Ukrainian Processing Center

Payment gateway “e- Commerce Connect Gateway”

Communication Interface

Guidance of the e-shop administrator

2011

Table of Contents

1. General positions

In the phase of the card purchasing capacity check the interaction of the e-shop with the payment gateway implements at the completion phase of the so-called “checkout”-process. For this phase, as a rule, it’s typical that the customer has already identified the list of purchases and services, their costs, delivery terms etc. and agreed to make a payment by means of a credit card. At this moment the task of the e-shop is to redirect the customer to the secure page of the payment server as well as to transmit all the necessary transaction data in the redirection line.

After the redirection to the gateway secure page interaction with the customer is implemented through the secure https protocol. For this purpose the payment gateway provided with the SSL-certificate granted by the certified agency (for example, the “VeriSign” agency). However for the authentication of the shop and data secure from the modification when in use of redirection all critical data protected by means of MAC (Message Authentication Code).

For the interaction with the gateway e-shop software ought to have such pages as:

1. Page with the prepared values for the request transfer to the payment gateway.
2. Page (**SUCCESS_URL**) for the user’s browser redirection in the case of successful transaction. The response parameters transmit the processing results.
3. Page (**FAILURE_URL**) for the user’s browser redirection in the case of unsuccessful transaction. The response parameters transmit the processing results.
4. Page (**NOTIFY_URL**) for the transaction result transfer from the gateway directly to the e-shop (optionally).

If page 4 is not in use, all the processing results are transferred through the browser page to the e-shop address (pages 2, 3). The deployment of this page makes it possible to transfer the transaction results directly to the merchant from the gateway. Thereby it allows to raise the security level – the merchant relies on the connection from the gateway side, as the address of such source is fixed, as opposed to the customer browser. In addition, after such an approval, by customer redirection (p. 2, 3), the response parameters transfer only the uncritical part of the processing results, thus ensure the concealment of the most critical data from the customer.

Dynamic elements are used for the URL formation in some software. As usual this happens when the server software or the browser do not support or switch off the cookies support system. In this case a merchant should grant the URL formation scheme.

2. Transfer of the authorization request parameters to the payment server

E-shop has to transfer a number of parameters when passing to a secure page of the gateway. Such parameters indicated at the table 1:

Table 1

Parameter	Structure	Format	Description of the parameter	Additional comments
Version	F	n4	Version of the interface SG	Version of the interface protocol. Current version 0001. This is a help parameter for the handler of the gateway incoming data. Is used to choose the better way for data processing.
MerchantID	L	an15	Merchant identifier	Assigned by processing bank.
TerminalID	F	an8	Terminal identifier	-- // --
TotalAmount	F	n12	Purchase amount	In the smallest currency units (kopecks, cents)
Currency	F	n3	Currency	Under the agreement with the processing bank.
PurchaseTime	F	n12	Time of the purchase in MMddhhmmss format	
Locale	F	a2	Language of the interface (en, ru, uk)	Language of the interface of the secured gateway page.
OrderID	L	ans…20	Number of the order up to 20 byte length	The value of the XID is determined on the basis of the OrderID. If OrderID can not be used, one should use XID parameter.
XID	F	ans28	Transaction identifier (number of the order augmented up to 20 byte)	The parameter stipulated by the 3-D Secure specification. Can be defined on the basis of the internal order numbering of the shop or ad arbitrium. The base requirement – uniqueness over a considerable period of time (minimum 6 months). Transferred to the Base64 encrypted and has 28 symbols.
SD (O)	Var	an...99	Session Data – session's data	Auxiliary parameter which can be used by e-shop to administer users' sessions.
PurchaseDesc (O)	L	ans...125	Brief description of the purchase	Optional parameter stipulated by the 3-D Secure specification.
Signature	Var	an….40	MAC-code value	The length of the parameter depends on the chosen scheme of MAC-code calculation.

Annotation:

A. *Structure description*

F – full field

L – left justified

R – right justified

S – filled with spaces

Z – filled with zeroes

Var –variable length field

B. *Format description*

n- numeric decimal digit, value 0..9,

an - alphabetic or numeric character, value 0..9 or A..Z or ..z,

ans - alphabetic, numeric or special character,

These parameters are transferred to a gateway page, in an appointed HTML-format using HTTPS/POST method, for a further input of the payment card details by the customer (cardholder).

