BML PAYMENT GATEWAY
DEVELOPER’S GUIDE
VERSION 1.1

Table of Contents

Confidentiality Statement

This document contains information that is confidential or proprietary to Bank of Maldives Plc (or its direct or indirect subsidiaries). By accepting this document, you agree that: (1) if there is any pre-existing contract containing disclosure and use restrictions between your company and Bank of Maldives Plc, you and your company will use this information in reliance on and submit to the terms of any such pre-existing context; or (2) if there is no contractual relationship between you or your company and Bank of Maldives Plc, you and your company agree to protect this information and not reproduce, disclose or use the information in any way, except as may be required by law.

Disclaimer

The screens and illustrations in this document are representative of those created by the software and are not always exact copies of what appears on the computer monitor. Companies, names and data used in examples herein are fictitious unless otherwise noted. The material in this document is for information only, and is subject to change without notice. Bank of Maldives Plc reserves the right to make changes in the product without reservation and without notice to users.

1 INTRODUCTION

1.1 Purpose

This document contains information relevant to online merchants who are interested in routing their online transactions to 3D Secure MPI for processing. 3D Secure MPI complies with Visa, MasterCard and Amex.

1.2 Transaction Flow

1. Cardholder selects to checkout at the merchant website, merchant server will prepare a checkout page with all the fields required in **PxReq** Depends on the merchant Interface type. If the merchant is Direct Merchant, Card Information capture screen (checkout page) should prompt by the merchant.

   If the merchant is Re-Direct Merchant, Card Information capture screen (checkout page) will prompt by the MPI on behalf of Merchant But still merchant needs to send the other required fields in PxReq.
   On the checkout page, cardholder is prompted to choose the card brand (Visa/MasterCard/American Express) for the transaction payment, refer below:

2. For Visa, MasterCard and American Express card brand, merchant/MPI will provide card information capture screen based on the interface type for cardholder to enter the card information like card number, expiry date and cvv2.

3. 3D Secure MPI receives **PxReq** and validates the transaction, if invalid transaction, PxRes will post to Merchant URL with error code “203”. Merchant server must response accordingly to cardholder browser.

4. 3D Secure MPI will process valid transaction to determine whether the credit card is enrolled for 3D authentication. If the card is enrolled, 3D authentication will be performed else normal authorization will be performed.

5. During 3D authentication, the cardholder is required to enter password.

6. If wrong password, **PxRes** will post to Merchant URL with error code “202”. Else when the transaction is successful; **PxRes** will post to Merchant URL with reason code “1”. Merchant server must response to cardholder browser accordingly.

2 TECHNICAL REQUIREMENT

2.1 Transport Security

The channel between cardholder and merchant server should be secured using HTTPS. The merchant web server has to have a valid SSL certificate signed by a certificate authority installed.

2.2 HTTPS Post

Interaction between 3D Secure MPI and Merchant server are through HTML form post. The checkout form prepares field in PxReq, and post to 3D Secure MPI. The response of the transaction PxRes is posted to a merchant server URL.

2.3 Cardholder Browser

Merchant must provide clear instruction at the checkout page to avoid cardholder close browser window/hit refresh button/hit back button while the transaction is processing.

If the cardholder close browser in the middle of 3D authentication, the payment summary may not be posted to the merchant server.

3D Secure MPI uses JavaScript for the redirection of 3D authentication. Cardholder should enable JavaScript at the browser.

2.4 Response Time Frame

Time taken for 3D authentication is variable. It depends on the network traffic, and response time from Visa/Master as well as ACS. Typically, 3D authentication should complete within 20 sec.

2.5 Settlement

3D Secure MPI will automatically settle all approved transactions with the acquiring bank daily at 10 pm.

2.6 URLs

2.6.1 PxReq URL

Merchant server will be given a URL to received PxReq. This URL is given to merchant solely for online transaction request. Merchant should not disclose the URL to any unrelated third party.

PxReq URL: https://ebanking.bankofmaldives.com.mv/bmlmpiuat/threed/MPI

2.6.2 PxRes URL

Merchant server needs to provide 3D Secure MPI a URL, which is capable of accepting response (PxRes) from 3D Secure MPI.

