

Version

1.16



NestPay®

Merchant Integration 3D Pay

www.asseco-see.com.tr

Document Information

Project/Product Name	
Project Manager	
Document Version No	

Document Code			
Prepared By		Preparation Date	05/11/2015
Reviewed By		Review Date	05/11/2015

Distribution List

From	Date	Phone/Fax

To	Action*	Due Date	Phone/Fax

* Action Types: Approve, Review, Inform, File, Action Required, Attend Meeting, Other (please specify)

Version History

Ver. No.	Ver. Date	Revised By	Description
1.4	11/5/2015	Okan GÜRBÜZ	Document format has been changed
1.5	1/2/2013	Nildem DEMİR	English parameters
1.6	1/29/2013	Nildem DEMİR	Recurring payment
1.7	4/12/2013	Selcuk Yılmaz	Explanation has been added about instalment and rnd.
1.8	15/7/2013	Nildem Demir	Address details
1.9	4/10/2013	Nihal Müstecaplıoğlu	"comments" parameter added.
1.10	5/16/2014	Erdem BEĞENİLMİŞ	Hash Ver 2 & Sample Codes
1.11	20/06/2014	Nihal Müstecaplıoğlu	Hash ver2 support
1.12	10/13/2014	Selcuk Yılmaz	Ver2 has been added in mandatory parameter
1.13	03/11/2014	Yiğit İPÇİOĞLU	ProcReturnCode
1.14	11/04/2014	Erdem BEĞENİLMİŞ	Card Brand
1.15	04/22/2015	Erdem BEĞENİLMİŞ	Hash Ver2 - Deleting Non-ISPC Paramaters
1.16	05/25/2015	Erdem BEĞENİLMİŞ	Removing Turkish Characters

Proprietary Notice

The information contained in all sheets of this document, proposal, or quotation constitutes trade secrets and/or information that is commercial or financial and is deemed confidential or privileged. It is furnished to prospective customer in confidence with the understanding that prospective customer will not, without the permission of Asseco-SEE Teknoloji A.Ş. (from now on called Asseco-SEE or Asseco), use or disclose for other than evaluation purposes the information contained herein which is solely confidential and proprietary to Asseco-SEE ("**Asseco-See Confidential Information**"). In the event a contract is awarded on the basis of this document, proposal, or quotation, prospective customer shall have the right to use and disclose Asseco-See Confidential Information to the extent provided in the contract. The restriction does not limit prospective customer right to use or disclose such information if obtained from another source without restriction.

Contents

1.	3D Pay Model	6
2.	Nestpay 3D Pay Model	7
3.	Quick Start Guide	7
3.1	Generate Hash for Client Authentication	7
3.1.1	Hash Version 2 – SHA 512 Algorithm	7
3.2	Posting Hidden Parameters.....	9
3.2.1	Sample HTTP form with mandatory parameter set	9
3.3	3D Authentication.....	10
3.4	Transaction Result	10
3.5	Merchant Success Page	10
3.5.1	Basic transaction response parameters for fully authenticated successful 3D transaction ...	10
4.	Integration Basics.....	11
4.1	HTTP Post Integration	11
4.1.1	Sample HTTP form with mandatory and optional parameters	11
4.2	Card Transactions	13
4.2.1	MPI Response Parameters	13
4.2.2	Possible mdStatus Values	13
4.2.3	Successful Transaction.....	13
4.2.4	Failed Transaction.....	13
4.2.5	Transaction Response Parameters.....	14
4.2.6	MPI Response Parameters	14
4.2.7	Possible Transaction Results.....	14
4.3	Hash Checking.....	15
4.3.1	Generating the plain text for Hash Ver-2	15
4.3.2	Security Notes	16
5.	Code Samples	17
5.1	ASP Code Sample	17

5.1.1	Request Sample Codes	17
5.1.2	Response Code Sample	17
5.2	.Net Code Sample	20
5.2.1	Request Sample Codes	20
5.2.2	Response Code Sample	27
5.3	JSP Code Sample	35
5.3.1	Request Sample Codes	35
5.3.2	Response Code Sample	38
5.4	PHP Code Sample	41
5.4.1	Request Sample Codes	41
5.4.2	Response Code Sample	44
6.	APPENDIX A: Gateway Parameters	48
6.1	Mandatory Input Parameters	48
6.2	Optional Input Parameters	49
6.3	Transaction Response Parameters.....	51
6.4	MPI Response Parameters	52

1.3D Pay Model

3D PAY model is the basic internet integration model supporting 3D transactions.

Basic Properties:

- Enables processing of 3D secure card transactions
- HTTP Post method is supported for merchant integration.
- Credit card page is hosted by the merchant.
- Payment is done automatically by NestPay.

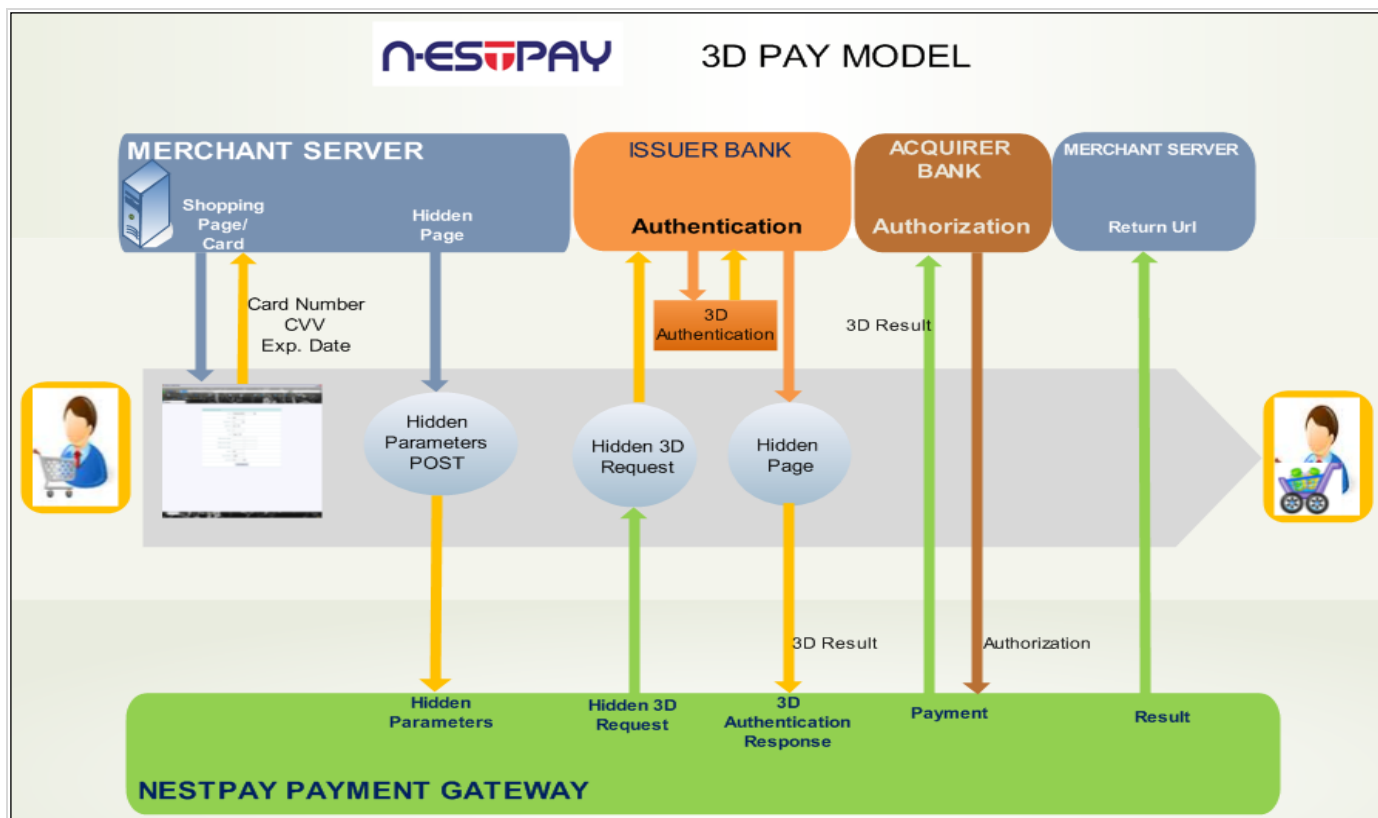
After obtaining all necessary shopping data from the customer like card number and expiry, order amount, currency, customer name/surname etc., merchant server generates a unique order ID. Necessary parameters are posted with HTTP Post method to NestPay gateway.

For card payment methods like Visa and MasterCard, merchant server needs to submit the card details like card number, CVV2, and expiry date information. After the order and card data is obtained from the user, 3D flow (enrolment and authentication queries) starts. In 3D flow, the 3D authentication information of the customer is queried by the issuer bank. The methods for 3D authentication can be different for different issuers. Examples of 3D authentication methods include using 3D secure passwords, one-time passwords, or security questions.

Using this model,

1. Integration process is easy.
2. Bank's SSL certificate is used. Therefore the software cannot be updated.
3. In addition to the obligatory parameters, merchant can POST its own data, such as username, user email or user id. This data is sent back to the merchant by the bank.

2. Nestpay 3D Pay Model



3D Pay Model Diagram

3. Quick Start Guide

This section will describe how to perform a successful Sale VISA transaction with **3D Pay Model**.

