Elektronikus Kártyatranzakciók az Interneten

A CIB Bank Zrt.

eCommerce

Internetes kártyaelfogadás szolgáltatás

technikai dokumentációja

Fejlesztői útmutató

CIB BANK

Tartalomjegyzék

CIB BANK

Előszó

Jelen leírás azoknak a fejlesztőknek szól, akik webáruházhoz terveznek fizetőmodult, mely képes a CIB Bank által rendelkezésre bocsátott API használatára.
A leírás szakmai jellegű, és feltételezi, hogy a megvalósítás alapját képező programnyelv ismert.

A dokumentum olvasása előtt ajánlott a referencia leírás elolvasása, mivel az itt említett lépések abban találhatóak meg részletesen kifejtve.

A leírásban szereplő konkrét példák PHP nyelven íródtak, mivel tapasztalataink szerint ezt a nyelvet választották eddig a legtöbben webes alkalmazások implementálására. A példákban a titkosításhoz ugyanaz a kulcs tartozik, ami a példa alkalmazásoknál is található.

CIB BANK

Fizetési folyamat

Előkészítés

A vásárló azonosítása

A banki tranzakció megkezdése előtt a vásárlót azonosítani kell. Az azonosítás tetszőleges, általánosan elfogadott módon történhet, amennyiben az megfelel az alábbi feltételeknek:

- A vásárló egyértelműen azonosítható
- Megállapítható a vásárló értesítési csatornája/címe

Az azonosítást a webáruház végzi el, a tranzakció ilyenkor egy belső azonosítót kap, ami vagy a vásárlót, vagy a vásárló által nyitott session-t azonosítja. Ezt a belső azonosítót kell összekötni a banki tranzakció azonosítójával (lásd következő fejezet).

A tranzakció adatainak előállítása

A már beazonosított vásárló, illetve a hozzá tartozó kosár, valamint a szerződéskor a bank által rendelkezésre bocsátott adatok segítségével elő kell állítani a banki tranzakcióhoz szükséges paramétereket:

PID

A szerződéskor a bank bocsátja a kereskedő rendelkezésére. Első 3 karaktere megegyezik a titkosító kulcs nevével, utolsó négy számjegye azonosítja a virtuális POS terminált. A dokumentumban használt példákban az IEB0001 érték látható, de ez az azonosító semmilyen körülmények között nem egyezhet meg a banki teszt és éles szerveren használhatóval.

TRID

A webáruház által generált egyedi tranzakció azonosító. 16 jegyű véletlenszám, mely a tranzakciót annak teljes élettartama alatt (authorizáció, illetve annak visszavonása vagy visszautalás) azonosítja. Egy esetleges jövőbeni tranzakció vizsgálat során szükség lesz a TRID értékére, így kérjük, hogy ezt minden esetben tárolják legalább egy évig. Amennyiben a webáruház saját azonosítót is használ akkor a két értéket kapcsolják össze.
Példa a generálásra:

```
mt_srand();
$trid="";
for ($i=0; $i<4; $i++)
    $trid .= mt_rand(1000, 9999);
```

Tekintettel arra, hogy az azonosító egyedi, a végleges értéket addig kell generálni, amíg olyan értéket nem ad, ami még egyetlen tranzakciónál sem létezett.

UID

Az ügyfélazonosító értéke a banki üzenet szempontjából nem releváns, célszerű azonban összekötni a webáruház vásárló azonosítójával (ügyelni kell azonban arra, hogy a banki üzenetben maximum 11 karakter hosszú lehet).

CUR

A tranzakcióban fizetendő összeg devizaneme ISO3166 szabvány szerinti 3 betűs formában. Jelenleg lehetséges értékei a következők:

- EUR
- HUF

AMO

A tranzakció összege magyar forint esetén egész szám, euró esetében pedig két tizedesjegyre kerekített tört szám kell legyen. Példa az összegre:

```
$cur = "EUR";
$orig_amo = 100;
switch ($cur)
{
    case "EUR":
        $amo = number_format($orig_amo, 2, ".", "");
        break;
    case "HUF":
        $amo = number_format($orig_amo, 0, ".", "");
        break;
    default:
        $amo = 0;  // NOTE: this should not happen
}
```

Megjegyzés: noha a vásárló kibocsátó bankja felé küldött authorizációs üzenetben ugyanaz az összeg szerepel, mint az AMO értékben megadott, az esetleges konverziós műveletek miatt (pl. a vásárló USD kártyával fizet) a kártyáról levont érték ettől eltérhet.