Example:

```
<FORM  ACTION="https:/ecs.upc.ua:8443/enter.do" METHOD="POST">
<INPUT TYPE="HIDDEN" NAME="Version" VALUE="0001">
<INPUT TYPE="HIDDEN" NAME="MerchantID" VALUE="6352045">
<INPUT TYPE="HIDDEN" NAME="TerminalID" VALUE="ECI62791">
```

```
<INPUT TYPE="HIDDEN" NAME="TotalAmount" VALUE="000000012550">
<INPUT TYPE="HIDDEN" NAME="Currency" VALUE="980">
<INPUT TYPE="HIDDEN" NAME="locale" VALUE="ru">
<INPUT TYPE="HIDDEN" NAME="SD" VALUE="584sds565hgj76GGjh6756248">
<INPUT TYPE="HIDDEN" NAME="OrderID" VALUE="HV-923452">
<INPUT TYPE="HIDDEN" NAME="PurchaseTime" VALUE="031227105500">
<INPUT TYPE="HIDDEN" NAME="PurchaseDesc" VALUE="Musical Disc ">
<INPUT TYPE="HIDDEN" NAME="Signature" VALUE="45F345FAFDE4455445AC">

</FORM>
```

Then at the gateway page the received data supplemented with the Card Number, ExpYear, ExpMonth, CVV2, and Card Type. Previously the gateway performs sequence of verifications (the existence of registration parameters of the merchant in the database, the correspondence of the currency to the registered value, the authorization limit of the merchant, verification of the electronic signature).

After that the gateway provides the customer browser with the page for payment card details input. At the same time the buyer can indicate the card type (on conditions that the merchant is able to accept one or another card type). Also the customer can input the CVV2 code (for cards MEASTRO this function is not submitted).

At the next stage the request processing is carried out using the 3D-Secure or a standard scheme (channel encryption e-commerce), on the basis of the parameters provided by the bank that provides services.

3. Back-off of the authorization request processing results to the e-shop

Processing results (transaction results) can be transferred in two ways:

- forwarding of the results to NOTIFY_URL address and redirection of the customer browser to the page “successful/ unsuccessful”

- forwarding of the results through the customer browser to the page “successful/ unsuccessful”

In the first case the processing results are transferred from the gateway to the e-shop page using HTTP/HTTPS POST method. Under such conditions the additional security can be achieved by e-shop with the limitation of the access to the particular URL to the gateway requests only.

The gateway at the session might receive a confirmation on the fact of shop notification concerning the state and parameters of the fulfilled transaction. One of the advantages is that no parameters of the reverse transaction will be at the customer browser page

A list of response parameters to the e-shop website:

Table 2

Parameter	Format	Description of the parameter	Additional comments
MerchantID	an15	Merchant identifier	Is similar to the data in the authorization request
TerminalID	an8	Terminal identifier	--- // ---
TotalAmount	n12	Purchase amount	--- // ---
Currency	n3	Currency	--- // ---
PurchaseTime	n12	Time of the purchase request (YYMMDDhhmmss)	--- // ---
OrderID	ans..20	Order ID	
XID	ans28	Transaction identifier (number of the order augmented up to 20 byte)	--- // ---
SD	an... 99	Session Data	--- // ---
ApprovalCode	n6	Host authorization code	
Rrn	n10	Retrieval Reference Number	Unique transaction number in the authorization and settlement system of the servicing bank
ProxyPan	N13…19	Lust 4 digits of the card number	PAN value (four lust digits) with the additional zeroes in front for the PAN length
TranCode	n3	Code of the transaction completion	See table 3
Signature	an…40	MAC-code value for the chosen scheme of the gateway/e-shop intercommunication	Parameter length depends on the chosen scheme of the MAC-code calculation

After the given session of the gateway with the shop host the concluding forwarding of the browser takes place. It looks like "approved"/"rejected". The minimum of parameters is transferred. Such as: OrderID , TranCode and SD.

Example:

```
<FORM NAME=”back” ACTION=”http://www.playboy.kiev.ua/shop/success.asp” METHOD="POST">

<INPUT TYPE=”HIDDEN” NAME=”Version” VALUE="0001">
<INPUT TYPE="HIDDEN" NAME="SD" VALUE="584sds565hgj76GGjh6756248">
<INPUT TYPE="HIDDEN" NAME="OrderID" VALUE="VHS-23684">
<INPUT TYPE="HIDDEN" NAME="TranCode" VALUE="000">

</form>


<noscript>
<center>
<h1>Return processing results</h1>
<h2>Your browser noes not support JavaScript or disabled</h2>
<h3>Click 'Submit' to continue with Transaction</h3>
<input type="submit">
</center>
</noscript>
```

Addresses of the e-shop Web Pages retrieved by the gateway from its Data Base, i.e. they have to be provided by the merchant beforehand – at the registration stage.

