Univerzita Hradec Králové
Fakulta informatiky a managementu
Katedra informačních technologií

Zabezpečení TLS

Bakalářská práce

Autor: Zdeněk Hejzlar
Studijní obor: Aplikovaná informatika

Vedoucí práce: Ing. Hana Švecová

Hradec Králové duben 2022

Prohlášení:

Prohlašuji, že jsem bakalářskou práci zpracoval samostatně a s použitím uvedené literatury.

Zdeněk Hejzlar

V Hradci Králové dne 25.4.2022 Jméno a příjmení

Poděkování:

Děkuji vedoucí bakalářské práce Ing. Haně Švecové za metodické vedení práce a konzultace týkající se tvorby mého bakalářského projektu. Děkuji Mgr. Tereze Pechové za odborné konzultace při psaní práce. Taktéž děkuji všem přátelům, blízkým a rodině za psychickou podporu.

Anotace

Cílem bakalářské práce je analýza aktuálně využívaných kryptografických algoritmů s následnou komparací, návrhem a vytvořením výukové aplikace pro výuku kryptografie.

Čtenář je seznámen s historickým vývojem kryptologie a jejími vědními disciplínami (kryptografie, kryptoanalýza) a dále s principy symetrického a asymetrického šifrování včetně nejvyužívanějších algoritmů.

Výuková aplikace by měla být přínosem pro výuku dílčích oblastí kryptografie na Fakultě informatiky a managementu Univerzity Hradec Králové.

Annotation

Title: Transport Layer Security

The aim of the bachelor thesis is to analyze currently used cryptographic algorithms with subsequent comparison, design and creation of an educational application for teaching cryptography.

The reader is introduced to the historical development of cryptology and its disciplines (cryptography, cryptanalysis), as well as the principles of symmetric and asymmetric encryption, including the most used algorithms.

The educational application should be beneficial for teaching partial areas of cryptography at the Faculty of Informatics and Management of the University of Hradec Králové.

Obsah

Seznam obrázků

1 Úvod

Bakalářská práce je rozdělena do části teoretické a praktické. V teoretické části je nastíněn historický kontext vývoje kryptologie a souvisejících vědních disciplín (kryptografie, kryptoanalýza) a dále jsou rozvedeny dílčí metody pro šifrování včetně nejvyužívanějších algoritmů.

„První kryptografické techniky (tzv. šifry) vznikly prakticky vzápětí po vzniku písma, neboť jejich účelem bylo utajit obsah písemných zpráv před nepovolanými čtenáři. Historicky nejstarším známým důkazem o praktickém využití kryptografie je hliněná destička ze starověké Mezopotámie z období asi 1 500 let před naším letopočtem, na níž je uveden šifrovaný popis technologie výroby glazurované keramiky“ [1].

Mezi související vědní obory řadíme kryptografii, která se zabývá využíváním matematiky pro šifrování a dešifrování dat. Za pomoci šifrování lze tato data předávat bezpečně mezi koncovými body přes nezabezpečené sítě (např. internet).

Posledním odvětvím je kryptoanalýza zabývající se analýzou a prolamováním šifrovacích algoritmů. Tato část vyžaduje zapojení analytického myšlení, aplikaci matematických nástrojů, ale také schopnost hledat dané vzorce [2].

Pro šifrování využíváme takzvaných kryptografických algoritmů. Tyto postupy lze rozdělit do dvou kategorií dle využitého klíče na symetrické a asymetrické [3].

Výuka dílčích částí kryptografie se postupně stává součástí odborných předmětů vyučovaných v informatických oborech, avšak pro názornou ukázku nejvyužívanějších šifrovacích algoritmů chybí přehledný výukový software, který by studentům poskytoval komplexní přehled z jednoho místa.

Výuková aplikace je zpracována/naprogramována ve vícejazyčné verzi (CZ, EN), aby tato aplikace mohla být využívána i zahraničními studenty.

2 Metodika zpracování a cíl práce

Přínosem a současně cílem této bakalářské práce je především zpracování výukové aplikace pro využití ve výuce na Fakultě informatiky a managementu Univerzity Hradec Králové.

Pro korektní zpracování této kvalifikační práce proběhla individuální analýza šifrovacích algoritmů, systémů a protokolů využívaných pro zabezpečení citlivých dat či informací. Při analýze bylo čerpáno z odborné literatury, zahraničních článků a dalších volně dostupných internetových zdrojů.

V první teoretické části bude nastíněn historický vývoj kryptologie a souvisejících disciplín (kryptografie, kryptoanalýza) včetně využívaných metod (symetrické a asymetrické šifrování) a nejvyužívanějších algoritmů.

V druhé praktické části bude popsána naprogramovaná výuková aplikace dostupná přes webové rozhraní na https://edu.uhk.cz/~hejzlzd1/, dále byla vytvořena i cizojazyčná verze v anglickém jazyce pro výuku zahraničních studentů.

Pro zpracování / naprogramování aplikace byl využit programovací jazyk PHP, který je zaměřen na serverové vykonávání kódu a zobrazování zpracovaných dat uživateli. Dále bylo využito jazyku HTML (Hyper Text Markup Language), zaměřeného na tvorbu statických webů. Takto připravený web byl následně stylizován pomocí kaskádových stylů (tzv. CSS) a skriptovacího jazyku s názvem Javascript.

Aplikace je provozována na webovém serveru Univerzity Hradec Králové na doméně edu.uhk.cz. Pro správné fungování projektu je využito i databázového serveru, který se nachází na stejnojmenné doméně. Pro databázové prvky je využito relační databáze s názvem MySQL.

3 Teoretická část

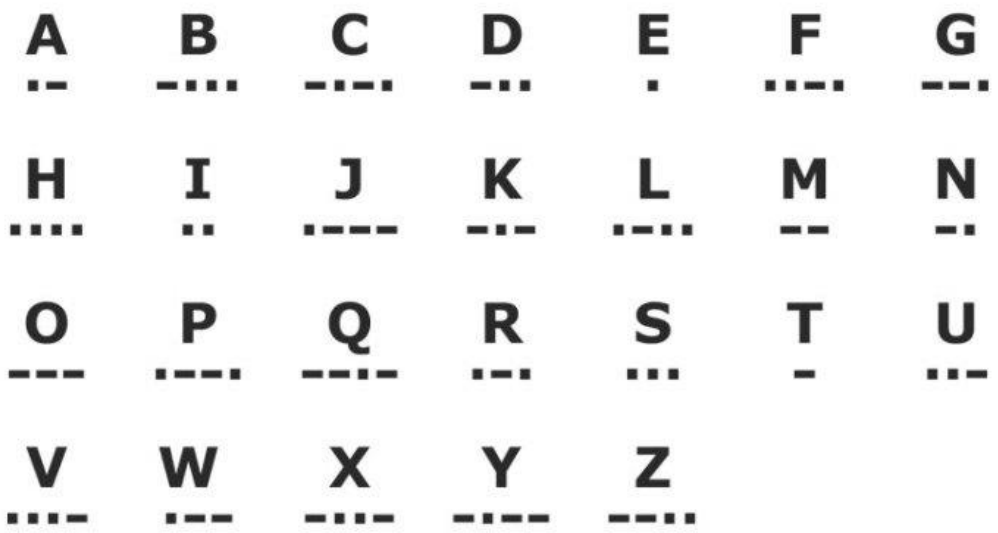
3.1 Kryptologie

Pro začátek je potřeba stanovit pojem „kryptologie“. Tímto pojmem označujeme vědu, která se zabývá šifrováním a dešifrováním zpráv. Tato vědní disciplína vychází z matematických věd, které se opírají o informační technologie. [4]. Bez této vědy by digitální data byla v nezabezpečené formě. V případě odcizení dat by bylo velmi jednoduchá data (aktiva) zneužít v rámci kybernetických útoků z důvodu snadno čitelného výstupu (formátu). S narůstajícími kybernetickými útoky se postupně zvyšuje i zájem o zabezpečení a utajování informací.

Historický vývoj kryptografie pochází z období Starověkého Egypta, kdy vznikaly první šifry v podobě hieroglyfů. Velký rozmach v této oblasti nastal především v průběhu první a druhé světové války. Kryptologie však nebyla využívána pouze ve vojenství, avšak své využití si našla i mezi běžnou populací.