3.1 Generate Hash for Client Authentication

3.1.1 Hash Version 2 – SHA 512 Algorithm

Hash for Version 2 is the base64-encoded version of the hashed text which is generated with SHA512 algorithm. For using Hash Version 2, "**hashAlgorithm**" parameter should be sent in the request with the value of "**ver2**".

To generate the hash for client authentication, the following parameter values should be appended in the order given below by using pipeline "|" as a separator. If a parameter is to be skipped, a double pipeline "||" can be used instead.

```
plaintext = clientid + | + oid + | + amount + | + okurl + | + failurl + | + transaction type + | +
instalment + | + rnd + ||| + currency + | + storeKey
```

Note: In case "|" character needs to be used in a parameter, "\" character can be used for escaping. Additionally, if the "\" character is also used in a parameter, we use another "\" character before it and then append it to hash plain text. An example is shown below:

```
Original Text : ORDER-256712jbs\j6b|
Escaped Text : ORDER-256712jbs\\j6b\\
```

- **Given parameters**

```
Clieid      : 9900000000000001
oid         : ORDER256712jbs\j6b|
amount      : 91.96
okurl       : https://www.teststore.com/success.php
failurl     : https://www.teststore.com/fail.php
transaction type : Auth
instalment  : 2
rnd         : asdf
currency    : 949
storekey    : AB123456\\
```

- **Hash**

```
Plaintext=
9900000000000001|ORDER256712jbs\\j6b\\|91.96|https://www.teststore.com/success.php|https://w
ww.teststore.com/fail.php|Auth|2|asdf|||949|AB123456\\|

Hash = Base64(SHA512(plaintext))
```


3.2 Posting Hidden Parameters

Mandatory input parameters are posted to NestPay Payment Gateway located at **https://host/fim/est3dgate** as hidden parameters.

clientid	: Merchant ID (given by Nestpay)
storetype	: "3D_PAY"
hash	: Hash value for client authentication
trantype	: "Auth"
amount	: amount transaction amount
currency	: ISO code of transaction currency (949 for TL)
oid	: Unique identifier of the order
okUrl	: Return URL to which NestPay Payment Gateway redirects the browser of the customer if transaction is completed successfully.
failUrl	: Return URL to which Nestpay Payment Gateway redirects the browser of the customer if transaction is completed unsuccessfully.
lang	: Language of the payment pages hosted by Nestpay ("tr" for Turkish, "en" for English)
pan	: Card number
Ecom_Payment_Card_ExpDate_Year	: Expiry year
Ecom_Payment_Card_ExpDate_Month	: Expiry month
cv2	: Cv2 value
encoding	: Page encoding
hashAlgorithm	: ver1 or ver2

3.2.1 Sample HTTP form with mandatory parameter set

```
<form method="post" action="https://host/fim/est3dgate">
  <input type="hidden" name="clientid" value="9900000000000001"/>
  <input type="hidden" name="storetype" value="3D_PAY" />
  <input type="hidden" name="hash" value="iej6cPOjDd4IKqXWQEznXWqLzLI=" />
  <input type="hidden" name="trantype" value="Auth" />
  <input type="hidden" name="amount" value="91.96" />
  <input type="hidden" name="currency" value="949" />
  <input type="hidden" name="oid" value="1291899411421" />
  <input type="hidden" name="okUrl" value="https://www.teststore.com/success.php"/>
  <input type="hidden" name="failUrl" value="https://www.teststore.com/fail.php" />
  <input type="hidden" name="lang" value="en" />
  <input type="hidden" name="rnd" value="asdf" />
  <input type="hidden" input name="pan" value="4242424242424242">
  <input type="hidden" input name="Ecom_Payment_Card_ExpDate_Year" value="28" >
  <input type="hidden" input name="Ecom_Payment_Card_ExpDate_Month" value="10">
  <input type="hidden" input name="cv2" value="000">
```

```
<input type="hidden" input name="encoding" value="utf-8">
<input type="hidden" name="hashAlgorithm" value="ver2">
</form>
```

3.3 3D Authentication

In 3D flow, the 3D authentication information of the customer is queried by the issuer bank. The methods for 3D authentication can be different for different issuers. Examples of 3D authentication methods include using 3D secure passwords, one-time passwords, and security questions.

3.4 Transaction Result

The transaction result will be displayed to the customer. If the transaction is successful, the authorization code will be displayed. The customer will be redirected to `okUrl` if refresh time has passed.

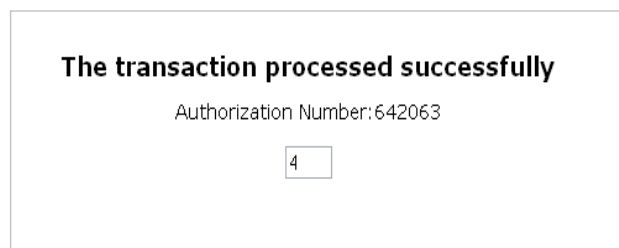


Fig-3

3.5 Merchant Success Page

If the transaction is successful, the customer will be redirected to `okUrl`, which is submitted on step 2 to NestPay Payment Gateway. All parameters posted by the merchant are returned back to the merchant. In addition to merchant parameters, gateway returns the transaction response parameters and MPI response parameters (related to 3D secure transaction flow) which can be found in Appendix A.

3.5.1 Basic transaction response parameters for fully authenticated successful 3D transaction

Response	: "Approved"
AuthCode	: Authorization code of the transaction
HostRefNum	: Host reference number

ProcReturnCode : "00"
TransId : Unique transaction ID
mdStatus : "1"

3.5.1.1 For the example transaction above the transaction response parameters would be

Response : "Approved"
AuthCode : 544889
HostRefNum : 034910000320
ProcReturnCode : "00"
TransId : 103491153310910033
mdStatus : "1"

4. Integration Basics

4.1 HTTP Post Integration

After receiving a valid order, parameters are post to NestPay payment gateway as hidden parameters using the HTTP form. In addition to mandatory parameters, the merchant can post order billing/shipping and order item details to payment gateway which can be viewed later on Merchant Administration Panel. For optional parameters explanations please refer to Appendix A.

The 28 byte-long base-64 encoded xid parameter is the unique Internet transaction ID which is required for 3D secure transactions. If it is not sent by the merchant, it will be generated automatically by the system.

4.1.1 Sample HTTP form with mandatory and optional parameters

```
<form method="post" action="https://host/fim/Nestpaygate">
  <input type="hidden" name="clientid" value="9900000000000001"/>
  <input type="hidden" name="storetype" value="3D_PAY" />
  <input type="hidden" name="hash" value="iej6cPOjDd4IKqXWQEznXWqLzLI=" />
  <input type="hidden" name="trantype" value="Auth" />
  <input type="hidden" name="amount" value="91.96" />
  <input type="hidden" name="currency" value="949" />
  <input type="hidden" name="instalment" value="">
  <input type="hidden" name="oid" value="1291899411421" />
  <input type="hidden" name="okUrl" value="https://www.teststore.com/success.php" />
  <input type="hidden" name="failUrl" value="https://www.teststore.com/fail.php" />
  <input type="hidden" name="lang" value="tr" />
```

```
<input type="hidden" name="rnd" value="asdf" />
<input type="hidden" input name="pan" value="4242424242424242">
<input type="hidden" input name="Ecom_Payment_Card_ExpDate_Year" value="28" >
<input type="hidden" input name="Ecom_Payment_Card_ExpDate_Month" value="10">
<input type="hidden" input name="cv2" value="000">
<input type="hidden" name="tel" value="012345678">
<input type="hidden" name="email" value="test@test.com">
```

<!-- Billing Parameters [All Optional]-->

```
<input type="hidden" name="printBillTo" value="true">
<input type="hidden" name="BillToCompany" value="Billing Company">
<input type="hidden" name="BillToName" value="Bill John Doe">
<input type="hidden" name="BillToStreet1" value="Address line 1">
<input type="hidden" name="BillToStreet2" value="Address line 2">
<input type="hidden" name="BillToStreet3" value="Address line 3">
<input type="hidden" name="BillToCity" value="Istanbul">
<input type="hidden" name="BillToStateProv" value="mystate">
<input type="hidden" name="BillToPostalCode" value="12345">
<input type="hidden" name="BillToCountry" value="616">
```

<!-- Shipping Parameters [All Optional]-->

```
<input type="hidden" name="printShipTo" value="true">
<input type="hidden" name="ShipToCompany" value="Shipping Company">
<input type="hidden" name="ShipToName" value="Ship John Doe">
<input type="hidden" name="ShipToStreet1" value="Address line 1">
<input type="hidden" name="ShipToStreet2" value="Address line 2">
<input type="hidden" name="ShipToStreet3" value="Address line 3">
<input type="hidden" name="ShipToCity" value="Istanbul">
<input type="hidden" name="ShipToStateProv" value="mystate">
<input type="hidden" name="ShipToPostalCode" value="12345">
<input type="hidden" name="ShipToCountry" value="616">
```

<!-- Order Item Parameters [All Optional]-->

```
<input type="hidden" name="ItemNumber1" value="a5">
<input type="hidden" name="ProductCode1" value="a5">
<input type="hidden" name="Qty1" value="3">
<input type="hidden" name="Desc1" value="a5 desc">
<input type="hidden" name="Id1" value="a5">
<input type="hidden" name="Price1" value="6.25">
<input type="hidden" name="Total1" value="7.50">
```

<!--Recurring Payment [All Optional]-->

```
<input type="hidden" name="RecurringPaymentNumber" value="6">
<input type="hidden" name="RecurringFrequencyUnit" value="M">
<input type="hidden" name="RecurringFrequency" value="1">
</form>
```

4.2 Card Transactions

Submission of the form including card data will start the 3D authentication flow with the customer. After the 3D authentication process is completed, the MPI response parameters and all parameters sent by merchant will be posted back to the merchant to make the payment. The payment will be done according to **mdStatus** field which shows the status code of the 3D secure transaction.