3 MESSAGE TYPE

3.1 Message Description

- **PxReq For Re-Direct Merchant**
  This is the request message from a Re-Direct Merchant type to MPI/MPG to initiate the transaction process. Please refer below for the details of fields. This excludes card information. In this case, MPI/MPG will prompt the card information capture screen.

- **PxRes to the Merchant**
  This is the response message of the transaction from MPI/MPG, which will be posted to the Merchant server to the URL given for each transaction.

3.2 PxReq for Re-Direct Merchant – HTML Form Fields (Visa/MasterCard/AMEX)

1. Version

This is the version of the MPG and currently system default the value as 1.1

2. MerID

This is your Merchant ID as set in MPG (will be provided by your MPG Provider).

3. AcqID

This is your Acquirer ID (will be provided by your MPG Provider).

4. MerRespURL (Not Required)

This is the URL of a Web Page on your server where the response will be sent.

This field is not required to be sent any more as it is a onetime setup in MPG for each merchant ID. So this URL is not required to be sent for every transaction.

5. PurchaseCurrency (Optional)

This is the standard ISO code of the currency used for the transaction.
This field is not required to be sent any more as it is a onetime setup in MPG for each merchant ID.

6. PurchaseCurrencyExponent (Optional)

This is the number of decimal places used in the amount (usually 2).
This field is not required to be sent any more as it is a onetime setup in MPG for each merchant ID.

7. OrderID

This is your own Order ID that will be used to match your Orders with MPG transactions.
Order ID should always be unique.

8. SignatureMethod

This is the signature method used to calculate the signature (explained below). It is SHA1

9. PurchaseAmt

This is the total amount of the purchase. The PurchaseAmt is straight forward, it is basically the normal amount eg: 169.80, but without the decimal place and left padded with zeros in order to make the number twelve characters long. Therefore the amount 169.80 should be 000000016980.

10. Signature

This is a digital signature that will verify that the contents of this Web Page will not be altered in transit. (MPG will verify this signature). Signature will be discussed later on in this document.

3.3 PxRes – HTML Form Fields (Visa/MasterCard/AMEX)

The response is made of several parts. The first is the Response Code which is the most important part of the equation.
This will equal to 1, 2 or 3. Basically 1 means that the transaction was authorized, 2 means it was declined and 3 means that an error has occurred.

Data Sent with Response Codes

Response Code 1

- Response Code
- Reason Code
- Reason Description
- Merchant ID
- Acquirer ID
- Merchant Order ID
- Signature
- Reference Number
- Card Number (padded)
- Authorization Code

Response Code 2

- Response Code
- Reason Code
- Reason Description
- Merchant Order ID

Response Code 3

- Response Code
- Reason Code
- Reason Description
- Merchant Order ID

4 REASON CODES/ERROR CODES

The following are reason/ error codes in the PxRes.

Reason/ Error Code	Description
1XX	Invalid input to 3D Secure MPI
2XX	Error related to 3D authentication
3XX	Error related to authorization
4XX	System error or timeout

Reason/ Error Code Listing

Reason/ Error code	**Description**	**Action / Remarks**
000	Transaction is successful	Merchant to display the confirmation page to cardholder
101	Invalid field passed to 3D Secure MPI	Merchant needs to check error description to find out what is wrong with the field. Authorization/Authentication not carried out.
201	Invalid ACS response format. Transaction is aborted.	Retry the transaction. If error persists, contact issuing bank.
202	Cardholder failed the 3D authentication, password entered by cardholder is incorrect and transaction is aborted	Merchant to display error page to cardholder
203	3D PaRes has invalid signature. Transaction is aborted	Retry the transaction. If error persists, contact issuing bank.
300	Transaction not approved	Transaction has failed authorization, e.g. due to insufficient credit, invalid card number, etc. The actual response code provided by acquiring host can be found via the View Transaction History web page available to merchants.
301	Record not found	• Merchant/User has submitted a transaction with invalid purchase ID. • Merchant/User tried to reverse a previously declined transaction
302	Transaction not allowed	▪ Purchase ID not unique due to mismatched card number and/or transaction amount ▪ System unable to process reversal due to transaction has been settled ▪ System unable to process reversal due to transaction type is CAPS

Reason/ Error code	Description	Action / Remarks
		▪ System unable to process previously voided transaction
	Transaction is aborted. Invalid Request	Transaction request (PxReq) not includes ‘CardNo’, ‘CardExpDate’, ‘CardCVV2’.
303	Invalid Merchant ID	Not a valid merchant account
304	Transaction blocked by error 901	Merchant to report error to acquiring bank
308	Transaction is aborted. The Transaction was cancelled by the user.	The transaction was cancelled by the user.
900	3D Transaction timeout	Timeout of 3D transaction due to late response from Issuer ACS, after the predefined 3D timeout set in the application.
901	System Error	System unable to complete transaction. Merchant to report error to acquiring bank.
902	Time out	Issuing/acquiring host timeout, transaction is not approved

Note: Above are general reason codes/error codes. More specific reason codes/error codes with description may exist on the response. So always retrieve the details from response data.

5 SIGNATURE REQUIREMENT

A hash signature is required when using either of the methods. The SHA1 hash is a security feature that enables your script to identify that the results of a transaction are actually from the appropriate authorization source and also for the Payment Gateway Server to make sure the integrity of data received on a transaction request. Using SHA1 algorithm, a unique signature or fingerprint of the transaction can be created. The mathematical algorithm used to construct this signature is designed in such a way that any change to the information used in the calculation of the signature will cause a completely different signature to be created.

Also, the information used in the calculation of the signature cannot be discovered through any analysis of the signature itself. This is done by using information from your account. Every transaction that is processed through the system has a corresponding hash signature of the transaction created during the transaction process.
This signature must be included in the request of every transaction.

The signature for a request is a hash of the following six fields:

1. Password (this is the password included in the mailer that we provided during your account setup).
2. Merchant ID
3. Acquirer ID
4. Order ID (the Order Id that your system generates)
5. Purchase Amount

6. Purchase Currency

The fields must be set in the following order:
(**Password** + **Merchant ID** + **Acquirer ID** + **Order ID** + **Purchase Amount** + **Purchase Currency**)

For an example:

If your password is "**orange**", your Merchant Id is "**9809999999**", your Acquirer Id is "**407387**", the Order Id is "**MPGORDID01154321**", the amount is "**12.00**" and Purchase Currency is "**462**".

The hash would be calculated on the following string:
orange9809999999407387MPGORDID01154321000000001200462

The resulting hash signature value equals to (using SHA1 algorithm and encoded as a base 64 string):
Mqds0pzeurxv17Jgd7VI6wZIhlE=

If the authorization request is successful MPG will create a second SHA1 hashed signature and include it in the response message. Therefore when you receive the response data you can verify that the values were not modified in transit. The hash signature for the response is a hash of the following fields:

1. Password (this is the password included in the mailer that we provided).
2. Merchant Id
3. Acquirer Id
4. Order Id (the Order Id that you included in the request)
5. Response Code
6. Reason Code/Error Code

The fields must be set in the following order:
(**Password** + **MerchantID** + **AcquirerId** + **OrderID + Response Code + Reason Code**)

For an example:
If your password was "**orange**", your Merchant Id was "**9809999999**", your Acquirer Id was "**407387**" and the Order Id was "**MPGORDID01154321**".

The hash would be calculated on the following string:
orange9809999999407387MPGORDID01154321

The resulting hash signature value equals to: (using SHA1 algorithm and encoded as a base 64 string):
nJ0FThY0MlrFoa3iUOwmvSCFEFA=

When MPG receives the request of the transaction order, it will check if the Hash value it will generate matches the value you have included. When your script receives the results of the transaction you can create the hash on your side and compare it to the one sent by MPG to be sure that both are the same. You will already know your password and Merchant Id, and will receive the Order Id as presented to us.

A developer would take the results of the transaction AFTER it was returned to your server, and run the hash algorithm on the fields mentioned above. The only way the results of the developer's processing can match the signature included with the transaction results is if the password used in the hash on the developer's end MATCHES the one used in the transaction.

The MPG password is a shared secret (between you and the server), and is one of the key pieces of information in the hash. One can be assured that if the signature generated on your end matches the one sent with the transaction, then the transaction has in fact been processed by our system, and has not been posted back to the merchant's server from any other location.
The MPG password is generated automatically by our system and is sent to you. This is also used to authenticate your server to our server during transaction posting. It is only known by the merchant and the System.
The SHA1 is called secure because it is computationally infeasible to find a message which corresponds to a given message digest, or to find two different messages which produce the same message digest. Any change to a message in transit will, with very high probability, result in a different message digest, and the signature will fail to verify.

5.1 Signature Generation Sample Code

```
public class PxSignatureUtil {
      private static MessageDigest SHA1;

      static {
          try {
              SHA1 = MessageDigest.getInstance("SHA1");

          } catch(NoSuchAlgorithmException nae) {
              throw new RuntimeException(nae);
          }
      }

      public static String getPxSignature(String signatureString) throws
Exception {
            String signature = "";
            try{
                  SHA1.reset();
            byte[] toEncrypt = signatureString.getBytes("UTF-8");

            SHA1.update(signatureString.getBytes());

            }catch(UnsupportedEncodingException uee){
                  throw new RuntimeException(uee);
            }

            byte[] encryptedRaw = SHA1.digest();
        byte[] encoded = BmlBase64.encode(encryptedRaw);

        try {
            signature = new String(encoded, "UTF-8");
        } catch (UnsupportedEncodingException uee) {
            throw new RuntimeException(uee);
        }
        return signature;
      }
   }
```

6 SAMPLE CODE

6.1 Sample Request Page

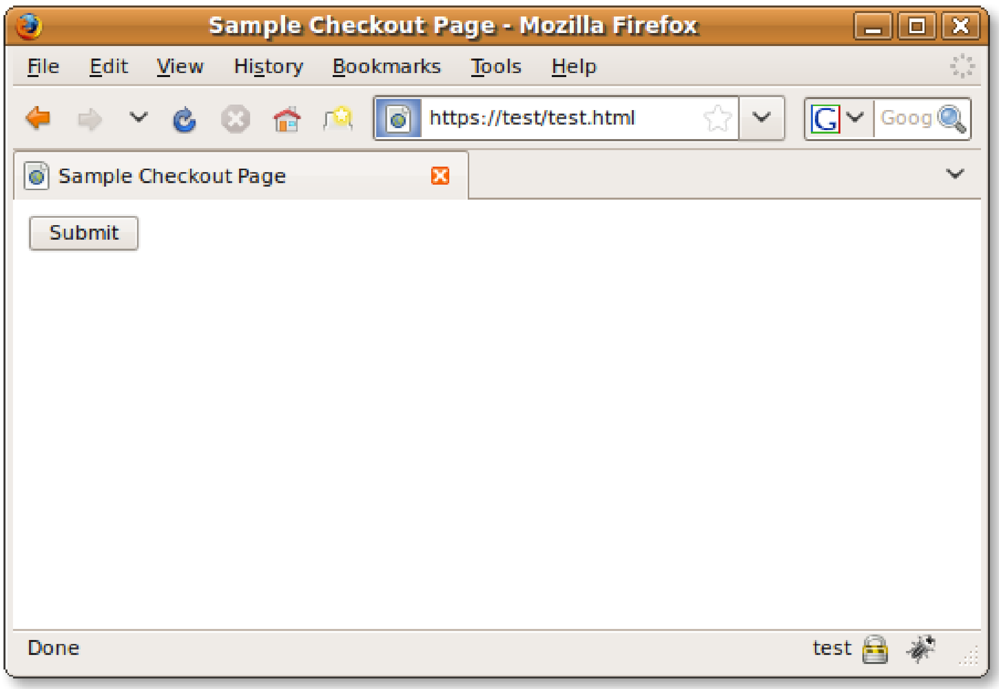
The way MPG works is that it waits for you to request a particular page with the required data. As soon as it receives a request for that page, MPG will validate the data that you sent and will start processing the transaction. Please find below a simple page that can be stored in your web server and when loaded will show only a checkout button. As soon as you press that button, the request will be sent to MPG and the processing will initiate (if you would like to experiment by using this example note that some of the fields' values will have to change according to the data that you Acquirer will provide otherwise the request will be rejected. You will also need to calculate the signature correctly).

Here is the HTML code of the page:

```
<html>
      <body>
      <formmethod=”post”
action=”https://ebanking.bankofmaldives.com.mv/bmlmpiuat/threed/MPI”>
             <inputtype=”hidden” name=”Version” value=”1.1” />
             <inputtype=”hidden” name=”MerID” value=”9809999999” />
             <inputtype=”hidden” name=”AcqID” value=”407387” />
             <inputtype=”hidden” name=”PurchaseCurrency” value=”462” />
             <inputtype=”hidden” name=”PurchaseCurrencyExponent” value=”2” />
             <inputtype=”hidden” name=”OrderID” value=”MPGORDID01154321” />
             <inputtype=”hidden” name=”SignatureMethod” value=”SHA1” />
             <inputtype=”hidden” name=”PurchaseAmt” value=”000000001200” />
             <inputtype=”hidden” name=”Signature” value=”MqdsOpzeurxv17Jgd7VI6wZIhlE=” />

             <inputtype=”submit” name=”submit” value=”Submit” />
      </form>
      </body>
</html>
```

The page is as simple as it can be and it should look like the one below:


Sample Checkout Page - Mozilla Firefox
File Edit View History Bookmarks Tools Help
https://test/test.html
Goog
Sample Checkout Page
Submit
Done
test

By pressing the submit button, the request will be sent to MPG for authorization. In a few fractions of a second the checkout page should load. On the checkout page card details must be entered to proceed and finally the transaction will complete.

6.2 Sample Response Page

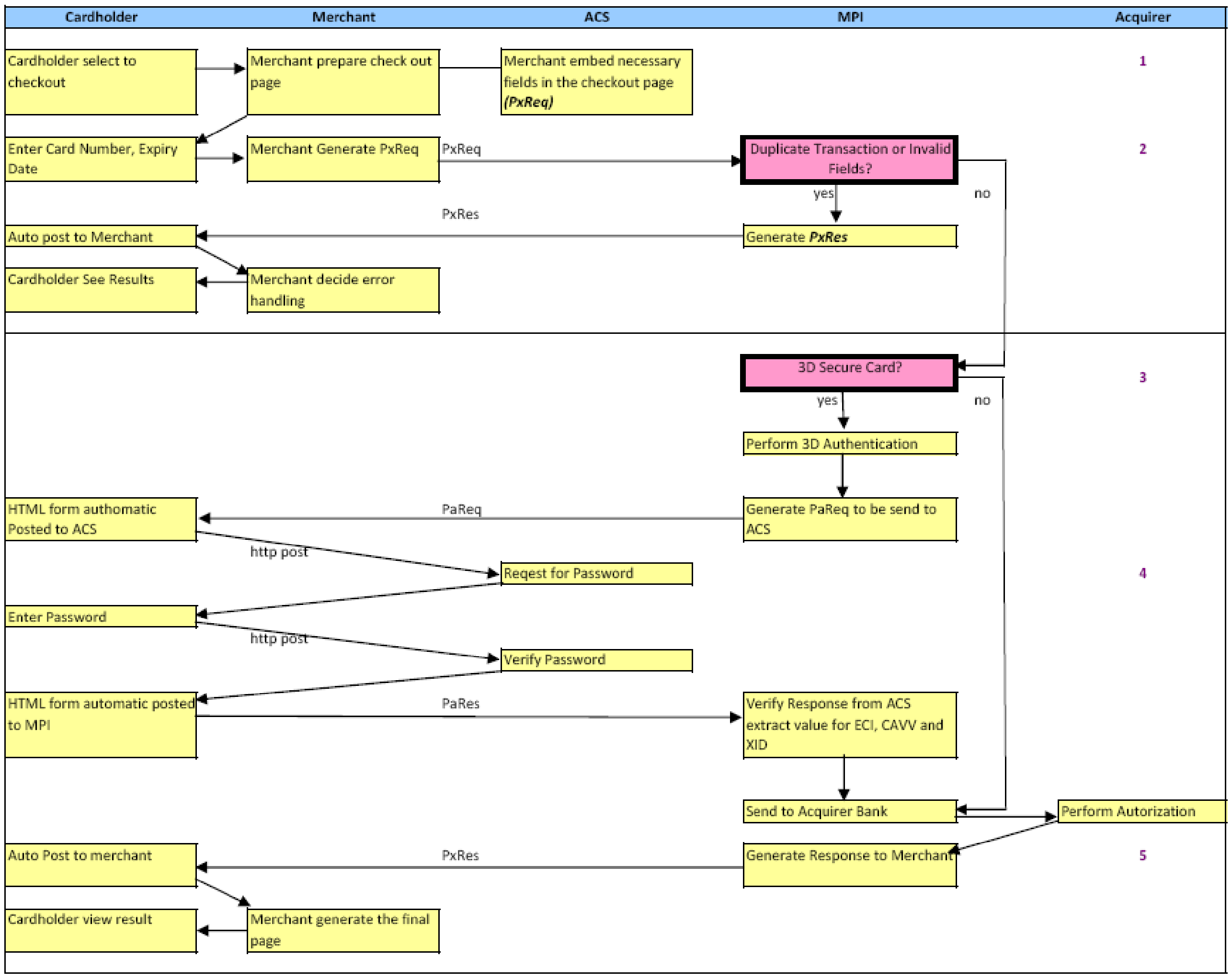
By this point you have all the information required to send a request to MPG. The only problem remaining is the response page. In order for you to know if the transaction was successful or not, MPG will send response to the Response URL or PxRes URL (preconfigured on your server and shared with MPG provider) when the authorization request completes.
Response page should read the response from MPG and should update the order of your customer and at the same time it should show a message indicating to the customer if the transaction was successful or not. The method used to update the order and display the information to the customer is up to you and there are several methods of achieving this, which can range from very simple to very complex depending on how your business logic is enforced and the complexity of your website.

The following VB.Net snippet will simply read the values and display them on a placeholder (label) on the form that the customer will see. This method is usually too simple to be used in a real environment but can nevertheless form the basis of a more complex and solid one.

```
Private Sub Page_Load(ByVal sender As System.Object, ByVal e As
                                   System.EventArgs) Handles MyBase.Load
      If Page.IsPostBack = False Then
                     Dim FormKey As String
                     Dim FormValue As String
                     lblAllFormItems.Text = “”
                     lblAllFormKeys.Text = “”
                     For Each FormKey In Request.Form
                            FormValue = Request.Form.Item(FormKey)
                            If FormValue <> String.Empty Then
                                   lblAllFormKeys.Text &= “<br /><b>” & FormKey & “:</b>”
                                   lblAllFormItems.Text &= “<br />” & FormValue
                            End If
                     Next
      End If
End Sub
```

In overall, these are the following required to develop from the Merchant Side:
A Web Page to call BML payment gateway
A Response Page (Please note that SSL is vital for your Response Page)

APPENDIX A – 3D SECURE TRANSACTION FLOW DIAGRAM


Cardholder
Merchant
ACS
MPI
Acquirer
1
Cardholder select to checkout
Merchant prepare check out page
Merchant embed necessary fields in the checkout page (PxReq)
2
Enter Card Number, Expiry Date
Merchant Generate PxReq
PxReq
Duplicate Transaction or Invalid Fields?
yes
no
PxRes
Auto post to Merchant
Generate PxRes
Cardholder See Results
Merchant decide error handling
3
3D Secure Card?
yes
no
Perform 3D Authentication
HTML form authomatic Posted to ACS
PaReq
Generate PaReq to be send to ACS
http post
Reqest for Password
4
Enter Password
http post
Verify Password
HTML form automatic posted to MPI
PaRes
Verify Response from ACS extract value for ECI, CAVV and XID
Send to Acquirer Bank
Perform Autorization
5
Auto Post to merchant
PxRes
Generate Response to Merchant
Cardholder view result
Merchant generate the final page