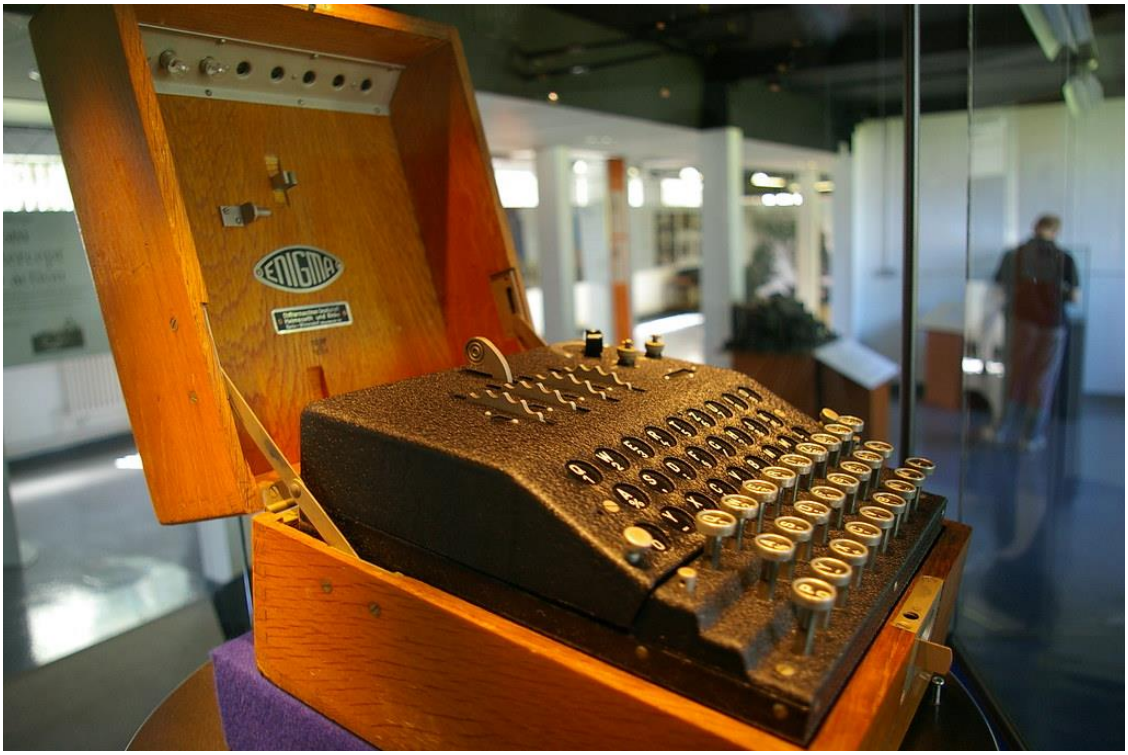
V současné době je kryptologie využívána prakticky v každém moderním zařízení (Smart technologie, IoT) či informačních systémech např. s využitím umělé inteligence (AI). Příkladem využití kryptologie může být šifrování osobních dat v databázích, šifrování hesel, šifrování digitálních podpisů či ý protokolu Hypertext Transfer Protocol Secure (HTTPS).

Do kryptologie zásadně přispěl také vynález telegrafu, díky kterému vznikla **Morseova abeceda**. Nejednalo se o klasickou šifru, avšak o jednoduchý šifrovací systém sloužící k jednoduchému přenosu informací. Tento vynález zjednodušil komunikaci pro další generace. Zpráva byla zašifrována pomocí tabulky do podoby krátkých a dlouhých zvukových signálů značených tečkou či pomlkou. Každý znak měl ustálenou kombinaci teček a pomlk, které se nedaly zaměnit. V případě telegrafu pak bylo jednoduché informace dostat od odesílatele k příjemci vyťukáváním daných znaků. Příjemce si dané zvukové signály zaznamenal na papír a pomocí znalosti či tabulky převedl tyto signály na otevřený čitelný text.



Obrázek 1 - Morseova abeceda (Poornesh Pathak) [5]

Dalším důležitým milníkem v šifrování byla druhá světová válka a vznik šifrovacího stroje s názvem **Enigma**. Tento stroj byl využíván německým námořnictvem za účelem utajit informace před nepřátelskou zpravodajskou službou. Šifrovací algoritmus byl však do dvou let prolomen, což významně zkrátilo druhou světovou válku. Svým vzhledem Enigma připomínala běžný psací stroj. Šifrování zpráv pak probíhalo formou psaní otevřeného textu a zápisu znaků, které se rozsvítily na stroji. K dešifrování zpráv byly zapotřebí dešifrovací klíče v podobě nastavení Enigmy. Po nastavení stačilo psát na stroji nesmyslný text, který byl obdržen. Originální znaky stačilo zapsat z osvětlených znaků, čímž vznikala dešifrovaná informace. Díky tomuto stroji začal vzrůstat zájem o utajení informací.



Obrázek 2 – Šifrovací stroj Enigma (Tim Gage) [6]

Tato věda se v minulosti začala rozdělovat do dvou částí – **kryptografie** a **kryptoanalýza**. Další část s názvem **steganografie** většinou není uváděna a je často opomíjena.

3.1.1 Kryptografie

Jednou z nejdůležitějších částí při utajování informací ve zprávě je šifrování. Při procesu šifrování je převáděna původní „otevřená“ čitelná zpráva na zašifrovaný text. Výsledný text je v závislosti na použitém algoritmu nesmyslný či přímo nečitelný pro entitu, která nezná pomocné informace (klíče) k dešifrování. Tento proces se provádí podle předem určených pravidel či pomocí konkrétního zvoleného algoritmu. Příkladem zašifrované zprávy může být náhodná posloupnost čísel, písmen, slov či nesmyslných vět.

Kryptografie se zabývá celkovou problematikou šifrování a dešifrování dat, dále zkoumá bezpečnost využívaných šifrovacích algoritmů a také bezpečnost šifrovacích systémů.

3.1.2 Kryptoanalýza

Součástí kryptologie je taktéž možnost dešifrování dat bez dešifrovacích informací. Touto činností se zabývá právě kryptoanalýza, pro kterou je charakteristický zpětný postup oproti kryptografii. Je to věda zkoumající metody získávání obsahu šifrovaných zpráv bez přístupu k tajným informacím. Další aktivitou je zkoumání vlastností otevřeného textu, zašifrovaného textu a šifrovacích klíčů. Osoba zabývající se kryptoanalýzou se nazývá kryptoanalytik.

Kryptoanalytik nečitelné zašifrované informace zkoumá pomocí metod kryptoanalýzy s cílem nalezení originálních čitelných dat. Prvním krokem může být rozpoznání typu šifry, v druhém kroku pak využívá její nedokonalosti či slabiny k prolomení. Náročnost této operace je závislá na složitosti použitého šifrovacího algoritmu. Celá operace může trvat i několik minut, v horším případě až několik let. Důležitým faktorem je taktéž schopnost kryptoanalytika rozpoznat danou šifrovací metodu a znalost, jak k ní přistupovat bez znalosti klíčů.

3.1.3 Steganografie

Je to věda zabývající se skrytím informace o existenci zprávy. I tato část je podstatná v případě, kdy je potřeba utajit informace o tom, že existuje zašifrovaná zpráva.

3.2 Symetrické a asymetrické kryptovací systémy

Systémy pro šifrování můžeme rozdělit podle typů využitých šifrovacích klíčů. Tyto klíče se využívají při převádění otevřeného textu na zašifrovanou informaci. Tyto systémy dělíme na **symetrické** a **asymetrické šifrovací systémy**. Dalším rozdělením pak může být i samotná délka klíče, která u různých systémů ovlivní zabezpečení prováděné šifry. Délku pak určujeme v bitech.

3.2.1 Symetrické klíče

Princip symetrických klíčů spočívá v tvorbě stejného klíče nejen pro šifrování otevřeného textu, ale i pro jeho následnou dešifraci. Tato metoda je méně náročná na výpočetní výkon, ale také méně bezpečná při využití v síti. Při prolomení klíče

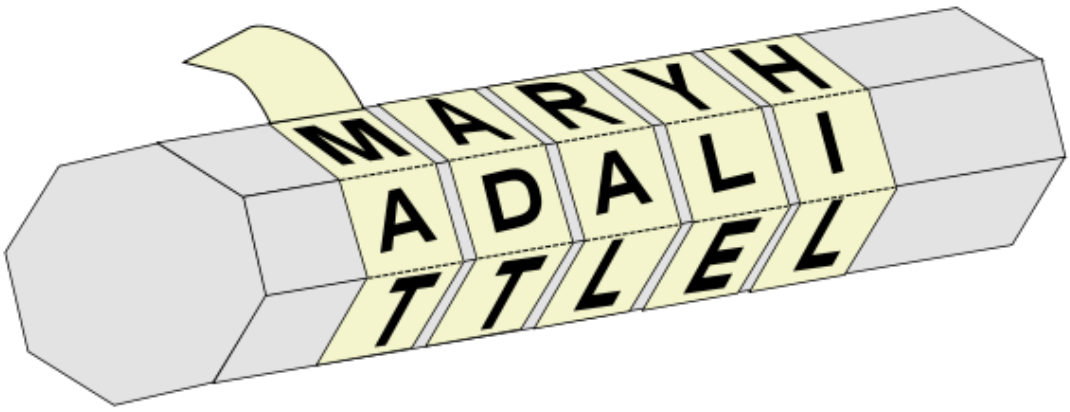
může dojít k úniku dat všude, kde byl využit stejný klíč, a k následnému podstrčení cizích dat.

