

Tool Paygate-Test

Introduction

Computop Paygate uses TLS encryption to ensure secure transmission of request and response data as well as for notifications sent from Paygate to your system.

Therefore request and response is encrypted either with Blowfish (default) or with AES encryption (upon request at [Computop Helpdesk](#)). The blowfish / AES-password is provided after Computop completed setup of your MerchantId.

Additionally sensitive data like MerchantId, amount and currency are hashed with a separate HMAC-password using SHA256-algorithm.

To check your implementation we [provide a tool](#)

- to validate MAC-value (API-parameter [MAC](#))
- to validate Len/Data value (API-parameter Len and Data)

You will find a short description how to use this tool.

Usage

The tool itself will start like this:

Encryption **A**

Decryption **B**

MAC calculation **C**

Base64-encoding **D**

Base64-decoding **E**

Base64 Encoder (extern) **F**

Base64 Decoder (extern) **G**

Test Paygate Encryption

Data are not stored on any server. Data are only stored in local cookies for your own convenience.

A short documentation of this tool can be found here: [Paygate-Test](#).
Do **not** send any real credit card numbers here! **Please use these test data**. For other test data refer to our [Test Guide](#).

Credit card with 3-D Secure 2.x:

- Parameter "MsgVer=2.0" is required to indicate that your implementation supports 3-D Secure processing.
- Some values (e.g. address, browser, credential on file (CoF), ...) must be sent in as KVP-JSON ([see our Dev-Docu](#)).
- Your MerchantID has to be setup for 3-D Secure 2.x.

Please replace 'Generic3DSTest' by your MerchantID and also use Encryption-Password assigned by Computop.

MerchantID **1** your MerchantID (mandatory) [Put your MerchantID](#) [Put Generic Test MerchantID](#)

EncryptionPassword **2** your EncryptionPassword (mandatory) [Blowfish/ECB](#)

HMacPassword **3** your HMacPassword (optional)

Plain Request **10**
Enter your parameter plain string (unencrypted); this input will be encrypted.

Request will be encrypted with your Encryption-Password.

[Set simple Call with MsgVer=2.0](#) **4**

[Set simple Payment Call](#) **5**

[Add response=encrypt](#) **6**

[Add URLs](#) **7**

[Add Simulation \(OrderDesc=test:0000\)](#) **8**

[Add CoF \(CIT/initial=true\)](#) **9**

[Add CoF \(MIT/initial=false\)](#)

[Set Capture Call](#)

Additional parameters (e.g. template) may be added plain. See next page.

Unencrypted **14**
This part will not be encrypted.

Data will not be encrypted.

[Set template=ct_responsive \(ONLY payssl.aspx\)](#) **11**

[Set React templates HPP, CC, SDD \(ONLY paymentpage.aspx\)](#)

[Add CustomField1..9 \(React templates\)](#) **12**

[Add language=en](#) [Add language=de](#) **13**

[Add language=fr](#) [Add language=es](#)

Encrypt **15**

Building an encrypted payment request

If you already have a Computop MerchantId, encryption-password and HMAC-password, you can enter them into fields (1), (2) and (3). You may also choose Blowfish or AES encryption, Blowfish is default and AES needs to be enabled by [Computop Helpdesk](#).

Then you start creating a basic request

- for creditcard requests using 3-D Secure 2.x with button "Set simple Call with MsgVer=2.0" (4)
- or simple payment calls (e.g. PayPal) with button "Set simple Payment Call" (5)

If you want an encrypted Paygate-response pls. add parameter "response=encrypt" with button "Add response=encrypt" (6).

If you want to use payment methods with forms or redirect you have to provide URLs, too. Some sample URLs can be added with button "" (8).

To enable simulation mode (i.e. no downstream-systems are required) pls. use button "Add Simulation (OrderDesc:0000)" (8). By using this option you can [simulate](#) all [response codes](#) just by replacing "0000" with your desired one.

To add additional parameters e.g. for recurring payments which are customer or merchant initiated Paygate supports [credential on file](#) which can be added with button "CoF (CIT/initial=true)" (i.e. customer initiated, initial payment) or button "CoF (MIT/initial=false)" (i.e. merchant initiated, subsequent payment). These key/values are sent as [base64](#)-encoded JSON-values. A list of [JSON-objects](#) can be found here.

After putting a basic payment request together you may modify e.g. amount from 123 (i.e. 1,23) into another value or change the currency from EUR (i.e. USD) - depending on your paymethod setup.

 if you want to use your own MerchantId pls. use the button "Put your MerchantID" and the generic test MerchantId will be replaced your yours.

 you are now basically ready to go and start encrypting your request with button "Encrypt" (15).

Adding unencrypted parameters to payment request

Paygate also supports [payment forms](#) - so the consumer can select a payment method or directly enter [credit card](#) or [bank account](#) data.

These payment forms use unencrypted data to

- modify their background or font color
- or display a merchant logo - depending on the template
- or select the template you want to use

Adding additional JSON parameters (base64-encoded)

Mostly for credit card payments additional parameters like e.g. [browser information](#) or [external 3-D Secure](#) data may be used. These are base64-encoded and then added to the field (10) for "Plain Request".

Therefore form "Base64-encoding" (D) can be used which already provides some samples in Paygate-[JSON format](#).

Base64-encoded values are also sent as key-value-pair:

- e.g.: credentialOnFile=base64({'type': { "unscheduled": "CIT" }, "initialPayment": false })
- will be sent as:
credentialOnFile=ew0KICAgInR5cGUiOiB7DQogICAgICAgICJ1bnNjaGVkdWxIZCI6ICJDSVQiDQogICAgfSwNCiAgICAiaW5pdGlhbFBheW1lbnQiOiBmYWxzZQ0KfQ==

Using predefined MerchantId Generic3DSTest

You can simply try and use a Paygate predefined test MerchantId "Generic3DSTest". You don't need to know Blowfish- and HMAC password, because the tool is already prepared to use it. Just keep the dummy value "set_automatically" in place.

Encrypt and send request

After you have built your plain (unencrypted) payment request and eventually added some template parameters by using:

- button "Set simple Call with MsgVer=2.0"
- button "Add response=encrypt"
- button "Add URLs"
- button "Add Simulation (OrderDesc=Test:0000)"
- button "Set React templates HPP, CC, SDD"

Please replace 'Generic3DSTest' by your MerchantID and also use Encryption-Password assigned by Computop.

MerchantID

[Put your MerchantID](#)

[Put Generic Test MerchantID](#)

1

Your MerchantID assigned by Computop.

EncryptionPassword

Your EncryptionPassword assigned by Computop.

HMacPassword

HMacPassword assigned by Computop.
If you enter this password,
the MAC value is calculated.

Plain Request

Request will be
encrypted with
your Encryption-
Password.

MsgVer=2.0&MerchantID=Generic3DSTest&TransID=YourTransId&RefNr=Ref123&Amount=123&
Currency=EUR&response=encrypt&URLBack=https://computop.com/paygate-test&
URLSuccess=https://computop.com/developer/blowfish-test/success.php&
URLFailure=https://computop.com/developer/blowfish-test/failure.php&
URLNotify=https://computop.com/developer/blowfish-test/notify.php&
OrderDesc=test:0000

[Set simple Call with MsgVer=2.0](#)

2

[Set simple Payment Call](#)

[Add response=encrypt](#)

3

[Add URLs](#)

4

[Add Simulation \(OrderDesc=test:0000\)](#)

5

[Add CoF \(CIT/Initial=true\)](#)

[Add CoF \(MIT/Initial=false\)](#)

[Set Capture Call](#)

Additional parameters (e.g. template) may be added plain. See next page.

Unencrypted

Data will not be
encrypted.

&Template=PaymentPageDropDown_v1&CCTemplate=Cards_v1&SDDTemplate=DirectDebit_v1&
language=en

[Set template=ct_responsive \(ONLY paypal.aspx\)](#)

[Set React templates HPP, CC, SDD \(ONLY
paymentpage.aspx\)](#)

6

[Add CustomField1..9 \(React templates\)](#)

[Add language=en](#)

[Add language=de](#)

[Add language=fr](#)

[Add language=es](#)

[Encrypt](#)

7

you are ready to go.

After pushing button "Encrypt" (7) the payment request is built and encrypted and shown on the next form:

Plain data (before encryption)

PlainText: MsgVer=2.0&MerchantID=Generic3DSTest&TransID=YourTransId&RefNr=Ref123&Amount=123&Currency=EUR&response=encrypt&URLBack=https://computop.com/paygate-test&URLSuccess=https://computop.com/developer/blowfish-test/success.php&URLFailure=https://computop.com/developer/blowfish-test/failure.php&URLNotify=https://computop.com/developer/blowfish-test/notify.php&OrderDesc=test:0000&MAC=ef01dbde5a5e5fd0a5bbec50ff3c9a453f49a7aa3d971f2885bb7d68790cf87e

MerchantID: Generic3DSTest

Len: 443

1

2

3

Encrypted data

Data: 43d86023d0e7e8f9e04c948e50eae3b84d485eec55a9cf4acb2394963d7fe6673b9de03f551ae49284379b3f4cf7bc2369581eec52ca3e2bbded1ec985fe8902bf74eaa702965855158d4a9ec262788b8a1f1f6c2ad08a79d7588f9916cf8790fdd5b24684b87711c9d3e1c6124aeedb2022af51ce6899961990d33e4b853227c39f0ebe5b0b733b588ae74811c5a9e8ebc47e0e61e82dfbc553701acfe4ec635f95cbda7b0498ed505cf6a1863c1fefdaf4aac69e578ffa6381fbb074dac732e9acf9379db430f9e932f128d1b6d02e018da75308af0ccfa146787fd915d56b1dad61837f5564c1f01674f96372b115ce300300fc9d5df4cdcbd82a8447f3881d5e7a60f5b2642a3818015c11c681dca610e3a0e444234c3ef54f912b0d348a64fd02de3eb9965626094e615718e1b4dbb4d7e4e43338ccd95d6be6e5ae299079806fc82cfe4a4e2170ddad213472f8a63f87bf72b408850792d2bdd670fd361401ea58541897dbb0c648dd99d1ba8f05822d75e6410a2f7f49a3b9a5cecc4c1c1285842eb8fee78296020b637a4ac6d9581cd96b70bf3f13277fee3de679549048a76c46c68a91ab2e93da963bcd88434bfd0f4e44708bb3c9d99140a0

4

These data will be added without encryption

Unencrypted: &Template=PaymentPageDropDown_v1&CCTemplate=Cards_v1&SDDTemplate=DirectDebit_v1&language=en

5

MAC calculated for request values - if HMAC-Password was given

PayId:

TransId: YourTransId

MerchantID: Generic3DSTest

Amount: 123

Currency: EUR

MAC: ef01dbde5a5e5fd0a5bbec50ff3c9a453f49a7aa3d971f2885bb7d68790cf87e

6

7

Common tips

Note: Your MerchantID is automatically added as a plain key-value-pair (i.e.: MerchantID=Generic3DSTest) to the request.

Note: If you get "Unexpected exception" please:

- check your MerchantId and Blowfish-Password/HMAC-Password
- doublecheck your template name - if specified
- ensure that you've provided URLFailure, URLSuccess, URLNotify.

Some Payment API-Calls - just try it out

8

9

Hosted Payment Page (HPP)	https://computop-paygate.com/paymentpage.aspx?MerchantID=Generic3DSTest&Len=443&Data=[encrypted Data]&Template=PaymentPageDropDown_v1&CCTemplate=Cards_v1&SDDTemplate=DirectDebit_v1&language=en	Call HPP	QR-Code
Credit Card Form (PaySSL)	https://computop-paygate.com/payssl.aspx?MerchantID=Generic3DSTest&Len=443&Data=[encrypted Data]&Template=PaymentPageDropDown_v1&CCTemplate=Cards_v1&SDDTemplate=DirectDebit_v1&language=en	Call PaySSL	QR-Code

Here you will see:

- (1) plain request data which will be encrypted into Len + Data
- (2) your MerchantId - or the Paygate default one
- (3) value for parameter "Len" of encrypted request
- (4) value for parameter "Data" of encrypted request
- (5) additional request data which will not be encrypted
- (6) values that are used for HMAC-calculation
- (7) the calculated HMAC-value itself
- (8) finally a button to initiate the payment request
- (9) or a button to show a QR-code which can be scanned by your smartphone to initiate a payment request there

i pls. note that the unencrypted parameters are used for templates with payment forms. The template name (here: "Template=PaymentPageDropDown_v1") needs to match the Paygate endpoint. So we have to call here "PaymentPage.aspx" (short "HPP"), "PaySSL.aspx" won't work, will cause a technical exception and the form won't show up.

Samples