TS

A webszervert futtató hardver rendszerideje ÉÉÉÉHHNNÓÓPPMM formátumban:

- ÉÉÉÉ: naptári év századdal együtt (1970-2050)
- HH: hónap (01-12)
- NN: nap (01-31)
- ÓÓ: óra (00-23)
- PP: perc (00-59)
- MM: másodperc (00-59)

A rendszeridővel szemben nem követelmény, hogy atomórához szinkronizált legyen, de a formátumtól eltérés sikertelen feldolgozáshoz vezethet. Példa az időpecsétre:

```
$ts = date("YmdHis");
```

AUTH

Az authorizáció típusa, web alapú tranzakciók esetén értéke kötelezően ’0’.

LANG

A vásárló átirányításakor a vásárló böngészőjében megjelenő fizetőoldal szövegének nyelve. Az elfogadott értékek megtalálhatóak a referencia leírásban. A leírásban nem szereplő nyelvkód használata hibás feldolgozáshoz vezet.

URL

A vásárlót a bankkártya adatok megadásához minden esetben a bank fizetőoldalára kell irányítani. Az authorizációt követően a vásárló az ebben a paraméterben megadott címre kerül visszairányításra. Az URL abszolút elérési címet kell tartalmazzon, http vagy https protokollal. Nem tartalmazhat paramétert (az összes üzenetben kötelezően jelen levő TRID érték alapján a vásárló session értéke könnyen visszakereshető). Az URL értékét a titkosítást megelőzően nem szabad urlkódolni.

CIB BANK

Authorizáció

Tranzakció inicializálás

Az előállított szükséges paramétereket titkosítás után paraméterként felhasználva meg kell hívni a bank *kereskedő* urljét. Ügyelni kell arra, hogy csak az üzenethez szükséges paraméterek küldhetőek, minden eltérés hibás feldolgozáshoz vezet. A bank kititkosított válaszában az RC érték jelzi, hogy sikerült-e az üzenet feldolgozása. Amennyiben a bank felé küldött üzenet teljesen értelmezhetetlen, a bank egy titkosítatlan hibakódot ad, mely utal a hiba okára. Példa az inicializálásra (eki_* függvények értelmezését lásd a Titkosítás illetve Kommunikáció fejezetekben):

```
$try = 0;
do {
    generateparams(); $try++
    $msgt10clear =  "PID=".$pid."&TRID=".$trid."&MSGT=10&UID=".$uid;
    $msgt10clear .=  "&AMO=".$amo."&CUR=".$cur."&TS=".$ts;
    $msgt10clear .=  "&AUTH=0&LANG=".$lang."&URL=".$url;
    $msgt10crypt = eki_encrypt($msgt10clear);
    $msgt11crypt = eki_call($eki_merchanturl, $msgt10crypt);
    $msgt11clear = eki_decrypt($msgt11crypt);
    parse_str($msgt11clear, $msgt11arr);
    $msgt11rc = $msgt11arr['RC'];
} while (($msgt11rc == "02") && ($try <= 3));
```

A példa alapján a ciklusból kiérve $msgt11rc lehetséges értékei és értelmezésük:

- 00: az inicializálás sikeres volt
- 01: az inicializálás egyéb technikai okok miatt nem sikerült 02: az inicializálás nem sikerült, mert a generált TRID foglalt

A bank esetlegesen visszaadott titkosítatlan hibakódja utalhat általános hibára, ezek elhárítását lásd a Hibakezelés részben.

Amennyiben $msgt11rc értéke ’01’, a tranzakciót **meg kell szakítani**, és a vásárlót tájékoztatni kell a fizetés meghiúsulásáról. ’02’-es kód esetén célszerű újabb TRID értékkel megismételni az inicializációt. Sikeres inicializálás esetén ($msgt11rc=’00’) a TRID értékét el kell menteni, mivel a tranzakcióhoz tartozó összes további üzenetben ezt kell majd szerepeltetni.