Populární algoritmy využívající této metody jsou např. AES, Camellia, DES, 3DES, Idea či Blowfish.

3.2.1.1 Transpoziční systém

„Uvádí se, že jeho první zaznamenaná forma byla popsána řeckým historikem Plutarchem. Jedná se o transpoziční systém zvaný Scytale“ [7].

Scytale využívali sparťanští vojevůdci k utajení svých zpráv s rozkazy a plány. Tato šifra byla jednoduchá, avšak účinná. Vyžadovala, aby příjemce i odesílatel měli dřevěnou hůl o stejném poloměru. Kolem dřevěné hole se omotal pergamen, na který se následně psalo svisle a krátce. Po odmotání zbyl dlouhý pruh pergamenu s náhodnými znaky, které nedávaly smysl. Tento pruh „papíru“ se následně odeslal příjemci, který pro dešifrování zprávy musel obmotat svou totožnou hůl. V případě, že zprávu získal nepřítel, nebyl schopný dešifrovat text, jelikož stejným postupem na jiné holi získal směs náhodných znaků, které dohromady neměly hlubší význam [7].



Obrázek 3 – Sparťanské Scytale (Devlin M. Gualtieri) [8]

Systém transpoziční vyžaduje přesná a předem určená pravidla k přeskupování písmen otevřeného textu. U této metody je známá především Polybiova šifrovací mřížka. Sloupce a první řádky jsou nadepsány znaky nebo

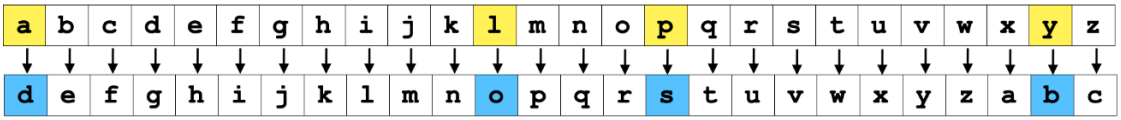
číslicemi. Zbytek tabulky je pak vyplněn abecedou. Šifrovaná zpráva vzniká výběrem znaku z abecedy a napsáním jeho pozice v tabulce. Dešifrování probíhá obráceným procesem – vyhledává se znak pomocí získaných souřadnic.

	1	2	3	4	5
1	A	B	C	D	E
2	F	G	H	I,K	K
3	L	M	N	O	P
4	Q	R	S	T	U
5	W	V	X	Y	Z

Obrázek 4 - Polybiusova mřížka (Zdeněk Hejzlar)

3.2.1.2 Substituční systém

Substituční systém funguje na principu nahrazování znaků otevřeného textu jinými znaky dle dohodnutého pravidla. Tato metoda šifrování je známá především díky *Caesarově šifře*. Tento snadný šifrovací algoritmus fungoval na jednoduchém principu posunutí abecedy o tři znaky doprava či doleva oproti originální abecedě. Šifrování otevřeného textu pak probíhalo zvolením znaku z první abecedy a zapsáním jeho druhé varianty v posunuté abecedě. Tento systém byl následně vylepšen na takzvaný polyabecední substituční systém, který je složen z několika substitučních šifer.



Obrázek 5 - Caesarova šifra (Omar Ibrahim) [9]

3.2.2 Asymetrické klíče

Asymetrické klíče zvyšují bezpečnost využití systému v síťové komunikaci. Při šifrování je oddělen klíč k šifrování od klíče určeného k dešifraci. Tyto klíče jsou nazývány jako veřejné a privátní. Systémy využívající asymetrické klíče mají většinou i funkci k zajištění bezpečného přenosu klíče k danému subjektu. Tvorba asymetrických klíčů je oproti jednodušším symetrickým klíčům (náhodným) značně výkonnostně náročnější.

Mezi systémy, které využívají tohoto způsobu, řadíme např. algoritmus RSA (Rivest–Shamir–Adleman). Postup tohoto systému spočívá ve čtyřech krocích – generování klíčů, distribuce, šifrování a dešifrování. Generace probíhá díky výběru dvou odlišných prvočísel, s kterými se následně provádí sofistikované matematické operace. Výstupem je pak privátní a veřejná informace. Soukromý klíč zůstává u jednoho z koncových bodů, veřejný se zasílá protistraně. Samotné šifrování pak probíhá převodem textu na číslo a využitím veřejného klíče. Privátní je následně využit na dešifrování získané zprávy.

Dalším komplexním systémem je Diffie-Hellman, který se oproti RSA nezabývá šifrováním textu, ale generací a zabezpečením přenosu informací. K tvorbě šifrovacích informací používá matematické operace, jako je modulo prvočísel

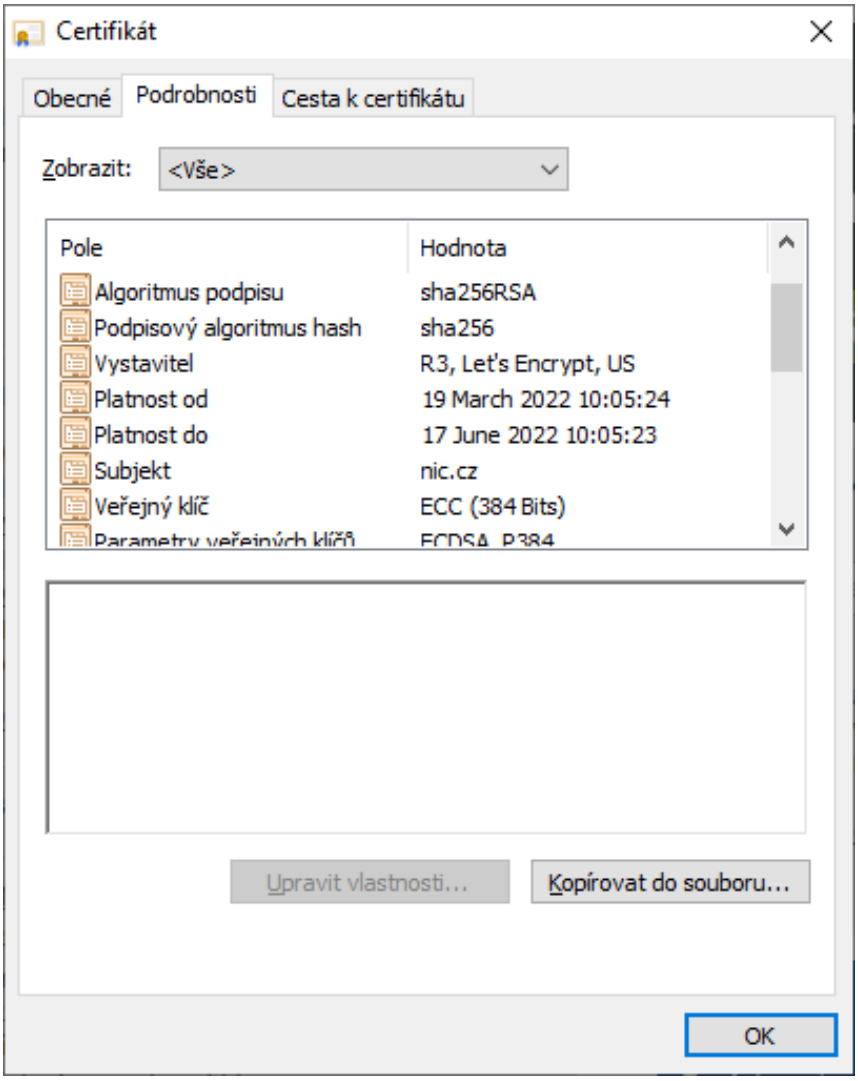
a mocniny. Výhodou je, že algoritmus využívá složitých operací, které bez upřesňujících dat trvají velice dlouho prolomit.

Taktéž sem lze zařadit populární veřejně využívaný DSA (Digital Signature Algorithm). Tento systém je využíván pro digitální podpisy a jejich validaci. První dva kroky má stejné jako RSA, avšak se složitější implementací. Za pomoci klíčů algoritmus ověřuje vytvořené podpisy u přenesených dat.

