Guide to LBP ePayment Web Service

Version : 1.0
Draft/Final as of : March 2015
Author : LANDBANK
Owner : TMG - eBSD

Document Information

Approvals

This document has been approved by:

Name	Position/Title	Approval Date	Signature
Arthur E. Dalampan	*ITM/Head–eBSD*		
Janise Supan-Bonggo	*Team Lead-eGov*		

Distribution

This document has been distributed to:

Name	Position/Title

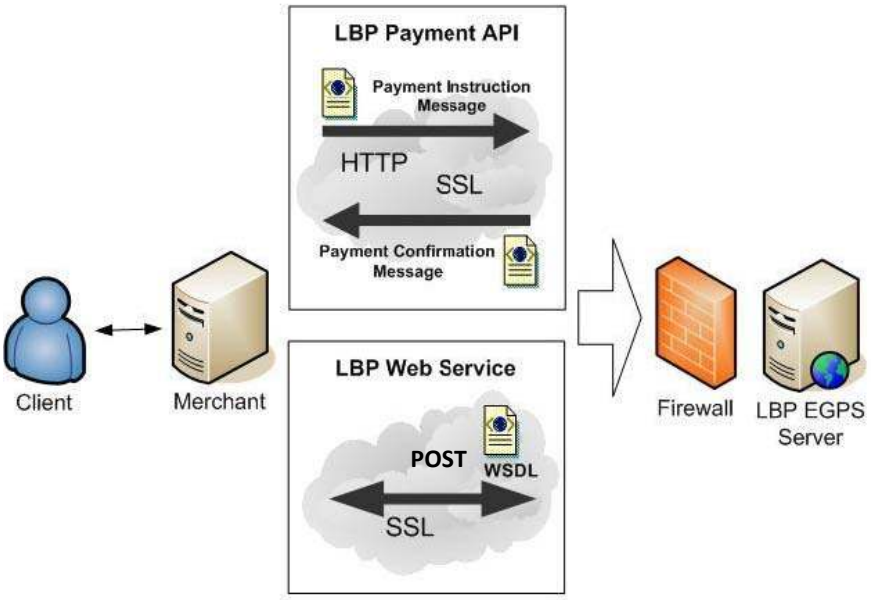
TABLE OF CONTENTS

1.0 Introduction

This document defines the logical flow and use of the LBP Security Library of the LBP ePayment Application. This shall serve as the basis for the construction and encryption of messages between Landbank and its partner Agencies/Merchants transacting via the LBP ePayment Gateway. The functions described below shall be incorporated in partner Agencies/Merchants' Applications and shall be used in accordance with the rules described below in sending or receiving messages to and from Landbank.

2.0 System Overview

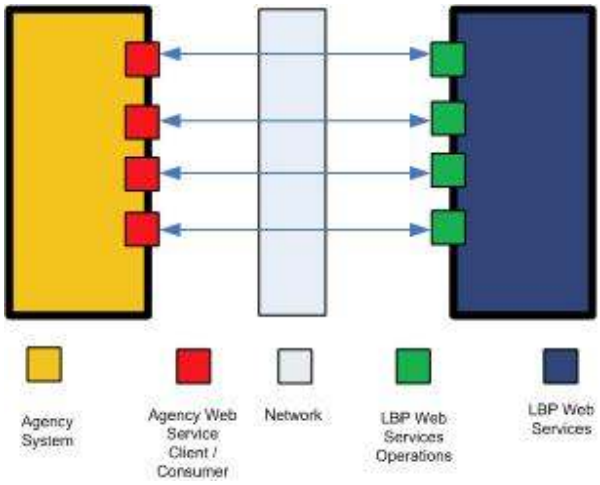
The Electronic Government Payment System is an Internet payment system of LANDBANK interfaced with the Payment of an Agency to provide the Bank's clients, both private and government entities, with an alternative payment collection of fees via the Internet.



2.1 High Level Interface Design

One of the bank's payment interfaces for the Electronic Government Payment System (EGPS) is implemented through calling of web services published over the internet. EGPS has several web services as part of its application.

The Agency's payment system shall call EGPS web services to initiate payment. Once received, EGPS shall process the request accordingly. The communication using web services is illustrated in a simple diagram below.



3.0 Design Assumptions and Constraints

The following shall be the working assumptions and constraints in coming up with the technical design of the bank payment interface:

1. The communication protocol between EGPS and the Agency System will be through Hypertext Transfer Protocol Secure (HTTPS)
2. The Agency System should be capable of interacting with EGPS Web Services.
3. The Agency System can be customized to meet the interface requirements of EGPS Services.
4. The Agency System should be able to handle situations of partial failure. This refers to situations wherein network, client or service becomes inaccessible or busy.

4.0 Detailed Interface Design

4.1 Web Service Information

EGPS functions using the following framework:

Web Service Types	SOAP / REST
SOAP Version	1.1
REST Version (RESTEasy)	2.2.1
Transport Protocol	HTTP / HTTPS
Web Service Publisher	IIS/Tomcat 6.0
HTTP Port	80 / 443 / 8080

4.2 Security Architecture

The following security mechanisms are applied in the web service and communication exchanges to ensure that payment data and its transmission will not be plainly visible and interrupted:

1. **Hashing**

   The application will require a signature string that has been hashed using a keystore file and some special functions (indicated below). This will then be compared to the signature generated by EGPS.