Vásárló átirányítása a fizetőoldalra

Sikeres inicializációt követően a vásárló böngészőjét át kell irányítani a bank *vásárló* urljére. Az átirányítás tetszőleges technológiával történhet, célja hogy a vásárló böngészőjében a banki fizetőoldal (és csak az) jelenjen meg. A banki url paramétere a titkosított átirányításhoz szükséges (20-as) üzenet. Példa:

```
$msgt20clear =  "PID=".$pid."&TRID=".$trid."&MSGT=20";
$msgt20crypt = eki_encrypt($msgt20clear);
header('Location: $eki_customerurl?$msgt20crypt');
```

A webszerver ennek hatására elveszíti a kapcsolatot a vásárlóval, aki átkerül a banki fizetőoldalra a kártyainformációk megadása érdekében.

CIB BANK

Vásárló visszaérkezésének lekezelése

A kártyainformációk megadását követően a bank a vásárlót visszairányítja az inicializációs üzenetben megadott URL paraméterben található címre. A visszairányítással együtt GET paraméterben érkezik a bank üzenete, melyben kititkosítás után megtalálható a tranzakció azonosítója. Ez alapján társítható ismét a webáruház session-je a banki tranzakcióval.

```
$msgt21clear = eki_decrypt(rawurlencode($_SERVER['QUERY_STRING']));
parse_str($msgt21clear, $msgt21arr);
$trid = $msgt21arr['TRID'];
```

Amennyiben a kapott $trid érték alapján nem sikerül a session-t azonosítani (feltéve hogy ez a két érték korábban mentésre került), a tranzakciót nem szabad folytatni, és a vásárlót tájékoztatni kell a hibáról.

Vásárló tájékoztatása

Amennyiben a vásárló sikeresen visszaérkezik a webáruházba, ott kötelező tájékoztatni a tranzakció eredményéről. Az eredmény lekérését és a tranzakció lezárását (MSGT32) az inicializációhoz hasonlóan szerver-szerver üzenetként szükséges implementálni:

```
$msgt32clear =  "PID=".$pid."&TRID=".$trid."&MSGT=32&AMO=".$amo;
$msgt32crypt = eki_encrypt($msgt32clear);
$msgt31crypt = eki_call($eki_merchanturl, $msgt32crypt);
$msgt31clear = eki_decrypt($msgt31crypt);
parse_str($msgt31clear, $msgt31arr);
$msgt31rc = $msgt31arr['RC'];
$msgt31rt = $msgt31arr['RT'];
$msgt31anum = $msgt31arr['ANUM'];
```

A visszaigazoló oldalon a következő paramétereket kell szerepeltetni:

- Tranzakció azonosító (TRID)
- Összeg (AMO)
- Devizanem (CUR)
- Eredménykód (RC)
- Eredmény szövegesen (RT)
- Engedélykód (ANUM)

Ajánlott a vásárló figyelmének felhívása a fenti adatok elmentésére, valamit kötelező az adatok alternatív úton (e-mail, sms) vásárlóhoz juttatása.
Tekintettel arra, hogy a vásárló mindennemű kapcsolatot megszakít a webáruházzal a fizetés idejére, elképzelhető, hogy (jellemzően technikai hiba miatt) nem is kerül vissza a webáruházba. A fizetés ennek ellenére megtörténhet, ám amennyiben a webáruház egy előredefiniált időintervallumon belül (alapértelmezés szerint 10-15 perc) nem zárja le az eredményt, a bank a tranzakciót *reverzálja* (a vásárló kártyáján történt foglalást feloldja). Amennyiben a webáruház a fizetés megtörténte előtt kéri le annak eredményét, a bank szervere hibaüzenetet ad. Az ebből adódó időzítési probléma feloldására a bank biztosít egy olyan lekérdező üzenetet (MSGT33), amely a maszkolt kártyaszámon kívül mindenben megegyezik az MSGT32 üzenettel., de minden esetben ad RC értéket. Ezt az üzenetet a hagyományos fizetési algoritmussal párhuzamosan kell alkalmazni, a vásárló bankba irányítását követően, percenként felhasználva. A MSGT33-as üzenetre visszaadott válaszban található RC kódok értelmezése:

- PR: a fizetés még nem történt meg, de nem is szakadt meg időtúllépés miatt
- TO: a fizetés időtúllépés miatt megszakadt
- 00: a fizetés sikeresen befejeződött
- Bármi egyéb: a fizetés sikertelenül befejeződött A legsűrűbben előforduló hibakódokat azok lehetséges okaival a 3. Melléklet tartalmazza, de ettől függetlenül tetszőleges hibakód kezelésére fel kell készíteni az alkalmazást egy általános hibaág létrehozásával, ami lezárja a tranzakciót.

Az MSGT33-as üzenetek küldését addig kell folytatni, míg a visszatérő üzenet RC értéke 'PR'. Amennyiben a kapott MSGT31 válaszüzenet RC értéke '00', a webáruház küldhet lezáró üzenetet (MSGT32), ezzel véglegesítve a sikeres fizetést. Ekkor a bank a megszabott időintervallum leteltével sem oldja fel a foglalást (pontosan úgy, mintha a vásárló sikeresen visszaért volna a webáruházba).

Ha a vásárló a lezárást követően tér csak vissza a webáruházba, tájékoztatni a már ismert adatok alapján kell, újabb lezárás nem lehetséges.

A webáruház a vásárló bankba küldése után a fizetés pillanatnyi állapota mellett lekérdezheti a tranzakció összes korábbi lépését is:

```
$msgt37clear =  "PID=".$pid."&TRID=".$trid."&MSGT=37&AMO=".$amo;
$msgt37crypt = eki_encrypt($msgt37clear);
$msgt38crypt = eki_call($eki_merchanturl, $msgt37crypt);
$msgt38clear = eki_decrypt($msgt38crypt);
parse_str($msgt38clear, $msgt38arr);
$msgt38rc = $msgt38arr['RC'];
$msgt38hist = $msgt38arr['HISTORY'];
```

A tömb 'RC' értéke jelzi, hogy az üzenet végrehajtása sikeres volt-e (00: sikeres, 01: hiba történt), sikeres végrehajtás esetén pedig a 'HISTORY' elemben levő string tartalmazza az egyes állapotokat, azok időbeni sorrendjében. A lehetséges értékek listáját a referencia leírás tartalmazza (3.1 Meződefiníciók, HISTORY). Az üzenet akkor is hibát jelez, ha a tranzakció inicializálva lett bár, de a fenti leírásban szereplő egyetlen állapot sem következett még be (pl még nem történt meg a vásárló banki fizetőoldalra irányítása).

Authorizáció visszavonása

Sikeres és lezárt foglalást (MSGT32-es üzenetre válaszként küldött MSGT31-es üzenet RC értéke ’00’) a webáruház is feloldhat. Ennek hatása ugyanaz, mintha a bank tette volna az időtúllépési periódust követően. A feloldás előtt meg kell győződni arról, hogy a tranzakció valóban feloldható-e. Erre a célra a MSGT70-es üzenet szolgál, mely a tranzakció authorizációs eredményét kiegészíti az elszámolás aktuális állapotával (MSGT71 STATUS mező). Amennyiben az állapot ’10’ (foglalt, de még nem terhelt), a MSGT74 segítségével a tranzakció reverzálható. Sikeres reverzálás esetén a válaszüzenetben (MSGT75) szereplő STATUS mező értéke ’40’, a továbbiakban a MSGT71 is ezt az értéket adja. Visszavont foglalás terhelésére nincs lehetőség.

```
if ($msgt31rc = $msgt31arr['RC']) {
   $msgt70clear =  "PID=".$pid."&TRID=".$trid."&MSGT=70&AMO=".$amo;
   $msgt70crypt = eki_encrypt($msgt70clear);
   $msgt71crypt = eki_call($eki_merchanturl, $msgt70crypt);
   $msgt71clear = eki_decrypt($msgt71crypt);
   parse_str($msgt71clear, $msgt71arr);
   if ($msgt71arr['STATUS'] == "10") {
          $msgt74clear =  "PID=".$pid."&TRID=".$trid."&MSGT=74&AMO=".$amo;
          $msgt74crypt = eki_encrypt($msgt74clear);
          $msgt75crypt = eki_call($eki_merchanturl, $msgt74crypt);
          $msgt75clear = eki_decrypt($msgt75crypt);
   }
}
```

CIB BANK

Visszautalás

A protokoll lehetőséget biztosít a már terhelt tranzakciókat részben vagy teljes egészében egy naptári éven belül visszautalni. A beállítható visszautalandó összeg értéke minimálisan 1 EUR vagy 100 HUF. Amennyiben az eredeti terhelés ennél kisebb összeggel történt, a tranzakcióra a visszautalás nem engedélyezett. A visszautalandó összeg beállítása (MSGT80) után a visszautalás kérésével (MSGT78) a bank a beállított összeget visszautalja a vásárló kártyájára. Alapértelmezés szerint (a MSGT80 üzenetet kihagyva) a visszautalás 0 összeggel történik, ezért a MSGT80-as üzenet használata visszautalás előtt erősen javasolt.

```
if ($msgt31rc = $msgt31arr['RC']) {
   $msgt70clear =  "PID=".$pid."&TRID=".$trid."&MSGT=70&AMO=".$amo;
   $msgt70crypt = eki_encrypt($msgt70clear);
   $msgt71crypt = eki_call($eki_merchanturl, $msgt70crypt);
   $msgt71clear = eki_decrypt($msgt71crypt);
   parse_str($msgt71clear, $msgt71arr);
   $status = $msgt71arr['STATUS'];
   if (($status == "20") || ($status == "30")) {
          $msgt80clear =  "PID=".$pid."&TRID=".$trid;
          $msgt80clear .= ."&MSGT=80&AMOORIG=".$amo."&AMONEW=".$amonew;
          $msgt80crypt = eki_encrypt($msgt80clear);
          $msgt81crypt = eki_call($eki_merchanturl, $msgt80crypt);
          $msgt81clear = eki_decrypt($msgt81crypt);
          $msgt78clear =  "PID=".$pid."&TRID=".$trid;
          $msgt78clear .= ."&MSGT=78&AMO=".$amo;
          $msgt78crypt = eki_encrypt($msgt78clear);
          $msgt79crypt = eki_call($eki_merchanturl, $msgt78crypt);
          $msgt79clear = eki_decrypt($msgt79crypt);
   }
}
```

A fenti példa lekérdezi a tranzakció státuszát, és amennyiben az terhelt ($amo értékben), beállítja a visszautalandó összeget ($amonew), majd visszautalást indít. A visszautalt tranzakció státusza ekkor ’50’-re változik. Korábban részben már visszautalt tranzakcióra nem lehet újabb visszautalást kezdeményezni, de a visszautalandó összeget többször is lehet módosítani visszautalás előtt.

CIB BANK

Titkosítás

A bankhoz küldött összes üzenetet küldés előtt titkosítani kell. A titkosítás lépéseit a referencia kézikönyv tartalmazza, ezen felül a dokumentáció mellékletét képezi titkosító példaalkalmazás JAVA, PHP és C# nyelveken. A példaalkalmazások fordításához szükséges, harmadik féltől származó könyvtárak listáját a források elején elhelyezett komment részek tartalmazzák.
A titkosításhoz az algoritmuson kívül szükséges egy titkosító kulcs is, ezt minden esetben a bank bocsátja a kereskedő rendelkezésére. A kulcsok nevei minden esetben megegyeznek az egyes üzenetekben használandó PID paraméter értékének első 3 karakterével. A .reg kiterjesztésű verzió windows alapú rendszerekhez ajánlott, regisztrálása után a kulcs a \HKLM\Software\IEB\Eki kulcs alá kerül, a kulcs neve megegyezik a file nevével, az egyetlen érték (des) a teljes kulcsfilet tartalmazza bináris formában. A .des kiterjesztésű filet inkább a nem windows rendszert használó webáruházak számára ajánljuk, a benne levő bináris tartalom pontosan megegyezik a registryben tárolható értékkel. A titkosítókulcs a file utolsó 24 bytejában található, key1, key2, iv sorrendben (a kulcsok és az inicializációs vektor rendre 8-8 byte hosszúak). A titkosítás 3DES CBC módszerrel végzendő. Példa a titkosításra (feltételezendő, hogy a kulcs file az aktuális könyvtárban található):

```
function eki_encrypt($cleartext) {
     $arr=explode("&",$cleartext);
     $ciphertext="";
     $pid="";
     for ($i=0;$i<count($arr);$i++) {
         if (strtoupper($arr[$i])!="CRYPTO=1")
             $ciphertext.="&".$arr[$i];
         if (substr(strtoupper($arr[$i]),0,4)=="PID=")
             $pid=substr(strtoupper($arr[$i]),4,7);
     }
     $ciphertext=substr($ciphertext,1);
     $ciphertext=rawurlencode($ciphertext);
     $ciphertext=str_replace("%3D","=",$ciphertext);
     $ciphertext=str_replace("%26","&",$ciphertext);
     $crc=str_pad(dechex(crc32($ciphertext)),8,"0",STR_PAD_LEFT);
     for ($i=0;$i<4;$i++)
         $ciphertext.=chr(base_convert(substr($crc,$i*2,2),16,10));
     $pad=8-(strlen($ciphertext) % 8);
     for ($i=0;$i<$pad;$i++)
         $ciphertext.=chr($pad);
     $f=fopen(substr($pid,0,3).".des","r");
     $keyinfo=fread($f,38);
     fclose($f);
     $key1=substr($keyinfo,14,8);
     $key2=substr($keyinfo,22,8);
     $iv=substr($keyinfo,30,8);
     $key=$key1.$key2.$key1;
     $td=mcrypt_module_open("tripledes","","cbc","");
     mcrypt_generic_init($td,$key,$iv);
     $ciphertext=mcrypt_generic($td,$ciphertext);
     mcrypt_module_close($td);
     $pad=3-(strlen($ciphertext) % 3);
     for ($i=0;$i<$pad;$i++)
         $ciphertext.=chr($pad);
     $ciphertext=base64_encode($ciphertext);
     $ciphertext=rawurlencode($ciphertext);
     $ciphertext="PID=".$pid."&CRYPTO=1&DATA=".$ciphertext;
     return $ciphertext;
 }
```

