GP webpay - Správa soukromého klíče a podepisování požadavků

Verze: 1.3

Global Payments Europe, s.r.o.

Vytvořeno **19.2.2016**

Poslední změna **6.12.2019**

SERVICE. DRIVEN. COMMERCE

globalpaymentsinc.com

Autor dokumentu	GPE Application Development
Správce dokumentu	
Schválil	
Verze	1.3
Stupeň utajení	Důvěrné

Historie dokumentu:

Verze	Datum	Provedl	Komentář
1.0	19.2.2016	GPE Application Development	První verze dokumentu
1.1	29.3.2016	GPE Application Development	Drobné opravy
1.2	3.5.2019	GPE Application Development	Sloučení s dokumentem „GP_webpay_Podepisovani_pozadavku_v1.0_CZ.docx“
1.3	25.11.2019	GPE Application Development	Nová verze GP webpay Keystore Managera, aktualizace příkladů

Obsah

1. Právní doložka

Tento dokument včetně všech případných příloh a odkazů je určen výhradně pro potřeby poskytovatele služeb e-shopu (dále jen „Zákazník“).

Informace v tomto dokumentu obsažené (dále jen „Informace“) jsou předmětem duševního vlastnictví a ochrany autorských práv společnosti Global Payments Europe, s.r.o. (dále jen „GPE“) a mají povahu obchodního tajemství v souladu s ust. § 504 zák. č. 89/2012 Sb., Občanský zákoník. Zákazník si je vědom právních povinností ve vztahu k nakládání s Informacemi.

Informace nebo kterákoliv její část nesmí být bez předchozího výslovného písemného souhlasu GPE poskytnuty nebo jakýmkoliv způsobem zpřístupněny třetí straně. Informace nesmí být zároveň využity Zákazníkem pro jiné účely, než pro účely ke kterému slouží. Pro vyloučení všech pochybností nesmí být Informace nebo kterákoliv část bez předchozího výslovného písemného souhlasu GPE poskytnuty nebo jakýmkoliv způsobem zpřístupněny ani společnostem poskytujícím služby zpracování plateb v prostředí internetu.

GPE si v rozsahu dovoleném platným právem, vyhrazuje veškerá práva k této dokumentaci a k Informacím v ní obsažených. Jakékoliv rozmnožování, použití, vystavení či jiné zveřejnění nebo šíření Informací nebo její části metodami známými i dosud neobjevenými je bez předchozího písemného souhlasu společnosti GPE přísně zakázáno. GPE není jakkoliv odpovědná za jakékoliv chyby nebo opomenutí v Informacích. GPE si vyhrazuje právo, a to i bez uvedení důvodu, jakoukoliv Informaci změnit nebo zrušit.

2. Úvod

Dokument popisuje princip zabezpečení zakládání plateb v prostředí platební brány GP webpay a autorizace následných operací s platbami.

2.1 Obecný princip zabezpečení GP webpay

Systém GP webpay pro své zabezpečení používá tzv. PKI (Public Key Infrastructure) model. Tento model využívá asymetrickou kryptografii, při které se používají dva rozdílné klíče.

1. Soukromý klíč – tato část je tajná a vlastní ji pouze oprávněná osoba
2. Veřejný klíč – veřejná část, kterou lze volně distribuovat jakýmkoli (i nezabezpečeným) kanálem – e-mail, veřejné úložiště klíčů …

Hlavní vlastností soukromého klíče je to, že žádné dva klíče na světě se neshodují – tj. každý klíč je originál.

2.1.1 Získání soukromého klíče

- Veřejná certifikační autorita – obecně přijímaná důvěryhodná komerční instituce zajišťující správu klíčů (vydávání, zneplatnění, obnovování …). Její veřejný klíč bývá umístěn přímo ve webových prohlížečích, popř. v různých run-timech (běhová prostředí pro ostatní software – např. Java, .NET…). Klíče vydané takovouto institucí jsou obecně přijímány jako důvěryhodné a používají se pro komunikaci s bankami a veřejnými institucemi – např. Thawte (https://www.thawte.com/), První certifikační autorita a.s. (http://www.ica.cz/).
- Různá obecná řešení – soukromé klíče nejsou všeobecně akceptovány, ale jsou postaveny na důvěře mezi klientem a konkrétním poskytovatelem klíče – např. Komerční banka má svoji certifikační autoritu a poskytuje klíče svým klientům pro komunikaci s internetovým bankovnictvím.
- GP webpay umožňuje svým klientů získání soukromého klíče prostřednictvím webového portálu. Tento klíč je možné použít pouze v prostředí GP webpay.

2.1.2 Účely využití PKI

- autentizace přístupu (ověření totožnosti uživatele)
- prověřování integrity zpráv (zpráva nebyla žádným způsobem změněna)
- nepopiratelnost – využití elektronického podpisu
- privátnost – šifrování zpráv, symetrické a asymetrické šifry

GP webpay využívá z těchto účelů pouze dva – ověření integrity a nepopiratelnost.

2.2 Využití PKI v GP webpay

2.2.1 Způsoby použití

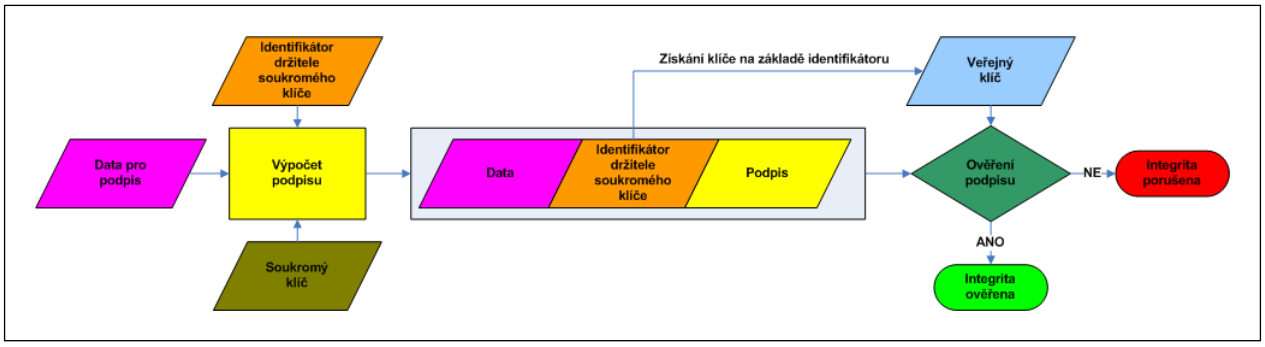
Soukromý klíč se používá pro výpočet podpisu veškerých zpráv umožňujících manipulaci s platbami. Ověřený podpis zaručuje integritu přenesených dat a správnou identitu (nepopiratelnost identity) zasilatele zprávy – neexistuje možnost vytvoření podpisu pomocí veřejné části klíče.

Typy zpráv:

- Zakládání nových plateb prostřednictvím standardního rozhraní HTTP
- Správa plateb v Portálu – stržení/vrácení peněžních prostředků držitele platební karty
- Správa plateb prostřednictvím služeb web-services – používá se při přímém propojení systému GP webpay s platebním systémem obchodníka

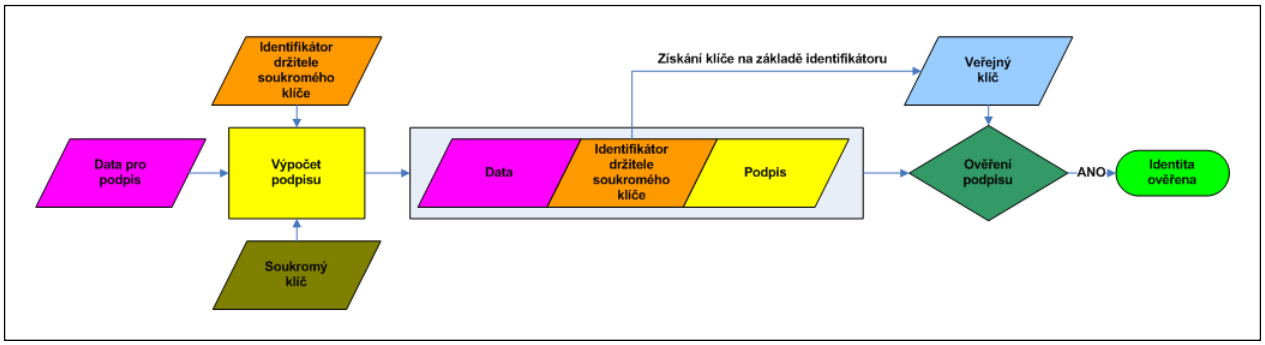
2.2.2 Ověření integrity zprávy

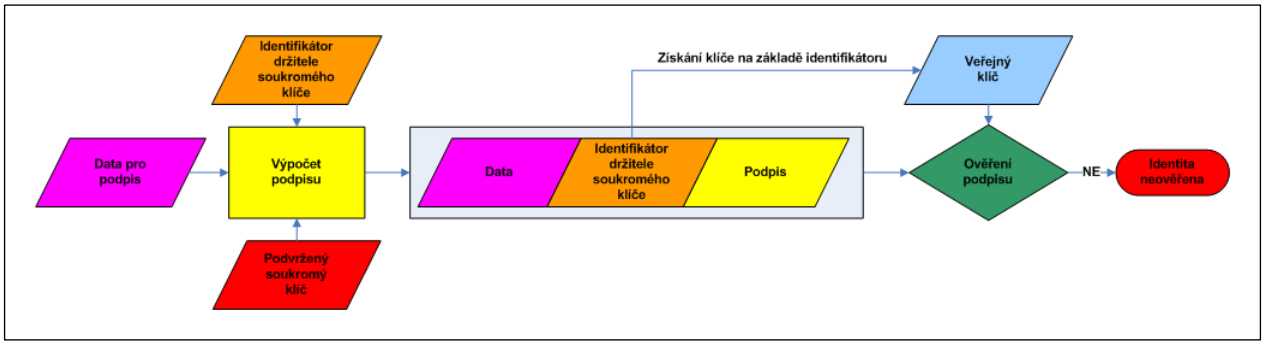
Každá zpráva modifikující data (ať jde o zakládání nebo operace s platbami) obsahuje kromě vlastních dat i pole pro podpis. Podpis vznikne na straně vlastníka soukromého klíče na základě vstupních dat a soukromého klíče s využitím obecného algoritmu pro výpočet podpisu. Po odeslání dat na server tento provede ověření obdobným způsobem, ale s využitím odpovídajícího veřejného klíče (veřejný klíč je vyhledán na základě identifikátoru původce zprávy zaslaného ve zprávě). Pokud se výpočet liší, tak došlo během přenosu dat k jejich narušení.



2.2.3 Ověření identity zasilatele zprávy

Součástí přenášených dat je také identifikátor původce zprávy. Na základě tohoto identifikátoru je vybrán odpovídající veřejný klíč na serveru. Pokud bylo možné podpis ověřit a za předpokladu, že neexistují dva shodné soukromé klíče, lze konstatovat, že daná data opravdu zaslal držitel soukromého klíče.


Identifikátor držitele soukromého klíče
Data pro podpis
Výpočet podpisu
Soukromý klíč
Získání klíče na základě identifikátoru
Veřejný klíč
Data
Identifikátor držitele soukromého klíče
Podpis
Ověření podpisu
ANO
Identita ověřena


Identifikátor držitele soukromého klíče
Data pro podpis
Výpočet podpisu
Podvržený soukromý klíč
Získání klíče na základě identifikátoru
Veřejný klíč
Data
Identifikátor držitele soukromého klíče
Podpis
Ověření podpisu
NE
Identita neověřena

3. Soukromý klíč a jeho správa

3.1 Soukromý klíč obecně

Soukromý klíč je základem bezpečnosti systému GP webpay. Tento klíč je ve výhradním vlastnictví držitele klíče a je nutné maximálně dodržovat bezpečnostní požadavky na jeho utajení:

- Uchovávat jej na bezpečném místě
- Vždy jej mít zabezpečen heslem
- Pokud dojde k jeho vyzrazení, je nutné získat klíč nový a o kompromitaci informovat všechny subjekty využívající ověřování identity pomocí jeho veřejné části

Soukromý klíč je uložen v datovém souboru. Tento soubor nazýváme úložiště, popř. keystore. Uložiště může obsahovat více soukromých i veřejných klíčů. Aby bylo možné jednotlivé klíče v úložišti odlišit, jsou k nim přiřazeny názvy – tzv. aliasy. Úložiště bývá chráněno centrálním heslem a každý soukromý klíč ještě svým vlastním heslem.

Existuje několik formátů úložišť. Pro naše účely budou postačovat tyto (dále popsaná konverzní aplikace podporuje právě tyto formáty):

JCEKS – úložiště ve formátu podporované programovacím jazykem JAVA

PFX – úložiště ve formátu podporované společností Microsoft (PKCS12)

PEM – úložiště ve formátu podporované programovacím jazykem PHP

K těmto typům se ještě váží formáty pro distribuci veřejného klíče:

PEM – úložiště v textovém formátu

DER – úložiště v binárním formátu

3.2 Získání soukromého klíče

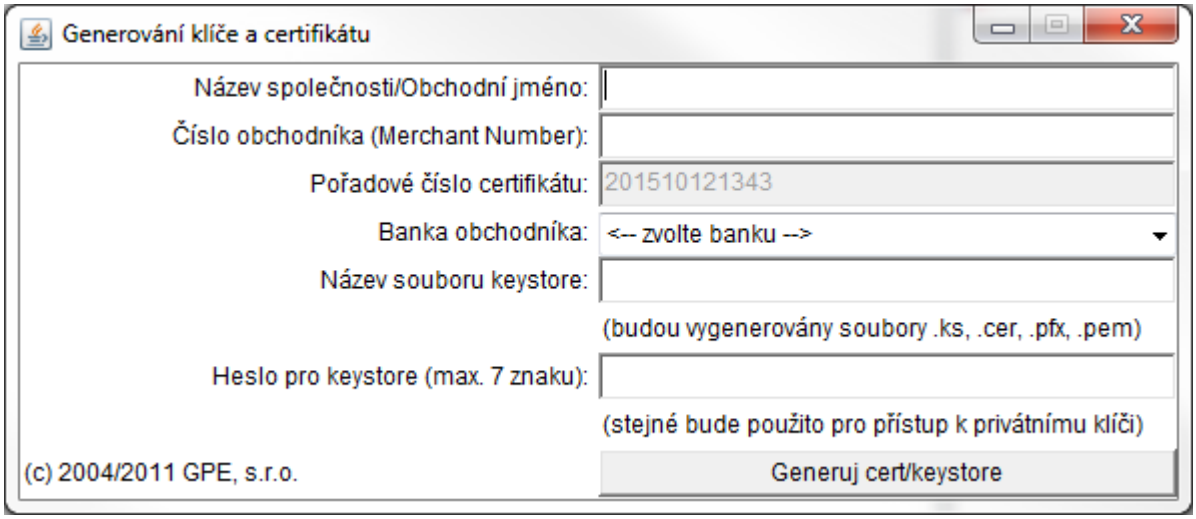
Jak již bylo zmíněno, lze soukromý klíč získat několika způsoby. Pro komerční využití, popř. pro komunikaci s veřejnou správou, je nutné klíč získat od uznávané certifikační autority.

Pro účely provozu systému GP webpay je dostačující jeho získání prostředky dostupnými v GP webpay.

Pokud již máte nějaký soukromý klíč zakoupen (existuje několik komerčních certifikačních autorit, které vydávají/prodávají soukromé klíče), je možné použít ten.

3.2.1 Historie

Od samého počátku fungování GP webpay byla možnost získání soukromého klíče pomocí samostatně dodávané aplikace „Generování klíče a certifikátu“. Tato aplikace byla dostupná ke stažení z uživatelského prostředí GP webpay GUI a také jako součást distribučního balíčku dokumentace.



Výsledkem generování jsou následující soubory:

`<jméno>.ks` – soubor keystore v Java formátu – obsahuje soukromý i veřejný klíč

`<jméno>.pfx` – soubor keystore ve formátu PKCS#12 – obsahuje soukromý i veřejný klíč

`<jméno>.pem` – soubor keystore ve formátu PEM – obsahuje soukromý i veřejný klíč – např. pro PHP aplikace

`<jméno>.cer` – soubor s veřejným klíčem

3.2.2 Současnost

Nové grafické rozhraní pro správu objednávek Portál GP webpay má v sobě zakomponovánu správu soukromých a veřejných klíčů jednotlivých e-shopů. Jejíž součástí je také možnost vygenerování soukromého klíče prostřednictvím webového prohlížeče.

Výsledkem generování je soubor „`gpwebpay-pvk.key`“ v textovém formátu PEM.

Tento klíč lze přímo vložit do používaného webového prohlížeče (prostřednictvím importu v Portálu).

Současně je nutné jej zachovat pro další použití – např. v jiném prohlížeči.

