Manual de Especificaciones Técnicas

MultiPagos Bajío

Contenido del Documento:

1. Objetivo:

Que el cliente que utilizará la funcionalidad de MultiPagos Bajío, cuente con un documento técnico de referencia para programar su solución, el intercambio de información y la implementación seguridad para integrar su desarrollo con la solución de Banco del Bajío.

2. Dirigido a:

Áreas de seguridad lógica (informática), áreas de arquitectura y desarrollo de software que participan en el desarrollo de la solución de MultiPagos Bajío por parte del cliente.

3. Alcance:

Este documento detallará los aspectos relacionados con la interacción entre el cliente y Banco del Bajío, así como el manejo de los mensajes, protocolos, cuestiones de seguridad y las instrucciones para la generación de las llaves asimétricas (RSA), un ejemplo de implementación y uso.

El alcance de los siguientes ejemplos está sujeto a los lenguajes de programación de Java y PHP. En caso de que el cliente requiera los algoritmos para otras plataformas, quedará bajo su responsabilidad el desarrollo y homologación de la funcionalidad correspondiente.

4. Descripción:

4.1. ¿Qué es Multipagos Bajío?

Es un Canal Electrónico de Pagos basado en web que ofrece Banco del Bajío a sus clientes el cual es utilizado para el servicio de cobranza, poniendo a su disposición un cajero virtual en el portal web del cliente.

Los usuarios del cliente de Banco del Bajío podrán realizar sus pagos con cargo a su tarjeta de crédito/débito, a su cuenta de Banco del Bajío o a su cuenta CLABE de cualquier otro banco (aplican restricciones), de manera transparente.

4.2. Esquema operacional:

a) El usuario o contribuyente ingresa al portal del Cliente (p.e. www.miempresa.com/pagos), consulta sus adeudos, selecciona los pagos a realizar y da clic en el botón o liga “Pagar”. Esta liga o botón redireccionará al usuario al portal de Multipagos Bajío y a su vez enviará los datos necesarios del pago a Banco del Bajío.

b) El portal de Multipagos Bajío guiará al usuario a realizar el pago indicado por el cliente.

c) En caso de éxito, el portal de Multipagos Bajío mostrará el comprobante correspondiente al usuario o contribuyente, a su vez notificará al portal del cliente el pago efectuado. Dado que la información del pago se envía desde el portal de Multipagos Bajío, cuando se termina la transacción y el usuario cierra esa ventana, el portal de Multipagos no redirigirá al usuario a la página del cliente, continuará en la página en la que inició el trámite dando clic al botón de Pagar.

4.3. Requerimientos del cliente de BanBajío:

Tener desarrollado su sistema de cobranza en línea (que los usuarios puedan autenticarse al sitio para poder identificar quien es el que va a pagar a través de Multipagos).

Contar con un portal con acceso a Internet.

Identificar previamente a los usuarios, creación de los mismos y contraseñas de acceso al sitio para consulta de sus adeudos.

Aplicar reglas de seguridad en Internet para proteger o evitar exponer sus llaves privadas del cliente.

Desarrollar una página que pueda ser consumida por BanBajío para notificaciones de pagos.

4.4. Requerimientos Técnicos:

Para asegurar la confidencialidad, integridad, seguridad de la información, el correcto funcionamiento y desarrollo; el cliente deberá cubrir los siguientes requerimientos técnicos:

4.4.1. Esquema de Desarrollo

El cliente deberá desarrollar y/o adecuar su solución web para intercambiar información entre Banco del Bajío y su aplicativo principal.

Protocolo de intercambio de información entre el cliente y Banco del Bajío: será a través un socket que se abrirá por el puerto 80, utilizando protocolos HTTP y TCP, por el cual se enviarán variables por el método POST.

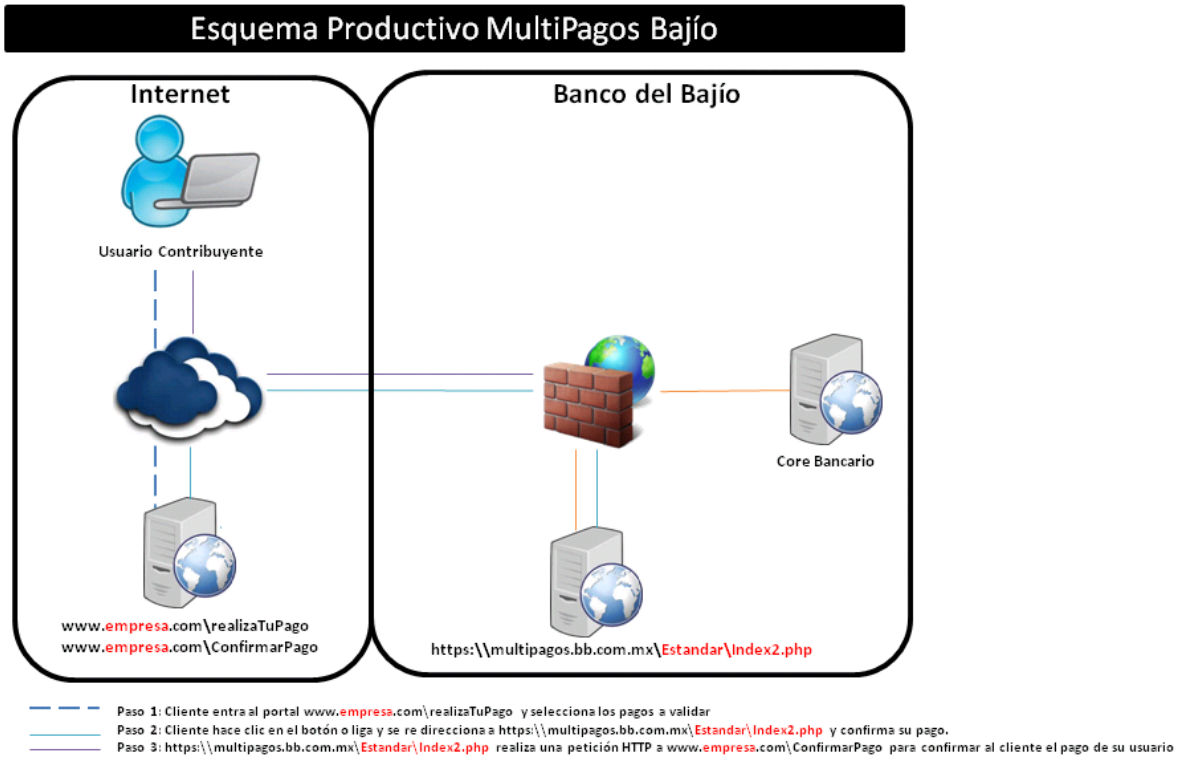
Plataforma de desarrollo:
Cualquiera que cumpla el estándar http: por ejemplo: PHP, Java, etc,

NOTA: se recomienda no utilizar frames, debido a bugs que se pueden generar al cambiar de dominios (errores en manejos de sesión), la recomendación en general es cumplir con los estándares de W3C (www.w3.org).

Esquema de Seguridad:
Criptografía asimétrica (RSA), *ver sección 4.4.3.*

4.4.2. Infraestructura:

Ambientes requeridos por parte del cliente: Productivo, contar con un servidor público en Internet donde sus usuarios se registren por medio de un usuario y contraseña, se les muestren los adeudos que tienen registrados, puedan seleccionar los adeudos de los cuales desean realizar el pago y donde Banco del Bajío pueda responder un pago exitoso (p.e. www.miempresa.com/notificaPago.php)



4.4.3. Seguridad

Dentro del sitio de Multipagos Bajío se estará utilizando el protocolo HTTPS, sin embargo y debido a que la arquitectura de comunicación entre Banco del Bajío y el cliente que redirige a su vez a su usuario o contribuyente para realizar los pagos, será a través del protocolo HTTP utilizando el mecanismo POST; los cuales serán encriptados y generando un hash, es recomendable implementar mecanismos de control de integridad de información.

A continuación se describe el proceso de control de integridad y autenticidad que se propone utilizar en el intercambio de información entre Banco del Bajío y el cliente.

4.4.3.1. Generación de llaves asimétricas

Este primer paso se detalla en el documento "Generación de llaves asimétricas.ppt" de forma visual para Windows y Linux.

El formato de salida de las llaves es PEM, el cual es un formato estándar para intercambio de llaves, el mismo contiene las llaves binarias convertidas a base64.

4.4.3.2. Intercambio de llaves

Banco del Bajío y el cliente deberán intercambiar sus respectivas llaves públicas. La forma de intercambio se recomienda realizarla por mensajería y de manera confidencial (en un CD).

Los destinatarios y direcciones se negociarán en su momento.

4.4.3.3. Firma Digital

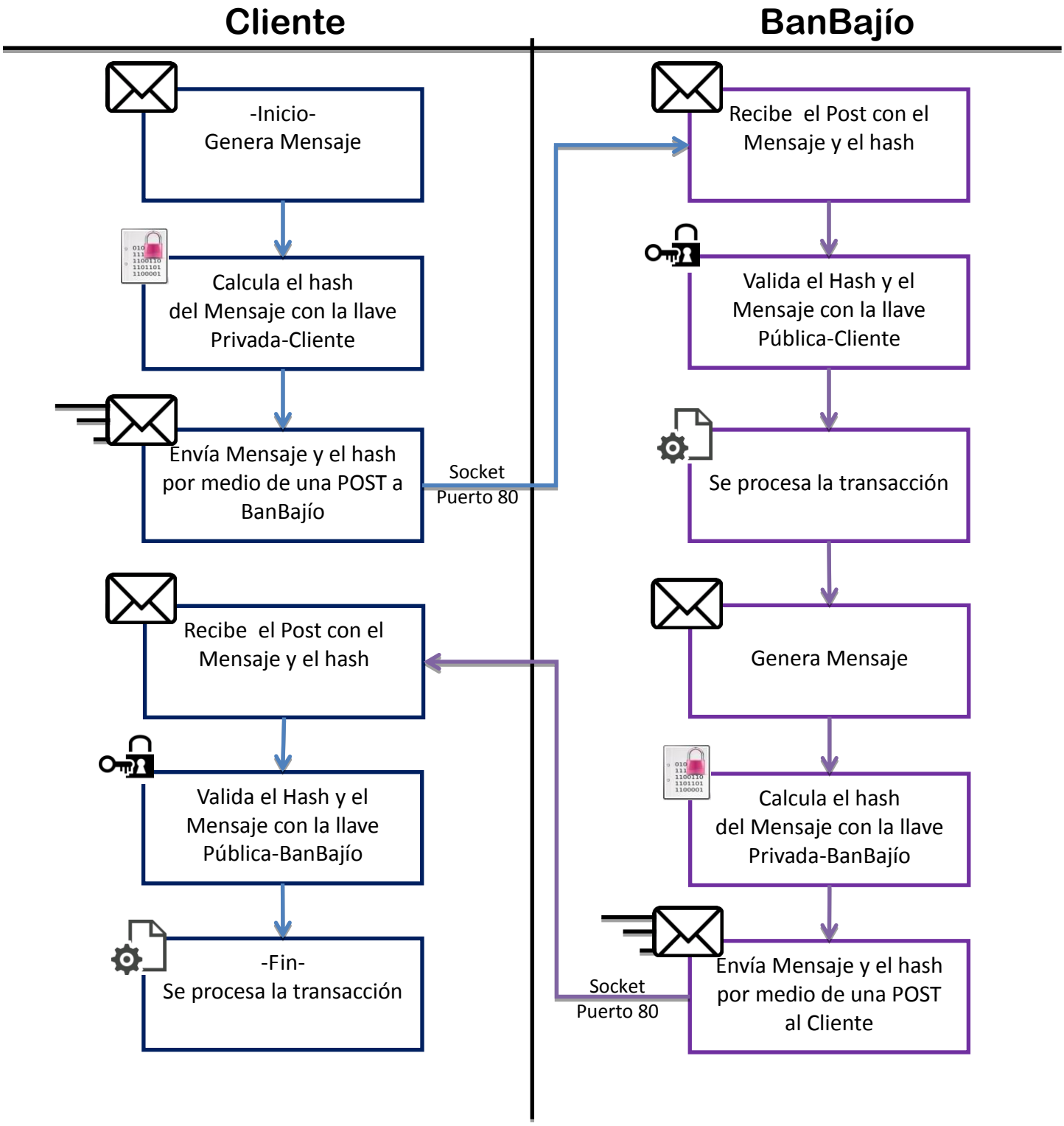
Una vez generadas las llaves correspondientes, el siguiente paso consiste en enviar el mensaje post acompañado de una firma digital. Esta firma digital dará validez e integridad a la información enviada a Banco del Bajío. La firma digital se compone de un hash cifrado con un algoritmo asimétrico como se detalla a continuación.