Hasonló módon értelmezendő a kititkosítás is (a bank minden válaszüzenetet titkosítva küld):

```
function eki_decrypt($ciphertext)
{
    $arr=explode("&",$ciphertext);
    $cleartext="";
    $pid="";
    for ($i=0;$i<count($arr);$i++) {
        if (substr(strtoupper($arr[$i]),0,5)=="DATA=")
            $cleartext=substr($arr[$i],5);
        if (substr(strtoupper($arr[$i]),0,4)=="PID=")
            $pid=substr(strtoupper($arr[$i]),4,7);
    }
    $cleartext=rawurldecode($cleartext);
    $cleartext=base64_decode($cleartext);
    $lastc=ord($cleartext[strlen($cleartext)-1]);
    $validpad=1;
    for ($i=0;$i<$lastc;$i++)
        if (ord(substr($cleartext,strlen($cleartext)-1-$i,1))!=$lastc)
            $validpad=0;
    if ($validpad==1)
        $cleartext=substr($cleartext,0,strlen($cleartext)-$lastc);
    $f=fopen(substr($pid,0,3).".des","r");
         $keyinfo=fread($f,38);
    fclose($f);
    $key1=substr($keyinfo,14,8);
    $key2=substr($keyinfo,22,8);
    $iv=substr($keyinfo,30,8);
    $key=$key1.$key2.$key1;
    $td=mcrypt_module_open("tripledes","","cbc","");
    mcrypt_generic_init($td,$key,$iv);
    $cleartext=mdecrypt_generic($td,$cleartext);
    mcrypt_module_close($td);
    $lastc=ord($cleartext[strlen($cleartext)-1]);
    $validpad=1;
    for ($i=0;$i<$lastc;$i++)
        if (ord(substr($cleartext,strlen($cleartext)-1-$i,1))!=$lastc)
            $validpad=0;
    if ($validpad==1)
        $cleartext=substr($cleartext,0,strlen($cleartext)-$lastc);
    $crc=substr($cleartext,strlen($cleartext)-4);
    $crch="";
    for ($i=0;$i<4;$i++)
        $crch.=str_pad(dechex(ord($crc[$i])),2,"0",STR_PAD_LEFT);
    $cleartext=substr($cleartext,0,strlen($cleartext)-4);
    $crc=str_pad(dechex(crc32($cleartext)),8,"0",STR_PAD_LEFT);
    if ($crch!=$crc)
        return "";
    $cleartext=str_replace("&","%26",$cleartext);
    $cleartext=str_replace("=","%3D",$cleartext);
    $cleartext=rawurldecode($cleartext);
    return $cleartext;
}
```

A bank az élesítés pillanatáig két kulcspárt bocsát ki:, teszt és éles párokat.. A teszt kulcspár a banki tesztszerverrel, az éles a banki éles szerverrel való kommunikációra alkalmas csak, felcserélésük esetén a banki szerver általános hibaüzenetet (RC=S01) ad. Tekintettel arra, hogy a teszt és az éles kulcspár nevei megegyeznek (a könnyebb élesbe állást segítendő), célszerű a kulcsok ellenőrzőösszegét (pl MD5) megjegyezni a későbbi egyértelmű azonosíthatóság érdekében.

CIB BANK

Kommunikáció

Webáruház-Bank kommunikáció

A banki szerverrel történő üzenetváltás minden kereskedői üzenet esetében http csatornán keresztül zajlik, a bank által meghatározott urlen keresztül. Az összeállított és titkosított paramétereket GET vagy POST metódus segítségével lehet a bank szerveréhez juttatni. A bank válasza minden esetben a kért üzenet content részében érkezik, text/plain formátumban. Amennyiben a kérés feldolgozása sikeresen fejeződik be, a kapcsolat http státusza 200, a content-ben érkező válasz pedig az EKI szabvány szerint titkosított. Hibás feldolgozás esetén a http státusz nem 200 (jellemzően 403 vagy 500), a contentben levő válasz nem titkosított, és csak a hiba kódját tartalmazza (pl. kititkosíthatatlan kérés esetén a http státusz 403, a content pedig 'RC=S01'). Példa a kereskedői üzenetek küldésére/fogadására:

```
function eki_call($url, $params)
{
   $ch = curl_init();
   curl_setopt($ch, CURLOPT_URL, ($url);
   curl_setopt($ch, CURLOPT_POST, 1);
   curl_setopt($ch, CURLOPT_POSTFIELDS, $params);
   curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
   curl_setopt($ch, CURLOPT_TIMEOUT, 50);
   $http_response = curl_exec ($ch);
   $http_status = curl_getinfo($ch, CURLINFO_HTTP_CODE);
   curl_close($ch);
   if ($http_status != "200") {
          echo "HTTP STATUS ".$http_status.", EKI error: ".$http_response;
          exit;
   }
   $ekiCryptedResponse = $http_response;
   return($ekiCryptedResponse);
}
```

Vásárló-Bank kommunikáció

A vásárló átirányítása a bank fizetőoldalára tetszőleges módszerrel történhet. PHP kód esetén, amennyiben még nem történt a böngésző számára content átadás, célszerű a header() függvényt használni:

```
   header("Location: ".$eki_customerurl."?".$msgt20crypt);
   exit;
```

Fentieken kívül használható a JavaScript window.location.replace(%customerurl%), illetve a html <meta http-equiv="refresh" content="0; url=%customerurl%" /> tag használatával, ahol %customerurl% a teljes, GET paraméterezett banki fizetőoldal url.

CIB BANK

Problémakezelés

A dokumentáció eddigi részeiben feltételeztük, hogy minden körülmény adott a sikeres tranzakció végrehajtásához. Éles környezetben azonban a fenti pontokban részletezett lépések bármelyike hibára futhat. Ez a fejezet a hibák feltárásához, illetve azok elhárításához igyekszik segítséget nyújtani.

Titkosítási problémák

A sikeres titkosításhoz/kititkosításhoz szükséges a bank által a kereskedő rendelkezésére bocsátott titkosítókulcs, a hibátlanul működő titkosító alkalmazás, és a helyesen összeállított feldolgozandó üzenet. Ezen feltételek bármelyikének nem teljesülése automatikusan magával vonja a művelet hibás működését.

Titkosító kulcsok

A bank a webáruház számára két kulcsot (kulcscsomagot) bocsát rendelkezésre. Mivel a banki titkosító algoritmus szinkron titkosítás, a teszt kulcs a banki teszt szerverhez, az éles kulcs pedig csak az éles szerverhez használható. A teszt és éles kulcsok eltérnek tartalmukban, de nevükben nem. Amennyiben a bank válaszkódja folyamatosan 403, a content pedig 'RC=Sxx', az nagy valószínűséggel a kulcs hibájára vezethető vissza. Elképzelhető (bár nem valószínű), hogy a kulcs a banki kiadás és a webszerveren való telepítés befejezése között sérült, ilyenkor célszerű az ellenőrző összeget (pl MD5) összehasonlítani.
A kulcsokat a bank két formátumban bocsátja a kereskedő rendelkezésére, .reg formátumban (jellemzően Windows alapú rendszerekhez), illetve .des formátumban (jellemzően unix alapú rendszerekhez). Előbbi telepítése a webáruház registryjébe szükséges.
Telepítéskor ügyelni kell arra, hogy a titkosítást végző (jellemzően a webszervert működtető) technikai felhasználó (és csak ő) olvashassa a kulcsfileokat. Ügyelni kell arra is, hogy a titkosító kulcs nevének első 3 karaktere pontosan (csupa nagybetű) meg kell egyezzen a titkosítandó/titkosított üzenetben szereplő PID paraméter értékének első 3 karakterével.