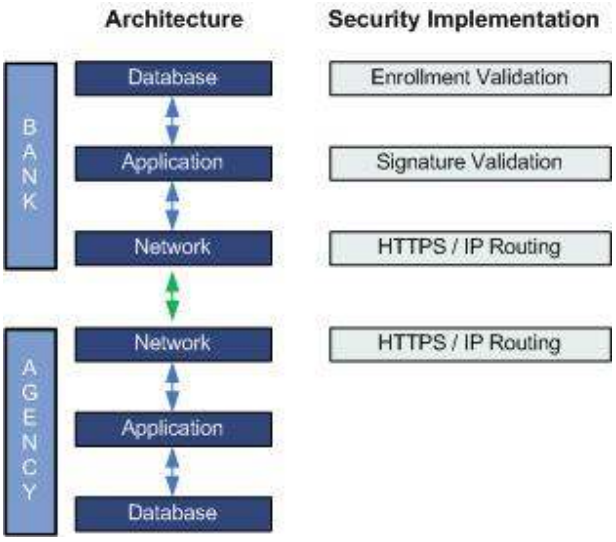
2. **HTTPS**

   The transportation of data will be via HTTPS. This will encrypt the data being passed over the internet.

3. **Point-to-Point Connection (IP Routing)**

   Moreover, it is recommended to implement IP routing or point-to-point connection between the Agency and the bank. The former will depend on the agency's readiness (i.e. all nodes connecting to central server)

Below is a simple diagram outlining the security mechanisms needed to be implemented on various levels of the EGPS and Agency's Payment System application.



4.3 Web Service Operations (SOAP)

LBPGatewayWS

Method	**processPayment**
Description	Web service method for processing ATM payment
Result	String (URL)

Parameter	**Data type**	**Description**
messageType	NUM(2)	LBP-assigned value
transactionType	NUM(2)	LBP-assigned value
paymentMode	NUM(2)	LBP-assigned value
currencyCode	NUM(3)	LBP-assigned value
username	ALPHANUM(30)	To be provided by the Merchant
password	ALPHANUM(30)	To be provided by the Merchant
productId	NUM(4)	LBP-assigned value
referenceNumber	ALPHANUM(30)	To be provided by the Merchant; used as identifier for duplicate transactions
transactionNumber	ALPHANUM(15)	To be provided by the Merchant
messageDate	ALPHANUM(8)	To be provided by the Merchant; format: CCMMDDYY
messasgeTime	ALPHANUM(6)	To be provided by the Merchant; format: HHMMSS
paymentDate	ALPHANUM(8)	To be provided by the Merchant; format: CCMMDDYY
paymentAmount	NUM(16,2)	To be provided by the Merchant; Sum of all payments
paymentDetail	ALPHANUM(30)	To be provided by the Merchant
payeeCode	ALPHANUM(4)	LBP-assigned value
payeeName	ALPHANUM(25)	To be provided by the Merchant
payerCode	ALPHANUM(4)	LBP-assigned value
payerName	ALPHANUM(25)	To be provided by the Merchant
creditAccountNumber	ALPHANUM(25)	LBP-assigned value
debitAccountNumber	ALPHANUM(25)	LBP-assigned value
toBankCode	ALPHANUM(6)	LBP-assigned value
fromBankCode	ALPHANUM(6)	LBP-assigned value
otherDetail1	ALPHANUM (100)	To be provided by the Merchant
otherDetail2	ALPHANUM (100)	To be provided by the Merchant
otherDetail3	ALPHANUM (100)	To be provided by the Merchant
otherDetail4	ALPHANUM (100)	To be provided by the Merchant
otherDetail5	ALPHANUM (100)	To be provided by the Merchant
signature	variable length	Signature for web service access

Method	**getPaymentStatus**
Description	Web Service method for payment inquiry
Result	String (ErrorCode)

Parameter	**Data type**	**Description**
messageType	NUM(2)	LBP-assigned value
transactionType	NUM(2)	LBP-assigned value
paymentMode	NUM(2)	LBP-assigned value
currencyCode	NUM(3)	LBP-assigned value
username	ALPHANUM(30)	To be provided by the Merchant
password	ALPHANUM(30)	To be provided by the Merchant
productId	NUM(4)	LBP-assigned value
referenceNumber	ALPHANUM(30)	To be provided by the Merchant; used as identifier for duplicate transactions
transactionNumber	ALPHANUM(15)	To be provided by the Merchant
messageDate	ALPHANUM(8)	To be provided by the Merchant; format: CCMMDDYY
messasgeTime	ALPHANUM(6)	To be provided by the Merchant; format: HHMMSS
paymentDate	ALPHANUM(8)	To be provided by the Merchant; format: CCMMDDYY
paymentAmount	NUM(16,2)	To be provided by the Merchant; Sum of all payments
paymentDetail	ALPHANUM(30)	To be provided by the Merchant
payeeCode	ALPHANUM(4)	LBP-assigned value
payeeName	ALPHANUM(25)	To be provided by the Merchant
payerCode	ALPHANUM(4)	LBP-assigned value
payerName	ALPHANUM(25)	To be provided by the Merchant
creditAccountNumber	ALPHANUM(25)	LBP-assigned value
debitAccountNumber	ALPHANUM(25)	LBP-assigned value
toBankCode	ALPHANUM(6)	LBP-assigned value
fromBankCode	ALPHANUM(6)	LBP-assigned value
otherDetail1	ALPHANUM (100)	To be provided by the Merchant
otherDetail2	ALPHANUM (100)	To be provided by the Merchant
otherDetail3	ALPHANUM (100)	To be provided by the Merchant
otherDetail4	ALPHANUM (100)	To be provided by the Merchant
otherDetail5	ALPHANUM (100)	To be provided by the Merchant
signature	variable length	Signature for web service access