Función de hash: MD5
Cifrado asimétrico: RSA con una librería, se referencia como función K.

Para firmar el proceso de envío de información del cliente hacia Banco del Bajío, el primero deberá hacerlo utilizando su llave privada.

Utilizando las llaves se logra obtener autenticidad de los mensajes, y por ende utilizando la semilla (hash) se logra controlar la integridad de los datos recibidos.

En el siguiente gráfico se describe el proceso de envío de información entre el Cliente y BanBajío, firmas correspondientes, así como la verificación de la autenticidad e integridad del mensaje recibido.



4.4.4. Intercambio de información

El intercambio de información se realizará a través del protocolo HTTP mediante el envío de mensajes POST, los cuales serán enviados a través de un SOCKET por el puerto 80.

Nota: Los nombres se deberán de respetar a fin de hacer coincidir los desarrollos.

Variables POST	CLIENTE → BANCO DEL BAJIO	
Nombre Var	**Tipo de dato**	**Descripción**
cl_folio	Numérico, longitud variable (1-20)	Folio para identificar la transacción, este será **el mismo que se envío al banco**
cl_referencia	Alfanumérico, longitud variable (1-34)	Referencia, para identificar los conceptos de pago generado por el cliente, este será **el mismo que se envió al banco**
dl_monto	Numérico (11,2)	Monto total del pago, con punto decimal sin formato monetario, ejemplo 154623.98**, este será el mismo que se envió al banco**
cl_concepto	Numérico (11), longitud variable	Concepto por el cual se realiza el pago, ejemplo 1=Pago de Colegiatura
servicio	Numérico (11), longitud variable	Id de contratación de servicio Multipagos proporcionado por Banco del Bajío, ejemplo 3
hash	Alfanumérico, longitud variable	Firma generada por el Cliente, para ser verificada por Banco del Bajío para asegurar integridad de las variables y no repudio, es decir, asegurar que el Cliente es quien está transmitiendo la información y que no ha sufrido alteraciones desde su emisión

Ejemplo de envío de información del Cliente a Banco del Bajío, se utiliza un formulario **HTML** (el cliente abre una ventana emergente donde se desplegará la información de Banco del Bajío.), este código deberá ser integrado dentro de la funcionalidad del cliente.

