Public key that was sent to the Acquirer PG in the response of Initiate/Initiate2 API call. This ensures that OTP page URL is being passed to genuine cardholder browser.

11.4 Separation of Card details/PAN and OTP authentication mechanism

In this model, authentication details on IAS web page and card details are collected via separate channels. Customer OTP is never revealed to NPCI, Acquirer or Merchant.

This is done for two reasons:

- The Merchant is never given access to the Cardholder's registered mobile number or the dynamic OTP being sent to customer. The Merchant has no need to access this data and if the merchant has been compromised by an unrelated attack, only card details are compromised.
- NPCI is never given access to actual OTP being sent to customer as there is no need for NPCI to access this information. Even after successful authentication, only an authentication status response from IAS is received which is stored against the transaction, neither mobile number nor the OTP.

11.5 SSL Connection

Separate SSL connection between Merchant and Acquirer; Acquirer and NPCI; and between Cardholder browser and NPCI. This means the MITM attack which may have compromised the Cardholder's browser to Merchant connection doesn't automatically imply that the attacker can ascertain all data between the Cardholder's Browser and NPCI.

With respect to SSL, NPCI enforces strong encryption standards. All SSL sessions terminate on hardware load-balancers in the NPCI environment.

12. Integration Requirements

Integrating the PaySecure process into the acquirer system involves the following points:

12.1 Software Requirements

SOAP 1.1 or higher Web services client

12.2 Connectivity to NPCI's PaySecure product

12.2.1 Web Service Security

There are several security parameters that are used to ensure that only valid acquirer/merchants are allowed accessing the PaySecure web service. These parameters are listed below and will be provided to the acquirer at the time of on-boarding process:

1. Token = "a8b7fe50-ffc6-48db-b42e-cd49b5cc05d0"
2. CallerID = "AcquirerOne"
3. Version = "1.1.0.0"
4. UserID = "Acquirer123"

5. Password = "P@sswerd!"
6. PartnerID = "Merchant1"

Below security parameters are used to ensure that only valid acquirers/merchants are allowed accessing the PaySecure system.

1. Token:

- a) Unique token created by PaySecure system for each acquirer.
- b) Assigned to the acquirer during the certification process.
- c) PaySecure will receive this token in every API call.
- d) PaySecure will validate the token for every API call.

2. Caller ID (Acquirer ID):

- a) Unique Acquirer ID created by NPCI for each acquirer.
- b) Assigned to the acquirer during the certification process.
- c) PaySecure will receive this caller id in every API call.
- d) PaySecure will validate the caller id for every API call.

3. Version:

- a) PaySecure version for which acquirer is certified.
- b) Assigned to the acquirer during the certification process.
- c) PaySecure will receive this version in every API call.
- d) PaySecure will validate the version for every API call.

4. User ID:

- a) Unique User ID created by PaySecure for each acquirer.
- b) Assigned to the acquirer during the certification process.
- c) PaySecure will receive this user id in every API call.
- d) PaySecure will validate the user id for every API call.

5. Password:

- a) The password that corresponds to the user id to validate the authenticity.
- b) Assigned to the acquirer during the certification process.
- c) PaySecure will receive this password in every API call.
- d) PaySecure will validate the password for every API call.

6. Partner ID:

- a) Unique Partner ID created by PaySecure for each merchant/aggregator to uniquely identify the merchant.
- b) Assigned to the merchant via acquirer in response to the merchant on boarding form.
- c) PaySecure will receive this partner id in every API call.
- d) PaySecure will validate the partner id for every API call.

7. Merchant Password:

- a) Merchant Password is created and assigned by the acquiring bank to the merchant/aggregator during merchant enrolment on the acquirer payment gateway.
- b) PaySecure will receive this merchant password in every API call.
- c) PaySecure will validate the merchant password for every API call.

8. Merchant ID:

- a) Merchant ID is created and assigned by the acquiring bank to the merchant/aggregator during merchant enrolment on the acquirer payment gateway.
- b) PaySecure will receive this merchant id in every API call.
- c) PaySecure will validate the merchant id for every API call.

9. IP Address Checking:

- a) PaySecure will accept web service calls only from a pre-configured valid IP address of Payment Gateway.
- b) Acquirer bank will supply IP address to NPCI during the certification process.
- c) Both PR & DR IP's to be shared.

10. URLs

- a) Acquirer is required to connect to PaySecure via Internet
- b) NPCI will publish the URL
- c) Certification URL for API calls is
<https://cert.mwsrec.npci.org.in/MWS/MerchantWebService.asmx>

"Production URL will be shared at the time of Production movement"

13. Implementation Considerations

- ▶ Development Phase
- ▶ Pre certification Testing Phase with NPCI
- ▶ Certification Phase with NPCI – Test cases will be provided by NPCI. Acquiring bank to execute the test cases & share the results with NPCI along with screenshots for review.
- ▶ Production Setup

14. Web-Services Requirements

Web Services provide a simple interface for serving requests to PaySecure. To summarize, the order in which the services are called should be in the following order:

1. Checkbin
2. Checkbin2
3. Initiate
4. Initiate2
5. Authorize
6. TransactionStatus

**** Please note that the Iframe/browser re-direction will take place between calls 3/4 and 5.**

This web service exposes a method to execute commands, and is designed to receive and return XML formatted data. The web service method is represented below:

Object	Description
WebService	Web Service reference that interfaces the Channel commands.
Method	Description
CallPaySecure(command, xml input)	This method executes the supplied command (i.e. checkbin, initiate, authorize, transactionstatus) with its XML defined input parameters. The xml input must be html encoded.

Example: The XML tags for the input and output of these methods is the same as the input and output parameter names of the commands in the next section. The root node XML tag is <PaySecure>. All input and output parameters are child node of the <PaySecure> node. And the XML message header part should contain "Content-Length=xxxx" which means xxxx bytes are being posted, should not use "**Transfer-Encoding = chunked**" method.

15. Web Service API Calls

Acquirer Bank Payment Gateway will invoke the web service API calls hosted by PaySecure. The API calls i.e. CheckBin, CheckBin2, Initiate, Initiate2 & Authorize should be triggered by Payment Gateway in same sequence for every transaction. If any API call fails, the next API call should not be triggered for same transaction.

15.1 CheckBIN

The CheckBIN call determines the eligibility for eCommerce transaction enablement of the card entered by the cardholder (PaySecure will evaluate the BIN and return the eligibility of the BIN)

Acquirer Bank Payment Gateway sends to PaySecure

CheckBin	
Request**	<pre><soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:mer="https://PaySecure/merchant.soap.header/" xmlns:mer1="https://PaySecure/merchant.soap/"> <soapenv:Header> <RequestorCredentials xmlns="https://PaySecure/merchant.soap.header/"> <Token>AF165FBD-4E85-4c66-BEB1- C54DC16CD48B</Token> <Version>1.0.0.0</Version> <CallerID>720200</CallerID> <UserCredentials> <UserID>test@acculynk.com</UserID></pre>

	<pre> <Password>Password!1</Password> </UserCredentials> </RequestorCredentials> </soapenv:Header> <soapenv:Body> <CallPaySecure xmlns="https://PaySecure/merchant.soap/"> <strCommand>checkbin</strCommand> <strXML>&lt;PaySecure&gt;&lt;partner_id&gt; ACCUTEST&lt;/partner_id&gt;&lt;merchant_password&gt;F7@ 2zM3a&lt;/merchant_password&gt;&lt;card_bin&gt;60738470 0&lt;/card_bin&gt;&lt;/PaySecure&gt;</strXML> </CallPaySecure> </soapenv:Body> </soapenv:Envelope> </pre>
Response##	<pre> <soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema"> <soap:Body> <CallPaySecureResponse>&lt;?xml version="1.0" encoding="utf-16"?&gt; &lt;PaySecure xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema"&gt; &lt;status&gt;success&lt;/status&gt; &lt;errorcode&gt;0&lt;/errorcode&gt; &lt;qualified_internetpin&gt;TRUE&lt;/qualified_internetpin&gt; &lt;/PaySecure&gt;</CallPaySecureResponse> </soap:Body> </soap:Envelope> </pre>
<p>**Please pay attention to highlighted bold text in the request. The value in the “strXML” tag is must be html encoded. (i.e.: The “<” symbol is represented by “&lt;”)</p> <p>##The response is made up of various elements but to determine the success of CheckBIN API call, three elements are considered.</p> <p>First element to consider is <status>, this tag is used to let the requester know if BIN is eligible for RuPay ecommerce transactions and PaySecure will return a success or failure basis the status of eligibility.</p> <p>The second element to consider is the <errorcode> element. This element will return a ‘0’ to indicate that requested BIN is eligible for RuPay eCommerce transactions.</p> <p>Only in case of successful BIN eligibility, PaySecure will respond with status as “success” and error code as ‘0’.</p> <p>The third element to consider is the <qualified_internetpin>, this tag is used to let the</p>	

requester know if BIN is eligible for RuPay ecommerce transactions and PaySecure will return **TRUE** or **FALSE** basis the eligibility.

15.2 CheckBIN2

The CheckBIN2 call determines the eligibility of the card entered by the cardholder (PaySecure will evaluate the BIN and return the eligibility of the BIN)

****Note:** Once Acquirer bank is certified for re-direction flow, **CheckBIN API** call will be absolute for respective Acquirer and **CheckBIN2 API** will only be invoked for all transactions.

Acquirer Bank Payment Gateway sends to PaySecure

CheckBin	
Request**	<pre> <soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:mer="https://PaySecure/merchant.soap.header/" xmlns:mer1="https://PaySecure/merchant.soap/"> <soapenv:Header> <RequestorCredentials xmlns="https://PaySecure/merchant.soap.header/"> <Token>AF165FBD-4E85-4c66-BEB1- C54DC16CD48B</Token> <Version>1.0.0.0</Version> <CallerID>720200</CallerID> <UserCredentials> <UserID>test@acculynk.com</UserID> <Password>Password!1</Password> </UserCredentials> </RequestorCredentials> </soapenv:Header> <soapenv:Body> <CallPaySecure xmlns="https://PaySecure/merchant.soap/"> <strCommand>checkbin2</strCommand> <strXML<paysecure>&lt;partner_id>itm2ist1</part ner_id>&lt;merchant_password>Npci@202</merchan t_password>&lt;card_bin>652851000</card_bin>& lt;/paysecure></strXML> </CallPaySecure> </soapenv:Body> </soapenv:Envelope> </pre>
Response##	<pre> <soap:Envelope </pre>

	<pre> xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema"> <soap:Body> <CallPaySecureResponse>&lt;?xml version="1.0" encoding="utf-16"?&gt; &lt;PaySecure xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema"&gt; &lt;status&gt;success&lt;/status&gt; &lt;errorcode&gt;0&lt;/errorcode&gt; &lt;errmsg&/&gt; &lt;qualified_internetpin&gt;TRUE&lt;/qualified_internetpin&gt; &lt; Implements_Redirect&gt;TRUE&lt;/ Implements_Redirect&gt; &lt;/PaySecure&gt;</CallPaySecureResponse> </soap:Body> </soap:Envelope> </pre>
<p>**Please pay attention to highlighted bold text in the request. The value in the “strXML” tag is must be html encoded. (i.e.: The “<” symbol is represented by “&lt;”)</p> <p>##The response is made up of various elements but to determine the success of CheckBIN API call, three elements are considered.</p> <p>First element to consider is <status>, this tag is used to let the requester know if BIN is eligible for RuPay ecommerce transactions and PaySecure will return a success or failure basis the status of eligibility.</p> <p>The second element to consider is the <errorcode> element. This element will return a ‘0’ to indicate that requested BIN is eligible for RuPay eCommerce transactions.</p> <p>Only in case of successful BIN eligibility, PaySecure will respond with status as “success” and error code as ‘0’.</p> <p>The third element to consider is the <qualified_internetpin>, this tag is used to let the requester know if BIN is eligible for RuPay ecommerce transactions and PaySecure will return TRUE or FALSE basis the eligibility.</p>	

15.3 Initiate

The Initiate call is used to first establish the transaction between the acquirer and PaySecure (command allows a Merchant / Acquirer to begin the transaction).

Acquirer Bank Payment Gateway sends to PaySecure

Initiate Call	
Request**	<pre> <soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:mer="https://PaySecure/merchant.soap.header/" xmlns:mer1="https://PaySecure/merchant.soap/"> <soapenv:Header> </pre>

	<pre> <RequestorCredentials xmlns="https://PaySecure/merchant.soap.header/"> <Token>AF165FBD-4E85-4c66-BEB1-C54DC16CD48B</Token> <Version>1.0.0.0</Version> <CallerID>720203</CallerID> <UserCredentials> <UserID>test@acculynk.com</UserID> <Password>Password!1</Password> </UserCredentials> </RequestorCredentials> </soapenv:Header> <soapenv:Body> <CallPaySecure xmlns="https://PaySecure/merchant.soap/"> <strCommand>initiate</strCommand> <strXML>&lt;PaySecure&gt;&lt;partner_id&gt;ACCUTEST&lt;/partne r_id&gt;&lt;merchant_password&gt;F7@2zM3a&lt;/merchant_passw ord&gt;&lt;card_no&gt;1111222233334444&lt;/card_no&gt;&lt;car d_exp_date&gt;122014&lt;/card_exp_date&gt;&lt;language_code&gt; en&lt;/language_code&gt;&lt;auth_amount&gt;9031&lt;/auth_amou nt&gt;&lt;currency_code&gt;356&lt;/currency_code&gt;&lt;cvd2&gt; 0123&lt;/cvd2&gt;&lt;transaction_type_indicator&gt;SMS&lt;/transa ction_type_indicator&gt;&lt;tid&gt;20692448&lt;/tid&gt;&lt;stan&gt; ;000001&lt;/stan&gt;&lt;tran_time&gt;112952&lt;/tran_time&gt;&lt; tran_date&gt;0924&lt;/tran_date&gt;&lt;mcc&gt;6012&lt;/mcc&gt; &lt;acquirer_institution_country_code&gt;356&lt;/acquirer_instituti on_country_code&gt;&lt;retrieval_ref_number&gt;226511000001&lt; ;/retrieval_ref_number&gt;&lt;card_acceptor_id&gt;4234442342241 23&lt;/card_acceptor_id&gt;&lt;terminal_owner_name&gt;Acculynk. com&lt;/terminal_owner_name&gt;&lt;terminal_city&gt;Atlanta&lt;/ terminal_city&gt;&lt;terminal_state_code&gt;MH&lt;/terminal_state_ code&gt;&lt;terminal_country_code&gt;IN&lt;/terminal_country_cod e&gt;&lt;merchant_postal_code&gt;876678678&lt;/merchant_postal _code&gt;&lt;merchant_telephone&gt;456745674567&lt;/merchant _telephone&gt;&lt;order_id&gt;Order1289348&lt;/order_id&gt;&lt;c ustom1&gt;cust1sample&lt;/custom1&gt;&lt;custom2&gt;cust2sam ple&lt;/custom2&gt;&lt;custom3&gt;cust3sample&lt;/custom3&gt; &lt;custom4&gt;cust4sample&lt;/custom4&gt;&lt;custom5&gt;cust5 sample&lt;/custom5&gt;&lt;/PaySecure&gt;</strXML> </CallPaySecure> </soapenv:Body> </soapenv:Envelope> </pre>
Response##	<pre> <?xml version="1.0" encoding="utf-8"?><soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema"><soap:Body><CallP aySecureResponse xmlns="https://PaySecure/merchant.soap/"><CallPaySecureResult>&lt; ?xml version="1.0" encoding="utf-16"?&gt; &lt;PaySecure xmlns:xsi="http://www.w3.org/2001/XMLSchema- instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema"&gt; </pre>

****Please pay attention to highlight the bold text in the request. The value in the “strXML” tag is must be html encoded. (i.e.: The “<” symbol is represented by “<”)**

##The response is made up of various elements but to determine the success of Initiate API call, two elements are considered.

First element to consider is <status>, this tag is used to let the requester know if transaction was successfully initiated and card details has been verified with IAS and PaySecure will return a success or failure basis the status of card details verification.

The second element to consider is the <errorcode> element. This element will return a ‘0’ to indicate that transaction was successfully initiated and card details has been verified with IAS.

Only in case of successful transaction initiation and card details verification, PaySecure will respond with status as “success” and error code as ‘0’.

The Initiate2 call is used to first establish the transaction between the acquirer and PaySecure (command allows a Merchant / Acquirer to begin the transaction).
Acquirer Bank Payment Gateway sends to PaySecure

Initiate Call	
Request**	<pre><soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:mer="https://PaySecure/merchant.soap.header/" xmlns:mer1="https://PaySecure/merchant.soap/"> <soapenv:Header> <RequestorCredentials xmlns="https://PaySecure/merchant.soap.header/"> <Token>AF165FBD-4E85-4c66-BEB1-C54DC16CD48B</Token> <Version>1.0.0.0</Version> <CallerID>720203</CallerID> <UserCredentials> <UserID>test@acculynk.com</UserID></pre>

	<pre> <Password>Password!1</Password> </UserCredentials> </RequestorCredentials> </soapenv:Header> <soapenv:Body> <CallPaySecure xmlns="https://PaySecure/merchant.soap/"> <strCommand>initiate2</strCommand> <strXML>&lt;PaySecure&gt; &lt;partner_id&gt;itm2ist1&lt;/partner_id&gt; &lt;card_no&gt;6528510000000040&lt;/card_no&gt; &lt;language_code&gt;en&lt;/language_code&gt; &lt;auth_amount&gt;6001&lt;/auth_amount&gt; &lt;card_exp_date&gt;122018&lt;/card_exp_date&gt; &lt;transaction_type_indicator&gt;DMS&lt;/transaction_type_indicato r&gt; &lt;currency_code&gt;356&lt;/currency_code&gt; &lt;stan&gt;478785&lt;/stan&gt; &lt;tran_time&gt;182904&lt;/tran_time&gt; &lt;tran_date&gt;0102&lt;/tran_date&gt; &lt;acquirer_institution_country_code&gt;356&lt;/acquirer_institutio n_country_code&gt; &lt;retrieval_ref_number&gt;800218478785&lt;/retrieval_ref_numbe r&gt; &lt;card_acceptor_id&gt;CG0000000000002&lt;/card_acceptor_id&gt; &lt;terminal_owner_name&gt;Acculynk.com&lt;/terminal_owner_nam e&gt; &lt;terminal_city&gt;Atlanta&lt;/terminal_city&gt; &lt;terminal_state_code&gt;MH&lt;/terminal_state_code&gt; &lt;terminal_country_code&gt;IN&lt;/terminal_country_code&gt; &lt;merchant_postal_code&gt;876678678&lt;/merchant_postal_code& gt; &lt;merchant_telephone&gt;45674567456752125626&lt;/merchant_t elephone&gt; &lt;merchant_password&gt;Npci@202&lt;/merchant_password&gt; &lt;order_id&gt;order2JE8DCB&lt;/order_id&gt; &lt;custom1&gt;cust 1 sample&lt;/custom1&gt; &lt;custom2&gt;cust 2 sample&lt;/custom2&gt; &lt;custom3&gt;cust 3 sample&lt;/custom3&gt; &lt;custom4&gt;cust 4 sample&lt;/custom4&gt; &lt;custom5&gt;cust 5 sample&lt;/custom5&gt; &lt;cvd2&gt;0387&lt;/cvd2&gt; &lt;tid&gt;20692448&lt;/tid&gt; &lt;mcc&gt;6012&lt;/mcc&gt; </pre>
--	--

```
<!--formsg/><br>  
<!--tran_id<br>1000000000000000000000000025236<br>/tran_id<br><!--RedirectURL<br>https://testnpci/redirect_otp/Home/IssuerReg?<br>AccuCardholderId=89172389132&AccuGuid=6298312e-c54c-1954-ba32-a9211038d150&AccuHkey=5629y50g-e743-0022-5i2b-9aw8de632896<br>/RedirectURL<br><br>AuthenticationNotRequired<br>FALSE<br>/AuthenticationNotRequire<br>d<br>
```


15.5 Authorize

The authorize call is used to authorize a transaction for which authentications was successfully completed and verified by Issuing bank.

This call is sent by Acquirer to PaySecure to complete the transaction

Authorize Call	
Request**	<pre> <soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:mer="https://PaySecure/merchant.soap.header/" xmlns:mer1="https://PaySecure/merchant.soap/"> <soapenv:Header> <RequestorCredentials xmlns="https://PaySecure/merchant.soap.header/"> <Token>AF165FBD-4E85-4c66-BEB1-C54DC16CD48B</Token> <Version>1.0.0.0</Version> <CallerID>720203</CallerID> <UserCredentials> <UserID>test@acculynk.com</UserID> <Password>Password!1</Password> </UserCredentials> </RequestorCredentials> </soapenv:Header> <soapenv:Body> <CallPaySecure xmlns="https://PaySecure/merchant.soap/"> <strCommand>authorize</strCommand> <strXML>&lt;PaySecure&gt;&lt;partner_id&gt;ACCUTEST&lt;/partne r_id&gt;&lt;merchant_password&gt;F7@2zM3a&lt;/merchant_passw ord&gt;&lt;tran_id&gt;100000000000000000000000025253&lt;/tr an_id&gt;&lt;auth_amount&gt;2014&lt;/auth_amount&gt;&lt;current cy_code&gt;356&lt;/currency_code&gt;&lt;/PaySecure&gt;</strXML> </CallPaySecure> </soapenv:Body> </soapenv:Envelope> </pre>
Response##	<pre> <?xml version="1.0" encoding="utf-8"?><soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema"><soap:Body><CallP aySecureResponse xmlns="https://PaySecure/merchant.soap/"><CallPaySecureResult>&lt; ?xml version="1.0" encoding="utf-16"?&gt; &lt;PaySecure xmlns:xsi="http://www.w3.org/2001/XMLSchema- instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema"&gt; &lt;status&gt;success&lt;/status&gt; &lt;errorcode&gt;00&lt;/errorcode&gt; &lt;errmsg /&gt; &lt;apprcode&gt;183217&lt;/apprcode&gt; &lt;/PaySecure&gt;</CallPaySecureResult></CallPaySecureResponse>< /soap:Body></soap:Envelope> </pre>
<p>**Please pay attention to the bold text in the request. The value in the "strXML" tag must be html encoded. (i.e.: The "<" symbol is represented by "&lt;")</p>	

Only in case of successful transaction authorization by Issuing bank, PaySecure will respond with status as "success" and error code as '00'.

The TransactionStatus call is used to request the state of a transaction from PaySecure. Acquire sends to PaySecure to know the transaction status. It is not necessary to trigger this call in every transaction. It is recommended to use this call in case Payment Gateway doesn't receive response from PaySecure within specified time-frame.

```

xmlns:mer1="https://PaySecure/merchant.soap.header/"
<soapenv:Header>
  <RequestorCredentials
    xmlns="https://PaySecure/merchant.soap.header/"
    <Token>AF165FBD-4E85-4c66-BEB1-
C54DC16CD48B</Token>
    <Version>1.0.0.0</Version>
  </RequestorCredentials>
</soapenv:Header>

```

	<pre> xmlns:xsd="http://www.w3.org/2001/XMLSchema"> <soap:Body> <CallPaySecureResponse xmlns="https://PaySecure/merchant.soap/"> <CallPaySecureResult> &lt;?xml version="1.0" encoding="utf-16"?&gt; &lt;PaySecure xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema- instance"&gt;&lt;status&gt;success&lt;/status&gt;&lt;errorCode& gt;00&lt;/errorCode&gt;&lt;errmsg&gt;&lt;history&gt;&lt;tran saction&gt;&lt;status&gt;AZ&lt;/status&gt;&lt;tran_id&gt;100000 000000000000000000000000000000000011817&lt;/tran_id&gt;&lt;apprcode&gt;S MS020&lt;/apprcode&gt;&lt;recurring&gt;FALSE&lt;/recurring& &lt;datetime&gt;12/31/2012 19:09:51&lt;/datetime&gt; &lt;amount&gt;8001&lt;/amount&gt; &lt;/transaction&gt; &lt;/history&gt; &lt;/PaySecure&gt; </CallPaySecureResult> </CallPaySecureResponse> </soap:Body> </soap:Envelope> </pre>
	<p>**Please pay attention to the bold text in the request. The value in the “strXML” tag must be html encoded. (i.e.: The “<” symbol is represented by “&lt;”)</p> <p><i>##The response is made up of various elements but to determine the success of Authorize API call, two elements are considered.</i></p> <p><i>First element to consider is <status>, this tag is used to let the requester know if transaction status API call was successfully processed by PaySecure. PaySecure will return a success or failure basis the status of request processing.</i></p> <p><i>The second element to consider is the <errorCode> element. This element will return a ‘00’ to indicate that transaction was successfully successfully processed by PaySecure.</i></p> <p><i>Only in case of successful processing of transaction status API call, PaySecure will respond with status as “success” and error code as ‘00’.</i></p>

16. Best Practices

This section provides recommendations for the Acquirer Payment Gateway & merchant to be followed as security measures.

16.1 Transaction Review and Filtering

1. Amount Filter – Uses lower and upper transaction amount thresholds to restrict high-risk transactions.
2. Velocity Filter – Limits the total number of transactions received per hour/day, preventing high-volume attacks common with fraudulent transactions.
3. Shipping Billing Mismatches – Identifies high-risk transactions with different shipping and billing addresses, potentially indicating fraudulent transactions.

4. Transaction IP Velocity – Isolates suspicious activity from a single source by identifying excessive transactions received from the same IP address.
5. Suspicious Transaction Filter – Review highly suspicious transactions using criteria identified as being indicative of fraud or known to be fraud. One example is the lists available that contain known shipping addresses of fraudsters.
6. IP Address Blocking – Blocks transactions from IP addresses known to have been used in fraudulent activity.
7. Require Complete Order Information - a full address and phone number.