3.2.3 Moderní zabezpečení TLS a SSL

Pomocí SSL/TLS certifikátů je ověřováno bezpečné připojení k serveru. Tyto certifikáty umožňují autentifikaci serveru, což se projevuje známým prefixem HTTPS a ikonickou ikonkou zámečku v internetových prohlížečích. Tímto způsobem se dá zabránit podstrkování cizích dat či odposlouchávání. Pro své fungování TLS/SSL využívá obou typů šifrovacích algoritmů v závislosti na vykonávané aktivitě.

Tvorba těchto certifikátů je v zásadě jednoduchá, lze využít veřejných implementací. Jedním z poskytovatelů certifikátů je např. webová stránka letsencrypt.org, kde stačí zadat potřebné údaje. Vygenerovaný certifikát obsahuje název domény, název organizace, název vystavitele certifikátu, datum vystavení, datum zániku, digitální podpis a veřejný klíč. Taktéž může obsahovat případné subdomény.



Obrázek 6 - SSL certifikát nic.cz (Zdeněk Hejzlar)

Při připojení k serveru pod šifrovaným portem je tento certifikát odeslán klientovi ve fázi „handshake“. Tato fáze se skládá z 10 kroků. Na začátku spojení informuje klient server o úmyslu komunikace. Server odpovídá a zasílá svůj certifikát s klíčem k ověření.

Obrázek 7 – Handshake (Zdeněk Hejzlar)

Klient přijatý certifikát otestuje na straně vydavatele před další komunikací. Tato kontrola zajistí, že je server tím, za koho se vydává. Pokud je certifikát platný a jeho ověření na straně vydavatele také, pak se přechází k výměně klíčů a specifikaci další komunikace. V opačném případě je komunikace přerušena či pokračuje nezabezpečeně. Technologie SSL/TLS je využita v protokolech HTTP (Hypertext Tranfer Protocol) na portu 80, FTP (File Transfer Protocol) na portu 20-21 a dalších. Většina pak získává ke svým zkratkám na konec písmeno S jako secure. Další využití se nachází např. v zabezpečení SMTP (Simple Mail Transfer Protocol – port 25), které zprostředkovává zasílání elektronické pošty. Před odesláním se validuje pravost serveru, na který se e-mail odesílá a komunikace probíhá šifrovaně.

4 Výuková aplikace

Webová vzdělávací aplikace je zaměřena na srozumitelné představení informací týkajících se kryptologie. Při tvorbě projektu nebylo využito žádného frameworku (knihovny), protože se jedná o jednoduchou aplikaci s nízkým počtem funkcí. Webová aplikace je rozdělena do pěti hlavních sekcí, které jsou popsány v následujícím textu.

V **první** sekci se nachází stručné informace ohledně kryptografie, symetrických a asymetrických systémů a využití šifrování. Taktéž zde lze nalézt informace týkající se protokolů, SSL, TLS a jejich certifikátů.

Druhá sekce obsahuje šifrování uživatelem zadaných textových zpráv pomocí pěti zpracovaných symetrických algoritmů. Každý z uvedených algoritmů je stručně popsán a lze si ho vyzkoušet vyplněním formuláře.

Třetí část aplikace se zabývá asymetrickými algoritmy a jejich šifrováním textu. V tomto případě se jedná o jeden algoritmus, který je možné otestovat pomocí dostupného formuláře a jeho tlačítek.

Ve **čtvrté** sekci jsou v tabulce porovnány vybrané algoritmy na základě rychlosti šifrování datových souborů. Výběr algoritmů byl omezen dostupností a výkonem na webovém serveru, pro který byla aplikace tvořena.

Poslední pátá sekce obsahuje tabulky s přístupem do databáze a tlačítka k jejím omezeným úpravám. Tato část ukazuje především důležitost šifrování dat v případě útoku a prolomení ochrany databázového systému.

Tvorba projektu byla inspirována již existující webovou stránkou na adrese https://www.cryptool.org.

4.1 Využité technologie pro zpracování výukové aplikace

Při tvorbě projektu byly využity technologie volně dostupné, taktéž byly použity některé předdefinované knihovny pro šifrování. Mezi využité technologie spadají následující:

- HTML (HyperText Markup Language)
- CSS (Cascading Style Sheets)
- Javascript

- PHP
- MySQL
- Apache

HTML je značkovací jazyk sloužící k tvorbě struktury statických webů. Jedná se o hlavní stavební kámen webových stránek či aplikací.
CSS aneb kaskádové styly jsou využívány k úpravě designu webu. Využívají se především v kombinaci s výše zmíněným HTML.
Název **Javascript** skrývá objektově orientovaný programovací jazyk řídicí se událostmi. Na webech je běžně využíván k úpravě obsahu v závislosti na uživatelských interakcích. V případě tohoto projektu je využit k animacím a reakcím na kliknutí některé ze záložek v menu.
PHP je skriptovací jazyk určený pro tvorbu dynamických webů a aplikací. Jeho funkce jsou prováděny na straně serveru a výsledek je následně prezentován uživateli. V případě vzdělávací aplikace je využit k připojení do databáze, k měření výkonu šifrovacích algoritmů a k samotnému šifrování v druhé sekci webu.
MySQL pracuje jakožto systém k řízení báze dat – zajišťuje databázi, která obsahuje množinu vybraných dat určených k uložení. Pro komunikaci s MySQL se využívá SQL (Structured Query Language), což je strukturovaný dotazovací jazyk. Na základě dotazů v tomto jazyce jsou pomocí systému pro řízení báze dat dodávány požadované informace. Propojení s php je zajištěno pomocí MySQLi.
Pod pojmem **Apache** lze nalézt webový server, který je světově využívaný. Server poskytnutý pro aplikaci spadá pod UHK (Univerzita Hradec Králové) a je na něm php verze 5.3.13, mysql o verzi 5.5.42

4.2 Použité knihovny

Výuková aplikace využívá veřejně dostupných knihoven pro úpravu designu a funkci některých prvků. Většina je získávána online pomocí CDN serverů. V následujícím textu jsou vypsány ty nejdůležitější:

- Bootstrap v5.1.3
- Jquery v3.6.0

- Datatables
- Openssl
- Aos

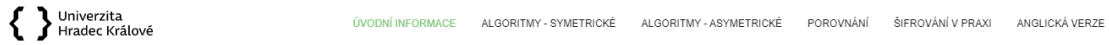
Bootstrap je front-end knihovna určena k jednoduché a responzivní tvorbě webových stránek. Tato knihovna je široce využívána díky svým předdefinovaným css stylům, moderní vizáži a užitečným funkcím. Taktéž se jednoduše využívá pomocí specifikování tříd či atributů u html prvků.
JQuery je knihovna v Javascriptu, která se využívá ke zjednodušení programování v daném programovacím jazyce. V projektu je zastoupena především kvůli závislosti Bootstrapu, Datatables a Aos. Tyto knihovny jquery využívají ke své funkčnosti.
Datatables zjednodušuje a vylepšuje tvorbu tabulek na webu. Jedná se o plugin pro Jquery, pomocí kterého lze vytvářet tabulky s vyhledáváním, řazením a s číslovanými stránkami s určitým počtem záznamů.
Openssl je modul funkcí v php určen pro symetrické a asymetrické šifrování a dešifrování. Pro funkčnost je potřeba mít správně nastavený webový server. Mezi poskytovanými funkcemi je i tvorba asymetrických klíčů.
Aos je určen pro aplikování animací na prvky. Implementace je snadná za pomoci atributu v html tagu.

4.3 Uživatelský interface

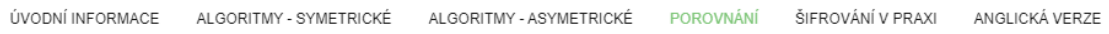
Pod pojmem user interface se nachází pojem, který popisuje metody, jak může uživatel komunikovat s aplikací. Tyto metody by měly být na první pohled zřejmé a jednoduché k pochopení, avšak u složitějších aplikací může nastat situace, kdy je UI nepřehledné. V rámci tvorby webové aplikace vzniklo jednoduché rozhraní, které je pro lepší pochopení v následujícím textu popsané.

4.3.1 Menu

V menu umístěném na vrchu stránky se nachází odkazy na jednotlivé celky. Po stisknutí některého z nadpisů je pohled převeden na daný obsah. V tomto menu se taktéž zvýrazňuje nadpis korespondující s obsahem, na kterém se nachází pohled uživatele.