ADALBPCaptureWS

Method	**processPayment**
Description	Web service method for processing ADA payment
Result	Pipe Delimited String: paymentMode referenceNumber transactionNumber paymentAmount payeeCode confirmationDate confirmationTime confirmationNumber tranStatus errorCode

Parameter	**Data type**	**Description**
messageType	NUM(2)	LBP-assigned value
transactionType	NUM(2)	LBP-assigned value
paymentMode	NUM(2)	LBP-assigned value
currencyCode	NUM(3)	LBP-assigned value
username	ALPHANUM(30)	To be provided by the Merchant
password	ALPHANUM(30)	To be provided by the Merchant
productId	NUM(4)	LBP-assigned value
referenceNumber	ALPHANUM(30)	To be provided by the Merchant; used as identifier for duplicate transactions
transactionNumber	ALPHANUM(15)	To be provided by the Merchant
messageDate	ALPHANUM(8)	To be provided by the Merchant; format: CCMMDDYY
messasgeTime	ALPHANUM(6)	To be provided by the Merchant; format: HHMMSS
paymentDate	ALPHANUM(8)	To be provided by the Merchant; format: CCMMDDYY
paymentAmount	NUM(16,2)	To be provided by the Merchant; Sum of all payments
paymentDetail	ALPHANUM(30)	To be provided by the Merchant
payeeCode	ALPHANUM(4)	LBP-assigned value
payeeName	ALPHANUM(25)	To be provided by the Merchant
payerCode	ALPHANUM(4)	LBP-assigned value
payerName	ALPHANUM(25)	To be provided by the Merchant
creditAccountNumber	ALPHANUM(25)	LBP-assigned value
debitAccountNumber	ALPHANUM(25)	LBP-assigned value
toBankCode	ALPHANUM(6)	LBP-assigned value
fromBankCode	ALPHANUM(6)	LBP-assigned value
otherDetail1	ALPHANUM (100)	To be provided by the Merchant
otherDetail2	ALPHANUM (100)	To be provided by the Merchant
otherDetail3	ALPHANUM (100)	To be provided by the Merchant
otherDetail4	ALPHANUM (100)	To be provided by the Merchant
otherDetail5	ALPHANUM (100)	To be provided by the Merchant
signature	variable length	Signature for web service access

4.4 Web Service Return Parameters (SOAP)

LBPGatewayWS

processPayment

Field	DataType/Length	Format	Description
String	variable length	URL	URL Address

getPaymentStatus

Field	DataType/Length	Format	Description
String	ALPHANUM(4)	String	EGPS Error Code

ADALBPCaptureWS (Pipe Delimited String)

processPayment

Field	DataType/Length	Description
paymentMode	NUM(2)	LBP-assigned value
referenceNumber	ALPHANUM(30)	To be provided by the Merchant; used as identifier for duplicate transactions
transactionNumber	ALPHANUM(15)	To be provided by the Merchant
paymentAmount	NUM(16,2)	Sum of all payments
payeeCode	ALPHANUM(4)	LBP-assigned value
confirmationDate	ALPHANUM(8)	LBP-assigned value; format: CCMMDDYY
confirmationTime	ALPHANUM(6)	LBP-assigned value; format: HHMMSS
confirmationNumber	ALPHANUM(20)	LBP-assigned value
tranStatus	ALPHANUM(1)	LBP-assigned value; A = successful transaction, R = rejected transaction
errorCode	ALPHANUM(4)	LBP-assigned value

4.5 Web Service Operations (ReST)

Description	Process Payment for ADA Transactions
URL	http://x.x.x.x/egps/rs/epayment/processada
HTTP Methods	POST
Result	Pipe Delimited String: PaymentMode ReferenceNum TransactionNum PaymentAmt PayeeCode ConfirmationDate ConfirmationTime ConfirmationNumber TranStatus ErrorCode

