ŠKODA AUTO VYSOKÁ ŠKOLA o.p.s.

Studijní program: Podniková ekonomika a manažerská informatika

Digitální podpisy a jejich implementace ve ŠKODA AUTO a.s.

Bakalářská práce

Štěpán HAVLAS

Vedoucí práce: Ing. Vladimír Beneš, Ph.D.

ŠKODA AUTO VYSOKÁ ŠKOLA o.p.s.
Katedra informatiky

Akademický rok: 2022/2023

ŠKODA AUTO Vysoká škola

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Zpracovatel: **Štěpán Havlas**

Studijní program: Podniková ekonomika a manažerská informatika

Název tématu: **Digitální podpisy a jejich implementace ve ŠKODA AUTO a.s.**

Cíl: Cílem závěrečné práce je podrobně popsat principy digitálního podpisu a s ním spojených digitálních certifikátů, zmapovat možná rizika používání certifikátů a digitálního podpisu. Pozornost bude věnována i kryptografii, která je s digitálními podpisy neslučitelně spojená.
V praktické části bude cílem popsat zpracování integrace Adobe Sign prostřednictvím API rozhraní do CMS (Contract management system) v rámci společnosti Škoda Auto.

Rámcový obsah:

1. 1 Teoretická část
   1.1 Úvod
2. 1.2 Papírové vs digitální podpisy
   1.3 Elektronický vs digitální podpis
3. 1.4 Symetrická kryptografie
   1.5 Asymetrická kryptografie
4. 1.6 Funkce šifrování
   1.7 Digitální podpis podrobně
5. 1.8 Infrastruktura veřejného klíče (PKI)
   1.9 Certifikáty
6. 1.10 Certifikační autorita (CA)
   1.11 Adresáře
7. 1.12 Seznamy odvolaných certifikátů
   1.13 Legislativa
8. 2 Praktická část
   Integrace Adobe Sign prostřednictvím API rozhraní do Contract management system (CMS)

Rozsah práce: 25 – 30 stran

Seznam odborné literatury:

1. KATZ, J. *Digital Signatures (Advances in Information Security).* USA: Springer, 2010. ISBN 978-03-872-7711-0.
2. VOHNOUTOVÁ, M. – DOSTÁLEK, L. *Velký průvodce infrastrukturou PKI a technologií elektronického podpisu.* ČR: Computer Press, 2010. ISBN 978-80-251-2619-6.
3. BALLAD, B. – BALLAD, T. – BANKS, E. *Access Control, Authentication, & Public Key Infrastructure.* USA: Jones & Bartlet, 2010. ISBN 978-07-637-9128-5.
4. SMART, Nigel. (2016) Cryptography Made Simple (Information Security and Cryptography). Springer, 2016, 1st ed. ISBN-13: 978-33-193-7309-6

Datum zadání bakalářské práce: prosinec 2021

Termín odevzdání bakalářské práce: prosinec 2022

L. S.

Elektronicky schváleno dne 23. 5. 2022
Štěpán Havlas
Autor práce

Elektronicky schváleno dne 23. 5. 2022
Ing. Vladimír Beneš, Ph.D.
Vedoucí práce

Elektronicky schváleno dne 24. 5. 2022
prof. Ing. Jiří Strouhal, Ph.D.
Garant studijního programu

Elektronicky schváleno dne 24. 5. 2022
doc. Ing. Pavel Mertlík, CSc.
Rektor ŠAVŠ

Prohlašuji, že jsem závěrečnou práci vypracoval samostatně a použité zdroje uvádím v seznamu literatury. Prohlašuji, že jsem se při vypracování řídil vnitřním předpisem ŠKODA AUTO VYSOKÉ ŠKOLY o.p.s. (dále jen ŠAVŠ) směrnicí Vypracování závěrečné práce.

Jsem si vědom, že se na tuto závěrečnou práci vztahuje zákon č. 121/2000 Sb., autorský zákon, že se jedná ve smyslu § 60 o školní dílo a že podle § 35 odst. 3 je ŠAVŠ oprávněna mou práci využít k výuce nebo k vlastní vnitřní potřebě. Souhlasím, aby moje práce byla zveřejněna podle § 47b zákona č. 111/1998 Sb., o vysokých školách.

Beru na vědomí, že ŠAVŠ má právo na uzavření licenční smlouvy k této práci za obvyklých podmínek. Užiji-li tuto práci, nebo poskytnu-li licenci k jejímu využití, mám povinnost o této skutečnosti informovat ŠAVŠ. V takovém případě má ŠAVŠ právo ode mne požadovat příspěvek na úhradu nákladů, které na vytvoření díla vynaložila, a to až do jejich skutečné výše.

V Mladé Boleslavi dne 5.12.2022

Obsah

Děkuji Ing. Vladimírovi Benešovi, Ph.D. za odborné vedení závěrečné práce a poskytování rad. Dále děkuji Bc. Filipu Štěrbovi, který mi pomohl při získávání informací v rámci ŠKODA AUTO.

Seznam použitých zkratek a symbolů

AES	Advanced Encryption Standard
API	Application programming interface
CA	Certifikační autorita
CRM	Customer relationship management
CRM	Customer relationship management
DES	Data Encryption Standard
eIDAS	electronic IDentification, Authentication and trust Services
HTTP	Hypertext Transfer Protocol
ISO	International Organization for Standardization
JSON	JavaScript Object Notation
MFA	Multi-factor authentication
PDF	Portable Document Format
PKI	Public key infrastructure
RA	Registrační autorita
REST	Representational state transfer
RSA	Rivest Shamir Adleman
SOAP	Simple Object Access Protocol
TLS	Transport Layer Security
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
USB	Universal Serial Bus
XML	Extensible Markup Language

Úvod

Digitalizace v současné době hraje čím dál větší roli v podnikové sféře. Podniky ve stále větší míře volí digitální podpisy z důvodu zvýšení efektivity a informační bezpečnosti. Systémy pro správu digitálních dokumentů mohou podniku pomoci ušetřit čas, peníze a prostor. Poskytují lepší zabezpečení spolu se snížením papírování. V této práci budou popsány základní principy, na kterých stojí digitální podpis, jako je například asymetrická kryptografie, dělení digitálních podpisů z pohledu legislativy ČR a infrastruktura PKI.

Hlavním cílem práce je popsat čtenáři implementaci služby Adobe Acrobat Sign, která využívá digitálního podpisu, pomocí technologie „Application Programming Interface“ aniž by disponoval hlubokými znalostmi v oblasti informačních technologií.

1 Historie podpisu

Podpis je ručně psaný symbol umístěný na dokument, kus papíru nebo jiný materiál, který umožňuje ověřit něčí identitu a souhlas s například nějakým závazkem. Může mít podobu personalizované kresby nebo jednoduché značky jako „X“ (Cambridge English Dictionary, nedatováno). Podpisy na písemných transakcích byly v židovských komunitách obvyklé přibližně od druhého století a mezi muslimy od Hegiry (tj. stěhování Mohameda a jeho následovníků do Medíny) v roce 622. V Evropě se podpisy datují od šestého století. Ale na tomto kontinentu se příliš nepoužívaly po dalších tisíc let, až do 16. a 17. století, kdy vzrůstalo vzdělání a gramotnost a bylo uzavíráno více písemných dohod. V Anglii byl stěžejní „Statute of Frauds“ z roku 1677, který stanovil, že smlouvy musí existovat písemně a musí být opatřeny podpisem. Podpisy se staly standardní formou potvrzování dohod – tato praxe byla přijata i v koloniální Americe (The history of the Signature, 2016).

Mezi 6. stoletím, kdy se poprvé objevily podpisy a 17. stoletím, kdy se podepisování běžně využívalo, používali Evropané k formalizaci smluv také různé zvyky. Voskové pečeti nesoucí vyraženou postavu byly běžné, zvláště mezi Francouzi, kteří tuto tradici přinesli do Anglie během Normanské invaze. V Bibli se také objevují pečeti a dodnes se používají v Číně, Japonsku a Koreji.

Jedním z oblíbených způsobů, jak vytvořit tyto otisky, bylo vtlačení pečetního kroužku do včelího vosku. Používaly se i samotné pečetní prsteny: král mohl například vyslat posla s ústním poselstvím cizí mocnosti a předat mu královský pečetní prsten, aby si příjemce zprávy byl jistý jejím původem (Palmer, Mangham, 2022).

Jiné, neformálnější dohody byly také uzavírány ve fyzické podobě. Věnování pramenu odstřižených vlasů někomu – to byl také jeden ze způsobů, jak uzavřít smlouvu. Kolem 13. století byly dohody někdy stvrzeny fackou, nebo jiným traumatickým činem. Teorie byla taková, že obě strany si budou pamatovat nejen zranění, ale i shodu, které bylo dosaženo při jeho způsobení.

Ačkoliv se zdá, že jedinečnost podpisu spolu s jeho čitelností by měly být základními rysy podpisu, ve skutečnosti tomu tak nikdy nebylo. Počínaje 9. a 10. století písaři ověřovali listiny pomocí znamení kříže. Tento způsob potvrzování byl způsoben nízkou gramotností. Naproti tomu u gramotných šlechticů nebyl podpis určen

k tomu, aby byl snadno rozeznatelný – spíše složitý a nečitelný podpis, než tištěný podpis, naznačoval vzdělání v ručním psaní (Hawkins, 2011).

Již dlouho se má za to, že věnovat velkou pozornost podpisům není zvláště dobrý způsob, jak odhalit, či vyloučit podvod. Například případ padělání peněžních poukázek v Anglii z roku 1772 závisel více na nevyzpytatelném chování údajného padělatele než na jemných rozdílech v rukopisu mezi jednotlivými dokumenty. Aby bylo možné rozpoznat krádež identity, moderní společnosti vydávající kreditní karty a banky věnují méně pozornosti podpisům svých zákazníků než jejich zvyklostem utrácet (The history of the Signature, 2016).

Zákony o náležitostech podpisu reagovaly na technologické změny za posledních několik století. Rozvoj tiskařského stroje – stejně jako v menším měřítku šílenství 60. a 70. let 19. století ve vlastnictví vlastního razítka s faksimile (velice přesné napodobení originálu) podpisu – donutil soudy v Británii i ve Spojených státech rozhodnout, zda pouhé razítko se jménem je považován za podpis.

Telegram také představoval podstatnou výzvu pro existující právo: V roce 1869 Nejvyšší soud New Hampshire v případu Howley V. Whipple prohlásil, že „nezáleží na tom, zda operátor píše ocelovým perem, nebo zda jeho perem je měděný drát dlouhý tisíce mil“. V prostředí dnešního internetu věci neustále „podepisujeme“, aniž bychom si to uvědomovali (Hawkins, 2011).

1.1 Historie digitálního podpisu

Historie digitálního podpisu se začala psát v roce 1976 – Whitfield Diffie a Martin Hellman přišli s myšlenkou asymetrického kryptosystému veřejného a soukromého klíče. Byli také první, kdo přišel s myšlenkou digitálního podpisu a pokusili se aplikovat teorii čísel. Jejich formulace používala sdílený tajný klíč. Ponechali však problém realizace jednosměrné funkce otevřený, protože obtížnost faktorizace (rozklad celého čísla na součin prvočísel) nebyla v té době dobře prostudována (Koestler, 2022).