16.2 Network and Infra Security

1. Install a firewall - A firewall is a hardware or software solution that monitors the activity of external connections (primarily the internet) to an internal network of servers. Firewalls help to eliminate unauthorized or unwanted external activity and safeguard your network and connections from outside threats.
2. Store all sensitive or confidential information separate from web servers.
3. Use Anti-Virus Software and Update It regularly.
4. Regularly Download and Install Security Updates.
5. Avoid Sending or Requesting Confidential Information via Insecure Methods.

16.3 Adhere to the PCI Data Security Standard Requirements

The Payment Card Industry (PCI) Data Security Standard is an industry-wide program designed to significantly increase security for storing, transmitting, and processing cardholder data. To maximize security, it is recommended that the Merchant thoroughly review and adhere to PCI requirements.

16.4 Suspicious Transactions Monitoring:

The following suspicious transactions need to be thoroughly monitored:

1. New customer: Since fraudsters generally look for newer merchants for fear of identification.
2. Unusual size of purchase: Because of the need to maximize value of transaction due to limited life-span of stolen card
3. Suspicious quantities of the product purchased: If more than normal quantities of an item are purchased, it might indicate a counterfeit or lost/stolen card
4. Items with maximum resale value form the majority of the purchases: Fraudsters are looking with high value items which also have a very high resale value
5. Inclination towards speedy shipping/readiness to pay unusual shipping charges: Fraudsters want quick delivery to avoid getting caught or their address traced.
6. International shipping: This is generally a high risk transaction

7. Patterns in account numbers used for transactions (e.g.: transactions using consecutive/similar account numbers): These indicate usage of software-generated account numbers
8. Transactions through multiple cards getting shipped at the same address, or multiple transactions on one card over a very short period of time, or multiple card transactions from the same billing address but to different shipping addresses or transactions using multiple cards, but from the same IP address. The fraudster might be making a high amount of small purchases through different counterfeit cards/addresses to evade detection

16.5 NPCI recommendations

1. The Acquirer may engage the services of a web-crawling provider to monitor the activity of the website and ensure that their merchants do not indulge in illegal, brand damaging or fraudulent activity.
2. The web-crawler may also be used to confirm the validity of the merchant category code (MCC) by monitoring the goods/services transacted by the site.
3. Some websites like ones that capture cardholder data and pass on the same to other website breaching the privacy of the customers, fraudulent websites that solicit donations, and fraudulent websites that pose as government site or financial institution website may be identified by the web-crawling service provider.
4. The web-crawler can keep a check on websites that indulge in sale of counterfeit or trademark infringing goods/services

16.6 Card Not Present (CNP) transactions acquisition

Since a physical presence of the card is not required for eCommerce transaction, a CNP transaction is exposed to a wider array of risks as compared to the merchant POS transaction. Even though card security features and systems are in place for CNP transactions, frauds might still occur. Since a physical location is not required for an E-Commerce site, it is easier for frauds to occur online or over the phone. Hence it is essential for the acquirer to take additional measures to safeguard it from frauds.

An acquirer must verify additional application information for card not present transaction processing merchants. This includes business plans, advertisement details and other marketing materials. Additionally, the acquirer must check whether the applicant is an existing merchant that wants to add a website, or a new merchant. The form should collect additional information like the URL, IP address, details of the customer service, terms & conditions of the sale, security guidelines and risk management guidelines which the e-commerce merchant should provide to acquirer.

Some guidelines to minimize the risk of CNP merchant acquisition are mentioned below:

- ▶ A robust CNP fraud prevention program is indispensable for mitigating CNP risk. It is imperative to collect data and monitor both the cardholder and type of transaction
- ▶ The information can be cross-referenced with internal databases and external sources to establish indicators that denote a low- or high-risk exposure for each transaction.
- ▶ Use simulations/test checks to analyse prospective E-Commerce/IVR transaction merchants by actually placing test orders.
- ▶ Thoroughly analyse any third-party service providers that cater to the merchant's shipment/delivery/transaction processes.
- ▶ Manage exposure to risk by choosing merchants based on a strict set of parameters. Check for the following:
 1. Previous merchant processing relationship with a foreign/offshore acquirer
 2. Usage of multiple acquirers
 3. Business less than one year old
 4. New merchant without proof to show acceptable charge-back/fraud records
 5. Accounts that have quickly exceeded approved processing volumes
 6. Arrangements with third-party processors or ISOs with suspicious activity
- ▶ For E-Commerce merchants, the acquirer must conduct a website inspection including:
 1. Content
 2. Products
 3. Privacy policy
 4. Refund policy
 5. Cancellation policy
 6. Terms & conditions
 7. Website data security
 8. Data storage
 9. SSL payment options over the internet
 10. Links to other sites
- ▶ Ensure that the merchant does not process any illegal transaction (e.g.: pornography, rape, terrorism, gambling in goods or services, illegal goods or services, or counterfeit goods and services)
- ▶ Develop a database to gauge the risk exposure for customer and third party processor.
- ▶ Use the significant cardholder verification resources available to keep a check on potential fraudulent purchases
- ▶ Collaborate with other members in the payment value chain for the same sector to share databases before acquiring merchants to help mitigate fraudulent transactions.

Note: For additional information regarding merchant application; members may refer the Merchant Management and TPP Compliance for Acquirers.

16.7 Merchant Training

It is essential that the merchant be educated in the risks involved in card transactions. This responsibility of training the merchants lies with the acquirer. The entire merchant organization should have a thorough understanding of the fraud risks involved in the transaction and well versed in the risk management strategy.

Merchant training is not a one-time exercise, but a continuous process. Regular follow up regarding training requirements is essential. It is also important that the entire merchant portfolio is covered for training.

Merchant training can be conducted based on generic or specified events. Generic merchant trainings can be conducted just after merchant is acquired, and then at regular intervals to keep the merchant updated. Specific merchant trainings can be conducted for a certain class/category of merchants regarding fraudulent activities relevant to their category or due to trends in fraudulent transactions specific to a particular merchant type.

e.g. Jewellery merchants are susceptible to counterfeit card frauds during high-business days like Akshay Tritiya or Dhan Teras etc.

At the same time, any new features must be notified to all merchants within the acquirer's portfolio who may get affected by it. Also care must be taken to ensure the merchant is ready to use those new features effectively.

An acquirer's reputation in the market is highly dependent on the performance of its merchant portfolio. Hence it must be ensured that fraudulent transactions due to unprepared or uninformed staff are kept to a minimum.

It must be ensured that the merchant is well-versed with best practices and procedures for all of the above aspects.

1. Ensure that the merchant is aware of the risks of card payments
2. The merchant should be able to fine-tune its policies, operational practices and fraud prevention tools and controls according to newer fraud systems, if implemented
3. The acquirer should provide instructions to avoid charge-backs related to authorizations and retrieval requests.
4. The merchant staff must be educated on terminal maintenance
5. The acquirer should ensure that the merchant is educated about the various reasons for charge-backs, including:
 - ▶ Cardholder disputes
 - ▶ Fraudulent transactions
 - ▶ Rules pertaining to expired authorization for unshipped goods
 - ▶ Transaction authorization requirements
 - ▶ Time limits for retrieval requests
6. Train the merchants in correct security procedures related to authorization and fraud prevention
7. The acquirer must ensure that the employees of the merchant have been sufficiently trained to prevent fraud and handle disputes.
8. The merchant must be trained on various fraud types (e.g. skimming) and how to effectively recognize and counter them.
9. After the training process is over, the acquirer may conduct seeding or shopping-simulations to identify any gaps in the training and further find ways to fill them.

16.8 Documents to be submitted

16.8.1 Documents shared by NPCI with member bank

- a) RuPay Global Clearing and Settlement Technical Message Specification
- b) RuPay Online Switching Interface Specification
- c) RuPay PaySecure Acquirer Integration Guide
- d) WSDL and HTML & “JAVA Script and HTML changes for Banks”
- e) Certification Scripts
- f) Acquirer On-Boarding Form – Duly filled and signed by Bank Officials along with Bank Seal.



Annexures A

Frequently Asked Questions

1. How are Key Logging attacks being addressed by this solution?
 - A. Since a dynamic OTP is being generated and sent to customer, key logging attacks are redundant.
2. How are Phishing attacks being addressed by this solution?
 - A. Phishing is perhaps one of the biggest challenges for ecommerce. It happens every day worldwide. Typically a Cardholder will receive an email that they believe to be genuine, requesting the Cardholder to click on a URL and provide data.
NPCI has developed an Anti-Phishing approach for iFrame flow to aid users in confirming the legitimacy of their interaction with NPCI. Cardholder is displayed with BIN based bank theme on the PIN pad for them to verify the authenticity of the PIN pad. This looks similar to the theme on their debit/credit card. This process provides assurance to the Cardholder that they are interacting with NPCI's PaySecure solution and not a Phishing website.
3. How is Man in the Middle (MITM) attack being addressed by this solution?
 - A. Scrambling PIN pad is a web service hosted at NPCI; MITM attacker cannot modify the interaction between the scrambling PIN Pad and NPCI. Attacker requires to see the Actual PIN Pad shown to the Cardholder. However, upon an "on-click" event, the [X, Y] coordinates are encrypted with public key – which the MITM cannot decrypt.
If they are hiding behind an anonymous IP address, then the PIN Pad will never be displayed due to Web Application firewall and GEO Checking.
Post submission of authentication details on bank's IAS web page, secure hash code is shared by Issuer with Acquirer for authentication status. If any value is changed in between then hash code will not match at Acquirer and transaction will be declined by Acquirer.
For successful authentication response at Acquirer end, NPCI securely verifies the status of authentication with IAS system via different channel before proceeding further. Thus, it is ensured that authentication has taken place for the transaction and validated by Issuer.
4. What does RuPay propose to use as the first factor for authenticating online transactions?
 - A. Card Verification data
5. How will Acquirer verify the merchant?
 - A. The interaction between the merchant and acquirer is only possible if the merchant authenticates correctly with the acquirer.
6. How is cardholder's browser authenticated?

- A. Once cardholder and NPCI is connected over a secure channel, Cardholder is first authenticated by matching the details Global Unique ID, Unique transaction ID etc.
7. What will be the messaging format between acquirer and PaySecure system?
- A. SOAP Web Services will be used.
8. Is there any hardware change at the acquirer end?
- A. No
9. What details are captured by the merchant?
- A. Purchase and Card (*Merchant should be PCI DSS compliant to capture the card details*) related details are captured.
10. Who will display the PIN pad?
- A. NPCI will display the scrambling PINPad to the cardholder.
11. Who will capture the OTP?
- A. Issuing Bank will capture the OTP directly from the cardholder. No OTP data will flow through the merchant or acquirer or NPCI.
12. Who will form the ISO block?
- A. NPCI will form the ISO block.
13. How does Acquirer communicate with NPCI e-commerce module?
- A. Acquirer will communicate with NPCI using SOAP based web services call.
14. What about the security in terms of communication and data?
- A. NPCI URL's are globally hosted. However, server to server communication is carried on 128-bit SSL connectivity which ensures data security.
- B. Furthermore, NPCI adheres to PCIDSS standards which ensure data handling in way specified by PCIDSS.
- C. NPCI is not exposed to OTP (Tech team to confirm if we are storing card details??)
15. What kind of communication is between Acquirer & PaySecure?
- A. Server to server call secured via SSL 128 bit.
16. What are the activities to be carried-out after Bank is ready with development?
- After development following major steps are carried out:
- A. IP whitelisting at firewall of bank and NPCI.
- B. Thereafter integration is verified.
- C. Comfort Testing
- D. Review of Comfort Testing
- E. UAT

F. Review of UAT

G. Production movement

The above mentioned process takes 4 week to complete.

17. What will be the session length for re-direction flow at PaySecure end?

A. PaySecure will be maintaining 15 minute session for each unique transaction.



Annexures B

MESSAGE FORMAT

1. CheckBIN

CheckBin() – API call will evaluate the BIN and return the eligibility of the BIN for PaySecure					
Acquirer Payment Gateway sends to PaySecure					
Input Parameters	Required	Type (Length)	Format	Possible Values	Contents
SOAPHeader – version	M	ANS(7-32)	Variable	1.0.0.0	Static - Assigned by PaySecure The version of the code PG is currently certified too.
SOAPHeader – callerid	M	N(1-11)	Variable	720200	Static - Assigned by NPCI. A unique identifier of the Acquirer. Assigned at time of implementation. Will match the Acquirer Institution Code as assigned by NPCI.
SOAPHeader – token	M	ANS(36)	Fixed	AF165FBD-4E85-4c66-BEB1-C54DC16CD48B	Static - Assigned by NPCI. The password that corresponds to the CallerID to validate authenticity. Assigned at time of implementation
SOAPHeader – userid	M	ANS(1-256)	Variable	test@acculynk.com	Static - Assigned by PaySecure A unique identifier of the Acquirer originating the message. Assigned at

					time of implementation.
SOAPHeader – password	M	ANS(8-36)	Variable	Password!1	Static - Assigned by PaySecure The password that corresponds to the userID to validate authenticity. Assigned at time of implementation
partner_id	M	AN(1-20)	Variable	ACCUTEST	Static - Assigned by PaySecure A unique ID used to identify the merchant of the transaction being requested.
merchant_password	M	ANS(8)	Fixed	F7@2zM3a	Static - Configured and setup at time of enrollment into PaySecure As assigned by Acquirer bank
card_bin	M	N(9)	Fixed	607384XXX	First 9 digits of a valid card The Bank Identification Number, BIN, of the card the consumer entered at time of checkout on the merchant site.
Outputs					
status	M	AN(7)	Fixed	SUCCESS / FAILURE	The status of the validation done by PaySecure for the request, SUCCESS or FAILURE
Errorcode	M	N(1-4)	Variable	0 /00– Success Other – Failure Refer WEB-SERVICE ERROR	This is a numeric code used to provide success or failure response

				CODES Annexures C	
qualified_internetpin	M	AN(4 or 5)	Variable	True / False	Identifies the BIN eligibility for RuPay eCommerce transactions. True represents BIN is eligible.
Errormsg	C	ANS(0-1024)	Variable	Description for the Error Code.	Refer WEB-SERVICE ERROR CODES Annexures C

M – Mandatory; C- Conditional; O – Optional

If **qualified_internetpin** is false merchant/Acquirer should not initiate the initiate API call.