Parameter	Data type	Description
MessageType	NUM(2)	LBP-assigned value
TransactionType	NUM(2)	LBP-assigned value
PaymentMode	NUM(2)	LBP-assigned value
CurrencyCode	NUM(3)	LBP-assigned value
Username	ALPHANUM(30)	To be provided by the Merchant
Password	ALPHANUM(30)	To be provided by the Merchant
ProductID	NUM(4)	LBP-assigned value
ReferenceNum	ALPHANUM(30)	To be provided by the Merchant; used as identifier for duplication transactions
TransactionNum	ALPHANUM(15)	To be provided by the Merchant
MessageDate	ALPHANUM(8)	To be provided by the Merchant; format: CCMMDDYY
MessasgeTime	ALPHANUM(6)	To be provided by the Merchant; format: HHMMSS
PaymentDate	ALPHANUM(8)	To be provided by the Merchant; format: CCMMDDYY
PaymentAmt	NUM(16,2)	To be provided by the Merchant; Sum of all payments
PaymentDetail	ALPHANUM(30)	To be provided by the Merchant
PayeeCode	ALPHANUM(4)	LBP-assigned value
PayeeName	ALPHANUM(25)	To be provided by the Merchant
PayerCode	ALPHANUM(4)	LBP-assigned value
PayerName	ALPHANUM(25)	To be provided by the Merchant
CrAcctNum	ALPHANUM(25)	LBP-assigned value
DrAcctNum	ALPHANUM(25)	LBP-assigned value
BankCodeTo	ALPHANUM(6)	LBP-assigned value
BankCodeFrom	ALPHANUM(6)	LBP-assigned value
OtherDetails1	ALPHANUM (100)	To be provided by the Merchant
OtherDetails2	ALPHANUM (100)	To be provided by the Merchant
OtherDetails3	ALPHANUM (100)	To be provided by the Merchant
OtherDetails4	ALPHANUM (100)	To be provided by the Merchant
OtherDetails5	ALPHANUM (100)	To be provided by the Merchant
Signature	variable length	Signature for web service access

Example:

```
http://192.168.1.1:8080/egps/rs/
epayment/processada?MessageType=02&TransactionType=01&PaymentMode=03&CurrencyCode=000&
Username=0002&Password=yCk9iB4&ProductID=0000&ReferenceNum=399285771714730&TransactionN
um=056610565104102&MessageDate=20140414&MessageTime=143904&PaymentTime=20140414&Pay
mentAmt=120&PaymentDetail=ePayment transaction&PayeeCode=0005&Payee Name=Test
Payee&PayerCode=9988&PayerName=Test
Company&CrAcctNum=1234567890&DrAcctNum=1234567890&BankCodeTo=069041&BankCodeFrom=0
69040&OtherDetails1=Other Details&OtherDetails2=Other Details&OtherDetails3=Other
Details&OtherDetails4=Other Details&OtherDetails5=Other
Details&Signature=10a3Mli8mmUxXjl+u3+0oXeFwXhgL620xc3968Dipf
```

Description	**Process Payment for ATM Transaction**
URL	http://x.x.x.x/egps/rs/epayment/processpayment
HTTP Methods	POST
Result	String (URL Address)

Parameter	**Data type**	**Description**
MessageType	NUM(2)	LBP-assigned value
TransactionType	NUM(2)	LBP-assigned value
PaymentMode	NUM(2)	LBP-assigned value
CurrencyCode	NUM(3)	LBP-assigned value
Username	ALPHANUM(30)	To be provided by the Merchant
Password	ALPHANUM(30)	To be provided by the Merchant
ProductID	NUM(4)	LBP-assigned value
ReferenceNum	ALPHANUM(30)	To be provided by the Merchant; used as identifier for duplication transactions
TransactionNum	ALPHANUM(15)	To be provided by the Merchant
MessageDate	ALPHANUM(8)	To be provided by the Merchant; format: CCMMDDYY
MessasgeTime	ALPHANUM(6)	To be provided by the Merchant; format: HHMMSS
PaymentDate	ALPHANUM(8)	To be provided by the Merchant; format: CCMMDDYY
PaymentAmt	NUM(16,2)	To be provided by the Merchant; Sum of all payments
PaymentDetail	ALPHANUM(30)	To be provided by the Merchant
PayeeCode	ALPHANUM(4)	LBP-assigned value
PayeeName	ALPHANUM(25)	To be provided by the Merchant
PayerCode	ALPHANUM(4)	LBP-assigned value
PayerName	ALPHANUM(25)	To be provided by the Merchant
CrAcctNum	ALPHANUM(25)	LBP-assigned value
DrAcctNum	ALPHANUM(25)	LBP-assigned value
BankCodeTo	ALPHANUM(6)	LBP-assigned value
BankCodeFrom	ALPHANUM(6)	LBP-assigned value
OtherDetails1	ALPHANUM (100)	To be provided by the Merchant
OtherDetails2	ALPHANUM (100)	To be provided by the Merchant
OtherDetails3	ALPHANUM (100)	To be provided by the Merchant
OtherDetails4	ALPHANUM (100)	To be provided by the Merchant
OtherDetails5	ALPHANUM (100)	To be provided by the Merchant
Signature	variable length	Signature for web service access