Ron Rivest, Adi Shamir a Leonard Adleman z Massachusettského technologického institutu přišli v roce 1977 s revolučním řešením. Jedná se o specifický typ kryptografie s veřejným klíčem (Public key cryptography), který zdokonalil řešení Diffieho a Hellmana. Vyvinuli algoritmus, který zajišťuje bezpečné šifrování

a dešifrování zpráv mezi komunikujícími stranami. Na rozdíl od předchozích metod, kdy bylo zapotřebí k šifrování a dešifrování použít sdíleného tajného klíče. RSA poskytla metodu šifrování a dešifrování, aniž by obě strany potřebovaly sdílený tajný klíč.

Clifford Cocks, anglický matematik pracující pro britskou zpravodajskou agenturu „Government Communications Headquarters“, popsal ekvivalentní systém v interním dokumentu v roce 1973. Vzhledem k tehdejším drahým počítačům byl však považován převážně za kuriozitu a nebyl nikdy nasazen (Palmer, Mangham, 2022).

Hned po RSA začalo mnoho společností budovat řešení založená na tomto algoritmu, aby byly digitální podpisy snadno dostupné. První široce prodávaný softwarový balík, který obsahoval digitální podpis, byl Lotus Notes 1.0. V roce 1987 byla publikována první verze softwaru. Sheldon Laube, CIO společnosti Price Waterhouse, byl nadchnut tímto řešením natolik, že koupil 10 000 kopií. To bylo do té doby považováno za největší prodej jednoho softwarového produktu. V roce 1995 IBM koupila Lotus, aby získala technologii Notes (Koestler, 2022).

Na přelomu milénia schválil Kongres Spojených států amerických zákon „Elektronický podpis v globálním a národním obchodu“. Tímto aktem vláda stanovila, že digitální podpisy budou mít stejnou právní sílu jako tradiční podpis.

V roce 2008 norma ISO 32000 činí PDF standardním formátem a zahrnuje digitální podpisy svou nedílnou součástí. Od té doby je primárním nástrojem pro podepisování dokumentů elektronicky.

2 Digitální podpis

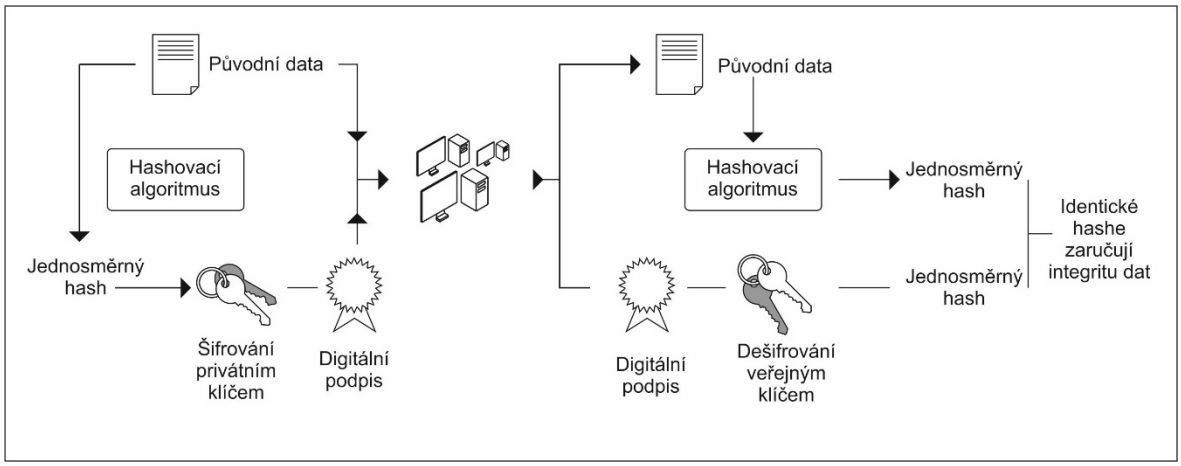
Digitální podpis je matematická technika používaná k ověření autentičnosti, taktéž integrity zprávy, softwaru nebo digitálního dokumentu. Jako digitální ekvivalent vlastnoručního podpisu nebo razítkové pečeti, digitální podpis nabízí mnohem větší zabezpečení a je určen k řešení problému předstírání identity (např. phishing) v digitální komunikaci (Katz, 2010).

Digitální podpisy navíc poskytují důkaz o původu, identitě a statusu elektronického dokumentu, transakce nebo zprávy a může potvrdit informovaný souhlas podepisujícího. V mnoha zemích, včetně České republiky, jsou digitální podpisy považovány za právně závazné stejně jako ručně psané podpisy. ČR se tak v roce 2000 stala 3. zemí na světě, kde vešel v platnost zákon upravující užívání elektronického podpisu.

Ze všech kryptografických principů je digitální podpis využívající kryptografii veřejného klíče považován za velmi důležitý a užitečný nástroj k dosažení informační bezpečnosti. Kromě schopnosti zajistit nepopiratelnost (autor podpisu nemůže tvrdit, že daný dokument nepodepsal) poskytuje digitální podpis také autentizaci zprávy a integritu dat (Lutkevich, 2021).

Vlastnosti digitálního podpisu:

- **Ověření zprávy:** když příjemce ověřuje digitální podpis pomocí veřejného klíče odesílatele, má jistotu, že podpis vytvořil pouze odesílatel, který vlastní odpovídající tajný soukromý klíč, a nikdo jiný.
- **Integrita dat:** v případě, že má útočník přístup k datům a upraví je, ověření digitálního podpisu na straně příjemce selže. Hash (hash je matematická funkce, která převádí vstup libovolné délky na výstup pevné délky) upravených dat a výstup poskytnutý ověřovacím algoritmem se nebudou shodovat. Příjemce tedy může bezpečně odmítnout zprávu za předpokladu, že byla narušena integrita dat.
- **Nepopiratelnost:** protože se předpokládá, že podpisový (soukromý) klíč zná pouze podepisující osoba, může vytvořit jedinečný podpis pouze na daných datech. Příjemce tak může poskytnout data a digitální podpis třetí straně jako důkaz, pokud v budoucnu dojde k nějakému sporu (Katz, 2010).



Zdroj: Vlastní zpracování

Obr. 1 Schéma digitálního podpisu

Obrázek 1 ukazuje dvě položky přenesené příjemci podepsané zprávy: původní zpráva a digitální podpis, což je hash původních dat zašifrovaných soukromým klíčem podepisujícího. K ověření integrity dat příslušný software nejprve použije veřejný klíč k dešifrování hashe. Poté použije stejný hashovací algoritmus, který vygeneroval původní hash, aby vygeneroval nový hash stejných dat. Informace o použitém hashovacím algoritmu jsou odeslány s digitálním podpisem. Nakonec software porovná nový hash s původním hashem. Pokud se oba hashe shodují, data se od podpisu nezměnila. Pokud se neshodují, data mohla být změněna od doby, kdy byla podepsána, nebo mohl být podpis vytvořen pomocí soukromého klíče, který nekoresponduje s veřejným klíčem, jenž prezentuje podepisující osoba.

2.1 Vlastnosti

Vlastnosti digitálního podpisu lze popsat následovně:

- algoritmus musí ověřit autora, datum a čas podpisu
- algoritmus musí ověřit obsah zprávy v době, kdy byl podpis vytvořen
- pro případ řešení sporů musí být podpis ověřitelný třetími stranami

2.2 Jak fungují digitální podpisy

Digitální podpisy jsou založeny na kryptografii veřejného klíče, známé také jako asymetrická kryptografie. Pomocí algoritmu veřejného klíče, jako je RSA,

lze vygenerovat dva klíče, které jsou matematicky propojeny: jeden soukromý a jeden veřejný.

Osoba, která vytváří digitální podpis, používá svůj vlastní soukromý klíč k šifrování dat souvisejících s podpisem; jediný způsob, jak tato data dešifrovat, je pomocí veřejného klíče podepisujícího. Takto se ověřují digitální podpisy.

Technologie digitálního podpisu vyžaduje, aby všechny strany důvěřovaly tomu, že jednotlivec vytvářející podpis byl schopen udržet svůj vlastní soukromý klíč v tajnosti. Pokud má někdo jiný přístup k soukromému klíči podepisujícího, může tato strana vytvořit podvodné digitální podpisy jménem držitele soukromého klíče (Lee, 2021).

2.3 Jak vytvořit digitální podpis

K vytvoření digitálního podpisu vytvoří příslušný software, například e-mailový program, jednosměrný hash elektronických dat, která mají být podepsána. Soukromý klíč se pak použije k zašifrování hashe. Zašifrovaný hash, spolu s dalšími informacemi, jako je hashovací algoritmus, je digitální podpis.

Důvodem zašifrování hashe místo celé zprávy, resp. dokumentu je, že hashovací funkce může převést libovolný vstup na hodnotu pevné délky, která je obvykle mnohem kratší (menší velikost). To šetří čas, protože hashování je mnohem rychlejší než podepisování. Hodnota hashe je pro hashovaná data jedinečná. Jakákoli změna v datech, dokonce i změna jednoho znaku, bude mít za následek jinou hodnotu. Tento atribut umožňuje ostatním ověřit integritu dat pomocí veřejného klíče podepisujícího k dešifrování hashe (Katz, 2010).

Pokud se dešifrovaný hash shoduje s druhým vypočítaným hashem stejných dat, dokazuje to, že data od doby, kdy byla podepsána, nebyla změněna. Pokud se dva hashe neshodují, data byla buď nějakým způsobem zmanipulována (narušena integrita) nebo byl podpis vytvořen pomocí soukromého klíče, který neodpovídá veřejnému klíči předloženému podepisovatelem (autentizace). Digitální podpis lze použít s jakýmkoli druhem zprávy (ať už je zašifrovaná nebo ne) jednoduše tak, aby si příjemce mohl být jistý identitou odesílatele a tím, že zpráva dorazila neporušená. Digitální podpisy znesnadňují podepisujícímu možnost popřít, že něco podepsal, a to za předpokladu, že jejich soukromý klíč nebyl kompromitován, protože digitální

podpis je jedinečný jak pro dokument, tak pro podepisujícího a spojuje je dohromady. Tato vlastnost se nazývá nepopiratelnost (Lutkevich, 2021).

Digitální podpisy nelze zaměňovat s digitálními certifikáty (bude blíže popsáno v následující kapitole). Digitální certifikát, elektronický dokument, který obsahuje digitální podpis vydávající certifikační autority, spojuje veřejný klíč s identitou a lze jej použít k ověření, že veřejný klíč patří určité osobě nebo subjektu. Většina moderních e-mailových programů podporuje používání digitálních podpisů a digitálních certifikátů, což usnadňuje podepisování všech odchozích e-mailů a ověřování digitálně podepsaných příchozích zpráv. Digitální podpisy jsou také široce používány k zajištění důkazu pravosti, integrity dat a nepopiratelnosti komunikace a transakcí prováděných elektronicky.

2.4 Úrovně digitálního podpisu

V současné době se právní úprava elektronického podpisu zakládá na nařízení Evropského parlamentu z roku 2014.

Nařízení pro elektronickou identifikaci, autentizaci a důvěryhodné služby, neboli eIDAS, bylo vytvořeno za účelem vytvoření důvěry v elektronické transakce mezi jednotlivci, organizacemi a vládními subjekty napříč Evropskými členskými státy. V rámci eIDAS mohou občané a podniky používat svá nativní národní schémata elektronické identifikace (eID) při přístupu k veřejným službám v jiných členských státech EU, které používají eID. Toto nařízení navíc zavádí standardy pro elektronické podpisy, časová razítka, elektronické pečetě a další autentizační prvky, včetně služeb elektronické certifikace a doporučeného doručování, které těmto elektronickým transakcím dávají stejný právní status, jako kdyby byly prováděny na papíře (Cryptomathic, 2018).