4.2.1 MPI Response Parameters

mdStatus	: Status code for the 3D transaction
txstatus	: 3D status for archival
eci	: Electronic Commerce Indicator
cavv	: Cardholder Authentication Verification Value, determined by ACS.
Md	: Hash replacing card number
mdErrorMsg	: Error Message from MPI

4.2.2 Possible mdStatus Values

- 1 = Authenticated transaction (Full 3D)
- 2, 3, 4 = Card not participating or attempt (Half 3D)
- 5, 6, 7, 8 = Authentication not available or system error
- 0 = Authentication failed

4.2.3 Successful Transaction

If the transaction is successful, authorization code will be displayed. The customer will be redirected to *okUrl* of the merchant server if *refreshtime* has passed. All input parameters along with transaction response parameters will be posted to *okUrl*. The *Response* parameter will be **"Approved"**

4.2.4 Failed Transaction

The failure message will be displayed. The customer will be redirected to *failUrl* of merchant server if *refreshtime* is over. All input parameters along with transaction response parameters will be posted to *failUrl*, and the *Response* parameter will be **"Declined"** or **"Error"**.

4.2.5 Transaction Response Parameters

Response	: "Approved", "Declined" or "Error"
AuthCode	: Authorization code of the transaction
HostRefNum	: Host reference number
ProcReturnCode	: Transaction status code
TransId	: Unique transaction ID
ErrMsg	: Error text (if <i>Response</i> is "Declined" or "Error")
ClientIp	: IP address of the customer
ReturnOid	: Returned order ID, must be the same as input oid
MaskedPan	: Masked credit card number
EXTRA.CARDBRAND	: Brand of Credit Card (Visa, MasterCard, etc...)
PaymentMethod	: Payment method of the transaction
Rnd	: Random string, will be used for hash comparison
HASHPARAMS	: Contains the field names used for hash calculation. Field names are appended with ":" character
HASHPARAMSVAL	: Contains the appended hash field values for hash calculation. Field values appended with the same order in <i>HASHPARAMS</i> field
HASH	: Hash value of <i>HASHPARAMSVAL</i> and merchant password field

4.2.6 MPI Response Parameters

mdStatus	: Status code for the 3D transaction
txstatus	: 3D status for archival
eci	: Electronic Commerce Indicator
cavv	: Cardholder Authentication Verification Value, determined by ACS.
mdErrorMsg	: Error Message from MPI (if any)
xid	: Unique Internet transaction ID

4.2.7 Possible Transaction Results

- **Response:** "Approved"

ProcReturnCode will be "00". This shows that the transaction has been authorized.

- **Response:** "Declined"

ProcReturnCode will be a 2 digit number other than "00" and "99" which corresponds to acquirer error code. This shows that the transaction has NOT been authorized by the acquirer. *ErrMsg* parameter will give the detailed description of the error. For detailed description of acquirer error codes for *ProcReturnCode*, refer to Appendix B.

- **Response:** "Error"

ProcReturnCode will be "99". This shows that the transaction has NOT reached to acquirer authorization step. *ErrMsg* parameter will give the detailed description of the error.

4.3 Hash Checking

After merchant receives the parameters, a hash check needs to be done at merchant's server for validating the parameters. Hash checking ensures that the message is sent by Nestpay only. "**hashAlgorithm**" parameter value should be set as "**ver2**" so that Hash Version 2 will be selected as hash calculation method.

4.3.1 Generating the plain text for Hash Ver-2

The parameters used for hash calculation are the following: *clientid*, *oid*, *AuthCode*, *ProcReturnCode*, *Response*, *rnd*, *md*, *eci*, *cavv*, *mdStatus*. Depending on the type of transaction a subset of these parameters will be included in the hash generation:

- HASHPARAMS, HASHPARAMSVAL
- Non 3D-secure card transactions
clientid, oid, AuthCode, ProcReturnCode, Response, rnd
- 3D secure card transactions
clientid, oid, AuthCode, ProcReturnCode, Response, mdStatus, eci, cavv, md, rnd

All the values corresponding to these parameters are appended with the same order. **Note that,** in generation of the hash, pipeline "|" character is used as a separator between parameters. The resulting string will be the same as *HASHPARAMSVAL* parameter values. The merchant password is appended as the final value to the end of this string. The resulting hash is the base64-encoded version of the hashed text which is generated with SHA-512 algorithm. Under normal conditions generated hash text must be the same as *HASH* parameter value posted by Nestpay payment gateway. If not, merchant should contact to Nestpay support team.

Note: In generation of the hash, pipeline "|" character is used as a separator between parameters. The resulting string will be the same as *HASHPARAMSVAL* parameter values. The merchant password is appended as the final value to the end of this string. The resulting hash is the base64-encoded version of the hashed text which is generated with SHA-512 algorithm. Under normal conditions, generated hash text must be the same as *HASH* parameter value posted by NestPay payment gateway. If not, merchant should contact to Nestpay support team.

Note: In case "\" character needs to be used in a parameter, "\\" character can be used for escaping. Additionally, if the "\" character is also used in a parameter, we use another "\" character before it and then append it to hash plain text. An example is shown below:

Original order id : ORDER-256712jbs\j6b|

Escaped order id : ORDER-256712jbs\\j6b\\|

Example: Non 3D card transaction

Assuming that the transaction response parameters are:

clientid, oid, AuthCode, ProcReturnCode, Response, rnd

clientid : 9900000000000001
oid : ORDER256712jbs\\j6b\\|
AuthCode : 213216
ProcReturnCode : 540000
Response : Approved
rnd : LdXGnw20ZupyXVr1XCu2
storeKey : AB123456\\|

HASHPARAMSVAL :
 9900000000000001|ORDER256712jbs\\j6b\\||213216|540000|Approved|LdXGnw20ZupyXVr1XCu2
HASHPARAMS : clientid|oid|AuthCode|ProcReturnCode|Response|rnd
HASH :
 UfrEVCY1IRoThXE571XkSK8a/LnoJOCLgijId+by7qGLKlyChhgnbwRwEJrFQSpSacH9DYk0Zk
 up2cmyFXoDLg==

The merchant hash text will be generated with clientid, oid, ProcReturnCode, Response, rnd (and store key of the merchant as secret hash element).

And the merchant hash is based64-encoded(SHA512(plain)). The result hash must be the same as the returning parameter *HASH*

Note that Merchant has to check Hash parameter of HASHPARAMS & HASHPARAMSVAL return values by the bank.

4.3.2 Security Notes

Merchant must check the mandatory parameters in **HASHPARAMS** parameter. For approved provisions, *ClientId*, *OrderId* and *Response* parameter names must be present in

HASHPARAMS. Merchant must also check the values of these mandatory return parameters. *ClientId* must be the clientId assigned to the merchant.

If these parameters are not present or have wrong values, merchant should not process this transaction and must report it to the Support Desk immediately.

5. Code Samples

The following procedure for 3D Pay Model areas. Values test purposes had been inserted. 3D Pay Model on edited code examples. Merchants, taking into account variables must define values for them. These codes reference purpose formed. ASP Code Sample

5.1 ASP Code Sample

5.1.1 Request Sample Codes

5.1.1.1 Hash Version 2

For ASP Classic, there is no built-in SHA-512 Hash Calculation library. Because of this reason, ASP classic code should be extended to use C# or VB.Net code which have a built-in library for SHA 512 hash generation. Below, you can find example C# and VB Code example for Hash Version 2.

5.1.2 Response Code Sample

```
<html>
<head>
<title>3D Pay Payment Page</title>
  <meta http-equiv="Content-Language" content="tr">
  <meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-9">
  <meta http-equiv="Pragma" content="no-cache">
  <meta http-equiv="Expires" content="now">

</head>

<body>
<!-- #include file = "hex_sha1_js.asp" -->
<h1>Payment Page</h1>

  <h3> Payment Response</h3>
  <table border="1">
    <tr>
```

```
<td><b>Parameter Name</b></td>
<td><b>Parameter Value</b></td>
</tr>
```

```
dim
obj,ok,mdstatus,hashparams,hashparamsval,hash,index1,index2,storekey,hashparam,val,hashval,par
amsval
dim paymentparams(5)
ok = 1
```

'hash checking parameters

```
storekey = "xxxxxx"
index1 = 1
index2 = 1
hashparams = request.form("HASHPARAMS")
hashparamsval = request.form("HASHPARAMSVAL")
hashparam = request.form("HASH")
paramsval = ""
paymentparams (0) = "AuthCode"
paymentparams (1) = "Response"
paymentparams (2) = "HostRefNum"
paymentparams (3) = "ProcReturnCode"
paymentparams (4) = "TransId"
paymentparams (5) = "ErrMsg"

for each obj in request.form
    ok = 1
    for each item in paymentparams
        if(item = obj) Then
            ok = 0
            exit for
        end if
    next
    if ok = 1 then
        response.write("<tr><td>"&obj & "</td><td>" & request.form(obj) & "</td></tr>")
    end if
next

</table>
<br>
<br>
```

'hash cheking

```

while index1 < Len(hashparams)
    index2 = InStr(index1,hashparams,":")
    xvalx = Mid(hashparams,index1,index2 - index1)
    val = request.form(xvalx)
    if val = null then
        val = ""
    end if
    paramsval = paramsval & val
    index1 = index2 + 1

Wend

hashval = paramsval & storekey
hash = b64_sha1(hashval)
'response.write("hash=" & hash & "<br/>hashparam=" & hashparam & "<br/>paramsval=" &
paramsval & "<br/>hashparamsval=" & hashparamsval )

if hash <> hashparam or paramsval <> hashparamsval then
    response.write("<h4>Security Alert. The digital signature is not valid.</h4>")
end if

mdstatus = Request.Form("mdStatus")
if mdstatus = 1 or mdstatus = 2 or mdstatus = 3 or mdstatus = 4 Then

<h5>3D Transaction is Success</h5><br/>
    <h3> Payment Host</h3>
    <table border="1">
        <tr>
            <td><b>Parameter Name</b></td>
            <td><b>Parameter Value</b></td>
        </tr>

for each item in paymentparams
    response.Write("<tr><td>" & item & "</td><td>" & request.form(item) & "</td></tr>")
next

</table>

if "Approved" = request.form("Response") Then

```

```

        Response.write("<h6>Transaction is Success</h6>")
    Else
        Response.write("<h6>Transaction is not Success</h6>")
    end if

    else
        Response.Write("<h6>3D not Approved </h6>")
    end if
</body>
</html>

```

5.2 .Net Code Sample

5.2.1 Request Sample Codes

5.2.1.1 Hash Version 2

5.2.1.1.1.Net - C# Code Sample

```

<%@ page language="C#" autoeventwireup="true"
    inherits="_3DModel, App_Web_fr4klrwv"%>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
<title>3D Model</title>
</head>
<body>

    <%
        // plaintext = clientid + | + oid + | + amount +| + okurl +| + failurl + | + transaction type + | +
        instalment +
        // | + rnd +|||| + currency +| + storeKey

        //unEscaped values
        String orgClientId = "9900000000000001";
        String orgOid = "ORDER256712jbs\j6b|";
        String orgAmount = "91.96";
        String orgOkUrl = "https://www.teststore.com/success.php";
    %>

```

```
String orgFailUrl = "https://www.teststore.com/fail.php";
String orgTransactionType = "Auth";
String orgInstallment = "";
String orgRnd = DateTime.Now.ToString();

String orgCurrency = "949";
// escaped values
String clientId = orgClientId.Replace("\\", "\\").Replace("|", "\\|");
String oid = orgOid.Replace("\\", "\\").Replace("|", "\\|");
String amount = orgAmount.Replace("\\", "\\").Replace("|", "\\|");
String okUrl = orgOkUrl.Replace("\\", "\\").Replace("|", "\\|");
String failUrl = orgFailUrl.Replace("\\", "\\").Replace("|", "\\|");
String transactionType = orgTransactionType.Replace("\\", "\\").Replace("|", "\\|");
String installment = orgInstallment.Replace("\\", "\\").Replace("|", "\\|");
String rnd = orgRnd.Replace("\\", "\\").Replace("|", "\\|");

String currency = orgCurrency.Replace("\\", "\\").Replace("|", "\\|");
String storeKey = "AB123456\\|".Replace("\\", "\\").Replace("|", "\\|");

String plainText = clientId + "|" + oid + "|" + amount + "|" + okUrl + "|" + failUrl + "|"
    + transactionType + "|" + installment + "|" + rnd + "||||" + currency + "|" + storeKey;

System.Security.Cryptography.SHA512 sha = new
System.Security.Cryptography.SHA512CryptoServiceProvider();
byte[] hashbytes = System.Text.Encoding.GetEncoding("ISO-8859-9").GetBytes(plainText);
byte[] inputbytes = sha.ComputeHash(hashbytes);
String hash = Convert.ToBase64String(inputbytes);

String description = "";
String xid = "";
String lang = "";
String email = "";
String userid = "";
%>

<center>

    <form method="post" action="https://<Host_Address>/<3dgate_path>">
        <table>
            <tr>

                <td>Credit Card Number</td>
```

```

        <td><input type="text" name="pan" size="20" />
    </tr>

    <tr>

        <td>CVV</td>

        <td><input type="text" name="cv2" size="4" value="" /></td>

    </tr>

    <tr>

        <td>Expiration Date Year</td>

        <td><input type="text" name="Ecom_Payment_Card_ExpDate_Year"
            value="" /></td>

    </tr>

    <tr>

        <td>Expiration Date Month</td>

        <td><input type="text"
            name="Ecom_Payment_Card_ExpDate_Month value="" /></td>

    </tr>

    <tr>

        <td>Choosing Visa Master Card</td>

        <td><select name="cardType">

            <option value="1">Visa</option>
            <option value="2">MasterCard</option>

        </select>

    </tr>

    <tr>

```

```

        <td align="center" colspan="2"><input type="submit"
            value="Complete Payment" /></td>

    </tr>

</table>

    <input type="hidden" name="clientid" value="<%=orgClientId%>">
    <input type="hidden" name="amount" value="<%=orgAmount%>">
    <input type="hidden" name="oid" value="<%=orgOid%>">
    <input type="hidden" name="okurl" value="<%=orgOkUrl%>">
    <input type="hidden" name="failUrl" value="<%=orgFailUrl%>">
    <input type="hidden" name="TranType" value="<%=orgTransactionType%>">
    <input type="hidden" name="Instalment" value="<%=orgInstallment%>">

    <input type="hidden" name="currency" value="<%=orgCurrency%>">
    <input type="hidden" name="rnd" value="<%=orgRnd%>">
    <input type="hidden" name="hash" value="<%=hash%>">
    <input type="hidden" name="storetype" value="3D_PAY">
    <input type="hidden" name="lang" value="tr">
    <input type="hidden" name="hashAlgorithm" value="ver2">

</form>

</center>

</body>

</html>

```

5.2.1.1.2.Net - VB Code Sample

```

<%@ page language="VB" autoeventwireup="true"
    inherits="_3DModel, App_Web_fr4klrww"%>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">

```

```

<title>3D Model</title>
</head>
<body>

    <%

        Dim orgClientId As String = "9900000000000001"
        Dim orgOid As String = "ORDER256712jbs\j6b|"
        Dim orgAmount As String = "91.96"
        Dim orgOkUrl As String = "https://www.teststore.com/success.php"
        Dim orgFailUrl As String = "https://www.teststore.com/fail.php"
        Dim orgTransactionType As String = "Auth"
        Dim orgInstallment As String = ""
        Dim orgRnd As String = DateTime.Now.ToString()

        Dim orgCurrency As String = "949"

        Dim clientId As String = orgClientId.Replace("\", "\\").Replace("|", "\\|")
        Dim oid As String = orgOid.Replace("\", "\\").Replace("|", "\\|")
        Dim amount As String = orgAmount.Replace("\", "\\").Replace("|", "\\|")
        Dim okUrl As String = orgOkUrl.Replace("\", "\\").Replace("|", "\\|")
        Dim failUrl As String = orgFailUrl.Replace("\", "\\").Replace("|", "\\|")
        Dim transactionType As String = orgTransactionType.Replace("\",
        "\\").Replace("|", "\\|")
        Dim installment As String = orgInstallment.Replace("\", "\\").Replace("|", "\\|")
        Dim rnd As String = orgRnd.Replace("\", "\\").Replace("|", "\\|")

        Dim currency As String = orgCurrency.Replace("\", "\\").Replace("|", "\\|")
        Dim storeKey As String = "AB123456\|.Replace("\", "\\").Replace("|", "\\|")

        Dim plainText As String = clientId + "|" + oid + "|" + amount + "|" + okUrl + "|"
        + failUrl + "|" + transactionType + "|" + installment + "|" + rnd + "||||" + currency + "|" +
        storeKey

        Dim result As Byte()
        Dim mixer As String
        Dim sha As New System.Security.Cryptography.SHA512Managed()

        result = sha.ComputeHash(System.Text.Encoding.ASCII.GetBytes(plainText))
        Dim hashValue As String = Convert.ToBase64String(result)
    %>

```



```

Dim description As String = "";
Dim xid As String = "";
Dim lang As String = "";
Dim email As String = "";
Dim userid As String = "";

%>

<center>

<form method="post" action="https://<Host_Address>/<3dgate_path>">
    <table>
        <tr>

            <td>Credit Card Number</td>

            <td><input type="text" name="pan" size="20" />
        </tr>

        <tr>

            <td>CVV</td>

            <td><input type="text" name="cv2" size="4" value="" /></td>
        </tr>

        <tr>

            <td>Expiration Date Year</td>

            <td><input type="text" name="Ecom_Payment_Card_ExpDate_Year"
                value="" /></td>
        </tr>

        <tr>

            <td>Expiration Date Month</td>

            <td><input type="text"
                name="Ecom_Payment_Card_ExpDate_Month value="" /></td>
        </tr>
    </table>
</form>

