Implementační manuál

Obsah:

1 Úvod

1.1 Co je to MojePlatba

MojePlatba je online služba přímého bankovnictví KB sloužící k tomu, aby klienti KB nakupující ve Vašem internetovém obchodu mohli své nákupy hradit online, a to přímo ze svého účtu v KB. Rozhraní služby MojePlatba je v tomto dokumentu označováno jako *platební brána KB*.

1.2 Pojmy a odkazy

Banka – Komerční banka

Bezpečnost webových aplikací – více například zde: http://en.wikipedia.org/wiki/OWASP

Certifikát – digitální certifikát v PKI. Více například zde: http://en.wikipedia.org/wiki/Public_key_infrastructure

CMS - Cryptographic Message Syntax, RFC 2630 viz http://www.ietf.org/rfc/rfc2630.txt

Internetový obchod – Váš online internetový obchod

KB - Komerční banka

Ostrá platební brána – platební brána KB v ostrém (čili produkčním) režimu, kdy probíhají opravdové finanční transakce. URL adresa Ostré platební brány je https://mojebanka.kb.cz/mojeplatba/ a přístup do ní mají po přihlášení pouze klienti KB. Rozhodně se nedoporučuje použít ve Vašem internetovém obchodu tuto adresu dříve, než proces platby plně odladíte pomocí *Ověřovací platební brány.* Stránky Ostré i Ověřovací platební brány jsou dostupné pouze při přístupu daným způsobem z Internetového obchodu, nikoli pouhým zadáním URL adresy do adresního řádku prohlížeče

Ověřovací platební brána - platební brána KB v ověřovacím (čili testovacím) režimu, kdy nedochází k žádným skutečným platbám a výsledek „platby“, který se vrací zpět do Internetového obchodu, je věrně nasimulován. Vstup do Ověřovací platební brány nevyžaduje přihlášení a je to tedy pohodlný a doporučený způsob, jak může vývojář Internetového obchodu odladit platbu službou KB MojePlatba. URL adresa je https://mojebanka.kb.cz/mojeplatba-demo/ a podobně jako Ostrá platební brána není přístupná pouhým zadáním adresy do prohlížeče

Platební brána – rozhraní služby KB MojePlatba

PKCS - Public Key Cryptography Standards. Více například zde: http://en.wikipedia.org/wiki/PKCS

PKCS#7 – veřejně přístupný kryptografický standard, který je použit pro podepisování a ověřování komunikace mezi platební bránou a Internetovým obchodem, konkrétně varianta PKCS#7 detached. Více se lze dovědět na www.rsa.com na adrese:
http://www.rsa.com/products/bsafe/documentation/certj212html/certj/dev_guide/group__CERTJ__PKCS7.html

PKI – Public Key Infrastructure. Více například zde: http://en.wikipedia.org/wiki/Public_key_infrastructure

1.3 Průběh platby přes platební bránu

Platba přes platební bránu KB v ostrém (produkčním) provozu probíhá z technického hlediska následovně:

1. Klient klepne ve Vašem Internetovém obchodě na příslušné tlačítko, čímž Internetový obchod odešle na pevně dané URL platební brány (https://mojebanka.kb.cz/mojeplatba/) HTML formulář metodou POST. Fomulář musí obsahovat správně pojmenované parametry, naplněné správnými hodnotami. Jedním z parametrů je i elektronický podpis této konkrétní platby ve formátu PKCS#7 detached, vygenerovaný Internetový obchodem.

2. Platební brána KB ověří korektnost těchto příchozích parametrů, včetně elektronického podpisu a faktu, zda Internetový obchod má k danému datu (čili v ten moment dnešnímu datu) v KB smluvně a technicky umožněno používání Ostré platební brány.

3. Platební brána zobrazí klientovi přihlašovací stránku.

4. Klient se přihlásí do platební brány.

5. Klient v platební bráně dokončí a autorizuje platbu.

6. Klient se odhlásí z platební brány příslušným tlačítkem „Odhlášení“, čímž platební brána odešle na URL Internetového obchodu HTML formulář metodou POST. Formulář bude obsahovat předem dané parametry, včetně výsledku platby, naplněné příslušnými hodnotami. Jedním z parametrů je i elektronický podpis výsledku této konkrétní platby ve formátu PKCS#7 detached, vygenerovaný platební bránou.

7. Internetový obchod ověří tyto příchozí parametry, včetně elektronického podpisu, a zobrazí klientovi stránku, ze které by měl poznat, že dokončil platbu, a to buď úspěšně, nebo naopak neúspěšně.

Pozor, může dojít i k následujícímu alternativnímu průběhu úspěšného i neúspěšného nákupu: klient namísto provedení kroku 6 výše může ukončit práci v platební bráně jinak než tlačítkem „Odhlášení“ v platební bráně, např. zavřením okna prohlížeče nebo vypršením uživatelské session (odejde na delší dobu od počítače). Internetový obchod pak nezíská informaci, zda platba byla úspěšně realizována, nebo nebyla, přestože klient mohl úspěšně zaplatit a peníze tedy Internetovému obchodu přijdou. V takovém případě ověříte platbu až po jejím zaúčtování na výpise z účtu.

Pozor, k implementaci a vyladění platby přes platební bránu KB z Vašeho Internetového obchodu doporučujeme použít nejprve **Ověřovací platební bránu**, která je pro vývoj a testování přímo určena!

2 Podrobný popis funkčnosti

2.1 Odeslání platby do platební brány

V Internetovém obchodu je třeba implementovat tlačítko, které způsobí odeslání HTML formuláře metodou POST na URL adresu platební brány, která je:

- https://mojebanka.kb.cz/mojeplatba/ v případě ostrého provozu (tj. používá se Ostrá platební brána)
- https://mojebanka.kb.cz/mojeplatba-demo/ v případě ověřovacího provozu během vývoje a testování (tj. používá se Ověřovací platební brána)

Formulář musí odesílat následující parametry:

Název parametru	Povinný	Délka parametru	Formát parametru	Význam parametru
SHOPID	Ano	Max. 10	Celé číslo nezačínající nulou	Číselné ID Vašeho internetového obchodu přesně tak, jak jste jej získali od KB
ORDERID	Ano	Max. 38	Alfanumerický – povoleny jsou číslice a velká a malá písmena bez diakritiky.	Jedinečný identifikátor transakce, generovaný Vaším internetovým obchodem. Identifikátor může být čistě číslený, ale lze použít i písmena. Identifikátory „a“ a „A“ jsou považovány za různé.
ACCOUNT	Ano	Max. 16	Celé číslo o délce max. 16 číslic Pokud Váš účet sestává z předčíslí a čísla účtu a za předčíslím následují více jak tři číslice, například 19-2735227, je potřeba pomlčku vynechat a číslo účtu doplnit zleva nulami na délku 10 znaků (číslic) následovně: 190002735227. Pokud účet obsahuje za předčíslím tři čísla, například 246-502, je potřeba pouze pomlčku vynechat, následovně 246502.	Číslo účtu Vašeho internetového obchodu pro příjem plateb z platební brány. Musí se jednat o jeden z účtů, zaregistrovaný pro tento účel v KB. Pozor, je nutné dodržet formát bez pomlčky, popsaný vlevo!
BANK	Ano	4	4 číslice, včetně případné počáteční číslice 0	Kód banky výše uvedeného účtu pro příjem plateb z platební brány. Účet včetně kódu banky musí být zaregistrován v KB
DESC	Ne	Max. 35	Povoleny jsou pouze následující znaky, spolu se znakem mezery (space): abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789-?:().,'+	Popis transakce pro příjemce - internetový obchod si může sám zvolit, co bude popisem transakce. Může to být například název či kód nakupovaného zboží či objednávky (bez diakritiky), apod
VS	Ne	Max. 10	Pouze číslice	Variabilní symbol pro platbu na výše uvedený účet pro příjem plateb z platební brány
KS	Ne	Max. 10	Pouze číslice a současně nesmí být v seznamu zakázaných KS	Konstantní symbol pro platbu na výše uvedený účet pro příjem plateb z platební brány
SS	Ne	Max. 10	Pouze číslice	Specifický symbol pro platbu na výše uvedený účet pro příjem plateb z platební brány

AMOUNT	Ano	Max. 14 číslic před desetinnou čárkou, max. 2 desetinná místa	Desetinné číslo s desetinnou čárkou či tečkou	Cena (číslo s možnou desetinnou čárkou) nakupovaného zboží. V případě, že uživatel nakupuje více zboží, je zde součet cen (celková částka za nakupované zboží)
BACKURL	Ano		Pouze platná URL adresa, začínající řetězcem http:// či https:// Parametr BACKURL musí přesně odpovídat URL adrese, kterou má Váš internetový obchod zaregistrovanou v KB, a to až do délky této zaregistrované URL adresy. Nad rámec této délky smí parametr BACKURL pokračovat již libovolně	URL adresa, na kterou je uživatel přesměrován zpět do Internetového obchodu po ukončení procesu placení v platební bráně KB.
SIGN	Ano		Base64 řetězec, v němž je zakódován elektronický podpis	Elektronický podpis privátním klíčem Internetového obchodu. Formát je PKCS#7 detached, zakódovaný kódováním base64
ai	Ano		Pro Ostrou platební bránu musí být poslána hodnota „pg001“ (bez uvozovek), pro Ověřovací platební bránu hodnota „pg001d“ (bez uvozovek). Jiná hodnota není přípustná.	Technický parametr, kterým se rozlišují funkčnosti. Slouží jako jakési potvrzení, že byla zamýšlena opravdu daná funkčnost a že nedošlo k omylu v URL, na které se formulář odesílá

Velikost písmen v názvu parametru je závazná, není tedy možné poslat například parametr „sign“ namísto parametru „SIGN“.

Parametry by měly být v HTML stránce Vašeho internetového obchodu implementovány jako hidden inputs, aby uživatel internetového obchodu neměl možnost políčka v prohlížeči editovat. **Pozor**, testovací formulář ve Vzorové implementaci má tyto parametry kvůli usnadnění testování implementovány jako text inputs, nikoli hidden inputs.

Příklad korektních hodnot parametrů:

Název parametru	**Hodnota parametru**
SHOPID	1000000003 (identifikátor obchodu, který jste obdrželi od banky e-mailem po zadministrování obchodu)
ORDERID	1000024685
ACCOUNT	123
BANK	0300
DESC	Nejaky popisek
VS	1111111111

KS	0558
SS	
AMOUNT	1550,50
BACKURL	https://www.obchod.cz/platba.html?param1=12345
ai	pg001
SIGN	MIAGCSqGSIb3DQEHAqCAMIACAQExCzAJBgUrDgMCGgUAMIAGC (cca 1900 znaků pro přehlednost vynecháno) C7Zs/t90RSdurM1UR5hg4F4Ip4yeGEtwGf8VqARpCB720EAAAAAAAA=

2.1.1 Vytvoření parametru SIGN

Nejprve je nutné vytvořit z aktuálních hodnot odesílaných parametrů textový řetězec, který vznikne zřetězením párů „název=hodnota“, oddělených znakem „|“ (ASCII 124), všech těchto parametrů (**i nepovinných**) v pevně daném pořadí:

SHOPID, ORDERID, ACCOUNT, BANK, DESC, VS, KS, SS, AMOUNT, BACKURL, ai.

Prázdná hodnota (u nepovinného parametru) je reprezentována jako prázdný řetězec způsobem „název=“.

Máme-li hodnoty parametrů jako v tabulce výše, pak výsledný textový řetězec k podepsání je:

SHOPID=1000000003|ORDERID=1000024685|ACCOUNT=123|BANK=0300|DESC=Nejaky popisek|VS=1111111111|KS=0558|SS=|AMOUNT=1550,50|BACKURL=https://www.obchod.cz/platba.html?param1=12345|ai=pg001

(řetězec neobsahuje žádné konce řádků).

Je vidět, že parametr SS není vynechán, přestože je prázdný, pouze jeho hodnota v řetězci chybí.

Tento řetězec nyní podepište privátním klíčem Vašeho obchodu ve formátu PKCS#7 detached s tím, že do PKCS#7 zahrňte (veřejný) certifikát Vašeho internetového obchodu, ale pokud možno nikoli nadřazené certifikáty v jeho certificate chainu, neboť by to zbytečně zvětšovalo velikost parametru SIGN.

Výsledný podpis PKCS#7 detached následně zakódujte pomocí kódování base64.

Konečný výsledek bude vypadat zhruba (ale ne přesně) takto – jedná se nicméně o zcela reálný příklad:

MIAGCSqGSIb3DQEHAqCAMIACAQExCzAJBgUrDgMCGgUAMIAGCSqGSIb3DQEHAQAAoIAwggQvMIIDF6ADAgECAgMAh8swDQYJKoZIhvcNAQEFBQAwVTELMAkGA1UEBhMCQ1oxFzAVBgNVBAoTDktvbWVyY25pIGJhbmthMRkwFwYDVQQLExBLQiBQS0kgRXhlY3V0aXZlMRIwEAYDVQQDEwlEQ1MgQ0EgS0IwHhcNMDgwOTI5MDgzODQ1WhcNMTAwOTI5MDgzODQ1WjBcMQswCQYDVQQGEwJDWjEQMA4GA1UEBwwHUHJhaGEgMTESMBAGA1UECwwJUHJhenNrYSAxMREwDwYDVQQLDAgxMDAwMDExMTEUMBIGA1UEAwwLUEcgS0xJRU5UIDEwgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAM8uVdkF99LNEJnVodRqyDgecCiYAfzZ4mLwArUFeSrN0jiJUS1fg8bpwCRbKAbai96IY65jiTXwdcJ/89c3euwMDCgpdqzeMq+WcFxiTyBcjGjlexLAJj5LUJ29YsnOlppGC4x0OG+XCC/zfLbqXXAhRSnl+UJfa9HiKJc7amLzAgMBAAGjggGDMIIBfzAJBgNVHRMEAjAAMBoGA1UdEQQTMBGBD2phbl93ZWJlckBrYi5jejCBuwYDVR0gBIGzMIGwMIGtBg0rgRqVzfc+hyMBDAADMIGbMHQGCCsGAQUFBwICMGgMZkNlcnRpZmlrYWNuaSBwb2xpdGlrYSBmaXJlbW5paG8gY2VydGlmaWthdHUgdnlkYW5laG8gZG8gc291Ym9ydSwgcyB2eXNva3ltIHN0dXBuZW0gb3ZlcmVuaSB0b3Rvem5vc3RpLjAjBggrBgEFBQcCARYXaHR0cDovL3d3dy5tb2plYmFua2EuY3owGwYKK4Ealc33PgEEAAQNMAsECTE0NTE2MDAxMzALBgNVHQ8EBAMCBPAwbgYDVR0jBGcwZaFgpF4wXDELMAkGA1UEBhMCQ1oxFzAVBgNVBAoTDktvbWVyY25pIGJhb

mthMR8wHQYDVQQLExZEaXJlY3QgQ2hhbm5lbCBTeXN0ZW1zMRMwEQYDVQQDEwpST09UIENBIEtCggEGMA0GC
SqGSIb3DQEBBQUAA4IBAQBp9muF7M/rdyt8kYulvlzj9gtMnIrdEprMD/tmUcilC4uav+wB6ykqHOsrm0SYGeVZUKZ/HgyL
DehaG0sfAgRizZD1hZPtMMPprvufU+3B3xViPJM9dHvrjyLEPqRCkuWas48/z42cj18FKLCteWArxGj3egKDUlPdl9R55/tS4
wvLmBTc6qMmGCtvHYgMbEx/50orryDKYpvV0kp805Xw+O+MLHb2JdjcMNl3jmWc8ltPdbYK6Sx1r4ma7tUuhgLbY6qiYc
VkKq5aivRXsMZKy8xeYF5D6xNkj3Y5QZwsNxTF05920wdMPB3r05dy/SzT88jvJbg/p0XX40DnhYNTAAAxggFhMIIBXQIB
ATBcMFUxCzAJBgNVBAYTAkNaMRcwFQYDVQQKEw5Lb21lcmNuaSBiYW5rYTEZMBcGA1UECxMQS0IgUEtJIEV4ZW
N1dGl2ZTESMBAGA1UEAxMJRENTIENBIEtCAgMAh8swCQYFKw4DAhoFAKBdMBgGCSqGSIb3DQEJAzELBgkqhkiG9
w0BBwEwHAYJKoZIhvcNAQkFMQ8XDTA4MTAxNTE0MDkwNlowIwYJKoZIhvcNAQkEMRYEFNF8xPYmIRUXzwaMqhnj
nUlyUOpVMA0GCSqGSIb3DQEBAQUABIGAWc7vevi02BElOOOTCZlBD/vZn+AzYI1buhLgC83H5SCfHtPRGX7/Y1n33C
dkQeKYDCLq74wONTspp52HDshYDjTYc/pVq84Z8DGoRRnjB2V1v8Aj81+euvVngtkrbEk3vSBkPOeqaKLyJ39ARACxBq
ccszuPIEMC4f4JXZa9Yr0AAAAAAAA=

a tento konečný výsledek použijte jako hodnotu parametru SIGN.

2.2 Příjem výsledku platby z platební brány

platební brána KB po ukončení platby, kdy klient klepne na příslušné tlačítko, odešle HTML formulář metodou POST na adresu BACKURL Vašeho Internetového obchodu.

Formulář bude obsahovat stejné hodnoty následujících parametrů, které odeslal Internetový obchod:

SHOPID, ORDERID, ACCOUNT, BANK, DESC, VS, KS, SS, AMOUNT, BACKURL, ai

a navíc ještě parametry RESULT a BANK_SIGN.

Popis jednotlivých parametrů:

Název parametru	**Povinný**	**Formát parametru**	**Význam parametru**
SHOPID	Ano	(zopakovaná hodnota, odeslaná Internetovým obchodem)	
ORDERID	Ano	(zopakovaná hodnota, odeslaná Internetovým obchodem)	
ACCOUNT	Ano	(zopakovaná hodnota, odeslaná Internetovým obchodem)	
BANK	Ano	(zopakovaná hodnota, odeslaná Internetovým obchodem)	
DESC	Ne	(zopakovaná hodnota, odeslaná Internetovým obchodem)	
VS	Ne	(zopakovaná hodnota, odeslaná Internetovým obchodem)	
KS	Ne	(zopakovaná hodnota, odeslaná Internetovým obchodem)	
SS	Ne	(zopakovaná hodnota, odeslaná Internetovým obchodem)	
AMOUNT	Ano	(zopakovaná hodnota, odeslaná Internetovým obchodem)	
BACKURL	Ano	(zopakovaná hodnota, odeslaná Internetovým obchodem)	
ai	Ano	(zopakovaná hodnota, odeslaná Internetovým obchodem)	

RESULT	Ano	Hodnota „SUCCESS“ (bez uvozovek) právě tehdy, pokud platba uspěla a klient nepřerušil práci tím, že by již neklepnul na poslední tlačítko v platební bráně (neodešel od počítače apod.). Hodnota „FAILED“ (bez uvozovek) právě tehdy, pokud platba neuspěla a klient nepřerušil práci tím, že by již neklepnul na poslední tlačítko v platební bráně (neodešel od počítače apod.). Jiná hodnota není přípustná.	Výsledek platební transakce z platební brány KB.
BANK_SIGN	Ano	Base64 řetězec, v němž je zakódován elektronický podpis	Elektronický podpis privátním klíčem platební brány KB. Formát je PKCS#7 detached, zakódovaný kódováním base64

2.2.1 Jak vzniká parametr BANK_SIGN

Parametr BANK_SIGN je elektronický podpis banky privátním klíčem platební brány KB nad obdobně sestavným řetězcem, jako byl parametr SIGN výše. Hodnota parametru BANK_SIGN vznikne následujícím způsobem:

Nejprve je vytvořen z aktuálních hodnot následujících parametrů textový řetězec, který vznikne zřetězením párů „název=hodnota“, oddělených znakem „|“ (ASCII 124), těchto parametrů v pevně daném pořadí:

SHOPID, ORDERID, ACCOUNT, BANK, DESC, VS, KS, SS, AMOUNT, BACKURL, ai, RESULT.

Máme-li hodnoty parametrů jako v předchozí kapitole a platba bude úspěšná, pak výsledný textový řetězec k podepsání (tj. následně i k ověření Internetovým obchodem) bude:

SHOPID=1000000003|ORDERID=1000024685|ACCOUNT=123|BANK=0300|DESC=Nejaky popisek|VS=1111111111|KS=0558|SS=|AMOUNT=1550,50|BACKURL=https://www.obchod.cz/platba.html?param1=12345|ai=pg001|RESULT=SUCCESS

(řetězec neobsahuje žádné konce řádků).

Je vidět, že prázdná hodnota, v tomto případě parametru SS, je reprezentována jako prázdný řetězec způsobem „název=“.

Tento řetězec nyní banka podepíše privátním klíčem platební brány KB ve formátu PKCS#7 detached s tím, že do PKCS#7 povinně zahrne přinejmenším (veřejný) certifikát platební brány KB, a volitelně případně i nadřazené certifikáty v jeho certificate chainu (nejednalo by se o chybu, pouze o méně optimální variantu).

Tento podpis PKCS#7 detached následně banka zakóduje pomocí kódování base64 a použije jako hodnotu odesílaného parametru BANK_SIGN.

Reálný příklad podoby parametry BANK_SIGN:

MIIOGgYJKoZIhvcNAQcCoIIOCzCCDgcCAQExCzAJBgUrDgMCGgUAMAsGCSqGSIb3DQEHAaCCDJ8wggQOMIIC9qADAgECAgMAg78wDQYJKoZIhvcNAQEFBQAwVTELMAkGA1UEBhMCQ1oxFzAVBgNVBAoTDktvbWVyY25pIGJhbmthMRkwFwYDVQQLExBLQiBQS0kgRXhlY3V0aXZlMRIwEAYDVQQDEwlEQ1MgQ0EgS0lwHhcNMDgwNjA5MDkxMzM3WhcNMDkwNjA5MDkxMzM3WjBdMQswCQYDVQQGEwJDWjEXMBUGA1UEChMOS29tZXJjbmkgYmFua2ExGTAXBgNVBAsTEEtCIFBLSSBFeGVjdXRpdmUxGjAYBgNVBAMTEUtCIFBsYXRlYm5pIGJyYW5hMIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQC3DVQ9H9rhGwDW731nUKF321dLwQ9UtspgPVjt392vpMqpQXPt75Np8D+6NJoFCR53PVjlqwCysRufcy8JvPR3WYRd2odSfnu2uMXkqIEZQojsZhHbUAXQWZIg60B+rwdHGE0lRSBYVoyZ/ePH9MzwQXJOLqXKKKMXoGAHqwIRMwIDAQABo4IBYTCCAV0wCQYDVR0TBAIwADAYBgNVHREEETAPgQ1pbmZvX2NhQGtiLmN6MIGZBgNVHSAEgZ

EwgY4wgYsGDSuBGpXN9z6BAwELAwIwejBTBggrBgEFBQcCAjBHGkVDZXJ0aWZpa2FjbmkgcG9saXRpa2EgY2VydGl
maWthdHUgcHJvIHBvZHBpc292eSBtb2R1bCB2eWRhbmVobyBkbyBIU00wIwYIKwYBBQUHAgEWF2h0dHA6Ly93d3cu
bW9qZWJhbmthLmN6MAsGA1UdDwQEAwIGwDBuBgNVHSMEZzBloWCkXjBcMQswCQYDVQQGEwJDWjEXMBUGA1
UEChMOS29tZXJjbmkgYmFua2ExHzAdBgNVBAsTFkRpcmVjdCBDaGFubmVsIFN5c3RlbXMxEzARBgNVBAMTClJPT1
QgQ0EgS0KCAQYwHQYDVR0OBBYEFAH83A3Bo5r9NP0KxRB2l5Lwf1GgMA0GCSqGSIb3DQEBBQUAA4IBAQCRT6+
6LQnyH60hM/kPunWi6C84jSKr6jYxhlb89scCVTNXdX0RWDnDJ+qa9yIWrMvg86czwuJyHAJxkflQ/TfJlJTn70kqkUU6aqW
NE4/dyV0T41XH1MC8OLyK8mVmbvwEjTVe4Xhn70rKuhQ4P9SaS7OlhoIrOKObsOh9sltVVJjVQg8gnVqJlJzuijHaEJwXq
BxI4DIY+GEVOwV28JIbrMmA/Tj+Bmy6CAE3bQMLAQfu3WaJ5g64qDf4oZUOtVUpwCjahMIZ96SReZj6kSzUc3ZRSsa7D
vNTaoMgk8oFjYyNIiBV3Ooo2OJBhBBDEe0mEVBsowstGHdYcoD0NloSMIIEhzCCA2+gAwIBAgIBBjANBgkqhkiG9w0BA
QUFADBcMQswCQYDVQQGEwJDWjEXMBUGA1UEChMOS29tZXJjbmkgYmFua2ExHzAdBgNVBAsTFkRpcmVjdCBDa
GFubmVsIFN5c3RlbXMxEzARBgNVBAMTClJPT1QgQ0EgS0IwHhcNMDMwNTE2MDYxODUzWhcNMTMwNTE2MDYxO
DM1WjBVMQswCQYDVQQGEwJDWjEXMBUGA1UEChMOS29tZXJjbmkgYmFua2ExGTAXBgNVBAsTEEtCIFBLSSBFe
GVjdXRpdmUxEjAQBgNVBAMTCURDUyBDQSBLQjCCASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBAL3z9N
fOiksEqLGRevWb5DPtOoTJD2uV8wQ4zNUVV1k+EKxiPoXGH9hJaRpN3YoGwy3LElTB4rT47n5RzzM43Va2cehZneDBa
+fZdC+MYVmlDwQNHDv8xah1SV2UER3ao8xdAZesqurkEt9oNuaa9azhtw+Z1zuQWmSG8NkZCBnNObzR5PSeq1Wysb
8D33EqfuJ/UxbBsYLXaufNdvPmI79U/ApBLeef76UZpBsnHQANDbTvd0M7fdCVtrS3r4Z4WoZVCqx9XciiSJ68VVt3e1Pfst
ntlwq195AhXncwXTwrvabyK7IwpDypzSGpfgKMxrwHF1fR83WKUoOZb1WNtMcCAwEAAaOCAVkwggFVMA8GA1UdEw
QIMAYBAf8CAQIwgYsGA1UdIASBgzCBgDB+BgwrgRqVzfc+HwEtAwAwbjBHBggrBgEFBQcCAjA7GjIDZXJ0aWZpa2Fjb
mkgcG9saXRpa2EgLSBwb2RyaXplbmEgY2VydGlmaWthY25pIGF1dG9yaXRhLiAwIwYIKwYBBQUHAgEWF2h0dHA6Ly
93d3cubW9qZWJhbmthLmN6MA4GA1UdDwEB/wQEAwIBxjCBhAYDVR0jBH0we4AUD68A0jg7Lwf8B8Y+OJ9V+0gjKTm
hYKReMFwxCzAJBgNVBAYTAkNaMRcwFQYDVQQKEw5Lb21lcmNuaSBiYW5rYTEfMB0GA1UECxMWRGlyZWN0IENo
YW5uZWwgU3lzdGVtczETMBEGA1UEAxMKUk9PVCBDQSBLQoIBATAdBgNVHQ4EFgQUCWDmcBndIt9LWkZwo3wUL
WTkC2AwDQYJKoZIhvcNAQEFBQADggEBAF9tJvDt1a/imcPII7WF0lV3nEWb25cQLdgVZAg3TC5u+/3aq5NEJlqnFplXdE
kYnwAGeL8PiFeJwJI1myjQXTtqD4quKya6wTXZcDJdOL+y+fC9ETu7S2GC0B9/mO3ou1xdR+20i7HNMorUJz1huZ+38+e
+j3o5mFEEiaeXbGdHNbNd1sEwhC3kTxSR9caz6aGNXP58uNL7gvjvU9HsHiioSe+V46RXNkxPu0/qfvQGvwG1xZYZlox5
J61EbOgewqz/ehcqPqMsVEKwSyV7h1117vo0jzyutLUsYArNE1kQxy2lC4/JGc/DMCx1tr8I7X2e4WWFkgmbyhxZne50vq4
wggP+MIIC5qADAgECAgEBMA0GCSqGSIb3DQEBBQUAMFwxCzAJBgNVBAYTAkNaMRcwFQYDVQQKEw5Lb21lcmN
uaSBiYW5rYTEfMB0GA1UECxMWRGlyZWN0IENoYW5uZWwgU3lzdGVtczETMBEGA1UEAxMKUk9PVCBDQSBLQjAe
Fw0wMzA1MTYwNjAwMjRaFw0yMzA1MTYwNTU5MTVaMFwxCzAJBgNVBAYTAkNaMRcwFQYDVQQKEw5Lb21lcmNu
aSBiYW5rYTEfMB0GA1UECxMWRGlyZWN0IENoYW5uZWwgU3lzdGVtczETMBEGA1UEAxMKUk9PVCBDQSBLQjCCA
SIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBAJgFZeJJ7kefMbqYCLLyWMJM8Y66Xjau6sOyZ1ho2OYGzD/fnX
0XYpbwyQOctCtPRf7X/u7EFqEeGB7vLjyVcQj6LfPuGRBnU/tYb++yf+RXJEKOBdJIfqNVhdCtKRPli3yvLKSY04RVRbz2Y
a2+6f/aOVDILHgwTa4Wy1jdsLwPRCWzuo8BhLaMTFsJ+tS7rAt/TZUW8m5IFmtBkw6/jOxe6FMkont3lw85h/LtQrRzhSHe
DfVmGgcL6ppdIhy7yY8aRs37ZbRLh6h6CzWnYGBxgpBKXVVMc+p6s0GCIacKE97zgCRb9nMHkE0Gzwuk7fMA+IwNn3
gFy3qmHBuaX4cCAwEAAaOByjCBxzASBgNVHRMBAf8ECDAGAQH/AgEDMIGBBgNVHSAEejB4MHYGDCuBGpXN9z4
fASwDADBmMD8GCCsGAQUFBwICMDMaMUNlcnRpZmljYXRlIFBvbGljeSAtIFJvb3QgQ2VydGlmaWNhdGlvbiBBdXRob
3JpdHkwIwYIKwYBBQUHAgEWF2h0dHA6Ly93d3cubW9qZWJhbmthLmN6MA4GA1UdDwEB/wQEAwIBxjAdBgNVHQ4E
FgQUD68A0jg7Lwf8B8Y+OJ9V+0gjKTkwDQYJKoZIhvcNAQEFBQADggEBAF9cEFEsa8cYutSTAzD8IzBZIUlegpVa8UeC
f0ZSZG+78SA+nK62JtqMTJgZmADDQV5JpiEA+Lzw4IpVB03iR1IsXLZRik2HOdWPAXSo17k2kadUTSmAb30kihmlJA38
0sjQ4kX9/6HmqfnQbmlmCwTmQZ+mQMLflyGBd80Fh6CY9nCFM34eJhdDDhJlibY9JpjexzRVKjo5S86ZAwEiXOHLZdVje
SQsVfLIjhlNfbOGcG0dOCv07hQxqdPjdAECkU/NwFjzDEZmvOGRYXkfRseB+gxI92CQ8509WNhbRIQUHyrpypvfUHn9Iy
QQ7IId2/D38po8RwH5pu4FS0TuloMxggFDMIIBPwIBATBcMFUxCzAJBgNVBAYTAkNaMRcwFQYDVQQKEw5Lb21lcmN
uaSBiYW5rYTEZMBcGA1UECxMQS0IgUEtJIEV4ZWN1dGl2ZTESMBAGA1UEAxMJRENTIENBIEtCAgMAg78wCQYFK
w4DAhoFAKA/MBgGCSqGSIb3DQEJAzELBgkqhkiG9w0BBwEwIwYJKoZIhvcNAQkEMRYEFDad96Fp4DJLBaznUpyHe
Mgzq8CAMA0GCSqGSIb3DQEBAQUABIGAXXOG6IDCXmc7uTgF9i+D26sYXXULoNnI7r3qs2iEvdmVjAswGHB2FgaH40
e/iYYnUhLI1B6UvsQECzKEel2w4LU04PSV1LXk3e1lqdWDVW93VYJLT798Bh0tpB1yl2hPcQCIK26ZOXBB5Vi8gJ3mp3B
9usZpHTIVi9uGAdrZ5zs=

2.2.2 Ověření příchozích parametrů

Internetový obchod musí porovnat, zda hodnoty příchozích parametrů z platební brány **SHOPID, ORDERID, ACCOUNT, BANK, DESC, VS, KS, SS, AMOUNT, BACKURL, ai** jsou rovny hodnotám, které Internetový obchod původně odeslal do platební brány. Dále je nutno ověřit platnost parametru BANK_SIGN.

2.2.3 Jak ověřit parametr BANK_SIGN

Je potřeba provést verifikaci PKCS#7 detached signatury náležitým způsobem (viz např. vzorové implementace Java / PHP / ASP.NET nebo viz přímo specifikace PKCS#7) s tím, že data, jejichž podpis se ověřuje, musí být sestavena Vaším internetovým obchodem z názvů a hodnot parametrů SHOPID, ORDERID, ACCOUNT, BANK, DESC, VS, KS, SS, AMOUNT, BACKURL, RESULT přesně způsobem, uvedeným výše. Ověřujete tedy podpis řetězce, podobného následujícímu:

SHOPID=1000000003|ORDERID=1000024685|ACCOUNT=123|BANK=0300|DESC=Nejaky popisek|VS=1111111111|KS=0558|SS=|AMOUNT=1550,50|BACKURL=https://www.obchod.cz/platba.html?param1=12345|ai=pg001|RESULT=SUCCESS

Pro ověření podpisu je **nutné** použít Vaši vlastní důvěryhodnou kopii veřejného certifikátu platební brány KB. Tj. certifikát, který jste získali předem a **nikoli** pouze extrahovali z příchozího PKCS#7 podpisu z platební brány KB. Certifikát pro ověření podpisu získáte na adrese https://mojebanka.kb.cz//file/cs/KB_Platebni_brana.crt To, že je certifikát důvěryhodný, mimo jiné znamená, že certifikát je platný a opravdu byl vydán příslušnou certifikační autoritou (CA) Komerční banky. Ověření platnosti certifikátu platební brány KB není popsáno v tomto dokumentu.

2.2.4 Jak pokračovat

Po zpracování a ověření příchozích parametrů by se měl uživatel dovědět, jak jeho platba dopadla a vidět rekapitulaci nákupu a platby spolu s dalšími kroky, které jej čekají.

Pozor, příchozí parametry je nutné důsledně prověřit tak, aby nemohlo dojít ke zneužití mechanizmu platby pomocí služby MojePlatba ve vašem Internetovém obchodu. Internetový obchod by si měl ukládat platby například do databáze a při přijetí platby z platební brány porovnat na základě identifikátoru ORDERID, zda výsledek platby objednávky s tímto ORDERID byl přijat se stejnými parametry, jako byl odeslán.

Pozor, může dojít i k možnosti, že klient nedokončí správně platbu v platební bráně a Internetovému obchodu vůbec nepřijde odpověď z platební brány. V takovém případě jsou možné oba výsledky: platba proběhla úspěšně, platba neproběhla úspšně. Toto je již zmíněno výše v tomto dokumentu v kapitole **Průběh platby přes platební bránu**. V takovém případě ověříte platbu až po jejím zaúčtování na výpise z účtu.

Pozor, není v žádném případě vhodné jakkoli podcenit aspekt bezpečnosti aplikace. Více například na http://en.wikipedia.org/wiki/OWASP.

2.2.5 Spuštění ostrého provozu

Pozor, internetový obchod má jako výchozí nastavení v systémech KB zapnutý pouze ověřovací provoz z důvodů bezpečnosti a omezení vzniku chyb.

Jakmile máte otestovanou implementaci proti ověřovací platební bráně, kontaktujte linku podpory přímého bankovnictví na emailové adrese mojeplatba@kb.cz. KB povolí ostrý provoz a o nastavení vás bude informovat. Po nastavení ostrého provozu budete mít k dispozici i ověřovací provoz.

2.2.6 Obchodní podpora služby MojePlatba

Součástí implementační sady jsou i bannery a tlačíka pro podporu služby MojePlatba. Pokud vystavujete banner na stránky bez přesměrování do služby MojePlatba (například na úvodní stránce nebo v sekci Platební metody), vložte na banner odkaz na stránky produktu MojePlatba https://www.kb.cz/cs/ostatni/nase-aplikace/aplikace/mojeplatba

3 Vzorové implementace

Na instalačním CD jsou k dispozici vzorové implementace pro platformy **Java, PHP a ASP.NET**. Lze je použít jako výchozí bod pro implementaci platby přes platební bránu KB do Vašeho internetového obchodu.

Velmi užitečnou a doporučovanou službou je **Ověřovací platební brána KB t** (https://mojebanka.kb.cz/mojeplatba-demo/), kterou lze de facto neomezeně používat k věrné simulaci platby přes Ostrou platební bránu (https://mojeplatba.kb.cz/) (z technického hlediska), aniž by vývojář či tester musel být klientem KB a aniž by v důsledku toho docházelo k jakýmkoli platbám. Pozor, v ostrém provozu nezapomeňte přepnout Vás Internetový obchod na URL Ostré platební brány a nezapomeňte posílat správnou hodnotu technického parametru **ai** tak, jak je uvedeno v kapitole **Odeslání platby do platební brány**!

Pozor, aplikace jsou sice plně funkční, ale i tak je nutné dodržet postup, že nejprve s pomocí vzorové aplikace (nebo i bez ní) vytvoříte implementaci platby přes platební bránu KB ve Vašem vlastním internetovém obchodu (eshopu), tu následně řádně otestujete (funkčnost, bezpečnost, výkonnost), stejně jako každou jinou aplikaci, a teprve poté použijete v ostrém provozu. Ve vzorových implementacích je uvedena url https://test.mojeplatba.cz, pro využití pro ověřovací a ostrou platební bránu je nutné v implementaci změnit url na https://mojebanka.kb.cz.

Pozor, věnujte maximální pozornost bezpečnosti uložení certifikátů a privátního klíče!

3.1 Vysvětlivky, nástroje

Base64 kodér a dekodér – online např. zde: http://www.motobit.com/util/base64-decoder-encoder.asp

Dumpasn1 - nástroj pro uživatelské zobrazení binárních ASN.1 dat: http://www.cs.auckland.ac.nz/~pgut001/dumpasn1.c

X.509 - Public Key Infrastructure Certificate

CER - Canonical Encoding Rules - formát souboru s certifikátem X.509 v kódování Base-64

DER - Distinguished Encoding Rules - formát souboru s certifikátem X.509 v binárním kódování DER

PEM - Privacy Enhanced Mail - formát souboru s certifikátem X.509 v kódování Base-64

PKCS#12 - Personal Information Exchange Syntax Standard – formát souboru s certifikátem X.509, více např. zde: http://en.wikipedia.org/wiki/PKCS12

3.2 Vzorová implementace Java

Vzorová implementace v jazyku Java je umístěna na instalačním CD v adresáři „implementace\shop_java“. Je zde k dispozici vzorový webový archiv eshop.war, který obsahuje i kompletní zdrojový kód a potřebné .jar knihovny. Nedílnou součástí je i knihovna bcpkix-jdk15on-154.jar, bcprov-jdk15on-154.jar, nacházející se na instalačním CD v adresáři „implementace\shop_java\jre_lib_ext“. Tato knihovna pochází z projektu **BouncyCastle.org** a tedy i podléhá příslušné licenci, stejně jako knihovna WEB-INF\lib\bcmail-jdk15on-1.54.jar, obsažená přímo v archivu eshop.war. Knihovna WEB-INF\lib\commons-io-2.4.jar pochází z projektu Apache Commons a také podléhá příslušné licenci.

Webový archiv eshop.war lze importovat jako projekt do vývojového prostředí (Eclipse, IntelliJ IDEA, NetBeans atd.). Webový archiv lze také nainstalovat jakožto standardní WAR soubor na aplikační server, případně servlet container, jako je například Apache Tomcat nebo IBM WebSphere.

Prosím, prostudujte si **dokumentaci JavaDoc** ve zdrojových souborech .java (v archivu eshop.war)!

Nejdůležitější třídy v projektu jsou následující:

- **PGOrder** – reprezentuje platbu, obsahuje její parametry a sestavuje řetězce k podpisu a ověření podpisu

- **PGSigner** – zajišťuje vytvoření a ověření PKCS#7 detached. Není závislá na konkrétní implementaci security algoritmu

- **PGSignerBCProv** – zajišťuje implementaci vytvoření a ověření PKCS#7 detached pomocí knihoven BouncyCastle.org

- **PGConfig** – singleton třída, ze které ostatní třídy a JSP získávají nastavení, certifikáty či privátní klíč. Při prvním použití si třída PGConfig načte konfiguraci ze souboru *eshop.properties* a následně certifikáty a privátní klíč, dané touto konfigurací. Toto není už dále znovu načítáno a proto při změně konfigurace je potřeba následně restartovat aplikaci.

I ostatní třídy jsou samozřejmě potřebné, s výjimkou PGTester, která je vhodná pouze pro případné pokusy. Všechny třídy se nacházejí v package *cz.kb.eshopsample*.

3.2.1 Minimální verze Java prostředí, potřebné pro běh eshop.war

- **Java** 1.6
- **Servlet** 3.0
- **JSP** 2.2

3.2.2 Konfigurace aplikace eshop.war

Uvnitř eshop.war je **konfigurační soubor WEB-INF\classes\eshop.properties**, jehož obsah vypadá zhruba takto:

```
CERT_PATH=c:/eshop/conf/eshop_certifikat.p12
#####################################################################
# !!! POZOR, v zadnem pripade nesmite v citelne podobe do souboru
#     ulozit heslo ke svemu certifikatu s privatnim klicem !!!
#####################################################################
CERT_PASSWORD=c:/eshop/conf/somefile.txt
TRUST_CERT_PATH=c:/eshop/conf/KB_Platebni_brana.cer
RANDOM_ORDER_ID=Y
RANDOM_AMOUNT=N
BANK_TARGET_URL_1=https://mojebanka.kb.cz/mojeplatba/
BANK_TARGET_URL_2=https://mojebanka.kb.cz/mojeplatba-demo/
SIGNATURE_ALGORITHM=SHA256withRSA
#toto je jen komentar, aplikace jej ignoruje
```

Soubor eshop.properties extrahujte a zeditujte v textovém editoru s použitím správných hodnot pro Vaše prostředí. Následně soubor zapište zpět do eshop.war, čímž přepíšete původní soubor WEB-INF\classes\eshop.properties v eshop.war, případně jinak zajistěte, že se bude do aplikace načítat nový konfigurační soubor.

Popis konfiguračních parametrů v eshop.properties:

Parametr	Význam
CERT_PATH	Plná cesta k souboru s certifikátem (včetně privátního klíče) Vašeho eshopu na souborovém systému serveru. Jedná se o lokální cestu, tedy na tom serveru, kde běží aplikační server s Vaším eshopem. Soubor je nutno na server přesně na tuto cestu nakopírovat a aplikační server musí mít do adresáře právo ke čtení. Soubor musí být ve formátu PKCS#12, jinak jej tato vzorová implementace nenačte. **Pozor** na bezpečnost uložení certifikátu. **Zvolíte-li jako zde souborový systém, je potřeba nastavit k souboru oprávnění tak, aby běžný uživatel neměl k souboru přístup, aby se soubor neocitl ve struktuře, stáhnutelné přes WWW, a aby proces aplikačního (případně webového) serveru měl přístup pouze ke čtení.** Máte-li možnost používat bezpečnější variantu vytváření a ověřování elektronického podpisu, například pomocí hardware security modulu (HSM), doporučujeme ji použít.
CERT_PASSWORD	Cesta k souboru, který obsahuje zašifrované heslo k PKCS#12 souboru s certifikátem, který je dán v parametru CERT_PATH. Zašifrované heslo (ve formátu base64) získáme jednoduše spuštěním metody cz.kb.eshopsample.PGEncrypter.main(), přičemž jako parametr v příkazové řádce zadáme naše heslo a výsledek uložíme do souboru s cestou danou parametrem CERT_PASSWORD. **Pozor, nedoporučuje se ukládat hesla do souboru v čitelné, nebo pro třetí osobu jinak dešifrovatelné podobě**. Je zde bezpečnostní riziko zneužití privátního klíče Vašeho certifikátu nepovolanou osobou (hackerem), což by Vám mohlo způsobit škody. Použijte proto pokud možno co nejbezpečnější možný mechanizmus pro vytváření a ověřování elektronického podpisu, který máte k dispozici. **Zvolíte-li jako zde souborový systém, je potřeba nastavit**

	k souboru oprávnění tak, aby běžný uživatel neměl k souboru přístup, aby se soubor neocitl ve struktuře, stáhnutelné přes WWW, a aby proces aplikačního (případně webového) serveru měl přístup pouze ke čtení.
TRUST_CERT_PATH	Plná cesta k souboru s důvěryhodným certifikátem KB platební brány na souborovém systému serveru. Jedná se o lokální cestu, tedy na tom serveru, kde běží aplikační server s Vaším eshopem. Certifikát je nutno na server přesně na tuto cestu nakopírovat a aplikační server musí mít do adresáře právo ke čtení. Soubor musí být v binárním formátu DER, jinak jej tato vzorová implementace nenačte. Přípona souboru často bývá „.cer“. **Pozor** na bezpečnost uložení certifikátu. Případný útočník (hacker) se může pokusit na souborový systém podstrčit soubor s jiným certifikátem. **Zvolíte-li jako zde souborový systém, je potřeba nastavit k souboru oprávnění tak, aby běžný uživatel neměl k souboru přístup, aby se soubor neocitl ve struktuře, stáhnutelné přes WWW, a aby proces aplikačního (případně webového) serveru měl přístup pouze ke čtení.**
RANDOM_ORDER_ID	Hodnota „Y“ znamená, že parametr ORDERID, čili ID příkazu, je při každém novém zobrazení formuláře generován náhodně. Pokud nastavíme hodnotu „N“, musíme ID příkazu editovat ručně. Použití náhodného ID v ostrém (produkčním) provozu není korektní, jedná se pouze o prostředek k testování. V produkčním Internetovém obchodu je potřeba zaručit unikátnost parametru ORDER_ID po celou dobu existence Vašeho Internetovém obchodu. Toho lze dosáhnout použitím sekvence v databázi, nebo jiného generátoru posloupnosti unikátních čísel, který bude fungovat korektně i ve vícevláknové (multithreaded) aplikaci.
RANDOM_AMOUNT	Hodnota „Y“ znamená, že parametr AMOUNT, čili částka v korunách, je při každém novém zobrazení formuláře generován náhodně. Pokud nastavíme hodnotu „N“, je potřeba částku ve formuláři editovat ručně. V produkčním provozu samozřejmě použití náhodné částky nepřipadá v úvahu, jedná se pouze o prostředek k testování.
BANK_TARGET_URL_1	**URL**, na které se odesílá formulář. Toto URL slouží pro **Ostrou platební bránu KB**, čili pro produkční (ostrý) provoz.
BANK_TARGET_URL_2	**URL**, na které se odesílá formulář. Toto URL slouží pro **Ověřovací platební bránu KB**, čili pro ověřovací (testovací) provoz.
BANK_TARGET_URL_X	(kde X je číslo od 3 do 10) Alternativní URL, na které se odesílá formulář. Zde specifikujte další URL, kam chcete formulář v rámci vývoje či testování posílat, například na Váš vlastní testovací server. Nemělo by to být ale nutné, neboť máte-li k dispozici připojení k internetu, lze k tomuto účelu použít **Ověřovací platební bránu KB**.
SIGNATURE_ALGORITHM	Algoritmus pro vytvoření podpisu. Preferovaný je „SHA256withRSA“ přičemž jistou dobu bude podporován také „SHA1withRSA“. Když hodnota v konfiguračním souboru nebude uvedena,

	použije se automaticky „SHA256withRSA“.

3.2.3 Příprava prostředí Java

Nakopírujte knihovnu jre_lib_ext**bcpkix-jdk15on-154.jar a** jre_lib_ext**bcprov-jdk15on-154.jar** z instalačního CD do podadresáře **jre/lib/ext/** v adresáři Javy, kterou používá vás aplikační server ke svému běhu.

Pokud má Vaše JVM omezenu sílu klíče, mělo by pomoci stažení a nainstalování **unlimited JCE (Java Cryptography Extension) policy** od společnosti, dodávající Vaši JVM (Sun, IBM, Oracle/BEA, ...).

Pozor, pokud na vývojovém počítači spouštíte jak aplikační server, tak i webový prohlížeč se službou MojePlatba či Mojebanka, může přestat fungovat přihlašovací Java aplet těchto služeb. Problém se může objevit, jestliže jak prohlížeč, tak i aplikační server sdílejí stejnou instalaci Javy. Konfigurace při vývoji vzorové implementace byla následující:

- Windows 7 Professional SP1
- Apache Tomcat 7.0.67 (v adresáři c:\tomcat) nad Sun JDK 1.6.0_27 (v adresáři c:\jdk1.6.0_27)
- MSIE 7 nad Sun JRE 1.8.0_65 (v adresáři c:\Program Files\Java\jre1.8.0_65)
- Knihovna bcpkix-jdk15on-154.jar a bcprov-jdk15on-154.jar v adresáři c:\jdk1.6.0_27\jre\lib\ext\ (nikoli tedy v Javě, kterou používá prohlížeč)

3.2.4 Instalace eshopu na aplikační server

Již nakonfigurovaný war soubor eshop.war nainstalujeme na aplikační server a zvolíme dosud nepoužitý Context Root, například “/eshop” (bez uvozovek).

Ujistíme se, že soubory s certifikáty dané konfiguračními parametry CERT_PATH a TRUST_CERT_PATH jsou na serveru skutečně přítomny, případně že jsou k dispozici na bezpečnostním zařízení, z kterého jej čteme.

URL aplikace eshop.war bude vypadat následovně: http://server:port/contextroot/

Konkrétně například:
http://127.0.0.1/eshop/
https://testovaci.server.mojefirma.cz:8080/testApps/demoEshop/
atd.

Zadejte URL aplikace do prohlížeče a můžete začít používat testovací formulář.

3.2.5 Pokud eshop.war neběží

Pokud se namísto úvodní stránky eshopu na výše zmíněném URL zobrazí Java výjimka, ujistěte se o správné instalaci knihovny **bcpkix-jdk15on-154.jar a bcprov-jdk15on-154.jar** (viz výše).

Pokud je knihovna tam, kde má být, **je možné, že Vaše JVM má omezení na sílu bezpečnostního klíče a algoritmu**. Tehdy doporučujeme řídit se údaji od dodavatele JVM (Sun, IBM, ...). Například následující Java výjimka (exception):

java.lang.SecurityException: Unsupported keysize or algorithm parameters

by měla být řešitelná stažením a nainstalováním **unlimited JCE (Java Cryptography Extension) policy** od společnosti, dodávající JVM (Sun, IBM, Oracle/BEA, ...). Tím se odstraní omezení na sílu (velikost) bezpečnostního klíče a algoritmu.

Pokud aplikace stále neběží ani po restartu aplikačního serveru, zkontrolujte další možné problémy, např. s načítáním konfiguračního souboru apod.

Pokud ani nyní aplikace nefunguje, konzultujte prosím domovskou stránku projektu **BouncyCastle.org**, ze kterého pochází knihovna **bcpkix-jdk15on-154.jar a bcprov-jdk15on-154.jar.**

3.2.6 Poznámky

Pozor, v eshop.war není při návratu z platební brány do Internetového obchodu, tj. na stránce target.jsp, ověřována session ani rovnost odesílaných parametrů **SHOPID, ORDERID, ACCOUNT, BANK, DESC, VS, KS, SS, AMOUNT** a **BACKURL** s nazpět přijatými. Toto by měl reálný Internetový obchod implementovat.

Pozor, v eshop.war je formulář na stránce eshop.jsp pojat spíše jako testovací, nikoli jako reálná stránka v Internetovém obchodu se shrnutím nákupu.

3.2.7 Reference

- http://en.wikipedia.org/wiki/Java_(software_platform)
- http://java.sun.com/javase/technologies/security/
- http://www.bouncycastle.org/

3.3 Vzorová implementace ASP.NET

Vzorová implementace v ASP.NET je umístěna na instalačním CD v adresáři „implementace\shop_net“. Pro práci s digitálním podpisem využívá namespace System.Security.Cryptography.Pkcs, který je dostupný v .Net Framework verze 2.0 (viz Reference [4]).

Nejdůležitější třídy v projektu jsou následující:

- **PGCrypter** – třída pro šifrování a dešifrování hesla certifikátu
- **PGSigner** – třída pro generování (parametr SIGN odesílané platby na platební bránu) a ověření (parametr BANK_SIGN ve výsledku platby z platební brány) PKCS#7 zprávy

Pro detailnější popis si prosím prostudujte komentáře ve zdrojovém kódu.

3.3.1 Minimální verze prostředí, potřebná pro běh

- **ASP.NET** 2.0

3.3.2 Popis implementace

Popis souborů vzorové implementace v adresáři „shop_net“

soubor	popis
Default.aspx	výchozí stránka, slouží pouze pro přesměrování na shop.aspx
results.aspx	testovací stránka pro ověření odpovědi z platební brány
shop.aspx	testovací stránka pro odeslání požadavku na platební bránu
Web.Config	konfigurační soubor, neměl by být dostupný z webu
results.aspx.cs	definice třídy pro stránku results.aspx, neměl by být dostupný z webu
shop.aspx.cs	definice třídy pro stránku shop.aspx, neměl by být dostupný z webu
App_Code\PGCrypter.cs	definice třídy PGCrypter, neměl by být dostupný z webu
App_Code\PGSigner.cs	definice třídy PGSigner, neměl by být dostupný z webu

3.3.3 Konfigurace implementace

Implementace se konfiguruje souborem Web.Config.

Příklad konfigurace v souboru “Web.Config”:

```
<configuration>
  <appSettings>
    <add key="signerShopCertPath" value="App_Data\cert\PG_KLIENT_1.p12"/>
    <add key="signerShopCertPasswordPath" value="App_Data\pass\pass.txt"/>
    <add key="signerGateCertPath" value="App_Data\cert\kb_platebni_brana.cer"/>
    <add key="caStoreDir" value="App_Data\ca"/>
    <add key="caStoreFiles" value="dcs_ca_kb.cer;root_ca_kb.cer"/>
    <add key="saveDataFilePath" value="App_Data\tmp\data.xml"/>
  </appSettings>
  …
</configuration>
```

Popis konfiguračních parametrů ve Web.Config:

Parametr	Význam
signerShopCertPath	Relativní cesta k souboru s X.509 certifikátem (včetně privátního klíče) Vašeho eshopu na Web serveru. Web server musí mít do adresáře právo ke čtení. Soubor by neměli být dostupný z webu. Soubor musí být ve formátu PKCS#12, jinak jej tato vzorová implementace nenačte.

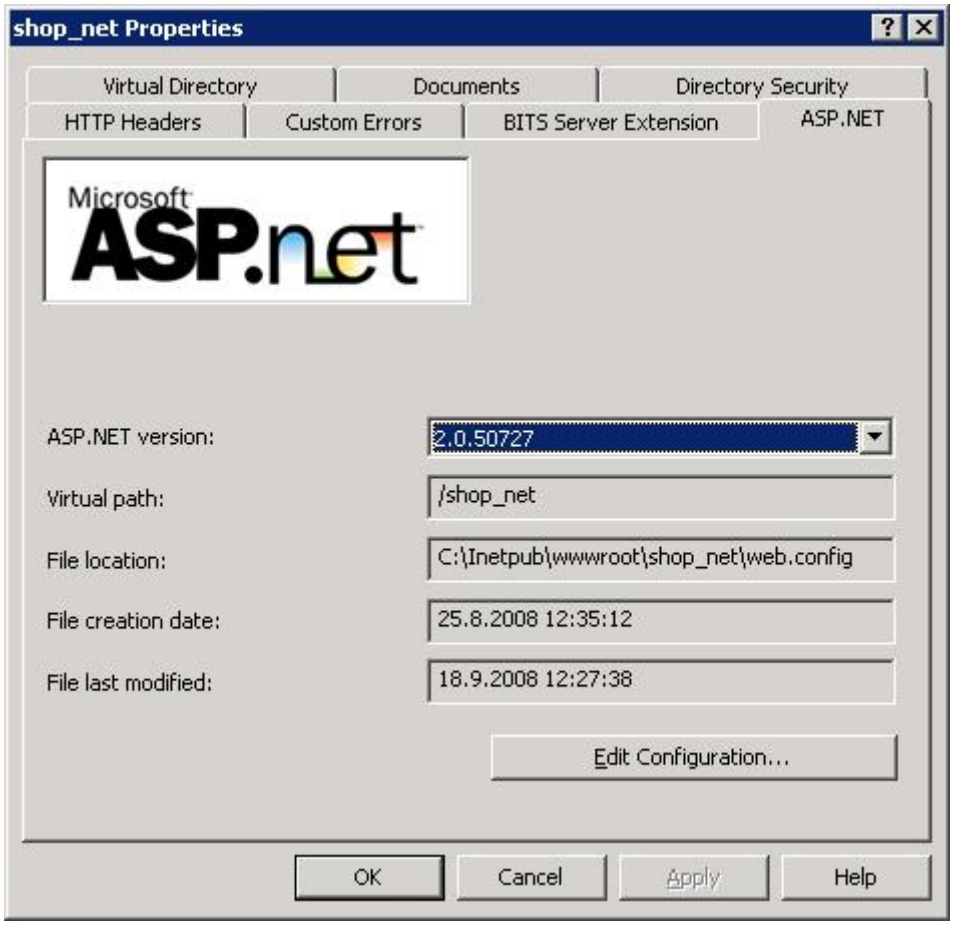
	Pozor na bezpečnost uložení certifikátu. **Zvolíte-li jako zde souborový systém, je potřeba nastavit k souboru oprávnění tak, aby běžný uživatel neměl k souboru přístup, aby se soubor neocitl ve struktuře, stáhnutelné přes WWW, a aby proces webového serveru měl přístup pouze ke čtení.** Máte-li možnost používat bezpečnější variantu vytváření a ověřování elektronického podpisu, například pomocí hardware security modulu (HSM), doporučujeme ji použít.
signerShopCertPasswordPath	Cesta k souboru, který obsahuje zašifrované heslo k PKCS#12 souboru s certifikátem, který je dán v parametru signerShopCertPath. Soubor by neměli být dostupný z webu. Zašifrované heslo (ve formátu base64) získáme zavoláním metody PGCrypter.EncryptPassword("heslo k certifikatu"), výsledek uložíme do souboru s cestou danou parametrem signerShopCertPasswordPath. **Pozor, nedoporučuje se ukládat hesla do souboru v čitelné, nebo pro třetí osobu jinak dešifrovatelné podobě**. Je zde bezpečnostní riziko zneužití privátního klíče Vašeho certifikátu nepovolanou osobou (hackerem), což by Vám mohlo způsobit škody. Použijte proto pokud možno co nejbezpečnější možný mechanizmus pro vytváření a ověřování elektronického podpisu, který máte k dispozici. **Zvolíte-li jako zde souborový systém, je potřeba nastavit k souboru oprávnění tak, aby běžný uživatel neměl k souboru přístup, aby se soubor neocitl ve struktuře, stáhnutelné přes WWW, a aby proces aplikačního (případně webového) serveru měl přístup pouze ke čtení.**
signerGateCertPath	Relativní cesta k souboru s důvěryhodným X.509 certifikátem KB platební brány na Web serveru. Web server musí mít do adresáře právo ke čtení. Soubor by neměli být dostupný z webu. Soubor musí být ve formátu CER nebo DER, jinak jej tato vzorová implementace nenačte. Přípona souboru často bývá „.cer“. **Pozor** na bezpečnost uložení certifikátu. Případný útočník (hacker) se může pokusit na souborový systém podstrčit soubor s jiným certifikátem. **Zvolíte-li jako zde souborový systém, je potřeba nastavit k souboru oprávnění tak, aby běžný uživatel neměl k souboru přístup, aby se soubor neocitl ve struktuře, stáhnutelné přes WWW, a aby proces aplikačního (případně webového) serveru měl přístup pouze ke čtení.**
caStoreDir	Relativní cesta k adresáři se soubory pro ověřeni platnosti certifikátu. Adresář by neměl být dostupný z webu.
caStoreFiles	seznam důvěryhodných X.509 certifikátů v adresáři uvedeného v parametru caStoreDir, krete jsou v ověřovací cestě certifikátu uvedeného v parametru signerCateCertPath. Soubory by neměli být dostupné z webu. Soubor musí být ve formátu CER nebo DER, jinak jej tato vzorová implementace nenačte. Přípona souboru často bývá „.cer“. **Pozor** na bezpečnost uložení certifikátů. Případný útočník (hacker) se může pokusit na souborový systém podstrčit soubor s jiným certifikátem. **Zvolíte-li jako zde souborový systém, je potřeba nastavit k souboru oprávnění tak, aby běžný uživatel neměl k souboru přístup, aby se soubor neocitl ve struktuře, stáhnutelné přes WWW, a aby proces aplikačního (případně webového) serveru měl přístup pouze ke čtení.**
saveDataFilePath	Relativní cesta k souboru pro uložení dat testovacího formuláře. Web server musí mít do adresáře právo k zápisu. Soubor by neměli být dostupný z webu.

3.3.4 Instalace eshopu na web server

Instalace vzorového eshopu v ASP.NET je popisována na Windows 2003 Server R2 SP2 s nainstalovaným IIS 6 a ASP.NET 2.0 (viz Reference [2]).

Nakopírujeme adresář „shop_net“ na web server např. do „C:\Inetpub\wwwroot\“. V IIS Manageru vytvoříme aplikaci „shop_net“ založenou na tomto adresáři (viz Reference [1]). Ve vlastnostech aplikace nastavíme na záložce ASP.NET verzi na 2.0 nebo vyšší (viz Obrázek 1). Parametry konfigurace lze měnit buď editací souboru web.config (viz Reference [3]) v adresáři aplikace nebo ve vlastnostech aplikace na záložce ASP.NET tlačítko „Edit Configuration“ (viz Obrázek 1).

Ověříme zda jsou všechny konfigurační parametry správně zadány a zda jsou nastavena příslušná uživatelská práva jak je popsáno u jednotlivých parametrů (viz výše). Vzorová implementace by měla být dostupná na url http://webserver/shop_net/.



Obrázek 1. Vlastnosti aplikace shop_net v IIS Manageru

3.3.5 Poznámky

Pozor, vzorová implementace je pojata spíše jako testovací, nikoli jako reálné stránky v Internetovém obchodu se shrnutím nákupu a tlačítkem odesílajícím platbu. Při návratu z platební brány do Internetového obchodu, není ověřována session ani rovnost odesílaných parametrů **SHOPID, ORDERID, ACCOUNT, BANK, DESC, VS, KS, SS, AMOUNT** a **BACKURL** s nazpět přijatými. Toto by měl reálný Internetový obchod implementovat.

3.3.6 Reference

[1] Creating Applications in IIS 6.0:
http://www.microsoft.com/technet/prodtechnol/WindowsServer2003/Library/IIS/1d1c9a73-b4c5-4cfb-ad69-b77fa2e17e19.mspx

[2] Troubleshooting an ASP.NET Installation (IIS 6.0): http://www.microsoft.com/technet/prodtechnol/WindowsServer2003/Library/IIS/7ecaa5f3-5499-4887-8c9d-00aba71125df.mspx
[3] ASP.NET Configuration Files Format (IIS 6.0): http://www.microsoft.com/technet/prodtechnol/WindowsServer2003/Library/IIS/8aaab093-772f-4db8-80b9-4fcae5ebbb75.mspx
[4] .Net Framework Class Library - System.Security.Cryptography.Pkcs Namespace: http://msdn.microsoft.com/en-us/library/system.security.cryptography.pkcs(VS.80).aspx

3.4 Vzorová implementace PHP

Vzorová implementace v PHP je umístěna na instalačním CD v adresáři „implementace\shop_php“. Pro práci s digitálním podpisem využívá extenzi PHP OpenSSL (viz Reference [6]) a pro šifrování extenzi PHP Mcrypt (viz Reference [9]).

Nejdůležitější třídy v projektu jsou následující:

- **PGCrypter** – třída pro šifrování a dešifrování hesla certifikátu
- **PGSigner** – třída pro generování (parametr SIGN odesílané platby na platební bránu) a ověření (parametr BANK_SIGN ve výsledku platby z platební brány) PKCS#7 zprávy
- **Trace** - pomocná třída pro trasování kódu, tato třída není vyžadována
- **ShopConfig** - třída pro získání konfigurace z config.ini

Pro detailnější popis si prosím prostudujte komentáře ve zdrojovém kódu.

3.4.1 Minimální verze prostředí, potřebné pro běh

- **PHP** 5.1
- **OpenSSL** 0.9.6
- **Mcrypt** 2.5.6

3.4.2 Popis implementace

Popis souborů vzorové implementace v adresáři „shop_php“

soubor	popis
config.inc	definice třídy ShopConfig, neměl by být dostupný z webu
results.php.inc	definice třídy ResultsPage pro stránku results.php, neměl by být dostupný z webu
shop.php.inc	definice třídy ShopPage pro stránku shop.php, neměl by být dostupný z webu
config.ini	konfigurační soubor, neměl by být dostupný z webu
index.php	výchozí stránka, slouží pouze pro přesměrování na shop.php
results.php	testovací stránka pro ověření odpovědi z platební brány
shop.php	testovací stránka pro odeslání požadavku na platební bránu
app_code\pgcrypter.inc	definice třídy PGCrypter, neměl by být dostupný z webu
app_code\pgsigner.inc	definice třídy PGSigner, neměl by být dostupný z webu
app_code\trace.inc	definice třídy Trace, neměl by být dostupný z webu

3.4.3 Konfigurace implementace

Implementace se konfiguruje souborem config.ini.

Příklad konfigurace v config.ini:

```
signerShopCertPath="app_data\cert\PG_klient_1.pem"
signerShopCertPasswordPath="app_data\pass\pass.txt"
signerGateCertPath="app_data\cert\kb_platebni_brana.cer"
caStoreDir="app_data\ca"
```

```
caStoreFile="dcs_ca_kb.cer;root_ca_kb.cer"
saveDataFilePath="app_data\tmp\data.ini"
tempPath="app_data\tmp"
```

Popis konfiguračních parametrů v config.ini:

Parametr	Význam
signerShopCertPath	Relativní cesta k souboru s X.509 certifikátem (včetně privátního klíče) Vašeho eshopu na Web serveru. Web server musí mít do adresáře právo ke čtení. Soubor by neměli být dostupný z webu. Soubor musí být ve formátu PEM, jinak jej tato vzorová implementace nenačte. **Pozor** na bezpečnost uložení certifikátu. **Zvolíte-li jako zde souborový systém, je potřeba nastavit k souboru oprávnění tak, aby běžný uživatel neměl k souboru přístup, aby se soubor neocitl ve struktuře, stáhnutelné přes WWW, a aby proces aplikačního (případně webového) serveru měl přístup pouze ke čtení.** Máte-li možnost používat bezpečnější variantu vytváření a ověřování elektronického podpisu, například pomocí hardware security modulu (HSM), doporučujeme ji použít.
signerShopCertPasswordPath	Cesta k souboru, který obsahuje zašifrované heslo k soukromému klíčí v PEM souboru s certifikátem, který je dán v parametru signerShopCertPath. Soubor by neměli být dostupný z webu. Zašifrované heslo (ve formátu base64) získáme zavoláním metody PGCrypter::EncryptPassword("heslo k certifikatu"), výsledek uložíme do souboru s cestou danou parametrem signerShopCertPasswordPath. **Pozor, nedoporučuje se ukládat hesla do souboru v čitelné, nebo pro třetí osobu jinak dešifrovatelné podobě**. Je zde bezpečnostní riziko zneužití privátního klíče Vašeho certifikátu nepovolanou osobou (hackerem), což by Vám mohlo způsobit škody. Použijte proto pokud možno co nejbezpečnější možný mechanizmus pro vytváření a ověřování elektronického podpisu, který máte k dispozici. **Zvolíte-li jako zde souborový systém, je potřeba nastavit k souboru oprávnění tak, aby běžný uživatel neměl k souboru přístup, aby se soubor neocitl ve struktuře, stáhnutelné přes WWW, a aby proces aplikačního (případně webového) serveru měl přístup pouze ke čtení.**
signerGateCertPath	Relativní cesta k souboru s důvěryhodným X.509 certifikátem KB platební brány na Web serveru. Web server musí mít do adresáře právo ke čtení. Soubor by neměli být dostupný z webu. Soubor musí být ve formátu CER, jinak jej tato vzorová implementace nenačte. **Pozor** na bezpečnost uložení certifikátu. Případný útočník (hacker) se může pokusit na souborový systém podstrčit soubor s jiným certifikátem. **Zvolíte-li jako zde souborový systém, je potřeba nastavit k souboru oprávnění tak, aby běžný uživatel neměl k souboru přístup, aby se soubor neocitl ve struktuře, stáhnutelné přes WWW, a aby proces aplikačního (případně webového) serveru měl přístup pouze ke čtení.**
caStoreDir	Relativní cesta k adresáři se soubory pro ověřeni platnosti certifikátu. Adresář by neměl být dostupný z webu.
caStoreFiles	seznam důvěryhodných X.509 certifikátů v adresáři uvedeného v parametru caStoreDir, krete jsou v ověřovací cestě certifikátu uvedeného v parametru signerCateCertPath. Soubory by neměli být dostupné z webu. Soubor musí být ve formátu CER, jinak jej tato vzorová implementace

	nenačte. **Pozor** na bezpečnost uložení certifikátů. Případný útočník (hacker) se může pokusit na souborový systém podstrčit soubor s jiným certifikátem. **Zvolíte-li jako zde souborový systém, je potřeba nastavit k souboru oprávnění tak, aby běžný uživatel neměl k souboru přístup, aby se soubor neocitl ve struktuře, stáhnutelné přes WWW, a aby proces aplikačního (případně webového) serveru měl přístup pouze ke čtení.**
saveDataFilePath	Relativní cesta k souboru pro uložení dat testovacího formuláře. Web server musí mít do adresáře právo k zápisu. Soubor by neměli být dostupný z webu.
tempPath	Relativní cesta k pracovnímu adresáři. Web server musí mít do adresáře právo k zápisu. Adresář by neměl být dostupný z webu.

3.4.4 Instalace eshopu na web server

Instalace vzorového eshopu v PHP je popisována na Windows 2003 Server R2 SP2 s nainstalovaným IIS 6 a PHP (viz Reference [10]) a instalované rozšíření OpenSSL v PHP (viz Reference [11]).

Nakopírujeme adresář „shop_php“ na web server např. do „C:\Inetpub\wwwroot\“. V IIS Manageru vytvoříme aplikaci „shop_php“ založenou na tomto adresáři (viz Reference [1]).

Pomoci aplikace OpenSSL převedeme dodaný certifikát ve formátu PKCS#12 do formátu PEM (viz Reference [7]). Příklad příkazu OpenSSL pro převod certifikátu do formátu PEM:

```
openssl.exe pkcs12 -in PG_klient1.p12 -clcerts -out PG_klient1.pem
```

Budete dotázáni na heslo k certifikátu *PG_klient1.p12* a následně k zadání a potvrzení ověřovací fráze k soukromému klíči v souboru *PG_klient1.pem*. Zadanou ověřovací frázi zadáme do konfiguračního parametru signerShopCertPassword.

Ověříme zda jsou všechny konfigurační parametry správně zadány a zda jsou nastavena příslušná uživatelská práva jak je popsáno u jednotlivých parametrů (viz výše). Vzorová implementace by měla být dostupná na url http://webserver/shop_php/.

3.4.5 Poznámky

Pozor, vzorová implementace je pojata spíše jako testovací, nikoli jako reálné stránky v Internetovém obchodu se shrnutím nákupu a tlačítkem odesílajícím platbu. Při návratu z platební brány do Internetového obchodu, není ověřována session ani rovnost odesílaných parametrů **SHOPID, ORDERID, ACCOUNT, BANK, DESC, VS, KS, SS, AMOUNT** a **BACKURL** s nazpět přijatými. Toto by měl reálný Internetový obchod implementovat.

3.4.6 Reference

[5] OpenSSL: http://www.openssl.org/
[6] OpenSSL on PHP: http://www.php.net/manual/en/book.openssl.php
[7] OpenSSL PKCS12: http://www.openssl.org/docs/apps/pkcs12.html
[8] Mcrypt: http://mcrypt.sourceforge.net/
[9] Mcrypt on PHP: http://www.php.net/manual/en/book.mcrypt.php
[10] PHP Installation on Windows systems: http://cz.php.net/install.windows
[11] PHP Installation of extensions on Windows: http://cz.php.net/manual/en/install.windows.extensions.php