WWW průvodce moderní kryptografií

Modern cryptography - Web guide

Michal Karger

Bakalářská práce
2008

Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky
Ústav aplikované informatiky
akademický rok: 2007/2008

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení:	**Michal KARGER**
Studijní program:	**B 3902 Inženýrská informatika**
Studijní obor:	**Informační technologie**
Téma práce:	**WWW průvodce moderní kryptografií**

Zásady pro vypracování:

1. **Vypracujte literární rešerši týkající se tvorby www stránek.**
2. **Seznamte se s dostupnými podklady o moderní kryptografii.**
3. **Navrhněte design www stránek, proveďte volbu vhodného software pro tvorbu stránek a do textu práce uveďte jeho stručný popis.**
4. **Vytvořte www stánky o moderní kryptografii.**
5. **Vámi vytvořené www stránky o moderní kryptografii umístěte na samostatné CD-ROM jako přílohu bakalářské práce.**

Rozsah práce:

Rozsah příloh:

Forma zpracování bakalářské práce: **tištěná/elektronická**

Seznam odborné literatury:

1. **An Introduction to Cryptography, Network Associates, Inc., 1998, Santa Clara, California, USA.**
2. **Chey Cobb (2004): Cryptography for Dummies. Wiley Publishing, Inc., Indianapolis, Indiana, USA.**
3. **Man Young Ree (2003): Internet Security: Cryptographic Principles, Algorithms and Protocols. John Wiley and Sons Ltd, Chichester, West Sussex, England, United Kingdom.**
4. **Wembo Mao (2003): Modern Cryptography: Theory and Practice. Prentice Hall, Inc., Upper Saddle River, New Jersey, USA & Hewlett-Packard Company & Person Education Ltd.**
5. **Fuller, R.G. & L.A. Ulrich (2004): HTML in 10 Simple Steps or Less. Wiley Publishing, Inc., Indianapolis, Indiana, USA.**

Vedoucí bakalářské práce: **Ing. Karel Perůtka, Ph.D.**
Ústav řízení procesů

Datum zadání bakalářské práce: **20. února 2008**

Termín odevzdání bakalářské práce: **5. května 2008**

Ve Zlíně dne 20. února 2008

UNIVERZITA TOMÁŠE BATI VE ZLÍNĚ
Fakulta aplikované informatiky
L.S.

prof. Ing. Vladimír Vašek, CSc.
děkan

doc. Ing. Ivan Zelinka, Ph.D.
ředitel ústavu

ABSTRAKT

Tato bakalářská práce se zabývá základními pojmy z oblasti moderní kryptografie, které jsou prezentovány pomocí WWW průvodce. Teoretická část obsahuje vysvětlení důležitých pojmů, rozdělení šifer a popis několika známých algoritmů doplněné příkladem. V závěru této části jsou popsány základy tvorby WWW stránek.

Praktická část se věnuje tvorbě webového průvodce. Dále následuje popis programu PSPad, jež byl použit k vytvoření webového průvodce. V závěru práce naleznete popis nastavení šifrovacího programu Truecrypt.

Klíčová slova: kryptografie, kryptoanalýza, klíč, otevřený text, šifrování, dešifrování, symetrické šifry, asymetrické šifry, digitální podpis, digitální certifikát, DES, RSA, El Gamalův systém, HTML, CSS, PSPad, TrueCrypt.

ABSTRACT

This bachelor thesis deals with basic conceptions of modern cryptography, which are presented by web guide. Theoretical part contains explanation of important conceptions, division of ciphers and a description of some known algorithms with an example included. In the end of this part the basics of web pages creation is described.

The practical part deals with the creation of the web guide. Then follows the description of a PSPad program, which was used to create the web guide. In the end of the thesis you will find a setup description of the Truecrypt encryption program.

Keywords: cryptography, cryptanalysis, key, plaintext, encryption, decryption,symmetric algorithms, public key cryptography, digital signature, digital certificates, DES, RSA, El Gamal cryptosystem, HTML, CSS, PSPad, Truecrypt.

Děkuji Ing. Karlu Perůtkovi, Ph.D. za odborné vedení bakalářské práce a ochotu při poskytování cenných rad.

Prohlašuji, že jsem na bakalářské práci pracoval samostatně a použitou literaturu jsem citoval. V případě publikace výsledků, je-li to uvolněno na základě licenční smlouvy, budu uveden jako spoluautor.

Ve Zlíně ……

Podpis diplomanta

OBSAH

ÚVOD

Kryptografie je velmi starý vědní obor, který v historii ovlivnil mnoho událostí. Od dob, kdy Julius Caesar posílal svým velitelům rozkazy zašifrované jednoduchým posunutím abecedy, se hodně změnilo, ale princip zůstal stejný - zajistit, aby přístup k dané informaci měla jen oprávněná osoba a informace se nemohla dostat do neoprávněných rukou. Postupem času se šifry zdokonalovaly, až dosáhly na úroveň dnešních propracovaných algoritmů, k jejichž aplikaci jsou potřeba počítače.

Čím dál častěji je kryptografie spojována v souvislosti s Internetem. S jeho masivním rozvojem a rostoucím počtem operací, které se uskutečňují přes Internet, přichází i nutnost střežit bezpečnost citlivých informací, aby nedošlo k jejich zneužití. Zde nachází kryptografie uplatnění při posílání e-mailů, přenosu souborů nebo při ochraně hesel. Kryptografie se neustále vyvíjí a vytváří se nové algoritmy. Nové algoritmy jsou však přijímány s jistou nedůvěrou, protože až delší časové období ukáže použitelnost algoritmu a jeho odolnost proti různým druhům útoků.

Cílem této práce je vytvořit přehledný WWW průvodce, který by zájemcům vysvětlil důležité pojmy moderní kryptografie, principy známých algoritmů a použití vybraného kryptografického softwaru.

I. TEORETICKÁ ČÁST

1 KRYPTOGRAFIE

1.1 Co je to kryptografie

Kryptografie je věda, která s využitím matematiky šifruje a dešifruje data. Umožňuje uschovat citlivé informace nebo je přenášet přes nezabezpečené sítě jako je internet [1].

„*Zatímco kryptografie je věda o vytváření šifrovacích systémů, kryptoanalýza je proces luštění původní zprávy ze zašifrovaného textu v případě, kdy není k dispozici příslušný klíč. Kryptologie je pak souhrnným označením pro kryptografii a kryptoanalýzu*.“ (PIPER, MURPHY, 2006, s.15-16)

1.2 Princip kryptografie

Data, která mohou být čtena a chápána bez jakýchkoliv opatření, se nazývají otevřený text [1].

Proces přeměny otevřeného textu do nečitelné podoby se nazývá šifrování. Šifrováním vzniká zašifrovaný text. Opakem je pak dešifrování, kdy je zašifrovaný text převeden zpět na otevřený text [10].

Obr. 1 – Šifrování a dešifrování [1]

Šifra je matematická funkce používaná v procesu šifrování a dešifrování. Šifra s využitím klíče zašifruje otevřený text. Klíč může být slovo, číslo nebo fráze. Z otevřeného textu se tak stane zašifrovaný text. Bezpečnost zašifrovaných dat závisí na síle šifry a na utajení klíče. Kryptosystém zahrnuje šifru spolu se všemi možnými klíči a protokoly, které jsou nutné pro její funkčnost [1].

2 KRYPTOANALÝZA

2.1 Kryptoanalytické metody

Kryptoanalýza se tedy zabývá luštěním šifer. Existuje mnoho kryptoanalytických metod, z nichž zde jsou ty nejčastější:

a) Útok hrubou silou – univerzální princip, kdy útočník zkouší všechny možné kombinace klíče. Z toho plyne, že čím delší je klíč, tím je útok více náročný na čas a výpočetní výkon. Například vyzkoušení všech kombinací 128-mi bitového klíče by při rychlosti 10^6 MIPS trvalo $5 \cdot 10^{28}$ let. Podle toho, jak dlouho má být informace utajena, volíme délku klíče [7].

   Tento typ útoku se používá poměrně často a to hlavně díky rostoucímu výkonu počítačů. V minulosti se pomocí něj podařilo prolomit některé známé šifry. V roce 1997 byl distribuovaným výpočetním výkonem prolomen 56-ti bitový RSA algoritmus RC5 za méně než 250 dní. Stejně dopadl i hojně používaný algoritmus DES, u kterého se předpokládalo, že by mohl takovému druhu útoku odolat [2].

b) Luštění se znalostí šifrovaného textu – útočník zde musí získat několik zpráv šifrovaných stejným algoritmem a podle stejného klíče. Analýzou šifrovaného textu a jeho aproximací lineární funkcí otevřeného textu může dojít k prolomení šifry. Tato metoda se nazývá lineární kryptoanalýza [7].

c) Luštění se znalostí otevřeného textu – útočník získal nejen šifrované zprávy, ale i k nim odpovídající otevřený text. Pak už stačí najít jen šifrovací klíč. Vyberou se páry zašifrovaného textu takové, u kterých jejich odpovídající páry otevřeného textu vykazují nějaké rozdíly. Zkoumá se, jak se tyto rozdíly během šifrovacího algoritmu mění. Analyzováním velkého počtu párů lze zjistit správný klíč. Tato metoda se nazývá diferenciální kryptoanalýza [7].

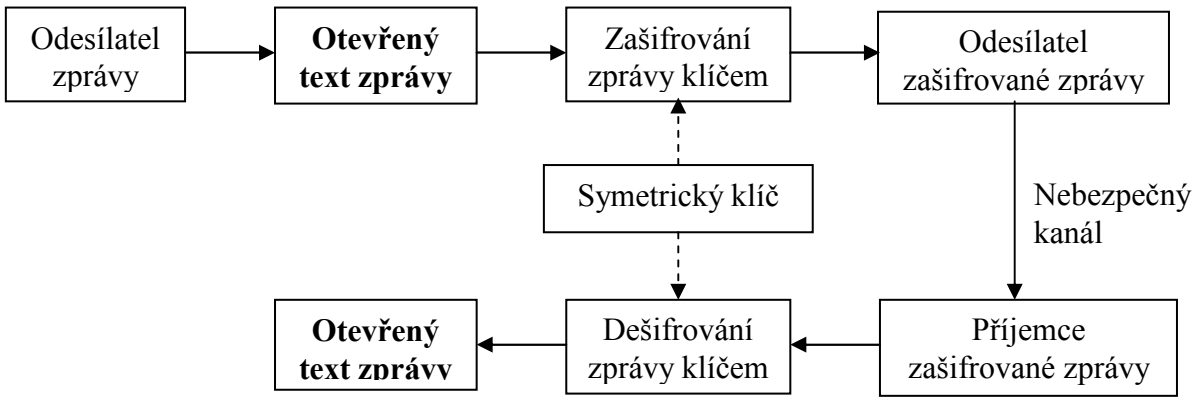
d) Luštění se znalostí vybraných otevřených textů – útočník si může vybrat otevřený text a jemu odpovídající zašifrovaný text. Pravděpodobnost, že získá klíč, je zde mnohem větší [7].

3 ROZDĚLENÍ ŠIFER

3.1 Symetrické šifry

Pro šifrování i dešifrování se používá stejný klíč. Pomocí šifrovací funkce a klíče se otevřený text převede na kód, který se odešle příjemci. Ten pomocí dešifrovací funkce a stejného klíče získá původní otevřený text. Tedy odesílatel i adresát vlastní stejný klíč. Proto je třeba zajistit bezpečné doručení klíče (např. po zabezpečeném přenosovém kanálu) [7].

Pro šifrování se využívají funkce, u kterých je i při znalosti vstupního a zakódovaného textu velmi těžké získat klíč. Bezpečnost zde závisí hlavně na délce klíče. V dnešní době se používají klíče, které mají délku 128 bitů [7].