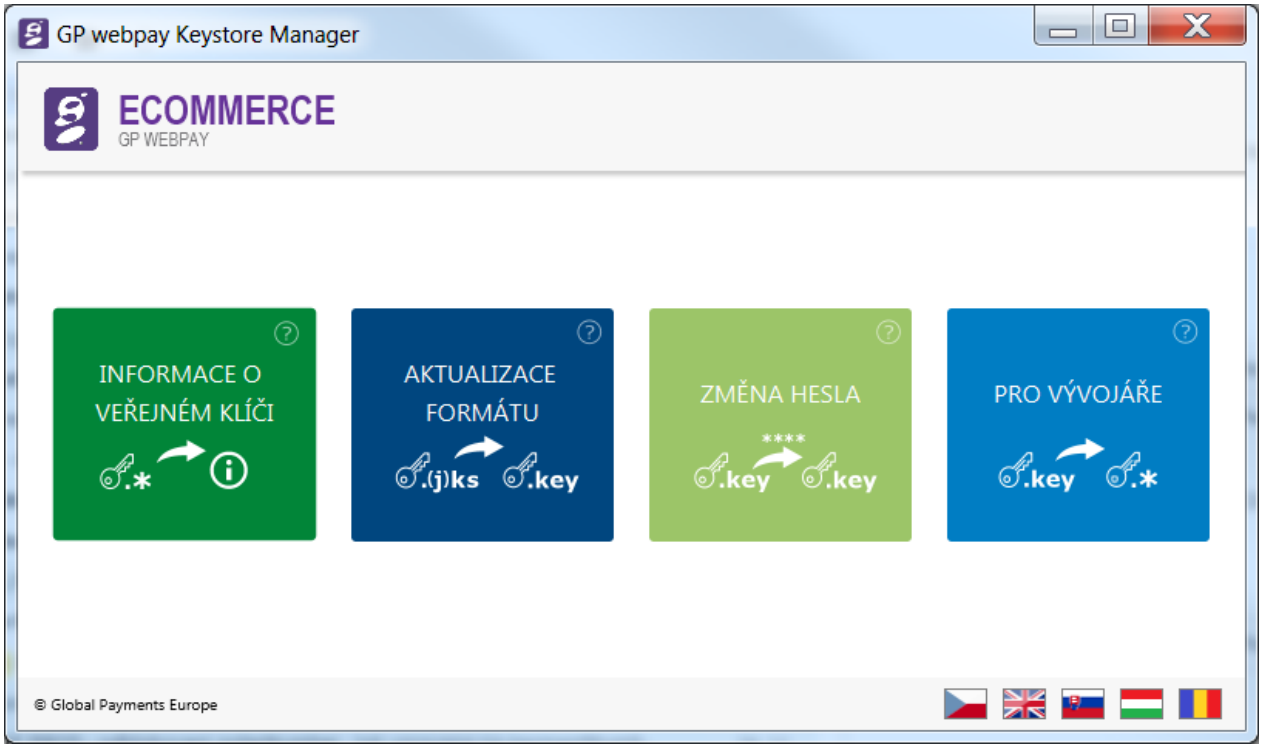
3.3 Správa soukromého klíče

Aby bylo možné pracovat s platbami ve webové aplikaci Portál GP webpay (dále pouze Portál), je nutné nahrát soukromý klíč do webového prohlížeče. Toto nahrání možné provést, po úspěšném přihlášení, přímo v prostředí Portálu. Soukromý klíč je nutné mít uložen v textovém formátu PEM, tento formát také vzniká při generování klíče v Portálu (soubor „`gpwebpay-pvk.key`“).

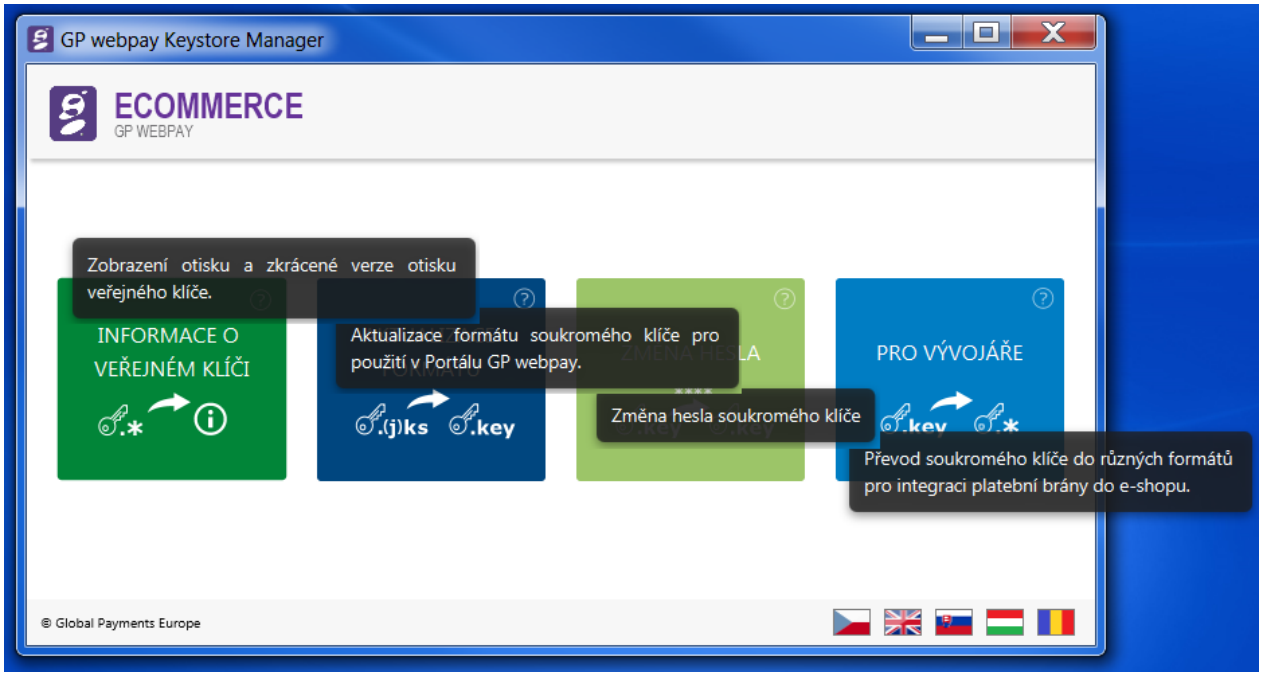
Pokud ovšem již máte soukromý klíč z dřívějška, je nutné původní formát aktualizovat do formátu nového. K této aktualizaci formátu slouží aplikace GP webpay Keystore Manager. Aplikace je dostupná v sekci „Ke stažení“ v Portálu a pro svůj běh vyžaduje nainstalované běhové prostředí jazyku Java (ke stažení z Oracle webu http://www.java.com).

Aplikace GP webpay Keystore Manager obsahuje tyto funkčnosti:

- Informace o veřejném klíči – zobrazení „otisku“ klíče – lze porovnat s hodnotou v GP webpay Portálu
- Aktualizace formátu – konverze formátu původního souboru se soukromým klíčem
- Změna hesla – změna hesla soukromého klíče v novém formátu
- Pro vývojáře – automatická konverze soukromého klíče v novém formátu do formátů podporovaných různými vývojářskými nástroji



Po najetí myší na „dlaždici“ se zobrazí stručný popis funkcionality:

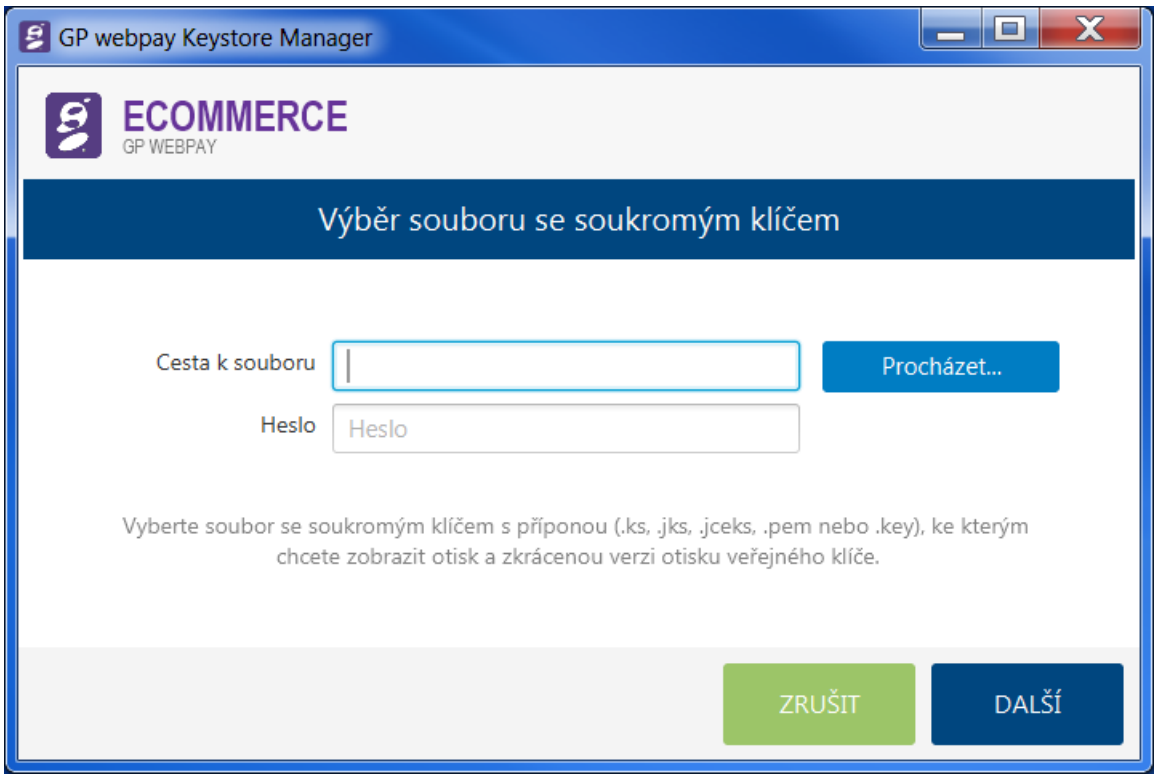


Aplikace podporuje několik jazykových variant. K jejich přepnutí slouží ikonky vlajek ve spodní části obrazovky.

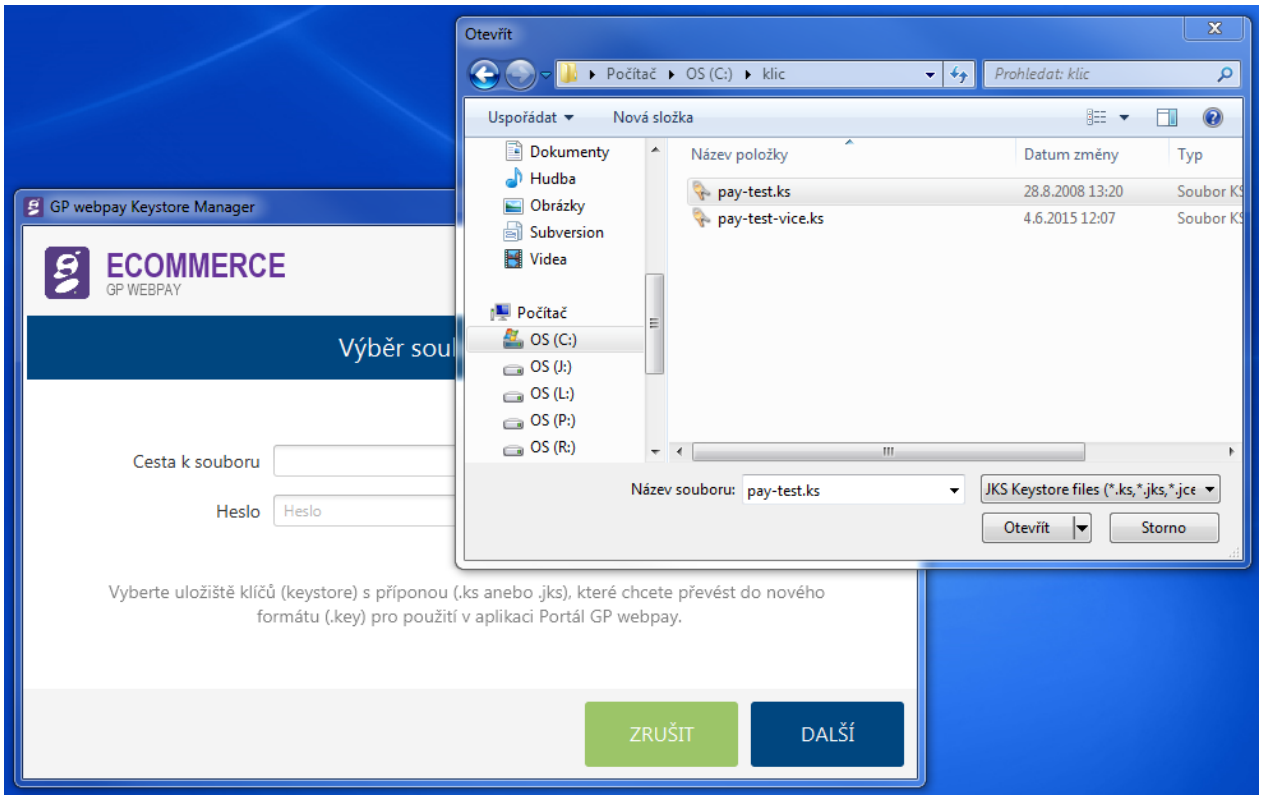
3.3.1 Informace o veřejném klíči

„Dlaždice“ slouží k zobrazení „otisku“ veřejného klíče. „Otisk“ lze následně porovnat s hodnotou v GP webpay portálu a ověřit shodnost verze klíče v keystore s verzí klíče uložené na serveru GP webpay.

Po kliknutí na dlaždici „INFORMACE O VEŘEJNÉM KLÍČI“ je zobrazeno okno pro výběr souboru se soukromým klíčem:



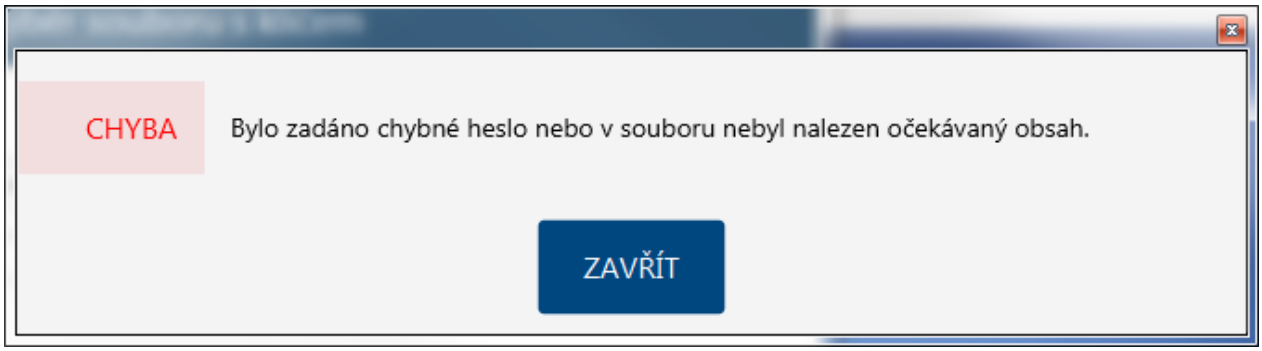
Ve vstupním poli „Cesta k souboru“ je potřeba, pomocí procházení adresářové struktury, najít soubor původního klíče.



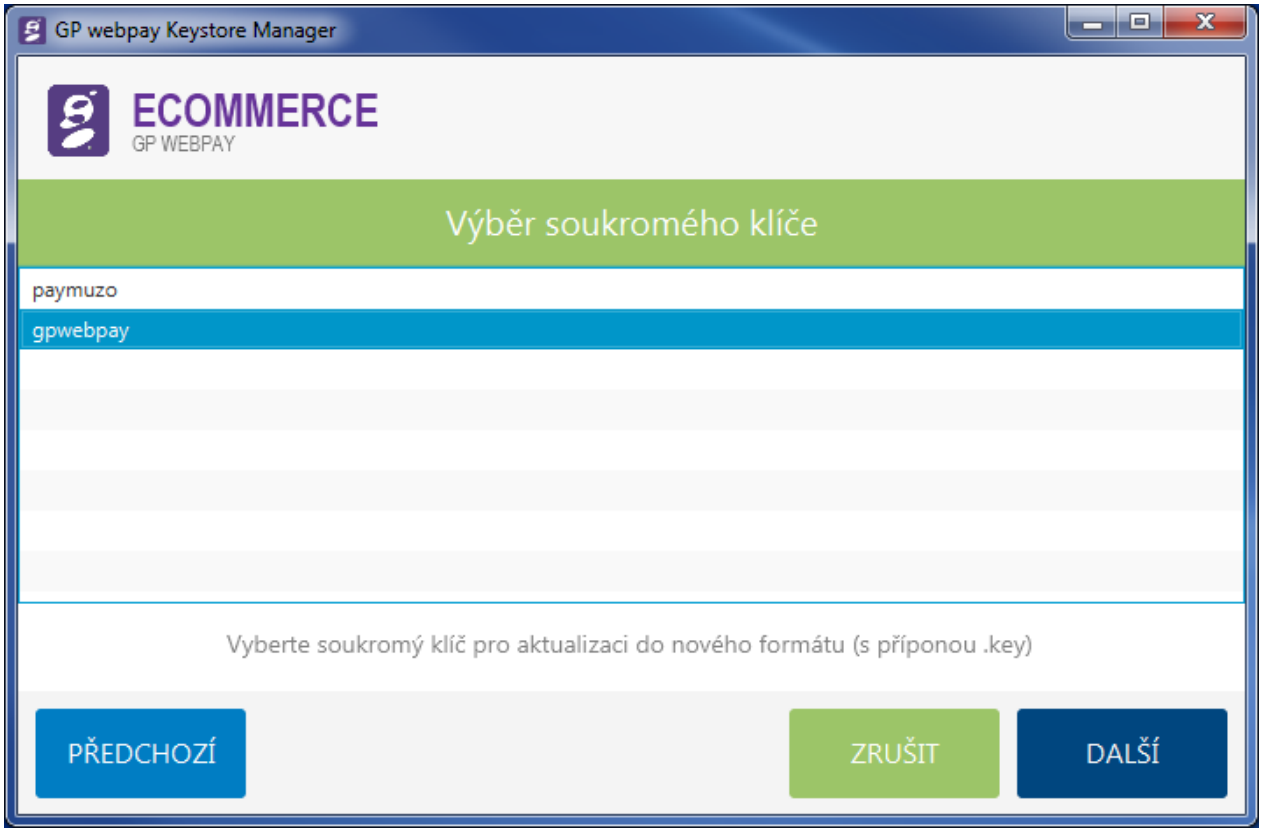
Potvrdit výběr souboru tlačítkem „Otevřít“.

Výběrové okno se uzavře a aplikace čeká na zadání hesla k úložišti soukromého klíče a stisk tlačítka „Další“. Následuje pokus o načtení obsahu souboru.

Pokud je chybně zadáno heslo, popř. soubor neobsahuje soukromý klíč, je zobrazeno hlášení:

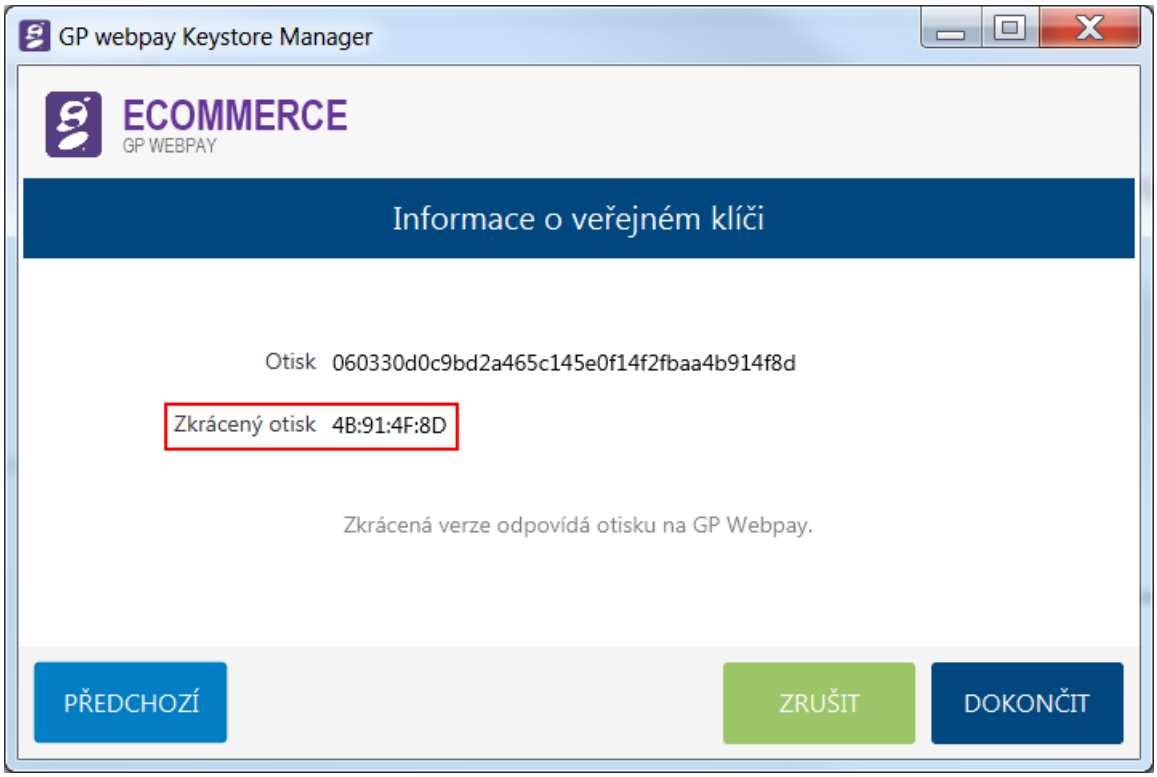


V případě, že soubor úložiště obsahuje více soukromých klíčů, dojde k zobrazení seznamu klíčů a je potřeba vybrat správný soukromý klíč:



a pokračovat tlačítkem „Další“. V případě existence pouze jednoho soukromého klíče je tato obrazovka přeskočena.

Následně je zobrazen „otisk“ vybraného klíče:



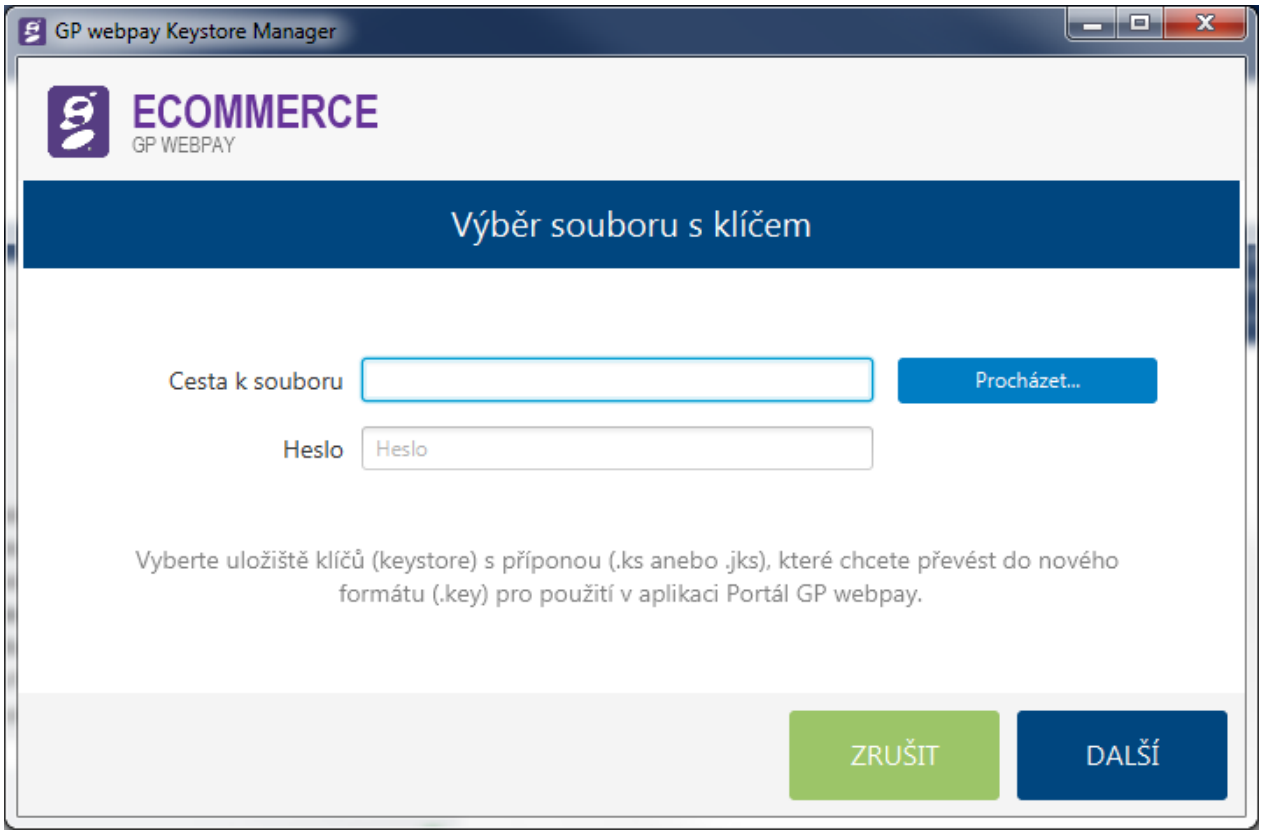
Zobrazená hodnota „Zkrácený otisk“ by měla odpovídat hodnotě v GP webpay Portálu:

A po stisku tlačítka „Dokončit“ se aplikace vrátí na úvodní obrazovku.

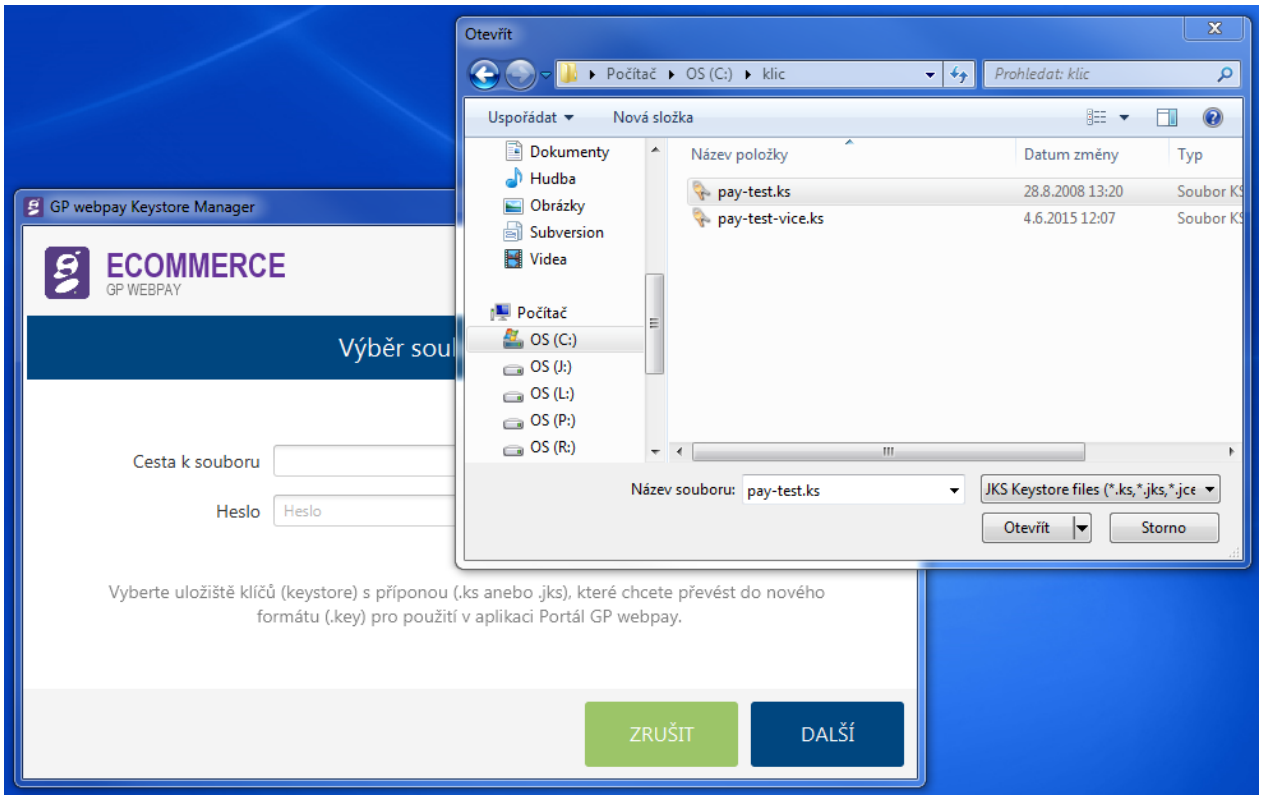
3.3.2 Aktualizace formátu

Tato „dlaždice“ slouží k aktualizaci formátu původního soukromého klíče, který se používal ve starém GUI pro obchodníky. Původní formát je v JAVA struktuře a má, většinou, příponu souboru „.ks“, „.jks“, „.jceks“. Nový formát je v PEM struktuře a soukromý klíč je uložen v souboru s názvem „gpwebpay-pvk.key“.

Po kliknutí na dlaždici „AKTUALIZACE FORMÁTU“ je zobrazeno okno pro výběr souboru se starým formátem soukromého klíče:



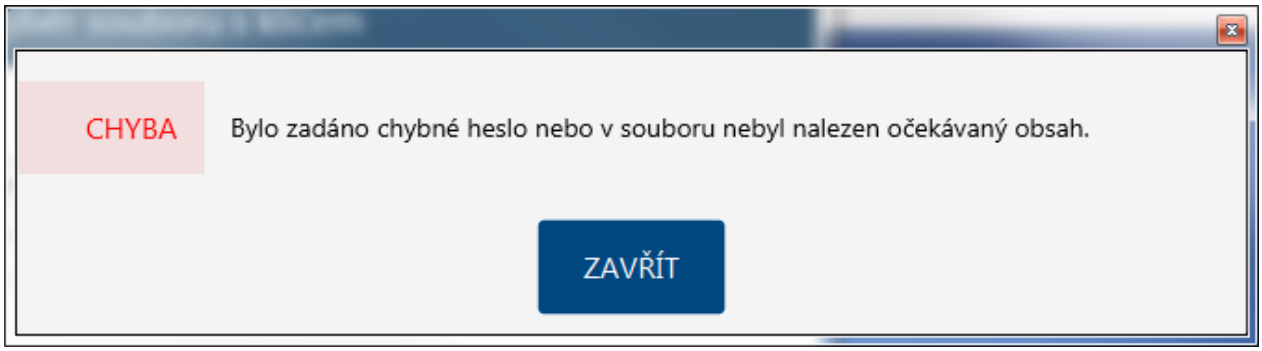
Ve vstupním poli „Cesta k souboru“ je potřeba, pomocí procházení adresářové struktury, najít soubor původního klíče.



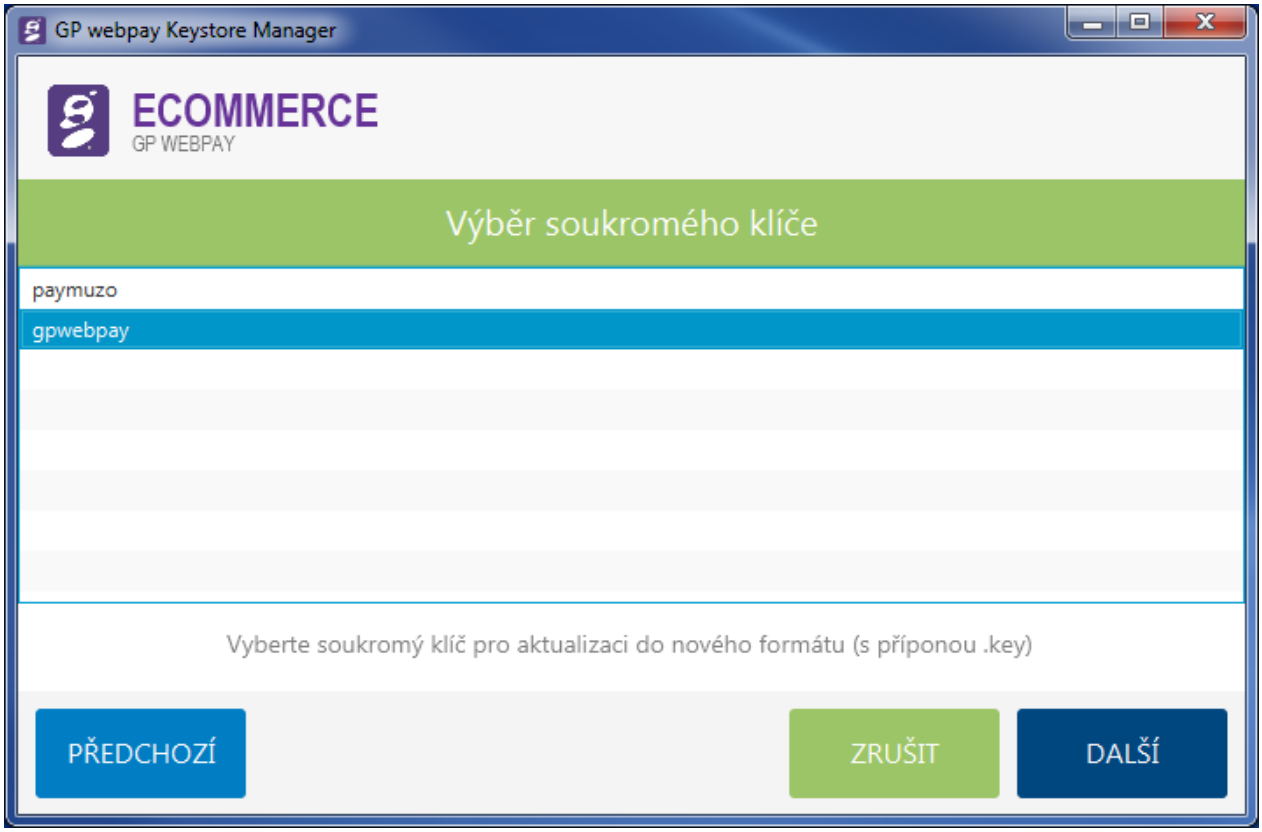
Potvrdit výběr souboru tlačítkem „Otevřít“.

Výběrové okno se uzavře a aplikace čeká na zadání hesla k původnímu úložišti soukromého klíče a stisk tlačítka „Další“. Následuje pokus o načtení obsahu souboru.

Pokud je chybně zadáno heslo, popř. soubor neobsahuje soukromý klíč, je zobrazeno hlášení:



V případě, že soubor úložiště obsahuje více soukromých klíčů, dojde k zobrazení seznamu klíčů a je potřeba vybrat správný soukromý klíč:

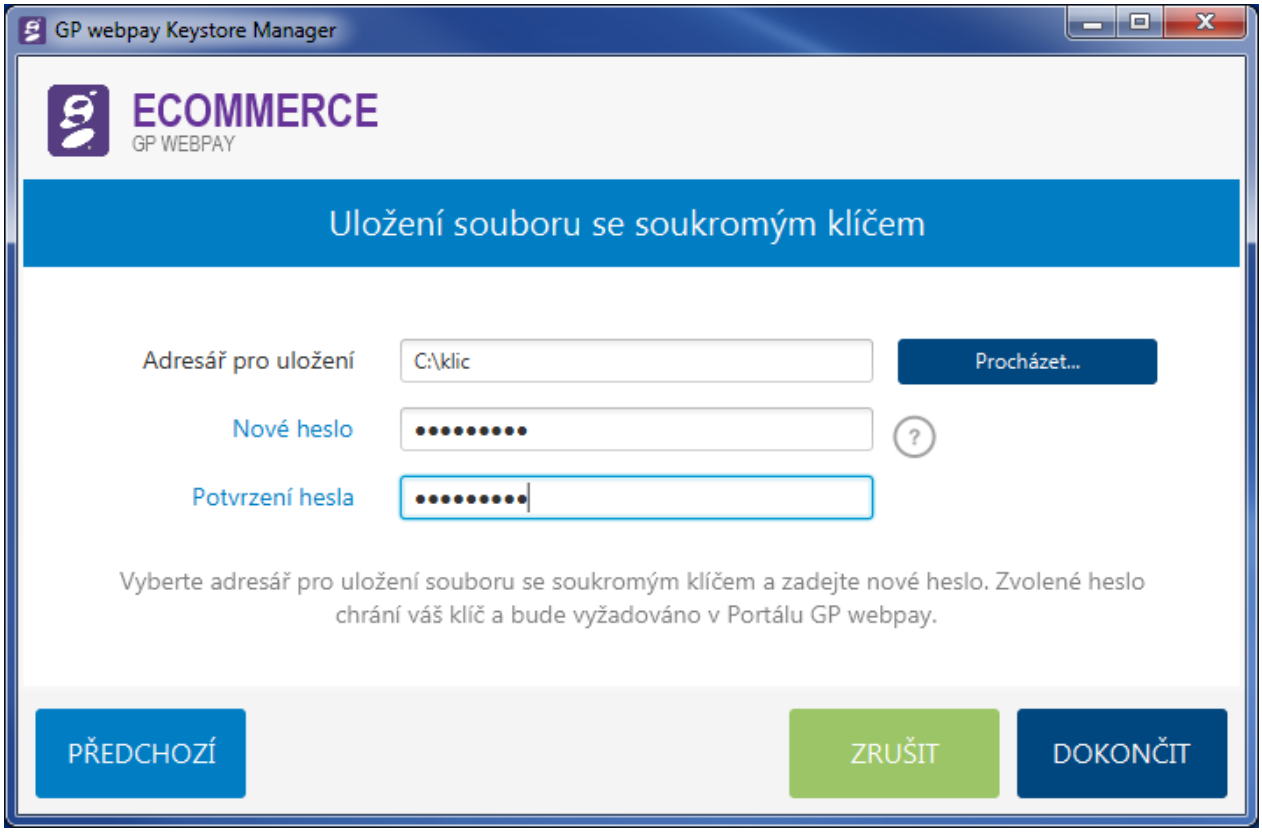


a pokračovat tlačítkem „Další“. V případě existence pouze jednoho soukromého klíče je tato obrazovka přeskočena.

Po ověření správnosti vstupního souboru, popř. potvrzení výběru klíče, je zobrazena výzva pro výběr cílového adresáře pro uložení konvertovaného souboru a požadavek na zadání nového hesla k soukromému klíči. Heslo musí být zadáno 2x, aby se předešlo překlepům.

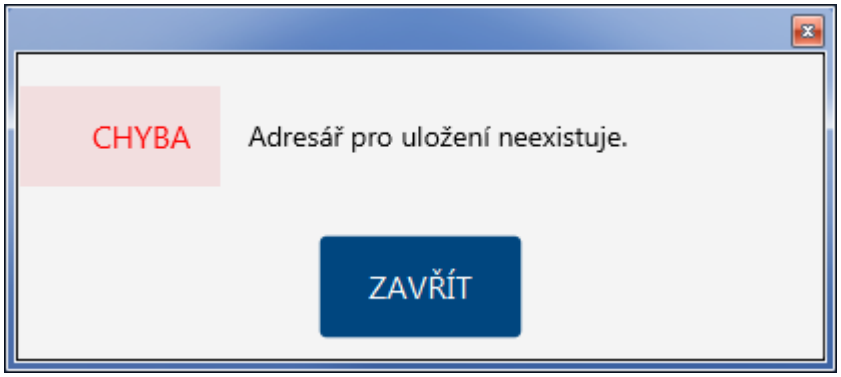
Heslo musí být dlouhé min. 8 znaků a obsahovat nejméně 3 typy z následujících požadovaných typů znaků:

- velké písmeno
- malé písmeno
- číslice
- speciální znak

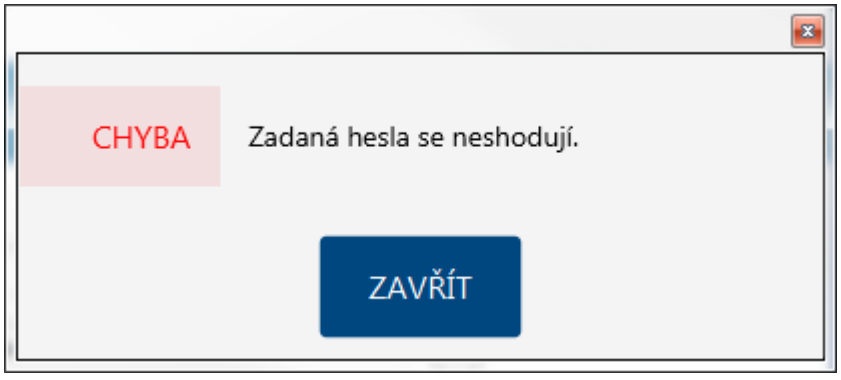


Po zadání všech potřebných informací je možné akci dokončit stiskem tlačítka „Dokončit“. Také je možno se vrátit k minulému kroku tlačítkem „Předchozí“, popř. pomocí tlačítka „Zrušit“ skočit zpět na úvodní obrazovku.

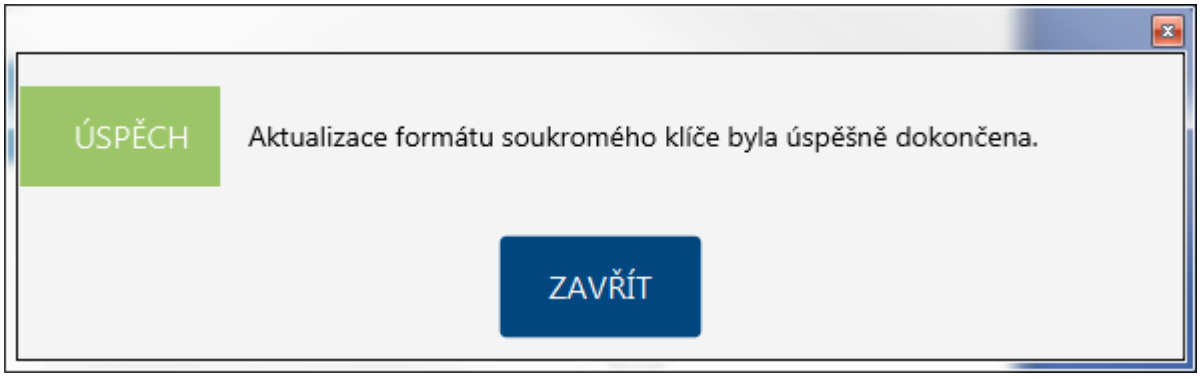
Pokud je zadán neexistující adresář, je při pokusu o pokračování zobrazeno hlášení:



V případě nerovnosti hesel je zobrazena informace:



Jestliže bylo vše zadáno v pořádku, dojde ke konverzi klíče a zobrazí se hlášení:



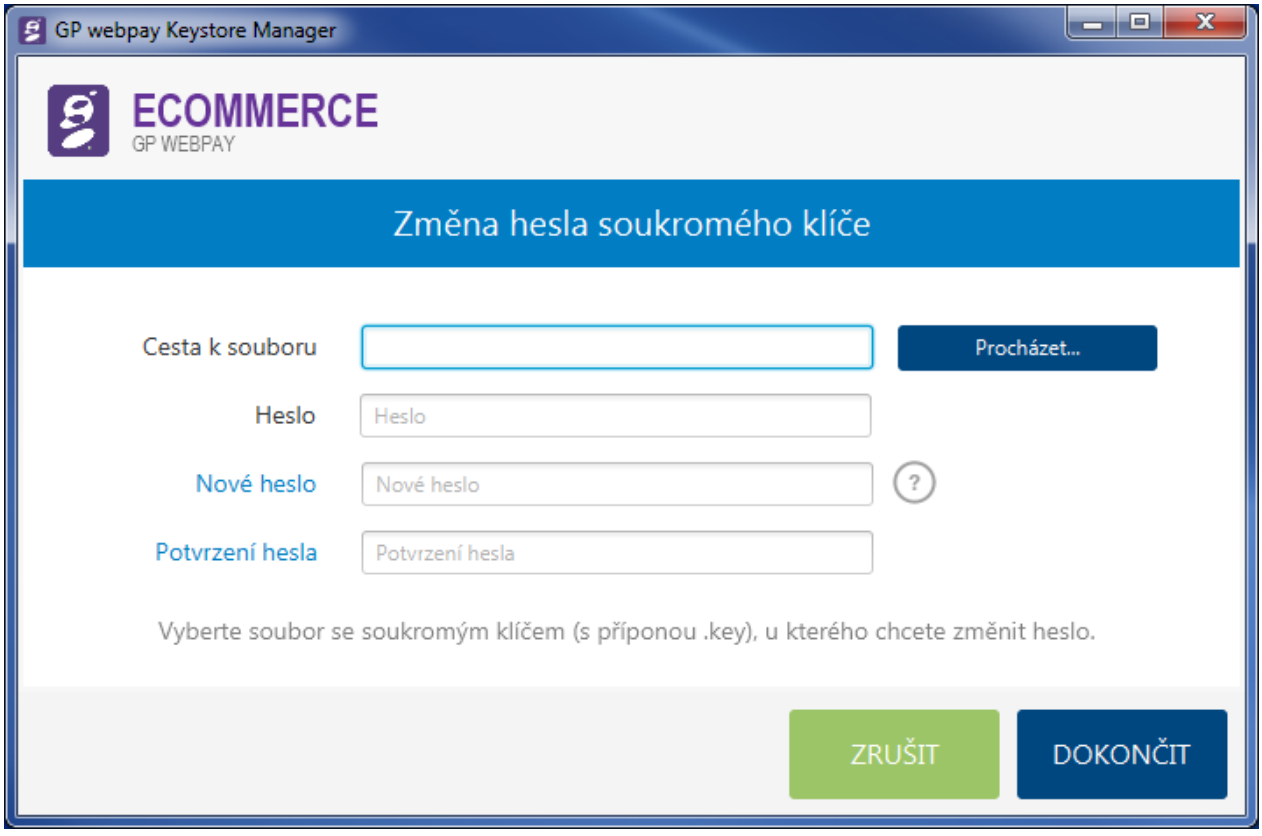
Ve zvoleném cílovém adresáři vznikne soubor se jménem „gpwebpay-pvk.key“. Soubor obsahuje soukromý klíč v textovém formátu PEM.

A po stisku tlačítka „Zavřít“ se aplikace vrátí na úvodní obrazovku.

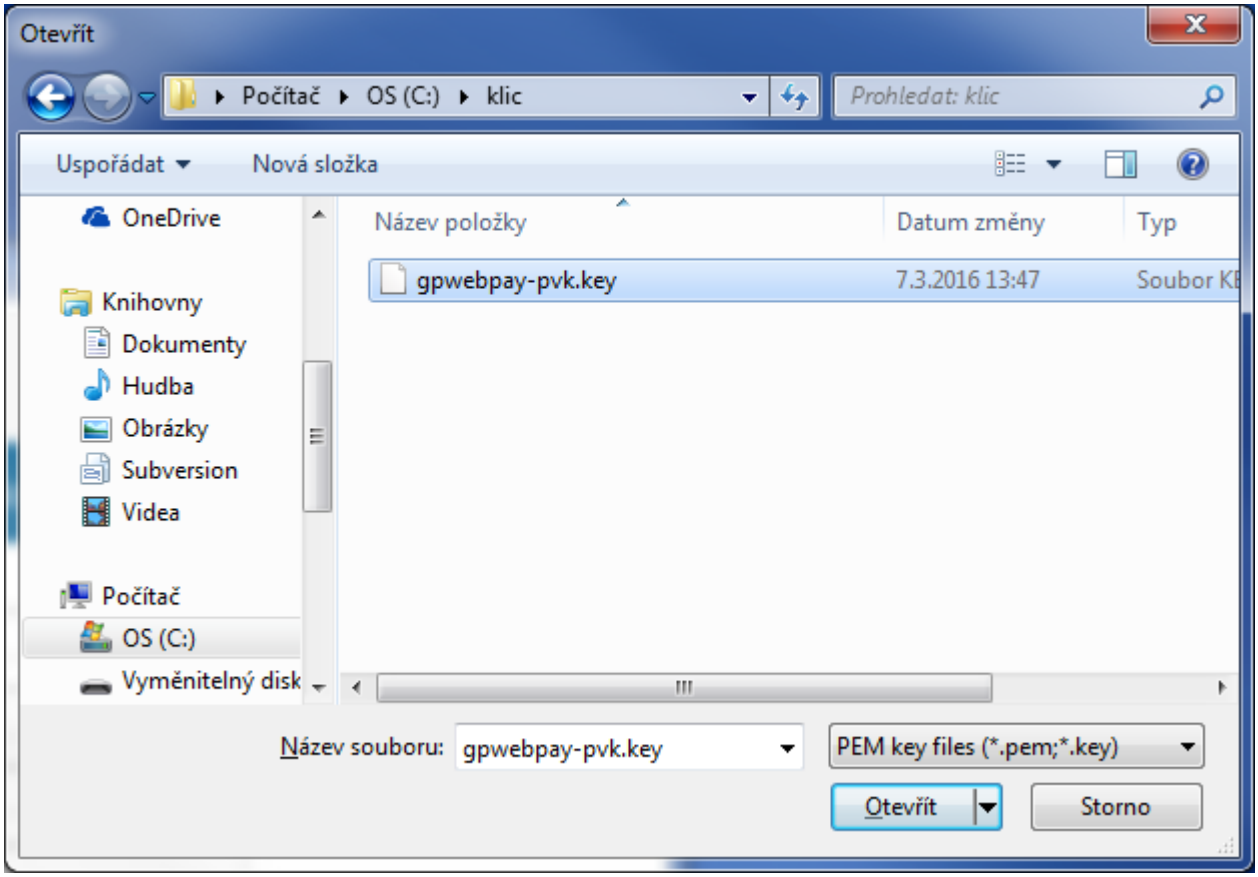
3.3.3 Změna hesla

Tato volba pracuje s novým formátem úložiště soukromého klíče a je nutné nejdříve soubor úložiště aktualizovat – viz předchozí kapitola, nebo použít soubor v novém tvaru získaný z Portálu GP webpay.

Po stisku „dlaždice“ dojde otevření nového okna pro zadání potřebných údajů:



Nejdříve je nutné, pomocí funkce „Procházet“ najít na souborovém systému adresář se souborem soukromého klíče:

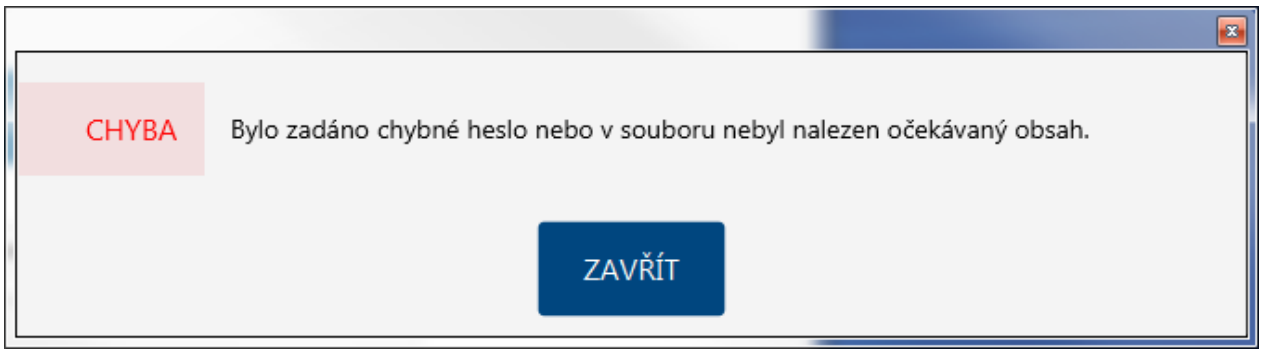


a patřičný soubor „Otevřít“. Dále je potřeba zadat heslo k původnímu klíči a heslo nové (samozřejmostí je ověření nového hesla duplicitním zadáním).

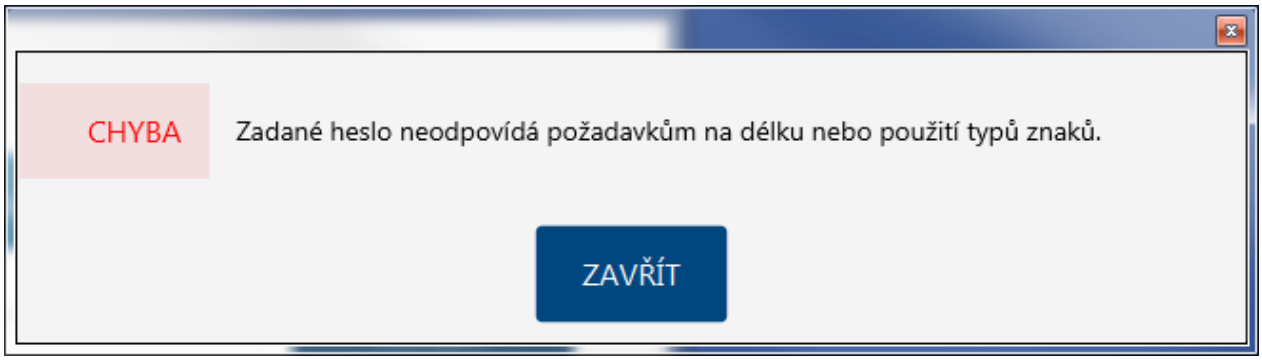
Heslo musí být dlouhé min. 8 znaků a obsahovat nejméně 3 typy z následujících požadovaných typů znaků:

- velké písmeno
- malé písmeno
- číslice
- speciální znak

Při chybném starém heslu nebo špatném formátu souboru je zobrazeno hlášení:

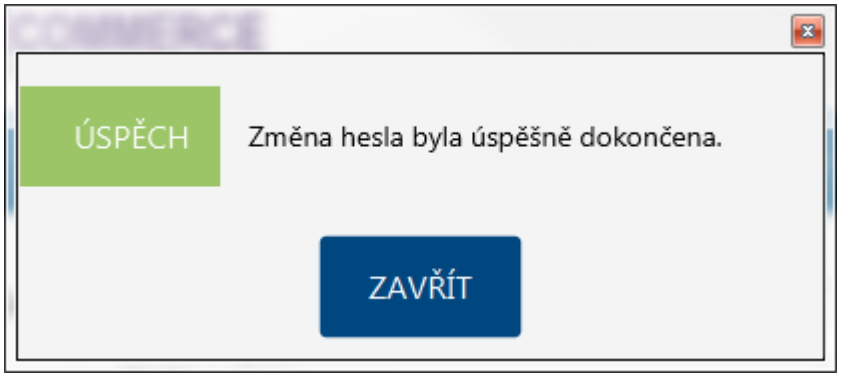


Pokud není zadáno nové heslo nebo nesplňuje potřebné bezpečnostní požadavky, tak je zobrazeno hlášení:



Jestliže se nové heslo neshoduje s potvrzením hesla, je tato situace indikována hlášením:

V případě správně zadaných hodnot je zobrazeno potvrzení o úspěšné změně hesla:



a následuje návrat na úvodní obrazovku.

3.3.4 Pro vývojáře

Sekce „PRO VÝVOJÁŘE“ je primárně určena pro programátory implementující platební bránu do e-shopu obchodníka.

Volba spustí proces konverze formátu úložiště soukromého klíče do další nejpoužívanějších formátů úložišť a současně uloží veřejnou část klíče do obecně použitelných formátů.

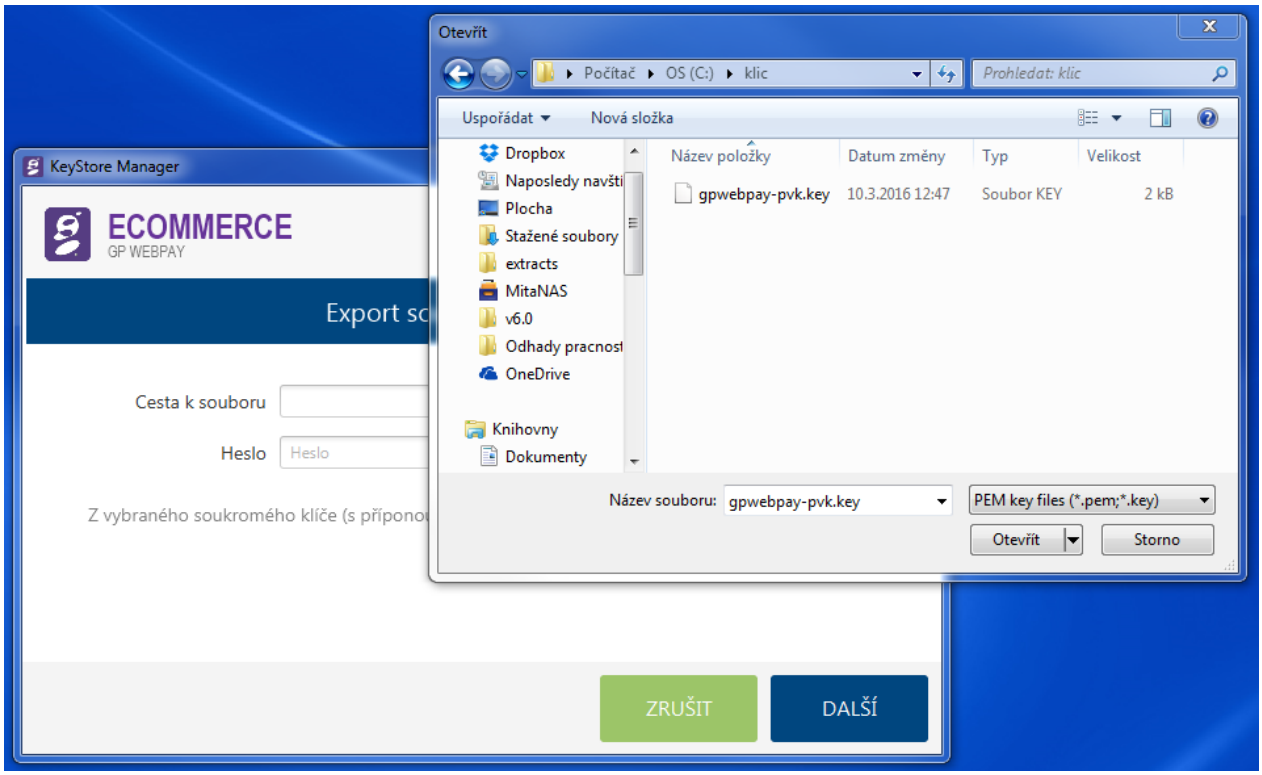
Vstupní formát úložiště soukromého klíče:

- textový formát PEM (PVK) – `gpwebpay-pvk.key`

Výstupní formáty úložišť a souborů:

- úložiště soukromého klíče:
    - JAVA formát JCEKS – `gpwebpay-pvk.jceks`
    - Microsoft PKCS12 – `gpwebpay-pvk.p12`
- soubor veřejného klíče:
    - textový formát PEM – `gpwebpay-pub.pem`
    - binární formát DER – `gpwebpay-pub.cer`

Prvním krokem konverze je výběr souboru úložiště soukromého klíče. Pomocí tlačítka „Procházet“ je nutné vybrat soubor s úložištěm soukromého klíče:

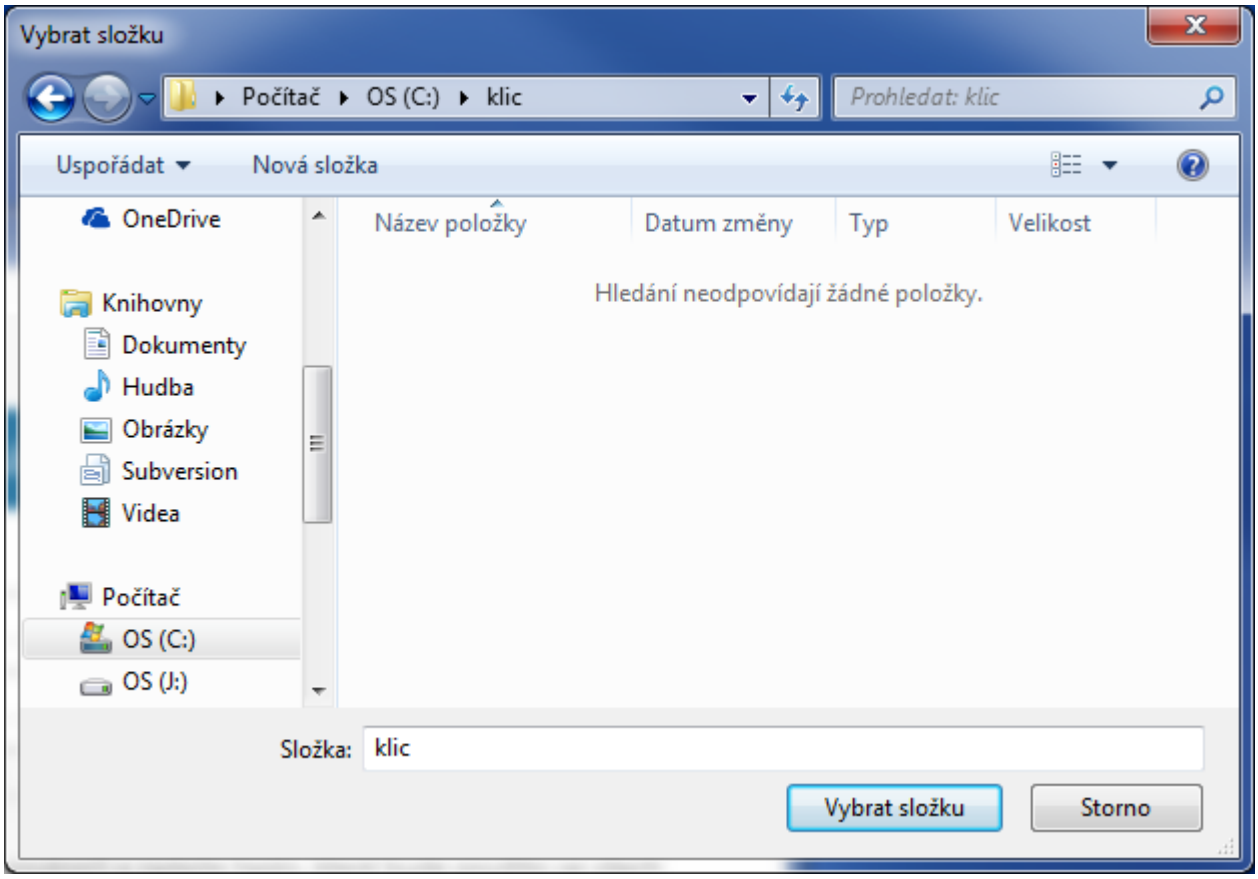


Soubor „Otevřít“, zadat heslo a stisknout tlačítko „Další“.

V případě chyby hesla nebo chybného vnitřního formátu souboru úložiště je zobrazeno hlášení:

V opačném případě následuje výzva k výběru výstupního adresáře pro uložení nově vzniklých souborů a zadání nového hesla k úložišti (to samé heslo se také použije k zabezpečení soukromého klíče v úložišti, lze použít i původní heslo).

Tlačítkem „Procházet“ se otevře okno pro výběr výstupního adresáře, po vyhledání potřebného místa v souborovém systému je potřeba volbu potvrdit tlačítkem „Vybrat složku“:



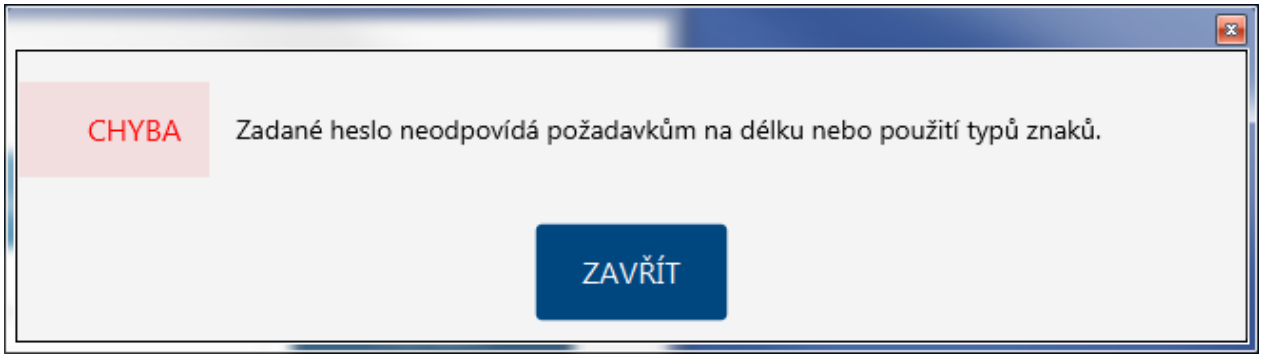
dále je potřeba zadat heslo a jeho potvrzení.

Heslo musí být dlouhé min. 8 znaků a obsahovat nejméně 3 typy z následujících požadovaných typů znaků:

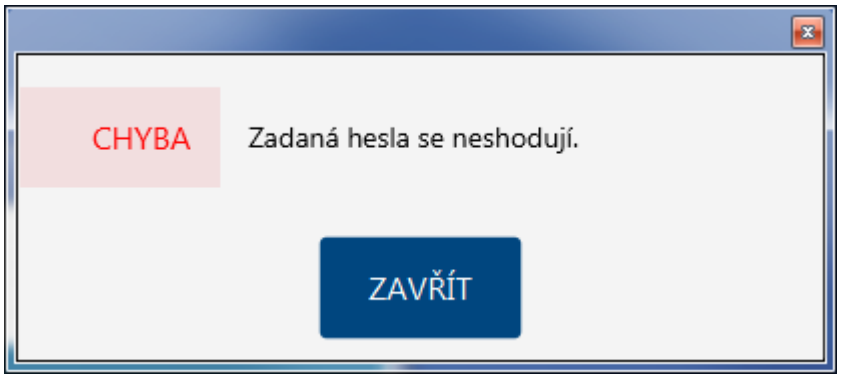
- velké písmeno
- malé písmeno
- číslice
- speciální znak

Následuje stisk tlačítka „Dokončit“.

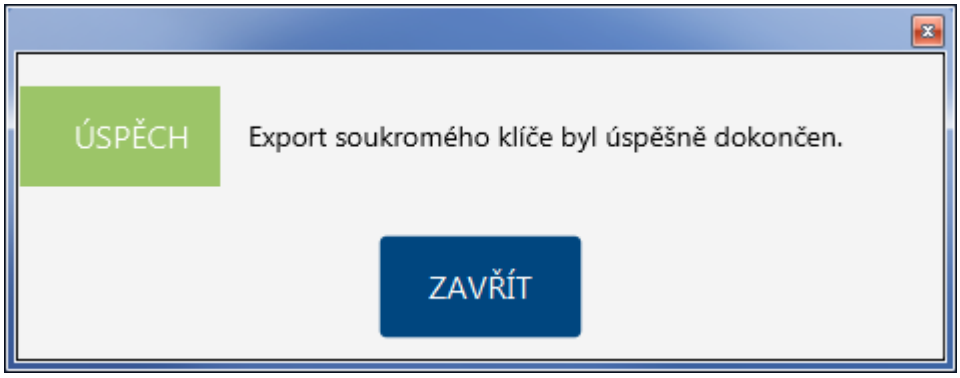
Pokud není zadáno nové heslo nebo nesplňuje potřebné bezpečnostní požadavky, tak je zobrazeno hlášení:



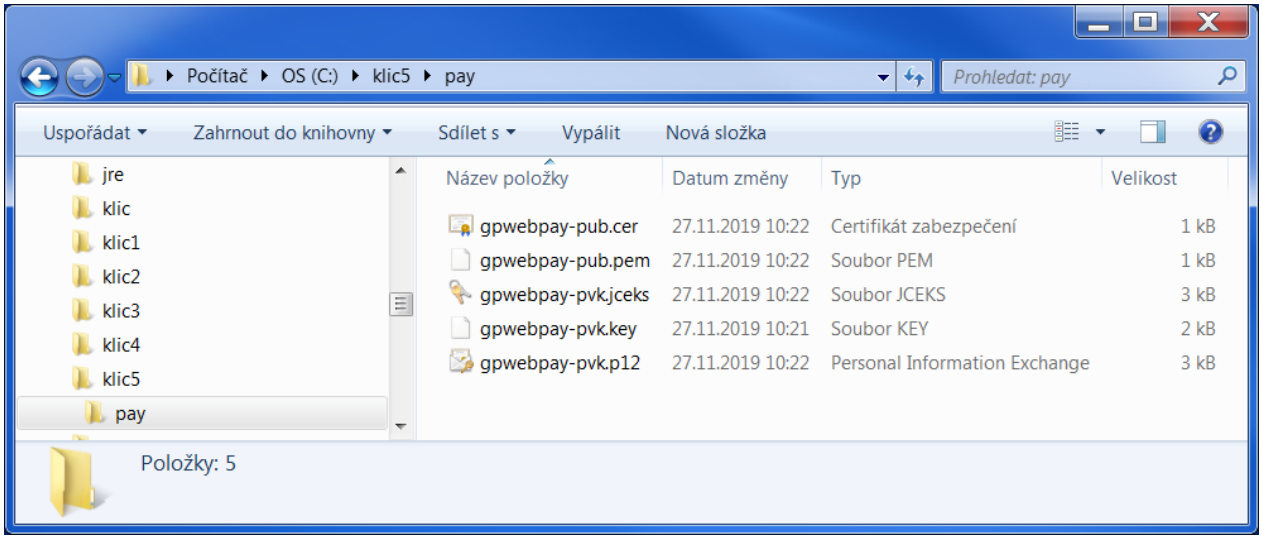
Jestliže se nové heslo neshoduje s potvrzením hesla, je tato situace indikována hlášením:



V případě správně zadaných hodnot je zobrazeno potvrzení o úspěšném exportu soukromého klíče:



Tímto je proces exportu soukromého klíče ukončen a v zadaném výstupním adresáři jsou vytvořeny tyto soubory:



- `gpwebpay-pvk.key` – originální soukromý klíč vygenerovaný v GP webpay Portálu
- `gpwebpay-pvk.jceks` – soukromý klíč v úložišti jazyku JAVA (JCEKS)

  – použitelný pro JSP/JAVA aplikace
- `gpwebpay-pvk.p12` – soukromý klíč v úložišti ve struktuře Microsoft (PKCS12 – P12)

  – použitelný pro aplikace .NET
- `gpwebpay-pub.pem` – textový PEM (PVK) formát veřejného klíče

  – použitelný pro PHP aplikace na ověření správnosti hodnoty podpisu vytvořeným pomocí soukromého klíče
- `gpwebpay-pub.cer` – binární DER formát veřejného klíče

  – použitelný pro .NET aplikace na ověření správnosti hodnoty podpisu vytvořeným pomocí soukromého klíče

  – formát pro zaslání veřejného klíče aplikační podpoře GP webpay, pokud selže nahrání veřejného klíče prostřednictvím Portálu GP webpay

Po stisku tlačítka „Zavřít“ se aplikace vrátí na úvodní obrazovku.

4. Podepisování zpráv

4.1 Obecný technický základ

4.1.1 Podepisování požadavku

(Kompletní příklad je uveden v kapitole Příklad podpisu.)

GP webpay API HTTP/WS přijme pouze ty požadavky, u kterých lze doložit, že původcem požadavku byl oprávněný subjekt, tedy obchodník, se kterým poskytovatel uzavřel smlouvu.

K prokázání původu požadavku slouží parametr DIGEST. Jeho obsah je vypočten na základě:

- zaslaných dat: tím je prokázáno, že obsah jednotlivých parametrů nebyl cestou změněn
- soukromého klíče: tím je prokázáno, že požadavek pochází od daného obchodníka

Při zahájení integrace obchodník vygeneruje s využitím portálu GP webpay soukromý klíč, který si obchodník bezpečně uloží a poskytne ho vývojáři pro integraci. Veřejný klíč obchodníka je během tohoto procesu automaticky uložen na server GP webpay a před přijetím požadavku od obchodníka se pomocí něj bude kontrolovat, zda obchodník podepsal požadavek svým soukromým klíčem.

Parametr DIGEST, obsažené v předávaných požadavcích, obsahuje elektronický podpis všech ostatních polí požadavku. Tento podpis zajišťuje integritu a nepopiratelnost předávaného požadavku.

Požadavky bez parametru DIGEST nebo s neodpovídajícím obsahem parametru DIGEST budou zamítnuty s důvodem:

- PRCODE=5 SRCODE=34 “Chybi povinne pole, DIGEST” nebo
- PRCODE =31 “Chybny podpis”.

Pro výpočet i ověření elektronického podpisu slouží jako datová zpráva řetězec sestavený jako součet (concatenation) textové interpretace hodnot všech parametrů (definovaných v API HTTP, ostatní parametry se ignorují) v zasílaném požadavku s výjimkou parametru DIGEST. Při sestavení vstupní zprávy je nutné dodržet stejné pořadí parametrů (viz tabulka v kapitole 3.1 Požadavek), jako v definici příkazu a oddělovat jednotlivé parametry oddělovačem “|“ (pipe, ascii 124, hexa 7C), kterému nesmí předcházet, ani nesmí být následován whitespace. URLEncode parametrů se použije pouze pro přenos dat, pro výpočet podpisu se musí použít původní data.

U příkazu CREATE_ORDER se tedy zdrojem pro výpočet parametru DIGEST stane hodnota, která vznikne zřetězením obsahů parametrů v tomto pořadí:

MERCHANTNUMBER + | + OPERATION + | + ORDERNUMBER + | + AMOUNT + | + CURRENCY + | + DEPOSITFLAG + | + MERORDERNUM + | + URL + | + DESCRIPTION + | + MD

V případě, že v požadavku není obsažen některý z nepovinných parametrů, parametr se přeskočí. Jestliže je zasílán parametr prázdný, pak je potřeba jej také zahrnout do výpočtu pro DIGEST a budou v řetězci dva oddělovače vedle sebe – ||.

Pokud obchodník posílá pouze povinné parametry, k výpočtu pole DIGEST slouží hodnota:

MERCHANTNUMBER + | + OPERATION + | + ORDERNUMBER + | + AMOUNT + | + CURRENCY + | + DEPOSITFLAG + | + URL

4.1.2 Ověření odpovědi

Všechny odpovědi z GP webpay obsahují také pole DIGEST, jehož obsah je vypočten:

- na základě údajů, obsažených v odpovědi
- a současně na základě soukromého klíče GP webpay

Při zahájení integrace si obchodník v portálu GP webpay stáhne veřejný klíč GPE, který mu slouží k ověření obsahu pole DIGEST.

Tímto způsobem se obchodník může přesvědčit, že:

- odpověď pochází skutečně od GP webpay
- odpověď nebyla cestou změněna.

Dále odpověď obsahuje také parametr DIGEST1, který dále zvyšuje bezpečnost odpovědi. Parametr DIGEST1 je tvořen stejně jako parametr DIGEST, ale je k parametrům pro ověření pole DIGEST přidán parametr „MERCHANTNUMBER“. Tento parametr není zasílán v odpovědi a obchodník si jej musí přidat sám, protože zná jeho hodnotu.

Výsledný řetězec pro ověření parametru DIGEST1 vypadá takto:

<řetězec pro parametr DIGEST> + | + MERCHANTNUMBER

4.1.3 Výpočet elektronického podpisu

Vstupy:

- datová zpráva (zpráva)
- privátní RSA klíč (s modulem délky K)

Výstupy:

- elektronický podpis (BASE64 kódovaný), délka přibližně K*1,5

Výpočet elektronického podpisu probíhá následujícím způsobem

a) ze zprávy je vypočtena hodnota hash funkce SHA-1 [3]

b) hash je zakódován na vstupní hodnotu pro RSA podpis algoritmem EMSA-PKCS1-v1_5-ENCODE podle části 9.2.1 [1]. Toto kódování je provedeno takto:

   01 | FF* | 00 | 30 21 30 09 06 05 2B 0E 03 02 1A 05 00 04 14 | hash

   kde znaky FF se opakují tolikrát, až je celková délka řetězce o jeden oktet kratší než modulus klíče. Znak | značí spojení řetězců (concatenation).

c) na výstupní hodnotě z b) je proveden RSA podpis v souladu s částí 8.1.1 [1] RSASSA-PKCS1-V1_5-SIGN

d) výstup c) je zakódován pomocí BASE64

4.1.4 Ověření elektronického podpisu

Vstupy:

- datová zpráva
- elektronický podpis (BASE64 kódovaný)
- veřejný RSA klíč

Výstupy:

- logická hodnota „ano“ – podpis je platný
- logická hodnota „ne“ – podpis není platný nebo nebylo jeho ověření možné.

Verifikace elektronického podpisu probíhá v souladu s částí 8.1.2 [1] v těchto hlavních krocích:

a) podle nastavení obchodníka v systému GPE je vybrán správný veřejný klíč a ověřena jeho integrita;

b) elektronický podpis je BASE64 dekódován;

c) výstup b) je dešifrován pomocí vybraného veřejného klíče;

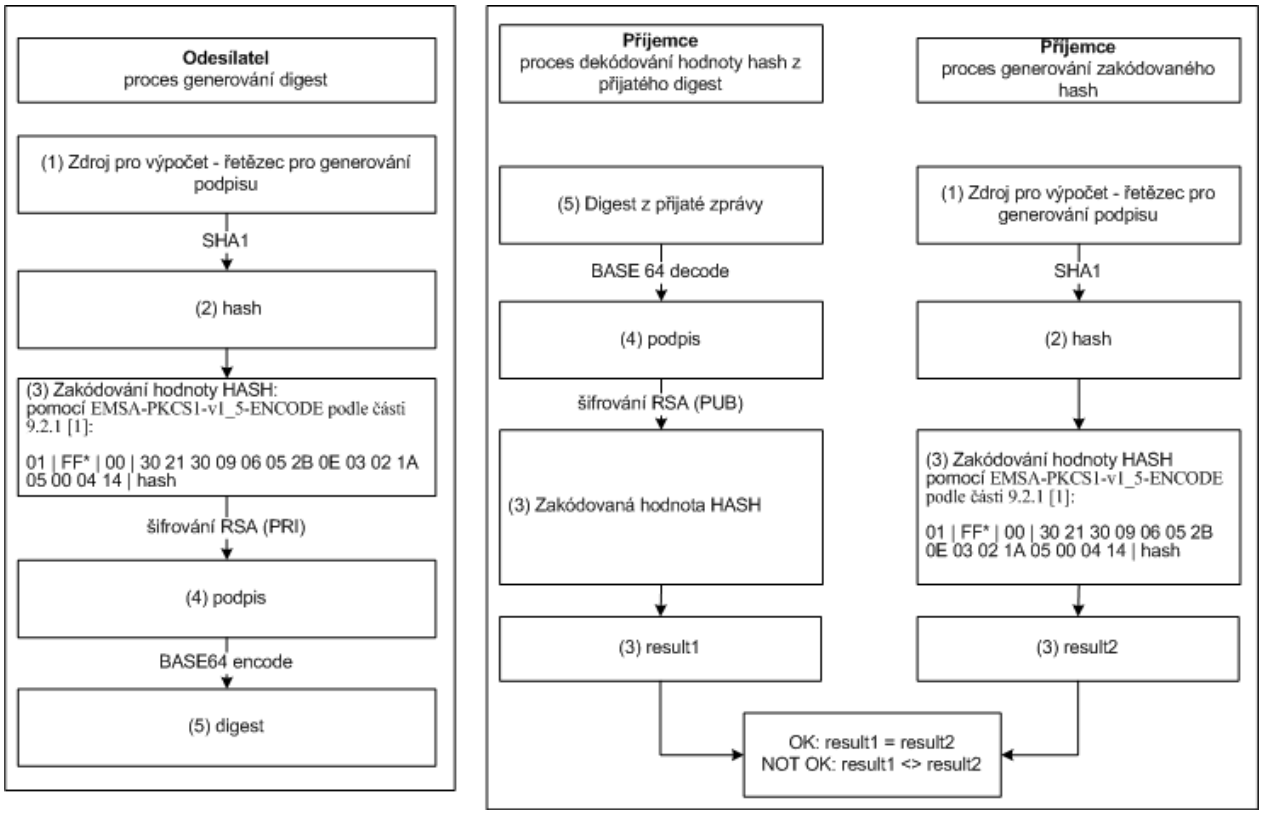
d) ze zprávy je vypočtena miniatura (hash) a zakódována v souladu s předchozí částí “Výpočet elektronického podpisu“ body a) b);

e) elektronický podpis dešifrovaný podle c) je porovnán s výsledkem podle d) a pokud jsou shodné, vrací funkce logickou pravdu (podpis je platný).

V opačném případě vrací funkce logickou nepravdu (podpis není platný).

Aplikace, která vyhodnocuje elektronický podpis, musí vyhodnotit podpis jako neplatný i v případě, kdy jeho ověření nebylo možné (například kvůli nedostupnosti klíče).

4.1.5 Grafické znázornění generování a ověření



4.1.6 Použité klíče

Pro vytvoření podpisu budou použity RSA klíče (keyPair) o délce modulu 2048 bitů. Při komunikaci mezi GP webpay a obchodníkem budou využity následující páry klíčů:

KeyPair GPE	Privátní klíč GPE (GPE_{PRI})	Použit pro výpočet elektronického podpisu zpráv odesílaných GPE.	
	Veřejný klíč (certifikát) GPE (GPE_{PUB})	Použit obchodníkem k ověření elektronického podpisu zpráv zasílaných GPE.	Bude předáván ve formě X509 certifikátu
KeyPair obchodníka	Privátní klíč obchodníka ($MERCH_{PRI}$)	Použit pro výpočet elektronického podpisu zpráv odesílaných obchodníkem.	
	Veřejný klíč (certifikát) obchodníka ($MERCH_{PUB}$)	Použit v GPE k ověření elektronického podpisu zpráv zasílaných obchodníkem.	Předáván ve formě X509 self-signed certifikátu

Funkce pro vytvoření soukromého klíče je součástí aplikace portál GP webpay. Lze použít i komerčně vydávané klíče, ale jejich platnost je omezena 1-2 roky (na rozdíl od klíče vytvořeného aplikací portál GP webpay, kde je platnost delší).

4.1.7 Logování

Aplikace, která ověřuje elektronický podpis, musí ve svých auditních záznamech uchovávat všechny informace o úspěšných i neúspěšných verifikacích elektronického podpisu.

Pro ověření záznamů je nutné logovat veškeré údaje nutné k ověření, respektive k opětovnému ověření elektronického podpisu. Jedná se především o elektronický podpis, pole, která byla využita pro jeho vytvoření a výsledek jeho ověření. V případě chybějících nebo nekompletních záznamů nebude možné uznat autentičnost takových transakcí.

4.1.8 Reference

Další informace o mechanismu výpočtu pole DIGEST lze nalézt v těchto dokumentech:

[1] RFC 2437, PKCS #1: RSA Cryptography Specifications, October 1998;

[2] XML-Signature Syntax and Processing, W3C Recommendation 12 February 2002, http://www.w3.org/TR/xmldsig-core/;

[3] RFC 3174 – US Secure Hash Algorithm 1 (SHA1), September 2001;

[4] RFC 2459 – Internet X.509 Public Key Infrastructure Certificate and CRL Profile, January 1999

Pro vytvoření elektronického podpisu je možné použít například následující kryptografické knihovny a komponenty:

JCE Cryptix: alternativní JCE Provider, poskytující algoritmus pro RSA/SHA1/PKCS#1 podpis, www.cryptix.org.

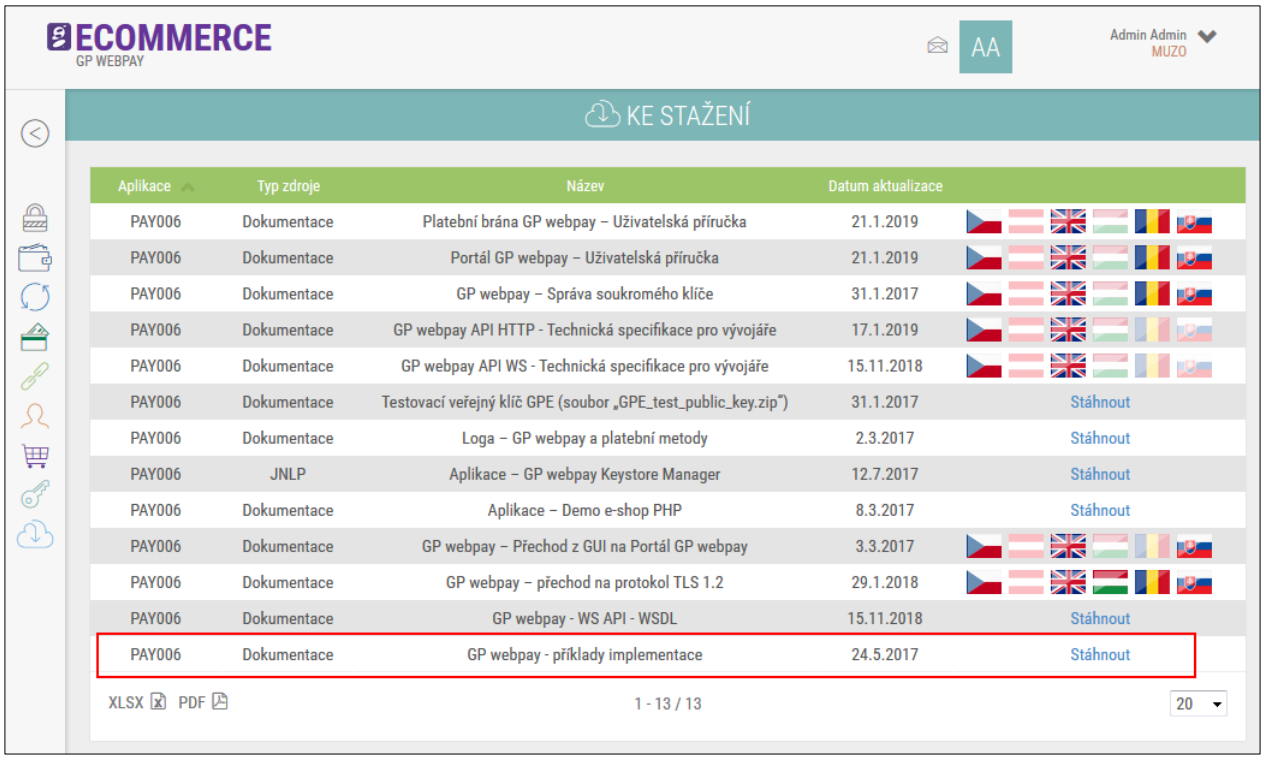
Bouncy Castle: alternativní JCA Provider, poskytující knihovny pro generování certifikátů a práci c PKCS#12 úložišti certifikátů, www.bouncycastle.org.

Crypto++ volně šiřitelná C++ knihovna kryptografických funkcí podporující také RSA/SHA1/PKCS#1 algoritmus, www.cryptopp.com

4.2 Příklady podpisu

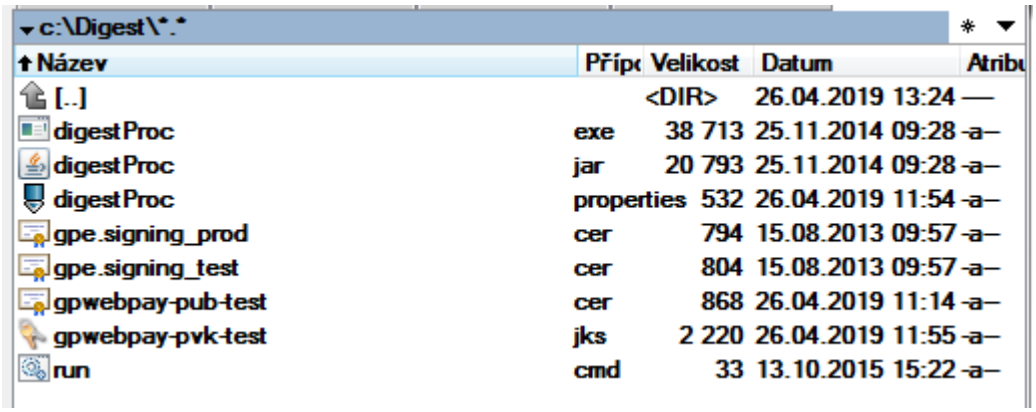
4.2.1 Testovací podepisovací klíč, test aplikace

Z bezpečnostních důvodů nelze vložit do dokumentu celou aplikaci (nelze ji pak zaslat mailem). Aplikaci je potřeba stáhnout ze sekce „KE STAŽENÍ“ z GP webpay Portálu:



Ze stažených příkladů, z adresáře “**overovani_podpisu-digest_verification**”, nakopírujte následující soubory “**run.cmd**”, “**digestProc.exe**”, “**digestProc.jar**” do adresáře s příklady.

Cílový adresář by měl obsahovat tyto soubory:



Soubory

`digestProc.exe` – aplikace spustitelná v prostředí MS Windows

`digestProc.jar` – Java archiv aplikace

`gpwebpay-pub-test.jks` – testovací keystore

`gpwebpay-pub-test.cer` – testovací veřejný klíč (certifikát)

`digestProc.properties` – konfigurační soubor

`gpe.signing_prod.cer` – produkční veřejný klíč GPE

`gpe.signing_test.cer` – testovací veřejný klíč GPE

Konfigurační soubor

Obsah souboru	Meaning in English
`` ####### Privatni klic ####### #nazev keystore s privatnim klicem keyStoreFile = gpwebpay-pvk-test.jks #heslo ke keystore keyStorePwd = Abcd1234 #jmeno (alias) privatniho klice privateKeyAlias = alias #heslo k privatnimu klici privateKeyPwd = Abcd1234 ####### Privatni klic ####### ####### Verejny klic ####### #jmeno souboru s verejnym klicem - cizi verejny klic publicKeyFile = gpwebpay-pub-test.cer #publicKeyFile = gpe.signing_test.cer #publicKeyFile = gpe.signing_prod.cer ####### Verejny klic ####### ####### URLEncoding ####### #predepsane enkodovani vstupnich parametru encoding = UTF-8 #digest na vstupu je URLEncoded # input_urlencoded = true input_urlencoded = false #na vyslednem digestu udelat URLEncode # output_urlencoded = true output_urlencoded = false ####### URLEncoding ####### ```	``` ####### Private key ####### #name of the keystore containing the private key keyStoreFile = gpwebpay-pvk-test.jks #password to the keystore keyStorePwd = Abcd1234 #name (alias) of the private key privateKeyAlias = alias #password to the private key privateKeyPwd = Abcd1234 ####### Private key ####### ####### Public key ####### #name of the file containing the public key - somebody else's public key publicKeyFile = gpwebpay-pub-test.cer #publicKeyFile = gpe.signing_test.cer #publicKeyFile = gpe.signing_prod.cer ####### Public key ####### ####### URLEncoding ####### #required encoding of input parameters encoding = UTF-8 #input digest is URLEncoded # input_urlencoded = true input_urlencoded = false #to make URLEncode over the output digest # output_urlencoded = true output_urlencoded = false ####### URLEncoding ####### ```

Spuštění

Java aplikace potřebují pro svůj běh tzv. Java Runtime. Java Runtime je běhové prostředí a je nutné jej do většiny operačních systémů doinstalovat. Java Runtime je zdarma dostupné na adrese: http://www.java.com

Z nabízených verzí je plně postačující verze Java SE JRE (Standard Edition, Java Runtime Enviroment). Aplikace je funkční i s Java Runtime jiných dodavatelů.

Pokud spustíte aplikaci bez parametrů:

```
digestProc.exe

java -jar digestProc.jar
```

tak se vypíše nápověda:

Zobrazená nápověda:	Meaning in English
Digest Processing, (c) 08/2004 Dimitrij R. Holovka Pouziti: - vypocet podpisu: java -jar digestProc.jar -s <retezec pro vypocet> - overeni podpisu: java -jar digestProc.jar -v <retezec pro vypocet> <podpis> nebo - vypocet podpisu: digestProc.exe -s <retezec pro vypocet> - overeni podpisu: digestProc.exe -v <retezec pro vypocet> <podpis> Pokud <retezec pro vypocet> obsahuje mezery je nutne jej dat do uvozovek.	Digest Processing, (c) 08/2004 Dimitrij R. Holovka Usage: - generation of signature: java -jar digestProc.jar -s <string for generation> - verification of signature: java -jar digestProc.jar -v < string for generation> <digest> or - generation of signature: digestProc.exe -s <string for generation > - verification of signature: digestProc.exe -v <string for generation > <digest> If <string for generation> contains spaces, it is necessary to put it into quotation marks.

Příklady:

Generování podpisu:

Spuštění: `digestProc.cmd -s "hello world"`

Výsledek:

Výsledek:	Meaning in English
Digest Processing, (c) 08/2004 Dimitrij R. Holovka ---- Nacitani parametru --- OK Soubor keyStore: test.ks Heslo keyStore: changeit Alias privatniho klice: paymuzo Heslo privatniho klice: changeit Privatni klic: OK Verejny klic: OK Soubor publicKeyStore: test.cer Verejny klic: OK URLEncoded podpis na vstupu: false Provadet URLEncode na vystupu: false Kodova stranka: WINDOWS-1250 ---------- Vytvoreni podpisu pomoci privatniho klice: Delka podpisu: 344 Soubor s podpisem: digestProc.sign Podpis:	Digest Processing, (c) 08/2004 Dimitrij R. Holovka ---- Reading of parameters --- OK File keyStore: test.ks KeyStore password: changeit Alias of the private key: paymuzo Private key password: changeit Private key: OK Public key: OK File publicKeyStore: test.cer Public key: OK Input URLEncoded signature: false To make URLEncode on the output: false Code page: WINDOWS-1250 ---------- Generation of signature using the private key: Length of signature: 344 File with signature: digestProc.sign Signature:

Podpis:

```
et2dmSt7+9y43r6pCjAAiLKzDws9VgAm/Qh1ZZxyy1QWS43WLHlywUHBS2f68hXPARAMPOaCpLiswz8iS3pameakq
wTdlmeff6hOAR2s1/ACToo/dICYZoCkXdXg8dKzHqcWVnqeiGBY+NmewyH2/KHg/IzxxqF6AeLbTJsH5drPlUBVsT
```

```
bRprUfNAzoCizLRLM3fLXvy8bvDte35cq7Syly6snFjo3crUBwEzXLLrjU/XEYPvzL/XwNo3IZ7PhGspdx8gSWYj7
7t2zasLh/AxpkI8qYqwQlEYzmBHKB3NFetjJoR9jDCvFFtxMvI4bHYHKnjJ1mPyBPxJp00sxp/w==
```

Aplikace vygeneruje – za použití privátního klíče „gpwebpay“ uloženého v keystore „test.ks“ – podpis textu „Hello world“. Podpis je zobrazen na obrazovce a současně uložen do souboru „digestProc.sign“.

Ověření podpisu:

Spuštění: `digestProc.cmd -v "hello world"`

```
"Cs6FVQJYZpHptn7/FC3SHlcfuz35slDYQZYC7i5YEoK24osGFGqD/KsI7VD3Rezqdz5I8CTI8CXOhTfnA0DwKjzH
8nRvxMTxytQdC+saMwpLviCnLFzL8v2hWf2Ef+YHYYitrHCL9iaE9ggusvwky8kUhA6DjBoOqQeOXk30RZOOr8feV
X0FDyvVCZxWcNMgHauvN+4o9okmaNs2btvmsd38LTnXdoWDicqx+xxjIEliJWutd0YLieJGp569SOAwHtYL5t07c8
wWzFzOIhO+nufSpiKUL1NCs8LtXshM3r1Ye2eWXKYv+h072/GXIHjgTrY7MhqINMNo7s/XR7FRpA=="
```

Výsledek:

Výsledek:	**Meaning in English**
Digest Processing, (c) 08/2004 Dimitrij R. Holovka ---- Nacitani parametru --- OK Soubor keyStore: test.ks Heslo keyStore: changeit Alias privatniho klice: paymuzo Heslo privatniho klice: changeit Privatni klic: OK Verejny klic: OK Soubor publicKeyStore: test.cer Verejny klic: OK URLEncoded podpis na vstupu: false Provadet URLEncode na vystupu: false Kodova stranka: WINDOWS-1250 ---------- Overeni podpisu pomoci verejneho klice: Vysledek overeni: true	Digest Processing, (c) 08/2004 Dimitrij R. Holovka ---- Reading of parameters --- OK KeyStore file: test.ks KeyStore password: changeit Alias of the private key: paymuzo Private key password: changeit Private key: OK Public key: OK File publicKeyStore: test.cer Public key: OK Input URLEncoded signature: false To make URLEncode on the output: false Code page: WINDOWS-1250 ---------- Verification of signature using the public key: Result of verification: true

Aplikace pomocí, **veřejného klíče** (certifikátu) uloženého v souboru „test.cer“, ověří podpis řetězce „hello world“. Výsledek ověření zobrazí na posledním řádku výpisu – „`Vysledek overeni: true`“.

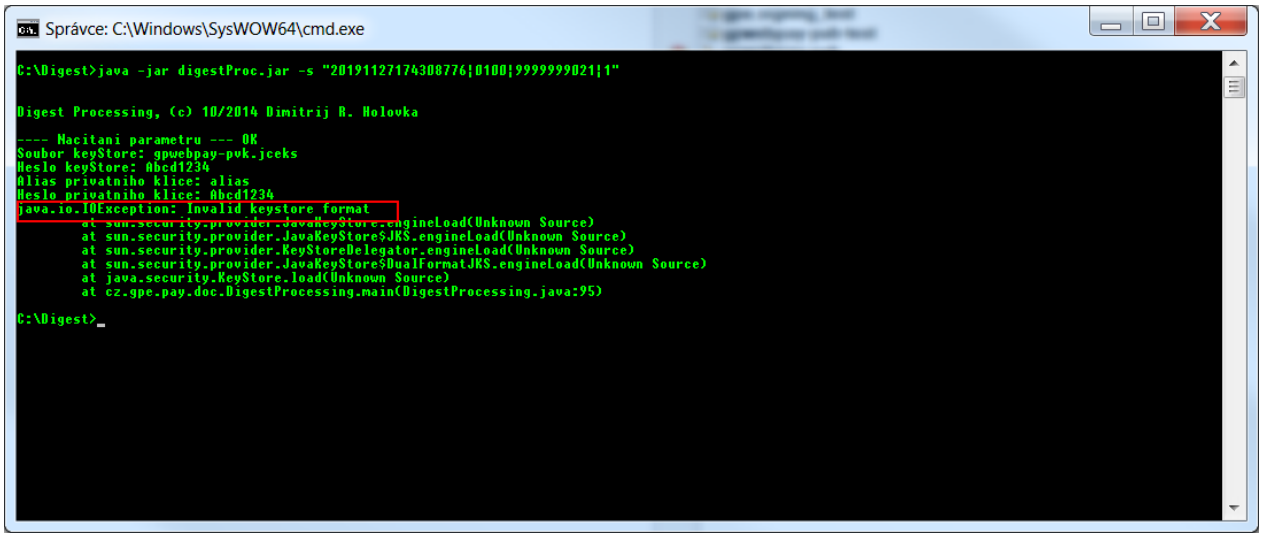
Pro ověření odpovědi ze systémů GP webpay je potřeba změnit v konfiguračním souboru parametr „`publicKeyFile`“ takto:

- testovací prostředí: `gpe.signing_test.cer`
- produkční prostředí: `gpe.signing_prod.cer`

4.2.2 Příklad podpisu

! Toto je pouze příklad, nefunguje v reálném testovacím prostředí (používá testovací klíč bez odpovídající veřejné části na GP webpay serveru) !

Pokud se při pokusu o podepsání/ověření řetězce objeví chyba „Invalid keystore format“, je použit jiný typ keystore (JCEKS, P12 …) než podporovaný „JKS“:



Je potřeba použitý keystore se soukromým klíčem převést na správný formát pomocí některého z programů pro práci s klíči – např. KeyStore Explorer, OpenSSL …

4.2.2.1 HTTP

4.2.2.1.1 Požadavek

HTTP požadavek
https://test.3dsecure.gpwebpay.com/pgw/order.do?MERCHANTNUMBER=9999999021&OPERATION=CREATE_ORDER&ORDERNUMBER=157487125803&AMOUNT=100&CURRENCY=203&DEPOSITFLAG=1&MERORDERNUM=155912254545&URL=https%3A%2F%2Flocalhost%3A443%2Fdemoshop%2Fpayment%2Fpayment.php&userparam2=59452C6A0381B48B3B164A80E202983F542759CC17AF36DE37B4CDB4B9908EB7&email=dholovka%40gpe.cz&DIGEST=BZlXwJJGku1m2uhyHU%2ByIyhzRv%2BVKfNdI%2F1Y5pSG61FfJb88JzYlLs7CGfhY6%2Bonle3q0sDH6V%2FM9EF4uIWMcSJxBpqzPmh7U6ud7nYa3UINr7ljSU3WC8FQd8qIc%2FLWeVFgh%2BPgxq0cdI47vvFZXuoBtzFpZQZWI0AOOQxhKAzZ4UrRafdNGYKtYI1BwZD4u5Aq3wSrOOBRBu%2BFompn%2BIWAN8xpAySA2A9VvBEHGvD1tFLSMIVlCCOMfqq0EEQj0TceXrWoQ8Y02gQoolRcCDb1geOsshELdaseuKUtcFnsZE49%2B700G1lxYz9%2F4RNMOTa%2FPpwTVBc00qlqlgfjWQ%3D%3D

Data pro podpis:

9999999021|CREATE_ORDER|157487125803|100|203|1|155912254545|https://localhost:443/demoshop/payment/payment.php|59452C6A0381B48B3B164A80E202983F542759CC17AF36DE37B4CDB4B9908EB7|dholovka@gpe.cz

Podpis:

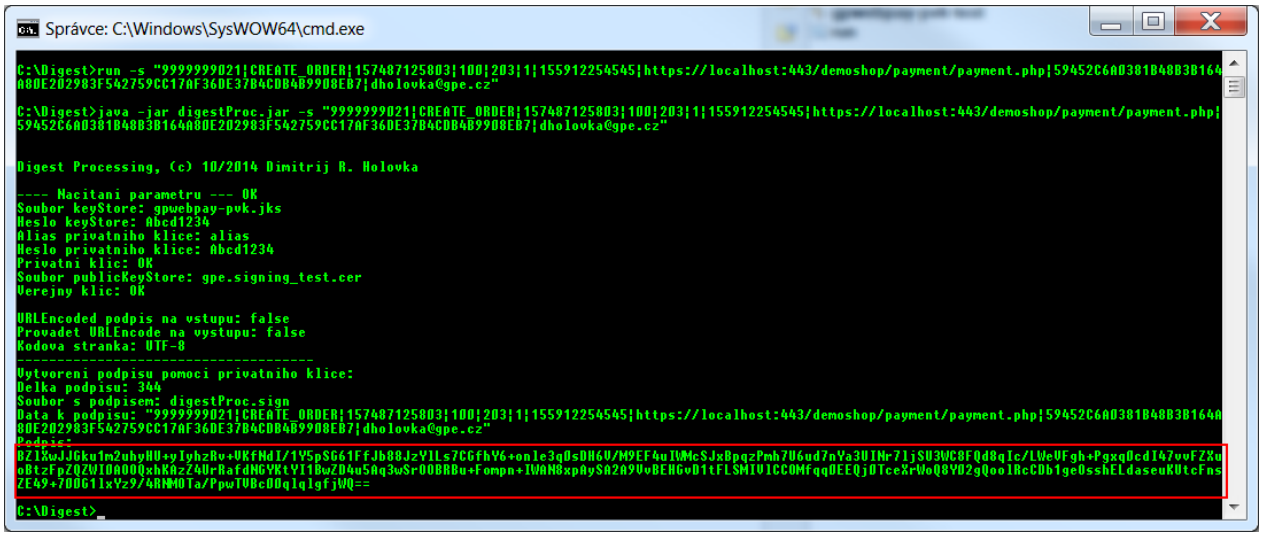
BZlXwJJGku1m2uhyHU+yIyhzRv+VKfNdI/1Y5pSG61FfJb88JzYlLs7CGfhY6+onle3q0sDH6V/M9EF4uIWMcSJxBpqzPmh7U6ud7nYa3UINr7ljSU3WC8FQd8qIc/LWeVFgh+Pgxq0cdI47vvFZXuoBtzFpZQZWI0AOOQxhKAzZ4UrRafdNGYKtYI1BwZD4u5Aq3wSrOOBRBu+Fompn+IWAN8xpAySA2A9VvBEHGvD1tFLSMIVlCCOMfqq0EEQj0TceXrWoQ8Y02gQoolRcCDb1geOsshELdaseuKUtcFnsZE49+700G1lxYz9/4RNMOTa/PpwTVBc00qlqlgfjWQ==

Příkaz pro spuštění aplikace:

java -jar digestProc.jar -s "9999999021|CREATE_ORDER|157487125803|100|203|1|155912254545|https

://localhost:443/demoshop/payment/payment.php|59452C6A0381B48B3B164A80E202983F542759CC17AF36DE37B4CDB4B9908EB7|dholovka@gpe.cz"

Výsledek je zobrazen na obrazovce a uložen v souboru "**digestProc.sign**":



4.2.2.1.2 Odpověď

HTTP odpověď
https://localhost:443/demoshop/payment/payment.php?OPERATION=CREATE_ORDER&ORDERNUMBER=157487125803&MERORDERNUM=155912254545&PRCODE=0&SRCODE=0&RESULTTEXT=OK&DETAILS=59452c6a0381b48b3b164a80e202983f8e9c5459c948e292465bc638b8be647d&USERPARAM1=2F89879EAF57B52B37E23DFD1D2B1BA6567A13BC2547F16DBB54EF5BE3A743A7&TOKEN=AA74E7D735D3201A926971BE5A92C8CE14D2E685DC399E4A3E2BE12C64605EC7&EXPIRY=2012&ACCODE=69Z4IV&ACSRES=A&PANPATTERN=405607******0016&DAYTOCAPTURE=04122019&ACRC=00&RRN=000001267633&DIGEST=o9uwRov2%2BWkF5QcZ5EhL0Ka7d8cObEW2n59j62WFDWcAL5HUD2g5%2F9lwfo3ZD2EKvaNtXmn9QYLXK4mwlgRWpdjCtuYgJxmHBa%2BsE%2FfUSRVBV3y2Qt47eLZNH8UGtSXlHy3IzGt%2FCX4UYeOwoZ%2BI4keiQr1F%2FRC7w6AOQCLM5rSEtNhM487eM8A5kI2E%2BpuzQeVtSr9X3nCa3N531N1Fm9p96bQKejpI7nX9%2FJJk8pk6n0MXBGzYbxX%2B0YNt2sUlCfFJrCg44LWQsBZYl1vLlxHdZHfKK6F6LBwMTAb0qD5zE1fQUdzLYPWFalb4tTtHC6srwv3p1e5A0Z3efWzPOA%3D%3D&DIGEST1=hudnVKmuIHpBG4U4BXsNGTdxoiOq9860K%2Fntgher8gwHab2TOavI5DCNs2hURSAC9nX4nDJa30g2E7FVigWI%2BYf45kHgQ2MQEASGQgllTDqyHRoYFZwqM69hXHoUgd3RXEZDmYnLm6VvJ4PvVKqCa8FBC2JUyo9saKN2rXVjv2yox265u724r7JoIlGg5ka%2F5Schu7bxRuG%2BdID4YNePR4gz9C6K9V8b2bWQ%2FifYOEHGtySMfZ7BrPIDXRNRJM5YrnGUlLghsYfpJwI%2Bp4d5kpUSHkiwQ5yNXhLPI8xER%2BCO%2Bmve9xV6n%2FvphiQo8T1RBcp2bAK4IOAr5%2FT7%2BTBVtQ%3D%3D

Data pro ověření:

CREATE_ORDER|157487125803|155912254545|0|0|OK|59452c6a0381b48b3b164a80e202983f8e9c5459c948e292465bc638b8be647d|2F89879EAF57B52B37E23DFD1D2B1BA6567A13BC2547F16DBB54EF5BE3A743A7|AA74E7D735D3201A926971BE5A92C8CE14D2E685DC399E4A3E2BE12C64605EC7|2012|A|69Z4IV|405607****0016|04122019|00|000001267633|9999999021**

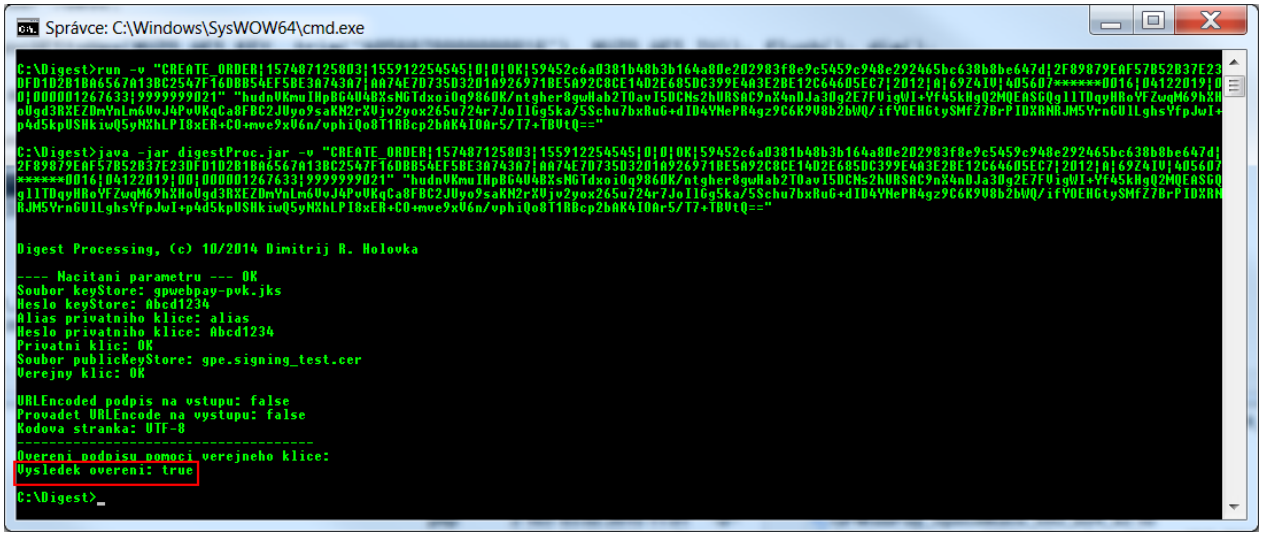
Podpis (DIGEST1 – URLDecoded):

hudnVKmuIHpBG4U4BXsNGTdxoiOq9860K/ntgher8gwHab2TOavI5DCNs2hURSAC9nX4nDJa30g2E7FVigWI+Yf45kHgQ2MQEASGQgllTDqyHRoYFZwqM69hXHoUgd3RXEZDmYnLm6VvJ4PvVKqCa8FBC2JUyo9saKN2rXVjv2yox265u724r7JoIlGg5ka/5Schu7bxRuG+dID4YNePR4gz9C6K9V8b2bWQ/ifYOEHGtySMfZ7BrPIDXRNRJM5YrnGUlLghsYfpJwI+p4d5kpUSHkiwQ5yNXhLPI8xER+CO+mve9xV6n/vphiQo8T1RBcp2bAK4IOAr5/T7+TBVtQ==

Příkaz pro spuštění aplikace:

```
java -jar digestProc.jar -v
"CREATE_ORDER|157487125803|155912254545|0|0|OK|59452c6a0381b48b3b1
64a80e202983f8e9c5459c948e292465bc638b8be647d|2F89879EAF57B52B37E2
3DFD1D2B1BA6567A13BC2547F16DBB54EF5BE3A743A7|AA74E7D735D3201A92697
1BE5A92C8CE14D2E685DC399E4A3E2BE12C64605EC7|2012|A|69Z4IV|405607**
****0016|04122019|00|000001267633|9999999021"
"hudnVKmuIHpBG4U4BXsNGTdxoiOq9860K/ntgher8gwHab2TOavI5DCNs2hURSAC9
nX4nDJa30g2E7FVigWI+Yf45kHgQ2MQEASGQgllTDqyHRoYFZwqM69hXHoUgd3RXEZ
DmYnLm6VvJ4PvVKqCa8FBC2JUyo9saKN2rXVjv2yox265u724r7JoIlGg5ka/5Schu
7bxRuG+dID4YNePR4gz9C6K9V8b2bWQ/ifYOEHGtySMfZ7BrPIDXRNRJM5YrnGUlLg
hsYfpJwI+p4d5kpUSHkiwQ5yNXhLPI8xER+CO+mve9xV6n/vphiQo8T1RBcp2bAK4I
OAr5/T7+TBVtQ=="
```

Výsledek je zobrazen na obrazovce:



4.2.2.2 WS

4.2.2.2.1 Požadavek

WS požadavek

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:v1="http://gpe.cz/pay/pay-ws/proc/v1" xmlns:type="http://gpe.cz/pay/pay-ws/proc/v1/type">
   <soapenv:Header/>
   <soapenv:Body>
      <v1:getPaymentStatus>
         <v1:paymentStatusRequest>
            <type:messageId>20191127174308776</type:messageId>
            <type:provider>0100</type:provider>
            <type:merchantNumber>9999999021</type:merchantNumber>
            <type:paymentNumber>1</type:paymentNumber>
            <type:signature>
lTFcQSdDX9BMwup7B3YgC9SWKUROQYbuIe4TZmtGIFEP4kAMsLkkHMmFCJjJcDtaU3x5NVTVSJTSel2J1Q8DIemhf/2eWWmDgPps
Dj/E/bg2mZQINr0R8c2OAMnXyffEX1ky3s1siAcu2aBDIYgpB9wi9fVvhmpIkLQY2QzDCrZtPMzsvLZ54v4uOScgXgTLwXxSeS04
xDZQlA8Ng6ojX/d+lIvdPEk4qD9RjDKfuMvRj1WO4i+cQPY9u9QN3aJtmda2VO3JicoMbVhUETlHdjHD/xAcARS6St8/xIfuzg25
SpoOrQnjCwB/dNNPRsx5F+NUrZvDY8uPxqjCGuwayA==</type:signature>
         </v1:paymentStatusRequest>
      </v1:getPaymentStatus>
   </soapenv:Body>
</soapenv:Envelope>
```

Data pro podpis: **20191127174308776|0100|9999999021|1**

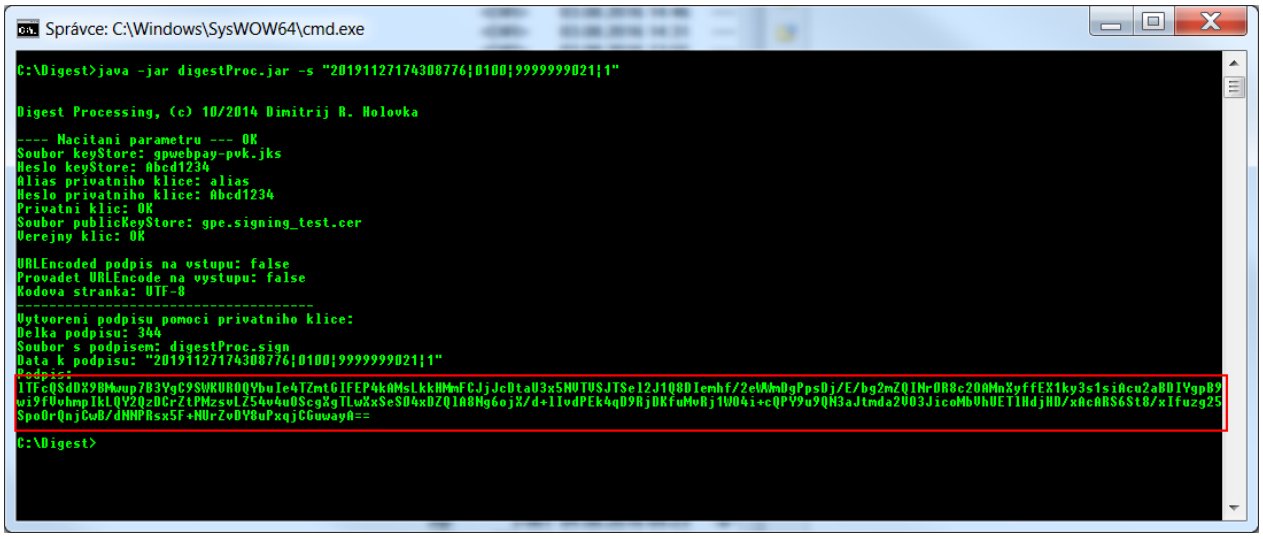
Podpis:

lTFcQSdDX9BMwup7B3YgC9SWKUROQYbuIe4TZmtGIFEP4kAMsLkkHMmFCJjJcDtaU3x5NVTVSJTSel2J1Q8DIemhf/2eWWmDgPpsDj/E/bg2mZQINr0R8c2OAMnXyffEX1ky3s1siAcu2aBDIYgpB9wi9fVvhmpIkLQY2QzDCrZtPMzsvLZ54v4uOScgXgTLwXxSeS04xDZQlA8Ng6ojX/d+lIvdPEk4qD9RjDKfuMvRj1WO4i+cQPY9u9QN3aJtmda2VO3JicoMbVhUETlHdjHD/xAcARS6St8/xIfuzg25SpoOrQnjCwB/dNNPRsx5F+NUrZvDY8uPxqjCGuwayA==

Příkaz pro spuštění aplikace:

java -jar digestProc.jar -s "20191127174308776|0100|9999999021|1"

Výsledek je zobrazen na obrazovce a uložen v souboru “**digestProc.sign**”:



4.2.2.2.2 Odpověď

WS odpověď

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
   <soapenv:Body>
      <ns4:getPaymentStatusResponse xmlns:ns4="http://gpe.cz/pay/pay-ws/proc/v1"
xmlns="http://gpe.cz/gpwebpay/additionalInfo/response"
xmlns:ns5="http://gpe.cz/gpwebpay/additionalInfo/response/v1" xmlns:ns2="http://gpe.cz/pay/pay-
ws/core/type" xmlns:ns3="http://gpe.cz/pay/pay-ws/proc/v1/type">
         <ns4:paymentStatusResponse>
            <ns3:messageId>20191127174308776</ns3:messageId>
            <ns3:state>1</ns3:state>
            <ns3:status>UNPAID</ns3:status>
            <ns3:subStatus>INITIATED</ns3:subStatus>

<ns3:signature>M49XARQA7zu9xPHzaTgVUbFjpgXSC7OOH+BiN46mVldTrfqJYoSoQ9yAmcVrwF9Czv82ubHpY5+Q14MeQLMc0
iK7vohROjZscHHnMDUrpqj3l5WAaU/LEN/c0SR6dJDqHlbD1wvW557dTrTh3yjZWHT/bAqCqNU5S9rGVGNjB4kPcnxEPUsrApBuM
Gb+/Nugyf9VMUTdWeEbfuvgYfa//7fRUwABa5bhryiJW+1dvslR9MFaMYgqfhLeGDr+q8SaRIfE4qd/4XGKKF8Aa1xlhWldPVCKV
4pfag0/RcimaQ7IBb2IYV2rdtYyKhRwPj0WSU4OnkDzkU6MnauD3iRySA==</ns3:signature>
         </ns4:paymentStatusResponse>
      </ns4:getPaymentStatusResponse>
   </soapenv:Body>
</soapenv:Envelope>
```

Data pro podpis: **20191127174308776|1|UNPAID|INITIATED**

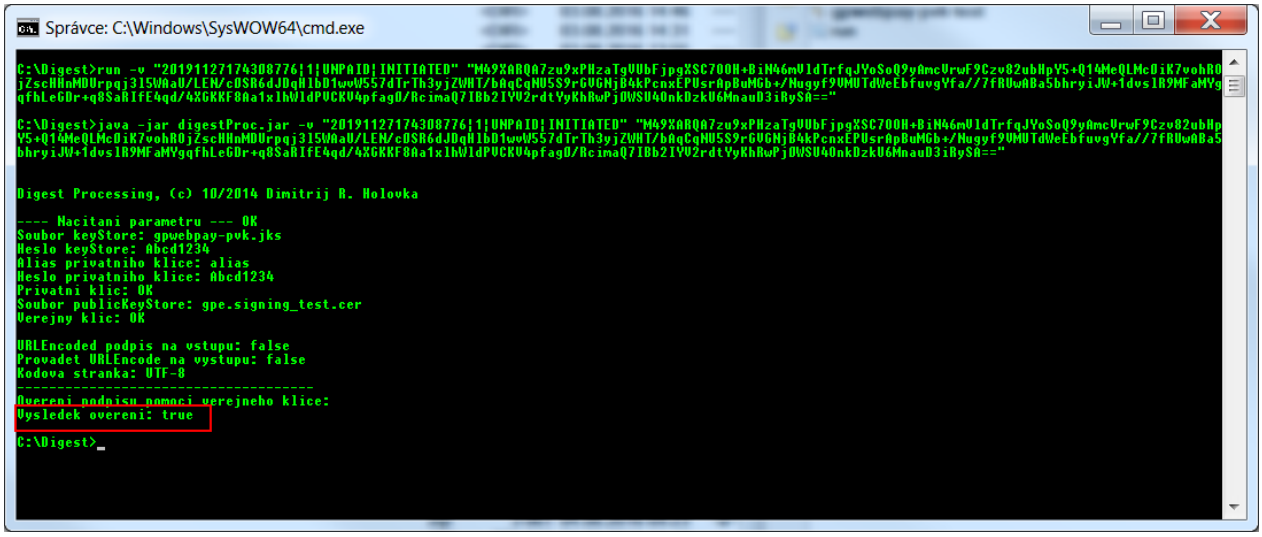
Podpis:

M49XARQA7zu9xPHzaTgVUbFjpgXSC7OOH+BiN46mVldTrfqJYoSoQ9yAmcVrwF9Czv82ubHpY5+Q14MeQLMc0iK7vohROjZscHHnMDUrpqj3l5WAaU/LEN/c0SR6dJDqHlbD1wvW557dTrTh3yjZWHT/bAqCqNU5S9rGVGNjB4kPcnxEPUsrApBuMGb+/Nugyf9VMUTdWeEbfuvgYfa//7fRUwABa5bhryiJW+1dvslR9MFaMYgqfhLeGDr+q8SaRIfE4qd/4XGKKF8Aa1xlhWldPVCKV4pfag0/RcimaQ7IBb2IYV2rdtYyKhRwPj0WSU4OnkDzkU6MnauD3iRySA==

Příkaz pro spuštění aplikace:

```
java -jar digestProc.jar -v "20191127174308776|1|UNPAID|INITIATED"
"M49XARQA7zu9xPHzaTgVUbFjpgXSC7OOH+BiN46mVldTrfqJYoSoQ9yAmcVrwF9Cz
v82ubHpY5+Q14MeQLMc0iK7vohROjZscHHnMDUrpqj3l5WAaU/LEN/c0SR6dJDqHlb
D1wvW557dTrTh3yjZWHT/bAqCqNU5S9rGVGNjB4kPcnxEPUsrApBuMGb+/Nugyf9VM
UTdWeEbfuvgYfa//7fRUwABa5bhryiJW+1dvslR9MFaMYgqfhLeGDr+q8SaRIfE4qd
/4XGKKF8Aa1xlhWldPVCKV4pfag0/RcimaQ7IBb2IYV2rdtYyKhRwPj0WSU4OnkDzk
U6MnauD3iRySA=="
```

Výsledek je zobrazen na obrazovce:


Správce: C:\Windows\SysWOW64\cmd.exe
C:\Digest>run -v "2019112717430877611UNPAID1INITIATED" "M49XARQA7zu9xPNzaTgVUbFjpgXSC7OOH+BiN46mVldTrfqJYoSoQ9yAmcVrwF9Czv82ubHpY5+Q14MeQLMcOiK7vohROjZscHHnMDUrpqj315WAaU/LEN/cOSR6dJDqN1bD1wvV557dTrTh3yjZWHT/bAqCqMU5S9rGVGMjB4kPcnxEPUsrApBuMGb+/NugyF9VMUTdVeEbfuvgVfa//7fRUwABa5bhryiJV+1dvs1R9MFaMYqqfhLeGDr+q8SaRIfE4qd/4XGKKF8Aa1x1hWldPUCKV4pfagO/RcimaQ7IBb2IYV2rdtVyKhRwPjOWSU4OnkDzkU6MnauD3iRySA==
C:\Digest>java -jar digestProc.jar -v "2019112717430877611UNPAID1INITIATED" "M49XARQA7zu9xPNzaTgVUbFjpgXSC7OOH+BiN46mVldTrfqJYoSoQ9yAmcVrwF9Czv82ubHpY5+Q14MeQLMcOiK7vohROjZscHHnMDUrpqj315WAaU/LEN/cOSR6dJDqN1bD1wvV557dTrTh3yjZWHT/bAqCqMU5S9rGVGMjB4kPcnxEPUsrApBuMGb+/NugyF9VMUTdVeEbfuvgVfa//7fRUwABa5bhryiJV+1dvs1R9MFaMYqqfhLeGDr+q8SaRIfE4qd/4XGKKF8Aa1x1hWldPUCKV4pfagO/RcimaQ7IBb2IYV2rdtVyKhRwPjOWSU4OnkDzkU6MnauD3iRySA==
Digest Processing, (c) 10/2014 Dimitrij R. Holovka
---- Nacitani parametru --- OK
Soubor keyStore: gpwebpay-pvk.jks
Heslo keyStore: Abcd1234
Alias privatniho klice: alias
Heslo privatniho klice: Abcd1234
Privatni klic: OK
Soubor publicKeyStore: gpe.signing_test.cer
Verejny klic: OK
URLEncoded podpis na vstupu: false
Provadet URLEncode na vystupu: false
Kodova stranka: UTF-8
Overeni podpisu pomoci verejneho klice:
Vysledek overeni: true
C:\Digest>