También se especifica la URL a la cual se deberán enviar los datos al banco:
https://multipagos.bb.com.mx/Estandar/index2.php

```
<form method=POST action="https://multipagos.bb.com.mx/Estandar/index2.php" target="popup"
onsubmit="window.open('', 'popup', 'width=870,height=600,menubar=no, scrollbars=yes,
directories=no')">
        Folio:          <input name='cl_folio'      type='text' value='1'> <br>
        Referencia:     <input name='cl_referencia' type='text' value='abc123'> <br>
        Monto:          <input name='dl_monto'      type='text' value='1500.00'> <br>
        Concepto:       <input name='cl_concepto'   type='text' value='1'> <br>
        Servicio:       <input name='servicio'      type='text' value='3'>
        Firma:          <input name='hash'          type='hidden' value='hash_generado'>
        <input type='submit' value='Pagar'>
</form>
```

Variables POST BANCO DEL BAJIO → CLIENTE		
Nombre Var	**Tipo de dato**	**Descripción**
cl_folio	Numérico, longitud variable (1-20)	Folio para identificar la transacción, este será **el mismo que se envió al banco**
cl_referencia	Alfanumérico, longitud variable (1-34)	Referencia, para identificar los conceptos de pago generado por el cliente, este será **el mismo que se envió al banco**
dl_monto	Numérico (11,2)	Monto total del pago, con punto decimal sin formato monetario, ejemplo 154623.98**, este será el mismo que se envió al banco**
dt_fechaPago	Fecha (yyyymmdd)	Fecha del pago, fecha en la que se realiza el pago
nl_tipoPago	Numérico (2)	**Tipo del Pago:** 01- Tarjeta de Crédito (Adquiriente) 02 - Cargo a cuenta Bajío 03 – Cargo a cuentas CLABEs de otros bancos (Domiciliación) 04 – Tarjeta de débito (Adquiriente) 05 – Tarjeta de Prepago (Adquiriente) 06 – Tarjeta de débito (Domiciliación)
nl_status	Numerico(2)	**Status del Pago:** Estado de la transacción. 01 = Cobrado 02 = Rechazo 03 = Procesado (este aplica para el tipo de pago 03)
hash	Alfanumérico, longitud variable	Firma generada por el Banco del Bajío, para ser verificada por el Cliente para asegurar integridad de las variables y no repudio, es decir, asegurar que Banco del Bajío es quien está transmitiendo la información y que no ha sufrido alteraciones desde su emisión

Todos los ejemplos de la función SignData (mostrados en la siguiente sección) dan como resultado el hash o firma digital y con formato en base64.

Para generar el hash cifrado o la firma digital por parte del cliente, el texto de entrada serán las variables post indicadas concatenadas por un pipe "|" (al final también se agrega el "|"). No existen espacios entre los campos y el carácter "|".

Cadena CLIENTE= "cl_folio|cl_referencia|dl_monto|cl_concepto|servicio|"
Cadena BAJIO= "cl_folio|cl_referencia|dl_monto|dt_fechaPago|nl_tipoPago|nl_status|"

Notas: Para verificar el hash o forma digital generada por el cliente, Banco del Bajío realizará la misma concatenación de las variables POST recibidas.

Para verificar el hash o forma digital generada por Banco del Bajío, el cliente realizará la misma concatenación de las variables POST recibidas.

Para validar si la firma digital o el hash cifrado es correcto, la función VerifyData recibe el hash cifrado, el texto en claro y la llave pública.

El código de las funciones SignData y VerifyData está definido en:

\FuncionesCriptograficas\mx.com.bajio.crypto-1.0.0.jar (para Java)
TestRSA.rar (para PHP)

4.4.4.1. Ejemplo de implementación en Java

Llave pública

Si las llaves se utilizarán en JAVA se deberá quitar la cabecera y el pié de página de la llave pública. O sea que se quitarán las líneas "-----BEGIN PUBLIC KEY----- " y "-----END PUBLIC KEY-----".

-----BEGIN PUBLIC KEY-----
MFwwDQYJKoZIhvcNAQEBBQADSwAwSAJBAKl/PCfGF6H303/Ecm/l3z8mNtSuqmDbz
Sms+f8IQRPn6P4JTzqWIGW3jOC0Q6cSGMja53/VNO6akHbrmqg5nPwECAwEAAQ==
-----END PUBLIC KEY-----

Llave privada

Para utilizar la llave privada generada con anterioridad se debe cambiar el formato de la misma utilizando OpenSSL.

openssl pkcs8 -topk8 -inform PEM -outform DER -in private_key.pem -out private_key.der -nocrypt

Las siguientes funciones serán implementadas por el cliente para enviar y recibir información

Para generar el hash o Firma digital:

```
package mx.com.bajio.crypto;

public class TestCrypto {

        public static void main(String[] args)
        {
                try
                {
                        String hash;
                        Hash = CryptoBajio.SignData("Cadena"," private_key.der ");
                }
                catch (Exception e)
                {
                        e.printStackTrace();
                }
        }
}
```

cl_folio|cl_referencia|dl_monto|cl_concepto |servicio|
Esta cadena la concatenará el CLIENTE con las variables POST a enviar a Banco del Bajío

Llave privada del CLIENTE

Para verificar el hash o Firma digital recibida de Banco del Bajío:

```
public class TestCrypto {

        public static void main(String[] ar
        {
                try
                {

                        String base64 =
"ZH9qgtr7MLlkNbRVIC                    hmJ0fuRD/p                    sKe0PW3/QjKWQ" +
"u56DcsF4I0qi3um2BL                    CR/vuRS986i                    RSH1WyoDg2lU/7sMvc6KJGrMi" +
"shFsF73Rz6ghF/x5of

                if(CryptoBajio.VerifyData(base64," Cadena "," public.key ") )
                                System.out.println("Datos validos.");
                        else
                                System.out.println("Datos invalidos.");

                }
                catch (Exception e)
                {
                        e.printStackTrace();
                }
        }
}
```

cl_folio|cl_referencia|dl_monto|dt_fechaPago|nl_tipoPago|nl_status|

Esta cadena la concatenará el CLIENTE con las variables POST enviadas por Banco del Bajío

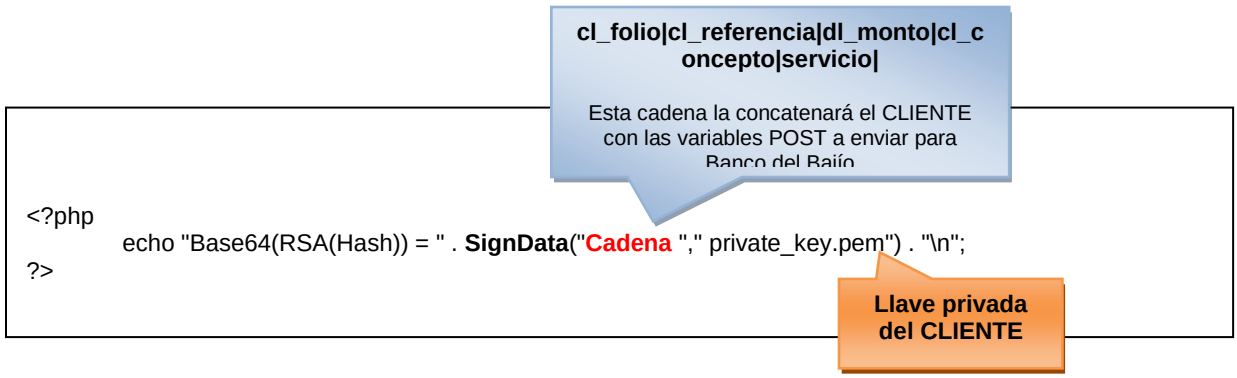
Hash generado por el Banco

Llave pública de Banco del Bajío

4.4.4.2. Ejemplo de implementación en PHP

Las siguientes funciones serán implementadas por el CLIENTE para enviar y recibir información

Para generar el hash o Firma digital:



Para verificar el hash o Firma digital recibida de Banco del Bajío: