UNIFIED PAYMENTS
...innovations that work for you

API DOCUMENT FOR CIPA SCHEMES CONSOLIDATION

Ecommerce platforms

INTRODUCTION

This document describes the steps to be taken by merchants that want to receive Scheme (VisaCard, MasterCard, Amex, VerveCard and PayAttitude) details on their respective sites and process the transactions on our Payment Gateway.

NOTE: This document is for merchants that have been setup (by E-Commerce Ops) on the payment gateway.

Preliminary Step

1. GO TO https://test.payarena.com/prospectivemerchants on your browser

2. Fill in details in the form that will be presented to you.

3. Submit the form

4. Test Configuration details will be forwarded to you via the email you provided. This will include the following:
   - Merchant ID
   - Cryptographic Key

The Merchant ID and Cryptographic key are used to identify a Webshop on UP MPI each time a Webshop sends a payment request, i.e. a create order request to UP MPI.

IMPLEMENTATION

For a seamless integration, kindly follow the outlined steps below:

Step 1: Create a json representation of data elements as follows:

```
{
         “id”: “MERCHANTID”,
         “description”: “payment for goods”,
        “amount”: 200,
        “fee”: 0,
         “currency”: “566”,
        “returnUrl”: “http://mywebsite.com/returnurl”,
        “secretKey”: “AJAHD45S45F4S4S45AS45D54S”,
        “scheme”: “”,
         “vendorId”: “”
         "parameter" : "",
        "count" : 0
         }
```

Step 2: Send the json data created above as a POST request to https://test.payarena.com/Aggregator, the header should specify Accept and Content-Type as application/json. This will return an id for the transaction posted above.

Step 3: Generate a SHA1 string of your secret key.

Step 4: Create a json representation of the card details as follows:

```
{
        “secretKey”: “AJAHD45S45F4S4S45AS45D54S”,
         “scheme”: “visa”,
        “cardNumber”: “4999082100029373”,
         “expiry”: “01/19”,
         “cvv”: “126”,
         “cardholder”: “”,
         “mobile”: “”,
         “pin”: “”,
        }
```

NOTE: *When the scheme is either Visa or Mastercard or Amex, fields- cardholder, mobile and pin should be empty.*

When scheme is Verve, all fields must be supplied but when scheme is ***PayAttitude****, only the secret key and mobile fields should be populated, all others should be empty as seen below.*

```
{
“secretKey”: “AJAHD45S45F4S4S45AS45D54S”,
“scheme”: “”,
“cardNumber”: “”,
 “expiry”: “”,
 “cvv”: “”,
 “cardholder”: “”,
 “mobile”: “07087419908”,
 “pin”: “”,
}
```

Step 5: Using the AES algorithm, sample code provided below, encrypt the json data in Step 4 above using the first 16 characters of the string obtained in Step 3. See an example below:

C#

```
        public static string Encrypt(byte[] dataToEncrypt, byte[] key, byte[] iv)
{
        using (var aes = new AesCryptoServiceProvider())
         {
        aes.Mode = CipherMode.CBC;
         aes.Padding = PaddingMode.PKCS7;
        aes.Key = key;
          aes.IV = iv;
        using (var memoryStream = new MemoryStream())
         {
        var cryptoStream = new CryptoStream(memoryStream, aes.CreateEncryptor(),
CryptoStreamMode.Write);
         cryptoStream.Write(dataToEncrypt, 0, dataToEncrypt.Length);
cryptoStream.FlushFinalBlock();
         return memoryStream.ToArray().Aggregate("", (current, txt) => current +
txt.ToString("X2"));
}

}

}
```

where dataToEncrypt is a byte array of the json card details in step 4 above and key and iv are byte arrays of the first 16 characters in step 3 above.