Alkalmazások

Az EKI titkosító protokoll minden lépése kötelező. Natív implementáció esetén előfordulhat, hogy a titkosítási lépéssor végeredménye eltér a várttól. Ez a lépések valamelyikének hibás működésére vezethető vissza. Az alkalmazás fejlesztésénél ügyelni kell a következőkre:

- Szöveges típusra alakításkor mindig meg kell adni a helyes kódolást (ASCII), illetve amíg lehet, célszerű bináris formában tárolni az eredményt
- CRC32 számításakor big endian sorrendet kell alkalmazni
- Amennyiben a dekódoláskor a számított CRC32 érték eltér a kapottól, az hibának számít
- Amennyiben a felhasznált 3DES titkosító függvény implicit paddelést használ, az üzenetet előtte nem szabad paddelni
- URL kódoláshoz az RFC 1738 szabványt kell alkalmazni

A fenti lista nem teljes, de tartalmazza a legsűrűbben előforduló hibákat.
Amennyiben lehetséges, ajánlott a mellékelt példakódok, illetve a sakide függvénykönyvtár (és alkalmazás) használata. Probléma esetén a sakide alkalmazás '-v' kapcsolóval történő futtatása részletesebb hibaelemzést ad, amennyiben az azt hívó alkalmazás csak a standard outputot képes kezelni, célszerű átirányítani a standard error csatornát:

```
/<path>/<to>/sakide -v -e -s"<string_to_encode>" 2>&1
```

Adatok

Amennyiben a kereskedői üzenetre a banki szerver http státuszkódja 500, és a válaszüzenet 'RC=Dxx', a hiba valószínűsíthetően a küldött adatokban van. Az adatok összeállításakor ügyelni kell a következőkre:

- Az üzenetnek tartalmaznia kell a referencia útmutatóban szereplő összes felsorolt paramétert
- Minden paraméternek rendelkeznie kell az útmutatóban szereplő formátumú értékkel
- Egy tranzakció teljes élettartama alatt azonos TRID értéket kell szerepeltetni minden hozzá tartozó üzenetben
- Az authorizálandó összegnek minden, azonos tranzakcióhoz tartozó üzenetben azonosnak kell lennie (pl. egy esetleg le nem kezelt hibaág miatt előfordulhat, hogy a megerősítő üzenetben 0 összeg szerepel)

Kommunikációs problémák

Az üzenetek küldése és fogadása standard protokollon keresztül (http) történik, viszont általában nem az annak alapértelmezetten kiosztott porton (80) keresztül. Ez időtúllépéshez vezethet, ami a tranzakció meghiúsulásával illetve a teljes fizetési metódus működésképtelenségével jár. Kommunikációs hiba észlelésekor ajánlott a következők ellenőrzése:

- A banki szerver nevének helyessége a hívás helyén (a banki teszt szerver címe nem egyezik meg az éles címmel)
- A banki szerver nevének feloldhatósága a webszerver számára
- A webszerver előtti esetleges forgalomkorlátozó eszközökön (pl tűzfal) a banki szerver ip címének és cél portjának elérhetősége

A kommunikációs hiba a webszerveren parancssori eszközökkel (curl, wget) általában reprodukálható. Kérjük hibabejelentés esetén ezek kimeneteit is csatolják.

CIB BANK

Support

A bank a kereskedő számára ingyenesen biztosít a problémás tranzakciók kivizsgálásához segítséget. A vizsgálatot e-mailben kell kérni (ecommerce@cib.hu, a tárgy „Tranzakció kivizsgálás kérés <boltazonosító>”), a levélben a következőket kell feltüntetni:

- A problémás tranzakció azonosítója
- A probléma lehető legpontosabb leírása
- Screenshot, amennyiben lehetséges
- A kereskedői szerver ip címe
- A küldött/fogadott titkosított üzenetek, időpecséttel ellátva