eManager user manual

Chapter 1. Introduction

1.1. General procedure

There are two requirements needed to fulfill, before the testing and development can start:

- Merchant has signed an e-commerce contract with the acquirer bank
- You have received an e-mail with the test account details. Nets Estonia creates a test account after receiving information from merchants acquirer bank and sends the account details to the contact person mentioned in the contract.

The iPay (Nets Estonia e-commerce solution) test environment is a copy of the live environment. Generally you need to implement a payment request with two data flows to iPay server:

Payment authorization request:
- Payment request to the iPay server. (Initiator your website, receiver "iPay server").
- Payment verification, this is handled by your web service. (Initiator "iPay servlet", receiver your website)

After the payment is implemented on the merchant side next steps are taken:

Merchant/developer ensures that the required functionality is obtained and the system reacts adequately to different payment scenarios.

Merchant/developer provides instructions by sending an e-mail to webpos@estcard.ee, on how can Nets Estonia AS perform test payments on the ready solution without causing negative impact to the merchant.

After the tests are successful, Nets Estonia submits a Live environment access request to the bank and after confirmation, the Live account is activated and its details are sent to the contact person mentioned in the contract.

1.2. Technical overview

Nets Estonia e-commerce gateway (iPay) works with http queries. To request an authorization form, customer is redirected from merchant’s e-shop to iPay with simple hidden html form.
iPay handles actual authorization and card data, ensuring that merchant never has to worry about securing customers card data, it is taken out of merchants domain and never revealed to merchant, making the whole process more secure.
After customer has completed payment, customer is redirected back to the merchants web store with information about the success of transaction.
For merchant iPay also provides online environment to view and search past transactions, making life simpler for merchant.
Regular reports over e-mails are also available. +

Chapter 2. Protocol specifications

2.1. Protocol version 004

Protocol version 004 is currently our default protocol.
Without extensions it provides simple authorization services.

2.1.1. Requesting authorization form

To request authorization page from Nets Estonia e-commerce gateway a simple hidden html form needs to be generated within a merchants webpage.

Table 1. Fields for authorization form request

Nr	Availability	Mac Value	Request Parameter	Definition	Format	Example value
1	M	N	lang	Language: ISO 639-1	char..2	en
2	M	N	action	Action ID: gaf	char..3	gaf
3	M	Y	ver	Protocol version (4)	int..3	004
4	M	Y	id	Merchant ID	char..10	12ABCD1223
5	M	Y	ecuno	Transactions unique ID. Format: date[YYYYMM] + random number(in range): 100000-999999 [1: Unique identifier that is in both request and response and which connects authorization request with authorization response. This needs to be unique within 24 hours. When transaction is declined or cancelled and customer wishes to try the same payment again this value needs to be renewed.]	int..12	201610280012
6	M	Y	eamount	Payment amounnt in cents	int..12	000000001234

Nr	Availability	Mac Value	Request Parameter	Definition	Format	Example value
7	M	Y	cur	Currency ISO-4217	char..3	EUR
8	M	Y	datetime	Timestamp. Format: [YYYYMMDDhhmmss] ISO-8601	int..14	20161028112930
9	M	N	charEncoding	Encoding	char	UTF-8
10	M	Y	FeedBackUrl	Feedback URL given by merchant	char..128	https://merchant.site/feedback
11	M	Y	delivery	1. position: Delivery description (Mandatory), **2. position: Pre-Authorization "P" (Optional. Usable only for pre-auhorization)** (S|T)(P)	char..2	First position S or T, second position P S (electronic delivery of product), T (physical delivery of product), P (preorder)
12	O	Y	additionalinfo	Additional information [2: Information that is displayed in merchant's report view. Also searchable. Good place to store some relevant data about the transaction. For example ticket number or client id or booking id. Can be up to 128 characters in total length. Key is separated from value using ":". key:value pairs are separated with ";"]	char..128 key:value;	Cust_id:121212;b_id:10001;100:ABC123;101:kala;001:jama;
13	M	-	mac	Digital Signature	hexadecimal	4d5e875a245d42.....

Posted data should be unpadded. However in MAC calculation all fields are padded to their maximum length. Ascii (char) fields are padded with trailing spaces and numeric fields are padded with leading zeroes.

2.1.1.1. Example data

action=gaf
lang=et
ver=004
id=8AEF7DBEF3
ecuno=784071201802
eamount=000000001100
cur=EUR
datetime=20180213113049
charEncoding=UTF-8
feedBackUrl=https://merchant.site/ecom-check.php
delivery=S
additionalinfo=100:ABC123;101:kala;001:jama;
(mac=9f7eab41d650.....) +

2.1.1.2. Example mac calculation

For mac calculation data fields will be padded to their max length and concatenated to create a single string. Order of the fields is important.

RSA with SHA1 (SHA1withRSA) [3: SHA1 will be deprecated in favour of sha256, however, currently SHA1 is used. New implementations should be capable of using sha256 if the need arises.]
MAC=RSA(prikey,
SHA1(version+merchant_id+ecuno+amount+currency+datetime+feedbackurl+delivery+additional_info))

Example input string for mac calculation from above example data: +

```
"0048AEF7DBEF3784071201802000000001100EUR20180213113049https://merchant.site/ecom-
check.php
S100:ABC123;101:kala;001:jama;
"
```

2.1.1.3. Example MAC value

For fast checking of validity of your mac calculation, you can use example mac value, calculated with mytestprivat.key.
Using input's from example below, MAC should be identical to the sample mac value below.

```
        Original, unpadded data
              ver : 004
               id : 8AEF7DBEF3
            ecuno : 531602201805
          eamount : 1100
         currency : EUR
         datetime : 20180524130931
      feedbackurl : https://feed.back.to/ecom/ecom-check.php
         delivery : S
  additional_info : 100:ABC123;101:kala;001:jama;
         === Calculated MAC ===
bb1635bd193d6fb04c075897af6ae72985a2f0d81c7f456167970d443d140dcadc93b3632e11475c182553
2300156c6f33467459c218fa16b9f9a9d94cda1c043168f708098b2170a8fbb9fcd4b88bc579334b55554a
f4202e88bb6796ff5c5698182169a53d5ef60c14f8f33292d52110da317f5198e26b624d066aab15525c
```

PHP example

In PHP MAC would be calculated as follows:

```
#
# Prepare key
#
$fp = fopen("$key", "r");
$fs = filesize("$key");
$priv_key = fread($fp, $fs);
fclose($fp);
$pkeyid = openssl_get_privatekey($priv_key);
#
# concatenate data for mac calculation
#
$data = $ver . $id . $ecuno . $eamount . $cur . $datetime . $feedbackurl . $delivery .
$addinfo_pad;
#
# calculate sha1 signature and sign it
#
$signature=sha1($data);
openssl_sign($data, $signature, openssl_get_privatekey($priv_key));
#
# As this provides binary signature, the
# signature needs to be converted into hex.
#
$mac=bin2hex($signature);
```

2.1.1.4. Example html form

All of the above put together, for requesting authorization page would look like this:

```
<form  name='form' action="https://test.estcard.ee/ecom/iPayServlet" method="post">
    <input type="submit" value="To payment page">
    <input type="hidden" name="lang" value="en">
    <input type="hidden" name="action" value="gaf">
    <input type="hidden" name="ver" value="004">
    <input type="hidden" name="id" value="8AEF7DBEF3">
    <input type="hidden" name="ecuno" value="784071201802">
    <input type="hidden" name="eamount" value="1000">
    <input type="hidden" name="cur" value="EUR">
    <input type="hidden" name="datetime" value="20180213113049">
    <input type="hidden" name="charEncoding" value="UTF-8">
    <input type="hidden" name="feedBackUrl" value="https://merchant.site/ecom-
check.php">
    <input type="hidden" name="delivery" value="S">
    <input type="hidden" name="additionalinfo" value="100:ABC123;101:kala;001:jama;">
    <input type="hidden" name="mac"
value="9f7eab41d650518e3156c1b96f8cb0738e2624f4e5b51bb3ea2081d531b4e4ed0e9fc4a992798db
da8d783899385190f133ce2b972d993740a14f66a8b5df44dae6af0a9a5a115c9745d9b94a8d41788c8aed
28987960a734bccc0373a19b9314f9dbf0a003b46207ffaee317d09811de5fe591507fb408c62b2ce238f9
79c5f">
</form>
```

2.1.2. Response

Response is calculated the same way as request, only it is done by iPay and Merchant’s webpage has to check validity of the answer.

RSA with SHA1 (SHA1withRSA). [3: SHA1 will be deprecated in favour of sha256, however, currently SHA1 is used. New implementations should be capable of using sha256 if the need arises.]
MAC=RSA(prikey,
SHA1(ver+id+ecuno+receipt_no+eamount+cur+respcode+datetime+msgdata+actiontext)) +

Table 2. Resonse request

Nr	Availability	Mac Value	Request Parameter	Definition	Format	Example value
1	M	Y	Action	Action ID: afb	Char..3	afb
2	M	Y	ver	Protocol version (4)	int..3	004
3	M	Y	id	Merchant ID	char..10	12ABCD1223
4	M	Y	ecuno	Transactions unique ID. Format: date[YYYYMM] + random number(in range): 100000-999999	int..12	201610280012
5	M	Y	receipt_no	Receipt number	int..6	000015
6	M	Y	eamount	Payment amounnt in cents	int..12	000000001234

Nr	Availability	Mac Value	Request Parameter	Definition	Format	Example value
7	M	Y	cur	Currency ISO-4217	char..3	EUR
8	M	Y	respcode	Response code. NETS payment system messaging standard table 39. Action codes	char..3	000
10	M	Y	datetime	Timestamp. Format: [YYYYMMDDhhmmss] ISO-8601	int..14	20161028112930
10	M	Y	msgdata	Payment description, cardholder name, etc.	char..40	
11	M	Y	actiontext	Description of response code	char..40	OK, approved
12	M	N	auto	Y - automatic feedback N - feedback via browser	char 1	OK, approved
13	M	-	mac	Digital signature	hexadecimal	4d5e875a245d42…..

2.1.2.1. Example feedback data

```
action=afb
ver=4
id=8AEF7DBEF3
ecuno=556173201802
receipt_no=02973
eamount=1100
cur=EUR
respcode=000
datetime=20180215115522
msgdata=Cardholder Name
actiontext=OK, tehing autoriseeritud
mac=6EE6B987374E5DE0FAAD9ABB0DEB3ABA52E1CA4C715D6B67D7AD50D59913A09BCD69475C71F29D99C0
7D9F1D578E4452E2A427C767B7DDDF4F06B197E071FC9621A11B94596BF27764D69D22FED06A28AA72535A
80ACA3238A3A0D82C7CE543A13B5C1AB17CB662CF2F5BAF535E58018B10C73F6FE36D947104B0F79FBB8DC
81
charEncoding=UTF-8
auto=N
```

PHP example

Please note that msgdata field might contain multibyte characters and threfore multibyte safe operations are needed. PHP’s sprintf is NOT multibyte safe. Custom function mb_sprintf could be used instead

```
#
# Concatenate all the fields together with maximum length
# for mac calculation
#
$data = sprintf("%03s", $ver) . sprintf("%-10s", "$id") .
sprintf("%012s", $ecuno) . sprintf("%06s", $receipt_no) . sprintf("%012s",
$eamount) . sprintf("%3s", $cur) . $respcode . $datetime . mb_sprintf("%-40s",
$msgdata) . mb_sprintf("%-40s", $actiontext);
#
# Load certificate
#
 function hex2str($hex) {
       $str = "";
       for($i=0;$i<strlen($hex);$i+=2)
       $str.=chr(hexdec(substr($hex,$i,2)));
 return $str;
}
$mac = hex2str($mac);
$key = nets_estonia_pub_key
$fp = fopen("$key", "r");
$fs = filesize("$key");
$pub_key = fread($fp, $fs);
fclose($fp);
$pubkeyid = openssl_get_publickey($pub_key);
$result = openssl_verify($data, $mac, $pubkeyid);
if ($result == 1) {
  echo "Signature check OK<br>";
} elseif ($result == 0) {
  echo "Signature NOT OK<br>";
} else {
  echo "error checking signature<br>";
}
#
# unload key from memory
#
openssl_free_key($pubkeyid);
```