2. CheckBIN2

CheckBin2() – API call will evaluate the BIN and return the eligibility of the BIN for PaySecure					
Acquirer Payment Gateway sends to PaySecure					
Input Parameters	Required	Type (Length)	Format	Possible Values	Contents
SOAPHeader – version	M	ANS (7 - 32)	Variable	Static assigned by PaySecure	The version of the code PG is currently certified too.
SOAPHeader – callerid	M	N(1 to 11)	Variable	Static assigned by NPCI	A unique identifier of the Acquirer. Assigned at time of implementation. Will match the Acquirer Institution Code as assigned by NPCI. Acquirer ID assigned by NPCI
SOAPHeader – token	M	ANS(36)	Fixed	Static Assigned by PaySecure	The password that corresponds to the CallerID to validate authenticity. Assigned at time of implementation
SOAPHeader – userid		AN(1-	Variable	Static	A unique identifier of

	M	256)		assigned by PaySecure	the Acquirer originating the message. Assigned at time of implementation.
SOAPHeader – password	M	AN(8-36)	Variable	Static assigned by PaySecure	The password that corresponds to the userID to validate authenticity. Assigned at time of implementation
partner_id	M	AN(1-20)	Variable	Static assigned by PaySecure	A unique ID used to identify the merchant of the transaction being requested.
merchant_password	M	ANS (8)	Fixed	Configured and setup at time of enrollment into PaySecure	As assigned by Acquirer bank
card_bin	M	N(9)	Fixed	First 9 digits of a valid card	The Bank Identification Number, BIN, of the card the consumer entered at time of checkout on the merchant site.
Outputs					
status	M	AN(7)	Fixed	SUCCESS / FAILURE	The status of the validation done by PaySecure for the request, SUCCESS or FAILURE
errorcode	M	N(1-4)	Variable	See Error Codes / Messages for details	This is a numeric code used to provide success or failure response
qualified_internetpin	M	AN(4 or 5)	Variable	True / False	Identifies the BIN eligibility for RuPay eCommerce transactions. True represents BIN is eligible.
Implements_Redirect	M	AN(4 or 5)	Fixed	True / False	Identifies of the Authentication for this BIN implements

					redirect or iframe method for authentication.
errormsg	C	AN(0-1024)	Variable	Description for the error code.	Refer WEB-SERVICE ERROR CODES Annexures C

M – Mandatory; C- Conditional; O – Optional

3. Initiate

Initiate() – API call allows a PG to initiate the transaction process.					
Acquirer Payment Gateway sends to PaySecure					
Input Parameters	Required	Type (Length)	Format	Possible Values	Contents
SOAPHeader – version	M	ANS(7-32)	Variable	1.0.0.0	Static - Assigned by PaySecure. The version of the code PG is currently certified to use.
SOAPHeader – callerid	M	N(1-11)	Variable	720200	Static - Assigned by NPCI. A unique identifier of the Acquirer. Assigned at time of implementation. Will match the Acquirer Institution Code as assigned by NPCI. Acquirer ID assigned by NPCI
SOAPHeader – token	M	ANS(36)	Fixed	AF165FB D-4E85- 4c66- BEB1- C54DC16 CD48B	Static - Assigned by PaySecure. The password that corresponds to the CallerID to validate authenticity. Assigned at time of implementation
SOAPHeader –	M	ANS(1-	Variable	test@acc ulynk.co	Static - Assigned by PaySecure. A unique

userid		256)		m	identifier of the Acquirer or system originating the message. Assigned at time of implementation.
SOAPHeader – password	M	ANS(8-36)	Variable	Password !1	Static - Assigned by PaySecure. The password that corresponds to the userID to validate authenticity. Assigned at time of implementation
partner_id	M	AN(1-20)	Variable	ACCUTES T	Static - Assigned by PaySecure. A unique ID used to identify the merchant of the transaction being requested.
merchant_password	M	ANS(8)	Fixed	F7@2zM3 a	Static - Configured and setup at time of enrollment into PaySecure As assigned by Acquirer bank
card_no	M	N(13-19)	Variable	6073849 8000049 61	Full length card number as embossed on card The full length card number, PAN, as entered on the Merchant website by the consumer.
card_exp_date	M	N(6)	Fixed	MMYYYY	The month and year the card expires as embossed on the card
language_code	M	AN(2-5)	Variable	en	Language code used

					to drive the language on the PaySecure pages. EN - English
auth_amount	M	N(1-12)	Variable	Numeric transaction amount in base units (no decimal)	The transaction authorization amount in base units (i.e., a decimal payment RS. 110.25 would be 11025 in base units)
currency_code	M	N(3)	Fixed	356	ISO standard 4217, currency code in which the auth_amount is expressed.
cvd2	M	N(3-4)	Variable	123	Card Verification Data
transaction_type_indicator	M	A(3)	Fixed	SMS	Identify whether it is a single message or dual message(i.e. "SMS" or "DMS" only)
Tid	M	ANS(8)	Fixed	1111111 1	Card acceptor terminal ID. Dynamic – Merchant wise
Stan	M	N(6)	Fixed	386133	System Trace Audit Number. Dynamic – Varies
tran_time	M	N(6)	Fixed	162020	HHMMSS Local time at which the transaction began at the card acceptor location.
tran_date	M	N(4)	Fixed	1215	MMDD Local date at which the transaction began at the card acceptor location

Mcc	M	N(4)	Fixed	5541	Merchant category code (i.e. "4132") - Dynamic Value Refer to ISO 18245 for list of MCC's
acquirer_institution_country_code	M	N(3)	Fixed	356	Country code of acquirer (i.e. "356") Refer to ISO 3166 for country code list.
retrieval_ref_number	M	AN(12)	Fixed	2348163 86133	Dynamic- YDDDDHHSSSSSS Y- Year (Last digit of current year) DDD- Julian Date HH- Hour SSSSSS- System Trace Audit Number. Should be unique for each transaction.
card_acceptor_id	M	ANS(15)	Fixed	4234442 3422412 3	Dynamic - Card acceptor operating the transaction. May be referred to as the MID or merchant id. Varies Merchant wise
terminal_owner_name	M	ANS(1-23)	Variable	Acculynk. com	Dynamic - Merchant name - Right padded with spaces. Refer version POS switching interface 1.4 Varies Merchant wise
terminal_city	M	A(1-13)	Variable	Mumbai	Dynamic - Merchant city - Right padded with spaces. Varies Merchant wise

terminal_state_code	M	A(2)	Fixed	MH	Dynamic - Merchant state code Refer version POS switching interface 1.4 Varies Merchant wise
terminal_country_code	M	A(2)	Fixed	IN	Dynamic - Merchant country code, ISO 3166 Varies Merchant wise
merchant_postal_code	M	ANS(9)	Variable	400064	Dynamic - Merchant postal code left padded with zero Varies Merchant wise
merchant_telephone	M	ANS(20)	Variable	6788947010	Dynamic - Merchant telephone (customer support) Right padded with spaces. Refer version POS switching interface 1.4 Varies Merchant wise
order_id	O	ANS(1-50)	Variable	order2JE8DCB	Order reference number. Over 50 in length will be truncated.
custom1 (Merchant URL, if there are multiple url for the same then it should be comma (,) separated)	O	AN(0-128)	Variable	Merchant Defined	Up to a 128-byte length of Alpha-Numeric data that will be stored by PaySecure. RFU
custom2	O	AN(0-128)	Variable	Merchant Defined	Up to a 128-byte length of Alpha-Numeric data that will be stored by PaySecure. RFU

custom3	O	AN(0-128)	Variable	Merchant Defined	Up to a 128-byte length of Alpha-Numeric data that will be stored by PaySecure. RFU
custom4	O	AN(0-128)	Variable	S	Value “S” needs to be populated only for Small Scale Merchants and for any other case this field will not be present in the request send by PG.
custom5	O	AN(0-128)	Variable	Merchant Defined	Up to a 128-byte length of Alpha-Numeric data that will be stored by PaySecure. RFU
Outputs					
tran_id	C	N(0 or 30)	Fixed		A unique ID assigned by PaySecure to the successful transaction and remains constant throughout the lifecycle of the transaction. “Transaction ID will not be generate for declined Transaction.”
Guid	C	ANS(0 or 36)	Fixed	GUID	A unique ID assigned to the successful transaction and used during the PIN capture processes. “Transaction ID will not be generate for declined Transaction.”

Modulus	C	AN(0 or 256)	Fixed	Alpha numeric value up to 256 characters in length	RSA encryption key value assigned to the successful. The value will be posted to the PIN Pad for use in securing the PIN data. "Transaction ID will not be generate for declined Transaction."
Exponent	C	N(0 or 6)	Fixed	NNNNNN	RSA encryption key value assigned to the successful transaction. The value will be posted to the PIN Pad for use in securing the PIN data. "Transaction ID will not be generate for declined Transaction."
status	M	AN(7)	Fixed	SUCCESS / FAILURE	A status indicator. Card details has been verified by Issuer and transaction has successfully been initiated by PaySecure, SUCCESS or FAILURE
errorcode	M	N(0-4)	Variable	Refer WEB-SERVICE ERROR CODES Annexure s C	This is a numeric code used to provide success or failure response
errormsg	C	ANS(0-1024)	Variable	Descripti on for the	Refer WEB-SERVICE

				error code.	ERROR CODES Annexures C
--	--	--	--	-------------	----------------------------

Note: Left or Right padding will be done as per RuPay switching Interface 1.4 document.

Note: PaySecure to do validation on combination of Caller id (Acquirer ID), Partner id, and Merchant Password.

4. Initiate2

		Initiate2() - API call allows a Merchant to begin the authentication process using the redirect method.			
		Acquirer Payment Gateway sends to PaySecure			
Input Parameters	Required	Type (Length)	Format	Possible Values	Contents
SOAPHeader – version	M	ANS (7 - 32)	Variable	Static assigned by PaySecure	The version of the code PG is currently certified to use.
SOAPHeader – callerid	M	N(1 to 11)	Variable	Static assigned by NPCI	A unique identifier of the Acquirer. Assigned at time of implementation. Will match the Acquirer Institution Code as assigned by NPCI. Acquirer ID assigned by NPCI
SOAPHeader – token	M	ANS(36)	Fixed	Static assigned by PaySecure	The password that corresponds to the CallerID to validate authenticity. Assigned at time

					of implementation
SOAPHeader – userid	M	AN(1-256)	Variable	Static assigned by PaySecure	A unique identifier of the Acquirer or system originating the message. Assigned at time of implementation.
SOAPHeader – password	M	AN(8-36)	Variable	Static assigned by PaySecure	The password that corresponds to the userID to validate authenticity. Assigned at time of implementation
partner_id	M	AN(1-20)	Variable	Static assigned by PaySecure	A unique ID used to identify the merchant of the transaction being requested.
merchant_password	M	ANS (8)	Fixed	Configured and setup at time of enrollment into PaySecure	As assigned by Acquirer bank
card_no	M	N(13-19)	Variable	Full length card number as embossed on card	The full length card number, PAN, as entered on the Merchant website by the consumer.
card_exp_date	M	N(6)	Fixed	MMYYYY	The month and year the card expires as embossed on the card
BrowserUserAgent	M	AN(512)	Variable		Acquirer will collect the UserAgent string of the cardholder

					browser and pass in the Initiate2 API call
IPAddress	M	AN(50)	Variable		IPv4 or IPv6 address of the cardholder
HTTPAccept	M	ANS (256)	Variable		Accept header is used by HTTP clients to tell the server what content types they will accept
language_code	M	AN(2-5)	Variable	EN only – potential for additional in future	Language code used to drive the language on the PaySecure pages. EN - English
auth_amount	M	N(1-12)	Variable	Numeric transaction amount in base units (no decimal)	The transaction authorization amount in base units (i.e., a decimal payment RS. 110.25 would be 11025 in base units)
currency_code	M	N(3)	Fixed	356 only – potential for additional in future	ISO standard 4217, currency code in which the auth_amount is expressed.
Cvd2	M	N(3-4)	Variable		Card Verification Data (i.e. “0310”)
transaction_type_indicator	M	A(3)	Fixed		Identify whether it is a single message or dual message(i.e. “SMS” or “DMS” only)
tid	M	ANS (8)	Fixed		Card acceptor terminal ID
stan	M	N(6)	Fixed		System Trace Audit Number
tran_time	M	N(6)	Fixed	HHmmss	Local time at which the transaction began at the card