In the second case the processing results are transferred through the browser page, where the corresponded form is transmitted to the merchant website address to the page “successful/unsuccessful”. The operation of the form starting carries out by Java Script. If the implementation of this language found impossible the message about the necessity of manually confirmation of form sending is to be input.

Example:

```
<FORM  NAME=”back” ACTION=”http://www.playboy.kiev.ua/shop/success.asp”  METHOD="POST">

<INPUT TYPE=”HIDDEN” NAME=”Version” VALUE="0001">
<INPUT TYPE="HIDDEN" NAME="MerchantID" VALUE="6352045">
<INPUT TYPE="HIDDEN" NAME="TerminalID" VALUE="ECI62791">
<INPUT TYPE="HIDDEN" NAME="TotalAmount" VALUE="12550">
<INPUT TYPE="HIDDEN" NAME="Currency" VALUE="980">
<INPUT TYPE="HIDDEN" NAME="SD" VALUE="584sds565hgj76GGjh6756248">
<INPUT TYPE="HIDDEN" NAME="OrderID" VALUE="VHS-23684">
<INPUT TYPE="HIDDEN" NAME="ApprovalCode" VALUE="554632">
<INPUT TYPE="HIDDEN" NAME="Rrn" VALUE="7753335670">
<INPUT TYPE="HIDDEN" NAME="ProxyPan" VALUE="0000000000005207">
<INPUT TYPE="HIDDEN" NAME="TranCode" VALUE="000">
<INPUT TYPE="HIDDEN" NAME="Signature" VALUE="45F345Fafde4455445Gvb550">

</form>
```

```
<noscript>
<center>
<h1>Return processing results</h1>
<h2>Your browser noes not support JavaScript or disabled</h2>
<h3>Click 'Submit' to continue with  Transaction</h3>
<input type="submit">
</center>
</noscript>

</form>

<script language="javascript">
<!--
  document.back.submit();
-->
</script>
```

For binding of the customer to the corresponded e-shop session and purchase the SD (Session Data) parameter is used, which is transferred through the customer browser in the process of backward redirection.

Example:

```
<INPUT TYPE="HIDDEN" NAME="SD" VALUE="584sds565hgj76GGjh6756248">
```

4. Transaction response codes

Transaction response codes are divided into several classes and subclasses and serve for an informing of the merchant about the transaction results. To indicate a successful transaction one response code is required. The major part of the response codes serves for the provision of the generalized information about the reasons of unsuccessful transaction to the merchant.

Table 3

Codes on basis of the authorization host responses		Comments
Integrated response codes for the e-shops	Interpretation of the codes	Response codes in the message 1110
000	Successful authorization	00x
105	Do not honor by the issuing bank	100, 103,104,105…107,
116	Insufficient funds	116
111	Non-existent card	111,125,200,202
108	Lost or stolen card	208,209
101	Invalid expiration date	101,201
130	Amount limit exceeded	121,123
290	Issuer is inaccessible	905…908,910
291	Technical or communicational problem	9xx (except indicated above)
Codes on basis of the responses generated by the payment server without referencing to the bank host		
Internal errors codes of the payment server in accordance with the processing method		
401	Format error	
402	Acquirer/Merchant parameters error	
403	Connection error to the payment system resource (DS)	
404	Customer authentication error	
405	Signature error	
501	Transaction canceled by the user	
502	Browser session is out of date	

5. Application of the hardware token for the MAC-code generation

After the reception of the request parameters from the merchant the payment gateway verifies data for the purpose of its integrity by means of verification of the merchant signature (MAC-code). The merchant has to generate MAC-code value and send it as the “Signature” parameter.

To generate a signature the cryptographic function of the token is used. Token should be installed by merchant together with the driver. Driver of the token accepts the request in the appropriate format and generates the results with the completion code and signature value, if the completion code is successful.

Driver of the token accepts the requests by TCP/IP protocol and generates the result.

The signature request format and response format correspond the following:

Request		
Data	Format	Description
‘A0’	an2	Token command
‘|’	an1	Separating character ‘|’
‘Merchant ID’	an…15	Merchant identifier, given during registration
‘|’	an1	Separating character ‘|’
Request parameters	an…99	Request parameters connected by ‘;’ symbol, in a following sequence: • Merchant ID • Terminal ID • Purchase Time • Order ID • XID • Currency • Total Amount • SD The parameter value is in use even if it’s empty
Response		
‘A1’	an2	Token response command
‘|’	an1	Separating character ‘|’
Completion code	N2	Completion codes: • 00 - successful, the signature is formed • 01 - error, the request is formed incorrectly • 02 - error, no object at the token that corresponds to the merchant • 03 - error of the implementation of the cryptographic function
‘|’	an1	Separating character ‘|’
Signature	An…32	Signature value, sixteen-digit value

Example:

Merchant ID = 6352045

- Request:

A0|6352045|6352045;ECI62791;031227105500;HV-923452;;980;12550;584sds565hgj76GGjh6756248

- Response:

A1|00|A878B97869D96989E986C8980980

6. Application of the token for the payment gateway response verification

After the completion of the transaction processing the payment server forms the transaction result and its verification MAC-code.

The integrity of the data received from the payment server can be checked thought the instrumentality of A2 command.

Request		
Data	Format	Description
‘A2’	an2	Token command
‘|’	an1	Separating character ‘|’
‘Merchant ID’	an…15	Merchant identifier, given during registration
‘|’	an1	Separating character ‘|’
Request parameters	an…99	Request parameters connected by ‘;’ symbol, in a following sequence: • Merchant ID • Terminal ID • Purchase Time • Order ID • XID • Currency • Total Amount • SD • TranCode - transaction code • ApprovalCode – authorization code ProxyPan - card number where all the digits except last 4 substituted for 0 The parameter value is in use even if it’s empty
‘|’	an1	Separating character ‘|’
Signature	An…32	Signature value, sixteen-digit value
Response		
‘A3’	an2	Token response command
‘|’	an1	Separating character ‘|’
Completion code	N2	Completion codes: • 00 - successful, the signature is formed • 01 - error, the request is formed incorrectly • 02 - error, no object at the token that corresponds to the merchant • 03 - error of the implementation of the cryptographic function

Example:

Merchant ID = 6352045

- Request

```
A2|6352045|6352045;ECI62791;031227105500;HV-923452;;980;12550;584sds565hgj76GGjh6756248;000;523453;0000000000005012
```

- Response

```
A3|00
```

7. Examples of the programmes

Example in Perl :

```
use Net::Telnet ();
use POSIX qw(strftime);


sub getPurchaseTime() {
  return strftime "%y%m%d%H%M%S", localtime ;
}

my $hostname     = "127.0.0.1" ;
my $hostport        = 27015 ;


$PurchaseTime = getPurchaseTime() ;

my $data =
"$MerchantID;$TerminalID;$PurchaseTime;$OrderID;$XID;$Currency;
$TotalAmount;$SD;" ;
print 'Data to sign : ' . $data . "\n" ;

my $hsm = new Net::Telnet (Telnetmode => 0);
$hsm->open(Host => $hostname,
                      Port => $hostport );

$hsm->put("A0|$MerchantID|$data");

$Response = $hsm->get() ;
$hsm->close() ;

my @hsmResult = split( /\|/, $Response ) ;

$Signature = $hsmResult[2] ;

print  "CCODE = " . $hsmResult[1] ."\n" ;
print  "Sign = "       . $Signature ."\n" ;
```

Example in PHP :

```
<?php

  $PurchaseTime = strftime ("%y%m%d%H%M%S") ;


     $data =
"$MerchantID;$TerminalID;$PurchaseTime;$OrderID;$XID;$Currency;
$TotalAmount;$SD;" ;

     echo "<br>Data to sign : $data" ;

     $address = "172.29.112.18" ;
     $service_port = 27015 ;

     /* Create a TCP/IP socket. */
     $socket = socket_create(AF_INET, SOCK_STREAM, SOL_TCP);
     if ($socket < 0) {
echo "socket_create() failed: reason: " . socket_strerror($socket) .
"\n";
     }

     echo "Attempting to connect to '$address' on port
'$service_port'...";
     $result = socket_connect($socket, $address, $service_port);
     if ($result < 0) {
echo "socket_connect() failed.\nReason: ($result) " .
socket_strerror($result) . "\n";
     } else {

     }

     $in = "A0|$MerchantID|$data" ;
     socket_write($socket, $in, strlen($in));

     $out = "" ;
     $recv = socket_recv($socket, $out, 2048, 0) ;

     list($cmd, $ccode, $signValue) = split('\|', $out);
     $Signature = $signValue ;

     socket_close($socket);

?>
```