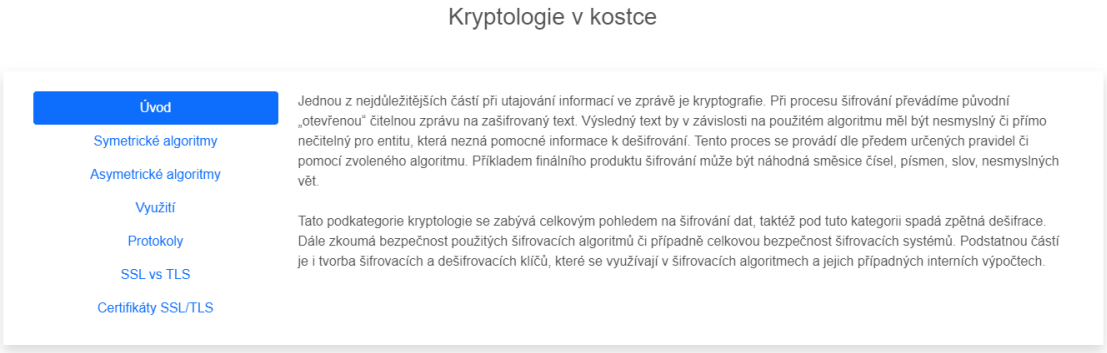
Obrázek 8 - UI – Menu (Zdeněk Hejzlar)



Obrázek 9 - UI – Menu detail (Zdeněk Hejzlar)

4.3.2 Sekce

Části stránky jsou rozmístěny do několika obsahových sekcí. Tyto sekce se vyznačují nadpisem a popisem dané části. Taktéž je možné je rozpoznat pomocí vržených stínů z objektů, které jsou pod nadpisem. Rozděleny jsou na několik tematických/funkčních okruhů.

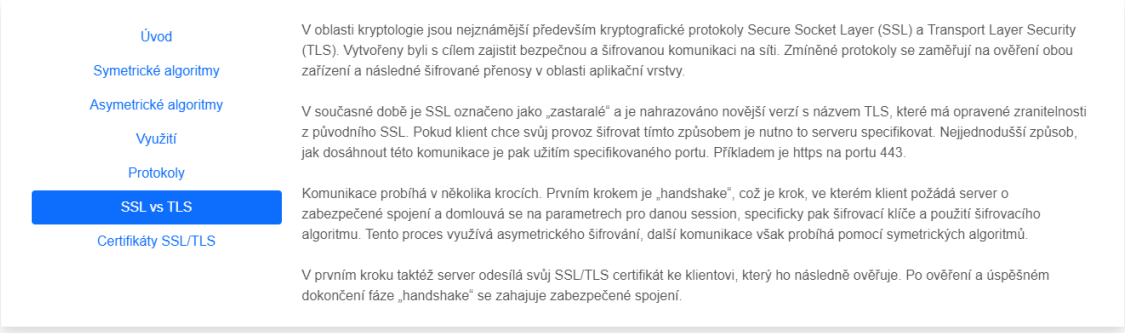


Obrázek 10 - UI – Ukázka sekce (Zdeněk Hejzlar)

4.3.3 Rozdělení webu

4.3.3.1 Úvodní informace

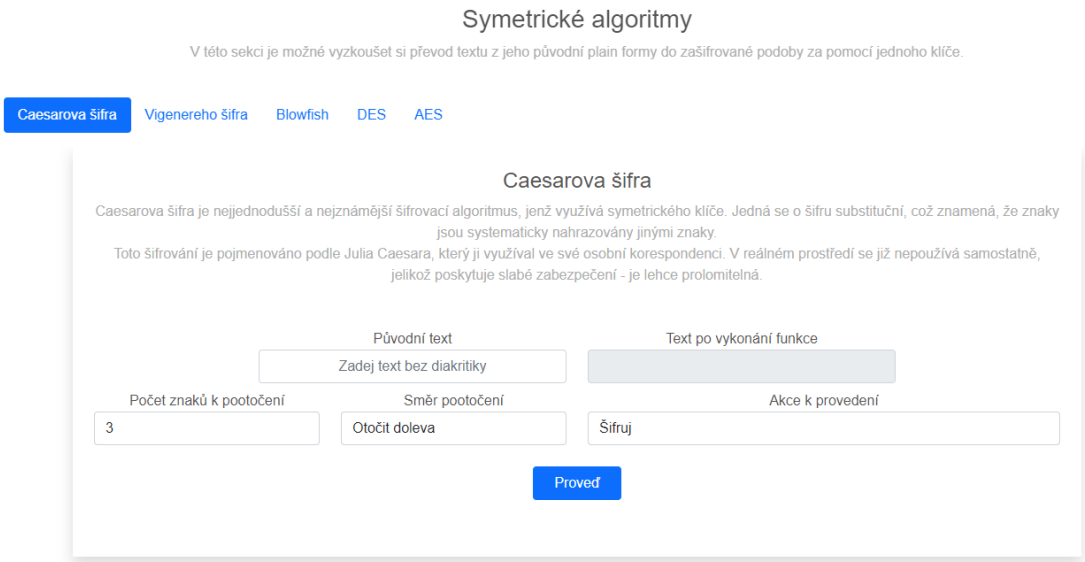
Tato sekce byla vytvořena s cílem informovat o teoretické stránce problematiky. Nachází se jako první viditelná sekce na webu, má nadpis „Kryptologie v kostce" a obsahuje rozklikávací menu s podtématy. První téma se zabývá kryptologií a souvisejícími vědními disciplínami. Další rozklikávací položky obsahují informace o rozdělení algoritmů na symetrické a asymetrické. Taktéž se zde nachází text o využití šifrování v reálném prostředí. Popsáno je také slovo protokol a jeho význam v kryptologii. Další dvě záložky se zabývají pojmem TLS/SSL a jejich certifikáty.



Obrázek 11 - UI – Úvodní informace (Zdeněk Hejzlar)

4.3.3.2 Algoritmy – symetrické

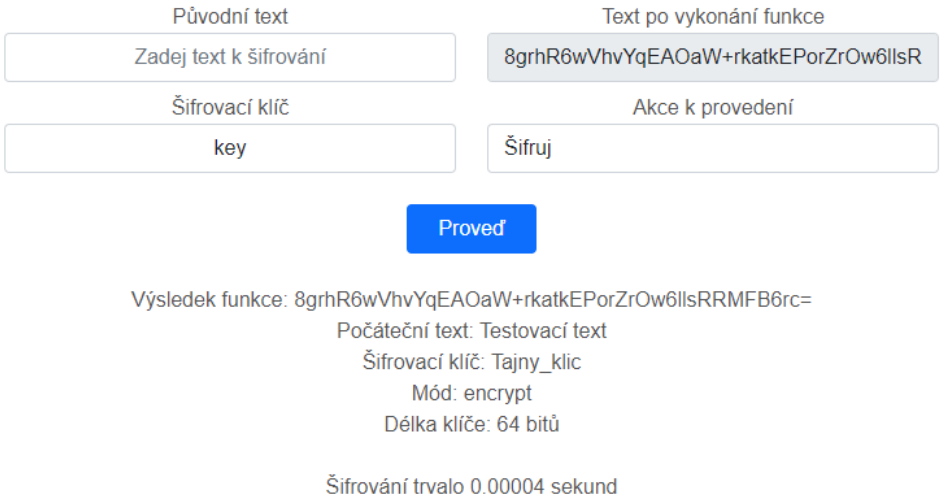
V části „Algoritmy – symetrické" se nachází implementace algoritmů, které využívají symetrických klíčů ke své funkci. Hlavní částí této sekce je menu, pomocí kterého lze proklikávat mezi jednotlivými algoritmy. Lze zde najít Caesarovu šifru, Vigenereho šifru, Blowfish, DES a AES. Z velké většiny se jedná o populární algoritmy, které se využívají a často zmiňují, a proto jsou i stručně popsány.