					acceptor location.
Tran_date	M	N(4)	Fixed	MMDD	Local date at which the transaction began at the card acceptor location
mcc	M	N(4)	Fixed		Merchant category code (i.e. "4132") Refer to ISO 18245 for list of MCC's
acquirer_institution_country_code	M	N(3)	Fixed		Country code of acquirer (i.e. "356") Refer to ISO 3166 for country code list.
Retrieval_ref_number	M	AN(12)	Fixed		YDDDDHHSSSSSS Y-Year (Last digit of current year) DDD-Julian Date HH-Hour SSSSSS-System Trace Audit Number
card_acceptor_id	M	ANS(15)	Fixed		Card acceptor operating the transaction. May be referred to as the MID or merchant id.
Terminal_owner_name	M	ANS(1-23)	Variable		Merchant name - Right padded with spaces. Refer version POS switching interface 1.4
terminal_city	M	A(1-13)	Variable		Merchant city - Right padded with spaces.
terminal_state_code	M	A(2)	Fixed		Merchant state code Refer version POS switching interface 1.4

terminal_country_code	M	A(2)	Fixed		Merchant country code, ISO 3166
merchant_postal_code	M	ANS(9)	Variable		Merchant postal code left padded with zero
merchant_telephone	M	AN(20)	Variable		Merchant telephone (customer support) Right padded with spaces. Refer version POS switching interface 1.4
order_id	O	ANS(1-50)	Variable		Order reference number. Over 50 in length will be truncated.
custom1 (Merchant URL, if there are multiple url for the same then it should be comma (,) separated)	O	AN(0-128)	Variable	Merchant Defined	Up to a 128-byte length of Alpha-Numeric data that will be stored by PaySecure. RFU
custom2	O	AN(0-128)	Variable	Merchant Defined	Up to a 128-byte length of Alpha-Numeric data that will be stored by PaySecure. RFU
custom3	O	AN(0-128)	Variable	Merchant Defined	Up to a 128-byte length of Alpha-Numeric data that will be stored by PaySecure. RFU
custom4	O	AN(0-128)	Variable	S	Value "S" needs to be populated only for Small Scale Merchants and for any other case this field will not be present in the request send by PG.

custom5	O	AN(0-128)	Variable	Merchant Defined	Up to a 128-byte length of Alpha-Numeric data that will be stored by PaySecure. RFU
Outputs					
tran_id	M	N(30)	Fixed		A unique ID assigned by PaySecure to the transaction and remains constant throughout the lifecycle of the transaction.
RedirectURL	M	ANS(10-24)	Variable	URL	The URL the cardholder needs to be redirected to initiate authentication process and parameter names. As part of this field AccuGUID, AccucardholderID, AccuHkey will be passed along with Issuer URL to redirect Note : Values are html encoded.
AuthenticationNotRequired	M	AN(4 or 5)	Fixed	True / False	If True, the acquirer can call the Authorize API call directly without the need for authentication. This will be set to true for international card brands supported on RuPay rails.
status	M	AN(7)	Fixed	SUCCESS / FAILURE	A status indicator. Card details has

					been verified by Issuer and if transaction has successfully been initiated by PaySecure, SUCCESS or FAILURE
errorcode	M	N(1-4)	Variable	Refer WEB-SERVICE ERROR CODES Annexures C	This is a numeric code used to provide success or failure response
errormsg	M	AN(0-1024)	Variable	Description for Error Code.	Refer WEB-SERVICE ERROR CODES Annexures C

5. Redirection Request Parameter

Method: POST

Parameter Name & Type: Hidden

Parameter Name	Required	Length	Format	Description
AccuCardholderId	M	36 (ANS)	Variable	Extract value from Initiate2 API Response in RedirectURL tag.
AccuGuid	M	36 (ANS)	Variable	Extract value from Initiate2 API Response in RedirectURL tag.
AccuReturnURL	M	512 (ANS)	Variable	Acquirer has to provide URL to which authentication status will be sent.
session	M	1024 (ANS)	Variable	To maintain and retrieve the transaction at acquirer end, Session value will be echo back in response.
AccuRequestId	M	128(ANS)	Variable	To avoid tampering of request data in transit, the acquirer is required to provide a hash code for the issuer to validate. This hash code can be generated by

				hashing TransactionID, AccuCardholderId, AccuGuid & session parameter value.
--	--	--	--	--

6. Redirection Response Parameter

Method: POST

Parameter Name & Type: Hidden

Parameter Name	Required	Length	Format	Description
AccuResponseCode	M	10(AN)	Variable	Authentication status code will be sent by Issuer/PaySecure
session	M	1024 (ANS)	Variable	Same value should be send in response without any modification.
AccuGuid	M	36 (ANS)	Variable	Same value will be send in response without any modification and it helps to match request & response.
AccuRequestId	M	128(ANS)	Variable	To avoid tampering of response data in transit, the issuer is required to provide a hash code for the acquirer to validate. This hash code can be generated by hashing TransactionID, AccuGuid, session & AccuResponseCode parameter value.

7. Authorize

Authorize() - used to authorize the payment with the issuer					
Acquire sends to PaySecure to complete the transaction					
Input Parameters	Required	Type (Length)	Format	Possible Values	Contents
SOAPHeader - version	M	ANS(7-32)	Variable	Static assigned by PaySecure	The version of the code PG is currently certified to use.
SOAPHeader - callerid	M	N(1- 11)	Fixed	Static assigned by NPCI	A unique identifier of the Acquirer. Assigned at time of implementation. Will match the Acquirer Institution Code as

					assigned by NPCI. Acquirer ID assigned by NPCI
SOAPHeader – token	M	ANS(36)	Fixed	Static assigned by PaySecure	The password that corresponds to the CallerID to validate authenticity. Assigned at time of implementation
SOAPHeader – userid	M	ANS(1-256)	Variable	Static assigned by PaySecure	A unique identifier of the Acquirer or system originating the message. Assigned at time of implementation.
SOAPHeader – password	M	ANS(8-36)	Variable	Static assigned by PaySecure	The password that corresponds to the userID to validate authenticity. Assigned at time of implementation
partner_id	M	AN(1-20)	Variable	Static assigned by PaySecure	A unique ID used to identify the merchant of the transaction being requested.
merchant_password	M	ANS(8)	Fixed		As assigned by Acquirer bank
tran_id	M	N(30)	Fixed	Transaction ID	A unique ID assigned by PaySecure, as a result of the Initiate call, to the transaction and remains constant throughout the lifecycle of the transaction.
** auth_amount	M	N(1-12)	Variable	Numeric transaction amount in base units (no decimal)	The transaction authorization amount in base units (i.e., a decimal payment RS. 110.25 would be 11025 in base units)
currency_code	M	N(3)	Fixed	ISO defined	Standard ISO 4217,

				Currency Code – 356	currency code in which the auth_amount is expressed.
Outputs					
Apprcode	C	AN(6)	Fixed		Populated from DE38 as received from the Issuer
status	M	AN(7)	Fixed	SUCCESS / FAILURE	A status indicator. Transaction was successfully authorized by Issuer. <i>To be checked in combination with <errorcode> value.</i>
errorcode	M	N(0-4)	Variable	Refer WEB-SERVICE ERROR CODES Annexures C	This is a numeric code used to provide success or failure response. <i>To be checked in combination with <status> value.</i>
errormsg	C	ANS(0-1024)	Variable	Description of Error Code.	Refer WEB-SERVICE ERROR CODES Annexures C

**** Auth amount received in Authorize API call would be treated as final.**

8. Transaction Status

Input Parameters	Required	Type (Length)	Format	Contents
SOAPHeader – version	M	ANS(7-32)	Variable	Static assigned by PaySecure
SOAPHeader – callerid	M	N(1-11)	Variable	A unique identifier of the Acquirer. Assigned at time of implementation. Will match the Acquirer Institution Code as assigned by NPCI. Acquirer ID assigned by NPCI
SOAPHeader – token	M	ANS(36)	Fixed	Static assigned by PaySecure
SOAPHeader –	M	ANS(1-	Variable	Static assigned by PaySecure

userid		256)		
SOAPHeader – password	M	ANS(8-36)	Variable	Static assigned by PaySecure
merchant_password	M	ANS (8)	Fixed	As assigned by Acquirer bank
tran_id	M	N(30)	Fixed	The unique transaction ID determined by PaySecure.
partner_id	M	AN(1-20)	Variable	The partner ID (i.e. “MYSHOP”, etc.)
Outputs				
History	M		Node	Transaction history with each transaction enclosed in a ‘<transaction>’ node. Please see the next table for details.
status	M	AN(7)	Fixed	The status of the API call processing by PaySecure, SUCCESS or FAILURE
errorcode	M	N(0-4)	Variable	The numeric error code (Refer WEB-SERVICE ERROR CODES Annexures C)
errormsg	M	ANS(0-1024)	Variable	Refer WEB-SERVICE ERROR CODES Annexures C

The table below describes the nodes that will be enclosed in the ‘<history>’ node.

Xml Node	Type (Length)	Purpose
<transaction>		Returns transaction details for a single transaction.
Child nodes		Contents
<tran_id>	N(30)	Transaction ID
<status>	AN(1-2)	Transaction status code (i.e.: AZ): AQ PIN Acquired AZ Authorized DC Declined I Initiated PE Prior to EFT

<apprcode>	AN(0 or 6)	Approval Code of the transaction
<datetime>	ANS(20)	The timestamp of the transaction in GMT. (i.e.: 11/20/2008 11:23:42)
<amount>	N(1-12)	The transaction amount in base units (i.e., a decimal payment RS. 110.25 would be 11025 in base units)
Sample Output		
<pre> <status>success</status> <errorcode>00</errorcode> <errmsg /> <history> <transaction> <status>I</status> <tran_id>a06ff65ef7f01ce79eadae4ae7713645</tran_id> <apprcode /> <datetime>10/10/2012 13:42:24</datetime> <amount>0</amount> </transaction> </history> </acculynk> </pre>		

Annexures C

WEB-SERVICE ERROR CODES

1. Error Codes / Status Codes For CheckBIN/CheckBIN2

Error Code	Description	Reference XML Tag
0	BIN eligibility validation was successful.	Entire SOAP envelop along with values of parameters passed PaySecure will successfully complete all technical as well as business validations on XML structure and values of all the parameters as present within tag to return code '0'
01	Missing Parameter	Entire SOAP envelop along with values of parameters passed PaySecure will check all the parameters and if any of the required parameter is not present then return this error code.
02	Invalid Command	<strCommand> PaySecure checks if command in this tag is as per specs and if not will decline with this error code.
400	General Error	If PaySecure is not able to process transactions for codes as specified in this document then PaySecure can return this code.
401	Command is Null or Empty	<strCommand> IAS checks tag and if NULL or Empty command found, then decline with this error code
402	XML is Null or Empty	<strXML> , If XML body in this tag is Null or Empty then return this message
406	Not Authenticated	<RequestorCredentials> , <partner_id> , <merchant_password> IAS has to check under SOAP header and validate User ID and Password if not match then IAS to return this error code
407	Not Authorized	<RequestorCredentials> And Source IP PaySecure checks for credentials passed by PG and source IP and if source IP is not from white listed IP addresses at application level then PaySecure will return this error code.
408	XML Data Error	<errorcode> under <strXML> PaySecure will validate entire XML structure/format and if validation fails then PaySecure will return this error code
410	Invalid BIN	<errorcode> under <strXML> PaySecure will return this error code if BIN is not available or eligible for RuPay e-commerce transactions.

2. Error Codes / Status Codes For initiate/initiate2

Error Code	Description	Reference XML Tag
0	Transaction was successfully initiated and card details verified by issuer.	Entire SOAP envelop along with values of parameters passed PaySecure will successfully complete all technical as well as business validations on XML structure and values of all the parameters as present within tag to return code '0'
01	Missing Parameter	Entire SOAP envelop along with values of parameters passed PaySecure will check all the parameters and if any of the required parameter is not present then return this error code.
02	Invalid Command	<strCommand> PaySecure checks if command in this tag is as per specs and if not will decline with this error code.
13	Amount Error	<auth_amount> under <strXML> If amount value send by PG is not greater than 0 then PaySecure will decline with this error code.
96	System Error	In case if due to any technical error PaySecure is unable to process transaction then PaySecure to return this error code.
400	General Error	If PaySecure is not able to process transactions for codes as specified in this document then PaySecure can return this code.
401	Command is Null or Empty	<strCommand> IAS checks tag and if NULL or Empty command found, then decline with this error code
402	XML is Null or Empty	<strXML> , If XML body in this tag is Null or Empty then return this message
406	Not Authenticated	<RequestorCredentials>, <partner_id>, <merchant_password> IAS has to check under SOAP header and validate User ID and Password if not match then IAS to return this error code
407	Not Authorized	<RequestorCredentials> And Source IP PaySecure checks for credentials passed by PG and source IP and if source IP is not from white listed IP addresses at application level then PaySecure will return this error code.
408	XML Data Error	<errorcode> under <strXML> PaySecure will validate entire XML structure/format and if validation fails then PaySecure will return this error code
410	Invalid BIN	<errorcode> under <strXML> PaySecure will return this error code if BIN is not available or eligible for RuPay e-commerce transactions.

412	Issuer Authentication Failure	<strXML> PaySecure passes all card details to Issuer for verification and if verification is failed by Issuer, PaySecure to send this error code.
-----	-------------------------------	---

3. Error Codes / Status Codes For Authorize

Error Code	Description	Reference XML Tag
00	Transaction was successfully authorized by Issuing bank.	Entire SOAP envelop along with values of parameters passed PaySecure will successfully complete all technical as well as business validations on XML structure and values of all the parameters present within tag and passes the same to Issuing bank. On successful response from Issuinp bank, PaySecure to return code '00'
01	Missing Parameter	Entire SOAP envelop along with values of parameters passed PaySecure will check all the parameters and if any of the required parameter is not present then return this error code.
02	Invalid Command	<strCommand> PaySecure checks if command in this tag is as per specs and if not will decline with this error code.
13	Amount Error	<auth_amount> under <strXML> If amount value send by PG is not greater than 0 then PaySecure will decline with this error code.
41	DECLINED (lost card)	<strXML> If card is marked as Lost at Issuer end, Issuer to pass this error code to PaySecure and PaySecure sends this error code to PG.
42	DECLINED (no account)	<strXML> If no account is found for the card number provided at Issuer end, Issuer to pass this error code to PaySecure and PaySecure sends this error code to PG.
43	DECLINED (stolen)	<strXML> If card is marked as Stolen at Issuer end, Issuer to pass this error code to PaySecure and PaySecure sends this error code to PG.
51	NON SUFFICIENT FUNDS	<strXML> If sufficient funds are not available in cardholder account to process that transaction amount, Issuer to pass this error code to PaySecure and PaySecure sends this error code to PG.
54	EXPIRED CARD	<strXML> If card is expired, Issuer to pass this error code to PaySecure and PaySecure sends this error code to PG.
55	WRONG PIN	<strXML> If cardholder has submitted wrong PIN, Issuer to pass this error code to PaySecure and PaySecure sends this error code to PG.
57	DECLINED (cardholder not allowed)	<strXML> If transactions are not allowed for the provided card at Issuer end, Issuer to pass this error code to PaySecure and PaySecure sends this error code to PG.

58	DECLINED (terminal not allowed)	<strXML> If transactions are not allowed for the provided terminal details at Issuer end, Issuer to pass this error code to PaySecure and PaySecure sends this error code to PG.
59	DECLINED (fraud)	<strXML> If transaction is match any RISK criterias/rules at Issuer end, Issuer to pass this error code to PaySecure and PaySecure sends this error code to PG.
60	DECLINED (contact acquirer)	<strXML>
61	DECLINED (exceeds with)	<strXML> If daily withdrawal amount limit is exceeded for provided card at Issuer end, Issuer to pass this error code to PaySecure and PaySecure sends this error code to PG.
62	DECLINED (restricted card)	<strXML> If card is marked as Restricted for usage at Issuer end, Issuer to pass this error code to PaySecure and PaySecure sends this error code to PG.
65	DECLINED (exceeds frequency)	<strXML> If daily withdrawal frequency limit is exceeded for provided card at Issuer end, Issuer to pass this error code to PaySecure and PaySecure sends this error code to PG.
91	ERROR	If NPCI switch does not receives response from Issuer switch within specified time limit, transactions is declined by NPCI switch with this error code.
92	NO ROUTING AVAILABLE	If no routing is available for provided BIN at NPCI switch end, NPCI switch to pass this error code to PaySecure and PaySecure sends this error code to PG.
96	SYSTEM ERROR	In case if due to any technical error PaySecure is unable to process transaction then PaySecure to return this error code.
96	PREVIOUSLY AUTHORIZED	Duplicate Authorize API Call ** Transaction was authorized at PaySecure and received duplicate Authorize request for same transaction ID from PG. It is completely on PG's discretion to decline/approve the transaction based on response received for duplicate Authorize API call and member entity is always advised to trigger the Transaction Status API call in case of response not received for first authorize API call instead of duplicate Authorize API call.

** As per the RuPay PaySecure specification PG is expected to send only one Authorize API call per transaction and in case of duplicate authorize call send by PG, NPCI reserves the right to decline/response to the request as per current functionality.

96	PREVIOUSLY DECLINED	<p>Duplicate Authorize API Call **</p> <p>Transaction was authorized at PaySecure and received duplicate Authorize request for same transaction ID from PG.</p> <p>It is completely on PG's discretion to decline/approve the transaction based on response received for duplicate Authorize API call and member entity is always advised to trigger the Transaction Status API call in case of response not received for first authorize API call instead of duplicate Authorize API call.</p>
110	NO ACCT	If not account is found for provided card at Issuer end, Issuer to pass this error code to PaySecure and PaySecure to send this error code to PG.
120	ACCT CLOSED	If not account is closed for the card provided at Issuer end, Issuer to pass this error code to PaySecure and PaySecure to send this error code to PG.
399	SYSTEM UNAVAILABLE	In case if interfacing systems are unavailable at PaySecure end, for some reasons resulting in non-processing of transaction then PaySecure to return this error code
400	General Error	If PaySecure is not able to process transactions for codes as specified in this document then PaySecure can return this code.
401	Command is Null or Empty	<p><strCommand></p> <p>IAS checks tag and if NULL or Empty command found, then decline with this error code</p>
402	XML is Null or Empty	<p><strXML>, If XML body in this tag is Null or Empty then return this message</p>
406	Not Authenticated	<p><RequestorCredentials>, <partner_id>, <merchant_password></p> <p>IAS has to check under SOAP header and validate User ID and Password if not match then IAS to return this error code</p>
407	Not Authorized	<p><RequestorCredentials> And Source IP</p> <p>PaySecure checks for credentials passed by PG and source IP and if source IP is not from white listed IP addresses at application level then PaySecure will return this error code.</p>
408	XML Data Error	<p><errorcode> under <strXML></p> <p>PaySecure will validate entire XML structure/format and if validation fails then PaySecure will return this error code</p>

** As per the RuPay PaySecure specification PG is expected to send only one Authorize API call per transaction and in case of duplicate authorize call send by PG, NPCI reserves the right to decline/response to the request as per current functionality.

4. Error Codes / Status Codes For Transaction Status

Error Code	Description	Reference XML Tag
00	Transaction Status API call was processed successfully by PaySecure.	Entire SOAP envelop along with values of parameters passed PaySecure will successfully complete all technical as well as business validations on XML structure and values of all the parameters as present within tag to return code '0'
01	Missing Parameter	Entire SOAP envelop along with values of parameters passed PaySecure will check all the parameters and if any of the required parameter is not present then return this error code.
02	Invalid Command	<strCommand> PaySecure checks if command in this tag is as per specs and if not will decline with this error code.
96	SYSTEM ERROR	In case if due to any technical error PaySecure is unable to process transaction then PaySecure to return this error code.
400	General Error	If PaySecure is not able to process transactions for codes as specified in this document then PaySecure can return this code.
401	Command is Null or Empty	<strCommand> IAS checks tag and if NULL or Empty command found, then decline with this error code
402	XML is Null or Empty	<strXML> , If XML body in this tag is Null or Empty then return this message
406	Not Authenticated	<RequestorCredentials> , <partner_id> , <merchant_password> IAS has to check under SOAP header and validate User ID and Password if not match then IAS to return this error code
407	Not Authorized	<RequestorCredentials> And Source IP PaySecure checks for credentials passed by PG and source IP and if source IP is not from white listed IP addresses at application level then PaySecure will return this error code.
408	XML Data Error	<errorcode> under <strXML> PaySecure will validate entire XML structure/format and if validation fails then PaySecure will return this error code

Annexures D

Test cases: For sample test cases refer certification guide. (Certification Guide would be shared subsequently)

Annexures E

State Code value in NPCI switch

State	Code
Andhra Pradesh	AP
Arunachal Pradesh	AR
Assam	AS
Bihar	BR
Chattisgarh	CG
Goa	GA
Gujarat	GJ
Haryana	HR
Himachal Pradesh	HP
Jammu & Kashmir	JK
Jharkhand	JH
Karnataka	KA
Kerala	KL
Madhya Pradesh	MP
Maharashtra	MH
Manipur	MN
Meghalaya	ML
Mizoram	MZ
Nagaland	NL
Orissa	OR
Punjab	PB
Rajasthan	RJ
Sikkim	SK
Tamil Nadu	TN
Telangana	TS
Tripura	TR
Uttarakhand (Uttranchal)	UK
Uttar Pradesh	UP
West Bengal	WB
Andaman & Nicobar	AN
Chandigarh	CH
Dadra and NagarHaveli	DN
Daman & Diu	DD
Delhi	DL
Lakshadweep	LD
Puducherry	PY

Annexures F

Valid MCC (Merchant Category Code) details

MCCODE	MCDISC
742	Veterinary Services
763	Agricultural Cooperative
780	Landscaping Services
1520	General Contractors
1711	Heating, Plumbing, A/C
1731	Electrical Contractors
1740	Masonry, Stonework, and Plaster
1750	Carpentry Contractors
1761	Roofing/Siding, Sheet Metal
1771	Concrete Work Contractors
1799	Special Trade Contractors
2741	Miscellaneous Publishing and Printing
2791	Typesetting, Plate Making, and Related Services
2842	Specialty Cleaning
3000	AIR UNITED AIRLINES
3001	AMERICAN AIRLINES
3002	PAN AMERICAN
3004	TRANS WORLD AIRLINES
3005	BRITISH AIRWAYS
3006	JAPAN AIRLINES
3007	AIR FRANCE
3008	LUFTHANSA
3009	AIR CANADA
3010	KLM (ROYAL DUTCH AIRLINES)
3011	AEROFLOT
3012	AIR QANTAS
3013	AIR ALITALIA
3014	SAUDI ARABIAN AIRLINES
3015	SWISSAIR
3016	AIRLINE SAS
3017	SOUTH AFRICAN AIRWAYS
3018	AIR VARIG (BRAZIL)
3020	AIR-INDIA
3021	AIR ALGERIE
3022	PHILIPPINE AIRLINES
3023	AIR MEXICANA
3024	PAKISTAN INTERNATIONAL
3025	AIR NEW ZEALAND
3026	EMIRATES AIRLINES
3027	AIRLINE UTA/INTERAIR
3028	AIR MALTA
3029	AIR SABENA

3030	AEROLINEAS ARGENTINAS
3031	OLYMPIC AIRWAYS
3032	AIR EL AL
3033	ANSETT AIRLINES
3034	AUSTRALIAN AIRLINES
3035	AIRLINE TAP (PORTUGAL)
3036	AIR VASP (BRAZIL)
3037	EGYPTAIR
3038	KUWAIT AIRWAYS
3039	AVIANCA
3040	GULF AIR (BAHRAIN)
3041	BALKAN-BULGARIAN AIRLINES
3042	FINNAIR
3043	AER LINGUS
3044	AIR LANKA
3045	NIGERIA AIRWAYS
3046	CRUZEIRO DO SUL (BRAZIL)
3047	IRLINE THY (TURKEY)
3048	AIR MAROC
3049	AIR TUNIS AIR
3050	ICELANDAIR
3051	AUSTRIAN AIRLINES
3052	AIR LANCHILE
3053	AIR AVIACO (SPAIN)
3054	AIR LADECO (CHILE)
3055	AIRLINE LAB (BOLIVIA)
3056	QUEBECAIRE
3057	EAST/WEST AIRLINES (AUSTRALIA)
3058	AIR DELTA
3060	AIR NORTHWEST
3061	CONTINENTAL
3063	U.S. AIR
3064	ADRIA AIR
3065	AIRINTER
3066	SOUTHWESTAIR
3067	AIRLINES VANAGUARD
3071	AIR BRITISH COLUMBIA
3075	SINGAPORE AIRLINES
3076	AEROMEXICO
3077	THAI AIRWAYS
3078	CHINA AIRLINES
3081	NORDAIR
3082	KOREAN AIRLINES
3083	AIR AFRIQUE
3084	EVA AIRLINES
3085	ARILINES MIDWEST EXP