Example:

http://192.168.1.1:8080/egps/rs
epayment/processpayment?MessageType=02&TransactionType=01&PaymentMode=03&CurrencyCode=000&Username=0002&Password=yCk9iB4&ProductID=0000&ReferenceNum=399285771714730&TransactionNum=056610565104102&MessageDate=20140414&MessageTime=143904&PaymentTime=20140414&PaymentAmt=120&PaymentDetail=ePayment transaction&PayeeCode=0005&Payee Name=Test Payee&PayerCode=9988&PayerName=Test Company&CrAcctNum=1234567890&DrAcctNum=1234567890&BankCodeTo=069041&BankCodeFrom=069040&OtherDetails1=Other Details&OtherDetails2=Other Details&OtherDetails3=Other Details&OtherDetails4=Other Details&OtherDetails5=Other Details&Signature=10a3Mli8mmUxXjl+u3+0oXeFwXhgL620xc3968Dipf

Description	**Check Transaction Status**
URL	http://x.x.x.x/egps/rs/epayment/inquiry
HTTP Methods	POST
Result	String (EGPS Error Code)

Parameter	**Data type**	**Description**
MessageType	NUM(2)	LBP-assigned value
TransactionType	NUM(2)	LBP-assigned value
PaymentMode	NUM(2)	LBP-assigned value
CurrencyCode	NUM(3)	LBP-assigned value
Username	ALPHANUM(30)	To be provided by the Merchant
Password	ALPHANUM(30)	To be provided by the Merchant
ProductID	NUM(4)	LBP-assigned value
ReferenceNum	ALPHANUM(30)	Reference Number of transaction for inquiry
TransactionNum	ALPHANUM(15)	To be provided by the Merchant
MessageDate	ALPHANUM(8)	Date of transaction for inquiry; format: CCMMDDYY
MessasgeTime	ALPHANUM(6)	Time of transaction for inquiry; format: HHMMSS
PaymentDate	ALPHANUM(8)	To be provided by the Merchant; format: CCMMDDYY
PaymentAmt	NUM(16,2)	Actual amount of transaction for inquiry
PaymentDetail	ALPHANUM(30)	To be provided by the Merchant
PayeeCode	ALPHANUM(4)	LBP-assigned value
PayeeName	ALPHANUM(25)	To be provided by the Merchant
PayerCode	ALPHANUM(4)	LBP-assigned value
PayerName	ALPHANUM(25)	To be provided by the Merchant
CrAcctNum	ALPHANUM(25)	LBP-assigned value
DrAcctNum	ALPHANUM(25)	LBP-assigned value
BankCodeTo	ALPHANUM(6)	LBP-assigned value
BankCodeFrom	ALPHANUM(6)	LBP-assigned value
OtherDetails1	ALPHANUM (100)	To be provided by the Merchant
OtherDetails2	ALPHANUM (100)	To be provided by the Merchant
OtherDetails3	ALPHANUM (100)	To be provided by the Merchant
OtherDetails4	ALPHANUM (100)	To be provided by the Merchant
OtherDetails5	ALPHANUM (100)	To be provided by the Merchant
Signature	variable length	Signature for web service access

Example:

http://192.168.1.1:8080/egps/rs/epayment/inquiry?MessageType=02&TransactionType=01&PaymentMode=03&CurrencyCode=000&Username=0002&Password=yCk9iB4&ProductID=0000&ReferenceNum=399285771714730&TransactionNum=056610565104102&MessageDate=20140414&MessageTime=143904&PaymentTime=20140414&PaymentAmt=120&PaymentDetail=ePayment transaction&PayeeCode=0005&Payee Name=Test Payee&PayerCode=9988&PayerName=Test Company&CrAcctNum=1234567890&DrAcctNum=1234567890&BankCodeTo=069041&BankCodeFrom=069040&OtherDetails1=Other Details&OtherDetails2=Other Details&OtherDetails3=Other Details&OtherDetails4=Other Details&OtherDetails5=Other Details&Signature=10a3Mli8mmUxXjl+u3+0oXeFwXhgL620xc3968Dipf

4.7 Security Functions

```
private  String getMac(String str, Key key) {

        Security.addProvider(new BouncyCastleProvider());

        Mac mac = Mac.getInstance("HmacSHA1");
        mac.init(key);
        byte[] digest = mac.doFinal(str.getBytes());

        return Arrays.toString(digest);
}


public static char[] encode(byte in[]) {

        int iLen = in.length();
        int oDataLen = (iLen * 4 + 2) / 3;
        int oLen = ((iLen + 2) / 3) * 4;
        char out[] = new char[oLen];
        int ip = 0;
        for(int op = 0; ip < iLen; op++)
        {
          int i0 = in[ip++] & 0xff;
          int i1 = ip >= iLen ? 0 : in[ip++] & 0xff;
          int i2 = ip >= iLen ? 0 : in[ip++] & 0xff;
          int o0 = i0 >>> 2;
          int o1 = (i0 & 3) << 4 | i1 >>> 4;
          int o2 = (i1 & 0xf) << 2 | i2 >>> 6;
          int o3 = i2 & 0x3f;
          out[op++] = map1[o0];
          out[op++] = map1[o1];
          out[op] = op >= oDataLen ? '=' : map1[o2];
          op++;
           out[op] = op >= oDataLen ? '=' : map1[o3];
        }

        return out;
}


//static character array map1[] initialization
static {
        map1= new char[64];
        int i = 0;
        for(char c = 'A'; c <= 'Z'; c++)
        map1[i++] = c;

        for(char c = 'a'; c <= 'z'; c++)
        map1[i++] = c;

        for(char c = '0'; c <= '9'; c++)
        map1[i++] = c;

        map1[i++] = '-';
        map1[i++] = '_';
}
```

```
private  String encrypt (String params) {

                Security.addProvider(new BouncyCastleProvider());

                MessageDigest md = MessageDigest.getInstance("SHA");
                md.update(params.getBytes());
                String md_value = new String(Base64.encode(md.digest()));

                //Generate SecretKey for this encryption
                KeyGenerator keygen = KeyGenerator.getInstance("DESede","BC");
                keygen.init(new SecureRandom());

                SecretKey skey = keygen.generateKey();
                String secretKey = new String(Base64.encode(skey.getEncoded()));

                Cipher cipher = Cipher.getInstance("DESede","BC");
                cipher.init(Cipher.ENCRYPT_MODE, skey);

                String encryptedParam = new String(Base64.encode(cipher.doFinal(params.getBytes())));
                String encryptAll = encryptedParam + secretKey + md_value;

                return encryptAll;
  }
```

4.8 Signature String Generation

The final signature string shall be the concatenated format of the prescribed fields passed through the Mac, which is afterwards Base64 encoded then finally encrypted.

*** signature = encrypt(encode(geMac(concatenated_params)))

Note:
As a result of these signature string generation algorithms, when processing the exact same set of data, no two signatures will be alike.

SAMPLE CODE (SOAP):

```
public void processRequest(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {

        LBPGatewayWSService lbpService = new LBPGatewayWSService();
        ALBPCaptureWS albpws =  lbpService.getLBPGatewayWSPort();
        char password                   =  "keystorepwd".toCharArray();
        String alias                    =  "keystorealias";
        String ksFileDir                =  "C:\\lbpkeystore.jks";
        String ksFileAccess             =  "true&jks&keystorepwd&keystorealias&SUN";
        FileInputStream fIn;
        PublicKey publicKey             =  null;
        Certificate cert ;

        //Open Keystore
        fIn = new FileInputStream(ksFileDir);
        KeyStore keystore = KeyStore.getInstance("JKS");
        keystore.load(fIn, password);
        cert = keystore.getCertificate(alias);
        Key key=keystore.getKey(alias,password);
        cert=keystore.getCertificate(alias);
        publicKey=cert.getPublicKey();

        arg0 = req.getParameter("MessageType");
        arg1 = req.getParameter("TransactionType");
        arg2 = req.getParameter("PaymentMode");
        arg3 = req.getParameter("CurrencyCode");
        arg4 = req.getParameter("UserName");
        arg5 = req.getParameter("Password");
        arg6 = req.getParameter("ProductID");
        arg7 = req.getParameter("ReferenceNumber");
        arg8 = req.getParameter("TransactionNumber");
        arg9 = req.getParameter("MessageDate");
        arg10 = req.getParameter("MessageTime");
        arg11 = req.getParameter("PaymentDate");
        arg12 = req.getParameter("PaymentAmount");
        arg13 = req.getParameter("PaymentDetail");
        arg14 = req.getParameter("PayeeCode");
        arg15 = req.getParameter("PayeeName");
        arg16 = req.getParameter("PayerCode");
        arg17 = req.getParameter("PayerName");
        arg18 = req.getParameter("CreditAccountNumber");
        arg19 = req.getParameter("DebitAccountNumber");
        arg20 = req.getParameter("ToBankCode");
        arg21 = req.getParameter("FromBankCode");
        arg22 = req.getParameter("OtherDetail1");
        arg23 = req.getParameter("OtherDetail2");
        arg24 = req.getParameter("OtherDetail3");
        arg25 = req.getParameter("OtherDetail4");
        arg26 = req.getParameter("OtherDetail5");

        // paymentmode + username + password + messagedate + messagetime
        // + referenceno + transactionnumber + amount + merchantcode
        String params = arg2 + arg4 + arg5 + arg9 + arg10 + arg7 + arg8 + arg12 + arg14;

        //CREATE MAC
        String mac;
        mac = getMac(params,publicKey);

        //BASE64ENCODE
        String encoded = new String(encode(mac.getBytes()));

        //ENCRYPT
        arg27 = encrypt(encoded);

        String s = albpws.processPayment(arg0, arg1, arg2, arg3, arg4, arg5, arg6, arg7, arg8, arg9, arg10, arg11, arg12, arg13, arg14, arg15,
        arg16, arg17, arg18, arg19, arg20, arg21, arg22, arg23, arg24, arg25, arg26, arg27);

        resp.sendRedirect(s);  }
```

SAMPLE CODE (REST):

```
public void processRequest(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {

        LBPGatewayWSService lbpService = new LBPGatewayWSService();
        ALBPCaptureWS albpws =  lbpService.getLBPGatewayWSPort();
        char password                   =  "keystorepwd".toCharArray();
        String alias                    =  "keystorealias";
        String ksFileDir                =  "C:\\lbpkeystore.jks";
        String ksFileAccess             =  "true&jks&keystorepwd&keystorealias&SUN";
        FileInputStream fIn;
        PublicKey publicKey             =  null;
        Certificate cert ;

        //Open Keystore
        fIn = new FileInputStream(ksFileDir);
        KeyStore keystore = KeyStore.getInstance("JKS");
        keystore.load(fIn, password);
        cert = keystore.getCertificate(alias);
        Key key=keystore.getKey(alias,password);
        cert=keystore.getCertificate(alias);
        publicKey=cert.getPublicKey();

        arg0 = req.getParameter("MessageType");
        arg1 = req.getParameter("TransactionType");
        arg2 = req.getParameter("PaymentMode");
        arg3 = req.getParameter("CurrencyCode");
        arg4 = req.getParameter("UserName");
        arg5 = req.getParameter("Password");
        arg6 = req.getParameter("ProductID");
        arg7 = req.getParameter("ReferenceNumber");
        arg8 = req.getParameter("TransactionNumber");
        arg9 = req.getParameter("MessageDate");
        arg10 = req.getParameter("MessageTime");
        arg11 = req.getParameter("PaymentDate");
        arg12 = req.getParameter("PaymentAmount");
        arg13 = req.getParameter("PaymentDetail");
        arg14 = req.getParameter("PayeeCode");
        arg15 = req.getParameter("PayeeName");
        arg16 = req.getParameter("PayerCode");
        arg17 = req.getParameter("PayerName");
        arg18 = req.getParameter("CreditAccountNumber");
        arg19 = req.getParameter("DebitAccountNumber");
        arg20 = req.getParameter("ToBankCode");
        arg21 = req.getParameter("FromBankCode");
        arg22 = req.getParameter("OtherDetail1");
        arg23 = req.getParameter("OtherDetail2");
        arg24 = req.getParameter("OtherDetail3");
        arg25 = req.getParameter("OtherDetail4");
        arg26 = req.getParameter("OtherDetail5");

        // paymentmode + username + password + messagedate + messagetime
        // + referenceno + transactionnumber + amount + merchantcode
        String params = arg2 + arg4 + arg5 + arg9 + arg10 + arg7 + arg8 + arg12 + arg14;

        //CREATE MAC
        String mac;
        mac = getMac(params,publicKey);

        //BASE64ENCODE
        String encoded = new String(encode(mac.getBytes()));

        //ENCRYPT
        arg27 = encrypt(encoded);

        URL url = new URL("http://x.x.x.x/egps/rs/epayment/processpayment");

        HttpURLConnection conn = (HttpURLConnection) url.openConnection();
        conn.setDoOutput(true);
```

```
conn.setDoInput(true);
conn.setInstanceFollowRedirects(false);
conn.setRequestMethod("POST");
conn.setRequestProperty("Content-Type", "application/x-www-form-urlencoded");
conn.setRequestProperty("charset", "utf-8");

String urlParameters = "MessageType=" + arg0 + "&" +
                       "TransactionType=" + arg1 + "&" +
                       "PaymentMode=" + arg2 + "&" +
                       "CurrencyCode=" + arg3 + "&" +
                       "Username=" + arg4 + "&" +
                       "Password=" + arg5 + "&" +
                       "ProductID=" + arg6 + "&" +
                       "ReferenceNum=" + arg7 + "&" +
                       "TransactionNum=" + arg8 + "&" +
                       "MessageDate=" + arg9 + "&" +
                       "MessageTime=" + arg10 + "&" +
                       "PaymentDate=" + arg11 + "&" +
                       "PaymentAmt=" + arg12 + "&" +
                       "PaymentDetail=" + arg13 + "&" +
                       "PayeeCode=" + arg14 + "&" +
                       "PayeeName=" + arg15 + "&" +
                       "PayerCode=" + arg16 + "&" +
                       "PayerName=" + arg17 + "&" +
                       "CrAcctNum=" + arg18 + "&" +
                       "DrAcctNum=" + arg19 + "&" +
                       "BankCodeTo=" + arg20 + "&" +
                       "BankCodeFrom=" + arg21 + "&" +
                       "OtherDetails1=" + arg22 + "&" +
                       "OtherDetails2=" + arg23 + "&" +
                       "OtherDetails3=" + arg24 + "&" +
                       "OtherDetails4=" + arg25 + "&" +
                       "OtherDetails5=" + arg26 + "&" +
                       "Signature=" + arg27;

OutputStreamWriter writer = new OutputStreamWriter(conn.getOutputStream());
writer.write(urlParameters);
writer.flush();
String line;
BufferedReader reader = new BufferedReader(new InputStreamReader(conn.getInputStream()));
while ((line = reader.readLine()) != null) {
    s = line;
}
writer.close();
reader.close();
conn.disconnect();

resp.sendRedirect(s);  }
```

5.0 WebService Endpoints and WSDL

SOAP

UAT Site	http://58.71.22.7/testegps/LBPGatewayWS?WSDL
	http://58.71.22.7/testegps/ADALBPCaptureWS?WSDL
Production	https://www.lbp-eservices.com/egps/LBPGatewayWS?WSDL
	https://www.lbp-eservices.com/egps/ADALBPCaptureWS?WSDL

ADA and Inquiry web service methods will return a String containing the return Error Code for the requested Reference Number upon proper authentication of signature and validation of the webservice parameter values.