Toto unijní nařízení dělí elektronické podpisy (Hanák, Pruška, 2020) na následující typy:

2.4.1 Elektronický podpis (prostý)

Tento druh elektronického podpisu je, jak název napovídá, nejjednodušší a také nejvíce používaný. V tomto druhu podpisu se používá spoustu druhů technického provedení. Například pouhé napsání jména a příjmení na konec dokumentu nebo emailové zprávy. Dále například když se na internetové stránce zaškrtne pole

„souhlasím“, podpis vytvořený elektronickým perem, nebo naskenování podpisu na konec dokumentu. U tohoto druhu podpisu se lze setkat především u dvou soukromoprávních subjektů, avšak záleží především na tom, jak se tyto subjekty dohodnou. Mohou požadovat u právního jednání některý z pokročilejších typů elektronického podpisu.

2.4.2 Zaručený elektronický podpis

Takzvaný zaručený elektronický podpis už zajišťuje to, že nelze změnit podepisovaný dokument, aniž by byla poškozena integrita dokumentu (jak již bylo popsáno v předchozí kapitole). Avšak i tento typ podpisu má svou nevýhodu a to tu, že nemusí být založen na kvalifikovaném certifikátu. To v praxi znamená, že takovýto typ podpisu si může vytvořit prakticky kdokoliv. Orgány veřejné moci z tohoto důvodu neuznávají elektronický podpis, který není založen na certifikátu vydaném kvalifikovanou certifikační autoritou.

2.4.3 Uznávaný elektronický podpis

Tento typ elektronického podpisu disponuje stejnými náležitostmi jako předchozí typ podpisu s jedním velkým rozdílem. A to ten, že musí být založen na kvalifikovaném certifikátu, který nelze vydat, aniž by nebyla ověřena totožnost žadatele o certifikát. Pro jednání s orgány veřejné moci je až tento typ podpisu akceptován.

2.4.4 Kvalifikovaný podpis

V současnosti je kvalifikovaný elektronický podpis nejvyšší formou elektronického podpisu. I v tomto případě dojde k ověření identity žadatele o certifikát s tím rozdílem, že část certifikátu je uložena buď na čipovou kartu (v případě ŠKODA AUTO se jedná o MFA průkaz), nebo na USB token.

K podepsání dokumentu tímto podpisem je zapotřebí právě čipová karta nebo USB token, který je taktéž vydán certifikační autoritou. Tímto řešením je zajištěna nejvyšší možná míra zabezpečení. Jedině tento druh podpisu je v rámci nařízení eIDAS uznáván ve všech zemích Evropské unie.

2.5 Výhody Digitální ho podpisu

Níže jsou uvedeny hlavní výhody používání digitálních podpisů:

Rychlost: Smlouvy lze snadno sepsat, vyplnit a podepsat všemi zúčastněnými stranami v krátkém čase bez ohledu na to, jak daleko jsou strany geograficky.

Náklady: Používání poštovních nebo jiných služeb pro papírové dokumenty je mnohem dražší ve srovnání s používáním digitálních podpisů na elektronických dokumentech.

Spolehlivost: V případě ztráty nebo poškození dokumentu v papírové podobě při přepravě je třeba tento dokument znovu vytisknout a podepsat. V případě digitálně podepsaného dokumentu se ztratí pouze kopie.

Zabezpečení: Používání digitálních podpisů a elektronických dokumentů snižuje riziko zachycení, přečtení, zničení nebo pozměnění dokumentů během přepravy.

Autenticita: Elektronický dokument podepsaný digitálním podpisem může u soudu obstát stejně dobře jako jakýkoli jiný podepsaný papírový dokument.

Sledování: Digitálně podepsaný dokument lze snadno sledovat a lokalizovat v krátkém čase.

Nepopiratelnost: Podepsáním elektronického dokumentu je podepisující jednoznačně identifikován jako signatář, a to nelze později popřít.

Prevence podvodů: Nikdo jiný nemůže zfalšovat váš digitální podpis ani odeslat elektronický dokument s tvrzením, že jste jej podepsali vy.

Časové razítko: Díky časovému razítku u digitálních podpisů budete jasně vědět, kdy byl dokument podepsán.

2.6 Nevýhody Digitálního podpisu

Stejně jako všechny ostatní elektronické produkty mají digitální podpisy nevýhody, které s nimi souvisí.

Tyto zápory zahrnují:

Vypršení platnosti: Digitální podpisy, stejně jako všechny technologické produkty, jsou vysoce závislé na technologii (kryptografii), na které jsou založeny. V době rychlého technologického pokroku má mnoho z těchto technologických produktů krátkou životnost.

Certifikáty: Aby bylo možné efektivně využívat digitální podpisy, musí si odesílatel i příjemci zakoupit digitální certifikáty od důvěryhodných certifikačních autorit.

Zákon: V některých státech a zemích jsou zákony týkající se kybernetických a technologických problémů slabé nebo dokonce neexistují. Obchodování v takových jurisdikcích se stává velmi riskantním pro ty, kdo používají digitálně podepsané elektronické dokumenty.

Kompatibilita: Existuje mnoho různých standardů digitálního podpisu a většina z nich je vzájemně nekompatibilních, což komplikuje sdílení digitálně podepsaných dokumentů.

3 Kryptografie v souvislosti s Digitálním podpisem

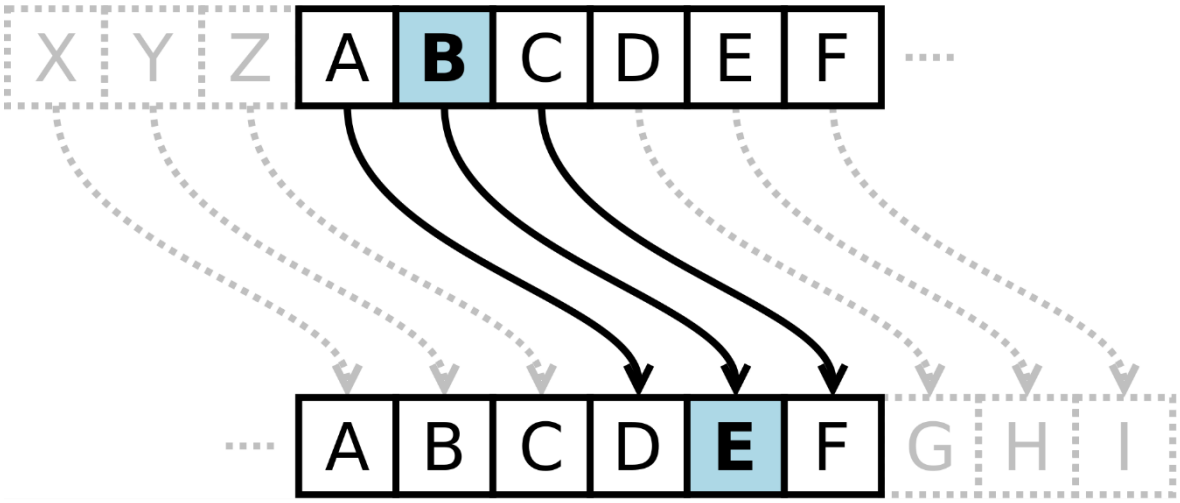
3.1 Úvod

Ačkoli se kryptografie používala po tisíce let ke skrývání tajných zpráv, její zkoumání jako vědy bylo započato teprve před zhruba sto lety.

První známý důkaz o používání kryptografie (v určité formě) byl nalezen v nápisu vytesaném kolem roku 1900 př. n. l. v hlavní komoře hrobky šlechtice Chnumhotepa [1]II. v Egyptě. Chnumhotepův písař tu nakreslil život svého pána na zdi jeho hrobky. Když kreslil hieroglyfy, používal řadu neobvyklých symbolů, aby zakryl význam nápisů. Tato metoda šifrování je příkladem substituční šifry. V substituční šifře je každý znak prostého textu (prostý text je zpráva, která musí být zašifrována) nahrazen jiným znakem, aby se vytvořil šifrovaný text (text šifry je zašifrovaná zpráva). Důkazy určitého použití kryptografie lze nalézt ve většině hlavních raných civilizací (Fleming, 2022).

V době kolem roku 100 př. n. l. Julius Caesar byl známý tím, že používal určitou formu šifrování k předávání tajných zpráv svým armádním generálům umístěným na válečné frontě (viz obr. 2). Byla využita substituční šifra, známá jako Caesarova šifra. Ta je pravděpodobně nejvíce zmiňovanou historickou šifrou v akademické literatuře. Podle Loshina (nedatováno) je šifra v kryptografii algoritmem pro šifrování a dešifrování dat. Varianta, kterou Caesar používal, byl každý znak posunut o 3 místa v abecedě, takže znak „A“ byl nahrazen „D“, „B“ nahrazeno „E“ a tak dále. Znaky na konci abecedy pokračovaly plynule na začátek, takže „X“ by bylo nahrazeno „A“ (Kotas, 2022).

[1] Chnumhotep II. byl šlechtic, úředník a kněz ve starověkém Egyptě za vlády faraonů Amenehmeta II. a Senusreta II.

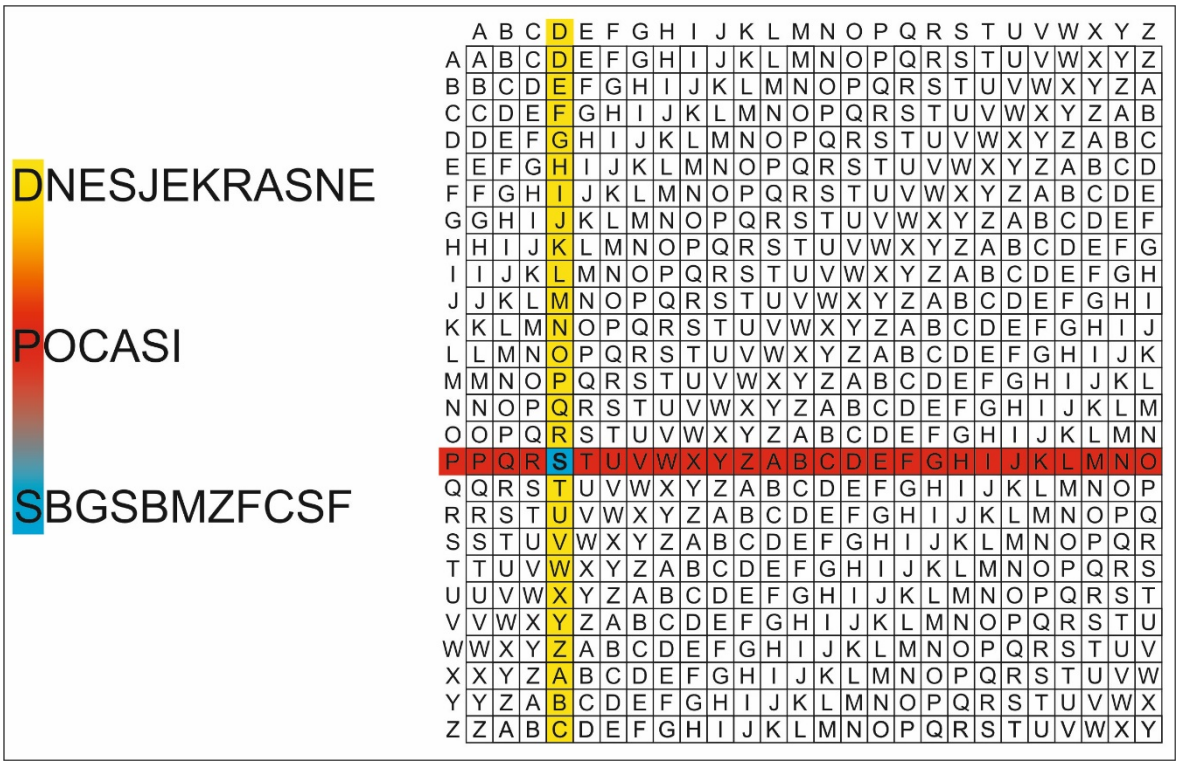


Zdroj: (Teachen, 2022)

Obr. 2 Caesarova šifra

Je snadné vidět, že takové šifry závisí na utajení šifrovacího systému, a ne na šifrovacím klíči. Jakmile je systém znám, lze tyto šifrované zprávy snadno dešifrovat. Ve skutečnosti lze substituční šifry prolomit pomocí frekvence písmen v jazyce.