3086	CARNIVAL AIR
3087	METRO AIRLINES
3088	CROATIA AIRLINES
3089	TRANSAERO
3090	UNI AIRWAYS
3092	MIDWAY AIRLINES
3094	ZAMBIA AIRWAYS
3096	AIR ZIMBABWE
3097	SPANAIR
3098	ASIANA AIR
3099	AIR CATHAY PACIFIC
3100	MALAYSIAN AIRLINE SYSTEM
3102	IBERIA AIR
3103	AIR GARUDA (INDONESIA)
3106	BRAATHENSAIR BRAATHENS S.A.F.E. (NORWAY)
3110	AIR WINGS AIRWAYS
3111	BRMIDLANDAIR
3112	WINDWRDISAIR
3115	TOWERAIR
3117	AIR VIASA
3118	VALLEY AIRLINES
3125	AIRLINE TAN
3126	TALAIR
3127	AIR TACA INTERNATIONAL
3129	SURINAM AIRWAYS
3130	SUN WORLD INTERNATIONAL
3132	FRONTIER AIR
3133	SUNBELT AIRLINES
3135	SUDAN AIRWAYS
3136	QATAR AIRWAYS COMPANY
3137	SINGLETONAIR SINGLETON
3138	SIMMONS AIRLINES
3143	SCENIC AIRLINES
3144	VIRGIN ATLANTIC
3145	SAN JUAN AIRLINES
3146	LUXAIR
3151	AIR ZAIRE
3154	PRINCEVILLE
3159	PBA AIRLINE
3161	NIPPON AIRWAYS
3164	NORONTAIR
3165	NEW YORK HELICOPTER
3170	AIR MOUNT COOK
3171	CANADIAN AIRLINES
3172	NATIONAIR
3175	MIDDLE EAST AIR

3176	METROFLIGHT AIRLINES
3178	MESA AIR
3181	AIR MALEV
3182	AIRLINE LOT (POLAND)
3184	AIRLINE LIAT
3185	AIRLINE LAV (VENEZUELA)
3186	AIRLINE LAP (PARAGUAY)
3187	AIR LACSA (COSTA RICA)
3190	AIR JUGOSLAV AIR
3191	AIR ISLAND AIRLINES
3192	AIR IRAN AIR
3193	INDIAN AIRLINES
3196	HAWAIIAN AIR
3197	HAVASU AIRLINES
3200	GUYANA AIRWAYS
3203	GOLDEN PACIFIC AIR
3204	FREEDOM AIR
3206	CHINA EAST AIR
3212	AIR DOMINICANA
3215	DAN AIR SERVICES
3216	CUMBERLAND AIRLINES
3217	AIRLINE CSA
3218	CROWN AIR
3219	AIRLINE COPA
3220	AIR COMPANIA FAUCETT
3221	TRANSPORTES AEROS MILITARES
3222	COMMAND AIRWAYS
3223	COMAIR
3228	CAYMAN AIRWAYS
3229	AIR SAETA ùSOCIAEDAD
3231	SAHSA ùSERVICIO AEREO DE
3233	CAPITOL AIR
3234	AIRLINE BWIA
3235	BROCKWAY AIR
3238	BEMIDJI AIRLINES
3239	BAR HARBOR AIRLINES
3240	BAHAMASAIR
3241	AIR AVIATECA (GUATEMALA)
3242	AIR AVENSA
3243	AUSTRIAN AIR SERVICE
3251	ALOHA AIRLINES
3252	AIRLINE ALM
3253	AIR AMERICA WEST
3254	US AIR SHUTTLE
3256	ALASKA AIRLINES
3259	AMERICAN TRANS AIR

3261	AIR CHINA
3262	RENO AIR INC.
3263	SERVICIO CARABOBO ASC AIRLINES
3266	AIR SEYCHELLES
3267	AIR PANAMA
3280	AIR JAMAICA
3282	AIR DJIBOUTI
3284	AERO VIRGIN ISLANDS
3285	AEROPERU
3286	AEROLINEAS NICARAGUENSIS
3287	AERO COACH AVIATION
3292	CYPRUS AIRWAYS
3293	ECUATORIANA
3294	ETHIOPIAN AIRLINES
3295	KENYA AIRWAYS
3297	ROMANIAN AIR
3298	AIR MAURITIUS
3299	WIDEROEÆS FLYVESELSKAP
3351	AFFILIATED AUTO RENTAL
3352	AMERICAN INTL RENT-A-CAR
3353	BROOKS RENT-A-CAR
3354	ACTION AUTO RENTAL
3357	HERTZ RENT-A-CAR
3359	PAYLESS CAR RENTAL
3360	SNAPPY CAR RENTAL
3361	AIRWAYS RENT-A-CAR
3362	ALTRA AUTO RENTAL
3364	AGENCY RENT-A-CAR
3366	BUDGET RENT-A-CAR
3368	HOLIDAY RENT-A-CAR
3370	RENT-A-WRECK
3376	AJAX RENT-A-CAR
3380	TRIANGLE RENT-A-CAR
3381	EUROP CAR
3385	TROPICAL RENT-A-CAR
3386	SHOWCASE RENTAL CARS
3387	ALAMO RENT-A-CAR
3389	AVIS RENT-A-CAR
3390	DOLLAR RENT-A-CAR
3391	EUROPE BY CAR
3393	NATIONAL CAR RENTAL
3394	KEMWELL GROUP RENT-A-CAR
3395	THRIFTY RENT-A-CAR
3396	TILDEN RENT-A-CAR
3398	ECONO-CAR RENT-A-CAR
3400	AUTO HOST CAR RENTALS

3405	ENTERPRISE RENT-A-CAR
3409	GENERAL RENT-A-CAR
3412	A-1 RENT-A-CAR
3414	GODFREY NATL RENT-A-CAR
3420	ANSA INTL RENT-A-CAR
3421	ALLSTATE RENT-A-CAR
3423	AVCAR RENT-A-CAR
3425	AUTOMATE RENT-A-CAR
3427	AVON RENT-A-CAR
3428	CAREY RENT-A-CAR
3429	INSURANCE RENT-A-CAR
3430	MAJOR RENT-A-CAR
3431	REPLACEMENT RENT-A-CAR
3432	RESERVE RENT-A-CAR
3433	UGLY DUCKLING RENT-A-CAR
3434	USA RENT-A-CAR
3435	VALUE RENT-A-CAR
3436	AUTOHANSA RENT-A-CAR
3437	CITE RENT-A-CAR
3438	INTERENT RENT-A-CAR
3439	MILLEVILLE RENT-A-CAR
3441	ADVANTAGE RENT-A-CAR
3501	HOLIDAY INN EXPRESS
3502	BEST WESTERN HOTELS
3503	SHERATON HOTELS
3504	HILTON HOTELS
3505	FORTE HOTELS
3506	GOLDEN TULIP HOTELS
3507	FRIENDSHIP INNS
3508	QUALITY INNS
3509	MARRIOTT HOTELS
3510	DAYS INNS
3511	ARABELLA HOTELS
3512	INTER-CONTINENTAL HOTELS
3513	WESTIN HOTELS
3514	AMERISUITES
3515	RODEWAY INNS
3516	LA QUINTA MOTOR INNS
3517	AMERICANA HOTELS
3518	SOL HOTELS
3519	PULLMAN INTL HOTELS PULLMAN INTERNATIONALHOTELS
3520	MERIDIEN HOTELS
3522	TOKYO HOTEL
3523	PENINSULA HOTEL
3524	WELCOMGROUP HOTELS

3525	DUNFEY HOTELS
3526	PRINCE HOTELS
3527	DOWNTOWNER-PASSPORT HOTEL
3528	RED LION HOTELS
3529	CP HOTELS
3530	RENAISSANCE HOTELS
3533	HOTEL IBIS
3534	SOUTHERN PACIFIC HOTELS
3535	HILTON INTERNATIONALS
3536	AMFAC HOTELS
3537	ANA HOTEL
3538	CONCORDE HOTELS
3539	SUMMERFIELD SUITES HOTEL
3540	IBEROTEL HOTELS
3541	HOTEL OKURA
3542	ROYAL HOTELS
3543	FOUR SEASONS HOTELS
3544	CIGA HOTELS
3545	SHANGRI-LA INTL HOTELS SHANGRI-LA INTERNATIONAL
3546	SIERRA SUITES HOTELS
3548	HOTELS MELIA
3549	AUBERGE DES GOUVERNEURS
3550	REGAL 8 INNS
3551	MIRAGE HOTEL AND CASINO
3552	COAST HOTELS
3553	PARK INNS INTERNATIONAL
3555	TREASURE ISLAND HOTEL & CASINO
3558	JOLLY HOTELS
3561	GOLDEN NUGGET
3562	COMFORT INNS
3563	JOURNEY'S END MOTELS
3564	SAMÆS TOWN HOTEL & CASINO
3565	RELAX INNS
3568	LADBROKE HOTELS
3570	FORUM HOTELS
3572	MIYAKO HOTELS
3573	SANDMAN HOTELS
3574	VENTURE INNS
3575	VAGABOND HOTELS
3577	MANDARIN ORIENTAL HOTEL
3579	HOTEL MERCURE
3581	DELTA HOTEL
3582	CALIFORNIA HOTEL & CASINO
3583	SAS HOTELS
3584	PRINCESS HOTELS INTL PRINCESS HOTELS INTERNATIONAL

3585	HUNGAR HOTELS
3586	SOKOS HOTELS
3587	DORAL HOTELS
3588	HELMSLEY HOTELS
3590	FAIRMONT HOTELS
3591	SONESTA HOTELS
3592	OMNI HOTELS
3593	CUNARD HOTELS
3595	HOSPITALITY INNS
3598	REGENT INTL HOTELS REGENT INTERNATIONAL HOTELS
3599	PANNONIA HOTELS
3603	NOAHÆS HOTELS
3612	MOVENPICK HOTELS
3615	TRAVELODGE
3620	BINIONÆS HORSESHOE CLUB
3622	MERLIN HOTELS
3623	DORINT HOTELS
3624	LADY LUCK HOTEL AND CASINO
3625	HOTEL UNIVERSALE
3628	EXCALIBUR AND CASINO
3629	DAN HOTELS
3631	SLEEP INNS
3632	THE PHOENICIAN
3633	RANK HOTEL
3634	SWISSOTEL
3635	RESO HOTELS
3636	SAROVA HOTELS
3637	RAMADA INNS
3638	HOWARD JOHNSON
3639	MOUNT CHARLOTTE THISTLE
3640	HYATT HOTELS
3641	SOFITEL HOTELS
3642	NOVOTEL HOTELS
3643	STEIGENBERGER HOTELS
3644	ECONO LODGES
3645	QUEENS MOAT HOUSES
3646	SWALLOW HOTELS
3647	HUSA HOTELS
3648	DE VERAÆ HOTELS
3649	RADISSON HOTELS
3650	RED ROOF INNS
3651	IMPERIAL LONDON HOTEL
3652	EMBASSY HOTELS
3653	PENTA HOTELS
3654	LOEWS HOTELS
3655	SCANDIC HOTELS

3656	SARA HOTELS
3657	OBEROI HOTELS
3658	OTANI HOTELS
3659	TAJ HOTELS INTERNATIONAL
3660	KNIGHTS INNS
3661	METROPOLE HOTELS
3662	CIRCUS CIRCUS HOTEL & CASINO
3663	HOTELES EL PRESIDENTE
3664	FLAG INN
3665	HAMPTON INNS
3666	STAKIS HOTELS
3667	LUXOR HOTEL AND CASINO
3668	MARITIM HOTELS
3669	ELDORADO HOTEL AND CASINO
3670	ARCADE HOTELS
3671	ARCTIA HOTELS
3672	CAMPANILE HOTELS
3673	IBUSZ HOTELS
3674	RANTASIPI HOTELS
3675	INTERHOTEL CEDOK
3676	MONTE CARLO HOTEL & CASINO
3677	CLIMAT DE FRANCE HOTELS
3678	CUMULUS HOTELS
3679	SILVER LEGACY & CASINO
3680	HOTEIS OTHAN
3681	ADAMS MARK HOTELS
3682	SAHARA HOTEL AND CASINO
3683	BRADBURY SUITES
3684	BUDGET HOST INN
3685	BUDGETEL HOTELS
3686	SUISSE CHALET
3687	CLARION HOTELS
3688	COMPRI HOTELS
3689	CONSORT HOTELS
3690	COURTYARD BY MARRIOTT
3691	DILLON INNS
3692	DOUBLETREE HOTELS
3693	DRURY INNS
3694	ECONOMY INNS OF AMERICA
3695	EMBASSY SUITES
3696	EXEL INNS
3697	FAIRFIELD HOTELS
3698	HARLEY HOTELS
3699	MIDWAY MOTOR LODGE
3700	MOTEL 6
3701	LA MANSION DEL RIO HOTEL

