

Metody zabezpečení dat přenášených pomocí elektronických sítí

Methods of data security transferred by electronic networks

Ondřej Píža

Bakalářská práce
2011



Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky
akademický rok: 2010/2011

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: Ondřej PÍŽA
Osobní číslo: A08252
Studijní program: B 3902 Inženýrská informatika
Studijní obor: Informační a řídicí technologie

Téma práce: **Metody zabezpečení dat přenášených pomocí elektronických sítí**

Zásady pro vypracování:

1. Vypracujte literární rešerši na dané téma.
2. Proveďte teoretický rozbor jednotlivých metod včetně statistik četnosti využívání pro firemní sektory.
3. Vypracujte ukázkové příklady.
4. Zhodnoťte efektivitu jednotlivých postupů v praxi.

Rozsah bakalářské práce:

Rozsah příloh:

Forma zpracování bakalářské práce: tištěná/elektronická

Seznam odborné literatury:

1. ZELENKA, J., ČAPEK, J., FRANCEK, J., JANÁKOVÁ, H. Ochrana dat, Kryptologie. Gaudeamus, září 2003. 171 s. SBN 80-7041-737-4.
2. ČANDÍK, Marek. Základy informační bezpečnosti. vyd. Zlín : Univerzita Tomáše Bati, 2004. 107 s. ISBN 8073182181.
3. DOSTÁLEK, L., VOHNOUTOVÁ, M., KNOTEK, M. Velký průvodce infrastrukturou PKI a technologií elektronického podpisu. Computer Press, a.s, 2009. ISBN 978-80-251-2619-6.
4. VONDRUŠKA, P. Kryptologie, šifrování a tajná písma. Albatros, 2006. ISBN 80-00-01888-8.
5. KATZ, Jonathan. Introduction to Modern Cryptography: Principles and Protocols. Chapman & Hall, 1 edition. 2007. 552 s. ISBN 978-1584885511.
6. MURPHY, Sean. Kryptografie - Průvodce pro každého. Dokořán, 2006. 157 s. ISBN 80-7363-074-5.
7. BITTO, O. Šifrování a biometrika. BEN, 2005. 168 s. ISBN 80-86686-48-5.
8. KLÍMA, Vlastimil; ROSA, Tomáš. Kryptologie pro praxi ? DSA, ECDSA. Dostupné z WWW: [http://crypto-world.info/klima/2004/st_2004_04_21_21.pdf].

Vedoucí bakalářské práce:

Ing. Roman Šenkeřík, Ph.D.

Ústav informatiky a umělé inteligence

Datum zadání bakalářské práce:

25. února 2011

Termín odevzdání bakalářské práce:

7. června 2011

Ve Zlíně dne 25. února 2011

prof. Ing. Vladimír Vašek, CSc.
děkan



prof. Ing. Vladimír Vašek, CSc.
ředitel ústavu

ABSTRAKT

Tato bakalářská práce je zaměřena na Metody zabezpečení dat přenášených pomocí elektronických sítí, tedy postupy a metody, jak zabránit získání citlivých dat nepovolaným subjektům.

Práce je členěna na dva hlavní celky. V teoretické části je rozebrán historický vývoj kryptologie od dob nejstarších po moderní, dále jsou zde teoreticky vysvětleny nejznámější kryptografické metody moderní doby a postupy, jakými se aplikují v praxi. Praktická část pak obsahuje popis vypracované ukázkové aplikace z hlediska programátora a dále je zde vysvětleno uživatelské rozhraní aplikace.

Klíčová slova:

Kryptografie, kryptoanalýza, hashovací funkce, symetrická šifra, asymetrická šifra, ASP.NET, SSH, SSL

ABSTRACT

This bachelor thesis is focused on security methods for data transferred by means of electronic networks, ie the procedures and methods to prevent acquisition of sensitive data to unauthorized parties.

The work is divided into two main parts. The theoretical part is aimed on analyzis of the historical development of cryptology from the ancient to modern times. Also the theoretical explanation of the most famous modern cryptographic methods and procedures, which are applied in practice is present there. The practical part includes a description of a sample application developed from the point of view of programmer and also the explanation of the user interface.

Keywords:

Cryptography, cryptanalysis, hash function, symmetric cipher, asymmetric cipher, ASP.NET, SSH,SSL

PODĚKOVÁNÍ

Děkuji vedoucímu bakalářské práce Ing. Romanu Šenkeříkovi Ph.D., za velmi užitečnou metodickou pomoc a cenné rady při zpracování bakalářské práce.

Prohlašuji, že

- beru na vědomí, že odevzdáním bakalářské práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb. o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších právních předpisů, bez ohledu na výsledek obhajoby;
- beru na vědomí, že bakalářská práce bude uložena v elektronické podobě v univerzitním informačním systému dostupná k prezenčnímu nahlédnutí, že jeden výtisk bakalářské práce bude uložen v příruční knihovně Fakulty aplikované informatiky Univerzity Tomáše Bati ve Zlíně a jeden výtisk bude uložen u vedoucího práce;
- byl/a jsem seznámen/a s tím, že na moji bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších právních předpisů, zejm. § 35 odst. 3;
- beru na vědomí, že podle § 60 odst. 1 autorského zákona má UTB ve Zlíně právo na uzavření licenční smlouvy o užití školního díla v rozsahu § 12 odst. 4 autorského zákona;
- beru na vědomí, že podle § 60 odst. 2 a 3 autorského zákona mohu užít své dílo – bakalářskou práci nebo poskytnout licenci k jejímu využití jen s předchozím písemným souhlasem Univerzity Tomáše Bati ve Zlíně, která je oprávněna v takovém případě ode mne požadovat přiměřený příspěvek na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše);
- beru na vědomí, že pokud bylo k vypracování bakalářské práce využito softwaru poskytnutého Univerzitou Tomáše Bati ve Zlíně nebo jinými subjekty pouze ke studijním a výzkumným účelům (tedy pouze k nekomerčnímu využití), nelze výsledky bakalářské práce využít ke komerčním účelům;
- beru na vědomí, že pokud je výstupem bakalářské práce jakýkoliv softwarový produkt, považují se za součást práce rovněž i zdrojové kódy, popř. soubory, ze kterých se projekt skládá. Neodevzdání této součásti může být důvodem k neobhájení práce.

Prohlašuji,

- že jsem na bakalářské práci pracoval samostatně a použitou literaturu jsem citoval. V případě publikace výsledků budu uveden jako spoluautor.
- že odevzdaná verze bakalářské práce a verze elektronická nahraná do IS/STAG jsou totožné.

Ve Zlíně

.....
podpis diplomanta

OBSAH

ÚVOD	9
I TEORETICKÁ ČÁST	10
1 HISTORIE VÝVOJE KRYPTOLOGIE	11
1.1 KLASICKÁ KRYPTOGRAFIE.....	11
1.2 STŘEDOVĚKÁ KRYPTOGRAFIE.....	11
1.2.1 Arabské země	11
1.2.2 Evropa.....	11
1.3 KRYPTOGRAFIE DO 2. SVĚTOVÉ VÁLKY	12
1.3.1 Vernamova šifra	12
1.4 MODERNÍ DOBA (OBDOBÍ STUDENÉ VÁLKY)	12
2 ROZDĚLENÍ A POPIS KRYPTOGRAFICKÝCH ŠIFER	13
2.1 SLOVNÍK VÝRAZŮ	13
2.2 MNOŽSTVÍ INFORMACE, ENTROPIE.....	13
2.3 REDUNDANCE (NADBYTEČNOST) JAZYKA, JAZYKOVÝ POMĚR	14
2.4 DĚLENÍ ŠIFER.....	15
2.5 VOLBA A SKRYTÍ KLÍČŮ	16
2.6 OBTÍŽNOST ŘEŠENÍ PROBLÉMU	16
2.7 HODNOCENÍ A POROVNÁNÍ KRYPTOSYSTÉMŮ	17
2.8 POUŽÍVANÉ ÚTOKY NA KRYPTOSYSTÉMY	17
3 MODERNÍ ŠIFRY	18
3.1 ASYMETRICKÉ ŠIFRY	18
3.1.1 RSA šifra	18
3.1.2 DSA (Digital Singature Algorithm)	19
3.2 SYMETRICKÉ ŠIFRY	21
3.2.1 Vernamova šifra	21
3.2.2 Feistelova Struktura (sítě).....	21
3.2.3 Šifra DES, 3DES	21
3.2.4 AES šifra	22
3.3 HASHOVACÍ FUNKCE	22
3.3.1 Hash algoritmus MD5	22
3.3.2 SHA-X algoritmy	23
4 SOUČASNÉ METODY PŘENOSU DAT V PRAKTICKÉM UŽITÍ	24
4.1 ÚVOD DO TERMINOLOGIE SSH.....	24
4.1.1 Postup autentizace	24
4.1.2 Nevýhody ssh	25
4.1.3 SFTP, SCP.....	25
4.2 SSL25	
4.2.1 Autentizace v SSL.....	26
4.2.2 Výhody a nevýhody SSL.....	26
II PRAKTICKÁ ČÁST	27
5 ŠIFROVACÍ APLIKACE	28

5.1	PLATFORMA.....	28
5.2	TESTOVACÍ NÁSTROJ CRYPTO TOOL 2.0.....	28
5.3	VÝBĚR ŠIFROVACÍCH METOD.....	29
5.4	POPIS APLIKACE.....	30
5.4.1	Webové šifrování.....	31
5.4.2	Rozhraní pro šifrování souborů.....	34
5.5	MOŽNOSTI BUDOUCÍHO VÝVOJE.....	35
5.6	EFEKTIVITA JEDNOTLIVÝCH METOD.....	36
	ZÁVĚR.....	38
	ZÁVĚR V ANGLIČTINĚ.....	39
	SEZNAM POUŽITÉ LITERATURY.....	40
	SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK.....	41
	SEZNAM OBRÁZKŮ.....	42
	SEZNAM TABULEK.....	43
	SEZNAM PŘÍLOH.....	44

ÚVOD

Už od dob starověkého Říma se začaly objevovat první náznaky určitého druhu myšlení, které se snažilo o utajení komunikace a skrytí důležitých informací, ať již se jednalo o vojenské informace (v této době byly vojenské a politické informace úzce spjaty), či později i o informace čistě politické a v moderní době taktéž obchodního charakteru.

Až 60. léta minulého století přinesla zlom v tomto oboru. Díky vzniku projektu ARPANET (později Internet), který začal s myšlenkou propojování jednotlivých počítačů nezávislou elektronickou sítí, jež by byla přístupná prakticky odkudkoliv, bylo nutné se začít také zabývat bezpečností dat, která v dnešní době často představují strategicky důležitá data, ať již z firemního nebo armádního či státního sektoru. Vzhledem k povaze takových dat vznikl zcela nový vědní obor, který se začal zabývat metodami, jak zabránit nepovolaným osobám k přístupu k podobným údajům.

Do této chvíle nebylo možné získat jakákoliv data bez fyzické přítomnosti osoby u zdroje nebo úložiště dat. S Internetem nicméně toto omezení velmi rychle padlo a začaly se objevovat případy útoků na důležitá datová centra.

Na internetu již nějakou dobu kolují škodlivé programy jako viry, červy, spyware, malware a další podobné nežádoucí programy. Jejich proniknutím do systému může dojít ke zničení veškerých dat.

V první kapitole je rozebrán historický vývoj kryptografických metod, od nejstarších (a tedy nejjednodušších) po moderní (nejsložitější).

Druhá kapitola popisuje teoretický poklad pro návrh šifry, dále obsahuje základní rozdělení šifer a taktéž jsou zde zmíněny metody užívané pro prolomení šifer a hodnocení bezpečnosti šifry.

Třetí kapitola podrobně popisuje funkci a výhody či nevýhody nejpoužívanějších šifer současné doby.

Čtvrtá kapitola se zabývá využitím těchto šifrovacích metod v praxi, tedy uvádí některé komunikační protokoly pro zabezpečený přenos souborů.

Pátá kapitola se celá věnuje praktické části této práce, tedy ukázkové aplikaci. Je zde vysvětlení jejího použití, její funkce, způsobu ovládání a teoretický rozbor návrhu, ať již jde o výběr platformy nebo určení omezení pro tento program.

I. TEORETICKÁ ČÁST

1 HISTORIE VÝVOJE KRYPTOLOGIE

Kryptologie, neboli souhrn postupů sloužící k ukrytí či utajení zprávy se v průběhu let vyvíjela nejrůznějšími směry tak, aby se snažila vycházet z dobových možností a poznatků. Proto hlavně v dávných dobách existovala pouze část této komplexní vědy, **kryptografie**, čili určitý postup, který převádí srozumitelnou zprávu do nesrozumitelné podoby a zpět. Nicméně součástí této vědy je taktéž **kryptoanalýza**, která slouží k prolomení šifer a k přečtení tajné zprávy.

1.1 Klasická kryptografie

Jedná o první skutečné šifry, které byly využity pro ukrytí důležitých znalostí a dat. Byly nalezeny záznamy u některých vyspělých kultur. Např. v římské říši asi nejznámější Caesarova šifra, která spadá mezi jednoduché substituční šifry. Dále Spartská transpoziční šifra anebo hebrejská substituční šifra Atbash.

- **substituční šifra** - je založena na záměně jedné skupiny symbolů za jinou skupinu symbolů, např. Polybiův čtverec
- **transpoziční šifra** - pouze přehodí pořadí znaků podle nějakého pravidla, např. sloupcová transpozice s úplnou tabulkou

1.2 Středověká kryptografie

1.2.1 Arabské země

Ve středověku, v arabských zemích, došlo k pravděpodobně nejdůležitějšímu objevu tehdejší doby v oblasti kryptoanalýzy. Jedná se o vynález frekvenční analýzy. Všechny dosud známé šifry byly s pomocí této metody v podstatě snadno prolomitelné.

1.2.2 Evropa

Tuto metodu překonal až Leon Alberti, který vynalezl polyalfabetickou šifru, tedy namísto jedné abecedy použil dvě. Každé písmeno tak bylo najednou zašifrováno dvakrát. Ačkoliv je historicky právě Alberti uznáván jako vynálezce této šifry, jsou dochovány záznamy, že Arabové znaly a používaly polyalfabetické šifry již 500 let před Albertim.

1.3 Kryptografie do 2. světové války

I přes svou neskutečně dlouhou dobu vývoje a používání, stále byly metody kryptografie stále vyvíjeny metodou pokusů a omylů. Edgar Alan Poe začal používat systematické metody k prolamování šifer už ve 40. letech 19. století a svou schopností prolomit prakticky jakýkoliv zašifrovaný text doslova šokoval veřejnost. Jeho postupy a metody pak posloužili britské rozvědné službě během první světové války k rozluštění německých kódů.

Asi nejvýznamnější objev této doby navrhl Gilbert Vernam, který vynalezl tzv. Vernamovu šifru, která pracuje s klíčem, který se stejně dlouhý jako šifrovaná zpráva, takže nedochází k opakování znaků. U této šifry jako jediné byl proveden důkaz, že je neprolomitelná.

1.3.1 Vernamova šifra

Vernamova šifra, jinými slovy jednorázová tabulková šifra, je založena na posunu každého písmene v šifrované zprávě o náhodný počet míst v abecedě. Tím se tedy dané písmeno nahradí jiným, které je zcela náhodné, z čehož plyne právě její neprolomitelnost.

1.4 Moderní doba (období studené války)

Současná situace ve světě potřebuje co nejdokonalejší utajení veškerých informací, kdy jednotlivé subjekty žádají téměř neprolomitelnou ochranu informací (téměř protože neexistuje stoprocentně spolehlivá a dokonalá ochrana).

Nicméně v době od druhé světové války (tato doba též nazývaná jako studená válka) začaly stále větší vliv získávat zpravodajské agentury jednotlivých zemí. Od počátečního boje dvou velmocí, jakou byly Spojené státy Americké a jim přidružené země a země Varšavské smlouvy, se postupně začaly angažovat i ostatní země.

Většina úspěchů těchto tajných služeb je neznámá a tajná a některé se podařilo "odhalit" až po delší době. Z těch nejznámějších můžeme jmenovat např. událost během egyptsko-izraelské války, kdy se podařilo izraelským tajným službám uvést protivníka v chaos vysláním falešných rozkazů, zatímco jejich vlastní jednotky mezitím začali postupovat.

V současné době (asi od 90. let 20. století) se začala stále více rozvíjet kvantová kryptografie, kterou již začíná postupně vytlačovat šifrování založené na teorii chaosu, fraktální šifrování a AI.

2 ROZDĚLENÍ A POPIS KRYPTOGRAFICKÝCH ŠIFER

Na úvod samotného rozdělení by bylo vhodné si definovat určitý soubor výrazů, které budu nadále v práci používat. Kryptologie je v tomhle velmi uvolněná a některé slova mohou mít více významů.

2.1 Slovník výrazů

kryptografie - věda o tvorbě šifer

kryptoanalýza - věda o prolamování šifer

kryptologie - souhrnný název pro kryptografii a kryptoanalýzu

otevřený (plain) text - původní (originální) nezašifrovaný text

zašifrovaný (cipher) text - zašifrovaný (skrytý) text

šifrování - proces přeměny otevřeného textu na zašifrovaný

dešifrování - proces přeměny zašifrovaného textu na otevřený

šifrovací algoritmus - popis postupu šifrování

dešifrovací algoritmus - popis postupu dešifrování

šifra - popis postupu umožňující šifrování a dešifrování

protokol - sada pravidel a postupů pro výměnu dat mezi více stranami

kryptosystém - systém umožňující šifrování a dešifrování

2.2 Množství informace, entropie

Pro další vysvětlení je důležité si definovat, co vlastně představuje množství informace. Tedy pro ilustraci si představme, že každý význam zprávy Z je stejně pravděpodobný. Potom množství informace definujeme jako počet bitů nutných pro zakódování všech možných významů této zprávy.

Toto množství zprávy se popisuje jako entropie. Značíme ji $H(Z)$. Pro tuto veličinu platí vztah $H(Z) = \log_2 n$, kde n je počet stavů.

Jinak řečeno, entropie je střední hodnota jednoho zakódovaného znaku. Entropie taktéž souvisí se sekvencí náhodných čísel, kdy absolutně náhodná čísla by měla maximální míru entropie.

2.3 Redundance (nadbytečnost) jazyka, jazykový poměr

Z předchozího odstavce se dá pochopit, že naše jazyky často obsahují více dat než skutečných informací (protože ne všechny kombinace hlásek dávají smysl a mají využití). Tedy naše jazyky jsou silně redundantní. V informatice se redundance naopak uměle využívá pro odhalování chyb (např. samoopravný kód).

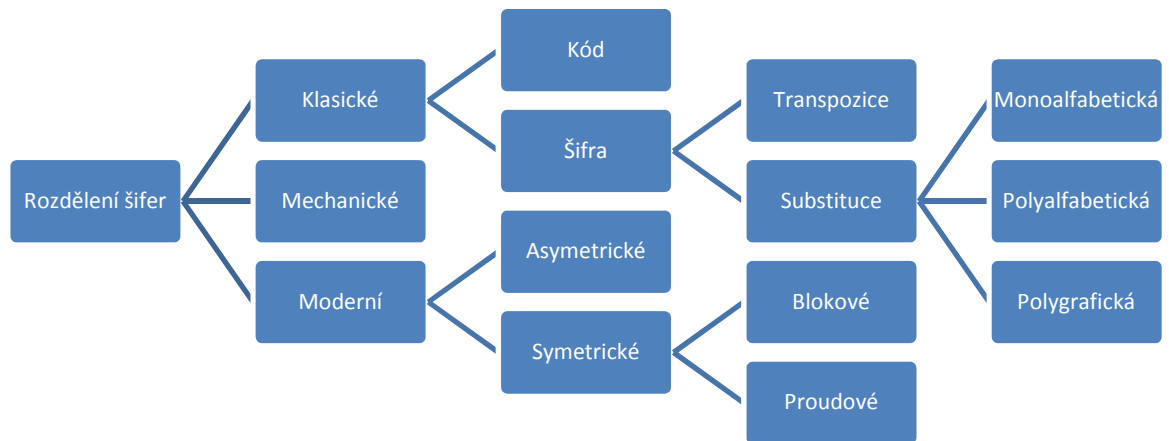
Mějme zprávu o velikosti N (N značí počet znaků zprávy). Poměr jazyka je pak definován jako $r = \frac{H(Z)}{N}$ a její jednotkou je *btc*, nebo-li bit per char. Z tohoto vztahu je tedy jasné patrné, že poměr jazyka nezáleží jen na volbě jazyka, ale i na délce zprávy.

- **Absolutní poměr jazyka** - je definován $R = \log_n L$, kde L značí počet znaků v abecedě.
- **Redundance** - $D = R - r$, je tedy udávána v bpc
- **Poměrná nadbytečnost** - $P = \frac{D}{R}$, např. anglický jazyk má kolem 72% nadbytečných znaků

Jak již bylo řečeno, redundance je vlastnost, která nahrává spíše útočící straně. Efekt a vliv redundance lze snížit použitím některými technikami.

- **komprese** - urychluje šifrování
- **zmatení** (confusion) - opakování substituce
- **rozptýlení** (diffusion) - transpozice

2.4 Dělení šifer



Obrázek 1 Rozdělení šifer

Vzhledem k tomu, že se tato práce má zabývat výhradně moderními metodami šifrování, nebude se již nadále zmiňovat o ostatních šifrách, které jsou součástí Obrázku 1, ale spadají do jiné kategorie než moderní.

Současné šifry se dělí na základě některých parametrů, podle kterých pracují. Jde především o množství dat, které jsou schopny najednou zpracovat (blokové a proudové šifry) a dále podle toho, zda musí napřed obě strany sdílet určitá "tajemství" před zahájením utajené komunikace (symetrické a asymetrické).

- **blokové šifry** - pracují s celými bloky dat, většinou v rozmezí 8 - 128 B
- **proudové šifry** - pracují s jednotlivými bity zprávy samostatně, obecně pomalejší a méně bezpečné
- **symetrické šifry** - odesílatel a příjemce sdílí jedno společné "tajemství" (klíč) nutné pro přečtení zprávy
- **asymetrické šifry** - odesílatel a příjemce šifrují zprávu různými klíči, tedy nemusí sdílet žádné klíče. Každá úloha, která jde realizovat pomocí symetrické kryptografie, lze provést i asymetrickém provedení. Cenou za tuto metodu je pomalejší šifrování (o několik řádů).

2.5 Volba a skrytí klíčů

Sebelepší algoritmus je k ničemu, pokud bude nevhodně zvolen klíč nebo jsou špatně uschovány. Např. pokud je za heslo zvoleno jméno uživatele nebo jeho rodné číslo, je velice pravděpodobné, že kdyby někdo chtěl z nějakého důvodu prolomit vytvořenou ochranu, bude úspěšný. Totéž platí o uložení takových hesel či klíčů.

Jednou ze zásad klíčů je jeho vhodná délka. Tato vlastnost je důležitá z důvodu budoucího rozšiřování o další aplikace, případně pokud se využívá dočasný klíč, tak aby nebylo možné využít duplicity již jednou použitého klíče.

2.6 Obtížnost řešení problému

V podstatě všechny šifrovací algoritmy používané v současnosti jsou založeny na určitém matematickém modelu (nebo tzv. matematickém problému), který musí úročník rozluštit, pakliže se chce dostat k původním datům. Složitost, celkový počet operací pro rozluštění a časová náročnost jednotlivých problémů jsou uvedeny v Tabulce 1.

- **Polynomiální problém** - k vyřešení tohoto problému je třeba provést $c * f(n)$ operací, kdy $f(n)$ je polynomiální funkce, c je vhodná konstanta a n je velikost vstupních dat.
- **NP - úplné problémy** - teorie k tomuto problému je složitá a dalece přesahuje rámec této práce. Pouze uvedu, že jsou výpočetně složitější a v kryptografii se využívají často.
- **Exponenciální problémy** - k jejich vyřešení je třeba provést $c^{f(n)}$ operací, kde $f(n)$ je polynomiální funkce a n je velikost vstupních dat.

Třída	Složitost	Počet operací ($n = 10^6$)	Doba zpracování při 10^6 op/s
Konstantní	$O(1)$	1	1 μ s
Lineární	$O(n)$	10^6	1 s
Kvadratická	$O(n^2)$	10^{12}	12 dní
Kubická	$O(n^3)$	10^{18}	32 000 let
Exponenciální	$O(2^n)$	10^{301030}	10^{301006} * stáří vesmíru

Tabulka 1 Časová náročnost jednotlivých operací

2.7 Hodnocení a porovnání kryptosystémů

Obecně lze systémy hodnotit a porovnávat na základě mnoha parametrů, jako je rychlost výpočtu, bezpečnost zašifrovaných dat nebo i na základě složitosti implementace celého systému. Všechny tyto pohledy jsou v podstatě rovnocenně důležité. V podstatě existují určitá pravidla, která by měla být při návrhu a výběru kryptosystému dodržována.

- Šifrování by nemělo navyšovat objem dat, které je nutné přenést.
- Implementace systému by měla být rozumně složitá.
- Taková implementace by měla být přiměřeně rychlá.
- Systém by neměl mít žádné omezení na vstupní data.
- V případě nějaké chyby v šifrování by se tato chyba neměla šířit.

Z hlediska bezpečnosti jsou šifry děleny na:

- **nepodmíněně bezpečné** - to jsou šifry, které nepodávají žádnou informaci o zprávě. Dosud jediná taková známá šifra je Vernamova šifra.
- **prolomitelné** - všechny současné systémy
 - **dokazatelně bezpečné** - k jejich prolomení je třeba vyřešit výpočetně složitý matematický model
 - **výpočetně bezpečné** - na jejich prolomení je třeba využít nesmyslně vysoký výpočetní výkon. Sem spadá většina moderních šifer
 - **bezpečné** - sem spadají šifry, k jejichž prolomení je třeba neúměrně drahý kryptoanalytický postup (neúměrně k ceně dat), anebo by kryptoanalytický postup neproběhl v krátkém čase.

2.8 Používané útoky na kryptosystémy

Z praxe jsou známy určité základní postupy, jak se pokusit o prolomení kryptosystému. Tyto postupy se dělí z hlediska znalosti útočníka na:

- **Cipher text only attack** - je nejobtížnější, používá se jen u slabých kryptosystémů
- **Known plain text attack** - ze znalosti plan textu a cipher textu se snaží zjistit tajemství
- **Chosen plan text attack** - na vstup jdou přiváděny určité zprávy a z výstupu se snaží zjistit slabiny a chování kryptosystému.

Snahou je vytvořit aplikaci, aby žádný kromě prvního způsobu nemohl být uplatněn.

3 MODERNÍ ŠIFRY

3.1 Asymetrické šifry

Jak již bylo zmíněno, u asymetrických šifer odpadá potřeba přenášet jakékoliv tajemství nebo klíč. Tím je docíleno poměrně vysoké bezpečnosti. Nevýhodou takových šifer je pomalejší šifrování, s tím jsou spojené vysoké nároky na výpočetní výkon (obzvláště, máme-li textu hodně nebo větší množství úloh), a nutnost dobře uschovat veřejný klíč. Velkou výhodou je fakt, že odpadá nutnost vymýšlet více klíčů při komunikaci s více stranami.

Při samotném šifrování se používá dvojice klíčů, konkrétně soukromý a veřejný. Veřejný klíč je k dispozici určitým lidem. Soukromý si uchovává osoba v tajnosti, přičemž s jeho pomocí je schopna dešifrovat zprávu zašifrovanou veřejným klíčem.

V této práci je největší prostor věnován šifře RSA, která patří k nejrozšířenějším v oblasti asymetrických šifer. U ostatních jsou zde uvedeny pouze základní vlastnosti. Bezpečnost těchto algoritmů je založena na vlastnosti některých matematických operací, u kterých se velmi těžko hledá invertní postup.

3.1.1 RSA šifra

Její název je odvozen ze jmen Rivest, Shamir, Adleman, kteří se podíleli na jejím vývoji. Jedná se o tzv. šifru s veřejným klíčem, tedy pro šifrování a dešifrování se používají odlišné klíče. Proto se tato šifra využívá u digitálního podpisu, protože lze jednoznačně identifikovat autora podpisu.

Matematický aparát (postup tvorby klíče)

Dva lidé spolu chtějí komunikovat prostřednictvím nezabezpečeného (otevřeného) kanálu. Martin chce poslat Petře zprávu. Petra si tedy musí napřed vyrobit soukromý a veřejný klíč.

1. Zvolí dvě velká náhodná různá prvočísla p a q .
2. Spočítá jejich součin $n = p * q$.
3. Spočítá se hodnota Eulerovy funkce $\varphi(n) = (p - 1) * (q - 1)$
4. Zvolí celé číslo e menší než $\varphi(n)$, které je s $\varphi(n)$ nesoudělné.
5. Nalezne se číslo d takové, aby platilo $d * e \equiv 1 \pmod{\varphi(n)}$
6. Pokud je e prvočíslo, pak $d = (1 + r * \varphi(n)) / e$, kde $r = (e - 1) * \varphi(n)^{e-2}$

Veřejným klíčem se pak rozumí dvojice (n, e) , kdy n představuje modul a e veřejný nebo šifrovací exponent. Tento klíč Petra pošle nezašifrovaně Martinovy.

Soukromým klíčem je pak dvojice (n, d) , kdy d je označován jako dešifrovací nebo soukromý exponent. Pomocí tohoto klíče si pak Petra může přečíst zprávu od Martina zašifrovanou poslaným veřejným klíčem.

Postup šifrování:

Jednoduchá matematická operace $c = m^e * \text{mod } n$, kde m je zpráva (plain text) a c je zašifrovaná zpráva.

Postup dešifrování:

Podobně jednoduchá matematická operace $m = c^d * \text{mod } n$

3.1.2 DSA (Digital Singature Algorithm)

DSA nebo-li algoritmus digitálního podpisu je standard americké vlády pro digitální podpis. Byl navržen americkým institutem NIST v roce 1991, od roku 1993 používán v protokolu DSS (Digital Signature Standart). Poslední úprava tohoto standartu byla v roce 2000, od kterého je veden jako FIPS 186-2.

Postup vytvoření klíčů:

- je proveden výběr kryptografické hashovací funkce (viz. dále), původně byla povinně SHA-1, nově je možnost využít i SHA-2
- rozhodne se o parametrech L a N , které určují délku klíče. Doporučení je používat tyto dvojice L a N : (1024,160);(2048,256);(3072,256)
- provede se výběr N -bitového prvočísla q . Velikost N musí být alespoň stejně velká jako výstup hashovací funkce.
- následně se provede výběr L -bitového čísla p tak, že $(p-1)$ je násobek q .
- nakonec se vybere číslo g takové, že odpovídá vzorci $g = h^{\frac{p-1}{q}}$, kdy číslo h se bere náhodně z intervalu $1 < h < p-1$, ale nejčastěji se volí $h=2$.
- všechny doposud vytvořené informace nejsou tajné a jsou sdíleny se všemi uživateli
- nyní se vybere náhodně číslo x splňující podmínku $0 < x < q$
- spočítá se $y = g^x \text{mod } p$
- veřejný klíč je pak soubor čísel (p, q, g, y) , soukromý klíč číslo x

Princip podepisování:

Označím si hashovací funkci jako H a zprávu jako z .

- vybere se náhodná hodnota k z intervalu $0 < k < q$
- spočítá se $r = (g^k \bmod p) \bmod q$
- dále $s = (k^{-1}(H(z) + x * r)) \bmod q$
- ve velice nepravděpodobném případě, kdy $r = 0$ nebo $s = 0$ se výpočet opakuje
- v opačném případě máme vytvořený digitální podpis (r, s)

Princip ověřování podpisu:

- pakliže neplatí tyto podmínky $0 < r < q$ a $0 < s < q$, je podpis automaticky odmítnut
- v opačném případě se spočítá $w = s^{-1} \bmod q$
- následně $u_1 = (H(z) * w) \bmod q$
- poté $u_2 = (r * w) \bmod q$
- a nakonec $v = ((g^{u_1} * y^{u_2}) \bmod p) \bmod q$
- podpis platí za podmínky $r = v$

3.2 Symetrické šifry

Výhodou těchto šifer je rychlost a jednoduchost. Nevýhodou je potom používání jednoho klíče po celou dobu komunikace, což může způsobovat značné potíže při komunikaci s více stranami.

Další velkým problémem může být nutnost nějakým způsobem předat onen tajný šifrovací klíč.

V základním principu se symetrické šifry dělí podle toho, zda pracují po jednotlivých bitech (proudové) anebo pracují s bloky celých dat (blokové). Blokované šifry se odlišují převážně stylem šifrování, kde se využívá rundové iterační funkce, kdy vstupem do této funkce je klíč a výstup z předchozí iterace.

3.2.1 Vernamova šifra

Vernamova šifra pochází z první poloviny 20. století. Je prakticky bezpečná, její jedinou nevýhodou je nepoužitelnost. Jelikož se ke zprávě pomocí operace XOR připojí klíč stejné délky jako zpráva sama, čímž se problém z přenesení zprávy přesune na problém přenesení klíče.

3.2.2 Feistelova Struktura (sítě)

Velká skupina používaných šifer. Nejedná se ani tak o algoritmy samotné, jako spíše o jejich strukturu. Během cyklu se několikrát data rozdělí na dvě poloviny, kdy je jedna polovina zpracována klíčem a druhá ne. Pak se v další rundě prochodí pořadí.

3.2.3 Šifra DES, 3DES

Původně vyvinuta v 60. letech 20. století. V současné době považována za nespolehlivou, hlavně díky bezpečnostním mezerám. Dále již bylo dokázáno, že je možné ji prolomit metodou brute force attack za méně než 24h, protože používá klíč o délce 56 bitů. V současnosti byla nahrazena šifrou AES.

DES šifruje data po blocích, kdy každý blok má 64 bitů. Algoritmus využívá dvou základních kryptografických technik, substituce a permutace. Princip DES je postaven na kombinaci těchto technik (na text je aplikována substituce a následně permutace). Tento proces je opakován šestnáctkrát. Kvůli prolomení šifry DES byla vynalezena šifra 3DES. Jedná se o stejný princip šifry, pouze je použita tato šifra třikrát na stejný text.

3.2.4 AES šifra

Byla to reakce na prolomení šifry DES. Využívá různých klíčů o délkách 128, 192, 256 bitů. Její nespornou výhodou je velmi vysoká rychlost a doposud stále platná neprolomitelnost.

Šifrování probíhá ve 4 krocích:

- SubBytes - jednoduchá substituce, každý bit je nahrazen jiným podle daného klíče, zajišťuje odolnost proti aritmetickým metodám
- ShiftRows - posun bitů v tabulce v cyklickém pořadí
- MixColumns - prohází sloupce a vynásobí je polynomem
- AddRoundKey - subklíč, který získá pomocí plánovacího algoritmu (stejná tabulka), smíchá s každým bitem zprávy

3.3 Hashovací funkce

Hashovací funkce představují takovou raritu v oblasti kryptografických funkcí. Už z požadavků na ně kladené je zřejmé, že se nejedná o žádné standardní funkce ani kombinace předchozích funkcí.

Po hashovacích funkcích se chce pevná délka výstupu při proměnném vstupu, malá změna na vstupu má vyvolat velkou změnu na výstupu a musí být co nejlépe invertibilní (tedy musí být téměř nulová šance z výstupu zjistit vstup). Proto se jim říká taktéž jednosměrné funkce. Taktéž je bezkolizní (neexistuje stejný hash pro dva různé texty).

3.3.1 Hash algoritmus MD5

Využívá se při kontrole integrity souborů nebo pro ukládání hesel. V roce 1996 byla objevena bezpečnostní díra, a i když nebyla zásadní, začalo se od užívání MD5 upouštět a byl postupně nahrazován algoritmem SHA.

Postup vytvoření hashe:

1. Zpráva je doplněna na délku dělitelnou 512, tak, že se přidá 1 a potom tolik nul, aby zbylo 64b. do těchto 64 bitů je uložena délka původní zprávy.
2. Pote je rozdělena do bloků po 512 bitech.
3. Každý blok je dále rozdělen na 128b subbloky a ty na 4 32b slova A,B, C, D.
4. Bloky A, B, C, D jsou inicializovány na fixní konstanty

5. Při každém zpracování subbloků dochází v 16 kolech k promíchání bloků A, B, C, D, za využití operací XOR OR AND a bitových posunů.
6. To, co zbude v blocích A, B, C, D na konci výpočtu je seřazeno (v definovaném sledu). Tím je součet hotov.

3.3.2 SHA-X algoritmy

V podstatě vychází z MD5, pouze jsou doplněny o více šifrovacích kol na subblok, delší výsledný a o složitější funkci na šíření chyby. V současné době je bezpečnost SHA-1 na spadnutí, doporučuje se používat SHA256, který existuje i ve verzích SHA384 a SHA512.

4 SOUČASNÉ METODY PŘENOSU DAT V PRAKTICKÉM UŽITÍ

4.1 Úvod do terminologie ssh

Ssh (Secure shell) je souhrnný název, jak pro protokol využívající zabezpečenou komunikaci přes nezabezpečený kanál (např. Internet), tak i pro programy využívající toho rozhraní.

Pro komunikaci s využitím ssh je potřeba běžící server (sshd) na straně serveru a ssh na straně klienta. Zajímavou vlastností tohoto protokolu je, že se prokazuje jak server klientovi, tak klient serveru. Protokol jako takový zajišťuje tedy autentizaci účastníků, transparentní šifrování přenášených dat, zajištění jejich integrity a bezztrátovou kompresi.

4.1.1 Postup autentizace

Nejprve autentizace pomocí veřejného klíče. Na začátku je vygenerován pár klíčů, veřejný a privátní. Není možné z veřejného klíče odvodit privátní. Privátní je určen pouze pro jeho majitele, který by jej měl z principu velmi dobře chránit (např. bezpečným heslem).

Pokud se chce klient přihlásit k serveru, obdrží veřejnou část obou klíčů. Pakliže tyto části zná, je server považován za známý. V opačném případě je na klientovi, aby prozkoumal klíč a posoudil, zda se opravdu jedná o server, na který se chtěl přihlásit. Klient je na tuto skutečnost důrazně upozorněn. Důvody tohoto stavu mohou být buďto útok typu man-in-the-middle anebo reinstalace sshd na straně serveru.

Poté klient vygeneruje tzv. session key, kterým se šifruje následující komunikace (symetricky). Pomocí veřejného klíče a svého privátního klíče je tento session key zašifrován a odeslán serveru. Tímto postupem je zajištěno, že server musí znát svůj tajný klíč a tedy se nemůže vydávat za někoho jiného.

Následně si server ověří identitu klienta. Může pomocí hesla, případně využít např. asymetrickou kryptografii. Hlavní je, že po ověření klienta je proces autentizace ukončen a může dojít ke komunikaci s klientem.

4.1.2 Nevýhody ssh

Samotný systém ssh (ssh1) je již v dnešní době považován za zastaralý a není doporučeno jej používat. Nicméně vzhledem ke stále přetrvávající míře využívání tohoto protokolu na větším počtu serverů nelze tento protokol zatím zcela odstranit.

Dalším důležitým faktorem je nutnost řádně ověřit neznámý veřejný klíč, jinak může dojít k dešifrování důležitých informací nebo k útokům typu man-in-the-middle. Dále má v sobě ssh možnost vytvářet tunelová spojení, čímž je možno přenášet velké množství dat (riziko pro firmy, kde ne vždy věří svým zaměstnancům).

Taktéž povolení ssh před firewall může být problém, protože některé implementace ssh umožňují využívat VPN, což umožňuje spojení dvou vzdálených ethernetových sítí, jako by byli na jednom switchi.

4.1.3 SFTP, SCP

SCP je protokol umožňující zabezpečený přenos souborů mezi dvěma počítači za využití ssh. Nicméně scp má omezené možnosti a proto je nahrazován protokolem sftp. Je nutné podotknout, že zabezpečení je vlastnost ssh, nikoliv scp. Protokol sám o sobě žádné šifrování neposkytuje.

SFTP je označení pro protokol umožňující zabezpečené přenášení souborů. Byl navržen jako multiplatformní, takže například expanzi některých znaků neponechává na implementaci serveru. Protokol sám o sobě nezajišťuje šifrování ani autentizaci. To má na starosti protokol ssh-2, novější verze ssh.

4.2 SSL

Protokol ssl (Secure Socket Layer) vznikl jako rozšíření webových klientů o autentizaci a šifrování. Nicméně není vázán striktně na http, je možné jej využít i pro jiné protokoly a aplikace. S jeho pomocí bylo vylepšeno mnoho do té doby ne zrovna ideálních protokolů, počínaje telnetem a konče ftp. Nicméně je třeba mít na paměti, že tyto protokoly od počátku nebyly navrženy na bezpečný přenos dat.

4.2.1 Autentizace v SSL

Autentizace probíhá podobně jako v případě ssh. Nicméně v případě ssl může být dokazována jednostranně nebo oboustranně na základě certifikátů. Certifikát si lze představit jako vazbu mezi autoritou a klíčem, kterým se identifikují.

4.2.2 Výhody a nevýhody SSL

Velkou výhodou je dobrá implementace ssl na úrovni programových kódů. Nicméně velkou nevýhodou je nemožnost používat přes systém ssl digitální podpisy, které začíná stále více systémů vyžadovat.

Další výhodou i nevýhodou je systém autentizace. Tento postup eliminuje nutnost prvotní instalace důvěrného klíče na server, nicméně neumožňuje uživateli jednoduše rozpoznat důvěryhodnost certifikátů a uživatel tak musí sám tuto vlastnost "vytušit".

II. PRAKTICKÁ ČÁST

5 ŠIFROVACÍ APLIKACE

V následující části je rozebrána aplikace na teoretické úrovni včetně popisu jednoduchého ovládání.

5.1 Platforma

Jako základna pro tuto aplikaci byla vybrána platforma ASP.NET, především z důvodu vhodnosti a podpory pro nejrůznější kryptometody. Tato platforma má již ve svém základu zabudovánu právě podporu pro zabezpečený přenos jakýchkoliv dat. Pakliže by ovšem situace vyžadovala naprogramovat si nějakou dosud neznámou šifru, nebyl by zde žádný problém rozšířit již stávající kolekci šifer o tuto novou šifrovací metodu.

Dalším významným důvodem pro výběr této platformy byla přenositelnost kódu, resp. jeho možnost spouštění na nejrůznějších systémech (Windows, Linux atd.). Vzhledem k tomu, že ASP.NET je programovací prostředí, které je využíváno pro internetové aplikace a stránky, je možné tuto aplikaci spustit na jakémkoliv moderním systému, který využívá některý sofistikovaný software pro přístup na webové stránky (textové prohlížeče již nejsou uvažovány jako aktivní).

Pro vývoj této aplikace bylo využito vývojového prostředí MS Visual Web Developer 2008 EE. Toto prostředí umožňuje efektivně ladit výsledný program a taktéž dokáže jednoduše simulovat webový server, takže odpadá jakákoliv nutnost instalace simulačního softwaru, který by dokázal věrohodně napodobit reakce serveru na osobním počítači.

5.2 Testovací nástroj CryptoTool 2.0

Jedná se o volně šiřitelný software, postavený na bázi technologií C# a .NET. Jeho primární funkcí je umožnit jednoduše sestavit některou z nejznámějších současných i starších šifer.

Uživatelské rozhraní je postaveno na systému Drag&Drop, tedy je zde prázdná plocha, na kterou se umístí jednotlivé ikony, představující modely šifer, případně jiné funkční objekty, např. textový vstup, vstup ze souboru, generátor náhodných znaků, textový výstup, výstup do souboru a jiné.

Tento program umožňuje realizovat např. následující šifry či funkce:

- ❖ Klasické
 - Caesar
 - Enigma
 - Playfair
 - Vernam
 - Vigenère
- ❖ Moderní
 - AES
 - DES
 - 3DES
 - WEP Protocol
 - RC2
- ❖ HASH
 - MD5
 - SHA
 - PKCS#5
- ❖ Kryptoanalitické
 - Frekvenční analýzu
 - Autokorelační funkci
 - Cube attack
 - WEP protocol attack
 - nejrůznější porovnávací a podpůrné funkce

Veškeré funkce jsou vlastně jakousi skládkou z jednotlivých komponent, které je jen potřeba správně vzájemně propojit, popř. nastavit parametry.

5.3 Výběr šifrovacích metod

Na ukázkou je zde užito několik šifrovacích technik, konkrétně DES,AES,RC2, SHA-1 a MD5 pro čistý text a DES,AES a RC2 pro textové soubory.

Důvody pro výběr právě těchto technik jsou rozšířenost, rychlost a relativní jednoduchost. Z klasických šifer jsou vybrány právě DES, AES a RC2. Jde o symetrické blokové šifry, přičemž za nejbezpečnější je považována právě šifra AES.

Protože hashovací šifry, jako je SHA-1 a MD5, vytvářejí šifru o konstantní délce, nebyly využity pro zašifrování souborů, jelikož se v tomto použití v praxi neuplatňují.

5.4 Popis aplikace

Obrázek 2 Náhled webové části aplikace

Aplikace se nachází na webové adrese <http://magiccode.aspone.cz>, na které je umístěn kompletní kód ukázkového příkladu. Jedná se o freewebhosting, tedy neplacený webhosting, což může přinášet určitá omezení, nicméně během dlouhodobého sledování chodu aplikace nebyly pozorovány žádné výpadky ani zpomalení chodu aplikace.

Jak je vidět na Obrázku 2, samotná aplikace má dvě části. První, která umožňuje šifrovat text na ukázkou přímo přes webové rozhraní a zobrazuje původní text, zašifrovaný text a výsledek po dešifrování. Oba dvě klíčové části, klíč a inic. vektor, jsou generovány programem, jelikož se v případě chyby (např. malý počet znaků) hůře opravují chyby, navíc je snahou používat pokud možno náhodné klíče a inic. vektory.

Druhá část aplikace umožňuje zašifrování a dešifrování celých souborů typu textový soubor. Soubor je během tohoto procesu nahrán na webový server a následně přepsán při dalším šifrování nebo dešifrování.

Pakliže dojde při šifrování nebo dešifrování k jakýmkoliv chybám, aplikace tuto možnost zachytí a zobrazí standardní chybové hlášení, které umístí do sekce, které odpovídá místo výskytu chyby.

Zatímco v první části nemá uživatel možnost zadávat vlastní šifrovací klíče a inicializační vektory, jelikož se jedná o zdlouhavou a náročnou práci na přesnost (zadat přesný počet

znaků může být někdy velmi obtížné), ve druhé části již uživatel může při dešifrování zadat své již získané klíče a vektory. Opět zde nebyla dána možnost zadat si své vlastní šifrovací klíče a vektory, aby systém nebyl náchylný na nevhodné znaky v klíči anebo na nevhodnou délku klíče či vektoru.

Samotný inicializační vektor je sada náhodných znaků u velikosti jednoho bloku, které umožňují "nastartovat" šifru. Jelikož většina šifer jsou rundové a vstupem bývá výstup z předchozí rundy, poslouží inic. vektor jako výstup z neexistující rundy a tedy pomůže zašifrovat první blok dat. Poté se již k ničemu nepoužívá.

Ukázka již hotového zdrojového kódu je v příloze P1. Je zde vybrána šifra RC2, taktéž obsluha uživatelských událostí, jako je výběr šifry a kliknutí na určité tlačítko.

5.4.1 Webové šifrování

Tato část je rozdělena do několika sekcí, jak je patrné z Obrázku 3.

The screenshot shows a web application interface for encryption and decryption, organized into three main sections:

- Data pro zašifrování (Data for encryption):** This section contains four input fields: "Šifrovací klíč:" (Encryption key), "Inicializační vektor:" (Initialization vector), "Typ šifry:" (Encryption type), and "Text k zašifrování:" (Text to encrypt). The "Typ šifry:" dropdown menu is open, showing options: DES (selected), AES, RC2, and SHA-1. Below the text input field is a "Zašifruj" (Encrypt) button.
- Zašifrovaná data (Encrypted data):** This section is a large empty text area intended for displaying the encrypted output.
- Dešifrovaná data (Decrypted data):** This section is a large empty text area intended for displaying the decrypted output.

Obrázek 3 Část pouze pro webové rozhraní

Na Obrázku 4 je vidět sekce pro vložená data, výběr šifry a zobrazení klíče.



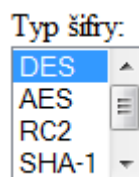
Obrázek 4 Výběr dat pro zašifrování

První část této sekce je zobrazena na Obrázku 5 a obsahuje atributy šifrovací klíč a inicializační vektor, které jsou vysvětleny a jejich funkce je popsána v kapitole 5.3.



Obrázek 5 První část sekce pro práci s daty

Dále je zde menu pro výběr typu šifry, viz. Obrázek 6. Celkový výčet všech použitých šifrovacích metod je v kapitole 5.2.



Obrázek 6 Menu pro výběr šifry

Jako první vybraný algoritmus je šifra DES. Celá tato první část je předpřipravena tak, aby uživatel mohl pouze vložit svá data a rovnou zkusit zašifrovat text.

Poslední část sekce obsahuje komponentu, do které je potřeba vložit data, která mají být zašifrována, jak je vidět z Obrázku 7.

Text k zašifrování

Prosím vložte text.

Obrázek 7 Část pro vkládání dat a spuštění šifrování

Po kliknutí na tlačítko "Zašifruj" dojde k vygenerování klíče a inicializačního vektoru (pokud je šifra vyžaduje, např. hashovací šifry nic takového nepotřebují). Poté jsou data zašifrována, výsledek je umístěn do části "Zašifrovaná data", viz. Obrázek 8.

Zašifrovaná data

Obrázek 8 Sekce pro zašifrovaná data

Poté je tento text dešifrován zpět do původní podoby a výsledek dešifrování je vidět v sekci "Dešifrovaná data" (opět hashovací funkce jsou jednosměrné, takže v jejich případě se v sekci "Dešifrovaná data" nic neobjeví).

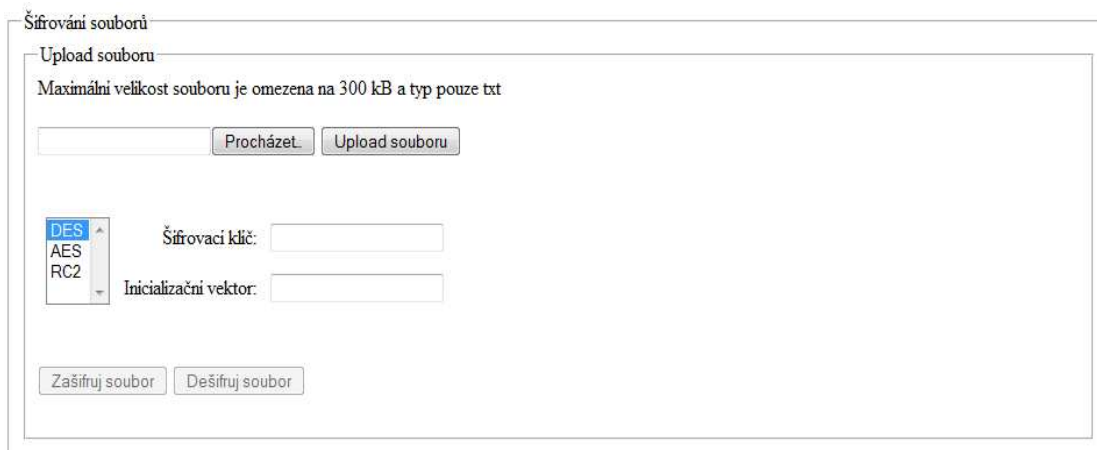
Dešifrovaná data

Obrázek 9 Sekce pro následně dešifrovaná data

Celá první část slouží spíše pro náhled na funkčnost šifer, z toho důvodu jsem neumožnil zadávat vlastní klíče a inicializační vektory.

5.4.2 Rozhraní pro šifrování souborů

V této části může uživatel nahrát svůj soubor z disku na server, ten poté zašifrovat, případně dešifrovat. Vzhled této části aplikace je vidět na Obrázku 10.

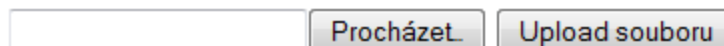


The screenshot shows a web interface titled "Šifrování souborů" (File Encryption). It features a section for uploading a file, with a text box and buttons for "Procházet..." (Browse...) and "Upload souboru" (Upload file). Below this, there is a dropdown menu for selecting encryption algorithms (DES, AES, RC2), with "DES" currently selected. To the right of the dropdown are input fields for "Šifrovací klíč:" (Encryption key) and "Inicializační vektor:" (Initialization vector). At the bottom of the interface are two buttons: "Zašifruj soubor" (Encrypt file) and "Dešifruj soubor" (Decrypt file).

Obrázek 10 Rozhraní pro šifrování a dešifrování souborů

První část této sekce obsahuje komponentu pro výběr souboru z lokálního disku a následné nahrání souboru na server. Velikost je omezena na 300 kB a textové soubory, jak lze vidět na Obrázku 11.

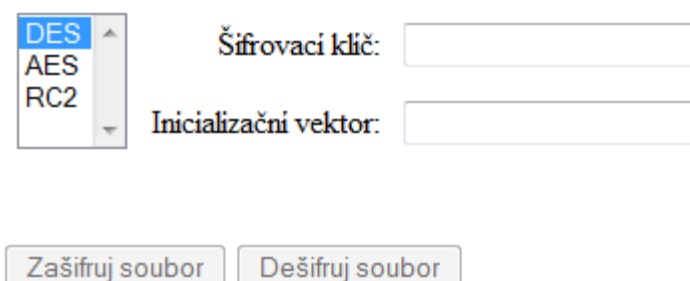
Maximální velikost souboru je omezena na 300 kB a typ pouze txt



This image shows a close-up of the file upload component. It consists of a text input field, a "Procházet..." (Browse...) button, and an "Upload souboru" (Upload file) button.

Obrázek 11 Komponenta pro nahrávání souborů

Samotné tlačítka "Zašifruj soubor" a "Dešifruj soubor" jsou neaktivní, dokud nedojde k prvnímu úspěšnému nahrání souboru na server (viz. Obrázek 12).



This image shows a close-up of the encryption controls. It includes a dropdown menu for encryption algorithms (DES, AES, RC2), with "DES" selected. To the right are input fields for "Šifrovací klíč:" (Encryption key) and "Inicializační vektor:" (Initialization vector). Below these are two buttons: "Zašifruj soubor" (Encrypt file) and "Dešifruj soubor" (Decrypt file).

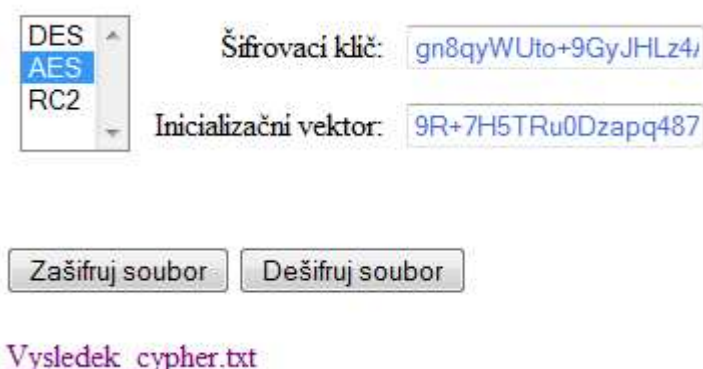
Obrázek 12 Ovládací rozhraní šifry

Poté jsou aktivní a pracují vždy s posledním nahraným souborem. Při každém stisku tlačítka "Zašifruj soubor" je vygenerován šifrovací klíč a inicializační vektor. Tyto dva

klíčové údaje jsou pak použity pro dešifrování. Takže pokud chce uživatel zašifrovat více než 1 soubor, musí si tyto údaje poznačit pro pozdější dešifrování.

Pokud se uživatel pokusí nahrát soubor větší než 300kB anebo jiný typ souboru než textový soubor, aplikace mu nedovolí pokračovat k funkci šifrování nebo dešifrování. Omezení pro tuto funkci jsou zavedeny z důvodu nepatřičného přetěžování serveru velkými soubory a taktéž omezeným místem pro uschování nahraných a dešifrovaných souborů.

Na Obrázku 13 je vidět výsledek po úspěšném nahrání a zašifrování souboru. Šifrovací klíč a inicializační vektor jsou vypsány napravo od menu výběru šifry. Právě údaje v těchto polích jsou potom použity pro dešifrování souboru. Aplikace neumožňuje žádné uschování nebo uložení těchto údajů na serveru, tedy při ztrátě již není možné soubor rozšifrovat.



DES
AES
RC2

Šifrovací klíč: gn8qyWUto+9GyJHLz4/

Inicializační vektor: 9R+7H5TRu0Dzapq487

Zašifruj soubor Dešifruj soubor

[Vysledek cypher.txt](#)

Obrázek 13 Výsledek po nahrání a zašifrování souboru

5.5 Možnosti budoucího vývoje

Jak již bylo řečeno v úvodu této kapitoly, možnosti platformy ASP.NET jsou velmi široké, takže budoucímu vylepšování aplikace není nijak bráněno a aplikace je psána s ohledem na budoucí rozšíření.

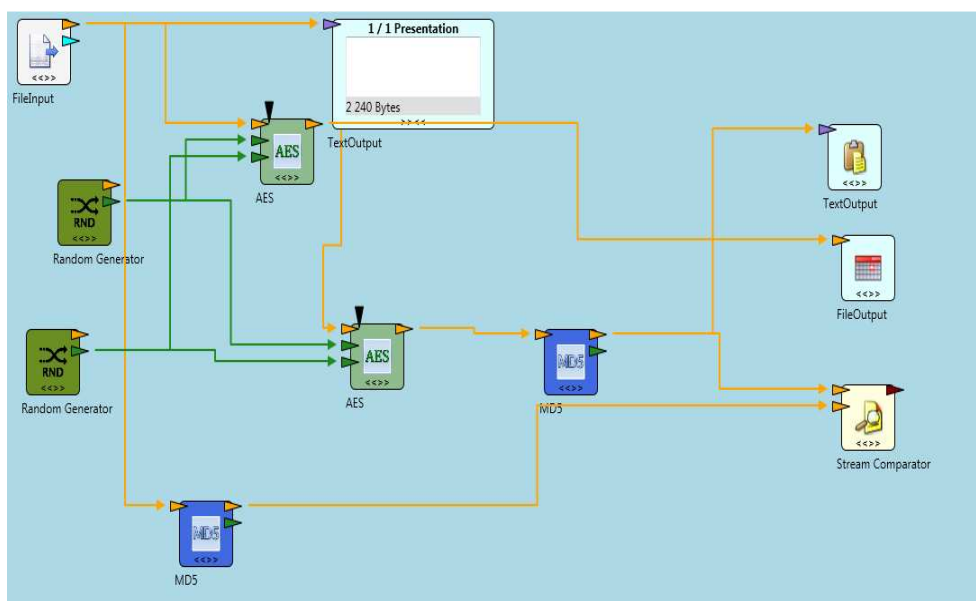
Prvním rozšířením by mohla být uživatelská databáze použitých šifrovacích údajů. Taktéž by bylo možné navrhnout uživatelské přihlašovací rozhraní, které by umožnilo šifrovat soubory pouze přihlášeným uživatelům. Pravděpodobná podoba by byla využití šifry SHA-1 pro zabezpečený přístup do databáze uživatelů.

V současné podobě není tato aplikace navržena pro šifrování většího množství souborů současně. Díky přihlášení uživatele by bylo možné rozlišit jednotlivé vstupy a vyvarovat se případnému přepsání souborů.

5.6 Efektivita jednotlivých metod

Pro porovnání efektivity jednotlivých šifrovacích postupů v praktické části byl využit volně dostupný program CryptoTool 2, který je postavený na základě technologií C# a .NET. Teoretický popis a rozbor aplikace je popsán v kapitole 5.2.

S jeho pomocí byly postupně šifrovány a následně dešifrovány různě velké soubory pomocí rozdílných šifrovacích algoritmů, které program obsahuje. Ukázkové sestavení šifry AES v tomto programu je na Obrázku 5.



Obrázek 14 Ukázka šifrovacího algoritmu AES v prostředí CryptoTool 2

Jak je možno vidět v Tabulce 2, některé šifry jsou pro určité aplikace nevhodné až nevyužitelné. Nejvhodnější metodou pro šifrování se jeví metoda AES, která ovšem patří k nejmodernějším metodám, jaké byly doposud vynalezeny a byla vyvinuta právě s ohledem na rychlost a spolehlivost a pro široké možnosti využití.

Postupně byly do zkonstruovaných šifrovacích bloků vkládány různé soubory o rozdílných velikostech. Cílem bylo zjistit, kdy některá šifra začne vykazovat znaky nedokonalého zpětného dešifrování anebo bude zabírat příliš mnoho strojového času. Aby byla zachována podobnost s vytvořenou aplikací, byly veškeré použité soubory pouze textové.

Algoritmus	Velikost šifrovaného souboru	
	malý (desítky kB)	velký (až jednotky MB)
AES	vhodný	vhodný
RSA	vhodný	nevhodný
3DES	vhodný	použitelný
RC2	vhodný	vhodný

Tabulka 2 Tabulka použitelnosti jednotlivých šifer

ZÁVĚR

Cílem této bakalářské práce bylo uvést, rozebrat a seznámit s moderními metodami zabezpečení dat, případně možnosti jejich prolomení a taktéž vývoj simulační aplikace, která by umožňovala vyzkoušet si tyto postupy v praxi.

Nejprve je v první části uveden historický postup jednotlivých etap vývoje kryptografie. Následně jsou zpracovány moderní metody a postupy, které se aplikují v současnosti především v oblasti bezpečnosti dat. Taktéž jsou v této části uvedeny základní teoretické a matematické myšlenkové postupy, které se využívají pro vylepšení návrhu šifry a mají pomoci zlepšit její bezpečnost. Tato část je zakončena pojednáním o moderních způsobech využití šifer pro praktické aplikace.

V druhé, a tedy praktické, části je směr zájmu zaměřen na vývoj aplikace a důvody volby platformy. Celá aplikace je pojata jako demonstrační program, proto běží na freehostingovém serveru. Ačkoliv jsou zde jistá omezení, např. max. velikost databáze a diskového prostoru, hlavní vlastnosti webhostingu, tedy dostupnost a rychlost, jsou zachovány.

Tato aplikace je psána s výhledem rozšíření do budoucna, nicméně pro širší využití veškerých možností této platformy (některé návrhy již byly publikovány s kapitole 5.4) by bylo nutné použít některé výkonnější verze webhostingu.

ZÁVĚR V ANGLIČTINĚ

The aim of this work was to introduce a modern security methods for data, or the possibility of their breaking and also the development of simulation applications, which would allow to test these procedures in practice.

The first part of this thesis discuss historical process stages of development of cryptography. Subsequently, modern methods and procedures which are currently mainly applied in the field of data security, are described. Also, in this section, there is given the basic theoretical and mathematical thought processes, which are used to improve ciphers and to help improve their security. This part ends with a treatise on modern uses of cryptograms for practical applications.

In the second, and so practical, the direction of interest is focused on developement of application and the reasons for choosing of used platform. The entire application is designed as a demonstration program that can run on freehosting server. Although there are some restrictions, eg maximum database size and disk space, hosting the main features, namely the availability and speed, are maintained.

This application is written with a future prospect to expansion, but for the wider use of all the possibilities of used platform would be necessary to use some of the more powerful versions of web hosting.

SEZNAM POUŽITÉ LITERATURY

- [1] ZELENKA, J., ČAPEK, J., FRANCEK, J., JANÁKOVÁ, H. Ochrana dat, Kryptologie. Gaudeamus, září 2003. 171 s. SBN 80-7041-737-4.
- [2] ČANDÍK, Marek. Základy informační bezpečnosti. vyd. Zlín : Univerzita Tomáše Bati, 2004. 107 s. ISBN 8073182181.
- [3] KATZ, Jonathan. Introduction to Modern Cryptography: Principles and Protocols. Chapman & Hall, 1 edition. 2007. 552 s. ISBN 978-1584885511.
- [4] BITTO, O. Šifrování a biometrika. BEN, 2005. 168 s. ISBN 80-86686-48-5.
- [5] KLÍMA, Vlastimil; ROSA, Tomáš. Kryptologie pro praxi – DSA, ECDSA. Dostupné z WWW: [http://crypto-world.info/klima/2004/st_2004_04_21_21.pdf].
- [6] KATZ, Jonathan. Introduction to Modern Cryptography: Principles and Protocols. Chapman & Hall, 1 edition. 2007. 552 s. ISBN 978-1584885511.
- [7] SINGH, Simon. Kniha kódů a šifer : Tajná komunikace od starého Egypta po kvantovou kryptografii. 1. vydání, Praha : Dokořán : Argo, 2003. 382 s. ISBN: 80-86569-18-7.

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

AI	Artificial intelligence
AES	Advanced Encryption Standart
ASP.NET	Active Server Pages .NET
DES	Data Encryption Standart
DSA	Digital Signature Algorithm
DSS	Digital Signature Standart
FTP	File Transfer Protocol
FIPS	Federal Information Processing Standart
MD5	Message Digest Algorithm
NIST	National Institute of Standart and Technology
SCP	Secure Copy
SFTP	Secure File Transfer Protocol
SHA-X	Secure Hask Algorithm
SSH	Secure Shell
SSL	Secure Sockets Layer
VPN	Virtual Private Network

SEZNAM OBRÁZKŮ

Obrázek 1 Rozdělení šifer	15
Obrázek 2 Náhled webové části aplikace.....	30
Obrázek 3 Část pouze pro webové rozhraní	31
Obrázek 4 Výběr dat pro zašifrování	32
Obrázek 5 První část sekce pro práci s daty.....	32
Obrázek 6 Menu pro výběr šifry	32
Obrázek 7 Část pro vkládání dat a spuštění šifrování.....	33
Obrázek 8 Sekce pro zašifrovaná data	33
Obrázek 9 Sekce pro následně dešifrovaná data	33
Obrázek 10 Rozhraní pro šifrování a dešifrování souborů	34
Obrázek 11 Komponenta pro nahrávání souborů.....	34
Obrázek 12 Ovládací rozhraní šifry	34
Obrázek 13 Výsledek po nahrání a zašifrování souboru.....	35
Obrázek 14 Ukázka šifrovacího algoritmu AES v prostředí CryptoTool 2	36

SEZNAM TABULEK

Tabulka 1 Časová náročnost jednotlivých operací.....	16
Tabulka 2 Tabulka použitelnosti jednotlivých šifer.....	37

SEZNAM PŘÍLOH

Veškeré přílohy se nacházejí na přiloženém CD nosiči, který obsahuje:

- elektronický text bakalářské práce ve formátu pdf
- šifrovací aplikaci včetně všech zdrojových souborů

PŘÍLOHA P1: UKÁZKA ZDROJOVÉHO KÓDU APLIKACE

```
using System;
using System.IO;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Security;
using System.Security.Cryptography;
using System.Text;
using System.Data;
using System.Data.SqlClient;

namespace Sifrovani
{
    public partial class _Default : System.Web.UI.Page
    {
        public int Sifra_RC2(String ptext,bool soubor)
        {
            String vysledek;
            RC2CryptoServiceProvider rc2 = new
RC2CryptoServiceProvider();
            byte[] rbData;
            rc2.GenerateKey();
            rc2.GenerateIV();
            RC2klic_s = Convert.ToBase64String(rc2.Key);
            RC2IV_s = Convert.ToBase64String(rc2.IV);
            if (soubor == false)
            {
                if (ptext.Length > 92160)
                {
                    vysledek = "Data přesáhla veliksot 90kB";
                    deciphertext.Text = vysledek;
                    return 0;
                }
                RC2klic = rc2.Key;
                RC2IV = rc2.IV;
                klic.Text = RC2klic_s;
                ptext = String.Format("{0,5:00000}" + ptext,
ptext.Length);
                delka_s = 5;
                rbData = System.Text.UTF8Encoding.UTF8.GetBytes(ptext);
            }
            else
            {
                String text = data_soubor;
                text = text.Length.ToString()+text;
                rbData = System.Text.UTF8Encoding.UTF8.GetBytes(text);
                delka_s = 6;
            }
            ICryptoTransform rc2crypt = rc2.CreateEncryptor(rc2.Key,
rc2.IV);
            MemoryStream mStream = new MemoryStream(rbData);
            MemoryStream mOut = new MemoryStream();
            CryptoStream cs = new CryptoStream(mStream, rc2crypt,
CryptoStreamMode.Read);
```

```

        int bytesRead;
        byte[] output = new byte[1024];
        do
        {
            bytesRead = cs.Read(output, 0, 1024);
            if (bytesRead != 0)
                mOut.Write(output, 0, bytesRead);
        } while (bytesRead > 0);

        if (mOut.Length == 0)
        {
            vysledek = "";
        }
        else
        {
            vysledek = Convert.ToBase64String(mOut.GetBuffer(), 0,
(int)mOut.Length);
            inicvektor.Text = Convert.ToBase64String(rc2.IV);
            klic.Text = Convert.ToBase64String(rc2.Key);
        }
        if (soubor == false)
        {
            ciphertext.Text = vysledek.ToString();
            return 0;
        }
        else
        {
            Soubor_key.Text = Convert.ToBase64String(rc2.Key);
            Soubor_IV.Text = Convert.ToBase64String(rc2.IV);
            String soubor_cesta =
Server.MapPath(@"~\www\files\vysledek_cypher.txt");
            if (File.Exists(soubor_cesta))
            {
                File.Delete(soubor_cesta);
            }
            System.IO.StreamWriter Stream_writer =
File.CreateText(soubor_cesta);
            Stream_writer.Write(vysledek);
            Stream_writer.Dispose();
            Stream_writer.Close();
            odkaz_url.Text = "Vysledek_cypher.txt";
            odkaz_url.NavigateUrl =
@"~\www\files\vysledek_cypher.txt";
            return 0;
        }
    }

    public int DSifra_RC2(String ctext, bool soubor)
    {
        String vysledek;
        int nReturn;

        byte[] bPlain;

        if (soubor == false)
        {
            bPlain = new byte[ctext.Length];
            try
            {
                bPlain =
Convert.FromBase64CharArray(ctext.ToCharArray(), 0, ctext.Length);

```

```

    }
    catch (Exception)
    {
        vysledek = "Vstupní data jsou špatně zakódována.";
        deciphertext.Text = vysledek.ToString();
        return 0;
    }
}
else
{
    RC2klic = Convert.FromBase64String(Soubor_key.Text);
    RC2IV = Convert.FromBase64String(Soubor_IV.Text);
    bPlain =
Convert.FromBase64CharArray(data_soubor.ToCharArray(), 0, data_soubor.Length);
}

    RC2CryptoServiceProvider rc2 = new
RC2CryptoServiceProvider();
    ICryptoTransform rc2decrypt = rc2.CreateDecryptor(RC2klic,
RC2IV);

    MemoryStream mOut = new MemoryStream();
    CryptoStream cs = new CryptoStream(mOut, rc2decrypt,
CryptoStreamMode.Write);

    long lRead = 0;
    long lTotal = bPlain.Length;

    try
    {
        do
        {
            cs.Write(bPlain, (int)mOut.Length,
((int)bPlain.Length - (int)mOut.Length));
            lRead = (int)mOut.Length;
        } while (lTotal != lRead);
        UTF8Encoding aEnc = new UTF8Encoding();
        vysledek = aEnc.GetString(mOut.GetBuffer(), 0,
(int)mOut.Length);
        String strLen = vysledek.Substring(0, delka_s);
        int nLen = Convert.ToInt32(strLen);
        vysledek = vysledek.Substring(delka_s, nLen);
        nReturn = (int)mOut.Length;
        deciphertext.Text = vysledek.ToString();
        if (soubor == true)
        {
            String soubor_cesta =
Server.MapPath(@"~\www\files\vysledek_decypher"+pripona);
            System.IO.StreamWriter Stream_writer =
File.CreateText(soubor_cesta);
            Stream_writer.Write(vysledek);
            Stream_writer.Dispose();
            Stream_writer.Close();
            odkaz_url.Text = "Vysledek_decypher.txt";
            odkaz_url.NavigateUrl =
@"~\www\files\vysledek_decypher.txt";
        }
        return 0;
    }
    catch (Exception)
    {

```

```

        vysledek = "Proces dekryptování skončil chybou,
pravděpodobně poškozená data nebo špatný klíč.";
        deciphertext.Text = vysledek.ToString();
        return 0;
    }
}

public void Nahraj_soubor(object sender, EventArgs e)
{
    vysledek_file.Text = "";
    if (soubor.HasFile)
    {
        HttpPostedFile myFile = soubor.PostedFile;
        if (myFile.ContentLength > 307200)
        {
            vysledek_file.Text = "Soubor přesahuje max.
velikost.";
        }
        else
        {
            try
            {
                nizev_souboru = soubor.FileName.ToString();
                pripona =
nizev_souboru.Substring(nizev_souboru.Length - 3, 3);
                if (pripona.Equals("txt"))
                {
                    soubor.SaveAs(MapPath(@"~\www\files\") +
soubor.FileName);
                    String cesta_souboru =
Server.MapPath(@"~\www\files\" + soubor.FileName);
                    StreamReader fp =
File.OpenText(cesta_souboru);
                    data_soubor = fp.ReadToEnd();
                    souborsifruj.Enabled = true;
                    desifrujsoubor.Enabled = true;
                }
                else
                {
                    vysledek_file.Text = "Nahrany soubor není
typu txt";
                }
            }
            catch (Exception ex)
            {
                vysledek_file.Text = "Chyba: " +
ex.Message.ToString();
            }
        }
    }
}

public void Desifruj_soubor(object sender, EventArgs e)
{
    switch (sifra_soubor.Text)
    {
        case "DES":
            DSifra_DES(plaintext.Text, true);
            break;
        case "AES":
            DesifrujAES(plaintext.Text, true);
    }
}

```

```
        break;
    case "RC2":
        DSifra_RC2(plaintext.Text, true);
        break;
    }
}

public void Zasifruj_soubor(object sender, EventArgs e)
{
    switch (sifra_soubor.Text)
    {
        case "DES":
            Sifra_DES(plaintext.Text, true);
            break;
        case "AES":
            Sifra_AES(plaintext.Text, true);
            break;
        case "RC2":
            Sifra_RC2(plaintext.Text, true);
            break;
    }
}

public void Page_Load(object sender, EventArgs e)
{
    klic.Text = "";
}

}
}
```