Během 16. století Vigenere navrhl šifru, která byla první šifrou, jenž používala šifrovací klíč. Po téměř tři staletí byla jeho šifra považována jako „le chiffre indéchiffrable“, což doslova znamená neprolomitelná šifra. Tato šifra používá tabulku o rozměrech 26x26 známou jako Vigenerova tabulka (viz obr. 3). První řádek této tabulky obsahuje písmena abecedy od A do Z. v každém dalším řádku se znaky abecedy posunou o jeden znak, takže když například B se posune na začátek, A se posune na konec. Dále je zapotřebí zmíněný šifrovací klíč, jenž se opakuje, aby byl stejně dlouhý jako šifrovaný text. K zašifrování textu se vybere příslušný znak z šifrovaného textu a šifrovacího klíče jako index řádku a index sloupce v tabulce. Průsečík těchto dvou indexů je zašifrovaný znak. Tento proces je opakován až je zašifrovaný celý text. K dešifrování se vybere příslušný řádek v tabulce podle klíče a k němu nalezneme sloupec pomocí zašifrovaného znaku, index tohoto sloupce je již dešifrovaný znak. Stejně jako u Caesarovy šifry lze i Vigenerovu šifru v dnešní době snadno prolomit. Vigenere však jako první přišel s myšlenkou zavedení šifrovacích klíčů. Ve srovnání s Caesarovou šifrou závisí utajení zprávy spíše na utajení šifrovacího klíče než na utajení šifrovacího systému (Smart, 2016).



Zdroj: Vlastní zpracování

Obr. 3 Vigenerova šifra

Na začátku 19. století, kdy docházelo k masivní elektrifikaci prakticky všeho, Hebern navrhl elektromechanické zařízení, které se nazývalo Hebernův rotorový stroj. Využívá rotor, ve kterém je tajný klíč zapuštěn do rotujícího disku. Klíč kóduje substituční tabulku a každé stisknutí klávesy z klávesnice vedlo k výstupu šifrovaného textu. To také otočilo disk o jeden zářez a pro další znak textu by se pak využila jiná tabulka. I toto řešení bylo prolomeno pomocí frekvenční analýzy (Simmons, 2009).

Stroj Enigma byl vynalezen německým inženýrem Arthurem Scherbiem na konci první světové války a během 2. světové války byl hojně využíván německými silami. Stroj Enigma používal 3 nebo 4 nebo dokonce více rotorů. Rotory se při psaní na klávesnici otáčejí různými rychlostmi a zobrazují příslušná písmena šifrovaného textu. V tomto případě bylo klíčové počáteční nastavení rotorů. Šifru stroje Enigma se nakonec spojencům podařilo prolomit (Kotas, 2022).

Až do druhé světové války byla většina práce na kryptografii pro vojenské účely, obvykle používaná ke skrytí tajných vojenských informací. Kryptografie však přitáhla

pozornost komerční sféry, kdy se podniky snažily zabezpečit svá data před konkurencí.

Na začátku 70. let si společnost IBM uvědomila, že jejich zákazníci požadují nějakou formu šifrování, a tak vytvořili „kryptoskupinu“ v čele s Horst-Feistelem. Navrhli šifru Lucifer. Tato šifra nakonec vyústila v Data Encription Standard (DES), který Spojené státy přijaly jako národní standard pro šifrování dat ve státních organizacích a dále pak i v soukromém sektoru až do prolomení této šifry v roce 1997. Hlavním problémem DES byla malá velikost šifrovacího klíče. Jak se výpočetní výkon zvyšoval, bylo snadné hrubou silou použít všechny různé kombinace klíče k získání zašifrované zprávy. Později byl tento standard zdokonalen a přijat Americkým úřadem pro standardizaci v roce 2001 (Fleming, 2022).

Dalším důležitým milníkem v historii kryptografie bylo navržení RSA algoritmu, již zmíněném v úvodní kapitole.

3.2 Symetrická kryptografie

V dnešní době se kryptografie dělí na symetrickou a asymetrickou.

Symetrická kryptografie používá tajný klíč pro šifrování i dešifrování. Tento přístup je opakem asymetrického šifrování, které používá jeden klíč k šifrování a jiný k dešifrování. Data jsou přeložena do formátu, který nemůže být interpretován nebo kontrolován někým, kdo nemá tajný klíč použitý k jejich zašifrování během této fáze (Smart, 2016).

Účinnost této metody určuje síla generátoru náhodných čísel použitého k vygenerování tajného klíče. Symetrická kryptografie, dnes běžně používaná na internetu, zahrnuje dva druhy algoritmů: bloková šifra, kde šifrovaný text je šifrován po blocích a šifra proudová, u které je text šifrován po jednom bajtu. Advanced Encryption Standard (AES) je dnes běžný šifrovací algoritmus. Tento typ šifrování je obvykle mnohem rychlejší než asymetrické šifrování, ale umožňuje, aby tajný klíč držel odesílatel i příjemce dat.

Symetrická kryptografie je založena na jediném sdíleném klíči, kterého jsou si vědomy všechny strany a který mohou používat k šifrování a dešifrování dat. Přestože je symetrické šifrování starším typem šifrování, je jednodušší a efektivnější

než asymetrické šifrování, které je náročnější na výpočetní výkon (Chandra, Bhattacharyya, Paira, Alam, 2022).

Vzhledem k tomu, že symetrické šifrování je plynulejší a rychlejší než asymetrické šifrování, běžně se používá pro šifrování velkých objemů dat, jako je například šifrování databází.

3.3 Asymetrická kryptografie

Asymetrická kryptografie, lépe známá jako kryptografie s veřejným klíčem, šifruje a dešifruje zprávu pomocí dvojice podobných klíčů, veřejného a privátního. Efektivní zabezpečení vyžaduje, aby privátní klíč zůstal v rukách majitele; veřejný klíč může být otevřeně distribuován bez ohrožení bezpečnosti. Kdokoli může použít veřejný klíč k zašifrování dokumentu tak, aby jej pomocí svého soukromého klíče mohl dešifrovat pouze příjemce.

Když se kdokoli pokusí odeslat zašifrovanou zprávu, použije sdílený adresář k získání veřejného klíče příjemce a použije jej k zašifrování zprávy. Zprávu pak příjemce dešifruje pomocí jeho přidruženého soukromého klíče. Na asymetrické kryptografii závisí řada protokolů, včetně protokolu Transport Layer Security (TLS), který zajišťuje bezpečnou komunikaci na internetu. U digitálního podpisu hraje asymetrická kryptografie klíčovou roli (Ohri, 2022).

3.4 RSA algoritmus

RSA, pojmenovaný po jeho autorech Rivest, Shamir, Adleman, je šifrovací algoritmus, který se používá k bezpečnému přenosu zpráv přes internet. Jedná se o asymetrickou kryptografii. Je založen na principu, že je snadné násobit velká čísla, ale rozklad velkého čísla na součin prvočísel je velmi obtížné. Například je snadné zkontrolovat, že násobek 31 a 37 je 1147, ale rozložit 1147 na součin dvou prvočísel je mnohem náročnější proces (Simmons, 2009).

Nejprve se zvolí pár dvou prvočísel *p* a *q*. Například 2 a 7 (1). Dále je třeba určit *N*, což je součin těchto dvou čísel (14). Nyní bude spočítán *T* podle vzorce (1).

$$T = (p-1)\mathrm{x}\,(q-1) \tag{1}$$

$$T = (2-1)\text{x}\,(7-1) \tag{2}$$

Po dosazení (2) je získána hodnota 6. Dále je zapotřebí určit čísla *e* a *d*. Musí platit vzorec (3).

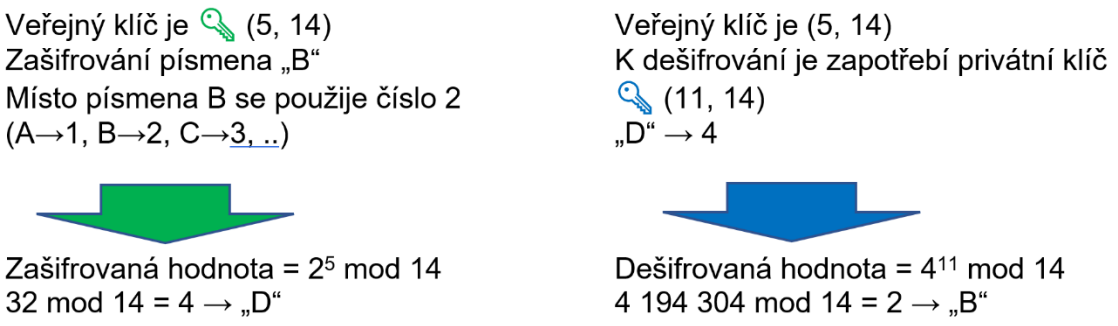
$$(e\ x\ d)\ mod\ T = 1 \tag{3}$$

(Modulo je matematická operace, jejímž výsledkem je zbytek po dělení jednoho čísla druhým). Tedy když součin těchto čísel je dělen 6 (T), musí být zbytek po dělení 1, *e* musí být menší než 6 a zároveň musí být nesoudělné s 6 (T) a 14 (N). v úvahu připadá pouze číslo 5. Z předchozí definice je patrné, že *d* může být například 11 (viz vzorec 4).

$$11 \rightarrow (5\ x\ 11)\ mod\ 6 = 1 \tag{4}$$

Nyní jsou pro RSA algoritmus všechny hodnoty spočítané a princip šifrování a dešifrování je popsán na obrázku 4.

V praxi se *p* a *q* volí tak velká čísla, aby rozklad jejich součinu na prvočísla nebyl dnešními výpočetními kapacitami spočítatelný. V současné době se za bezpečnou délku klíče považuje alespoň 2048 bitů dlouhý klíč. To znamená, že *p* a *q* musí být alespoň 308 cifer dlouhé číslo, takže jejich součin je přibližně 617 cifer dlouhé číslo. Do dnešního dne je délka nejdelšího prolomeného klíče 829 bitů.



Zdroj: Vlastní zpracování

Obr. 4 Princip RSA algoritmu

3.5 SHA algoritmus

SHA znamená bezpečný hashovací algoritmus (Secure Hashing Algorithm). SHA je upravená verze algoritmu MD5 a používá se pro hashování dat a certifikátů. Hashovací algoritmus vytváří ze vstupních dat výstup (hash) pevné délky. V případě dnes používaného algoritmu SHA-2 to může být až 512 bitů. Hashování je podobné šifrování, hlavní rozdíl mezi hashováním a šifrováním je v tom, že hashování je jednosměrná funkce, což znamená, že jakmile jsou data hashovaná, je téměř nemožné z výsledného hasne rekonstruovat původní data. Když se v hashovaném textu změní pouze jediný znak, výsledný hash je zcela jiný (viz obr. 5). Toho algoritmu se využívá právě v digitálním podpisu při ověřování integrity (Landman, 2022).

Zdroj: (sha1-online, 2022)

Obr. 5 Výsledek hashovacího algoritmu

4 Infrastruktura PKI

PKI (infrastruktura veřejného klíče) je základní rámec, který umožňuje uživatelům a serverům bezpečně vyměňovat informace pomocí digitálních certifikátů. Subjekty, které umožnují a používají PKI, obvykle zahrnují obecné uživatele internetu, webové klienty nebo prohlížeče a firemní servery (Vohnoutová, Dostálek, 2010).

Slovo infrastruktura je použito záměrně, protože PKI se nevztahuje na jedinou fyzickou entitu. Místo toho odkazuje na komponenty používané k šifrování dat a ověřování digitálních certifikátů. Tyto komponenty zahrnují hardware, software, zásady, procedury a entity potřebné k bezpečné distribuci, ověřování a revokaci certifikátů.

PKI dvouklíčový asymetrický kryptosystém, který podporuje různé systémy informačních technologií ve snaze o vysokou úroveň důvěrnosti informací, šifrování a důvěryhodnosti. Dva klíče jsou v tomto případě také dvěma hlavními částmi, které usnadňují tuto bezpečnou správu dat: veřejný klíč a soukromý klíč (Fruhlinger, 2022).

Certifikát je informace odkazující na veřejný klíč, který byl digitálně podepsán certifikační autoritou. Informace, které se běžně nacházejí v certifikátu, odpovídají standardu X.509. Certifikáty vyhovující tomuto standardu obsahují informace o zveřejněné identitě vlastníka odpovídajícího soukromého klíče, délce klíče, použitém algoritmu a souvisejícím hashovacím algoritmu, datech platnosti certifikátu a akcích, pro které lze klíč použít. Do určité míry je možné certifikát připodobnit občanskému průkazu (viz tab. 1). Certifikát není nezbytný pro provozování PKI, nicméně určité schéma je nezbytné k nalezení informací o držiteli soukromého klíče (Vohnoutová, Dostálek, 2010).

Veřejné klíče jsou uloženy v digitálních certifikátech spolu s dalšími relevantními informacemi (informace o uživateli, datum vypršení platnosti, použití, kdo certifikát vydal atd.). CA (Certifikační autorita) zadává informace obsažené v certifikátu při jeho vydání a tyto informace nelze změnit. Vzhledem k tomu, že certifikát je digitálně podepsán a všechny informace v něm obsažené mají být veřejně dostupné, není třeba bránit v přístupu k jeho čtení (Ionos, 2022).

Položka certifikátu	Položka občanského průkazu
Verze certifikátu	Verze formátu občanského průkazu
Pořadové číslo	Číslo občanského průkazu
Algoritmus podpisu	Typy ochranných prvků
Vydavatel	Vydal
Platnost	Platnost
Předmět: jméno, adresa, …	Jméno a adresa
Veřejný klíč	-
-	Fotografie
Rozšíření certifikátu	Nepovinné údaje
Elektronický podpis	Ochranné prvky

Zdroj: (Dostálek, 2020)

Tab. 1 Porovnání občanského průkazu a certifikátu

4.1 Komponenty PKI

Infrastruktura veřejného klíče je vytvořena kombinací řady služeb a technologií:

4.1.1 Certifikační autorita

Certifikáty vydává a ověřuje CA. CA přebírá odpovědnost za správnost identifikace osoby žádající o vydání certifikátu a zajišťuje správnost informací obsažených v certifikátu a digitálně jej podepisuje.

- Generování klíče: CA může vygenerovat veřejný klíč a soukromý klíč nebo si osoba žádající o certifikát může vygenerovat svůj vlastní pár klíčů a poslat podepsanou žádost obsahující jejich veřejný klíč CA k ověření. Osoba žádající o certifikát může preferovat generování vlastního páru klíčů, aby bylo zajištěno, že soukromý klíč bude stále pod kontrolou pouze držitele certifikátu a v důsledku toho bude méně pravděpodobné, že bude dostupný komukoli jinému.
- Vydávání certifikátů: Pokud není vygenerován vlastní certifikát (některé softwarové aplikace to umožňují), bude nutné si jej zakoupit, obvykle od CA. Než CA vydá certifikát, provede různé kontroly, aby si ověřila, že jste tím, za koho se vydáváte. Vydání certifikátu se do určité míry dá přirovnat k vydání pasu, kde je také nutné prokázat totožnost žadatele. Potom co si CA ověří identitu žadatele je certifikát podepsán, aby bylo zabráněno změnám v certifikátu.

- Jednotlivec může mít libovolný počet certifikátů vydaných libovolným počtem CA. Různé webové aplikace mohou vyžadovat, aby byly použity certifikáty vydané pouze určitými CA. Například banka může trvat na tom, aby byl použit certifikát, který vydala, zatímco veřejná webová stránka může přijmout jakýkoli certifikát, který jí je předložen. CA může být entita v rámci organizace (například koncern VW), nebo nezávislý subjekt. V ČR například První certifikační autorita, Česká pošta nebo eIdentity.

4.1.2 Registrační autorita

CA může využít třetí stranu – Registrační autoritu (RA) – k provedení nezbytných kontrol osoby nebo společnosti požadující certifikát, aby se ujistila, že jsou tím, za koho se vydávají. Tato RA se může žadateli o certifikát jevit jako CA, ale ve skutečnosti nepodepisuje vydaný certifikát.

4.1.3 Odvolání certifikátu

Tam, kde se systém spoléhá na publikování certifikátů, aby lidé mohli mezi sebou komunikovat, musí existovat systém, který uživatelům dá vědět, když se certifikáty stanou neplatnými. K tomuto účelu slouží seznam zneplatněných certifikátů (certificate revocation list). Je to list certifikátů, které byli odvolány CA před plánovaným datem vypršení platnosti a nejsou nadále důvěryhodné. Tyto seznamy jsou v pravidelných intervalech aktualizovány. K odvolání certifikátu může dojít z různých důvodů.

Například:

- došlo k úniku privátního klíče uživatele
- uživatel, pro kterého byl certifikát vydán, opustil organizaci (např. byl propuštěn)
- informace o uživateli byly změněny (např. změna jména)
- nahrazení certifikátu jiným

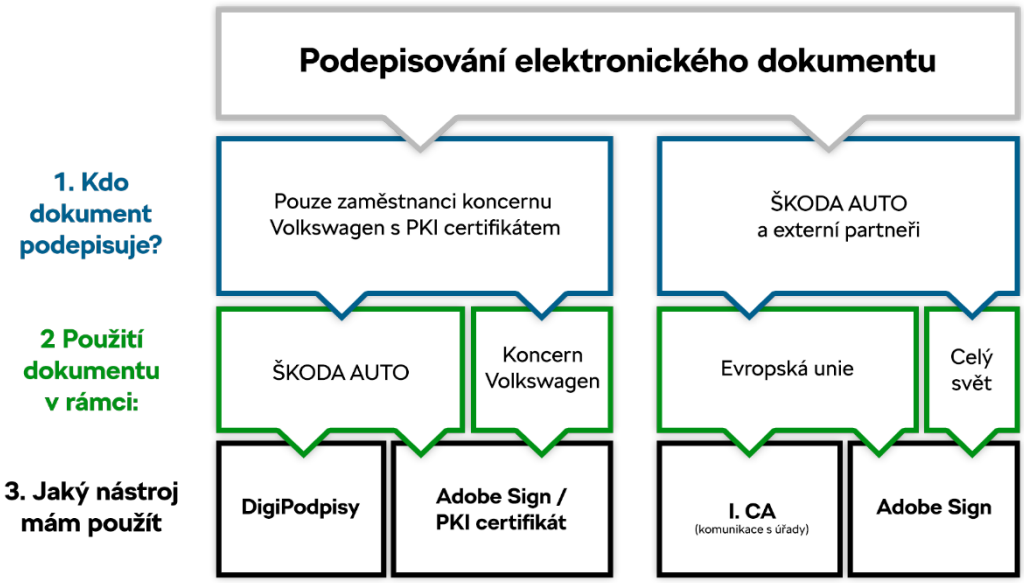
4.1.4 Systém správy certifikátů (Certificate Management System)

Tento termín označuje systém řízení, jehož prostřednictvím jsou certifikáty zveřejňovány, dočasně nebo trvale pozastaveny, obnovovány nebo rušeny. Systémy správy certifikátů certifikáty běžně neodstraňují, protože může být

v budoucnu nutné v určitém okamžiku prokázat jejich stav, možná z právních důvodů.

5 Implementace digitálního podpisu

V praxi existuje celá řada druhů implementace digitálního podpisu. Tato práce se zabývá implementací služby Adobe Acrobat Sign skrze rozhraní Application programming interface. Ve ŠKODA AUTO a.s. je podepisování elektronického dokumentu rozděleno do několika úrovní (viz obr. 6).



Zdroj: (Zaměstnanecký portál ŠKODA space)

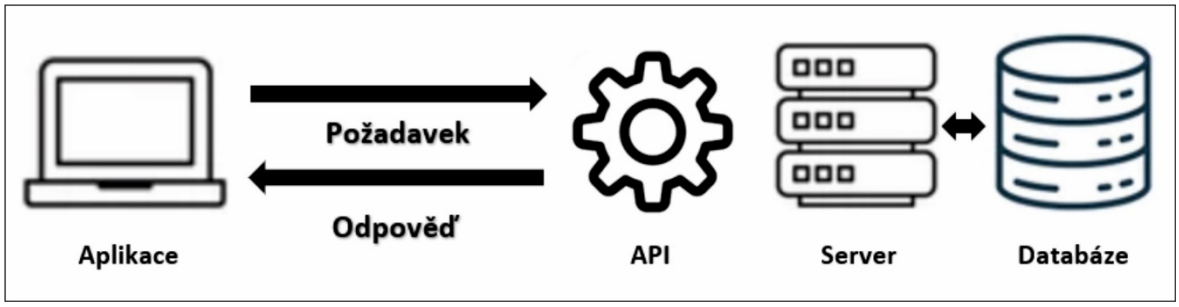
Obr. 6 Digitální podpisy ve ŠKODA AUTO a.s.

5.1 Application Programming Interface

API je software, který zajišťuje komunikaci a výměnu dat mezi dvěma aplikacemi. Toto rozhraní umožňuje společnostem zpřístupnit data a funkce svých aplikací vývojářům třetích stran. Anebo oddělením v rámci jejich firmy. Službám a produktům toto umožňuje komunikovat mezi sebou a vzájemně využívat svá data a funkce prostřednictvím různých rozhraní. Vývojáři nepotřebují vědět jak je API implementováno a jednoduše používají toto rozhraní ke komunikaci s jinými službami. Používání API za poslední desetiletí vzrostlo až do té míry, že mnoho webových aplikací by dnes bez tohoto rozhraní nemohlo fungovat.

5.1.1 Jak funguje API?

API je sada definovaných pravidel, která vysvětlují, jak spolu počítače nebo aplikace komunikují. Rozhraní API jsou umístěna mezi aplikací a webovým serverem a fungují jako zprostředkovatelská vrstva, která zpracovává přenos dat mezi systémy (viz obr. 7).



Zdroj: Vlastní zpracování

Obr. 7 Princip fungování API

Základní funkcionality API:

- Klientská aplikace vyšle požadavek, takzvané „volání API“, za účelem získání informací. Tento požadavek je zpracován z aplikace na webový server
- Po obdržení platného požadavku, API kontaktuje externí program nebo server
- Server odešle odpověď API s požadovanými informacemi
- Rozhraní API přenese data do původní aplikace, která vyslala požadavek

I když se přenos dat bude lišit v závislosti na používané webové službě, tento proces požadavků a odpovědí probíhá prostřednictvím rozhraní API. Zatímco grafické uživatelské rozhraní poskytuje uživatelům přístup k datům a funkcím aplikace, API poskytuje přístup k počítačům nebo jiným aplikacím.

5.1.2 Proč použít API?

Mezi hlavní benefity patří například:

- **Vylepšená konektivita napříč aplikacemi**
  V posledních letech v podnikové sféře strmě narůstá počet cloudových aplikací. Rozhraní API umožňují integraci, takže tyto platformy a aplikace

mohou mezi sebou bezproblémově komunikovat. Díky této integraci mohou společnosti automatizovat pracovní postupy a zlepšit spolupráci na pracovišti.

- **Jednodušší inovace**
  API nabízejí flexibilitu, která společnostem umožňuje navazovat spojení s novými obchodními partnery, nabízet nové služby na jejich stávajícím trhu a v konečném důsledku přistupovat na nové trhy, které mohou generovat masivní výnosy a řídit digitální transformaci.

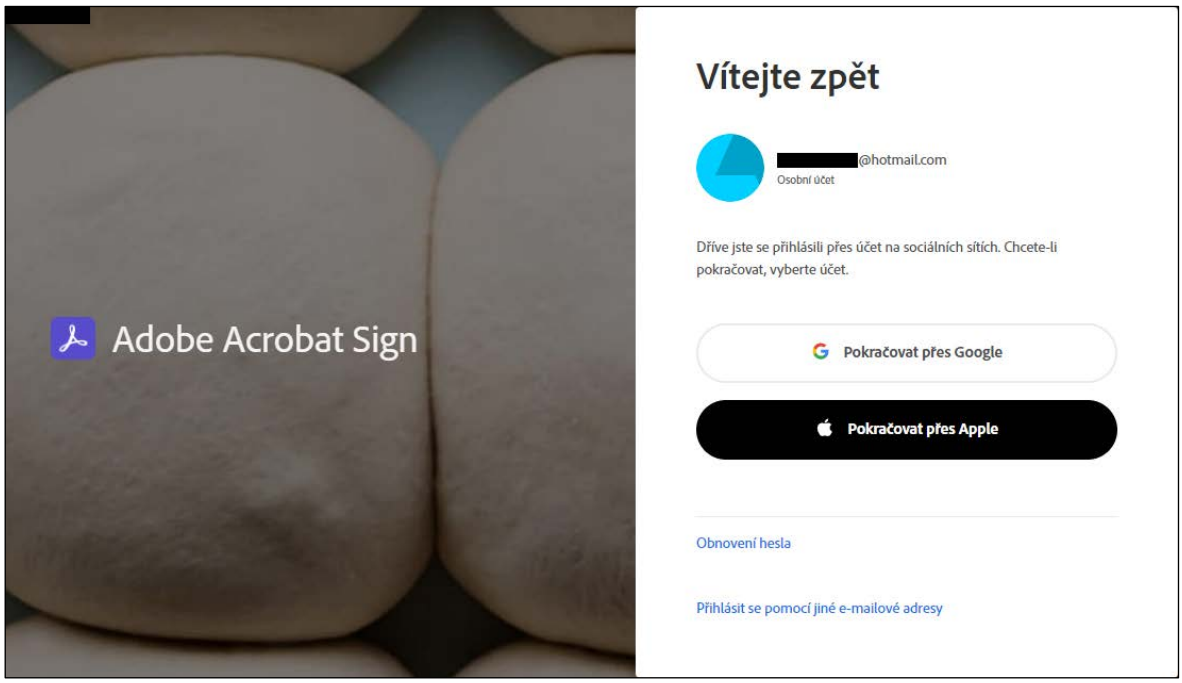
- **Přidaná vrstva zabezpečení**
  API rozhraní vytvářejí další vrstvu ochrany mezi podnikovými daty a serverem. Vývojáři mohou dále posílit zabezpečení pomocí autentizačních tokenů a šifrování protokolem Transport Layer Security (TLS).

5.1.3 Příklady využití API

Protože API umožňují společnostem otevřít přístup ke svým zdrojům při zachování bezpečnosti a kontroly, staly se cenným aspektem dnešních služeb a aplikací.

- **Univerzální přihlašování**
  Populárním příkladem API je funkce, která uživatelům umožňuje přihlašovat se na webové stránky pomocí účtů sociálních sítí. Například Google nebo Facebook. Tato funkcionalita umožňuje libovolné webové stránce využít API například Googlu (viz obr. 8) k rychlé autentizaci uživatele, což šetří čas a námahu s nastavováním nového profilu pro každou webovou službu. Toto autentizační schéma je také známe pod názvem „Single sign-on“.

- **Srovnání letů nebo ubytování**
  Služby pro rezervace letů nebo ubytování shromažďují tisíce letů a prezentují uživateli nejlevnější možnosti pro dané datum a destinaci. Tato služba je umožněna prostřednictvím API, která například službě Kiwi poskytuje přístup k nejnovějším informacím o dostupnosti letů daných leteckých společností. Díky automatizované výměně dat a požadavků API dramaticky zkracují čas a úsilí spojené s kontrolou dostupných letů nebo ubytování.



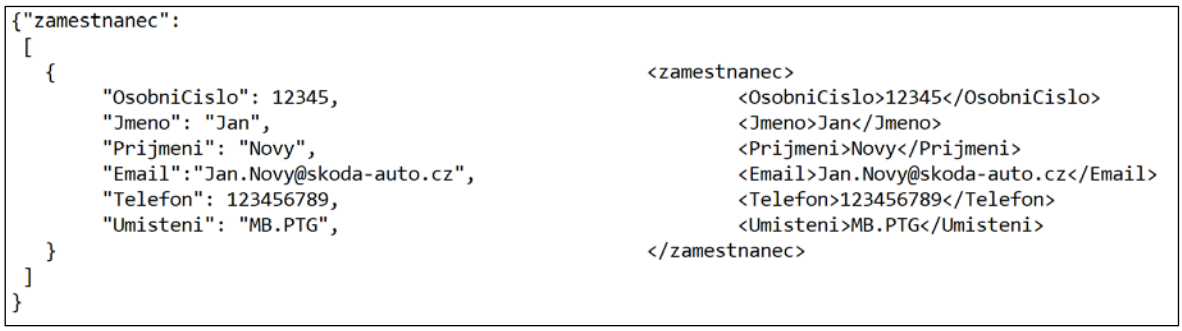
Zdroj: Vlastní zpracování

Obr. 8 Přihlašování pomocí účtu sociálních sítí

5.1.4 Typy API

S rostoucím využíváním webových API rozhraní byly vyvinuty protokoly, které uživatelům poskytují sadu definovaných pravidel, která specifikují přijímané datové typy a příkazy. Tyto protokoly API ve skutečnosti usnadňují standardizovanou výměnu informací.

- **SOAP** (Simple Object Access Protocol)
  Jedná se o protokol API postavený na XML (jazyk používaný pro výměnu dat mezi aplikacemi), který uživatelům umožňuje odesílat a přijímat data prostřednictvím protokolu HTTP (internetový protokol určený pro komunikaci s WWW servery).
- **JSON-RPC**
  Tento protokol namísto XML používá JSON (javaScript Open Notation). API volání mohou obsahovat více parametrů, avšak pouze jeden výsledek. Na obrázku 9 je srovnán XML spolu s JSON.



Zdroj: Vlastní zpracování

Obr. 9 Porovnání JSON a XML

- **REST** (Representational State Transfer)
  REST je sada architektonických omezení, nikoli protokol nebo standard. Data jsou nejčastěji přenášena ve formátu JSON. Tento druh API je využíván službou Adobe Acrobat Sign.

5.2 REST API

REST API je založeno na technologii „Representational State Transfer“ (REST), což je přístup ke komunikaci často používaný při vývoji webových služeb. Technologie REST je obecně preferována před jinými podobnými technologiemi. Především z důvodu že využívá menší datový tok. REST API lze také vyvíjet pomocí programovacích jazyků, jako je JavaScript nebo Python.

Technologie rest lze používat za tzv. „jazyk internetu“. S rostoucím využíváním cloudu jsou API využívána k organizaci přístupů k webovým službám. REST je logickou volbou pro vytváření rozhraní API, která uživatelům umožňují flexibilní připojení ke cloudovým službám, jejich správu a interakci s nimi v distribuovaném prostředí. REST API používají takové stránky jako Amazon, Google nebo Twitter.

Mezi základní principy REST patří:

- **Jednotné rozhraní**
  Je základem návrhu jakékoli webové služby REST. Indikuje, že server přenáší informace ve standardním formátu. Formátovaný zdroj se v REST nazývá reprezentace. Tento formát se může lišit od interní reprezentace zdroje na serverové aplikaci. Server může například ukládat data jako text, ale odesílat je ve formátu HTML.

- **Bezestavovost**

  V architektuře REST se bezstavovostí rozumí způsob komunikace, při kterém server dokončí každý požadavek klienta nezávisle na všech předchozích požadavcích. Klienti mohou žádat o zdroje v libovolném pořadí a každý požadavek je bezstavový nebo izolovaný od ostatních požadavků.

- **Vrstevnatost**

  Klient se může připojit k dalším autorizovaným zprostředkovatelům mezi klientem a serverem a stále bude přijímat odpovědi ze serveru. Servery mohou také předávat požadavky na jiné servery. Webová služba REST se může navrhnout tak, aby běžela na několika serverech s více vrstvami. Tyto vrstvy zůstávají pro klienta neviditelné.

- **Možnost ukládat do mezipaměti**

  Webové služby REST podporují ukládání do mezipaměti. Například dojde-li k navštívení webu, který má na každé stránce společné obrázky záhlaví a zápatí tak pokaždé, když uživatel navštíví novou webovou stránku, server musí znovu odeslat stejné obrázky. Aby se tomu zabránilo, klient uloží tyto obrázky po první odpovědi serveru a poté používá tyto obrázky přímo z mezipaměti.

5.2.1 Hlavní benefity REST API

Mezi benefity technologie REST API patří:

- **Škálovatelnost**

  Systémy, které implementují REST API, se mohou efektivně škálovat, protože REST optimalizuje interakce klient-server. Bezestavovost odstraňuje zatížení serveru. Server nemusí uchovávat informace o požadavcích klienta z minulosti. Dobře spravované ukládání do mezipaměti částečně nebo úplně eliminuje některé interakce klient-server. Všechny tyto funkce podporují škálovatelnost, aniž by způsobovaly překážky v komunikaci, které snižují výkon.

- **Flexibilita**

  Webové služby REST podporují úplné oddělení klient-server. Zjednodušují a oddělují různé komponenty serveru tak, aby se každá část mohla vyvíjet nezávisle. Změny platformy nebo technologie v serverové aplikaci nemají vliv

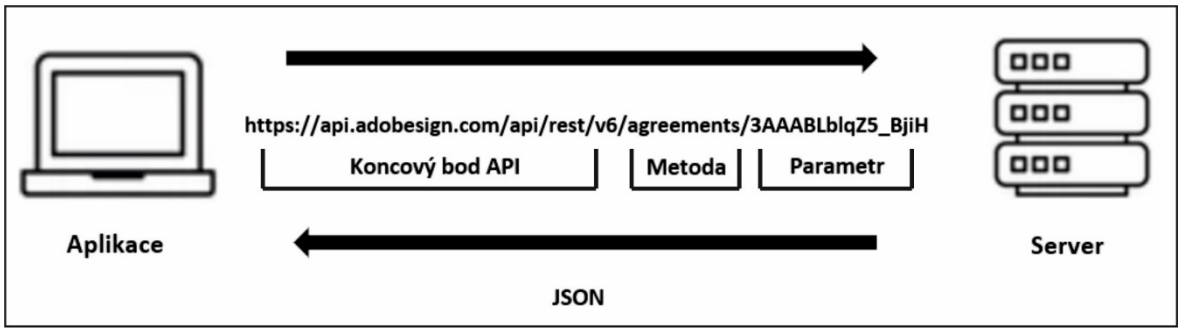
na klientskou aplikaci. Schopnost vrstvit aplikační funkce ještě dále zvyšuje flexibilitu. Vývojáři mohou například provádět změny v databázové vrstvě, aniž by museli přepisovat vrstvu aplikační.

- **Nezávislost**

  REST API jsou nezávislá na použité technologii. Klientské i serverové aplikace mohou být psány v různých programovacích jazycích, aniž by to ovlivnilo návrh rozhraní API. Také může být změněna základní technologie na obou stranách, aniž by to ovlivnilo komunikaci.

5.2.2 Jak funguje REST API?

Základní funkce REST API je stejná jako procházení internetu. Klient kontaktuje server pomocí rozhraní API, když vyžaduje nějaký zdroj. Vývojáři určují, jak by měl klient používat rozhraní REST API v dokumentaci příslušné aplikace. Obrázek 10 popisuje základní princip fungování REST API.



Zdroj: Vlastní zpracování

Obr. 10 Základní princip fungování REST API.

5.2.3 Komponenty požadavku klienta

Mezi komponenty požadavku klienta patří:

- **Unikátní identifikátor zdroje (URI)**

  Server identifikuje každý zdroj pomocí URI. U služeb REST server obvykle provádí identifikaci zdrojů pomocí adresy URL (Uniform Resource Locator). Adresa URL je podobná adrese webové stránky, kterou uživatel zadává do prohlížeče při návštěvě jakékoli webové stránky. Adresa URL se také nazývá koncový bod požadavku (endpoint) a jasně určuje serveru, co klient požaduje.

- **Metoda**

Vývojáři často implementují rozhraní REST API pomocí protokolu HTTP (Hypertext Transfer Protocol). Metoda HTTP říká serveru, co má udělat se zdrojem. Používají se metody „GET“, „PUT“, „POST“ a „DELETE“.

- **„GET“**
  Klienti používají „GET“ pro přístup k prostředkům, které jsou umístěny na zadané adrese URL na serveru.
- **„POST“**
  Klienti používají „POST“ k odesílání dat na server. Zahrnují reprezentaci dat s požadavkem. Odeslání stejného požadavku „POST“ vícekrát má za následek vytvoření duplicit.
- **„PUT“**
  Klienti používají „PUT“ k aktualizaci existujících zdrojů na serveru. Na rozdíl od „POST“ poskytuje vícenásobné odeslání stejného požadavku „PUT“ pouze aktualizaci zdroje.
- **„DELETE“**
  Klienti používají požadavek „DELETE“ ke smazání zdroje.

- **Parametry**
  Požadavky REST API mohou obsahovat parametry, které poskytují serveru další podrobnosti o tom, co je třeba udělat.

5.2.4 Komponenty odpovědí serveru

Mezi komponenty požadavku serveru patří:

- **Stavový řádek**
  Stavový řádek obsahuje třímístný stavový kód, který oznamuje úspěch nebo selhání požadavku. Například kódy 2XX označují úspěch, ale kódy 4XX a 5XX označují chyby.
- **Tělo zprávy**
  Tělo odpovědi obsahuje reprezentaci zdroje. Server vybere vhodný formát reprezentace na základě toho, co obsahují hlavičky požadavků. Klienti mohou požadovat informace ve formátech XML nebo JSON, které definují, jak jsou data zapsána v prostém textu.

5.2.5 Autentizační metody

Webová služba musí před odesláním odpovědi autentizovat požadavky. Autentizace je proces ověření identity. Totožnost může být prokázána například předložením občanského nebo řidičského průkazu. Podobně musí klienti služby REST prokázat svou identitu serveru, aby byla nastolena důvěra.

- **Základní autentizace**
  Uživatel zašle v záhlaví požadavku uživatelské jméno a heslo.
- **Autentizace držitele (bearer authentication)**
  Termín autentizace držitele odkazuje na proces poskytování přístupu držiteli tokenu. Token držitele je obvykle zašifrovaný řetězec znaků, který server generuje jako odpověď na žádost o přihlášení. Klient odešle token v záhlaví požadavku pro přístup ke zdrojům.
- **OAuth**
  Kombinuje hesla a tokeny pro vysoce bezpečný přihlašovací přístup do jakéhokoli systému. Server nejprve požaduje heslo a poté požádá o další token k dokončení procesu autorizace.

5.3 Adobe Acrobat Sign

Adobe Acrobat Sign je cloudová služba, která pomáhá organizacím a jednotlivcům nahradit inkoustový podpis podpisem digitálním. Tato služba umožňuje odesílat, podepisovat, sledovat a spravovat proces podpisu a elektronicky podepisovat digitální dokumenty na jakémkoli zařízení a z libovolného místa.

Zdroj: Adobe Acrobat Sign

Obr. 11 Aplikace ve službě Adobe Acrobat Sign

Po vytvoření aplikace (viz obr. 11) ve vývojářském účtu je třeba učinit následující kroky:

1. **Konfigurace OAuth**
   Adobe Sign používá ověřovací protokol OAuth k autorizaci požadavků pro jakýkoli „endpoint“ Adobe Sign API. Proces OAuth požaduje, aby si klientská aplikace vyžádala oprávnění od koncového uživatele, než za něj provede jakoukoli akci. Uživatelé se přesměrují do aplikace Acrobat Sign, kde se autentizují a udělí se jim požadovaná oprávnění. Aplikace poté přesměruje uživatele zpět do klientské webové aplikace. Proces OAuth vrací přístupové tokeny, které lze použít k volání jednotlivých REST API. K tomuto procesu jsou třeba ID aplikace a tajný klíč.
2. **Konfigurace identifikátoru URI**
   Dále je třeba vytvořit veřejný identifikátor URI pro přesměrování, který dokáže zachytit podrobnosti účtu a kód odeslaný z požadavku na autorizaci aplikace, aby došlo k propojení účtu Adobe Sign.
3. **Konfigurace oprávnění**

Nyní je třeba nastavit oprávnění, které aplikace bude mít při interakci s rozhraními Adobe Sign API. Oprávnění popisují, k jakým zdrojům a akcím bude mít aplikace přístup. Je třeba povolit alespoň minimální oprávnění.

4. **Přidání odkazu pro žádost o autorizaci**

   Aplikace musí obsahovat odkaz, který uživatelé používají k zahájení procesu požadavku OAuth. Proces OAuth začíná tím, že klient provede požadavek na koncový bod /public/oauth/v2 s požadovanými parametry.

5. **Vygenerování přístupového tokenu**

   Je třeba zažádat o přístupový token zasláním autorizačního kódu spolu s ID klienta a tajným klíčem klienta do služby Sign (viz obr. 12). Klient provede požadavek typu POST do koncového bodu /oauth/v2/token.

```
POST /oauth/token HTTP/1.1
Host: api.na1.adobesign.com
Content-Type: application/x-www-form-urlencoded
Cache-Control: no-cache

  grant_type=authorization_code&
  code=CBNCKBAThIsIsNoTaReAlcs_sL4K32wCzs4N&
  client_id=CBAThIsIsNoTaReAlmPBvPF&
  client_secret=319UThIsIsNoTaReAl2-4OxkVo9ycU&
  redirect_uri=https://myserver.com HTTP/1.1
```

Zdroj: Vlastní zpracování

Obr. 12 Požadavek pro vygenerování přístupového tokenu

Na obrázku 13 je vidět odpověď ve formátu JSON obsahující přístupový token typu OAuth.

```
{
   "access_token":"3AAABLblThIsIsNoTaReAlToKeNPr6Cv8KcZ9p7E93k2Tf",
   "refresh_token":"3AAABLblThIsIsNoTaReAlToKeNWsLa2ZBVpD0uc*",
   "token_type":"Bearer",
   "expires_in":3600,
   "api_access_point ":"https://api.na1.adobesign.com/",
   "web_access_point":" https://secure.na1.adobesign.com/"
}
```

Zdroj: Vlastní zpracování

Obr. 13 Odpověď ve formátu JSON

Customer Relationship Management (CRM) může odesílat/nahrávat dokumenty k podpisu, a to buď automaticky, nebo prostřednictvím akcí iniciovaných uživatelem. Když dokument podepíšou všechny strany, aplikace může získat kopii podepsaného dokumentu (smlouvy) ve formátu PDF.

Nahrání dokumentu

K nahrání PDF dokumentu k podpisu je zapotřebí odeslat požadavek typu POST na koncový bod „transientDocuments“ (viz obr. 14). Odpovědí je ID dokumentu, které jedinečně reprezentuje daný dokument. Aplikace musí specifikovat příjemce a další možnosti odesílání potřebné pro odeslání dokumentu k podpisu. Aplikace může také určit adresu URL zpětného volání, kterou bude aplikace Acrobat Sign používat k upozornění na dokončení procesu podpisu.

```
POST /api/rest/v6/transientDocuments HTTP/1.1
Host: api.na1.echosign.com
Authorization: Bearer MvyABjNotARealTokenHkYyi
Content-Type: multipart/form-data
Content-Disposition: form-data; name=";File"; filename="MojePDF.pdf"

<PDF CONTENT>
```

Zdroj: Vlastní zpracování

Obr. 14 Požadavek pro nahrání dokumentu

```
{
    "transientDocumentId":"3AAABLblqZhBVYbgJbl--NotArEaLID_zjaBYK"
}
```

Zdroj: Vlastní zpracování

Obr. 15 Odpověď ve formátu JSON 2

Odeslání Dokumentu

Pro odeslání dokumentu k podpisu je třeba vytvořit požadavek typu POST, který bude odeslán do koncového bodu „agreements“ (viz obr. 15).

```
POST /api/rest/v6/agreements HTTP/1.1
Host: api.na1.echosign.com
Authorization: Bearer 3AAABLblNOTREALTOKENLDaV
Content-Type: application/json

{
    "fileInfos": [{
        "transientDocumentId": "<id-dokumentu-z-predchoziho-api-volani>"
    }],
    "name": "Moje-testovaci-smlouva",
    "participantSetsInfo": [{
        "memberInfos": [{
            "email": "podepisujici@skoda-auto.cz"
        }],
        "order": 1,
        "role": "SIGNER"
    }],
    "signatureType": "ESIGN",
    "state": "IN_PROCESS"
}
```

Zdroj: Vlastní zpracování

Obr. 15 Požadavek na odeslání dokumentu k podpisu

Na obrázku 16 je vidět odpověď, což je ID smlouvy, které musí být použito jako odkaz na smlouvu ve všech následujících volání API.

```
{
    "id": "<id-generovane-adobe-sign>"
}
```

Zdroj: Vlastní zpracování

Obr. 16 Odpověď ve formátu JSON 3

```
GET /api/rest/v6/agreements/3AAABLblqZNOTREALAGREEMENTID5_BjiH HTTP/1.1
Host: api.na1.echosign.com
Authorization: Bearer 3AAANOTREALTOKENMS-4ATH
```

Zdroj: Vlastní zpracování

Obr. 17 Požadavek na kontrolu stavu podepisovaného dokumentu

Kontrola stavu podepisovaného dokumentu

Adobe Sign umožňuje zkontrolovat stav dokumentu, který byl odeslán k podpisu, a to pomocí odeslání požadavku (viz obr. 17) typu GET na koncový bod „agreements/{agreementid}“.

Na obrázku 18 jsou je vidět odpověď, což jsou konkrétní informace o podepisovaném dokumentu

```
{
  "id": "<id-generovane-adobe-sign>",
  "name": "Moje-testovaci-smlouva",
  "participantSetsInfo": [{
    "memberInfos": [{
      "email": "podepisujici@skoda-auto.cz",
      "securityOption": {
        "authenticationMethod": "NONE"
      }
    }],
    "role": "SIGNER",
    "order": 1
  }],
  "senderEmail": "odesilatel@skoda-auto.cz",
  "createdDate": "2022-11-16T08:13:16Z",
  "signatureType": "ESIGN",
  "locale": "eu_CZ",
  "status": "OUT_FOR_SIGNATURE",
  "documentVisibilityEnabled": false
}
```

Zdroj: Vlastní zpracování

Obr. 18 Odpověď ve formátu JSON 4

Stažení smlouvy

Jakmile je smlouva podepsána, aplikace může získat podepsanou kopii PDF a uložit ji. Na obrázku 19 je popsán dotaz pro stažení smlouvy.

```
GET /api/rest/v6/agreements/3AAA5NOTREALIDiH/combinedDocument HTTP/1.1
Host: api.na1.echosign.com:443
Authorization: Bearer 3AAABLblqZhB9BF
```

Zdroj: Vlastní zpracování

Obr. 19 Požadavek na stažení smlouvy

Odpovědí je obsah PDF souboru, který může být uložen lokálně.

Závěr

Cílem práce bylo popsat čtenáři základní principy digitálního podpisu. K tomuto cíli autor dospěl popisem těchto principů v prvních čtyřech kapitolách, kde autor mimo jiné popisuje historii podpisu jako takového.

V praktické části práce se autor věnuje popisu technologie „Application Programming Interface“ pomocí které se implementují služby poskytující digitální podepisování. Dále autor popisuje službu Adobe Acrobat Sign a její implementaci pomocí API, kde jsou i uvedené praktické příklady využití této technologie.

V důsledku nasazení technologie digitálního podpisu podnik spotřebuje výrazně méně papíru čímž šetří nejen náklady, ale i životní prostředí.

Další směr, kterým by mohl podnik rozvíjet digitalizaci je spojení služby Adobe Acrobat Sign s Contract Management Systémem. CRM je proces, ve kterém podnik spravuje své interakce se zákazníky, obvykle pomocí analýzy dat ke studiu velkého množství informací. Propojení Adobe Sign s například službou Salesforce je možno docílit ještě vyšší automatizace čímž podnik šetří další prostředky.

Seznam literatury

AJAY, Ohri. *Symmetric And Asymmetric Key Cryptography* [online]. 2022 [cit. 03.09.2022]. Dostupné z: https://www.jigsawacademy.com/blogs/cyber-security/symmetric-and-asymmetric-key-cryptography

BALLAD, Bill - BALLAD, Tricia - BANKS, Erin. *Access Control, Authentication, & Public Key Infrastructure.* USA: Jones & Bartlet, 2010. ISBN 978-07-637-9128-5.

LUTKEVICH, Ben. *Digital Signature* [online] 2021 [cit. 05.08.2022] Dostupné z: https://www.techtarget.com/searchsecurity/definition/digital-signature
Cambridge English Dictionary. *Signature* [online] Nedatováno [cit. 01.08.2022]. Dostupné z: https://dictionary.cambridge.org/dictionary/english/signature

LEE, Caroline*. What Is a Digital Signature and How It Works* [online] 2021 [cit. 02.08.2022]. Dostupné z: https://www.entrepreneur.com/en-in/technology/what-is-a-digital-signature-and-how-it-works/374465

Cryptomathic. *What is eIDAS?* [online] 2018 [cit. 01.08.2022]. Dostupné z: https://www.cryptomathic.com/products/authentication-signing/digital-signatures-faqs/what-is-eidas

SIMMONS, Gustavus*. RSA encryption* [online] 2009 [cit. 20.08.2022] Dostupné z: https://www.britannica.com/topic/RSA-encryption

SIMMONS, Gustavus*. Vigenere cipher* [online] 2009 [cit. 02.09.2022]. Dostupné z: https://www.britannica.com/topic/Vigenere-cipher

HAWKINS, Chris. (2011) *A History of Signatures: From Cave Paintings to Robo-Signings*. USA: CreateSpace Independent Publishing Platform, 2011. ISBN 978-1460978443

Ionos. *Public key encryption: What is public key cryptography?* [online] 2022 [cit. 27.08.2022]. Dostupné z: https://www.ionos.com/digitalguide/server/security/public-key-encryption/

HANÁK, Jakub – PRUŠKA, Lukáš. *Elektronický podpis pohledem aktuální právní úpravy* [online] 2020 [cit. 20.08.2022]. Dostupné z: https://www.epravo.cz/top/clanky/elektronicky-podpis-pohledem-aktualni-pravni-upravy-110560.html

FRUHLINGER, Josh. *What is PKI? And how it secures just about everything* [online] 2020 [cit. 25.08.2022]. Dostupné z: https://www.csoonline.com/article/3400836/what-is-pki-and-how-it-secures-just-about-everything-online.html
Jscrambler. *Hashing Algorithms* [online]. Copyright © [cit. 02.09.2022].

Dostupné z: https://blog.jscrambler.com/hashing-algorithms

KOESTLER, Karolin. *Electronic Signatures: A Brief History* [online] 2022 [cit. 05.08.2022]. Dostupné z: https://www.foxit.com/blog/the-history-of-electronic-signatures/

KATZ, J. *Digital Signatures (Advances in Information Security).* USA: Springer, 2010. ISBN 978-03-872-7711-0.

Legalesign, *The history of the Signature* [online] 2016 [cit. 01.08.2022]. Dostupné z: https://legalesign.com/blog/history-of-signatures/

PALMER, Lucy – MANGHAM, Gabriella. *The Evolution Of The Signature: From The Sumerians To DocuSign* [online] 2022 [cit. 14.08.2022]. Dostupné z: https://www.azeusconvene.co.uk/blog/the-evolution-of-the-signature

LANDMAN, Nathan – WILLIAMS, Christopher - ROSS Eli. *Secure Hash Algorithms* [online] 2022 [cit. 14.09.2022] Dostupné z: https://brilliant.org/wiki/secure-hashing-algorithms/

MCDONALD, Nicholas. *Past, present, and future methods of cryptography and data encryption* [online] 2009 [cit. 14.08.2022]. Dostupné z: https://my.eng.utah.edu/~nmcdonal/Tutorials/EncryptionResearchReview.pdf

FLEMING, Sean. *A brief history of cryptography and why it matters* [online] 2019 [cit. 02.09.2022]. Dostupné z: https://www.weforum.org/agenda/2019/02/a-brief-history-of-cryptography-and-why-it-matters/

SHA1-online. *SHA1 and other hash functions online generator* [online] 2022 [cit. 01.09.2022] Dostupné z: http://www.sha1-online.com

SMART, Nigel. (2016) *Cryptography Made Simple (Information Security and Cryptogra- phy).* Springer, 2016, 1st ed. ISBN-13: 978-33-193-7309-6

CHANDRA, Sourabh. *A study and analysis on symmetric cryptography* [online] 2014 [cit. 02.09.2022]. Dostupné z: https://ieeexplore.ieee.org/abstract/document/7043664

VOHNOUTOVÁ, Marta – DOSTÁLEK, Libor. *Velký průvodce infrastrukturou PKI a technologií elektronického podpisu.* ČR: Computer Press, 2010. ISBN 978-80-251-2619-6.

KOTAS, William August. *A Brief Hist A Brief History of Cryptography* [online] 2019 [cit. 10.08.2022]. Dostupné z: https://trace.tennessee.edu/cgi/viewcontent.cgi?article=1398&=&context=utk_chanhonoproj&=&sei-redir=1&referer=https%253A%252F%252Fscholar.google.com%252Fscholar%253Fhl%253Den%2526as_sdt%253D0%25252C5%2526q%253Dhistory%252Bof%252Bcryptography%2526btnG%253D#search=%22history%20cryptography%22

Seznam obrázků a tabulek

Seznam obrázků

Seznam tabulek

ANOTAČNÍ ZÁZNAM

AUTOR	**Štěpán Havlas**		
STUDIJNÍ PROGRAM/OBOR/SPECIALIZACE	**Podniková ekonomika a manažerská informatika**		
NÁZEV PRÁCE	**Digitální podpisy a jejich implementace ve ŠKODA AUTO a.s.**		
VEDOUCÍ PRÁCE	**Ing. Vladimír Beneš, Ph.D.**		
KATEDRA	**KI - Katedra informatiky**	**ROK ODEVZDÁNÍ**	**2022**

POČET STRAN	**49**
POČET OBRÁZKŮ	**19**
POČET TABULEK	**1**
POČET PŘÍLOH	**0**

STRUČNÝ POPIS	**Hlavním cílem práce je popsat čtenáři implementaci služby Adobe Acrobat Sign, která využívá digitálního podpisu, pomocí technologie „Application Programming Interface“ aniž by disponoval hlubokými znalostmi v oblasti informačních technologií.** **V praktické části práce se autor věnuje popisu zmíněné technologie pomocí které se implementují služby poskytující digitální podepisování. Dále autor popisuje službu Adobe Acrobat Sign a její implementaci pomocí API, kde jsou i uvedené praktické příklady využití této technologie.**
KLÍČOVÁ SLOVA	**Asymetrická kryptografie, Application programming interface, Infrastruktura veřejného klíče, Veřejný klíč, Privátní klíč, Bezpečný hashovací algoritmus**

ANNOTATION

AUTHOR	**Štěpán Havlas**		
FIELD	**Business Informatics**		
THESIS TITLE	**Implementation of digital signatures in ŠKODA AUTO .a.s**		
SUPERVISOR	**Ing. Vladimír Beneš, Ph.D.**		
DEPARTMENT	**KI - Department of Informatics**	**YEAR**	**2022**

NUMBER OF PAGES	**49**
NUMBER OF PICTURES	**19**
NUMBER OF TABLES	**1**
NUMBER OF APPENDICES	**0**

SUMMARY	**The main goal of the work is to describe to the reader the implementation of Adobe Acrobat Sign services, which uses a digital signature, using the "Application Programming Interface" technology without having deep knowledge in the field of information technology.** **In the practical part of the work, the author describes the mentioned technologies that are used in implementation of digital signing services. Furthermore, the author describes the Adobe Acrobat Sign service and its implementation using the API, where practical examples of the use of this technology are presented.**
KEY WORDS	**Asymmetric cryptography, Application programming interface, Public key infrastructure, Public key, Private key, Secure hash algorithm**