3702	THE REGISTRY HOTEL
3703	RESIDENCE INNS
3704	ROYCE HOTELS
3705	SANDMAN INNS
3706	SHILO INNS
3707	SHONEY'S INNS
3708	VIRGIN RIVER HOTEL & CASINO
3709	SUPER 8 MOTELS
3710	THE RITZ CARLTON HOTELS
3711	FLAG INNS (AUSTRALIA)
3712	BUFFALO BILLÆS HOTEL & CASINO
3713	QUALITY PACIFIC HOTEL
3714	FOUR SEASONS HOTEL(AUSTRALIA)
3715	FAIRFIELD INN
3716	CARLTON HOTELS
3717	CITY LODGE HOTELS
3718	KAROS HOTELS
3719	PROTEA HOTELS
3720	SOUTHERN SUN HOTELS
3721	CONRAD
3722	WYNDHAM HOTELS
3723	RICA HOTELS
3724	INTER NOR HOTELS
3725	SEAPINES PLANTATION
3726	RIO SUITES
3727	BROADMOOR HOTEL
3728	BALLYÆS HOTEL AND CASINO
3729	JOHN ASCUAGAÆS NUGGET
3730	MGM GRAND HOTEL
3731	HARRAHÆS HOTELS AND CASINOS
3732	OPRYLAND HOTEL
3733	BOCA RATON RESORT
3734	HARVEY/BRISTOL HOTEL
3735	MASTERS ECONOMY INNS
3736	COLORADO BELLE/EDGEWATER
3737	RIVIERA HOTEL AND CASINO
3738	TROPICANA RESORT & CASINO
3739	WOODSIDE HOTELS & RESORT
3740	TOWNPLACE SUITES
3741	MILLENNIUM BROADWAY HOTEL
3742	CLUB MED
3743	BILTMORE HOTEL & SUITES
3744	CAREFREE RESORTS
3745	ST REGIS HOTEL
3746	THE ELIOT HOTEL
3747	CLUBCORP/CLUB RESORTS

3748	WELLESLEY INNS
3749	BEVERLY HILLS HOTELS
3750	CROWNE PLAZA HOTEL
3751	HOMEWOOD SUITES
3752	PEABODY HOTELS
3753	GREENBRIAR HOTELS
3754	AMELIA ISLAND
3755	THE HOMESTEAD
3756	SOUTH SEAS RESORT
3757	CANYON RANCH
3758	KAHALA MANDARIN ORIENTAL HOTEL
3759	THE ORCHID AT MAUNA LANI
3760	HALEKULANI HOTEL/WAIKIKI PARC
3761	PRIMADONNA HOTEL & CASINO
3762	WHISKEY PETEÆS HOTEL & CASINO
3763	CHATEAU ELAN WINERY & RESORT
3764	BEAU RIVAGE HOTEL & CASINO
3765	BELLARIO
3766	FREMONT HOTEL AND CASINO
3767	MAIN STREET HOTEL & CASINO
3768	SILVER STAR HOTEL & CASINO
3769	STRATOSPHERE HOTEL AND CASINO
3770	SPRINGHILL SUITES
3771	CEASARS HOTEL AND CASINO
3772	NEMACOLIN WOODLANDS
3773	VENETIAN RESORT HOTEL & CASINO
3774	NEW YORK NY HOTEL & CASINO
3775	OCEAN DUNES RESORT AND VILLAS
3776	NEVELE GRANDE RESORT & COUNTRY CLUB
3777	MANDALAY BAY RESORT
3780	DISNEYLAND HOTELS
4011	Railroads
4111	Commuter Transport, Ferries
4112	Passenger Railways
4119	Ambulance Services
4121	Taxicabs/Limousines
4131	Bus Lines
4214	Motor Freight Carriers and Trucking - Local and Lo
4215	Courier Services
4225	Public Warehousing and Storage û Farm Products, Re
4411	Cruise Lines
4457	Boat Rentals and Leases
4468	Marinas, Service and Supplies
4511	Airlines, Air Carriers
4582	Airports, Flying Fields
4722	Travel Agencies, Tour Operators

4723	TUI Travel - Germany
4784	Tolls/Bridge Fees
4789	Transportation Services
4812	Telecommunication Equipment and Telephone Sales
4813	Key-entry Tele Merchant providing single
4814	Telecommunication Services
4815	VISA PHONE
4816	Computer Network Services
4821	Telegraph Services
4829	Wires, Money Orders
4899	Cable, Satellite, and Other Pay Television and Rad
4900	Utilities
5013	Motor Vehicle Supplies and New Parts
5021	Office and Commercial Furniture
5039	Construction Materials
5044	Photographic, Photocopy, Microfilm Equipment, and
5045	Computers, Peripherals, and Software
5046	Commercial Equipment I
5047	Medical, Dental, Ophthalmic, and Hospital Equipmen
5051	Metal Service Centers
5065	Electrical Parts and Equipment
5072	Hardware, Equipment, and Supplies
5074	Plumbing, Heating Equipment, and Supplies
5085	Industrial Supplies (Not Elsewhere Classified)
5094	Precious Stones and Metals, Watches and Jewelry
5099	Durable Goods (Not Elsewhere Classified)
5111	Stationary, Office Supplies, Printing and Writing
5122	Drugs, Drug Proprietaries, and Druggist Sundries
5131	Piece Goods, Notions, and Other Dry Goods
5137	Uniforms, Commercial Clothing
5139	Commercial Footwear
5169	Chemicals and Allied Products (Not Elsewhere Class
5172	Petroleum and Petroleum Products
5192	Books, Periodicals, and Newspapers
5193	Florists Supplies, Nursery Stock, and Flowers
5198	Paints, Varnishes, and Supplies
5199	Nondurable Goods
5200	Home Supply Warehouse Stores
5211	Lumber, Building Materials Stores
5231	Glass, Paint, and Wallpaper Stores
5251	Hardware Stores
5261	Nurseries, Lawn and Garden Supply Stores
5271	Mobile Home Dealers
5300	Wholesale Clubs
5309	Duty Free Stores
5310	Discount Stores

5311	Department Stores
5331	Variety Stores
5399	Miscellaneous General Merchandise
5411	Grocery Stores, Supermarkets
5422	Freezer and Locker Meat Provisioners
5441	Candy, Nut, and Confectionery Stores
5451	Dairy Products Stores
5462	Bakeries
5499	Miscellaneous Food Stores û Convenience Stores and
5511	Car & Truck Dealers (New & Used) Sales, Service, R
5521	Car & Truck Dealers (Used Only) Sales, Service, Re
5531	Auto and Home Supply Stores
5532	Automotive Tire Stores
5533	Automotive Parts and Accessories Stores
5541	Service Stations
5542	Automated Fuel Dispensers
5551	Boat Dealers
5561	Camper, Recreational and Utility Trailer Dealers
5571	Motorcycle Shops and Dealers
5592	Motor Homes Dealers
5598	Snowmobile Dealers
5599	Miscellaneous Auto Dealers
5611	MenÆs and BoyÆs Clothing and Accessories Stores
5621	WomenÆs Ready-To-Wear Stores
5631	WomenÆs Accessory and Specialty Shops
5641	ChildrenÆs and InfantÆs Wear Stores
5651	Family Clothing Stores
5655	Sports and Riding Apparel Stores
5661	Shoe Stores
5681	Furriers and Fur Shops
5691	MenÆs, WomenÆs Clothing Stores
5697	Tailors, Alterations
5698	Wig and Toupee Stores
5699	Miscellaneous Apparel and Accessory Shops
5712	Furniture, Home Furnishings, and Equipment Stores,
5713	Floor Covering Stores
5714	Drapery, Window Covering, and Upholstery Stores
5718	Fireplace, Fireplace Screens, and Accessories Stor
5719	Miscellaneous Home Furnishing Specialty Stores
5722	Household Appliance Stores
5732	Electronics Stores
5733	Music Stores-Musical Instruments, Pianos, and Shee
5734	Computer Software Stores
5735	Record Stores
5811	Caterers
5812	Eating Places, Restaurants

5813	Drinking Places
5814	Fast Food Restaurants
5912	Drug Stores and Pharmacies
5921	Package StoresùBeer, Wine, and Liquor
5931	Used Merchandise and Secondhand Stores
5932	Antique Shops
5933	Pawn Shops
5935	Wrecking and Salvage Yards
5937	Antique Reproductions
5940	Bicycle Shops
5941	Sporting Goods Stores
5942	Book Stores
5943	Stationery Stores, Office, and School Supply Store
5944	Jewelry Stores, Watches, Clocks, and Silverware St
5945	Hobby, Toy, and Game Shops
5946	Camera and Photographic Supply Stores
5947	Gift, Card, Novelty, and Souvenir Shops
5948	Luggage and Leather Goods Stores
5949	Sewing, Needlework, Fabric, and Piece Goods Stores
5950	Glassware, Crystal Stores
5960	Direct Marketing - Insurance Services
5962	Direct Marketing - Travel
5963	Door-To-Door Sales
5964	Direct Marketing - Catalog Merchant
5965	Direct Marketing - Combination Catalog and Retail
5966	Direct Marketing - Outbound Tele
5967	Direct Marketing - Inbound Tele
5968	Direct Marketing - Subscription
5969	Direct Marketing - Other
5970	ArtistÆs Supply and Craft Shops
5971	Art Dealers and Galleries
5972	Stamp and Coin Stores
5973	Religious Goods Stores
5975	Hearing Aids Sales and Supplies
5976	Orthopedic Goods - Prosthetic Devices
5977	Cosmetic Stores
5978	Typewriter Stores
5983	Fuel Dealers
5992	Florists
5993	Cigar Stores and Stands
5994	News Dealers and Newsstands
5995	Pet Shops, Pet Food, and Supplies
5996	Swimming Pools Sales
5997	Electric Razor Stores
5998	Tent and Awning Shops
5999	Miscellaneous Specialty Retail

6010	Manual Cash Disburse
6011	Automated Cash Disburse
6012	Financial Institutions
6051	Non-Fl, Money Orders
6211	Security Brokers/Dealers
6300	Insurance Underwriting, Premiums
6399	Insurance - Default
6513	Real Estate Agents and Managers - Rentals
7011	Hotels, Motels, and Resorts
7012	Timeshares
7032	Sporting/Recreation Camps
7033	Trailer Parks, Campgrounds
7210	Laundry, Cleaning Services
7211	Laundries
7216	Dry Cleaners
7217	Carpet/Upholstery Cleaning
7221	Photographic Studios
7230	Barber and Beauty Shops
7251	Shoe Repair/Hat Cleaning
7261	Funeral Services, Crematories
7273	Dating/Escort Services
7276	Tax Preparation Services
7277	Counseling Services
7278	Buying/Shopping Services
7296	Clothing Rental
7297	Massage Parlors
7298	Health and Beauty Spas
7299	Miscellaneous General Services
7311	Advertising Services
7321	Credit Reporting Agencies
7332	BLU
7333	Commercial Photography, Art and Graphics
7338	Quick Copy, Repro, and Blueprint
7339	Secretarial Support Services
7342	Exterminating Services
7349	Cleaning and Maintenance
7361	Employment/Temp Agencies
7372	Computer Programming
7375	Information Retrieval Services
7379	Computer Repair
7392	Consulting, Public Relations
7393	Detective Agencies
7394	Equipment Rental
7395	Photo Developing
7399	Miscellaneous Business Services
7511	Truck Stop

7512	Car Rental Agencies
7513	Truck/Utility Trailer Rentals
7519	Recreational Vehicle Rentals
7523	Parking Lots, Garages
7531	Auto Body Repair Shops
7534	Tire Retreading and Repair
7535	Auto Paint Shops
7538	Auto Service Shops
7542	Car Washes
7549	Towing Services
7622	Electronics Repair Shops
7623	A/C, Refrigeration Repair
7629	Small Appliance Repair
7631	Watch/Jewelry Repair
7641	Furniture Repair, Refinishing
7692	Welding Repair
7699	Miscellaneous Repair Shops
7829	Picture/Video Production
7832	Motion Picture Theaters
7841	Video Tape Rental Stores
7911	Dance Hall, Studios, Schools
7922	Theatrical Ticket Agencies
7929	Bands, Orchestras
7932	Billiard/Pool Establishments
7933	Bowling Alleys
7941	Sports Clubs/Fields
7991	Tourist Attractions and Exhibits
7992	Golf Courses - Public
7993	Video Amusement Game Supplies
7994	Video Game Arcades
7995	Betting/Casino Gambling
7996	Amusement Parks/Carnivals
7997	Country Clubs
7998	Aquariums
7999	Miscellaneous Recreation Services
8011	Doctors
8021	Dentists, Orthodontists
8031	Osteopaths
8041	Chiropractors
8042	Optometrists, Ophthalmologists
8043	Opticians, Eyeglasses
8049	Chiropodists, Podiatrists
8050	Nursing/Personal Care
8062	Hospitals
8071	Medical and Dental Labs
8099	Medical Services

8111	Legal Services, Attorneys
8211	Elementary, Secondary Schools
8220	Colleges, Universities
8241	Correspondence Schools
8244	Business/Secretarial Schools
8249	Vocational/Trade Schools
8299	Educational Services
8351	Child Care Services
8398	Charitable and Social Service Organizations - Fund
8641	Civic, Social, Fraternal Associations
8651	Political Organizations
8661	Religious Organizations
8675	Automobile Associations
8699	Membership Organizations
8734	Testing Laboratories
8911	Architectural/Surveying Services
8931	Accounting/Bookkeeping Services
8999	Professional Services
9211	Court Costs, Including Alimony and Child Support -
9222	Fines - Government Administrative Entities
9223	Bail and Bond Payments
9311	Tax Payments - Government Agencies
9399	Government Services (Not Elsewhere Classified)
9402	Postal Services - Government Only
9405	U.S. Federal Government Agencies or Departments
9700	Automated Referral Service
9701	Visa Credential Service
9702	EMERGENCY SERVICES
9751	U.K. Supermarkets - Electronic Hot File
9752	U.K. Petrol Stations - Electronic Hot File
9950	Intra-Company Purchases