

# **Payment Page Security Features Specification**

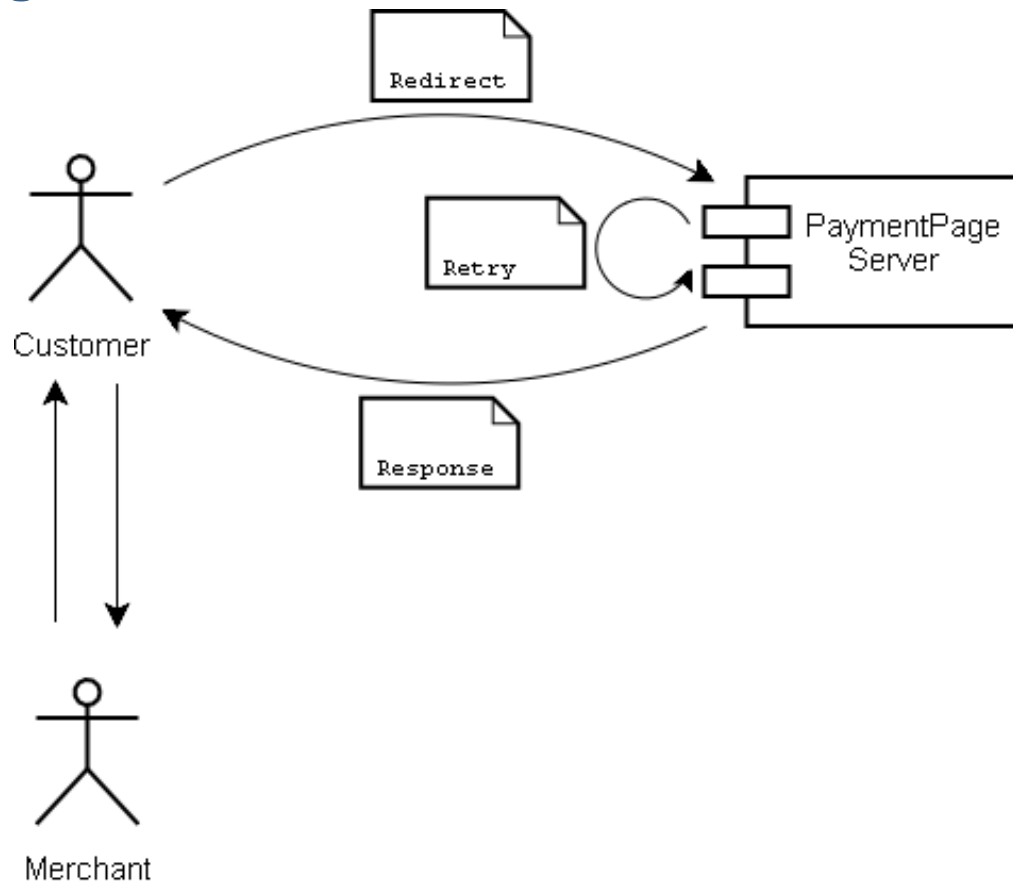
## Contents

Introduction .....	3
Messages .....	4
Payment Page Request .....	4
Payment Page Response .....	4
Message security .....	5
Payment Page Request/Response Envelope .....	5
Retry Data.....	6
Message content .....	6
URL parameters.....	6
Appendixes .....	7
Appendix A: Request/Response Envelope – version 6 .....	7

## Introduction

This specification describes the security features applied on sensitive data managed and transmitted by the Payment Page.

## Messages



### Payment Page Request

This is the first message, created by the merchant in order to initiate a payment. It mainly contains information about the merchant and the order. It is important that this message is tamperproof (integrity) and can only be created by the merchant (non-repudiation).

### Payment Page Response

This message contains the result of a payment attempt submitted by the consumer. This will usually contain either OK or USER\_CANCEL, since the consumer is given multiple attempts to enter his/her details. However, on multiple failures or severe transaction failures the response will contain NOK. This message must be tamperproof for integrity and only be valid when originating from DRWP (Digital River World Payments) for non-repudiation.

# Message security

## Payment Page Request/Response Envelope

The Payment Page Request and the Payment Page Response is protected using a digital envelope and a message authentication code (MAC). This is a slimmed down version of a PKCS#7 message, reduced in size due to the limitations of the transmission medium, i.e. being sent as a parameter in a URL. The MAC ensures integrity and non-repudiation, which are the most important aspects.

### Pack envelope

The following is performed by the Payment Page API when creating a redirect URL.


1. Retrieve the serial numbers from the most recent certificate for the sender and the recipient.
2. Compress the payload content using GZIP.
3. Generate a symmetric key (AES-128) to be used by DRWP when unencrypting the request (with AES encryption, CBC cipher mode and PKCS#5 Padding, e.g. AES/CBC/PKCS5Padding).
4. Encrypt the plaintext using the generated symmetric key and the Initialization Vector provided by DRWP (CBC, PKCS#5 padding).
5. Create a message digest of the encrypted data (SHA-1).
6. Encrypt the generated symmetric key with the public key of the recipient (with RSA encryption, ECB cipher mode and PKCS#1 Padding, e.g. RSA/ECB/PKCS1Padding).
7. Encrypt the message digest with the private key of the sender (ECB, PKCS#1 padding).
8. Package the serial numbers, encrypted key, signature and cipher text according to the assigned envelope container layout.
9. The entire envelope container is encoded using Base64url (see <http://www.apps.ietf.org/rfc/rfc4648.html#sec-5>).

### Unpack envelope

The following is performed by the Payment Page API when unpacking the Payment Page response from DRWP.

1. Decode the Base64url envelope.
2. Identify the certificate serial numbers, encrypted key, signature and cipher text according to the assigned envelope container layout.
3. Verify that the serial numbers match local certificates.
4. Verify the signature using the public key of the sender, identified by the serial number of its certificate.
5. Decrypt the key using the private key of the recipient, identified by the serial number of its certificate (with RSA encryption, ECB cipher mode and PKCS#1 Padding, e.g. RSA/ECB/PKCS1Padding).
6. Decrypt the cipher text using the decrypted key and the Initialization Vector provided by DRWP (with AES encryption, CBC cipher mode and PKCS#5 Padding, e.g. AES/CBC/PKCS5Padding).
7. Inflate the content by decompressing the plaintext using GZIP.

See [Appendix A](#) for more info on current envelope container layout.



Since some of the information in this container (e.g. amount and order description) might be revealed on the payment details entry page (depending on template design), transmission of this data should always be performed over a secure channel, i.e. TLS/SSLv3 that meet the intent of strong cryptography (e.g. AES with 128 bits or higher).

## Retry Data

The payment details are encrypted using symmetric encryption. Merchants never create or access this data; it is only used internally by Payment Page. Therefore the keys used to encrypt the payload are only known to DRWP.

## Message content

This information is only of interest for those who are implementing calls to Payment Page themselves, i.e. not using the Java reference implementation provided by DRWP.

All content must support UTF-8 encoded data.

## URL parameters

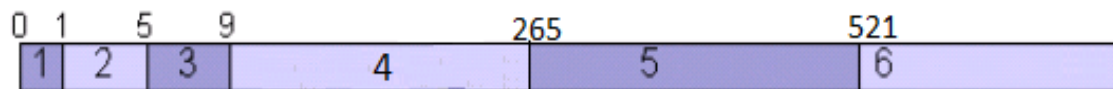
The individual fields sent in the Request and Response are packed as name-value pairs on the form:

`<Name>=<Value>;<Name>=<Value>; etc...`

For an updated and correct list of names that are used for Payment Page request/response parameters, see the Java reference implementation or the Payment Page Integration Guide.

# Appendixes

## Appendix A: Request/Response Envelope – version 6



Field	Size (bytes)	Content
1	1	Envelope version identifier. Always 6.
2	4	Receiver certificate serial number. Unsigned integer. (E.g. {0x0,0x0,0x0,0x1}->1, {0xFF,0xFF,0xFF,0xFF} -> 4294967295)
3	4	Sender certificate serial number. Unsigned integer. (E.g. {0x0,0x0,0x0,0x1}->1, {0xFF,0xFF,0xFF,0xFF} -> 4294967295)
4	256	Generated AES-128 key for payload encryption. Encrypted with RSA-2048 key, ECB mode and PKCS#1 padding.
5	256	Encrypted payload signature. SHA-1 for digest, RSA-2048 with ECB and PKCS#1 padding for encryption.
6	1..n	Encrypted payload padded with PKCS#5. Encrypted with AES-128 key from field 4, using CBC mode. Contact Digital River World Payment for Initialization Vector info.