Obrázek 12 - UI – Symetrické šifrování (Zdeněk Hejzlar)

Pro vyzkoušení jednotlivých implementací se zde nachází formulář, který lze vyplnit a zaslat pomocí tlačítka „Proveď". Ve formuláři je možnost zadat původní text, specifikovat parametry daného algoritmu a zvolit, zda se jedná o šifrování či

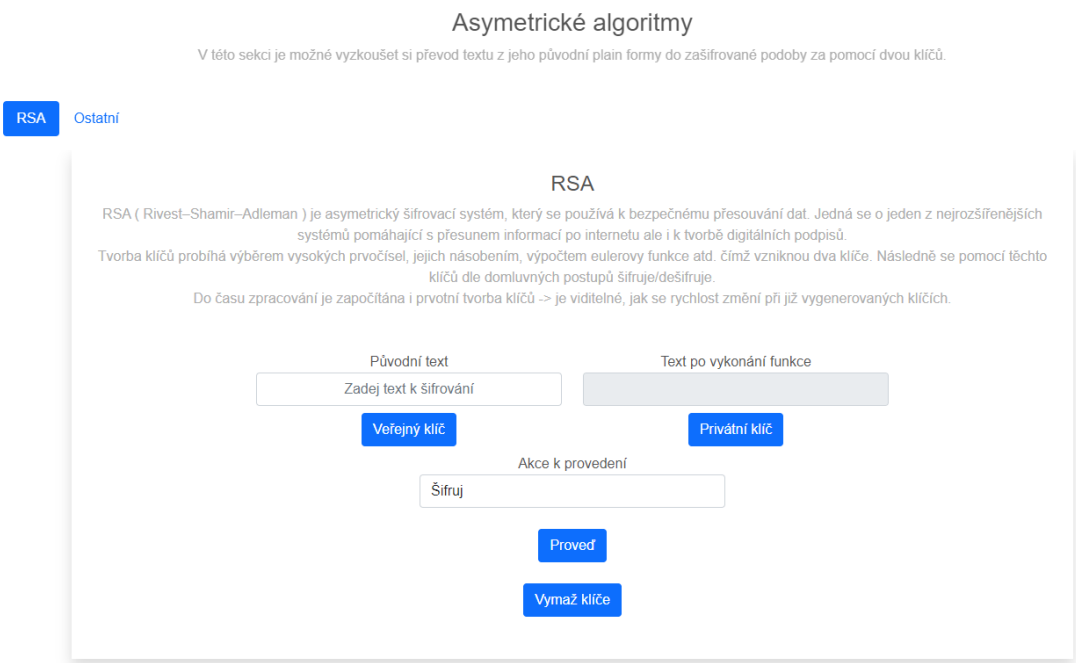
dešifrování. Po provedení se zobrazí finální výsledek v poli „Text po vykonání funkce“. Taktéž se ukáže část s doplňujícími informacemi.



Obrázek 13 - UI - DES doplňkové info (Zdeněk Hejzlar)

4.3.3.3 Algoritmy – asymetrické

Tato část aplikace je v mnohém totožná se sekcí symetrických algoritmů. Rozdílem je implementace jediného algoritmu (RSA). Ostatní algoritmy jsou využity k předávání klíčů nebo jejich implementace za daných podmínek byla příliš složitá. V menu se tedy nachází položka „RSA“ a „ostatní“, která popisuje, proč nejsou další algoritmy implementovány.



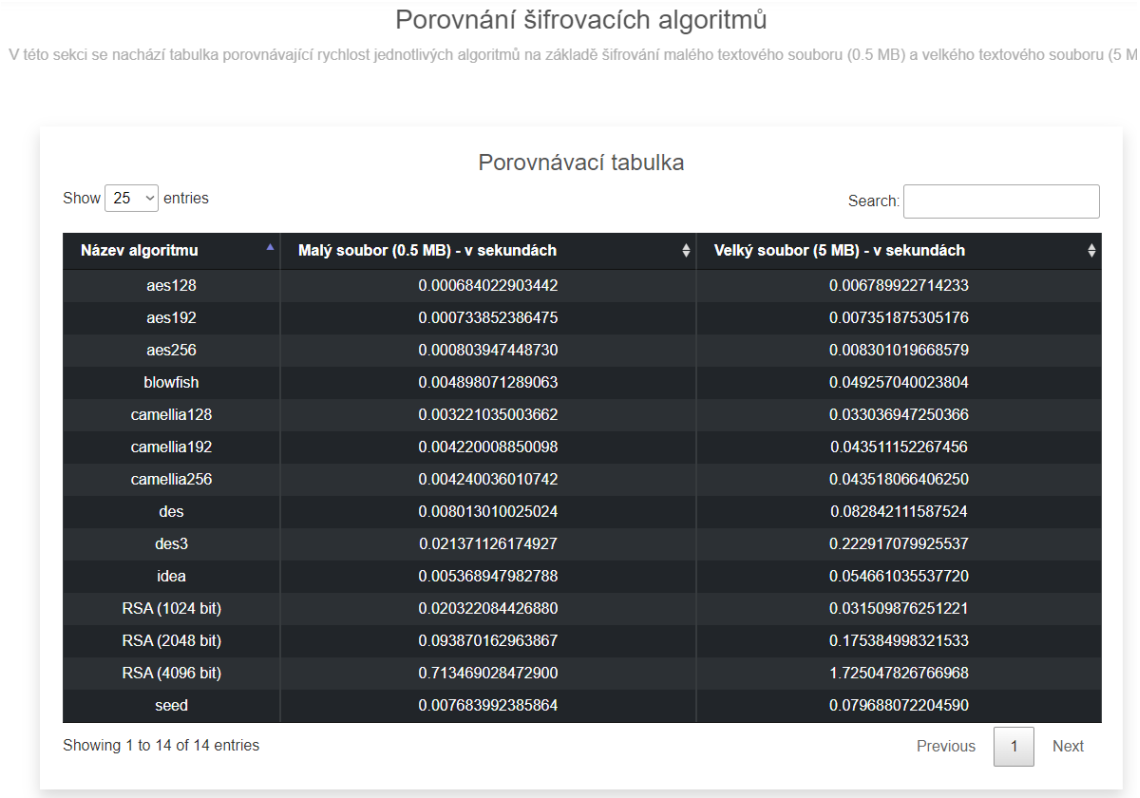
Obrázek 14 - UI – Asymetrické algoritmy (Zdeněk Hejzlar)

Formulář ke zkoušení šifrování je v případě RSA doplněn o několik tlačítek. Tlačítka s názvy „veřejný klíč“ a „privátní klíč“ zobrazí daný vygenerovaný klíč v podobě vyskakovacího okna. Tyto akce jsou přístupné až po prvním šifrování, před vytvořením ukazují tlačítka hlášku s pokynem k prvnímu zašifrování dat. Funkce „vymaž klíče“ dělá odpovídající akci a smaže vygenerované informace.

Obrázek 15 - UI – Veřejný klíč (Zdeněk Hejzlar)

4.3.3.4 Porovnání

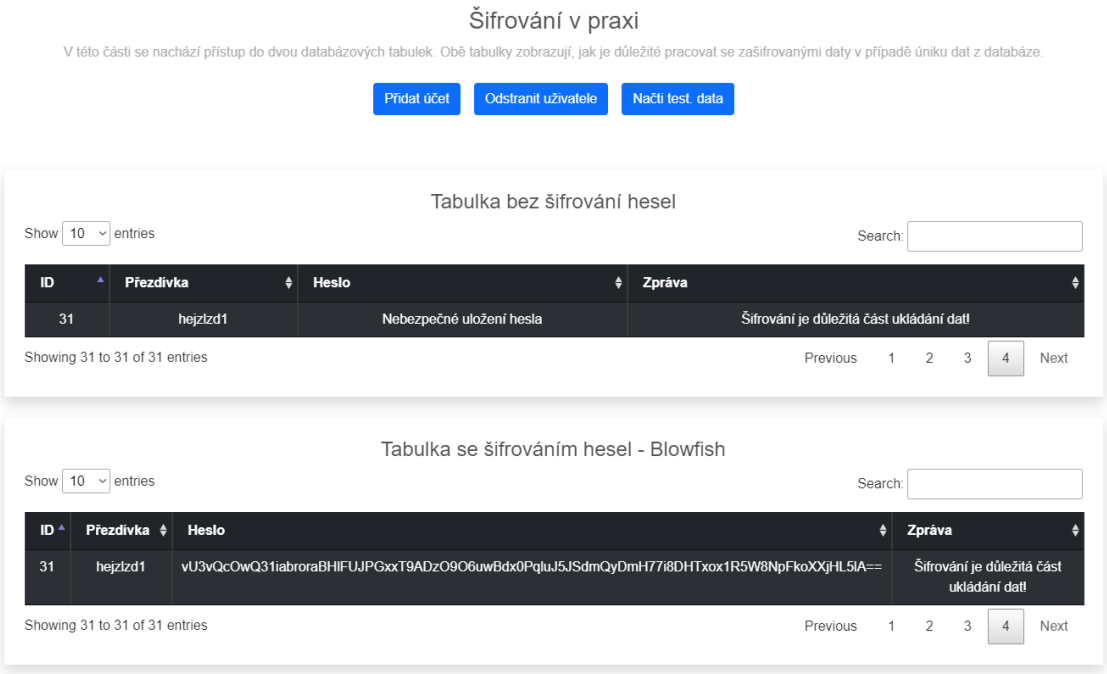
Aplikace porovnává vybrané populární algoritmy na základě času stráveného šifrováním předem nadefinovaných dat. Zdroje dat jsou dva, jeden o velikosti 0,5 MB a druhý 5 MB. Obsahem těchto souborů je text „lorem ipsum“ generovaný do dané velikosti. Srovnání je vizualizováno formou jednoduché tabulky obsahující název algoritmu, čas strávený šifrováním malého souboru a čas vynaložený u velkého souboru. Data lze vyhledávat dle hodnot či řadit dle libovolného sloupce.