```

```

        <tr>

            <td>Choosing Visa Master Card</td>
                <td><select name="cardType">

                    <option value="1">Visa</option>
                    <option value="2">MasterCard</option>

                </select>
            </td>

        </tr>

        <tr>

            <td align="center" colspan="2"><input type="submit"
                value="Complete Payment" /></td>
        </tr>

    </table>

    <input type="hidden" name="clientid" value="<%=orgClientId%>">
        <input type="hidden" name="amount" value="<%=orgAmount%>">
        <input type="hidden" name="oid" value="<%=orgOid%>">
        <input type="hidden" name="okurl" value="<%=orgOkUrl%>">
        <input type="hidden" name="failUrl" value="<%=orgFailUrl%>">
        <input type="hidden" name="TranType" value="<%=orgTransactionType%>">
        <input type="hidden" name="Instalment" value="<%=orgInstallment%>">

        <input type="hidden" name="currency" value="<%=orgCurrency%>">
        <input type="hidden" name="rnd" value="<%=orgRnd%>">
        <input type="hidden" name="hash" value="<%=hash%>">
        <input type="hidden" name="storetype" value="3D_PAY ">
        <input type="hidden" name="lang" value="tr">
        <input type="hidden" name="hashAlgorithm" value="ver2">

    </form>

</center>

</body>

```

</html>

5.2.2 Response Code Sample

5.2.2.1 .Net - C# Sample Code

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
<title>3d Pay Payment Page</title>
</head>
<body>
    <h1>3D Payment Page</h1>
    <h3>Payment Response</h3>
    <table border="1">

        <%
String originalClientId = "xxxxxx";
String [] mustParameters = new String[] { "clientid", "oid", "Response" };
boolean isValid = true;
for(int i=0;i<mustParameters.length;i++)
{
    if(Request.Form[mustParameters[i]] == null || Request.Form[mustParameters[i]] == "" )
    {
        if(mustParameters[i].equals("oid")){
            if(Request.Form["ReturnOid"] == null || Request.Form["ReturnOid"] == "" ){
                isValid = false;
                Response.Write("<tr><td>Missing Required Param</td>"+ "<td>oid /
ReturnOid</td></tr>");
            }
        }else{
            isValid = false;
            Response.Write("<tr><td>Missing Required
Param</td>"+ "<td>"+mustParameters[i]+"</td></tr>");
        }
    }
}
if(!Request.Form.Get("clientid").Equals(originalClientId)){
    Response.Write("<h4>Security Alert. Incorrect Client Id.</h4>");
    return;
}
```

```

        if(!isValid){
            Response.Write("<h4>Security Alert. The digital signature is not valid. Required Paramaters are
missing.</h4>");
            return;
        } else {

%>
        <tr>
            <td><b>Parameter Name</b></td>
            <td><b>Parameter Value</b></td>
        </tr>
        <%
            String[] paymentparams = new String[] { "AuthCode", "Response", "HostRefNum",
"ProcReturnCode",
            "TransId", "ErrMsg" };
            IEnumerator e = Request.Form.GetEnumerator();
            while (e.MoveNext()) {
                String xkey = (String) e.Current;
                String xval = Request.Form.Get(xkey);
                boolean ok = true;
                for (int i = 0; i < paymentparams.Length; i++) {
                    if (xkey.Equals(paymentparams [i])) {
                        ok = false;
                        break;
                    }
                }
                if (ok)
                    Response.Write("<tr><td>" + xkey + "</td><td>" + xval + "</td></tr>");
            }
        %>
    </table>

    <%
        String hashparams = Request.Form["HASHPARAMS"];
        String hashparamsval = Request.Form["HASHPARAMSVAL"];
        String hash = "";
        String storekey = "xxxxxx";
        String paramsval = "";
        String hashval = "";
        int index1 = 0, index2 = 0;

        if (Request.Form.Get("hashAlgorithm").Equals("ver2")){
            string[] parsedParams = hashparams.Split('|');

```

```

        foreach (string parsedParam in parsedParams)
        {
            String val = Request.Form.Get(parsedParam) == null ? "" :
Request.Form.Get(parsedParam);
            paramsval += val.Replace("\\", "\\").Replace("|", "\\|") + "|";
        }

        hashval = paramsval + storekey.Replace("\\", "\\").Replace("|", "\\|");
        String hashparam = Request.Form.Get("HASH");

        System.Security.Cryptography.SHA512 sha = new
System.Security.Cryptography.SHA512CryptoServiceProvider();
        byte[] hashbytes = System.Text.Encoding.GetEncoding("ISO-8859-
9").GetBytes(hashval);
        byte[] inputbytes = sha.ComputeHash(hashbytes);
        hash = Convert.ToBase64String(inputbytes);

    } else {
        do
        {
            index2 = hashparams.IndexOf(":", index1);
            String val = Request.Form.Get(hashparams.Substring(index1, index2-index1))
==
            null ? "" : Request.Form.Get(hashparams.Substring(index1, index2-index1));
            paramsval += val;
            index1 = index2 + 1;
        }
        while (index1 < hashparams.Length);

        hashval = paramsval + storekey;
        String hashparam = Request.Form.Get("HASH");

        System.Security.Cryptography.SHA1 sha = new
System.Security.Cryptography.SHA1CryptoServiceProvider();
        byte[] hashbytes = System.Text.Encoding.GetEncoding("ISO-8859-
9").GetBytes(hashval);
        byte[] inputbytes = sha.ComputeHash(hashbytes);

        hash = Convert.ToBase64String(inputbytes);
    }

    if (!paramsval.Equals(hashparamsval) || !hash.Equals(hashparam)) {
        Response.Write("<h4>Security Alert. The digital signature is not valid.</h4>");
    }

```

```

        Response.Write("<h4>Generated Hash Val : " + paramsval + "</h4>");
        Response.Write("<h4>Original Hash Val : " + hashparamsval + "</h4>");
    }

    String mdStatus = Request.Form["mdStatus"];
    if (mdStatus.Equals("1") || mdStatus.Equals("2") || mdStatus.Equals("3") ||
mdStatus.Equals("4")) {
%>
<h5>3D Transaction is Success</h5>
<br />
<h3>Payment Response</h3>
<table border="1">
    <tr>
        <td><b>Parameter Name</b></td>
        <td><b>Parameter Value</b></td>
    </tr>

    <%
        for (int i = 0; i < paymentparams.Length; i++) {
            String paramname = paymentparams [i];
            String paramval = Request.Form.Get(paramname);
            Response.Write("<tr><td>" + paramname + "</td><td>" + paramval +
"</td></tr>");
        }
    %>
</table>

<%
    if ("Approved".Equals(Request.Form["Response"])) {
%>
<h6>Transaction is Success</h6>
<%
    } else {
%>
<h6>Transaction is not Success</h6>
<%
    }
    } else {
%>

<h5>3D Transaction is not Success</h5>

<%

```

```

    }
}
%>
</body>
</html>

```

5.2.2.2 .Net - VB.Net Sample Code

```

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
<title>3d Pay Payment Page</title>
</head>
<body>
    <h1>3D Payment Page</h1>
    <h3>Payment Response</h3>
    <table border="1">

        <%
            Dim originalClientId As String = "xxxxxx"
            Dim mustParameters() As String = {"clientid","oid","Response"}
            Dim isValid As Boolean = True

            For Each paramName As String In mustParameters
                Dim paramValue As String = Request.Form(paramName)
                If String.IsNullOrEmpty(paramValue) Then

                    If paramName.Equals("oid") Then
                        Dim returnOidValue As String = Request.Form("ReturnOid")
                        If String.IsNullOrEmpty(returnOidValue) Then
                            isValid = False
                            Response.Write("<tr><td>Missing Required Param</td>"+ "<td>oid /
ReturnOid</td></tr>")
                        End If
                    Else
                        isValid = false;
                        Response.Write("<tr><td>Missing Required Param</td>"+ "<td>"+
paramName+"</td></tr>")
                    End If
                End If
            Next

            If Not Request.Form("clientid").Equals(originalClientId) Then

```

```

        Response.Write("<h4>Security Alert. Incorrect Client Id.</h4>")
    End If

    If Not isValid Then
        Response.Write("<h4>Security Alert. The digital signature is not valid. Required Paramaters are missing.</h4>")
    Else

%>
        <tr>
            <td><b>Parameter Name</b></td>
            <td><b>Parameter Value</b></td>
        </tr>
    <%
        Dim paymentparams() As String = { "AuthCode", "Response", "HostRefNum", "ProcReturnCode",
"TransId", "ErrMsg" }
        Dim allKeys() As String = Request.Params.AllKeys

        For Each xKey As String In allKeys
            Dim xval As String = Request.Form(xkey)
            Dim ok As Boolean = True

            For Each paymentParam As String In paymentparams
                If xkey.Equals(paymentParam) Then
                    ok = False
                End If
            Exit For
            End If
            Next

            If ok Then
                Response.Write("<tr><td>" + xkey + "</td><td>" + xval + "</td></tr>")
            End If
        Next
    %>
</table>

<%
    Dim hashparams As String = Request.Form("HASHPARAMS")
    Dim hashparamsval As String = Request.Form("HASHPARAMSVAL")
    Dim storekey As String = "xxxxxx"
    Dim paramsval As String = ""
    Dim index1 As Integer = 0
    Dim index2 As Integer = 0

```



```

Dim hash As String = ""
Dim hashparam As String = ""
Dim hashval As String = ""

If Request.Form("hashAlgorithm") == "ver2" Then
    Dim parsedParams() As String = hashparams.split("|")

    For Each parsedParam As String In parsedParams
        If String.IsNullOrEmpty(parsedParam) Then
            val = ""
        Else
            val = parsedParam
        End If
        paramsval += val.Replace "\", "\\").Replace("|", "\\|") + "|"
    Next

    Dim hashval As String = paramsval + storekey.Replace "\", "\\").Replace("|", "\\|")
    Dim hashparam As String = Request.Form("HASH")

    Dim result As Byte()
    Dim sha As New System.Security.Cryptography.SHA512Managed()
    result = sha.ComputeHash(System.Text.Encoding.ASCII.GetBytes(hashval))
    hash = Convert.ToBase64String(result)

Else

    Do While index1 < hashparams.Length
        index2 = hashparams.IndexOf(":", index1)
        Dim hashedParamValue As String = Request.Form(hashparams.Substring(index1,
index2 - index1))
        Dim val As String
        If String.IsNullOrEmpty(hashedParamValue) Then
            val = ""
        Else
            val = hashedParamValue
        End If
        paramsval += val
        index1 = index2 + 1
    Loop

    Dim hashval As String = paramsval + storekey
    Dim hashparam As String = Request.Form("HASH")

```

```

        Dim result As Byte()
        Dim sha As New System.Security.Cryptography.SHA512Managed()
        result = sha.ComputeHash(System.Text.Encoding.ASCII.GetBytes(hashval))
        hash As String = Convert.ToBase64String(result)

    End If

    If (Not paramsval.Equals(hashparamsval)) Or (Not hash.Equals(hashparam)) Then
        Response.Write("<h4>Security Alert. The digital signature is not valid.</h4>")
        Response.Write("<h4>Generated Hash Val : " + paramsval + "</h4>");
        Response.Write("<h4>Original Hash Val : " + hashparamsval + "</h4>");
    End If

    String mdStatus = Request.Form("mdStatus")
    If mdStatus.Equals("1") Or mdStatus.Equals("2") Or mdStatus.Equals("3") Or
mdStatus.Equals("4") Then
%>
<h5>3D Transaction is Success</h5>
<br />
<h3>Payment Response</h3>
<table border="1">
    <tr>
        <td><b>Parameter Name</b></td>
        <td><b>Parameter Value</b></td>
    </tr>

    <%
    For Each paramname As String In paymentparams
        Dim paramval As String = Request.Form(paramname)
        Response.Write("<tr><td>" + paramname + "</td><td>" + paramval + "</td></tr>")
    Next
    %>
</table>

<%
    If "Approved".Equals(Request.Form("Response")) Then
%>
<h6>Transaction is Success</h6>
<%
    Else
%>
<h6>Transaction is not Success</h6>

```

```

    <%
        End If
    Else
    %>

    <h5>3D Transaction is not Success</h5>

    <%
        End If
    End If
    %>
</body>
</html>

```

5.3 JSP Code Sample

5.3.1 Request Sample Codes

5.3.1.1 Hash Version 2

```

<%@page contentType="text/html; charset=ISO-8859-9"%>
<%@page import="org.apache.commons.codec.binary.Base64" %>
<%@page import="java.security.MessageDigest"%>

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">

<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-9">
<title>Ver2 Request</title>
</head>
<body>
    <%
        String storeType="3D_PAY ";
        //unEscaped values
        String orgClientId = "xxxxxxxxxx";
        String storeType="3D_PAY";
        String orgOid = "";
        String orgAmount = "95.93";
        String orgOkUrl = "http://localhost:8080/SampleCodeJSPTTest/GateResponseControl.jsp";
    %>

```

```
String orgFailUrl = "http://localhost:8080/SampleCodeJSPTest/GateResponseControl.jsp";
String orgTransactionType = "Auth";
String orgInstallment = "";
String orgRnd = new java.util.Date().toString();

String orgCurrency = "949";
// escaped values
String clientId = orgClientId.replace("\\", "\\\\").replace("|", "\\|");
String oid = orgOid.replace("\\", "\\\\").replace("|", "\\|");
String amount = orgAmount.replace("\\", "\\\\").replace("|", "\\|");
String okUrl = orgOkUrl.replace("\\", "\\\\").replace("|", "\\|");
String failUrl = orgFailUrl.replace("\\", "\\\\").replace("|", "\\|");
String transactionType = orgTransactionType.replace("\\", "\\\\").replace("|", "\\|");
String installment = orgInstallment.replace("\\", "\\\\").replace("|", "\\|");
String rnd = orgRnd.replace("\\", "\\\\").replace("|", "\\|");

String currency = orgCurrency.replace("\\", "\\\\").replace("|", "\\|");
String storeKey = "AB123456".replace("\\", "\\\\").replace("|", "\\|");

String plainText = clientId + "|" + oid + "|" + amount + "|" + okUrl + "|" + failUrl + "|"
    + transactionType + "|" + installment + "|" + rnd + "||||" + currency + "|" + storeKey;

MessageDigest messageDigest = MessageDigest.getInstance("SHA-512");
messageDigest.update(plainText.getBytes());
String hash= new String(Base64.encodeBase64(messageDigest.digest()),"UTF-8");

String description = "";
String xid = "";
String lang="";
String email="";
String userid="";
%>
<center>
    <form method="post" action="http://localhost:8080/fim/est3dgate">
        <table>
            <tr>
                <td>Credit Card Number</td>
                <td><input type="text" name="pan" size="20" value="" />
            </tr>
            <tr>
                <td>CVV</td>
                <td><input type="text" name="cv2" size="4" value="" /></td>
            </tr>
        </table>
    </form>
</center>
```

```

        </tr>
        <tr>
            <td>Expiration Date Year</td>
            <td><input type="text" name="Ecom_Payment_Card_ExpDate_Year"
                value="" /></td>
        </tr>
        <tr>
            <td>Expiration Date Month</td>
            <td><input type="text" name="Ecom_Payment_Card_ExpDate_Month"
                value="" /></td>
        </tr>
        <tr>
            <td>Choosing Visa / Master Card</td>
            <td><select name="cardType">
                <option value="1">Visa</option>
                <option value="2">MasterCard</option>
            </select>
        </tr>
        <tr>
            <td align="center" colspan="2"><input type="submit"
value="Complete Payment" /></td>
        </tr>
    </table>

    <input type="hidden" name="clientid" value="<%=orgClientId%>">
    <input type="hidden" name="amount" value="<%=orgAmount%>">
    <input type="hidden" name="oid" value="<%=orgOid%>">
    <input type="hidden" name="okurl" value="<%=orgOkUrl%>">
    <input type="hidden" name="failUrl" value="<%=orgFailUrl%>">
    <input type="hidden" name="TranType" value="<%=orgTransactionType%>">
    <input type="hidden" name="Instalment" value="<%=orgInstallment%>">

    <input type="hidden" name="currency" value="<%=orgCurrency%>">
    <input type="hidden" name="rnd" value="<%=orgRnd%>">
    <input type="hidden" name="hash" value="<%=hash%>">
    <input type="hidden" name="storetype" value="<%=storeType%>">
    <input type="hidden" name="lang" value="tr">
    <input type="hidden" name="hashAlgorithm" value="ver2">

</form>
</center>
</body>
</html>

```

5.3.2 Response Code Sample

```
<%@page import="java.util.Enumeration" %>
<%@page import="org.apache.commons.codec.binary.Base64" %>
<%@page import="java.security.MessageDigest"%>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-9">
<title>Ver2 Response</title>
</head>
<body>
    <h1>Payment Page</h1>
    <h3>Payment Response</h3>
    <table border="1">
    <%
String originalClientId = "xxxxxxxxx";
String [] mustParameters = new String[] {"clientid","oid","Response"};
boolean isValid = true;

for(int i=0;i<mustParameters.length;i++){
    if(request.getParameter(mustParameters[i]) == null ||
request.getParameter(mustParameters[i]) == "" ){
        if(mustParameters[i].equals("oid")){
            if(request.getParameter("ReturnOid") == null || request.getParameter("ReturnOid") == ""
){
                isValid = false;
                out.println("<tr><td>Missing Required Param</td>"+ "<td>oid /
ReturnOid</td></tr>");
            }
        }else{
            isValid = false;
            out.println("<tr><td>Missing Required
Param</td>"+ "<td>"+mustParameters[i]+"</td></tr>");
        }
    }
}

if(!request.getParameter("clientid").equals(originalClientId)){
    out.println("<h4>Security Alert. Incorrect Client Id.</h4>");
    return;
}
```

```

    if(!isValid){
        out.println("<h4>Security Alert. The digital signature is not valid. Required Paramaters are
missing.</h4>");
        return;
    } else {
%>
        <tr>
            <td><b>Parameter Name</b></td>
            <td><b>Parameter Value</b></td>
        </tr>
        <%
Enumeration enu = request.getParameterNames();
while(enu.hasMoreElements()){
    String param = (String)enu.nextElement();
    String val = (String)request.getParameter(param);
    out.println("<tr><td>"+param+"</td>"+<td>"+val+"</td></tr>");
} %>
</table>
<br>
<%
String hashparams = request.getParameter("HASHPARAMS");
String hashparamsval = request.getParameter("HASHPARAMSVAL");
String storekey="AB123456";
String paramsval="";
String hashval = "";
String hash = "";
int index1=0,index2=0;
if(request.getParameter("hashAlgorithm").equals("ver2")){