For non-ADA and non-Inquiry processes, upon proper authentication of signature and validation of the webservice parameter values, the LBP WebService will return a URL to the WSClient. The Agency shall redirect the user to this URL which will be returned by the LBP WebService.

REST

UAT Site	http://58.71.22.7/testegps/rs/epayment/processada?
	http://58.71.22.7/testegps/rs/epayment/processpayment?
	http://58.71.22.7/testegps/rs/epayment/inquiry?
Production	https://www.lbp-eservices.com/egps/rs/epayment/processada?
	https://www.lbp-eservices.com/egps/rs/epayment/processpayment?
	https://www.lbp-eservices.com/egps/rs/epayment/inquiry?

6.0 Message Types

MESSAGE TYPE	DESCRIPTION
01	API
02	REST
03	SOAP

7.0 Payment Modes

PAYMENT MODE	DESCRIPTION
01	Data Entry System Payment
02	ATM Payment
03	Auto Debit Arrangement (ADA) Payment
04	Payer to Payor (P2P) Payment
05	Payment Warehousing (Maker)
06	Payment Warehousing (Authorizer)
98	Payment Inquiry
99	Bancnet Payment

8.0 Error Codes

ERRORCODE	DESCRIPTION
0000	Transaction Completed Successfully
0205	Do not honor / error Transaction Denied, Contact your Bank
0208	Invalid IP Address
0209	Balance File Error
0212	Invalid Transaction
0213	Invalid Amount
0214	Invalid Card Number
0217	Customer Cancel
0219	System Error, Reenter transaction
0220	Invalid Response
0221	No Action Taken
0223	Unacceptable Transaction Fee
0225	Unable to locate record on file/ record for reversal not found (normal transaction, reversal was done during SSF cannot reverse transaction)
0226	Duplicate File
0230	Invalid Message Format
0231	Bank not supported by switch
0233	Expired Card Not Allowed
0236	Card is restricted
0238	Allowable PIN retries exceeded
0239	No Credit Account
0241	Lost Card
0243	Card was reported Stolen
0245	Card not valid for Credit function
0246	Invalid Discretionary Data
0248	No check activity
0249	Closing - Only Debits allowed
0250	Only Credits Allowed
0251	Denied for Insufficient Funds
0252	No Checking Account
0253	No Savings Account
0254	Expired Card
0255	Invalid Pin
0256	No Card Record
0257	Transaction Not Permitted To Issuer
0261	Exceeds withdrawal limit
0262	Restricted Card
0263	Security Violation
0264	Original Amount Incorrect
0265	Transaction Exceeds Number Limit
0267	Hot Card/Retain Card
0268	Late Response/Time Out
0269	Inactive Account
0270	Dormant Account
0271	Closed Account
0272	Terminated Account

ERRORCODE	DESCRIPTION
0273	Purged Account
0274	Frozen Account
0275	Allowable Number of PIN tries exceeded
0276	Acquirer not Available
0277	PIN Key Sync Error
0278	MAC Key Sync Error
0279	Invalid Business Date
0280	System Not Available
0281	Unable to Process at this Time
0282	Switch Transaction not available
0283	Error Accessing CMSXRF file
0284	Error accessing Merchant file
0285	Unable to reverse transaction.
0291	Destination processor not available
0294	Duplicate Transmission / Double reversal (do reposting)
0296	System Error
0297	Memo I/O error. Transaction found, reposting not possible. (reposting transaction)
0299	Could Not Process Transaction
0500	General Error Encountered
0501	Invalid Payment Message Instruction
0502	Invalid Message Type
0503	Invalid Transaction Type
0504	Invalid Payment Mode
0505	Invalid Reference Number
0506	Invalid Transaction Number
0507	Invalid Message Date
0508	Invalid Message Time
0509	Invalid Payment Date
0510	Invalid Payment Amount
0511	Invalid Payment Detail
0512	Invalid Payee Code
0513	Invalid Payee Name
0514	Invalid Payer Code
0515	Invalid Payer Name
0516	Inva22lid Control Number
0517	Invalid AccountTo Number
0518	Invalid AccountFrom Number
0527	Error parsing payment instruction message
0528	Transaction fee breakdown do not match
0803	Transaction Cancelled By User
0806	Transaction Timed Out
0807	Invalid Account Number
0808	Invalid Joint Account Indicator
0809	Invalid Payer User
0900	Invalid IP / IP not registered
0901	System currently not Available

ERRORCODE	DESCRIPTION
0902	System is under maintenance
0903	Transaction with Merchant Currently Disabled
0904	Security Module Error Occurred
0905	Error in decryption message
0906	Encryption Error
0907	Payment instruction message control number does not match
0999	General Exception Encountered
0809	Invalid Payer User
0810	Debit Account not enrolled in bank
0811	Payer/User not enrolled in bank
0812	Allowable no. of Account No. tries exceeded

9.0 Transaction Status

STATUS	DESCRIPTION
A	Accepted Transaction
R	Rejected Transaction