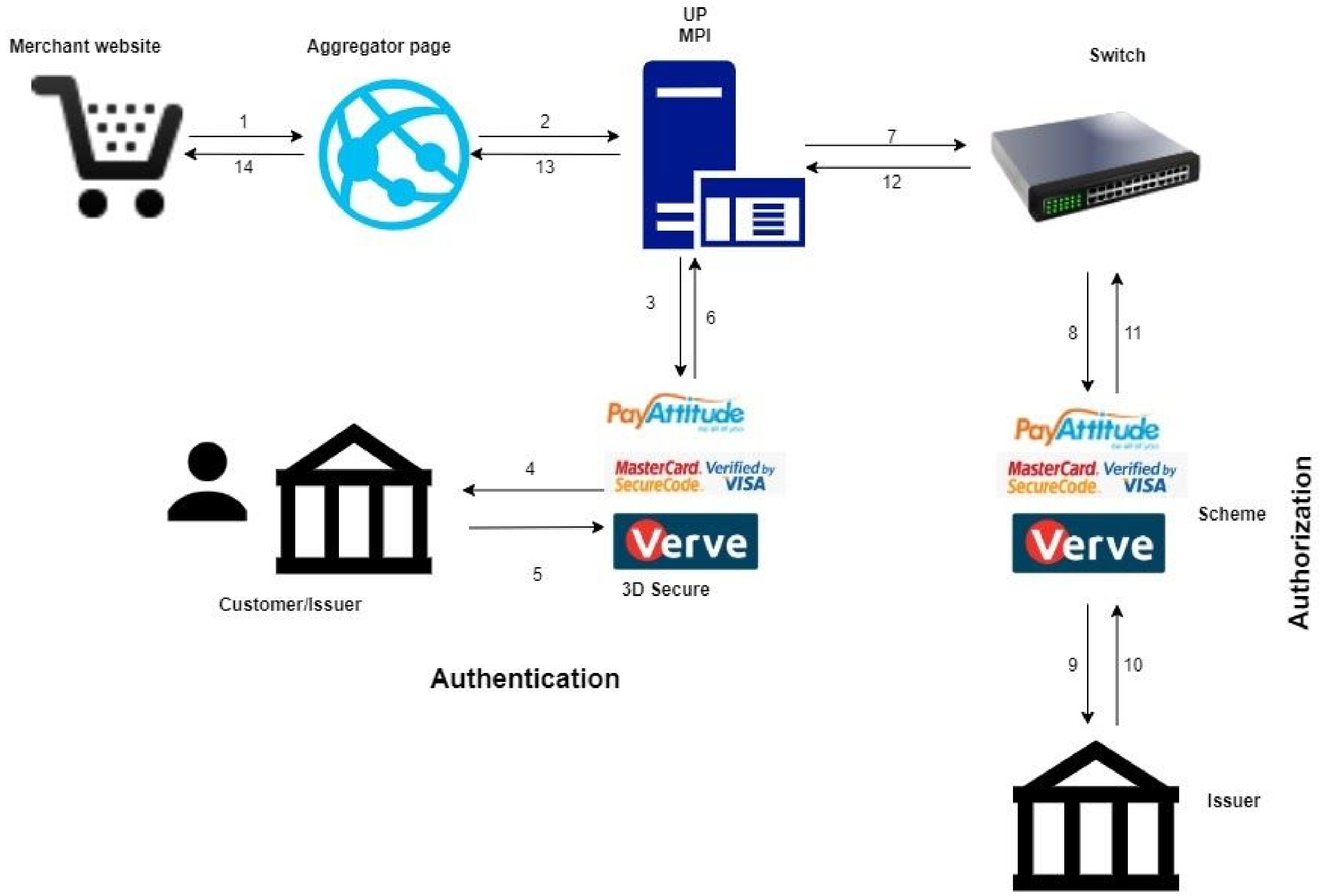
Step 6: Redirect the customer to the endpoint below as a GET request to complete the transaction:

https://test.payarena.com/Home/TransactionPost/TRANSACTIONID?mid=MERCHANTID&payload=ENCRYPTEDDATA

where

- Transactionid was gotten in step 2 above,
- Merchantid is unique ID used to identify the merchant an
- Encrypteddata as gotten in step 5 above.

Transaction Process Flow



Transaction Process Flow

1. Customer checks out from WebShop and enters payment details
2. The aggregator captures the data and sends it to UP MPI (in the case of card data its encrypts the data before sending to the UP MPI)
3. . UP MPI collects card details and sends authentication requests to the 3DSecure Provider.
4. The 3D Secure Provider contacts the issuer/cardholder to authenticate the transaction
5. The issuer/cardholder authenticates transaction and sends a response to 3D Secure Provider
6. The 3D Secure Provider sends the authentication response to UP MP
7. UP MPI sends an authorization request to the Switch
8. UP Switch sends the authorization request to Scheme
9. The Scheme contacts issuer to authorize
10. The Issuer authorizes the the transaction and sends the authorization response to the scheme
11. The Scheme sends the authorization response to the Switch
12. The Switch sends the authorization response to UP MPI.
13. UP MPI sends the payment confirmation response to the aggregator.
14. The aggregator sends the payment confirmation response to the webshope. The Webshop logs the response and displays payment confirmation to the customer with the details received from UP MPI.

TRANSACTION STATUS QUERY

To query the status of a transaction, send a GET request in the format below.

https://test.payarena.com/Status/Transactionid . Where transactionID is the ID received as a response after the order was created.

The status is the status of the transaction, this could be either of the following in the table below:

Status	**Description**
Approved/Approved Successful	A transaction is approved when the customer is successfully debited and value for transaction is received.
Cancelled	A transaction is cancelled when the customer decides to not enter payment details and returns back to the merchant site.
Declined	A transaction is declined when one of the following happens: 1.Unsucccessful authentication 2. Unsuccessful authorization 3. System error
Initiated	A transaction is initiated when the customer abandons a transaction.