    String[] parsedParams = hashparams.split("|");

    for(String parsedParam: parsedParams){
        String val = request.getParameter(parsedParam) == null ? "" :
request.getParameter(parsedParam);
        paramsval += val.replace("\\", "\\").replace("|", "\\|") + "|";
    }

    hashval = paramsval + storekey.replace("\\", "\\").replace("|", "\\|");
    String hashparam = request.getParameter("HASH");

    MessageDigest messageDigest = MessageDigest.getInstance("SHA-512");
    messageDigest.update(hashval.getBytes());
    hash= new String(Base64.encodeBase64(messageDigest.digest()),"UTF-8");

```

```

} else {
    do{
        index2 = hashparams.indexOf(":",index1);
        String val = request.getParameter(hashparams.substring(index1,index2)) == null ? "" :
request.getParameter(hashparams.substring(index1,index2));
        paramsval += val;
        index1 = index2 + 1;
    }
    while(index1<hashparams.length());

    hashval = paramsval + storekey;
    String hashparam = request.getParameter("HASH");

    MessageDigest messageDigest = MessageDigest.getInstance("SHA-512");
    messageDigest.update(hashval.getBytes());
    hash= new String(Base64.encodeBase64(messageDigest.digest()),"UTF-8");

}

if(!paramsval.equals(hashparamsval)|| !hash.equals(hashparams)) {
    out.println("<h4>Security Alert. The digital signature is not valid.</h4>");
    out.println("<h4>Generated Hash Val : " + paramsval + "</h4>");
    out.println("<h4>Original Hash Val : " + hashparamsval + "</h4>");
}

String mdStatus = request.getParameter("mdStatus");
if(mdStatus!=null && (mdStatus.equals("1") || mdStatus.equals("2") || mdStatus.equals("3")||
mdStatus.equals("4"))){

%>
<h5>3D Transaction is Success</h5>
<br />
<h3>Payment Response</h3>
<table border="1">
    <tr>
        <td><b>Parameter Name</b></td>
        <td><b>Parameter Value</b></td>
    </tr>
<%
    String [] paymentparams = new String[]
{"AuthCode","Response","HostRefNum","ProcReturnCode","TransId","ErrMsg"};

    for(int i=0;i< paymentparams.length;i++){

```



```

        String paramname = paymentparams [i];
        String paramval = request.getParameter(paramname);
        out.println("<tr><td>"+paramname+"</td><td>"+paramval+"</td></tr>");
    }
    %>
</table>
<%
if("Approved".equalsIgnoreCase(request.getParameter("Response"))){ %>
<h6>Transaction is Success</h6>
<%
    }else{ %>
<h6>Transaction is not Success</h6>
<% }
} else { %>
<h5>3D Transaction is not Success</h5>
<%}
} %>
</body>
</html>

```

5.4 PHP Code Sample

5.4.1 Request Sample Codes

5.4.1.1 Hash Version 2

```

<html>

<head>

<title>3D</title>

<meta http-equiv="Content-Language" content="tr">

<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-9">

<meta http-equiv="Pragma" content="no-cache">

<meta http-equiv="Expires" content="now">

```

```
</head>
```

```
<body>
```

```
<?php
```

```
$orgClientId = "9900000000000001";
$orgOid = "ORDER256712jbs\j6b|";
$orgAmount = "91.96";
$orgOkUrl = "https://www.teststore.com/success.php";
$orgFailUrl = "https://www.teststore.com/fail.php";
$orgTransactionType = "Auth";
$orgInstallment = "";
$orgRnd = microtime();
$orgCurrency = "949";
```

```
$clientId = str_replace("|", "\\|", str_replace("\\", "\\\\", $orgClientId));
$oid = str_replace("|", "\\|", str_replace("\\", "\\\\", $orgOid));
$amount = str_replace("|", "\\|", str_replace("\\", "\\\\", $orgAmount));
$okUrl = str_replace("|", "\\|", str_replace("\\", "\\\\", $orgOkUrl));
$failUrl = str_replace("|", "\\|", str_replace("\\", "\\\\", $orgFailUrl));
$transactionType = str_replace("|", "\\|", str_replace("\\", "\\\\", $orgTransactionType));
$installment = str_replace("|", "\\|", str_replace("\\", "\\\\", $orgInstallment));
$rnd = str_replace("|", "\\|", str_replace("\\", "\\\\", microtime()));
$currency = str_replace("|", "\\|", str_replace("\\", "\\\\", $orgCurrency));
$storeKey = str_replace("|", "\\|", str_replace("\\", "\\\\", "AB123456\\|"));
```

```
$plainText = $clientId . "|" . $oid . "|" . $amount . "|" . $okUrl . "|" . $failUrl . "|" .
$transactionType . "|" . $installment . "|" . $rnd . "||||" . $currency . "|" . $storeKey;
```

```
$hashValue = hash('sha512', $plainText);
$hash = base64_encode (pack('H*', $hashValue)) ;
```

```
$description = "";
$xid = "";
$lang="";
$email="";
$userid="";
```

```
?>
```

<center>

<form method="post" action="https://<host_address>/<3dgate_path>">

<table>

<tr>

<td>Credit Card Number</td>

<td><input type="text" name="pan" size="20" />

</tr>

<tr>

<td>CVV</td>

<td><input type="text" name="cv2" size="4" value="" /></td>

</tr>

<tr>

<td>Expiration Date Year</td>

<td><input type="text" name="Ecom_Payment_Card_ExpDate_Year" value="" /></td>

</tr>

<tr>

<td>Expiration Date Month</td>

<td><input type="text" name="Ecom_Payment_Card_ExpDate_Month" value="" /></td>

</tr>

<tr>

<td>Choosing Visa / Master Card</td>

<td><select name="cardType">

<option value="1">Visa</option>

<option value="2">MasterCard</option>

</select>

```

        </tr>

        <tr>

            <td align="center" colspan="2"><input type="submit"
                value="Complete Payment" /></td>

        </tr>

    </table>

    <input type="hidden" name="clientid" value="<?php echo $orgClientId ?>">
    <input type="hidden" name="amount" value="<?php echo $orgAmount ?>">
    <input type="hidden" name="oid" value="<?php echo $orgOid ?>">
    <input type="hidden" name="okurl" value="<?php echo $orgOkUrl ?>">
    <input type="hidden" name="failUrl" value="<?php echo $orgFailUrl ?>">
    <input type="hidden" name="TranType" value="<?php echo
$orgTransactionType ?>">
    <input type="hidden" name="Instalment" value="<?php echo $orgInstallment
?>">

    <input type="hidden" name="currency" value="<?php echo $orgCurrency ?>">
    <input type="hidden" name="rnd" value="<?php echo $orgRnd ?>">
    <input type="hidden" name="hash" value="<?php echo $hash ?>">
    <input type="hidden" name="storetype" value="3D_PAY">
    <input type="hidden" name="hashAlgorithm" value="ver2">
    <input type="hidden" name="lang" value="tr">

    </form>

</center>

</body>

</html>

```

5.4.2 Response Code Sample

```

<html>
<head>
<title>3D</title>
<meta http-equiv="Content-Language" content="tr">
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-9">
<meta http-equiv="Pragma" content="no-cache">
<meta http-equiv="Expires" content="now">

```

```

</head>
<body>
<h1>3D Payment Page</h1>
<h3>Payment Response</h3>
<table border="1">

<?php
    $originalClientId = "xxxxxx";
    $mustParameters = array("clientid","oid","Response");
    $isValid = true;
    for($i=0;$i<3;$i++)
    {
        if($_POST[$mustParameters[$i]] == null || $_POST[$mustParameters[$i]] == "" )
        {
            if($mustParameters[$i] == "oid"){
                if($_POST["ReturnOid"] == null || $_POST["ReturnOid"] == "" ){
                    $isValid = false;
                    echo "<tr><td>Missing Required Param</td>"+<td>oid / ReturnOid</td></tr>";
                }
            }else{
                $isValid = false;
                echo "<tr><td>Missing Required Param</td>"+<td>"+
                $mustParameters[$i]+</td></tr>";
            }
        }
    }
    if($_POST["clientid"] != $originalClientId){
        echo "<h4>Security Alert. Incorrect Client Id.</h4>";
        return;
    }

    if(!$isValid){
        echo "<h4>Security Alert. The digital signature is not valid. Required Paramaters are
        missing.</h4>";
        return;
    } else {

?>

<tr>
    <td><b>Parameter Name</b></td>
    <td><b>Parameter Value</b></td>
</tr>

```

```

<?php
    $paymentparams =
    array("AuthCode","Response","HostRefNum","ProcReturnCode","TransId","ErrMsg");
    foreach($_POST as $key => $value)
    {
        $check=1;
        for($i=0;$i<6;$i++)
        {
            if($key == $paymentparams[$i])
            {
                $check=0;
                break;
            }
        }
        if($check == 1)
        {
            echo "<tr><td>".$key."</td><td>".$value."</td></tr>";
        }
    }
?>
</table>
<br>
<br>
<?php
    $hashparams = $_POST["HASHPARAMS"];
    $hashparamsval = $_POST["HASHPARAMSVAL"];
    $hashparam = $_POST["HASH"];
    $storekey="xxxxxx";
    $paramsval="";
    $index1=0;
    $index2=0;
    $escapedStoreKey = "";

    if ($_POST["hashAlgorithm"] == "ver2"){
        $parsedHashParams = explode("|", $hashparams);
        foreach ($parsedHashParams as $parsedHashParam) {
            $vl = $_POST[$parsedHashParam];
            if($vl == null)
                $vl = "";

            $escapedValue = str_replace("\\", "\\\\", $vl);
            $escapedValue = str_replace("|", "\\|", $escapedValue);
        }
    }

```

```

        $paramsval = $paramsval . $escapedValue . "|";
    }
    $escapedStoreKey = str_replace("|", "\\|", str_replace("\\", "\\\\", $storekey));
    $hashval = $paramsval.$escapedStoreKey;
    $hash = base64_encode(pack('H*',hash('sha512', $hashval)));
} else {
    while($index1 < strlen($hashparams))
    {
        $index2 = strpos($hashparams,":",$index1);
        $vl = $_POST[substr($hashparams,$index1,$index2- $index1)];
        if($vl == null)
            $vl = "";
        $paramsval = $paramsval . $vl;
        $index1 = $index2 + 1;
    }
    $escapedStoreKey = $storeKey;
    $hashval = $paramsval.$escapedStoreKey;
    $hash = base64_encode(pack('H*',sha1($hashval)));
}
$hashparamsval = $hashparamsval . "|". $escapedStoreKey;

if($hashval != $hashparamsval || $hashparam != $hash) {
    echo "<h4>Security Alert. The digital signature is not valid.</h4>" . " <br />\r\n";
    echo "Generated Hash Value : ". $hashval . " <br />\r\n";
    echo "Sent hash value : " . $hashparamsval. " <br />\r\n";
    echo "Generated Hash : ". $hash . " <br />\r\n";
    echo "Sent hash : " . $hashparam. " <br />\r\n";
}

```

```

$mdStatus = $_POST["mdStatus"];
$ErrMsg = $_POST["ErrMsg"];
if($mdStatus == 1 || $mdStatus == 2 || $mdStatus == 3 || $mdStatus == 4)
{
    echo "<h5>3D Transaction is Success</h5><br/>";
}

```

?>

```

<h3>Payment Response</h3>
<table border="1">
    <tr>
        <td><b>Parameter Name</b></td>
        <td><b>Parameter Value</b></td>
    </tr>

```

```
<?php

for($i=0;$i<6;$i++)
{
    $param = $paymentparams[$i];
    echo "<tr><td>".$param."</td><td>".$_POST[$param]."</td></tr>";

}

?>

</table>

<?php
    $response = $_POST["Response"];
    if($response == "Approved")
    {
        echo "Payment Process is Successfull.";
    }
    else
    {
        echo "Transaction is not Success. Error = ".$ErrMsg;
    }

}

else
{
    echo "<h5>3D Transaction is not Success</h5>";
}

}

?>

</body>
</html>
```

6.APPENDIX A: Gateway Parameters

6.1 Mandatory Input Parameters

Parameter	Description	Format
clientid	Merchant ID	Maximum 15 characters
storetype	Merchant payment model	Values is "3D_PAY"
trantype	Transaction type	Set to "Auth" for

		authorization, "PreAuth" for preauthorization
amount	amount transaction amount	Use "." or "," as decimal separator, do not use grouping character
currency	ISO 3 digit code for transaction currency	3 characters (example: 949 for TL)
oid	Unique identifier of the order	Maximum 64 characters
Pan	Card number	Maximum 20 digits
Cv2	Cv2 value	3 or 4 digits
Ecom_Payment_Card_ExpDate_Year	Card expiry year	4 digits
Ecom_Payment_Card_ExpDate_Month	Card expiry month	2 digits
okUrl	The return URL to which NestPay redirects the customer if transaction is completed successfully.	Example: http://www.test.com/ok.php
failUrl	The return URL to which NestPay redirects the customer if transaction is completed unsuccessfully.	Example: http://www.test.com/fail.php
lang	Language of the payment pages hosted by NestPay	"tr" for Turkish, "en" for English
rnd	Random string, will be used for hash comparison	Fixed length, 20 characters
hash	"Hash value for client authentication	
hashAlgorithm	Hash version	"Ver2"

6.2 Optional Input Parameters

Parameter	Description	Format
refreshtime	Redirection counter value (to okUrl or failUrl) in seconds.	
encoding	Encoding of the posted data. Default value is "utf-8" if not sent	Maximum 32 characters
description	description	Maximum 255 characters

comments	Kept as "description" for the transaction	Maximum 32 characters
instalment	Instalment count	Maximum 255 characters
GRACEPERIOD	Grace period, postpones the payment for given months	Number (months)
email	Customer's email address	Maximum 64 characters
tel	Customer phone	Maximum 32 characters
BillToCompany	BillTo company name	Maximum 255 characters
BillToName	BillTo name/surname	Maximum 255 characters
BillToStreet1	BillTo address line 1	Maximum 255 characters
BillToStreet2	BillTo address line 2	Maximum 255 characters
BillToCity	BillTo city	Maximum 64 characters
BillToStateProv	BillTo state/province	Maximum 32 characters
BillToPostalCode	BillTo postal code	Maximum 32 characters
BillToCountry	BillTo country code	Maximum 3 characters
ShipToCompany	ShipTo company	Maximum 255 characters
ShipToName	ShipTo name	Maximum 255 characters
ShipToStreet1	ShipTo address line 1	Maximum 255 characters
ShipToStreet2	ShipTo address line 2	Maximum 255 characters
ShipToCity	ShipTo city	Maximum 64 characters
ShipToStateProv	ShipTo state/province	Maximum 32 characters
ShipToPostalCode	ShipTo postal code	Maximum 32 characters
ShipToCountry	ShipTo country code	Maximum 3 characters
idl	Id of item #l, required for item #l	Maximum 128 characters
itemnumberl	Item number of item #l	Maximum 128 characters
productcodeI	Product code of item #l	Maximum 64 characters
qtyl	Quantity of item #l	Maximum 32 characters
descl	Description of item #l	Maximum 128 characters
pricel	Price of item #l	Maximum 32 characters
amount	Subtotal of item #l	Maximum 32 characters
RecurringPaymentNumber	Total number of payments for recurring payment	Number
RecurringFrequencyUnit	Frequency unit for recurring payment	1 char: D=Day,W=Week,M=Month,Y=Year
RecurringFrequency	Frequency of recurring payment	Number

printBillTo	Print BillTo address fields on payment page	"true" or "false"
printShipTo	Print ShipTo address fields on payment page	"true" or "false"

6.3 Transaction Response Parameters

Parameter	Description	Format
AuthCode	Transaction Verification/Approval/Authorization code	6 characters
xid	Internet transaction identifier	28 characters, base64 encoded
Response	Payment status	Possible values: "Approved", "Error", "Declined"
HostRefNum	Host reference number	12 characters
ProcReturnCode	Transaction status code	Alphanumeric, 2 chars, "00" for authorized transactions, "99" for gateway errors, others for ISO-8583 error codes
TransId	NestPay Transaction Id	Maximum 64 characters
ErrMsg	Error message	Maximum 255 characters
ClientIp	IP address of the customer	Maximum 15 characters formatted as "###.###.###.###"
ReturnOid	Returned order ID, must be same as input orderId	Maximum 64 characters
MaskedPan	Masked credit card number	12 characters, XXXXX***XXX
EXTRA.TRXDATE	Transaction Date	17 characters, formatted as "yyyyMMdd HH:mm:ss"
rnd	Random string, will be used for hash comparison	Fixed length, 20 characters
HASHPARAMS	Contains the field names used for hash calculation. Field names are appended with ":" character	Possible values "clientid:oid:AuthCode:ProcReturnCode:Response:rnd:" for non-3D transactions, "clientId:oid:AuthCode:ProcReturnCode:Response:mdStatus:cavv:"

		eci:md:rnd:" for 3D transactions
HASHPARAMSVAL	Contains the appended field values for hash calculation. Field values should be appended with the same order in HASHPARAMS field	Fixed length, 28 characters
HASH	Hash value of HASHPARAMSVAL and merchant password field	Fixed length, 20 characters

6.4 MPI Response Parameters

Parameter	Description	Format
mdStatus	Status code for the 3D transaction	1=authenticated transaction 2, 3, 4 = Card not participating or attempt 5,6,7,8 = Authentication not available or system error 0 = Authentication failed
merchantID	MPI merchant ID	15 characters
txstatus	3D status for archival	Possible values "A", "N", "Y"
iReqCode	Code provided by ACS indicating data that is formatted correctly, but which invalidates the request. This element is included when business processing cannot be performed for some reason.	2 digits, numeric
iReqDetail	May identify the specific data elements that caused the Invalid Request Code (so never supplied if Invalid Request Code is omitted).	
vendorCode	Error message describing <i>iReqDetail</i> error.	
PAResSyntaxOK	If PARes validation is syntactically correct, the value is true. Otherwise value is false.	"Y" or "N"

ParesVerified	If signature validation of the return message is successful, the value is true. If PAREs message is not received or signature validation fails, the value is false.	"Y" or "N"
eci	Electronic Commerce Indicator	2 digits, empty for non-3D transactions
cavv	Cardholder Authentication Verification Value, determined by ACS.	28 characters, contains a 20 byte value that has been Base64 encoded, giving a 28 byte result.
xid	Unique internet transaction ID	28 characters, base64 encoded
cavvAlgorithm	CAVV algorithm	Possible values "0", "1", "2", "3"
md	MPI data replacing card number	Alpha-numeric
Version	MPI version information	3 characters (like "2.0")
sID	Schema ID	"1" for Visa, "2" for Mastercard
MdErrorMsg	Error Message from MPI (if any)	Maximum 512 characters