Obrázek 16 - UI – Porovnávací tabulka (Zdeněk Hejzlar)

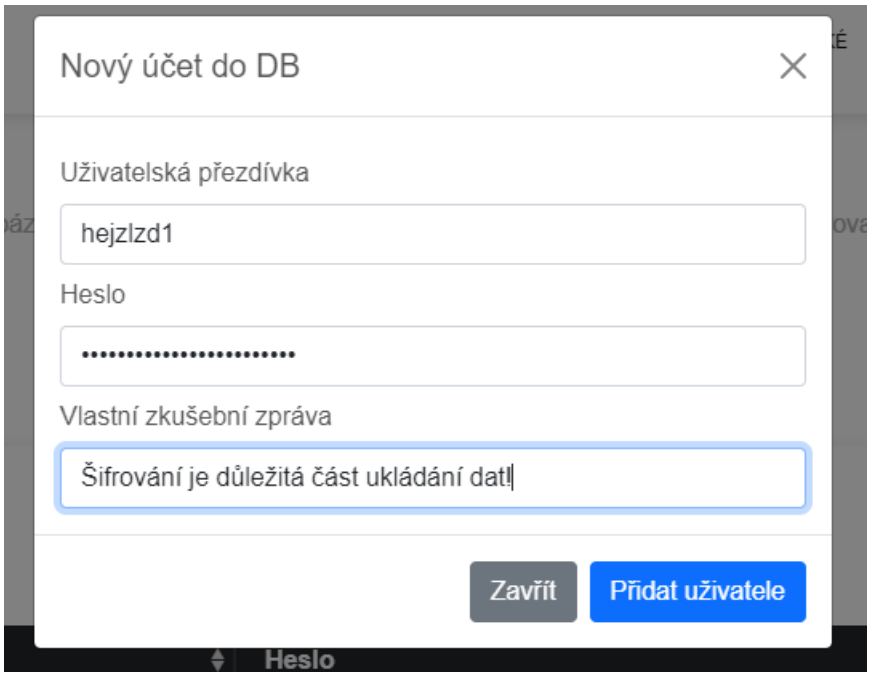
4.3.3.5 Šifrování v praxi

Poslední částí webu je ukázka, jak vypadá šifrování v reálném prostředí a případný únik dat. Pro tuto část je implementována databáze s předem nahranými položkami. Nachází se zde dvě tabulky, jedna obsahuje data v nezašifrované formě, druhá má data šifrovaná. Tabulka představuje uživatelské profily s vlastní zprávou. Tato část má také interaktivní prvky v podobě tlačítek, pomocí kterých lze s daty v tabulce manipulovat. Možnostmi jsou např. načtení původních dat, přidání uživatele či odstranění všech záznamů.

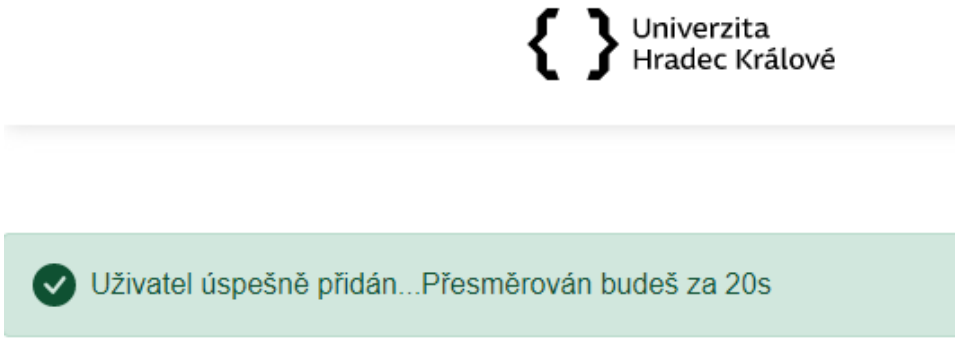


Obrázek 17 - UI – Tabulka šifrování (Zdeněk Hejzlar)

Pro správnou demonstraci je uživatelům umožněno vkládat záznamy do databáze. Tyto záznamy lze pak jednoduše smazat a nahrát původní zkušební data. Zadávání dat je zprostředkováno pomocí vyskakovacího okna s požadovaným formulářem. V případě úspěšného přidání či chyby je uživatel přesměrován na stránku s odpovídajícím upozorněním.



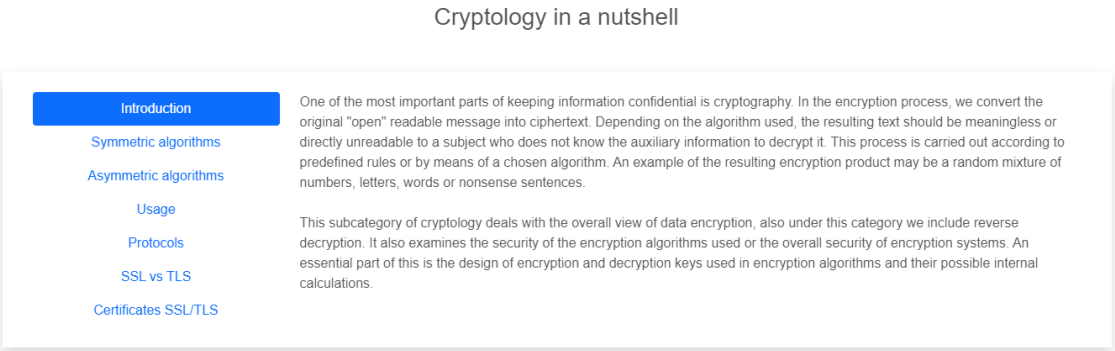
Obrázek 18 - UI – Nový uživatel (Zdeněk Hejzlar)



Obrázek 19 - UI – Info hláška (Zdeněk Hejzlar)

4.4 Anglická verze

Za účelem využití aplikace při výuce kryptografie byla přidána úplná lokalizace do anglického jazyka. Přeložená varianta webu se nachází v menu pod záložkou „Anglická verze“ či na adrese https://edu.uhk.cz/~hejzlzd1/en/.



Obrázek 20 - UI – Anglicky (Zdeněk Hejzlar)

4.5 Zdrojový kód

Zdrojový kód lze stáhnout a využít za pomoci verzovacího nástroje z veřejného online repositáře na adrese https://github.com/hejzlzd1/bachelor_crypto_app. Aplikace je optimalizovaná především pro běh na webovém serveru od UHK, pro vlastní využití je potřeba úprav a přidání souboru s nastavením databáze.

5 Analýza šifrovacích algoritmů

K vyhodnocení efektivnosti šifrovacích algoritmů je zapotřebí vzít na vědomí nejen bezpečnost, ale i rychlost šifrování a generace klíčů. V projektu jsou hodnoceny populární algoritmy pomocí tabulky rychlosti šifrování dvou odlišně velkých textových souborů.

U symetrických algoritmů je pevně stanovený klíč „secretkey“. Jsou zde zkoumány algoritmy s různou délkou daného klíče. O doplnění klíče do požadované délky se stará knihovna openssl, která daný řetězec kopíruje do stanovené délky dle bitů. Asymetrický algoritmus RSA má klíče generované dle využité délky.

Délka klíčů se pohybuje v hodnotách 128, 192, 256, 1024, 2048, 4096 bitů. Tento parametr stanovuje bezpečnost systému. Čím je větší bitová délka, tím více bezpečné je šifrování.

Vyhodnocované algoritmy jsou vybrané dle popularity a složitosti implementace. Vybrané algoritmy jsou AES, Blowfish, DES, DES3, Idea, Camellia, Seed a RSA.

Jednotlivé naměřené hodnoty jsou doplňovány do tabulky v sekci „Porovnání“, kde jsou řazeny abecedně.

6 Shrnutí výsledků

Tabulka níže zobrazuje rychlost algoritmů seřazenou abecedně. Ze získaných dat lze vyhodnotit, že větší soubor se šifruje u symetrických systémů značně déle než soubor menší. U RSA je časový interval hodně ovlivněn délkou klíčů a jejich generací. Z hlediska bezpečnosti a rychlosti jsou více využitelné pro větší objem dat. Oproti tomu symetrické algoritmy je vhodné využít u menšího množství dat.