Obr. 2 – Symetrická šifra [7]

Symetrické šifry můžeme rozdělit na 2 kategorie podle toho, jak zpracovávají otevřený text: [8]

a) Proudové šifry - šifrují jednotlivé bity a mohou tak zpracovávat zprávu libovolné délky. Proudové šifry využívají náhodný generátor, který podle symetrického klíče generuje výstupní hodnoty. Výstup z generátoru je pak kombinován s otevřeným textem. Často se zde využívá funkce XOR [7].

b) Blokové šifry - šifrují celý blok dat. Velikost bloku bývá obvykle 64 bitů a má velký vliv na bezpečnost algoritmu. Pokud by byla velikost bloku příliš malá, bylo by pak při daném klíči možné vytvořit slovník vstupních a výstupních hodnot algoritmu, a tím by se narušila bezpečnost celého algoritmu. V rámci normy, která se připravuje pro 21. století budou algoritmy zpracovávající bloky o délce 128 bitů [7].

Symetrické šifry jsou rychlé a nenáročné na výpočetní výkon. Používají se i pro zašifrování dat, která se nikam neposílají. Například pro zašifrování dat na počítači.

Velkou nevýhodou je, že je třeba si bezpečným kanálem předávat klíč, což může být někdy velmi obtížné. Další nevýhodou je velký počet klíčů. Pro počet klíčů N pro n osob platí vztah:

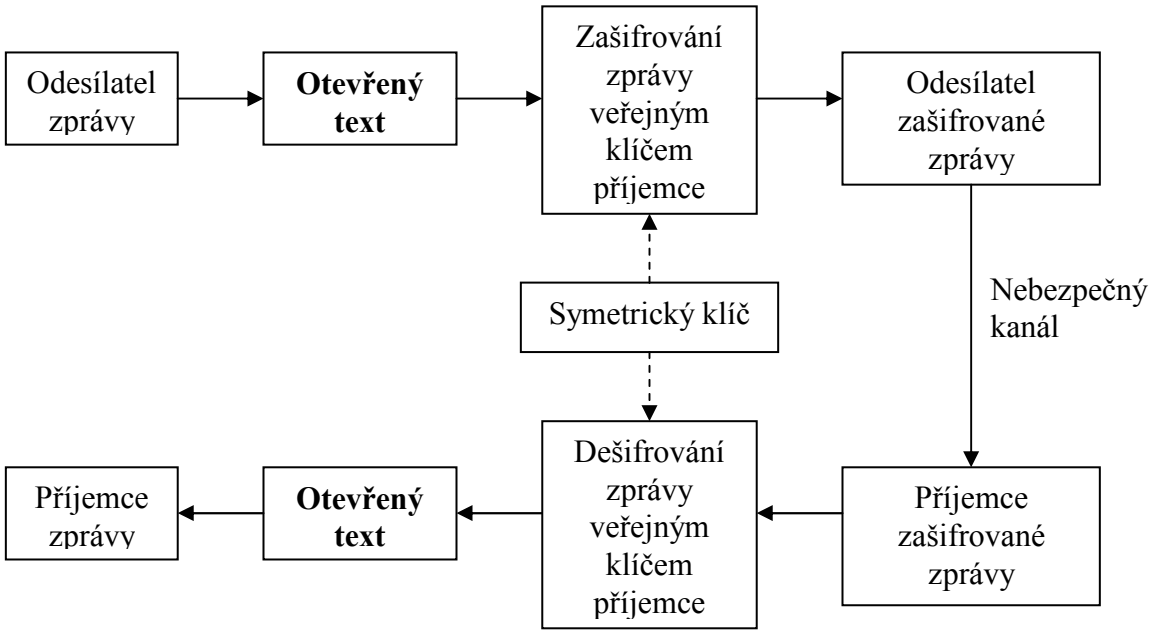
$$N = \frac{n \cdot (n-1)}{2} \tag{1}$$

Mezi symetrické šifry patří: algoritmus DES (klíč o délce 56 bitů), jeho vylepšená verze 3DES (klíč o délce 168 bitů), IDEA (klíč o délce 128 bitů), BlowFish, Cast [7].

3.2 Asymetrické šifry

Též označované jako algoritmy veřejného klíče [8].

U asymetrických šifer se používá jiný klíč pro šifrování a jiný klíč pro dešifrování. Tyto klíče se dohromady nazývají párem klíčů. Pár klíčů je většinou generován současně. Pomocí veřejného klíče se zpráva zašifruje a soukromým klíčem se zpráva dešifruje [7].



Obr. 3 – Asymetrická šifra [7]

Výhodou je, že není třeba posílat tajný klíč a riskovat tak jeho vyzrazení. Také je potřeba méně klíčů než u symetrických šifer. Pro jednu osobu postačuje jeden pár klíčů. Asymetrické šifry jsou však hodně pomalé. V porovnání se symetrickými šiframi jsou asi 100-krát pomalejší [7].

Proto se často používá kombinace obou druhů šifer tak, kdy jsou z každého druhu šifry využity její výhody. Zpráva je nejprve šifrována symetrickým klíčem, který je dále zašifrován asymetrickým klíčem. Při dešifrování je nejprve pomocí soukromého klíče získán šifrovaný symetrický klíč, kterým je pak dešifrována celá zpráva. Toto řešení je méně náročné na výpočetní výkon a je zde dosažena poměrně velká bezpečnost přenosu zprávy [7].

3.3 Požadavky na délku klíčů

Délky klíčů u symetrických a asymetrických algoritmů jsou naprosto odlišné. Ekvivalentem 80-ti bitového symetrického klíče je 1024-bitový asymetrický klíč [1].

Pro bezpečný přenos dat jsou v současnosti požadovány tyto minimální délky klíčů:

a) U symetrických kryptografických mechanismů je minimální délka klíče 80 bitů. Mechanismy s délkou klíče 40 bitů jsou označovány jako nevyhovující.

b) U asymetrických kryptografických mechanismů závisí minimální délka na konkrétním algoritmu. Pro algoritmus RSA jsou tyto minimální délky klíčů:

- 768 bitů pro běžné klíče
- 1024 bitů pro klíče organizací
- 2048 bitů pro zvláště důležité klíče [7]

4 DIGITÁLNÍ PODPIS

Velkou výhodou asymetrické kryptografie je, že umožňuje implementaci digitálního podpisu. Digitální podpis slouží ke stejnému účelu jako běžný ruční podpis. Ruční podpis však lze jednoduše padělat. Digitální podpis je mnohem kvalitnější a jeho padělání je téměř nemožné [1].

Digitální podpis zajišťuje:

a) Autenticitu – důkaz, že odesílatel dokument skutečně podepsal. Jde o potvrzení totožnosti.

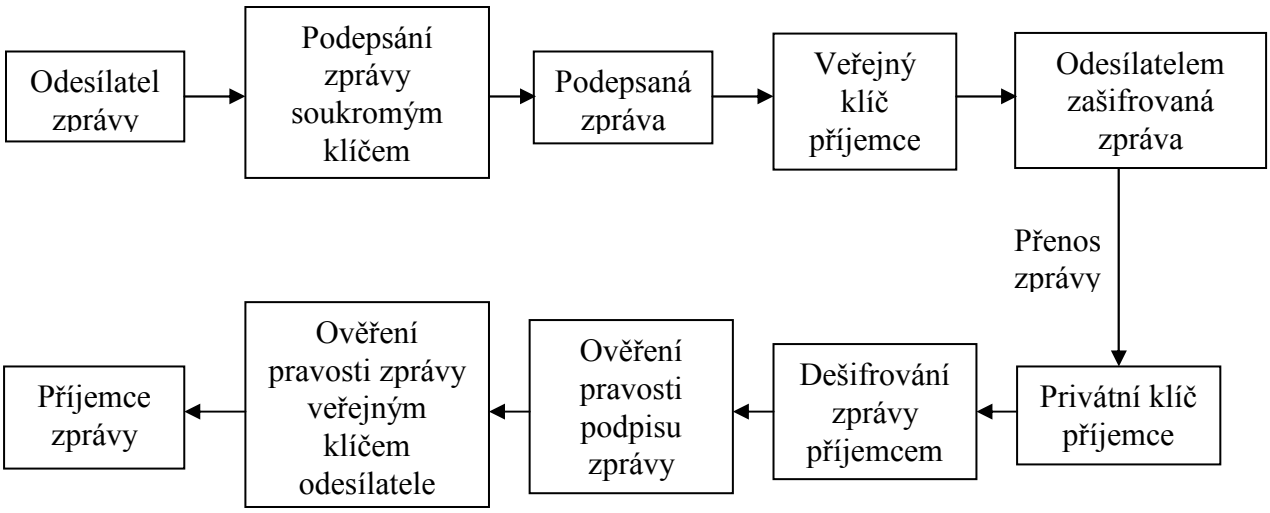
b) Integritu zprávy – po podpisu již nelze dokument upravovat.

c) Jednorázovost použití – podpis nelze aplikovat na jiný dokument.

d) Neodmítnutelnost zodpovědnosti – odesílatel se nemůže zbavit zodpovědnosti za dokument, který je podepsán jeho jménem. Podpis má právní závaznost a může sloužit jako důkaz v právním sporu [7].

4.1 Definice digitálního podpisu

Digitální podpis je binární číslo, jehož délka bývá až 4096 bitů. Jeho výpočet nebo ověření nelze provádět ručně, protože jde o složité matematické operace. Digitální podpis vzniká matematickým spojením 2 čísel. První číslo je část dokumentu získaná matematickým výpočtem a označuje se jako hash dokumentu. Druhé číslo se nazývá souhrnný elektronický klíč. Takto vzniklý digitální podpis se pak připojí k dokumentu [7].



Obr. 4 – Schéma šifrování a podpisu zprávy digitálním podpisem [7]

4.2 Vlastnosti digitálního podpisu

a) Elektronický podpis je propojen s konkrétním dokumentem a potvrzuje jak jeho původ, tak totožnost autora.

b) Může ho vytvořit pouze ten, kdo zná soukromý klíč.

c) Pravost podpisu může kdokoliv ověřit bez nutnosti znát soukromý klíč [7].

5 DIGITÁLNÍ CERTIFIKÁT

5.1 Co je to digitální certifikát

Digitální certifikát je informace připojená k veřejnému klíči, který patří nějaké osobě. Tento certifikát tak pomáhá ostatním ověřit, že tento veřejný klíč je platný. Digitální certifikáty se používají ve snaze zabránit nahrazování něčího klíče za jiný [1].

5.2 Jaké informace digitální certifikát obsahuje

- Verze certifikátu
- Sériové číslo certifikátu
- Použitý algoritmus pro digitální podpis
- Jméno certifikační autority
- Doba platnosti certifikátu
- Veřejný klíč
- Jméno a údaje o vlastníkovi certifikátu [2]

Certifikát dále obsahuje jeden nebo i více digitálních podpisů. Jejich smyslem je potvrdit, že informace které certifikát obsahuje, byly ověřeny nějakou osobou či objektem [1].

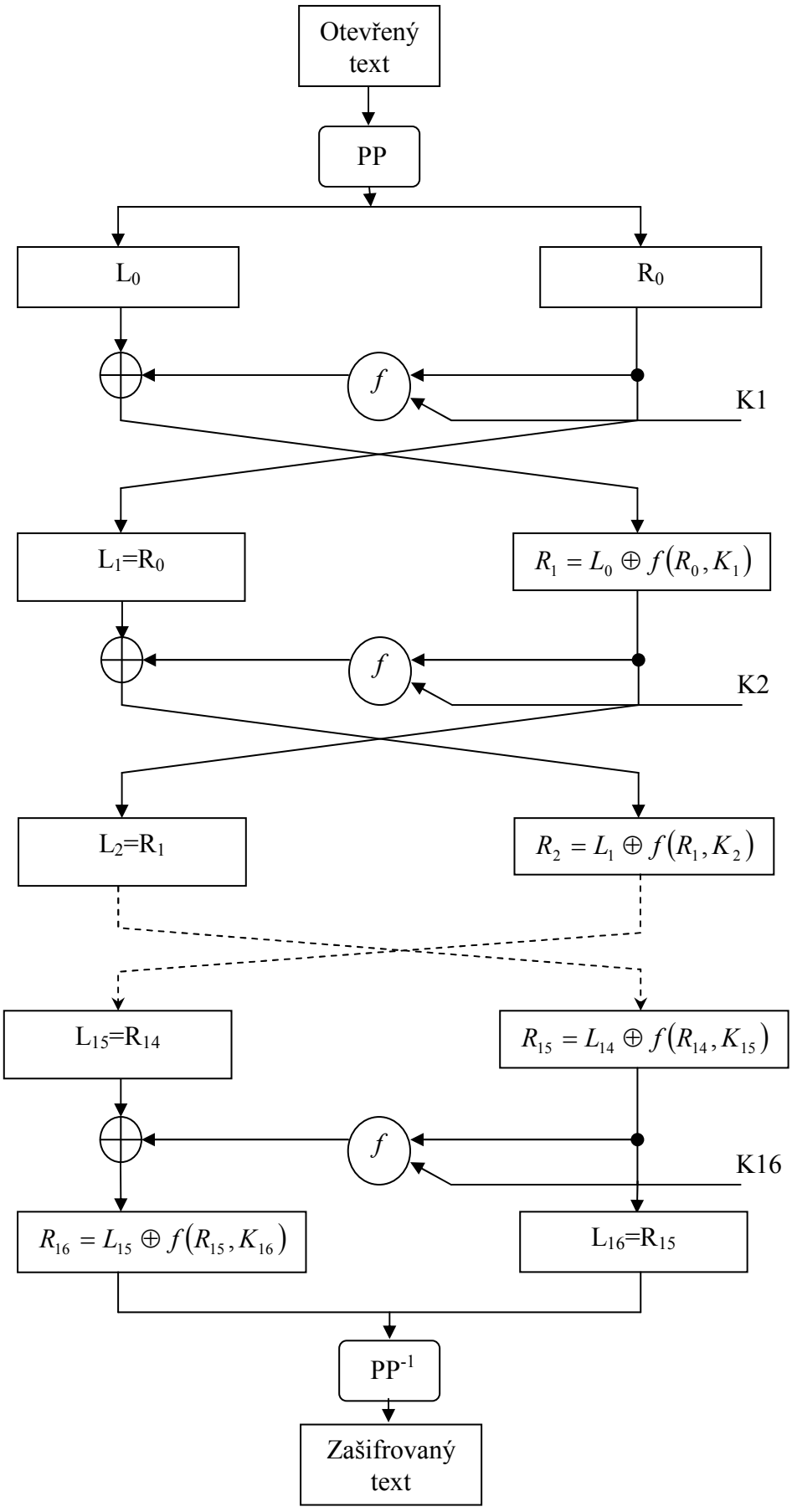
6 ŠIFROVACÍ NORMA DES

Šifrovací norma DES (Digital Encryption Standard) vznikla v roce 1975. Od roku 1981 se stala široce používaným standardem [2].

DES je symetrická bloková šifra. Pro šifrování a dešifrování se využívá stejného klíče. Délka bloku je 64 bitů. Na vstupu algoritmu je 64-bitový blok otevřeného textu. Na jeho výstupu pak dostáváme 64-bitový blok zašifrovaného textu. Klíč tvoří 64-bitové číslo, z něhož se však každý osmý bit ignoruje, protože slouží jako paritní zabezpečení. Využíváme tak pouze 56-ti bitů klíče. Existuje několik tzv. slabých klíčů, které by se měly vyloučit, protože veškerá bezpečnost algoritmu závisí na jeho klíči. Základní stavební částí algoritmu DES je substituce následovaná permutací, která se s pomocí klíče aplikuje na otevřený text. Tato celá část se nazývá runda. Algoritmus DES obsahuje 16 rund [8].

6.1 Popis algoritmu

Prvním krokem je počáteční permutace. Počáteční a konečná permutace nemá vliv na bezpečnost. Slouží pouze k jednoduššímu zavádění otevřeného a zašifrovaného textu do mikročipů DES. Poté je blok rozdělen na levou a pravou polovinu o délce 32 bitů. Pak následuje 16 rund, v nichž jsou prováděny identické operace dávající dohromady funkci f. Tyto operace kombinují klíč s otevřeným textem [8].

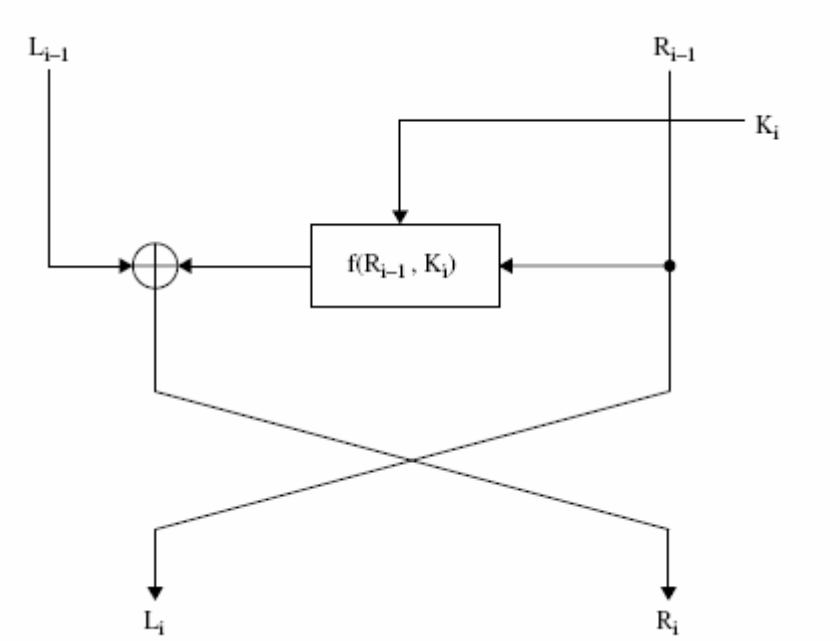


Obr. 5 – Schéma algoritmu DES [8]

6.2 Průběh rundy

Nejprve jsou posunuty bity klíče a následně je z něj vybráno 48 bitů. Pravá polovina dat je pomocí expanzní permutace rozšířena na 48 bitů a zkombinována pomocí funkce XOR s nově vzniklým 48-mi bitovým klíčem. Pak jsou data zpracována osmi S-boxy, které mají na výstupu dohromady 32 bitů. Tyto bity jsou pak dále permutovány a vzniká funkce f [8].

Výstup z této funkce je pomocí funkce XOR zkombinován s levou polovinou. Výsledkem je nová pravá polovina. Novou levou polovinu dostáváme ze staré pravé poloviny. Tento postup se opakuje 16-krát a tvoří 16 rund algoritmu DES [8].



Obr. 6 – Jedna runda algoritmu DES [4]

Rundu lze vyjádřit i takto:

$$\begin{aligned} L_i &= R_{i-1} \\ R_i &= L_{i-1} \oplus f\left(R_{i-1}, K_i\right) \end{aligned} \tag{2}$$

Kde:

L_i je levá polovina dat v i-tém kroku

R_i je pravá polovina dat v i-tém kroku

K_i je klíč v i-tém kroku

f je funkce popsaná výše [8]

6.2.1 Transformace klíče

Vyřazením každého osmého bitu dostaneme 56 bitů klíče. Těchto 56 bitů se rozdělí na dvě 28-mi bitové poloviny. Tyto poloviny se posunou rotací o jeden nebo dva bity vlevo podle aktuální rundy [8].

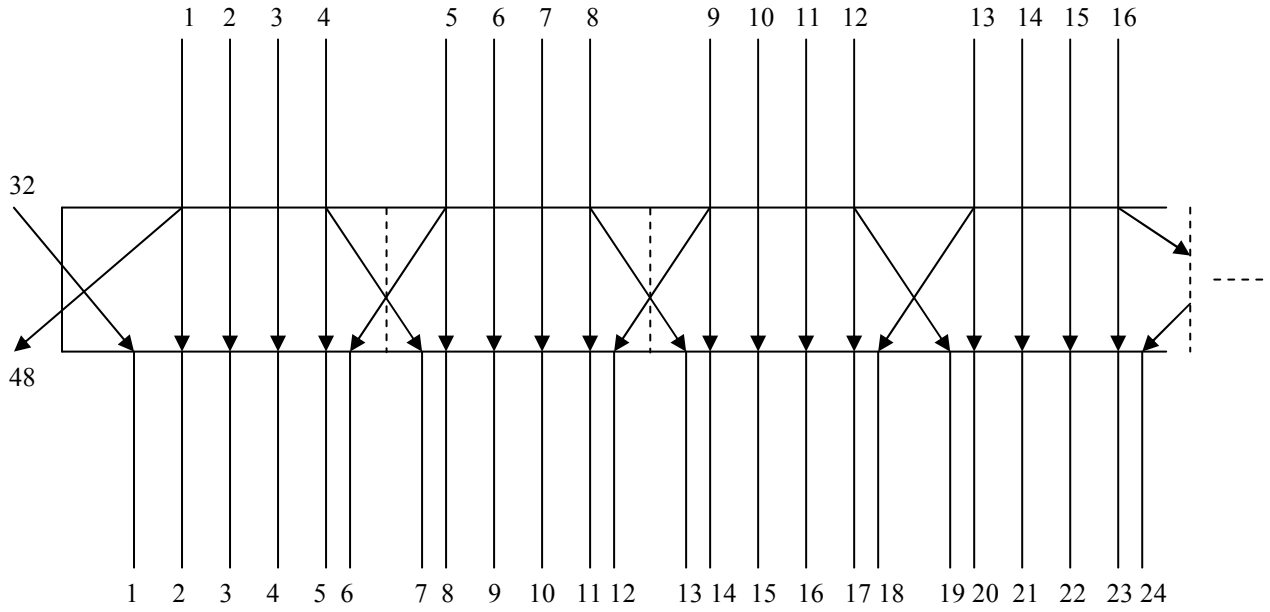
Tab. 1 – Rotace klíče na základě aktuální rundy [8]

Runda	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Počet	1	1	2	2	2	2	2	2	1	2	2	2	2	2	2	1

Poté se provede kompresní permutace, kdy se mění uspořádání bitů a zároveň se vybere z klíče 48 bitů, se kterými se dále pracuje. V každé rundě je tak pomocí posunů vytvořena jiná podmnožina bitů klíče [8].

6.2.2 Expanzní permutace

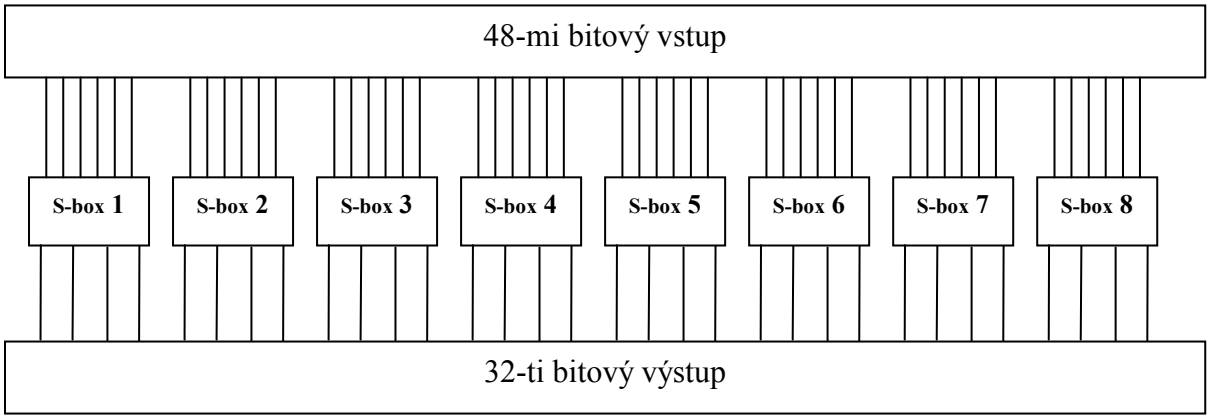
Expanzní permutace (někdy nazývaná E-box) rozšiřuje pravou polovinu dat z 32 na 48 bitů. Každému prvnímu a čtvrtému vstupnímu bitu odpovídají dva výstupní bity. Každému druhému a třetímu vstupnímu bitu odpovídá jen jeden výstupní bit. Pravá polovina dat tak bude mít stejnou délku jako klíč. Tato operace však nemá žádný kryptografický účel [8].



Obr. 7 – Expanzní permutace [8]

6.2.3 S-box substituce

Pravá polovina dat se pomocí funkce XOR zkombinuje s klíčem a následuje S-box substituce. Substituce je vykonávána v osmi S-boxech neboli substitučních boxech. Každý S-box má 6 vstupů a 4 výstupy. Celkový výstup ze všech S-boxů pak bude mít 32 bitů [8].



Obr. 8 – Substituce v S-boxech [8]

S-box je vlastně tabulka, která má 4 řádky a 16 sloupců. Vstupní bity slouží jako index pro vyhledání výstupů v tabulce. Kombinace prvního a šestého bitu ukazuje řádek a prostřední bity 2-5 ukazují sloupec [8].

Tab. 2 – S-boxy [4]

		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	0	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
S_1	1	0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
	2	4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
	3	15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13
	0	15	1	8	14	6	11	3	4	9	7	2	13	12	0	5	10
S_2	1	3	13	4	7	15	2	8	14	12	0	1	10	6	9	11	5
	2	0	14	7	11	10	4	13	1	5	8	12	6	9	3	2	15
	3	13	8	10	1	3	15	4	2	11	6	7	12	0	5	14	9
	0	10	0	9	14	6	3	15	5	1	13	12	7	11	4	2	8
S_3	1	13	7	0	9	3	4	6	10	2	8	5	14	12	11	15	1
	2	13	6	4	9	8	15	3	0	11	1	2	12	5	10	14	7
	3	1	10	13	0	6	9	8	7	4	15	14	3	11	5	2	12
	0	7	13	14	3	0	6	9	10	1	2	8	5	11	12	4	15
S_4	1	13	8	11	5	6	15	0	3	4	7	2	12	1	10	14	9
	2	10	6	9	0	12	11	7	13	15	1	3	14	5	2	8	4
	3	3	15	0	6	10	1	13	8	9	4	5	11	12	7	2	14
	0	2	12	4	1	7	10	11	6	8	5	3	15	13	0	14	9
S_5	1	14	11	2	12	4	7	13	1	5	0	15	10	3	9	8	6
	2	4	2	1	11	10	13	7	8	15	9	12	5	6	3	0	14
	3	11	8	12	7	1	14	2	13	6	15	0	9	10	4	5	3
	0	12	1	10	15	9	2	6	8	0	13	3	4	14	7	5	11
S_6	1	10	15	4	2	7	12	9	5	6	1	13	14	0	11	3	8
	2	9	14	15	5	2	8	12	3	7	0	4	10	1	13	11	6
	3	4	3	2	12	9	5	15	10	11	14	1	7	6	0	8	13
	0	4	11	2	14	15	0	8	13	3	12	9	7	5	10	6	1
S_7	1	13	0	11	7	4	9	1	10	14	3	5	12	2	15	8	6
	2	1	4	11	13	12	3	7	14	10	15	6	8	0	5	9	2
	3	6	11	13	8	1	4	10	7	9	5	0	15	14	2	3	12
	0	13	2	8	4	6	15	11	1	10	9	3	14	5	0	12	7
S_8	1	1	15	13	8	10	3	7	4	12	5	6	11	0	14	9	2
	2	7	11	4	1	9	12	14	2	0	6	10	13	15	3	5	8
	3	2	1	14	7	4	10	8	13	15	12	9	0	3	5	6	11

Operace s S-boxy jsou nejdůležitější fází algoritmu. Tyto operace jsou jako jediné v algoritmu DES nelineární a proto zajišťují bezpečnost celého algoritmu [8].

6.2.4 P-box permutace

Výstupy z S-boxů jsou permutovány v P-boxech. Tato permutace bývá někdy označována jako přímá permutace. Výsledek permutace se pomocí funkce zkombinuje s počáteční levou polovinou dat a stává se z něj nová pravá polovina. Nová levá polovina je tvořena starou pravou polovinou. Tímto krokem končí jedna runda. Posledním krokem algoritmu, který následuje po provedení všech 16ti rund, je konečná permutace [8].

6.3 Dešifrování

Algoritmus DES je navržen tak, aby byl univerzální. Pro dešifrování tak používáme stejnou funkci, pouze musíme používat klíče v opačném pořadí [8].

6.4 Slabé a poloslabé klíče

Některé klíče jsou označovány jako slabé, protože i přes počáteční rozdělení na dvě poloviny a následnou rotaci zůstávají klíče beze změny po celou dobu algoritmu [8].

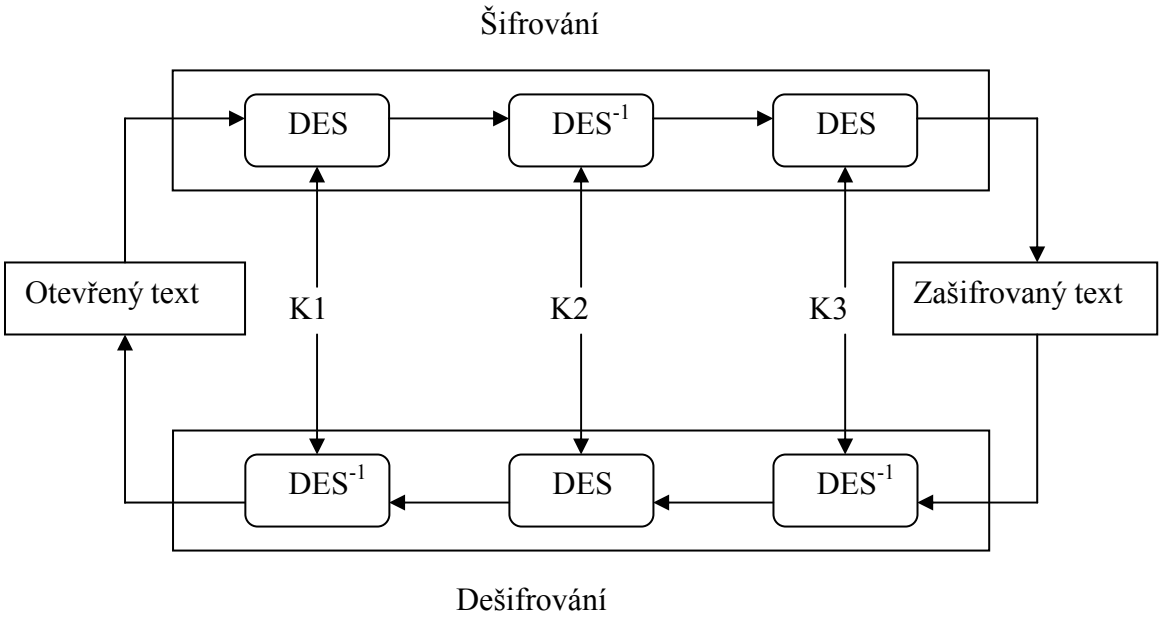
Tab. 3 – Slabé klíče DES v hexadecimálním vyjádření [8]

Hodnota slabého klíče (s paritními bity)				Skutečný klíč	
1010	1010	1010	1010	0000000	0000000
1F1F	1F1F	0E0E	0E0E	0000000	FFFFFFF
0E0E	0E0E	1F1F	1F1F	FFFFFFF	0000000
FEFE	FEFE	FEFE	FEFE	FFFFFFF	FFFFFFF

Poloslabé klíče jsou takové páry klíčů, které šifrují otevřený text na úplně stejný zašifrovaný text [8].

6.5 Trojnásobný DES (3DES)

Již dlouho předtím než byl DES prolomen bylo pracováno na jeho nástupci. Mnoho softwaru a hardwaru bylo vytvořeno pro původní DES. Z důvodu kompatibility bylo výhodnější vylepšit původní algoritmus než vytvořit zcela nový. 3DES využívá tří klíčů k zašifrování bloku otevřeného textu. Otevřený text je postupně šifrován třemi 56-ti bitovými klíči [2].



Obr. 9 – Trojnásobný DES [8]

Odesílatel zašifruje otevřený text pomocí klíče K1, pak dešifruje pomocí klíče K2 a poté opět zašifruje pomocí klíče K3. Při dešifrování je třeba použít klíče v obráceném pořadí [4].

Počet pokusů při útoku hrubou silou tak vzrostl z 2^{56} na 2^{112}. Tím byla značně zvýšena bezpečnost algoritmu [8].

7 ALGORITMUS RSA

Algoritmus RSA je pojmenovaný po svých tvůrcích – Ronu Rivestovi, Adi Shamirovi a Leonardu Adlemanovi. RSA je ze všech algoritmů veřejného klíče tím nejjednodušším, nejsrozumitelnějším a přesto skutečně bezpečným [8].

Algoritmus byl vymyšlen v roce 1977 a používá se jak pro šifrování, tak pro digitální podpisy [4].

Algoritmus RSA je založen na faktorizaci velkých čísel. Veřejný a soukromý klíč je generován ze dvou obrovských (až několika set místných) prvočísel [8].

7.1 Popis algoritmu

Obsah této kapitoly byl převzán z [4] a upraven.

- Nejprve se náhodně zvolí dvě velká prvočísla p a q.
- Poté se vypočte číslo n

$$n = pq \tag{3}$$

- Zvolí se šifrovací klíč e tak, aby platilo:

$$NSD(e, \varphi(n)) = 1 \tag{4}$$

kde: $\varphi(n) = (p-1)(q-1)$

- Pomocí rozšířeného Euklidova algoritmu se vypočte dešifrovací klíč d.

$$d = e^{-1} \bmod(\varphi(n)) \tag{5}$$

Tuto rovnici lze také zapsat jako:

$$ed = 1 \bmod(\varphi(n)) \tag{6}$$

Čísla e a n slouží jako veřejný klíč, číslo d pak jako soukromý klíč [4].

- Výraz pro šifrování, kde c_i jsou jednotlivé bloky zprávy:

$$c_i = m_i^e \bmod n \tag{7}$$

- Pro dešifrování pak platí:

$$m_i = c_i^d \bmod n \tag{8}$$

Pro jednotlivé bloky zprávy m_i musí platit $m_i < n$. Čísla p a q uschováme a také je nesmíme zveřejnit. Pro větší bezpečnost se volí tyto čísla přibližně stejně velká. Zprávu lze stejně dobře zašifrovat i klíčem *d* a dešifrovat klíčem *e* [8].

7.2 Vlastnosti algoritmu

Číslo *n* by mělo být větší než 129 míst, protože současná technologie nedokáže rozložit více jak 129-ti místný dekadický modul [8].

Pokud by se někdo pokoušel prolomit algoritmus pomocí útoku hrubou silou, tak tato metoda je ještě méně efektivní než se pokusit faktorizovat číslo *n* [4].

RSA je však asi 1000-krát pomalejší než DES při hardwarové realizaci a 100-krát pomalejší při softwarové realizaci [4].

Algoritmus se dá urychlit vhodnou volbou hodnoty *e*. Nejpoužívanější jsou hodnoty 3, 17, 65537. Číslo 65537 totiž obsahuje v binárním vyjádření pouze 2 jedničky, čímž se urychluje výpočet [8].

8 EL GAMALŮV SYSTÉM

El Gamalův systém patří také mezi algoritmy veřejného klíče a byl navržen roku 1985. Algoritmus může být použit jak pro šifrování, tak pro digitální podpis [4].

Bezpečnost tohoto algoritmu závisí na složitosti výpočtu diskrétních logaritmů [8].

8.1 Popis algoritmu

Obsah této kapitoly byl převzán z [4] a upraven.

Uvedený algoritmus se používá pro šifrování.

- Nejprve se zvolí prvočíslo p a dvě náhodná čísla g a x tak, aby $g < p$ a $x < p$
- Poté se dosadí do vzorce:

$$y = g^x \pmod p \tag{9}$$

- Veřejný klíč je tvořen čísly (y,g,p) a soukromým klíčem je číslo x.

  K zašifrování zprávy m, je třeba si zvolit náhodné číslo k tak, aby platilo:

$$NSD(k, p-1) = 1 \tag{10}$$

- Pro délku zprávy platí:

$$0 \le m \le p-1 \tag{11}$$

- Zašifrovanou zprávu tvoří následující pár (r,s):

$$\begin{aligned} r &= g^k \pmod p \\ s &= \left(y^k \pmod p\right)\left(m \pmod{p-1}\right) \end{aligned} \tag{12}$$

- Pro dešifrování je třeba vypočíst:

$$m = s / r^x \pmod p \tag{13}$$

Nevýhodou tohoto systému je, že zašifrovaný text má dvojnásobnou délku jako otevřený text. Délka prvočísla *p* se doporučuje nejméně 768 bitů. Pro dlouhodobější účely pak 1024 bitů i více [5].

8.2 Ukázka šifrování pomocí El Gamalova systému

- Zvolíme si prvočíslo $p = 11$ a čísla $g = 4$, $x = 8$. Obě čísla jsou menší než prvočíslo *p*.
- Poté vypočteme $y = g^x \pmod p = 4^8 \pmod{11} = 9$
- Veřejný klíč je tvořen čísly $y = 9$, $g = 4$, $p = 11$.

  Soukromý klíč je tvořen číslem $x = 8$.
- Chceme zašifrovat zprávu $m = 5$. Zvolíme si náhodné číslo $k = 7$. Číslo *k* nesmí mít žádné společné součinitele s číslem $(p - 1)$.
- Vypočteme pár (*r*,*s*):

  $r = g^k \pmod p = 4^7 \pmod{11} = 5$

  $s = \left(y^k \pmod p\right)\left(m \pmod{p-1}\right) = \left(9^7 \pmod{11}\right)\left(5 \pmod{10}\right) = 4 \cdot 5 = 20$

  Pár (*r*,*s*) tvoří zašifrovaný text.
- Výpočtem $m = s / r^x \pmod p = 20 / 4 = 5$ dostáváme zpět původní zprávu [4].

9 ZÁKLADY JAZYKA HTML

9.1 Pravidla pro zápis tagů

Jazyk HTML je založen na značkách, které se nazývají tagy. Díky těmto tagům pak příslušný webový prohlížeč pozná, jak zobrazit obsah stránky.

Základní pravidla pro zápis tagů:

- Začátek prvku se označí otevíracím tagem tak, že se napíše příslušná zkratka uzavřená z obou stran symboly „<“ a „>“.
- Konec prvku se označí tzv. uzavíracím tagem. Uzavírací tag se liší od otevíracího tagu pouze přidáním lomítka před zkratku. Ne všechny tagy mají příslušný uzavírací tag. Příkladem může být tag `<br>` pro zalomení řádku, pro který neexistuje tvar `</br>`.
- Atributy tagu se vpisují do otevíracího tagu oddělené mezerou. Hodnota atributu se vkládá za znak “=“ a je uzavřena uvozovkami. Uzavírací tag nemá žádné atributy [3].

9.2 Struktura HTML dokumentu

Všechny HTML dokumenty mají stejnou výchozí strukturu, která tvoří základ stránky:

```
<html>
     <head>
          <title>Název stránky</title>
     </head>
     <body>
          Obsah
     </body>
</html>
```

- tag `<html>` vymezuje začátek a konec dokumentu a poznáme podle něj, že jde o HTML dokument.
- tag `<head>` definuje hlavičku. Hlavička obsahuje informace o dokumentu, které nebudou zobrazeny webovým prohlížečem.
- tag `<title>` určuje název dokumentu, zobrazuje se v záhlaví stránky.
- tag `<body>` určuje tělo dokumentu [3].

9.3 Nadpisy

V jazyku HTML je definováno 6 úrovní nadpisů od největšího `<h1>` až po nejmenší `<h6>`. Jediný možný atribut u nadpisů je vlastnost `align` [3].

9.4 Obrázky

Při použití tagu `<img>` vloží webový prohlížeč do dokumentu obrázek, který specifikujeme v tagu `<img>` pomocí atributu `src`. Další atributy tagu `<img>` jsou:

- `width` – šířka obrázku v pixelech
- `height`- výška obrázku v pixelech
- `border` – tloušťka rámečku kolem obrázku v pixelech
- `alt` – alternativní popis. Tento atribut se hodně používal v minulosti. Rychlosti připojení byly mnohem pomalejší než dnes a stahování obrázků bylo časově náročné. Pomocí alternativního popisu získal uživatel základní informace o obrázku, aniž by ho musel zobrazovat. Nyní alternativní popis slouží zrakově postiženým, kteří k prohlížení Internetu používají mluvící prohlížeče [3].

9.5 Odkazy

Odkazy jsou neodmyslitelnou součástí webové stránky. Vytvoří se pomocí tagu `<a>`. Obsah mezi tagy `<a>` a `</a>` je přeměněn na oblast, která po kliknutí odkazuje na dokument specifikovaný atributem `href`. Mezi tagy `<a>` a `</a>` se může nacházet text nebo obrázek [3].

9.5.1 Způsoby definování adresy odkazu

Obsah této kapitoly byl převzán z [3] a upraven.

- Odkaz na dokument, který je ve stejném adresáři jako aktuální dokument. Hodnota atributu `href` bude stejná jako název souboru, na který chceme odkázat.

```
<a href="strana2.html">Odkaz</a>
```

- Odkaz na dokument, který je umístěn ve složce, která je ve stejné složce jako aktuální dokument. Hodnota atributu `href` bude název složky a za lomítkem následuje název souboru.

```
<a href="složka/strana2.html">Odkaz</a>
```

- Odkaz na dokument, který je umístěn o úroveň výš v souborové struktuře.

```
<a href="../strana2.html">Odkaz</a>
```

- Relativní odkaz, který odkazuje na dokument umístěný na jiném webovém serveru.

```
<a href="http://www.seznam.cz/">odkaz</a>
```

9.6 Tabulky

Tabulka je uspořádaný prvek, složený z řádků a sloupců, které obsahují buňky. Obsah buněk může být text, obrázek nebo i další tabulka. Tagy `<table>` a `</table>` určují, kde začíná a končí tabulka. Tagy `<tr>` a `</tr>` určují řádek a tagy `<td>` a `</td>` určují buňky uvnitř řádku. Tag pro sloupce není definován [3].

Vytvoření tabulky o dvou řádcích a dvou sloupcích:

```
<table>
      <tr><td>buňka1</td><td>buňka2</td></tr>
      <tr><td>buňka3</td><td>buňka4</td></tr>
</table>
```

9.7 Číslovaný seznam

Číslovaný seznam vytvoříme pomocí tagu `<ol>` a položky seznamu tagem `<li>`. Oba tagy jsou párové. Základní nastavení číslování je arabskými číslicemi. Pomocí atributu `type` může nastavit různé druhy číslování:

- `type="A"` – označení položek velkými písmeny
- `type="a"` – označení položek malými písmeny
- `type="I"` – číslování velkými římskými číslicemi
- `type="i"` – číslování malými římskými číslicemi [3]

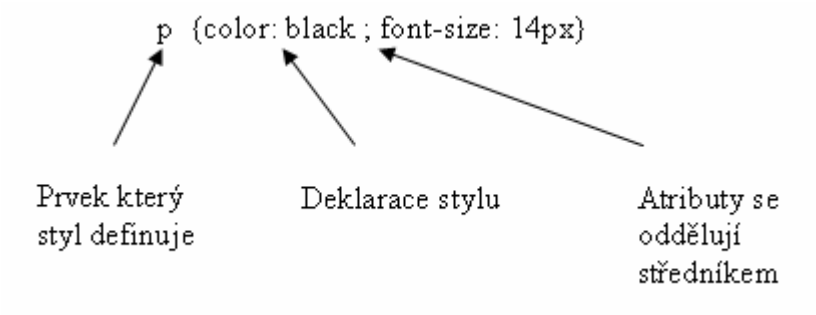
9.8 Nečíslovaný seznam

Nečíslovaný seznam vytvoříme pomocí tagu `<ul>` a položky seznamu opět tagem `<li>`. Počet položek stejně jako u číslovaného seznamu není nijak omezen. Pomocí atributu `type` lze nastavit druh odrážek:

- `type="square"` – odrážky budou ve tvaru čtverečku
- `type="circle"` – odrážky budou ve tvaru kolečka
- `type="disc"` – odrážky budou ve tvaru vyplněného kolečka [3]

10 KASKÁDOVÉ STYLY CSS

Smyslem kaskádových stylů CSS (Cascading Style Sheets) je oddělit styl od struktury dokumentu. Od vzniku CSS je doporučováno nepoužívat HTML k úpravě vzhledu a použít kaskádové styly. CSS má odlišnou syntaxi než HTML [3].



Obr. 10 - Zápis vlastností odstavce pomocí CSS

10.1 Možnosti zápisu CSS stylů

- Přímo v html kódu pomocí atributu `style`:

```
<p style="color: black ; font-size: 14px">
```

- V hlavičce HTML dokumentu pomocí tagu `<style>`.
- Pomocí externího stylopisu. Pokud zapíšeme definici stylů přímo do HTML dokument, pak jsou tyto styly použitelné jen v rámci jednoho dokumentu. Výhoda externího stylopisu je, že můžeme jednoduše definovat styl pro několik HTML dokumentů.

  Nejprve musíme vytvořit soubor s příponou *.css* a do něj umístíme všechny definice stylů. Do hlavičky každého HTML dokumentu, kterému chceme přiřadit daný styl, vložíme odkaz na vytvořený CSS soubor:

```
<link rel="stylesheet" type="text/css" href="styly2.css">
```

  Úprava takto vytvořených stylů je velmi jednoduchá, protože stačí upravit CSS soubor a změna se projeví ve všech dokumentech, které mají ve své hlavičce odkaz na daný CSS soubor [3].

10.2 Definování tříd stylů

Pomocí třídy si vytvoříme vlastní prvek, kterému přiřadíme definici stylu. Třídy mohou být aplikovány k jakémukoliv tagu pomocí atributu `class`. Vytvoření třídy popisek:

```
.popisek {font-style: italic;text-align: center}
```

Přiřazení třídy popisek k tagu `<div>` :

```
<div class=”popisek”>
```

Třídu můžeme také definovat jen pro použití s určitým tagem. Následující příklad demonstruje definici třídy, která může být aplikována jen na odstavce:

```
p.popisek {font-style: italic;text-align: center}
```

Následujícím způsobem můžeme definovat ID třídu a pak ji aplikovat na tag pomocí atributu ID:

```
#menu {width: 226px ; float: left}
<div ID=”menu”> </div>
```

ID třída však může být aplikována pouze na jeden prvek HTML dokumentu [3].

10.3 Písma

Písmo nastavíme pomocí vlastnosti `font-family`. Můžeme zde zadat i několik druhů písem za sebou. V případě, že návštěvník stránky nemá písmo nainstalované, prohlížeč použije další písmo ze seznamu.

Velikost písma upravíme vlastností `font-size`. Velikost můžeme zadat v příslušných jednotkách nebo pomocí klíčových hodnot: `xx-small`, `x-small`, `small`, `medium`, `large`, `x-large` a `xx-large`.

Nastavení druhu a velikosti písma :

```
h1 { font-family: Arial, Helvetica, sans-serif ; font-size:
                                medium }
```

10.4 Barvy

Obsah této kapitoly byl převzán z [3] a upraven.

Barvu prvku nastavíme pomocí vlastnosti `color`. Barvu pozadí prvku pak pomocí vlastnosti `background-color`. Hodnotu barvy lze zapsat několika způsoby:

- hexadecimálním vyjádřením:

```
h2  {color: #FF0000}
```

- pomocí předdefinovaných názvů:

```
h2 { color: red }
```

- pomocí RGB modelu:

```
h2 { color: rgb(255, 0, 0) }
```

10.5 Vlastnost padding

Vlastnost `padding` slouží ke zvětšení nebo zmenšení vnitřního okraje prvku. Narozdíl od jazyka HTML, kde se hodnota vlastnosti `padding` mění na všech stranách zároveň, je u CSS možnost nastavit šířku okraje každé strany zvlášť. Nulová hodnota znamená zrušení okrajů.

Pomocí vlastností `padding-top`, `padding-right`, `padding-bottom` a `padding left` lze nastavit jednotlivé okraje zvlášť. Použitím vlastnosti `padding` a zápisem všech 4 hodnot dosáhneme úpravy okrajů v pořadí : vrchní okraj, pravý okraj, spodní okraj a levý okraj [3].

Nastavení vybraného okraje: `p { padding-top: 100px}`

Nastavení všech 4 okrajů: `p { padding: 10px 20px 30px 40px}`

10.6 Vlastnost margin

Vlastnost `margin` slouží ke zvětšení nebo zmenšení vnějšího okraje prvku. Pomocí vlastností `margin-top`, `margin-right`, `margin-bottom` a `margin-left` lze nastavit jednotlivé okraje zvlášť. Použitím vlastnosti `margin` a zápisem všech 4 hodnot dosáhneme úpravy okrajů v pořadí : vrchní okraj, pravý okraj, spodní okraj a levý okraj [3].

10.7 Vlastnost float

Pomocí vlastnosti float nastavujeme obtékání prvku.

Hodnoty vlastnosti float:

- `right` – zarovná prvek k pravému okraji a text bude obtékat z levé strany
- `left` – zarovná prvek k levému okraji a text bude obtékat z pravé strany
- `none` – zruší obtékání prvku [3]

II. PRAKTICKÁ ČÁST

11 POPIS TVORBY WEBOVÉHO PRŮVODCE

11.1 Design webových stránek

Stránky byly navrženy tak, aby působily jednoduše, přehledně a přesto zaujaly svým designem. Pro dobrou čitelnost textu jsem zvolil bílé pozadí a poměrně velké rozměry stránky. Položky grafického menu reagují na přejetí kurzoru změnou barvy.

Obr. 11 - Design webového průvodce

11.2 Struktura stránek

Základní stránka _Index_.html obsahuje všechny důležité součásti jako hlavičku, grafické menu a rám, ve kterém se zobrazuje obsah. Ostatní stránky už pak tvoří konkrétní obsah bez grafických prvků.

Hlavička, menu i rám s obsahem jsou každý zvlášť definovány tagem `<div>`, na který je aplikována příslušná třída. Položky menu odkazují vždy na první stránku dané kapitoly. Pohyb v rámci kapitoly pak realizují šipky v pravém horním rohu.

11.3 Meta tagy

V hlavičce každého HTML souboru je pomocí meta tagu nastaveno správné kódování češtiny:

```
<meta http-equiv="content-type" content="text/html;
charset=windows-1250">
```

Následující meta tag ukazuje, že stránka byla vytvořena v programu PSPad:

```
<meta name="generator" content="PSPad editor, www.pspad.com">
```

11.4 Užití tříd kaskádových stylů

Při tvorbě jsem využil externího stylopisu kaskádových stylů. Definice tříd a stylů je tak ve dvou zvláštních souborech. Styly pro stránku _Index_.html jsou umístěny v souboru styly.css a styly pro obsah stránek jsou v souboru styly2.css. V každé stránce jsou pak aplikovány pomocí odkazu na externí soubor:

```
<link rel="stylesheet" type="text/css" href="styly.css">
```

Toto řešení je velmi přehledné a následná úprava stylů jednotlivých prvků je velmi jednoduchá.

Každý prvek má svůj styl definovaný pomocí třídy kaskádového stylu. Například třída `ram` definuje pozadí, na kterém se zobrazuje obsah, šířku prvku a zarovnání vpravo. Okraje `padding` a `margin` jsou nastaveny na nulovou hodnotu a rámeček též nebude vykreslen:

```
.ram {
background:url(bg.jpg);
width: 774px;
float: right;
margin: 0;
padding: 0;
border: 0;
}
```

11.5 Popis menu

Po spuštění stránek je výchozí pozice v menu první kapitola s nadpisem kryptografie.

Změna barvy položky při přejetí kurzorem je realizována pomocí tříd. Každá položka menu má svou třídu, ve které je definován obrázek jako grafické pozadí tlačítka. Obrázek má však dvojnásobnou šířku než menu, protože obsahuje i vzhled při přejetí kurzorem. Pomocí pseudotřídy odkazu `.hover` je nastaveno, aby se při přejetí kurzoru obrázek posunul o polovinu. Tímto je docíleno reakce menu na kurzor bez nutnosti nahrávat další obrázek. Pro správnou funkčnost je třeba aplikovat posunutí obrázku i pro pseudotřídu `.active`.

```
.zaklad a {
   background: url(polozka1.jpg);
   display:block;
   height: 44px;
     width: 212px;
   }
.zaklad a:hover {
   background-position: -212px 0;
   }
.zaklad a:active {
   background-position: -212px 0;
   }
```

Obr. 12 - Obrázek sloužící jako pozadí položky základní pojmy

Posunutí obrázku namísto nahrávání nového je výhodnější, protože při prohlížení stránek není třeba čekat na stažení nového obrázku a reakce na pohyb kurzoru je okamžitá.

11.6 Zobrazení obsahu

Pomocí tagu `<iframe>` je vytvořen vnořený rám, ve kterém se zobrazují jednotlivé stránky s obsahem. Tag `<iframe>` má definovaný název pomocí vlastnosti `name`. Tento název je pak využit pro zacílení odkazů z menu.

Pozadí rámu je definováno v třídě `.ram`. Okraje zobrazovaného textu pak v třídě `.text`:

```
.text {
     margin: 50px 50px 0px 65px;
     padding: 0;
  border: 0;
     }
```

Pohyb v rámci kapitoly je realizován pomocí odkazů s obrázkem šipky. Na poslední stránce se objeví pouze šipka pro pohyb zpět a naopak.

Vytvoření odkazů pro pohyb v rámci kapitoly:

```
<div align="right"><a href="Des1.html"><img
src="sipka_leva.jpg" alt="předchozí strana"></a><a
href="Des3.html"><img src="sipka_prava.jpg" alt="další
strana"></a></div>
```

Na stránkách jsou použity nadpisy úrovní `<h1>` až `<h3>`. Nadpis `<h1>` slouží jako nadpis kapitoly, nadpis `<h2>` jako klasický nadpis a `<h3>` jako podnadpis.

Všechny nadpisy včetně obrázků jsou vycentrovány.

Styly zobrazovaného obsahu jsou definovány v souboru styly2.css:

```
h1  {color: #000000;font-weight:bold;text-align: center}
h2  {color: #0099CC;font-weight:bold;text-align: center}
h3  {color: #0099CC;font-weight:bold;text-align: center}
p   {text-align: justify}
div.popisek {font-style: italic;text-align: center}
img{borderstyle:none;color:white;border:0px;margin:0px;border
-width: 0px;border:0}
div.hlavni  {width:620px}
```

11.7 Zdrojový kód hlavní stránky

Pro doplnění zde uvádím zdrojový kód hlavní stránky _Index_.html:

```
!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01
Transitional//EN">
<html>
  <head>
  <meta http-equiv="content-type" content="text/html;
   charset=windows-1250">
  <meta name="generator" content="PSPad editor,
   www.pspad.com">
  <link rel="stylesheet" type="text/css" href="styly.css">
  <title>Průvodce moderní kryptografií</title>
  </head>
<body>

<div class="hlavni">
 <div class="hlavicka">
 <img src="hedka.jpg">
 </div>
 <div class="menu">
  <div class="kryptografie"><a href="Kryptografie1.html"
  target="obsahframe"></a>
  </div>
  <div class="kryptoanalyza"><a href="Kryptoanalyza1.html"
  target="obsahframe"></a>
  </div>
  <div class="rozdeleni"><a href="Rozdeleni1.html"
  target="obsahframe"></a>
  </div>
  <div class="podpis"><a href="Podpis1.html"
  target="obsahframe"></a>
  </div>
  <div class="certifikat"><a href="Certifikat1.html"
  target="obsahframe"></a>
  </div>
```

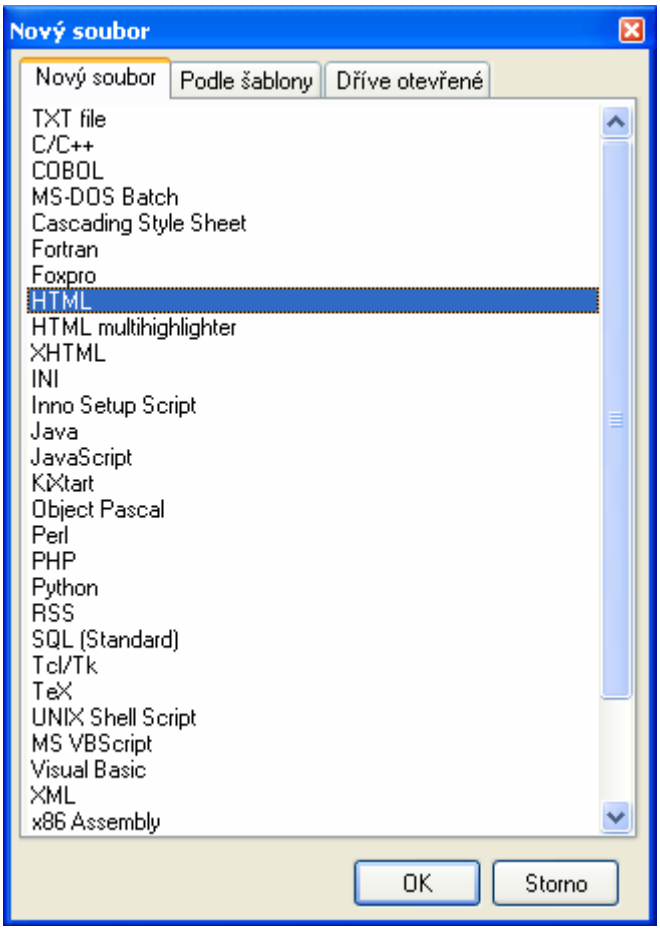
```
  <div class="des"><a
   href="Des1.html"target="obsahframe"></a>
  </div>
  <div class="rsa"><a
   href="Rsa1.html"target="obsahframe"></a>
  </div>
  <div class="elgamal"><a href="Eg1.html"
   target="obsahframe"></a>
  </div>
  <div class="autor"><a href="Autor.html"
   target="obsahframe"></a>
  </div>
 </div>
 <div class="obsah">
  <div class="text">
  <iframe src="Kryptografie1.html" width="650" height="1200"
  frameborder="0" scrolling="no" name="obsahframe">
  </iframe>
  <br><br><br><br>
  </div>
 </div>
</div>
</body>
</html>
```

12 POPIS PROGRAMU PSPAD

PSPad je volně šiřitelný program, který umožňuje efektivní práci s textem, vytváření webových stránek nebo může sloužit jako vývojové prostředí pro vybraný programovací jazyk. Práce s programem PSPad je velmi snadná. Obsahuje mnoho funkcí pro rychlejší a efektivnější práci.

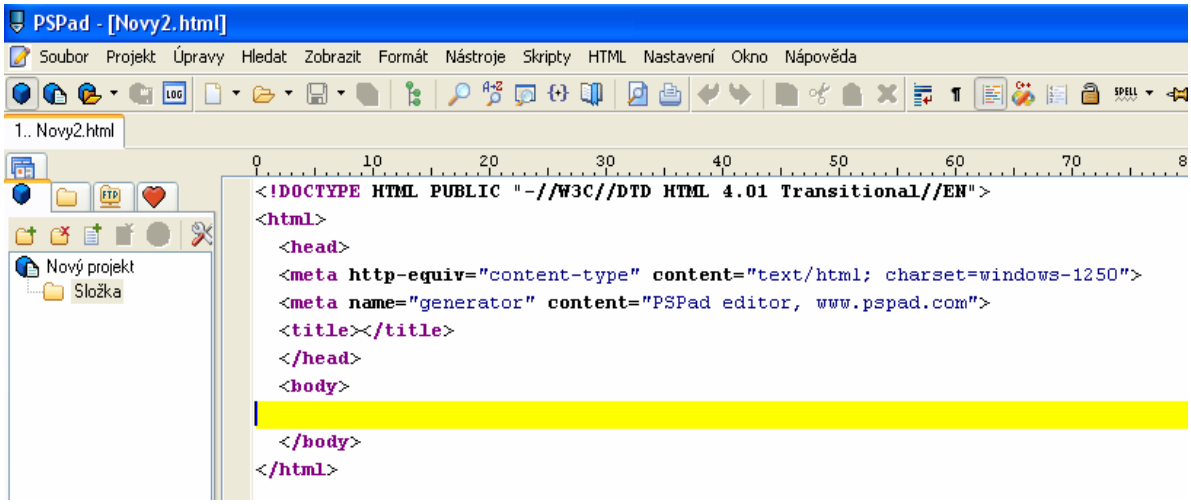
12.1 Vytvoření nového dokumentu

Po kliknutí na menu soubor a zvolení položky nový se objeví okno s výběrem formátu. Na ostatních záložkách si můžeme vybrat z velkého množství přednastavených šablon nebo vybrat jeden z dříve otevřených souborů.



Obr. 13 – Vytvoření nového souboru

Po výběru formátu se vygeneruje nový dokument s již předdefinovanou základní strukturou, do které můžeme okamžitě psát zdrojový text.



Obr. 14 – Prostředí programu PSPad s nově vytvořeným HTML souborem

12.2 Práce s dokumentem

Program PSPad zvýrazňuje syntaxi. Klíčová slova jsou graficky zvýrazněna, takže kód vždy působí velmi přehledně.

Užitečné funkce přístupné pomocí ikon:

hledání v textu – najde a zvýrazní zadané slovo v textu

nahrazení řetězce zadaným – nahradí zadaný řetězec jiným. Při každém výskytu hledaného řetězce se program dotazuje, zda chceme tento řetězec nahradit. Můžeme tedy nahradit i jen některé z nich.

hledání související závorky – pokud nastavíme kurzor před závorku, funkce nám najde jí odpovídající druhou část

vrácení poslední provedené operace – návrat je možný až o 1024 kroků

odvolání vrácení poslední změny

zapnutí nebo vypnutí zvýraznění syntaxe

změna zvýraznění syntaxe na zvolený formát – zobrazí se okno se seznamem formátů. Změny jsou viditelné ihned při označení vybraného formátu.

uzamknutí dokumentu pouze pro čtení

 upravení HTML kódu pro větší čitelnost – odsadí každý řádek od levého okraje tak, aby byl kód více přehledný.

 otevření dialogu pro výběr barvy – na dialogu si můžeme vybrat:

- bezpečné barvy
- barvy z 16-ti barevné palety
- stupně šedi
- pojmenované barvy
- barvy používané ve Windows

Po najetí kurzorem na barvu se zobrazí její hexadecimální vyjádření a slovní pojmenování, pokud existuje.

 zobrazení barvy – umožňuje zobrazení barvy po zadání jejího hexadecimálního tvaru nebo namíchání vlastní barvy

 převod HTML na TXT – vytvoří nový textový dokument, ve kterém bude pouze text bez tagů

 náhled HTML stránky v prohlížeči

Velmi efektivní je používání automatického dokončování. Po napsání prvního písmene nebo části výrazu a stisku CTRL + J nabídne program seznam možností na dokončení výrazu seřazených vzestupně dle abecedy.

Program PSPad nabízí také velké množství rozšíření. Například po stažení příslušného slovníku umožňuje kontrolu pravopisu.

13 NASTAVENÍ ŠIFROVACÍHO PROGRAMU TRUECRYPT

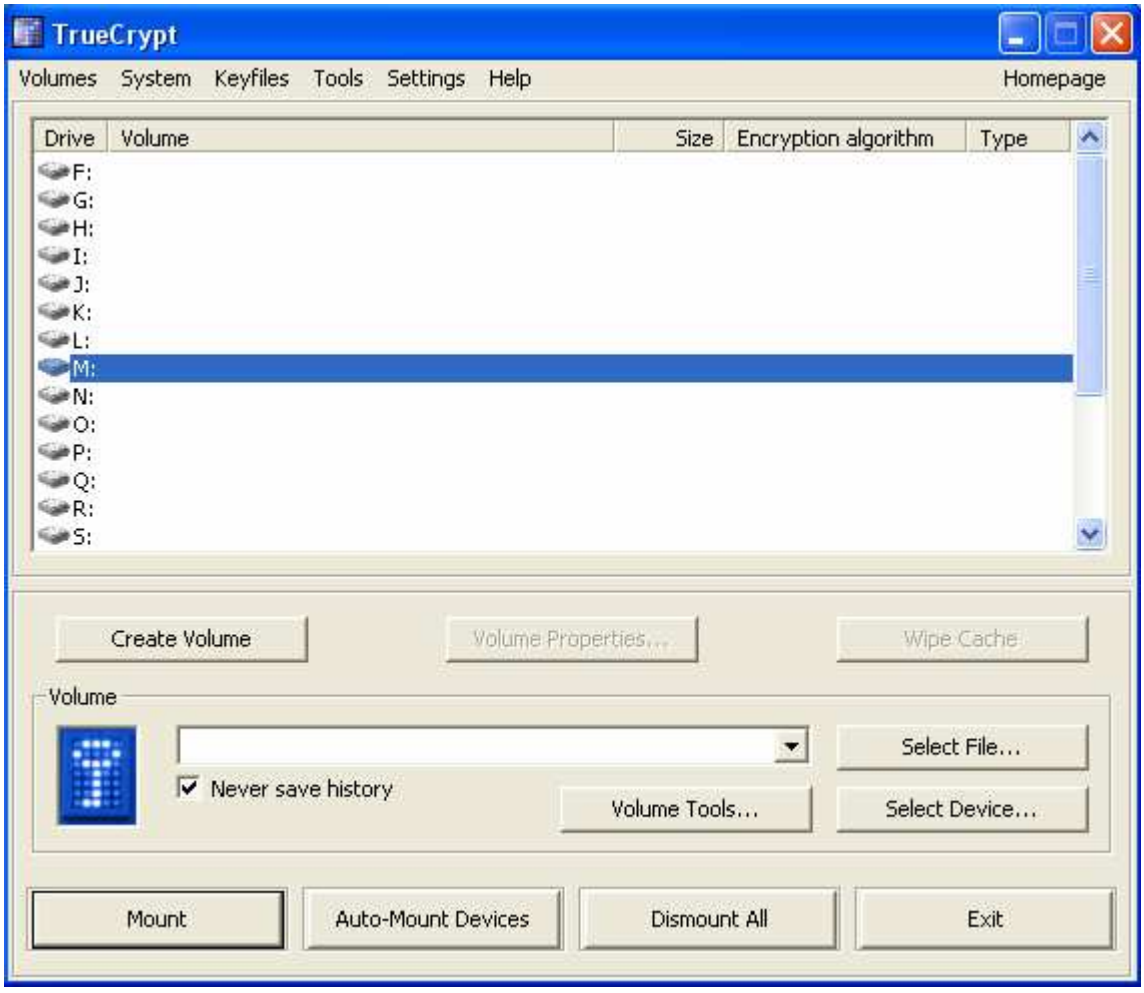
TrueCrypt je volně šiřitelný program pro zašifrování dat na počítači. Vytvoří virtuální logickou jednotku a vše co se nachází uvnitř této jednotky je zašifrováno. Program umožňuje šifrování algoritmy - AES, Serpent a Twofish [9].

Program TrueCrypt je ke stažení na této adrese:

http://www.truecrypt.org/downloads.php

13.1 Vytvoření virtuální jednotky

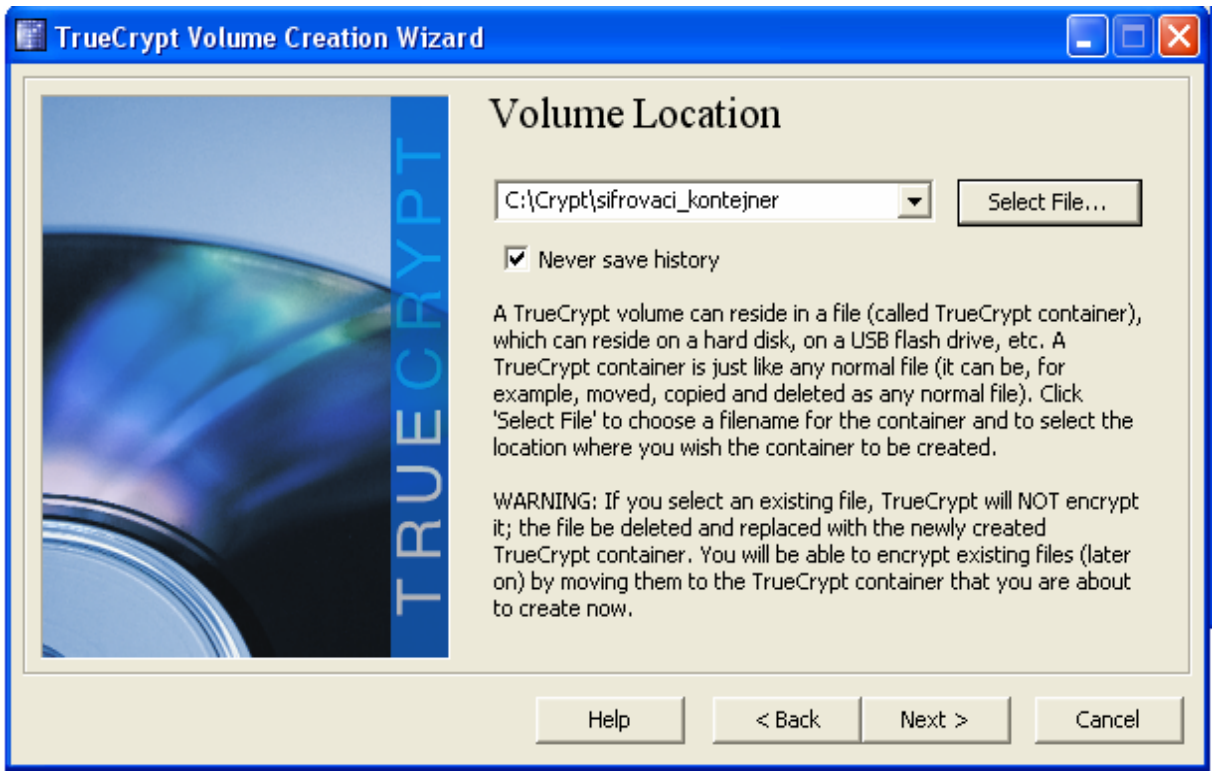
Po nainstalování a spuštění programu se objeví úvodní obrazovka:



Obr. 15 - Úvodní obrazovka programu TrueCrypt

Pro vytvoření virtuální jednotky klikneme na tlačítko „Create volume“ a následně potvrdíme volby „Create a file container“ a „Standard TrueCrypt volume“.

Objeví se další okno, ve kterém zvolíme „Select file“ a vybereme, kde chceme jednotku vytvořit a jak se bude jmenovat. Cestu k souboru uložíme pomocí tlačítka „Save“ a tlačítkem „Next“ se posuneme dále.



Obr. 16 - Umístění virtuální jednotky

V následujícím okně si vybereme algoritmus, kterým chceme data šifrovat. K dispozici jsou algoritmy AES, Serpent a Twofish nebo jejich vzájemné kombinace. Při kombinaci AES-Twofish-Serpent jsou 128-mi bitové bloky dat nejdříve zašifrovány pomocí algoritmu Serpent, poté algoritmem Twofish a nakonec AES. Každý algoritmus používá k šifrování svůj vlastní klíč.

V dolní části okna si pak můžeme zvolit hashovací funkci. Hashovací funkce jsou využívány náhodnými generátory čísel pro generování klíčů [9].

Obr. 17 - Výběr algoritmů

Tlačítkem „Next“ se posuneme na další okno, kde zadáváme velikost virtuální jednotky v kilobytech nebo v megabytech. Po zvolení vhodné velikosti opět klikneme na tlačítko „Next“.

Obr. 18 - Nastavení velikosti virtuální jednotky

Následuje okno, ve kterém zadáváme heslo. Toto heslo budeme potřebovat při každém připojení virtuální jednotky. Doporučuje se zadávat heslo delší než 20 znaků a využívat kombinace písmen, číslic a speciálních znaků.

Obr. 19 - Zadávání hesla

Poté generátor náhodných čísel vytvoří množinu náhodných čísel, ze které se vygenerují klíče.

Tato množina má délku 640B a její hodnoty jsou získávány na základě aktuálního pohybu myši, stisknutých kláves a dalších náhodných veličin. Čím delší a náhodnější bude pohyb kurzoru, tím bezpečnější budou vygenerované klíče [9].

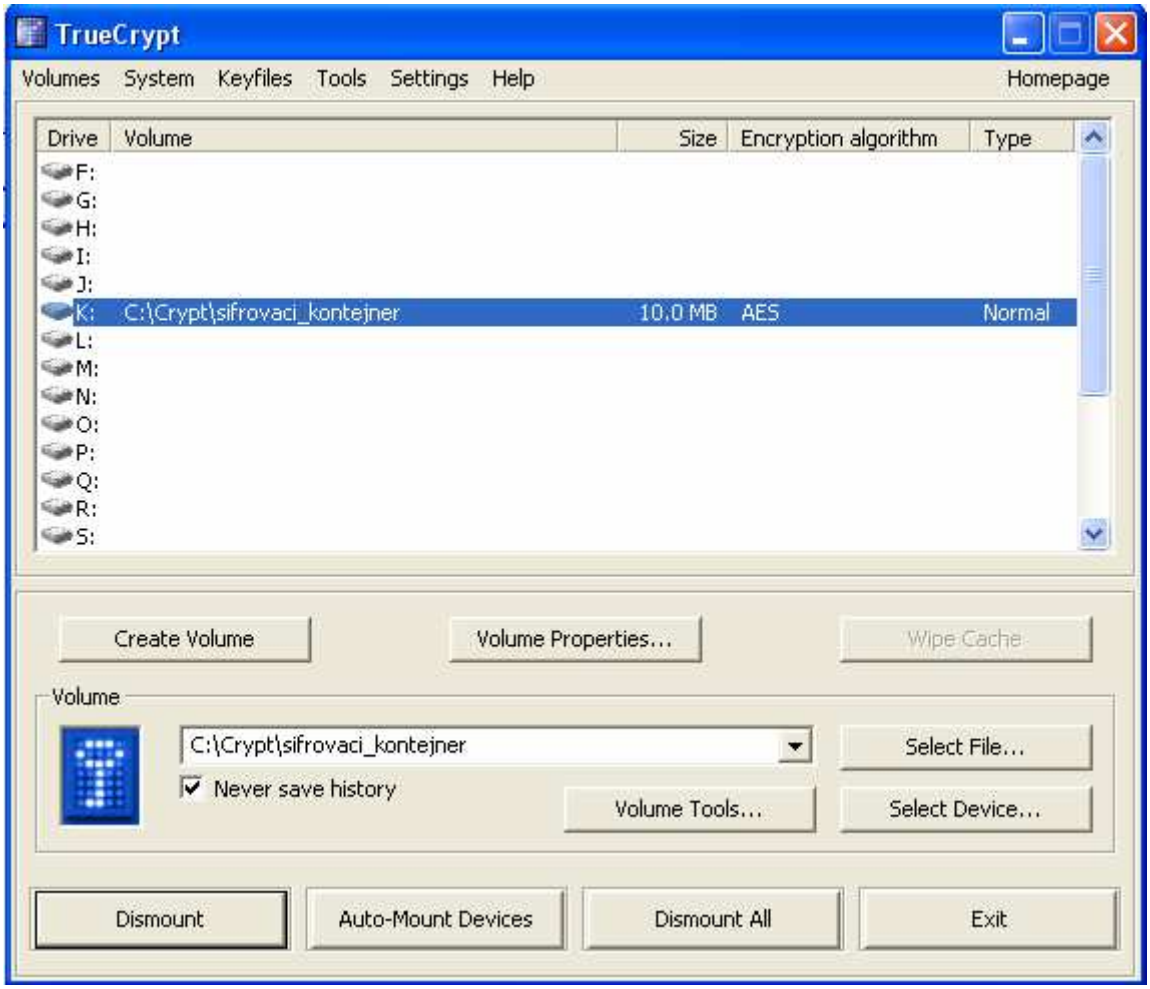
Obr. 20 - Generování klíčů

Po kliknutí na tlačítko „Format“ se zobrazí okno, které oznamuje úspěšné vytvoření virtuálního oddílu na námi zvoleném místě. Tlačítkem „Exit“ se vrátíme zpět do úvodní obrazovky.

13.2 Připojení virtuální jednotky

Dalším krokem bude samotné připojení virtuální jednotky. Ze seznamu svazků si vybereme ten, ke kterému chceme připojit virtuální jednotku. Klikneme na „Select File...“ a zvolíme námi vytvořenou jednotku. Tlačítkem „Mount“ připojíme jednotku.

Po zadání správného hesla se provede připojení virtuální jednotky. Pokud vše proběhlo správně, tak se u vybraného svazku zobrazí připojená jednotka.



Obr. 21 - Připojená virtuální jednotka

Celá tato virtuální jednotka je šifrována včetně jmen souborů a chová se jako opravdový disk. Soubory, které nahrajeme do této jednotky, jsou během nahrávání „za letu“ šifrovány.

Při kopírování souborů mimo jednotku jsou data zpět dešifrována. Pokud otevřeme např. obrázek umístěný uvnitř této jednotky, tak se jeho obsah dešifruje do paměti RAM a prohlížeč ho zobrazí jako normální obrázek. Dešifrované soubory nejsou nikdy ukládány na disk mimo jednotku, pouze do paměti RAM.

Pokud chceme ukončit práci a zajistit, aby se k souborům nikdo nedostal, stačí zvolit položku „Dismount“ nebo restartovat počítač. Virtuální položka zmizí a k souborům nebude možné se dostat. Pokud budeme chtít se soubory znovu pracovat, je třeba opět připojit virtuální jednotku a zadat správné heslo [9].

Při práci s programem TrueCrypt jsem nenarazil na žádné potíže a mohu ho vřele doporučit všem zájemcům, kteří hledají jednoduchý a přesto efektivní nástroj pro zašifrování dat na svém počítači.

ZÁVĚR

Cílem této bakalářské práce bylo vytvořit WWW průvodce, který srozumitelně vysvětlí důležité pojmy z oblasti moderní kryptografie a ukáže princip známých algoritmů. Bylo uvedeno rozdělení na symetrické a asymetrické šifry s výpisem jejich klíčových vlastností. Každý z obou druhů se svými vlastnostmi hodí k jinému účelu a proto se i v praxi používá kombinace obou druhů. Průvodce se také zabývá klíči a hesly, jež jsou pro kryptografii velmi důležité.

Algoritmů existuje velké množství a proto nebylo možné zde popsat všechny důležité algoritmy. Z tohoto důvodu jsem vybral jen 3 nejzajímavější. Algoritmy DES a RSA jsem vybral, protože oba jsou nejpoužívanější algoritmy ve své kategorii. El Gamalův systém pak z důvodu jeho jednoduchosti a využitelnosti pro digitální podpis. Charakteristika digitálního podpisu je rovněž uvedena včetně schémata principu.

Součástí teoretické části je také popis základů jazyka HTML a kaskádových stylů. V praktické části byl popsán postup při tvorbě WWW průvodce, který byl vytvořen v programu PSPad. Tento program se ukázal jako velmi vhodný a efektivní nástroj pro vytváření webových stránek. Na závěr práce je uveden popis nastavení šifrovacího programu TrueCrypt, jež je také součástí průvodce. Tento program se osvědčil a mohu jej vřele doporučit.

Průvodce by měl sloužit jako pomůcka pro každého, kdo se chce dozvědět o principech moderní kryptografie. Průvodce je optimalizován pro prohlížeče Internet Explorer a Mozilla Firefox.

CONCLUSION

The purpose of this bachelor thesis was to create a web guide, which will simply explain the basic conceptions of modern cryptography and will show the principle of known algorithms.

Basic division of ciphers into symmetric ciphers and the public key cryptography was included along with the description of the main features. Each of the cipher type fits different purposes and that is the reason why combination of both types is practically used. The guide also deals with the keys and passwords, which are highly important for cryptography.

There is big number of important algorithms, that is why it was not possible to describe all of them. Because of that I have chosen 3 most interesting algorithms. I have chosen DES and RSA, because they are both most used algorithms in their cathegory. El Gamal cryptosystem was chosen because of its simplicity and availablity for digital signature. The characteristics of digital signature with the scheme of principle is also included.

The basics of HTML language and cascade style sheets is also described in the theoretical part. The process of creation of the web guide is described in the practical part. The web guide was created in PSPad program. This program showed to be suitable and effective tool for web pages creation. In the end of the thesis is a setup description of the encryption program TrueCrypt, which is also part of the guide. This program was approved and I can warmly recommend it.

The web guide should serve as a help for everyone, who wants to learn about modern cryptography principles. Web guide is optimized for Internet Explorer and Mozilla Firefox browsers.

SEZNAM POUŽITÉ LITERATURY

[1] *An Introduction to Cryptography*, Network Associates, Inc., 1998, Santa Clara, California, USA.

[2] Chey Cobb (2004): *Cryptography for Dummies*. Wiley Publishing, Inc., Indianapolis, Indiana, USA.

[3] Fuller, R.G. & L.A. Ulrich (2004): *HTML in 10 Simple Steps or Less*. Wiley Publishing, Inc., Indianapolis, Indiana, USA.

[4] Man Young Ree (2003): *Internet Security: Cryptographic Principles, Algorithms and Protocols*. John Wiley and Sons Ltd, Chichester, West Sussex, England, United Kingdom

[5] Menezes, Alfred, J., Van Oorschot, Paul, C., Vanstone, Scott, A. *Handbook of Applied Cryptography*. 5th edition. [s.l.] : CRC Press, 1996. 816 s. Dostupný z WWW: <http://www.cacr.math.uwaterloo.ca/hac/>. ISBN 0-8493-8523-7.

[6] Piper, Fred, Murphy, Sean. *Kryptografie*. Pavel Mondschein. 1. vyd. [s.l.] : Dokořán, 2006. 158 s. Průvodce pro každého. ISBN 80-7363-074-5.

[7] Požár, Josef. *Informační bezpečnost*. [s.l.] : Aleš Čeněk - vydavatelství a nakladatelství, 2005. 311 s. ISBN 80-86898-35-5.

[8] Přibyl, Jiří, Kodl, Jindřich. *Ochrana dat v informatice*. 1. vyd. Praha (Česká republika) : Vydavatelství ČVUT, 1996. 299 s. ISBN 80-01-01664-1.

[9] TrueCrypt Foundation. *TrueCrypt - Free Open-Source On-The-Fly Disk Encryption Software for Windows Vista/XP, Mac OS X and Linux - Documentation* [online]. c2003-2008 , 31.3.2008 [cit. 2008-04-18]. Dostupný z WWW: <http://www.truecrypt.org/docs/>.

[10] Wembo Mao (2003): *Modern Cryptography: Theory and Practice*. Prentice Hall, Inc., Upper Saddle River, New Jersey, USA & Hawlett-Packard Company & Person Education Ltd.

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

MIPS Million Instructions Per Second

XOR Exclusive OR

AES Advanced Encryption Standard

RAM Random Access Memory

SEZNAM OBRÁZKŮ

SEZNAM TABULEK

SEZNAM PŘÍLOH

CD-ROM – WWW průvodce moderní kryptografií