Zapotřebí je taktéž započítat výkon serveru, který dané služby poskytuje. Na odlišném hardwaru bude možné dosáhnout lepších či horších hodnot v závislosti na přiřazených zdrojích.

Název algoritmu	Malý soubor (0.5 MB) - v sekundách	Velký soubor (5 MB) - v sekundách
aes128	0.000921010971069	0.007346868515015
aes192	0.000781059265137	0.007174015045166
aes256	0.000766992568970	0.007941961288452
blowfish	0.004724025726318	0.048825979232788
camellia128	0.003187894821167	0.032649040222168
camellia192	0.004174947738647	0.043208122253418
camellia256	0.004169940948486	0.043133974075317
des	0.007992029190063	0.082464218139648
des3	0.021390199661255	0.221854925155640
idea	0.005295991897583	0.054210901260376
RSA (1024 bit)	0.021179914474487	0.033765077590942
RSA (2048 bit)	0.163172006607056	0.245417833328247
RSA (4096 bit)	0.725915908813477	0.306478023529053
seed	0.007668018341064	0.079422950744629

Obrázek 21 - Tabulka porovnání systémů (Zdeněk Hejzlar)

6.1 Detailní porovnání rychlosti algoritmů

U symetrických algoritmů vítězí v rychlosti šifrování malého souboru algoritmus AES s délkou 192 bitů. V případě velkého souboru je rychlejší 128bitový AES.

Nejpomalejší je DES3 kvůli svému trojitému procházení šifrovacím algoritmem. Čas provedení je podstatně delší oproti původnímu DES.

Asymetrický RSA o délce 1024 bitů předhání některé symetrické algoritmy v šifrování velkých souborů.

Název algoritmu	Malý soubor (0.5 MB) - v sekundách	Velký soubor (5 MB) - v sekundách
aes192	0.000738859176636	0.007313013076782
aes256	0.000771999359131	0.008724927902222
aes128	0.000779151916504	0.006560087203979
camellia128	0.003204107284546	0.033578872680664
camellia192	0.004240036010742	0.044393062591553
blowfish	0.004791021347046	0.049319982528687
camellia256	0.005044937133789	0.043785095214844
idea	0.005636930465698	0.055706977844238
seed	0.007791996002197	0.081361055374146
des	0.007991075515747	0.083337068557739
RSA (1024 bit)	0.015698909759522	0.044568061828613
des3	0.021683931350708	0.226622104644775
RSA (2048 bit)	0.045547962188721	0.116168975830078
RSA (4096 bit)	2.208295106887817	0.660804986953735

Obrázek 22 - Porovnání – malé soubory (Zdeněk Hejzlar)

Název algoritmu	Malý soubor (0.5 MB) - v sekundách	Velký soubor (5 MB) - v sekundách
aes128	0.000779151916504	0.006560087203979
aes192	0.000738859176636	0.007313013076782
aes256	0.000771999359131	0.008724927902222
camellia128	0.003204107284546	0.033578872680664
camellia256	0.005044937133789	0.043785095214844
camellia192	0.004240036010742	0.044393062591553
RSA (1024 bit)	0.015698909759522	0.044568061828613
blowfish	0.004791021347046	0.049319982528687
idea	0.005636930465698	0.055706977844238
seed	0.007791996002197	0.081361055374146
des	0.007991075515747	0.083337068557739
RSA (2048 bit)	0.045547962188721	0.116168975830078
des3	0.021683931350708	0.226622104644775
RSA (4096 bit)	2.208295106887817	0.660804986953735

Obrázek 23 - Porovnání – velké soubory (Zdeněk Hejzlar)

7 Závěry a doporučení

Výsledky porovnání algoritmů značně ovlivnil výkon poskytnutého serveru. K dosažení lepších a přesnějších výsledků by bylo zapotřebí získat data z více serverů s odlišným hardwarem.

Data získaná testováním rychlosti nasvědčují, že většina symetrických algoritmů je značně rychlejší v šifrování menších souborů. V převodu větších souborů je pak výhodnější využít asymetrického algoritmu o správné délce klíče.

V kategorii zabezpečení jasně vítězí asymetrické systémy, jelikož je lze využít na síťové vrstvě pro bezpečnou komunikaci. Taktéž mají funkce pro bezpečné generování klíčů a jejich přenos mezi uživateli. Symetrické systémy jsou doporučeny v případě, že data ukládáme na lokální úrovni a přístup k nim není sdílen mezi více uživateli. Bezpečnost lze jednoznačně ovlivnit délkou generovaných klíčů. Mezi nejbezpečnější algoritmy spadají především AES a RSA.

Další studium by se mohlo zabývat detailnějším rozborem jednotlivých algoritmů, generací klíčů, prolamováním šifrovaného textu a detailním zkoumáním vlivu délky klíče na rychlost generace. Do projektu by také bylo vhodné přidat dostatečné množství dalších funkcí a v závislosti pak zvolit vhodný framework a aplikaci do něj převést. Také by mohlo být směřováno k optimalizaci a přeformátování současného kódu do efektivnější podoby.

8 Seznam použité literatury

[1] „Kryptografie_okolo_nas.pdf". Viděno: 3. duben 2022. [Online]. Dostupné z: https://knihy.nic.cz/files/edice/Kryptografie_okolo_nas.pdf

[2] O. Bitto, „HISTORIE KRYPTOLOGIE". https://www.fi.muni.cz/usr/jkucera/pv109/2003/xbitto.htm (viděno 4. duben 2022).

[3] C. Paar a J. Pelzl, *Understanding Cryptography: A Textbook for Students and Practitioners*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010. doi: 10.1007/978-3-642-04101-3.

[4] M. Oulehla a R. Jasek, *Modern?? kryptografie*. 2017.

[5] P. Pathak, „HISTORY OF MORSE CODE", *International Journal of Research (IJR)*, 15. červenec 2020. https://internationaljournalofresearch.com/2020/07/15/history-of-morse-code-2/ (viděno 19. duben 2022).

[6] T. Gage, *Enigma Machine (Bletchley Park)*. 2006. Viděno: 19. duben 2022. [Photo]. Dostupné z: https://www.flickr.com/photos/timg_vancouver/200625463/

[7] M. Kajínek, „Tajemství šifer – po stopách kryptografie a steganografie I", 19. červen 208n. l. http://www2.epochtimes.cz/200806195364/Tajemstvi-sifer-po-stopach-kryptografie-a-steganografie-I.html (viděno 21. říjen 2021).

[8] DMGualtieri, *English: A scytale, a physical mechanism for encryption and decryption of a transposition cipher. This example shows the plaintext rendering of „Mary had a little lamb..."* 2012. Viděno: 19. duben 2022. [Online]. Dostupné z: https://commons.wikimedia.org/wiki/File:Scytale.png

[9] O. Ibrahim, „A3: Encryption Machine", *CSE 142*, 27. srpen 2021. https://courses.cs.washington.edu/courses/cse142/21su/assessments/a3/ (viděno 19. duben 2022).

UNIVERZITA HRADEC KRÁLOVÉ
Fakulta informatiky a managementu
Akademický rok: 2020/2021

Studijní program: Aplikovaná informatika
Forma studia: Prezenční
Obor/kombinace: Aplikovaná informatika (ai3-p)

Podklad pro zadání BAKALÁŘSKÉ práce studenta

Jméno a příjmení:	**Zdeněk Hejzlar**
Osobní číslo:	**I1900182**
Adresa:	Sedlice 64, Sedlice, 39601 Humpolec, Česká republika
Téma práce:	Zabezpečení TLS
Téma práce anglicky:	Transport Layer Security
Vedoucí práce:	Ing. Hana Švecová Katedra informačních technologií

Zásady pro vypracování:

Zásady pro vypracování Téma: Kryptografické algoritmy ve výukové aplikaci **Postup pro vypracování:**

- Analýza kryptografických algoritmů
- Sestavení seznamu výhod/nevýhod daných algoritmů
- Návrh realizace aplikace
- Implementace algoritmů ve výukové aplikaci
- Otestování algoritmů a odvození efektivity na základě dostupných informací
- Závěr

Seznam doporučené literatury:

1. J. R. Vacca, Ed., *Computer and information security handbook*, Third edition. Cambridge, MA: Morgan Kaufmann Publishers, an imprint of Elsevier, 2017
2. M. Oulehla a R. Jašek, *Moderní kryptografie*. 2017
3. M. Novák, V. Šebesta, Z. Votruba, České vysoké učení technické v Praze, a Dopravní fakulta, *Bezpečnost a spolehlivost systémů*. Praha: Vydavatelství ČVUT, 2003

Podpis studenta: Datum:

Podpis vedoucího práce: Datum: