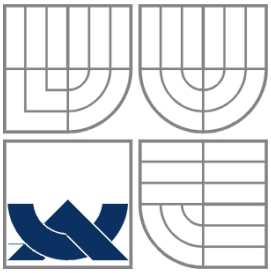
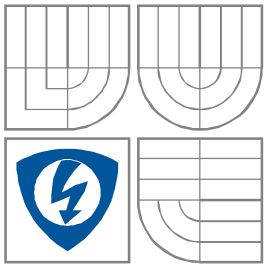


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF TELECOMMUNICATIONS

PORTÁL PRO PODPORU VÝUKY KRYPTOGRAFIE

PORTAL TO SUPPORT TEACHING CRYPTOGRAPHY

DIPLOMOVÁ PRÁCE
MASTER THESIS

AUTOR PRÁCE BC. TOMÁŠ FORMAN
AUTHOR

VEDOUCÍ PRÁCE DOC. ING. VÁCLAV ZEMAN, CSC.
SUPERVISIOR

BRNO 2010

Zadání DP

Abstrakt

Cílem diplomové práce je vybudování webového portálu pro prezentaci základních kryptografických algoritmů. Ty budou nejprve vysvětleny po teoretické stránce a následně demonstrovány pomocí skriptů.
Součástí projektu je vypracování zjednodušeného teoretického základu pro základní naplnění portálu daty. Dále pak vytvoření webového portálu pomocí jednoho z volně dostupných CMS systémů. Jako prostředek pro tvorbu demonstračních skriptů bude použit programovací jazyk Java a animační nástroj Flash.

Cílem vytvořeného webového portálu je vytvoření komunity odborné veřejnosti. Ta by mohla přispívat novými články, skripty a poznatky. Tímto přístupem byl portál udržován stále aktuální. Součástí portálu bude také sekce, která bude obsahovat slabiny nejpoužívanějších algoritmů spolu s návody, jak tyto slabiny eliminovat.

Klíčová slova

kryptografie, kryptoanalýza, šifrování, šifra, hashování, hash, asymetrická kryptografie, symetrická kryptografie, CMS, Joomla, Java, webdesign, SEO

Abstract

The main goal of this master's thesis is building of web portal for presentation basic cryptography algorithms. Those algorithms would be explained in the theoretical page in the first place. After that, they would be demonstrated by scripts.
One part of this project is designing simplified theoretical element for basic impletion portal of information. Next part is creating web portal by one of the free available CMS´s systems. Programming language JAVA would be used as an instrument for creating demonstration scripts. For creating animations will be used the Flash animation tool

Target of formed web portal is creating community of expert public. It would make new articles, scripts and knowledge. This way, the portal would be kept current. The section which would include failure the most widely used algorithms and instructions how to eliminate it will be part of portal.

Keywords

cryptography, cryptanalysis, encryption, cipher, hash, asymmetric cryptography, symmetric cryptography, CMS, Joomla, Java, web design, SEO

Bibliografická citace diplomové práce

FORMAN, T. *Portál pro podporu výuky kryptografie*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2010. 96 s., 5 s. příloh Vedoucí diplomové práce doc. Ing. Václav Zeman, Ph.D.

Prohlášení

Prohlašuji, že svou diplomovou práci na téma PORTÁL PRO PODPORU VÝUKY KRYPTOGRAFIE jsem vypracoval samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené diplomové práce dále prohlašuji, že v souvislosti s vytvořením tohoto projektu jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení § 152 trestního zákona č. 140/1961 Sb.

V Brně dne 24. května 2010 .....

podpis autora

Poděkování

Děkuji vedoucímu diplomové práce doc. Ing. Václavu Zemanovi, Ph.D.za odbornou pomoc a další cenné rady při zpracování mé diplomové práce.

V Brně dne 24. května 2010 .....

podpis autora

OBSAH

Seznam obrázků

Seznam tabulek

Úvod

Cílem této diplomové práce je zprostředkování, vysvětlení a demonstrace základních i pokročilých metod kryptografie pomocí internetu. Tyto informace budou podány a prezentovány tak, aby byly pochopitelné a mohly sloužit k výukovým účelům. Výsledkem bude internetový portál, který zpřístupní výukové informace všem, kdo budou mít o studium problematiky kryptografie zájem.

Na internetovém portálu bude rozebrána kryptografie od svých prvopočátků až po současnost tak, jak se postupně vyvíjela. Data budou na portále uspořádána do logických kategorií, jejichž obsah spolu souvisí. Pro podporu výuky a pochopení základních principů a algoritmů budou na portále k dispozici také názorné animované ukázky a jednoduché programy pro vyzkoušení a aplikaci. Tímto obsahem by portál měl být schopen nejen pomoct pochopit kryptografii a její principy, ale také přilákat odbornou veřejnost, čímž by se dostal do povědomí a mohl by se rozrůstat o kvalitní články a příspěvky.

Protože v dnešní době je internetových portálů zabývajících se kryptografií mnoho, bude nejdříve nutné udělat rozbor stávající situace na internetu a vyhledání prázdného místa pro zbudování nového portálu.

V první části práce bude rozebrána teorie tak, jak by měla být později prezentována na portále. Členění by mělo odpovídat postupnému rozvíjení znalostí o kryptografii.

Druha hlavní část celé práce se bude věnovat internetovému portálu jako takovému. Budou nastíněny trendy v dnešním webdesignu a informace o technologiích použitých při tvorbě webu i jeho interaktivního obsahu.

1. Analýza stavu na internetu

Vzhledem k tomu, že na internetu se v současnosti vyskytuje mnoho webových portálů, které jsou zaměřeny podobným směrem jako plánovaný portál, je třeba před celkovým návrhem portálu provést analýzu. Ta by neměla být zaměřená pouze na současný stav a webové portály se zaměřením čistě na kryptografii, ale také na všeobecné encyklopedie, které jsou v poslední době velmi oblíbené.

Na základě analýzy bude stanovena oblast, která není na internetu dobře pokryta, nebo je pokrytá nedostatečným způsobem.

1.1 Současný stav

Na internetu se dá najít mnoho internetových stránek zaměřených na odborná témata. Některé z těchto webů jsou zaměřeny čistě na jeden daný obor nebo jedno dané aktuální téma, některé se zabývají například celou oblastí. Výjimku tvoří všeobecné encyklopedie, které obsahují obrovskou databázi znalostí. I proto jsou tyto encyklopedie mezi veřejností stále oblíbenější i přes riziko méně kvalitních a často neaktuálních informací.

Na webové portály se dá nahlížet také z jiné strany a to z hlediska kvalitních informací v českém jazyce. Odborná veřejnost by sice měla disponovat alespoň základní jazykovou znalostí angličtiny, jakožto nejpoužívanějšího technického jazyka, ale i přesto je pro lepší a pohodlnější pochopení dané problematiky, mít materiály v rodném jazyce.

1.1.1 Odborné portály

Z hlediska dohledatelnosti kvalitních materiálů o kryptografii je na tom český internet velice špatně. Pouze minimum webových portálů se zabývá pouze kryptografií a jí příbuzným oborům komplexně. Většina portálů je zaměřena obecněji a na svých stránkách publikují pouze nejnovější trendy – neboli to, co přitahuje čtenáře. Pokud český uživatel hledá komplexní informace v daném oboru, vesměs se mu to nepodaří, nebo bude muset zkombinovat informace z mnoha portálů dohromady tak, aby dostal o kryptografii ucelenější obrázek. V tomto české portály velmi zaostávají za těmi zahraničními, které jsou vesměs všechny v anglickém jazyce.

Pokud zadáme do vyhledávače slovo kryptologie, je poskytnutý výpis plný odkazů do encyklopedií a odkazů na jednotlivé články, které jsou publikovány na webech zabývajících se technikou. Žádný z odkazů na předních pozicích bohužel nevede na portál, kde by uživatel dostal komplexní informace od historie až po současnost, aniž by musel navštívit mnoho portálů a skládat kousky informací jako skládanku. Nehledě na to, že informace z různých zdrojů jsou psány různými styly a jejich terminologie může obsahovat různá označení – to komplikuje ucelenost a možnost pochopit danou problematiku.

Příkladem odborných portálů mohou být například portál securityworld.cz, který se zabývá především publikací článků o bezpečnosti dat a možnostem, jak svá data zabezpečit. Nejde tedy přímo o odborný portál, který by se zabýval kryptografií obecně, ale má charakter zejména upozorňovací a tedy zaměřený na širokou veřejnost. Některé z článků jsou však velice zajímavé a proto se nabízí použití tohoto webu jako zdroj RSS zpráv pro budoucí portál.

Obr. 1.1 Ukázky webů crypto-world.info a securityworld.cz

Druhým příkladem odborného webu je crypto-world.info. Ten je již zaměřený odborněji a po zorientování se na portále je možno dohledat i zajímavé informace. Tento web je tvořen jako prezentace některých předních českých vědců z oblasti kryptologie. Nevýhodou webu je jeho nepřehlednost a i přes to, že obsahuje zajímavé informace, mnoho uživatelů bude odrazeno neschopností se ke hledaným informacím dostat.

Analýza českého internetu tedy nedopadla nejlépe a je zde mnoho co zlepšovat. Věřím, že mnoho odborníků, nebo vysokých škol má svoje portály s velice dobrými informacemi a zpracováním. Jejich nevýhodou však je nechtěná a někdy bohužel i chtěná izolace od všeobecné veřejnosti. Právě zde je místo pro nový portál, který by takovéto služby poskytoval všem.

1.1.2 Encyklopedie

V dnešní době jde o jednu z nejoblíbenějších možností, jak získat základní informace o tom, co uživatele právě zajímá. Do velké míry za to může jednoduchost a názornost vysvětlovaných informací. Uživatel, který chce získat hlubší informace, však většinou neuspěje. Důvodem je právě to, že encyklopedie jsou zaměřeny na tak širokou oblast věcí, že je v podstatě nemožné, aby obsahovaly větší detaily. A právě zde by měly nastupovat odborné portály a znalost informací dále prohlubovat.

Můžeme si položit i otázku, proč uživatel raději následuje odkaz do encyklopedie, než na odborný portál i v oblastech, kde je portálů dostatek. Důvodem je propracovaná optimalizace encyklopedií, která umožňuje jejich odkazy dostávat na první příčky ve vyhledávání. Pro uživatele je pak jednodušší následovat první odkaz, než listovat stránkami odkazů.

Obr. 1.2 Ukázka encyklopedie Wikipedia a portálu Cryptography

Nejznámější a nejpoužívanější encyklopedií je pravděpodobně Wikipedia. Její výhodou je, že je tvořena veřejností. Každý se tak může podílet na její tvorbě a prohlubování informací. Vzhledem k tomu, že Wikipedia je dnes snad nejpoužívanější online encyklopedií, obsahuje informace o veškerém dění a ze všech oborů. Bohužel pro technické obory je však tato část encyklopedie stále nedostatečně zpracovaná, alespoň v české lokalizaci. Anglická a tedy domovská verze obsahuje tzv. portály, které se zabývají širšími celky zájmu. Najdeme zde tedy portál o matematice, o fyzice a dokonce také o kryptografii. Z české verze Wikipedie je však tento portál nepřístupný. Pokud však chceme získat základní informace o problematice kryptografie, je Wikipedia ideálním odrazovým můstkem.

1.1.3 Softwarové portály

Samostatnou kapitolou jsou webové portály propagující kryptografické programy jak demonstrační, přes výukové, po plně funkční a pro šifrování použitelné. I přesto, že se tyto portály nezabývají problematikou kryptografie jako takovou, poskytují uživatelům možnost stáhnutí jejich programu a vyzkoušení kryptografie v praxi. Praktická zkušenost je to nejlepší, čeho může uživatel dosáhnout. Může si tak vyzkoušet jednotlivé algoritmy, jejich vstupy a výstupy na vlastních datech.

Obr. 1.3 Ukázka webu propagujícího program CrypTool

K nejznámějším softwarovým produktům, které jsou volně dostupné je program CrypTool. Uživatel si v něm může po stažení a instalaci vyzkoušet základy kryptoanalýzy i kryptografie. Tato aplikace umožňuje uživateli vše poznat jednoduchým a nenáročným způsobem.

Druhým velice dobře známým programem (standardem) je OpenPGP. Ten umožňuje šifrování a podepisování pomocí algoritmu RSA. Nejčastěji je tento standard používán pro šifrování a podepisování e-mailových zpráv. Nejde tedy o výukový program, ale o aplikaci, která se přímo využívá v praxi. Její ovládání a pochopitelnost je však na takové úrovni, že práci s ní zvládne i začátečník, zejména pokud je použita jedna z grafických nástaveb, jako je například Kleopatra.

1.2 Nový portál

Podle výsledků analýzy by nově plánovaný portál měl patřit do první probírané skupiny, tedy mezi odborné portály, které se zabývají kryptografií více do detailu. Ideální však je vybudování portálu, který bude částečně zahrnovat i výhody dalších dvou uvedených druhů portálů. Výsledkem této diplomové práce by měl být základ portálu, který bude obsahovat základní databázi znalostí, pojmů a výukových programů. Tento portál, postavený na jednom z GNU/GPL redakčních systémů, tak bude připraven poskytnout nejen základní, ale i pokročilé informace a funkce.

Samozřejmostí bude možnost přispívání články od registrovaných členů portálové komunity. Programátoři budou moci tvořit applety a animace, které budou moci na webu prezentovat a zjednodušit tak pochopení kryptografie od základů až po složité systémy.

2. Obsah portálu

V následující rozsáhlé části diplomové práce bude rozebrána teorie, která by měla tvořit základní bázi znalostí pro vytvářený portál. Její členění by logicky mělo odpovídat postupnému rozvoji znalostí tak, jak by jimi neznalý uživatel měl procházet.

Na začátku kapitoly budou rozebrány pojmy jako kryptologie, kryptografie a kryptoanalýza. Následovat pak bude historie kryptografie, počínaje steganografií, jakožto předchůdcem dnešní kryptografie. Poté bude rozbor historie pokračovat kryptografií jako takovou a to od dob římských a Caesarovy šifry až po mechanické šifrovací stroje z druhé světové války.
V další podkapitole budou položeny základní matematické znalosti potřebné k tomu, aby byly pochopitelné následující složitější algoritmy. Následovat pak bude teoretický rozbor digitálních podpisů a hashovacích funkcí, zakončený pohledem do kvantové kryptografie a nástinem jejího fungování.
V závěru pak budou nastíněny možné typy útoků na algoritmy a jejich slabiny.

2.1 Úvod do kryptologie

Kryptologie je věda, zabývající se šifrováním a dešifrováním zpráv. Jinými slovy tedy zabezpečením komunikace mezi dvěma a více komunikujícími stranami. Jejími hlavními obory jsou kryptografie a kryptoanalýza.

První jmenovaný obor se zabývá nejen šifrovacími nástroji a algoritmy, ale také hardwarovou konstrukcí šifrovacích strojů.

Kryptoanalýza se v poslední době dostává čím dál více do popředí, protože se zabývá tím, jak šifrované zprávy luštit. Hraje tedy důležitého oponenta kryptografii jako takové.

Právě kryptoanalýza je důležitá pro odhalování chyb v algoritmech. Díky ní jsou odhalovány chyby, které se v algoritmech vyskytují. Jejich nedostatky pak mohou být zavčas vyřešeny ještě před tím, než způsobí bezpečnostní problém [16].

2.2 Historie kryptografie

Jelikož je kryptografie jako obor velice stará, stojí za zmínku probrat alespoň základní historický vývoj kryptografických metod a jejich předchůdců.

2.2.1 Steganografie

Steganografie je jednoduše řečeno ukrývání důležitých informací do běžně známých věcí. Její původ je znám již z antického Řecka. Steganografie jako taková nemá s kryptografií nic společného. Přesto slouží ke stejnému účelu – ochraně důležitého obsahu před zneužitím, ať už je předmětem obsahu cokoliv.

Princip steganografie je velice jednoduchý. Pokud jej vztáhneme k dnešní době - vezmeme naprosto obyčejnou věc, jako je například digitální fotografie a ukryjeme do ní pro nás důležitá data tak, že pokud o těchto ukrytých datech nebude nikdo vědět, není šance jejich prozrazení [2, 5].

Praktické využití

Nevýhodou použití samotné steganografie spočívá v tom, že pokud se potencionální útočník o ukrytých datech dozví, nic mu nebrání je z nosiče extrahovat. Data nejsou nikterak chráněna. Proto se v dnešní době kombinuje steganografie právě s kryptografií, kdy se data nejdříve zašifrují a poté vloží do nosiče.

Pokud bychom použili pouze kryptografii, jsou tato zašifrovaná data charakteristicky nápadná při internetovém provozu a jsou tedy z pohledu útočníka „lákavá“. Pokud si chceme být jisti, že zprávu skryjeme a zároveň ji ochráníme před přečtením dostatečně silným algoritmem, použijeme právě kombinaci steganografie-kryptografie [2, 5].

Princip funkce

Vezmeme-li výše uvedený příklad fotografie a skrývaného souboru, pro správnou funkci a zajištění dobrého utajení je třeba dodržovat několik základních pravidel.

Prvním a zásadním je to, že by velikost vkládaného souboru neměla přesahovat 1:4 velikosti fotografie (myšleno v bytech). Proto jsou pro ukrývání zpráv vhodné jako nosiče fotografie uložené ve formátu bitmapy. Zde je každý pixel obrazu představován jedním bytem paměti (v případě 8-bitové barevné hloubky). Pokud v každém pixelu, jeho bytu dat, pozměníme jeden bit, je tato změna pro lidské oko téměř nerozeznatelná. Získáme tak na každý pixel obrazu jeden bit pro naši ukrývanou zprávu.
Jednoduše řečeno, pro fotografii o velikosti 1024*768 to je cca. 100kB dat, což postačí například pro 100.000 znaků textu.
Samozřejmě platí přímá úměra. Čím větší je ukrývaný soubor, tím více jsou jeho stopy vidět v obraze nosiče a naopak, pokud je ukrývaný soubor dostatečně malý, nejsou stopy v obraze nosiče viditelné a ten je k nerozeznání od originálu.

Je samozřejmé, že jako nosiče nemusíme využívat pouze fotek a obrázků, ale v podstatě jakéhokoliv souboru. Nejlepší jsou takové formáty souborů, které obsahují pouze konkrétní data, a nikdo by od nich nečekal nic jiného. Například hudební a video soubory [5].

Teoreticky lze také ukrýt například fotografii do fotografie. Neměl by však být porušen poměr 1:4, aby nebyly viditelné obě fotografie zároveň.

2.2.2 Začátky kryptografie

Substituční šifra

V podstatě jde o jednoduchý princip záměny jednoho znaku za jiný podle předem určeného pravidla. Pravidlem může být například jednoduché nahrazení znaků abecedy jinými znaky tak, aby zašifrovaná zpráva nedávala smysl.

Podtypem substituční šifry je tzv. *Caesarova šifra*. Ta využívá posun písmen abecedy o určitý počet tak, že zašifrovaná zpráva nedává smysl.

Vylepšením Caesarovy šifry o tzv. Tabulku záměn dosáhneme jakéhosi primitivního „zašifrování“ naší zprávy podle hesla. V následující tabulce jsou k vidění dva způsoby použití Tabulky záměn [4].

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
H E S L O A B C D F G I J K M N P Q R T U V W X Y Z

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
H E S L O P Q R T U W V X Y Z A B C D F G I J K M N

Aditivní šifra

Prvním případem použití aditivní *Vigenérovy šifry* byla ve speciálním případě již Caesarova šifra. Vigenérova šifra spočívá ve volbě hesla (slova), se kterým se postupně sčítají znaky zprávy. Pokud je zpráva delší, heslo se opakuje do té doby, dokud nebude sečteno se všemi znaky zprávy.

otevřený text	S T A S T N E A V E S E L E
klíč	**H E S L O** H E S L O H E S L
šifrový text	A Y T E I V J T H T A J E Q

Tato metoda je vhodnější než jednoduchá substituční šifra, protože z kryptogramu nedokážeme odhadnout četnost znaků zprávy.

Prozatím jedinou neprolomitelnou šifrou je tzv. *Vernamova šifra*. Ta také spočívá ve sčítání znaků zprávy s heslem. To je však složeno z jednorázově vygenerovaného náhodného pole znaků a je stejně dlouhé jako zpráva. Není tedy způsob, jak zjistit vztahy mezi znaky a šifru tedy rozluštit. Pro náročnost se toto šifrování používá pouze pro extrémně důležité zprávy [4].

Transpoziční šifra

Principem *Transpoziční šifry* je opět záměna znaků podle určitého algoritmu. Její výhodou je jednoduchost, se kterou se šifruje i dešifruje. Z toho však vyplývá i její hlavní nevýhoda – jednoduchá rozluštitelnost bez znalosti principu. Jednou ze základních transpozičních metod je jednoduchá sloupcová transpozice, při matici, do

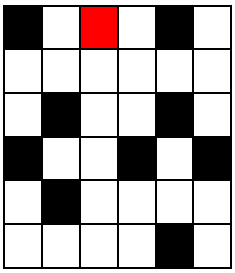
které je zapsaný text, transponujeme. Pokud však zachováme počet sloupců a řádků jako v následujícím příkladu, můžeme vidět, že rozluštění této šifry není nikterak složité.

Ukázka: ahoj ja jsem sifrovany text

AHOJJAJSEMSIFROVANYTEXT

A H O J J	A A S V E
A J S E M	H J I A X
S I F R O	O S F N T
V A N Y T	J E R Y X
E X T X X	J M O T X

Další možností v transpozičním šifrování je použití tzv. Transpoziční mřížky. Jde opět v podstatě o matici políček, z nichž jsou některé vystřižené. Do vystřižených políček vpisujeme text. Po vyplnění všech políček otočíme matici o 90 stupňů a pokračujeme. Po dokončení otočení o 270 stupňů by měly být všechna políčka zaplněna a náš text vzájemně zašifrován.
Pro použití tohoto stylu je třeba, aby byla matice čtvercová (vzhledem k otáčení) použít lze také jakýkoliv jiný středově souměrný předmět (mnohoúhelníky a podobně) [4].



Obr. 2.1 Příklad transpoziční mřížky [4]

Kombinovaná šifra
Vzhledem k nedokonalosti výše popsaných šifrovacích metod je vhodné použít jejich kombinaci. Tím dokážeme docílit toho, že se nám míra bezpečnosti znásobí. Pro útočníka tak není zcela jednoduché zprávu jednoduchým způsobem rozluštit.

Šifrovací stroje
Historickou kapitolu šifrování na nižší úrovni a také milník uzavřelo strojové kódování, které se hojně využívalo pro kódování vojenských zpráv za 2. světové války. Příkladem takovéhoto šifrovacího stroje je Enigma, která byla používána Němci [2, 4].

2.3 Matematické základy

V současné kryptografii přestala dostačovat základní matematika, jako je sčítání a násobení. Bylo jasné, že pro další rozvoj kryptografie bude nutné použít vyšší matematiku. V průběhu let se ukázalo, že pro kryptografii mají speciální význam

poznatky, které se týkají diskrétní matematiky – neboli oboru, který se zabývá celými čísly a diskrétními (nespojitými hodnotami).

Základy pro současnou kryptografii jsou rozděleny mezi několik oborů diskrétní matematiky. Patří mezi ně:

- Modulární aritmetika
- Grupy, okruhy, tělesa a pole
- Konečná pole
- Teorie čísel

2.3.1 Modulární aritmetika

Tento obor diskrétní matematiky se zabývá celočíselným dělením nad oborem celých čísel Z. Na tuto operaci se dá nahlížet z několika možných úhlů pohledu. V následující části textu budou rozebrány ty nejdůležitější operace a poznatky, které jsou potřeba v dnešní kryptografii [17].

Dělitelnost čísel

Definice 2.1. Nechť čísla $a, b \in$ Z. Číslo b je dělitelné číslem a právě tehdy, když existuje také číslo $q \in$ Z, které splňuje podmínku:

$$b = a \cdot q \tag{2.1}$$

Pokud takové číslo existuje, pak platí, že číslo a je dělitelem čísla b a číslo b je dělitelné číslem a. Tento stav se označuje jako $a \mid b$. Pokud takové číslo q neexistuje, můžeme říci, že číslo b není dělitelné číslem a beze zbytku. Tento stav se označuje $a \nmid b$.

Na základě tohoto poznatku můžeme definovat také číslo r, které je zbytkem po dělení čísel a a b. Platí tedy:

$$a = b \cdot q + r \tag{2.2}$$

V tomto případě existuje pouze jedna dvojice čísel q a r. [17]

Největší společný dělitel

Definice 2.2. Nechť čísla $a, b \in$ Z. Největším společným dělitelem čísel a, b je nejvyšší celé číslo d, pro které platí $d \mid a$ a zároveň $d \mid b$.

Největší společný dělitel je nejčastěji označován zkratkou **gcd(*a, b*)**.
Zároveň také můžeme stanovit pojem **nesoudělnost**. Ten můžeme použít tehdy, pokud platí následující rovnost [17]:

$$\gcd(a, b) = \gcd(b, a) \tag{2.3}$$

Euklidův algoritmus

Pokud máme malá čísla *a*, *b* tak je hledání největšího společného dělitele nenáročné. Pokud však jsou čísla *a*, *b* velká, je třeba pro výpočet gcd(*a*, *b*) použít Euklidův algoritmus.

Princip Euklidova algoritmu spočívá v několikanásobném použití operace dělení se zbytkem. Algoritmus lze zapsat jako posloupnost následujících kroků:

$$\begin{gathered} a = b \cdot q_1 + r_2 \\ b = r_2 \cdot q_2 + r_3 \\ r_2 = r_3 \cdot q_3 + r_4 \\ \vdots \\ r_{n-2} = r_{n-1} \cdot q_{n-1} + r_n \\ r_{n-1} = r_n \cdot q_n + 0 \end{gathered} \tag{2.4}$$

$$\gcd(a, b) = r_n$$

Z tohoto zápisu je jasné, že největším společným dělitelem je číslo r_n získané v předposledním kroku celého algoritmu [18, 17].

Vlastnosti modulární aritmetiky

V modulární aritmetice se ve většině případů nevyužívá zápis (2.2). Ekvivalentem právě pro tento zápis je velice často používaný a známý tvar:

$$a = b \cdot q + r \equiv a \bmod b = r \tag{2.5}$$

Číslo *b* se v tomto případě nazývá modulárním operátorem.

Pokud dvě čísla $a, b \in \mathrm{N}$ dělená číslem $n \in \mathrm{N}$ mají stejný zbytek *r*, nazýváme tyto čísla **kongruentními** podle modulu *n* [17]:

$$\begin{gathered} a \bmod n = b \bmod n \\ a \equiv b \bmod n \end{gathered} \tag{2.6}$$

V modulární aritmetice dále platí následující pravidla:

$$[(a \bmod n) + (b \bmod n)] \bmod n = (a + b) \bmod n \tag{2.7}$$
$$[(a \bmod n) - (b \bmod n)] \bmod n = (a - b) \bmod n \tag{2.8}$$
$$[(a \bmod n) \cdot (b \bmod n)] \bmod n = (a \cdot b) \bmod n \tag{2.9}$$

[17]

Ukázky operací modulo

Tab. 2.1 Ukázka sčítání modulo 4

+	0	1	2	3	4
0	0	1	2	3	0
1	1	2	3	0	1
2	2	3	0	1	2
3	3	0	1	2	3
4	0	1	2	3	0

Tab. 2.2 Ukázka násobení modulo 4

.	0	1	2	3	4
0	0	0	0	0	0
1	0	1	2	3	0
2	0	2	4	2	0
3	0	3	2	1	0
4	0	0	0	0	0

2.3.2 Grupy, okruhy a tělesa

Grupa

Grupa je množina čísel, která splňuje určité axiomy, tedy podmínky, které jsou pevně dány a nedokazují se. Jako takové mají velký význam právě v kryptografii.

Grupou je nazývána množina G, společně s matematickou binární operací, která je na ní prováděna. Grupy můžeme rozdělit na aditivní a multiplikativní a to dle operace, kterou na ní provádíme. Kromě axiomu uzavřenosti musí každá grupa splňovat následující tři axiomy [19]:

Aditivní notace:

asociativita $a + (b + c) = (a + b) + c$ (2.10)

neutrální prvek $(\exists 0)(\forall a)\ a + 0 = 0 + a = a$ (2.11)

inverzní prvek $(\forall a)(\exists b)\ a + b = b + a = 0$ (2.12)

Multiplikativní notace:

asociativita $f \cdot (g \cdot h) = (f \cdot g) \cdot h$ (2.13)

neutrální prvek $(\exists e)(\forall g)\ g \cdot e = e \cdot g = g$ (2.14)

inverzní prvek $(\forall g)(\exists h)\ g \cdot h = h \cdot g = e$ (2.15)

Výše uvedená pravidla jsou velmi jednoduchá. Abychom však grupy řádně pochopili, je důležité si platnost těchto pravidel uvědomit v aplikaci na množinu prvků.

Okruh

Pokud na množině $B \neq 0$ definujeme binární operace + a ·, tedy množina zapsaná jako (A; +,·), můžeme ji nazvat okruhem, pokud [17]:

(A; +) je komutativní grupa – pro její prvky platí $a * b = b * a$
(A; ·) je pologrupa – tedy splňuje axiomy uzavřenosti a asociativity

Těleso

Okruh (A; +,·) můžeme nazvat tělesem, pokud platí, že (A-{0}; ·) je grupou [17].

2.3.3 Konečná pole

Konečné pole se označuje jako GF (p). Jde o systém celých čísel, nad kterým lze provádět matematické operace sčítání a násobení, obě modulo p. Celý systém lze tedy zapsat jako (Z_p; +, ·), pokud Z_p = {0, 1, 2, 3, …, p-1}.

Konečná pole mají velký význam právě v kryptografii, nejen však v jeho základní verzi zapsané výše, ale také jako speciální poddruh zapisovaný jako GF (p^n), kde prvočíslo p=2, číslo $n \in$ N.

Příkladem nejjednoduššího konečného pole je GF (2^1), tedy pole s prvky 0, 1 a modulem 2:

+	0	1
0	0	1
1	1	0

·	0	1
0	0	0
1	0	1

Pokud je $n \geq 2$, nazývá se konečné pole polynomiálním. Příkladem je GF (2^2), tedy pole s prvky 0, 1 a modulem 2:

+	0	1	2	3
0	0	1	2	3
1	1	0	3	2
2	2	3	0	1
3	3	2	1	0

·	0	1	2	3
0	0	0	0	0
1	0	1	2	3
2	0	2	3	1
3	0	3	1	2

Při sčítání se postupuje standardním způsobem, při násobení je třeba použít tzv. redukční polynom stupně n s koeficienty {0, 1, …, p-1}, který je nerozložitelný [17].

2.3.4 Teorie čísel

Tvoří základní stavební kámen v asymetrické kryptografii, tedy v kryptografii, kde se používají dva rozdílné klíče. Teorie čísel hovoří zejména o elementárních zákonech matematiky. Jak bude dále vysvětleno, na těchto základních principech stojí bezpečnost asymetrických algoritmů jako takových.

Přirozená čísla
Množina přirozených čísel se označuje písmenem N.

Definice 2.3. Přirozené číslo $n > 1$ můžeme nazvat prvočíslem, pokud má pouze triviální dělitele. Pokud má číslo n také netriviální dělitele, potom mluvíme o čísle složeném.

Množinu N tedy můžeme rozdělit na celkem tři části. První část tvoří číslo 1, to je dělitelné pouze samo sebou. Druhou skupinu tvoří vlastní prvočísla. Třetí skupinu tvoří čísla složená [17].

Prvočísla
Otázkou zůstává celkový počet prvočísel. Euklidův důkaz říká, že množina prvočísel N je nekonečně veliká. Pro stanovení hustoty prvočísel neboli jejich počet v různě velkých množinách celých čísel, lze použít funkci 2.16, přičemž funkce $\pi(n)$ stanovuje absolutní počet prvočísel menších než číslo n. [17]

$$\pi(n) \approx \frac{n}{\log(n)} \tag{2.16}$$

Porovnání přesnosti funkce 2.16 si můžeme vidět v tabulce 2.3.

Tab. 2.3 Porovnání přesnosti funkce 2.16 s $\pi(n)$ [17]

n	$\pi(n)$	$n / \log(n)$
10^3	168	144,8
10^4	1229	1085,7
10^5	9592	8685,9
10^6	78498	72382,4

Z tabulky 2.3 můžeme poznat, že s rostoucím číslem n je odhad počtu prvočísel menších než n přesnější.

Jak bylo řečeno výše, základem výpočtu klíčů v asymetrické kryptografii jsou velká prvočísla. K tomu, abychom byli schopni najít prvočísla vysokého řádu, musíme být schopni je detekovat s vysokou pravděpodobností. Ve skutečnosti detekce prvočíselnosti pracuje na principu náhodného generování vysokých čísel, která jsou následně testována na prvočíselnost za pomoci některého z algoritmů. Pro detekci prvočísla můžeme využít například Fermatovu větu. Pokud chceme zvětšit pravděpodobnost, že námi generované číslo je opravdu prvočíslo, pak je doporučeno provést test na jedno číslo několikanásobně [17].

Základní věta aritmetiky
Jedná se o Euklidův princip, který tvrdí, že základy množiny celých čísel Z-{0, -1, 1} tvoří prvočísla. Kombinací prvočísel tak můžeme vytvořit jakékoliv celé číslo.

Z tohoto také vyplývá, že každé celé číslo (kromě uvedené výjimky) lze rozložit na součin dvou a více prvočísel. Této matematické operaci se říká kanonický rozsah [17].

Fermatova věta

Využívá se pro zjištění nejmenšího kladného zbytku mocniny v operaci modulo. Doslova je Fermatova věta vyjádřena vztahy 2.17, pokud a∈ Z a p ∤ a [17].

$$a^{p-1} \equiv 1 \bmod p$$
$$a^{p} \equiv a \bmod p \qquad (2.17)$$

Eulerova funkce

Jde o funkci, která stanovuje počet všech přirozených čísel *k* za podmínky $1 \leq k \leq n$, která jsou nesoudělná s číslem *n*. Eulerova funkce se zapisuje jako:

$$\varphi(n) = k \qquad (2.18)$$

Pro výpočet Eulerovy funkce se používá rozklad čísla *n* na součin prvočísel. Není však prozatím znám žádný efektivní způsob výpočtu Eulerovy funkce bez znalosti rozkladu čísla *n.* Pokud by byl objeven algoritmus, pomocí kterého by bylo možno prvočíslo rozdělit na součin prvočísel, byla by dramaticky ohrožena bezpečnost algoritmu RSA, která je na této současné neschopnosti založena [17].

Diskrétní logaritmus

Tato funkce tvoří jednu ze základních podmínek bezpečnosti asymetrických algoritmů. V případě, že *a, g, μ, n* jsou přirozená čísla, je výpočet členu *a* velice jednoduchý. Každý výpočet čísla *μ* (vzhledem k modulu jich je nekonečně mnoho) je pak diskrétním logaritmem o základu *g* z *a*.

$$a \equiv g^{\mu} \pmod{n} \qquad (2.19)$$

Pro kryptografii má velký význam náročnost výpočtu právě diskrétního logaritmu *μ*, i pokud známe ostatní členy rovnice. [23]

Základní logické operace

Logické operace tvoří základ počítačových systémů. Bez jejich použití by se dnes neobešla žádná výpočetní technika a stejně tomu tak je i v kryptografii. Zde se kromě standardních operací jako jsou sčítání, odčítání, násobení a dělení členů využívají i logické operace jako NOT, AND, OR a XOR.

NOT

Jde o logickou operaci negace. Vstupní bit je tedy logicky negován a poslán na výstup. Výstupní hodnota je pravda, když je vstupní hodnota nepravda a naopak. V následující tabulce je pravdivostní tabulka funkce NOT společně s možnými zápisy této operace:

X	Y
0	1
1	0

$Y = \bar{X}$
$Y = \neg X$

AND

Logická operace součinu neboli konjunkce. Výstupní hodnota je pravda pouze tehdy, pokud jsou obě vstupní hodnoty rovny hodnotě pravda. V následující tabulce je pravdivostní tabulka funkce AND společně s možnými zápisy této operace:

A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

$Y = A \cdot B$
$Y = A\&B$
$Y = A \wedge B$

OR

Logická operace součtu neboli disjunkce. Výstup této logické operace je pravda, pokud alespoň jeden ze vstupů je pravda. V následující tabulce je pravdivostní tabulka funkce OR společně s možnými zápisy této operace:

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1

$Y = A + B$
$Y = A \vee B$

XOR

Logická operace exkluzivního součtu neboli exkluzivní disjunkce. Výstup této logické operace je pravda, pokud jsou oba vstupy různé. V následující tabulce je pravdivostní tabulka funkce XOR společně s možnými zápisy této operace:

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	0

$Y = A \oplus B$

Tato funkce je v kryptografii ze všech nejpoužívanější. Vidět jí můžeme například v algoritmech DES a AES, nebo také v módech blokových šifer.

2.4 Současná kryptografie

S nástupem počítačové techniky a internetu dostal pojem kryptologie zcela nový rozměr. Význam tohoto oboru roste neuvěřitelnou rychlostí tak, jak se vyvíjí počítačová technika a s tím spojený výpočetní výkon. Ten je totiž v případě kryptologie spíše nepřítel, protože umožňuje tzv. útoky hrubou silou pomocí testování všech iterací klíčů. S výpočetním výkonem se tak zkracuje doba potřebná k takovémuto rozluštění kryptogramu.

Proto jsou neustále vyvíjeny lepší a silnější algoritmy, používající stále vícebitová klíče. Dalo by se říci, že moderní kryptologie se vydává 3 odlišnými cestami, z nichž se každá hodí k něčemu jinému. Jsou to *symetrická kryptografie*, *asymetrická kryptografie*. Můžeme zmínit také *kvantovou kryptografii*, jež se využívá je pro bezpečný přenos klíčů [4].

2.4.1 Úlohy a využití kryptografie

Kryptografie jako taková se v dnešní době používá v podstatě k 3 zásadním úlohám.

Důvěryhodnost

První a pravděpodobně nejznámější je tzv. *udržení důvěryhodnosti dat.* V praxi to znamená zamezení neoprávněným uživatelům dostat se k utajeným datům. To lze zajistit právě nejrůznějšími kryptografickými metodami spojenými s doplňkovou ochranou. Typickým příkladem by zde mohl být zašifrovaný dopis uložený v trezoru, ke kterému budou mít klíč pouze oprávněné osoby.

Integrita

Druhé využití kryptografie se využívá zejména v bankovním sektoru a v místech, kde je nutné, aby zpráva dorazila neporušená (nepozměněná) z místa vyslání do místa určení. Tomuto typu ochrany se říká *zabezpečení integrity dat*. Problém porušení tohoto zabezpečení by mohl mít fatální následky pro komunikaci vůbec. Pokud si nemůže být příjemce zprávy jist, že k němu zpráva dorazila nepozměněná, nemůže potom brát zprávu jako relevantní. Jako příklad bych uvedl elektronické bankovnictví, kde se odesílají jak čísla účtů, tak převáděné částky. Pokud by v této zprávě došlo k jakékoliv úpravě, mělo by to za následek doručení peněz na špatný účet, nebo v horším případě vyšší částku. Je proto nepřípustné, aby bylo něco takovéhoto možné. Proto je bankovní komunikace jednou z nejzabezpečenějších.

Autentizace

Třetí základní možností využití kryptografie je tzv. *autentizace*. Ta slouží k vzájemnému ověření totožnosti při elektronické komunikaci. Je to základní prvek bezpečné komunikace a jako uživateli mi to zajistí to, že při správné autentizaci budu vždy vědět, s kým komunikuji. Nemusí však jít pouze o ověření totožnosti osob, ale také programů, procesů, počítačů atp.

Není tedy možné zneužití například elektronického bankovnictví. Samozřejmě za předpokladu, že je použit vhodný kryptografický algoritmus s dostatečně silným a

velkým klíčem. S výše uvedenými metodami úzce souvisí využití kryptografie, přičemž se ve většině případů kombinuje více těchto metod.
Za autentizaci však nemusíme považovat pouze to, že uživatel zná například přihlašovací údaje a heslo. Toto je pouze jeden ze 4 typů autentizace. Druhou možností autentizace je využití toho, co uživatel vlastní. Autentizace je tedy možná například pomocí USB klíčenky s certifikátem, nebo čipové karty. Třetí možností jsou vlastní tělesné parametry uživatele. K takovéto autentizaci se používá například skener rohovky nebo biometrický senzor otisků prstů. Čtvrtou a poslední možností je kontrola znalostí uživatele – tedy správná odpověď na zadanou otázku.
V pravém slova smyslu však s počítačovou kryptografií souvisí pouze první 2 možnosti. Třetí a čtvrtá možnost jsou vysoce specifické vlastnosti, které není třeba šifrovat [3].

- Autentizace dat – slouží k ověření pravosti a identity dat, jejich původ, autor, obsah, datum vzniku atd.
- Řízení přístupu – zajišťuje oprávněnost přístupu ke chráněným datům
- Nepopiratelnost – pokud je předmětem sporu oboustranně digitálně podepsaný dokument, lze poté říci, že daný dokument byl odeslán, doručen, přečten atd.
- Vzájemná autentizace – ověření totožnosti v rámci digitálních podpisů
- Podpis smluv – smlouvy lze podepisovat více stranami najednou a to pouze v digitální rovině => výše uvedená nepopiratelnost [11, 12]

2.5 Symetrické šifry

Symetrická šifra používá ke kódování i dekódování stejný klíč, nebo 2 různé klíče, které jsou však jeden z druhého odvoditelné. Pro klíče K_1 a K_2 tedy musí platit:

$$K_1 = K_1' \tag{2.20}$$

Výhodou symetrické kryptografie je rychlost jejího kódování i dekódování. Nevýhodou pak je distribuce klíčů. Jelikož obě strany používají stejný klíč, nemají možnost si tento klíč bezpečně sdělit. Pro distribuci klíčů symetrického šifrování je možné využít asymetrického šifrování, o kterém se zmíním níže.

Symetrické šifry se dělí na 2 základní větve podle typu zpracování otevřeného textu.

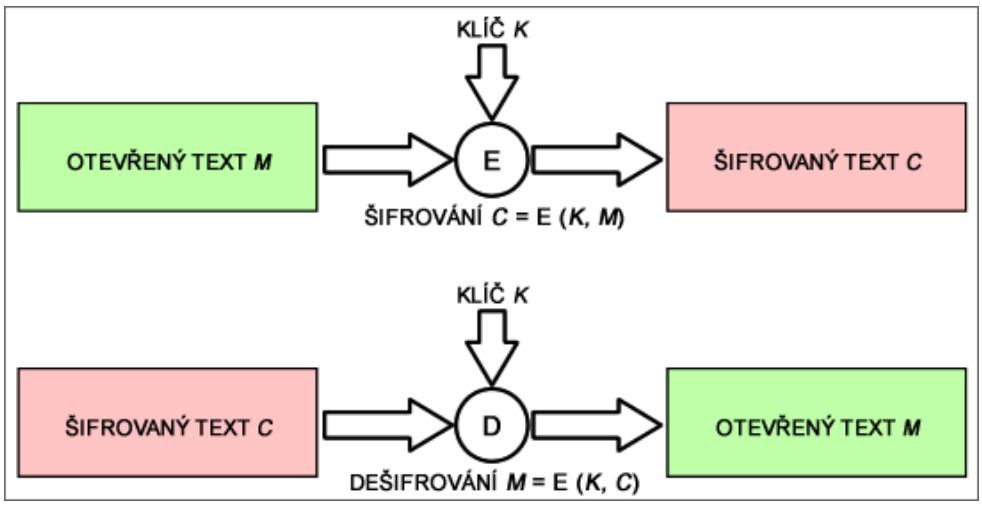
- Proudové šifry – zpracovávají otevřený text po bitech (RC4, FISH)
- Blokové šifry – zpracovávají otevřený text po stejně velkých blocích (AES, DES, 3DES)

Základní vlastnosti symetrického šifrovacího algoritmu:

Úplnost
Výstupní bity jsou dány kombinací všech vstupních bitů podle určité funkce, která není lineární. Pokud by tato funkce byla lineární, byly by výstupní bity pouze jednoduchou odvozeninou vstupních bitů a k rozluštění by stačila pouze soustava rovnic.

Neexistence korelace
Jedním ze základních pravidel je neexistence vztahu mezi otevřeným textem a zašifrovanou zprávou (kryptogramem). Nesmí také existovat vztah mezi kryptogramem a šifrovacím klíčem. Pokud by tato podmínka nebyla splněna, stačilo by pouze nalézt vztah mezi klíčem a kryptogramem. Jeho rozluštění by poté nebyl větší problém.

Lavinovitost
Tento pojem znamená to, že i změna jediného bitu ve vstupním bloku dat vyvolá na výstupu změnu ve více než jednom bitu. Toto znesnadňuje možnost zpětného odvození vstupních stavů [6].

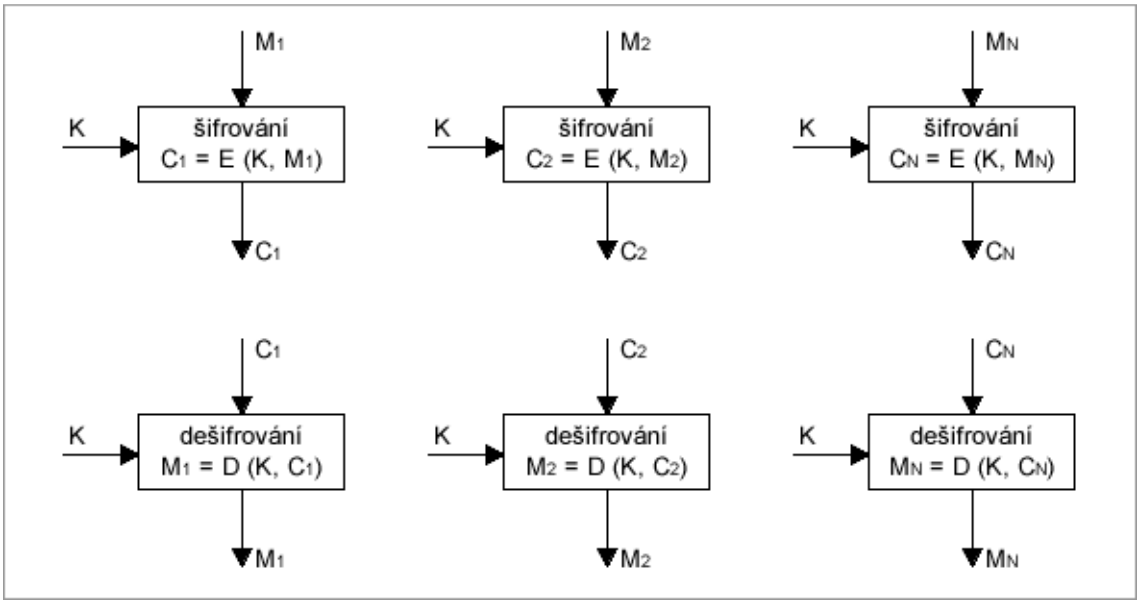


Obr. 2.2 Šifrování a dešifrování pomocí symetrické šifry

2.5.1 Režimy blokových šifer
Blokové režimy symetrických algoritmů můžeme realizovat více způsoby. Po realizaci návrhu algoritmu DES byly 4 režimy doplněny na konečných 5, které se využívají i v dnešním standardu pro symetrickou kryptografii, algoritmus AES. Některé z režimů blokových šifer jsou velmi jednoduché a méně bezpečné, jiné zase složitější, ale časově náročnější. V následující podkapitole budou jednoduše vysvětleny všechny režimy od těch nejjednodušších po složitější [17, 21].

Elektronická kódová kniha (režim ECB)
Nejjednodušší režim blokových šifer, kdy je vstupní text rozdělen na stejně dlouhé bloky. Poslední blok bývá kratší, a proto bývá doplněn na stejnou délku. Algoritmus následně šifruje jednotlivé bloky vstupního textu stejným způsobem za použití stejného

klíče pro všechny bloky. Jednotlivé bloky se šifrují postupně, nejsou tedy proházeny. Dešifrování probíhá analogicky.



Obr. 2.3 Režim blokové šifry ECB, šífrování a dešifrování [17]

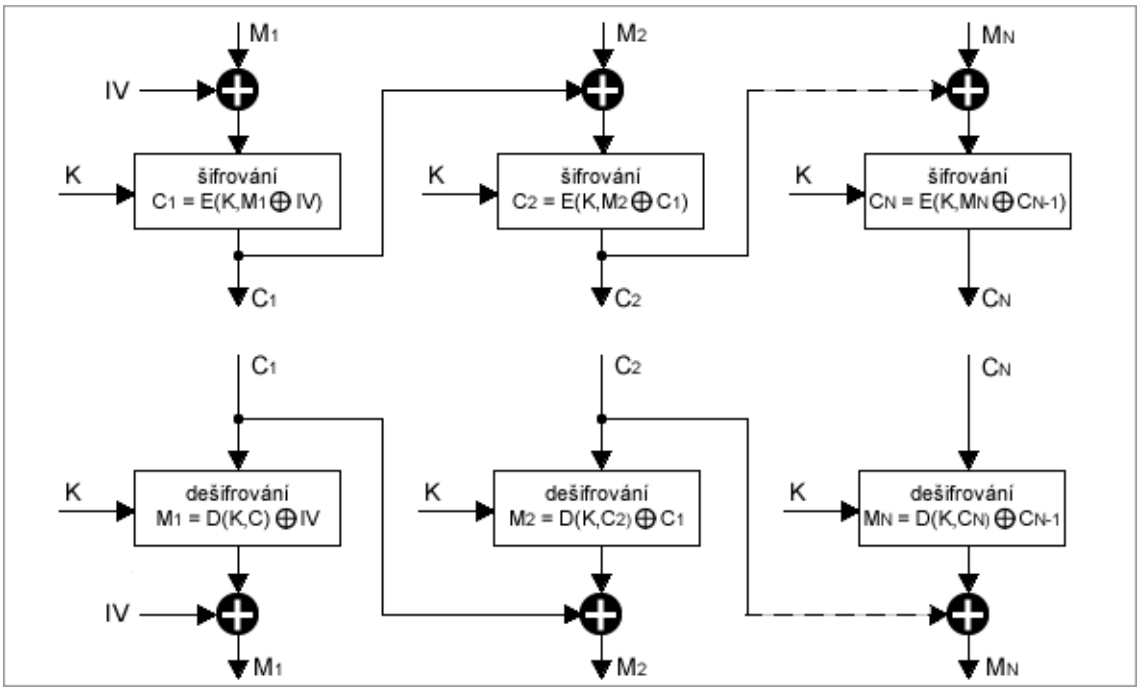
Nevýhodou systému je, že vstupní blok textu i výstupní blok šifry je stejně velký. To velice zjednodušuje kryptoanalýzu a tím snižuje bezpečnost celého algoritmu.

Tento režim blokové šifry je tedy vhodné použít, pokud je šifrovaný text kratšího charakteru. V kratším textu totiž pomocí kryptoanalýzy nelze detekovat dostatečný počet opakování bloků na to, aby mohlo být heslo rozluštěno a bylo to způsobeno právě režimem blokové šifry [17, 21].

Zřetězení zašifrovaného textu (režim CBC)

Tento režim blokové šifry nám umožňuje získat ze stejných vstupních bloků různé výstupy. Toho je dosaženo tím, že výstup jednoho bloku je přiveden na vstup následujícího bloku, kde je s ním provedena operace XOR. Vstupní text do šifrátoru je tedy v každém bloku jiný, i když jsou vstupní zprávy stejné.

Dešifrování probíhá přesně opačným způsobem. Vstup do dešifrátoru je zároveň přiveden na výstup dalšího bloku, kde je opět provedena operace XOR. Na obrázku 2.4 můžeme vidět na vstupech prvního bloku označení IV. Toto označení je inicializační vektor. Ten je třeba pro zahájení šifrování i dešifrování v režimu blokové šifry CBC. Inicializační vektor na straně šifrování i dešifrování musí být stejný.



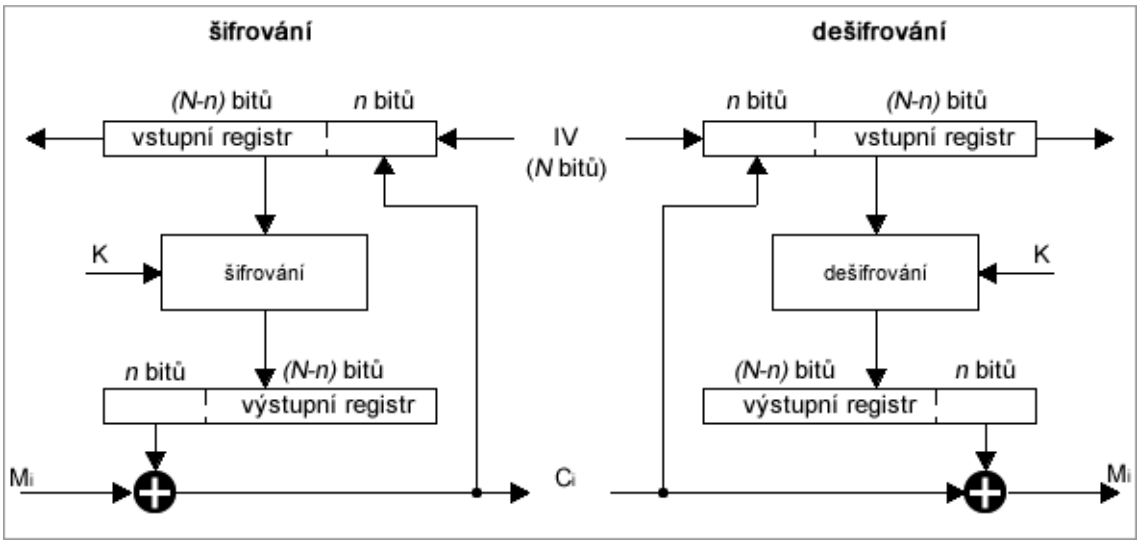
Obr. 2.4 Režim blokové šifry CBC, šifrování a dešifrování [17]

Tento režim blokové šifry se tedy hodí i pro kratší zprávy, kde se navíc často opakují bloky textu, nebo jsou velice podobné [17, 21].

Zpětná vazba ze zašifrovaného textu (režim CFB)

Tento režim blokové šifry rozdělí bloky vstupního textu na subbloky, které následně šifruje. Pokud tedy například máme blok velký 64 bitů, pak se šifrování provádí po blocích velkých 8 bitů, tedy po jednom bajtu. Pokud bychom šli ještě do větší hloubky, tak blok zprávy velký 8 bitů bude rozdělen na subbloky velké 1 bit. To znamená, že pomocí tohoto režimu lze použít blokové šifry jako proudové – tedy šifrování po jednom bitu.

Na začátku se vstupní text rozdělí na bloky o příslušné velikosti, tak, jak je tomu u blokových šifer zvykem. Každý blok se následně rozdělí na subbloky, se kterými se následně pracuje. Na vstupu algoritmu je vstupní posuvný registr. Celý registr je vybrán a šifrován pomocí klíče. Na výstupu je opět posuvný registr, jehož nejvyšších *n* bitů je přivedeno na operaci XOR společně s původním subblokem textu. Výsledkem je subblok šifrované zprávy.



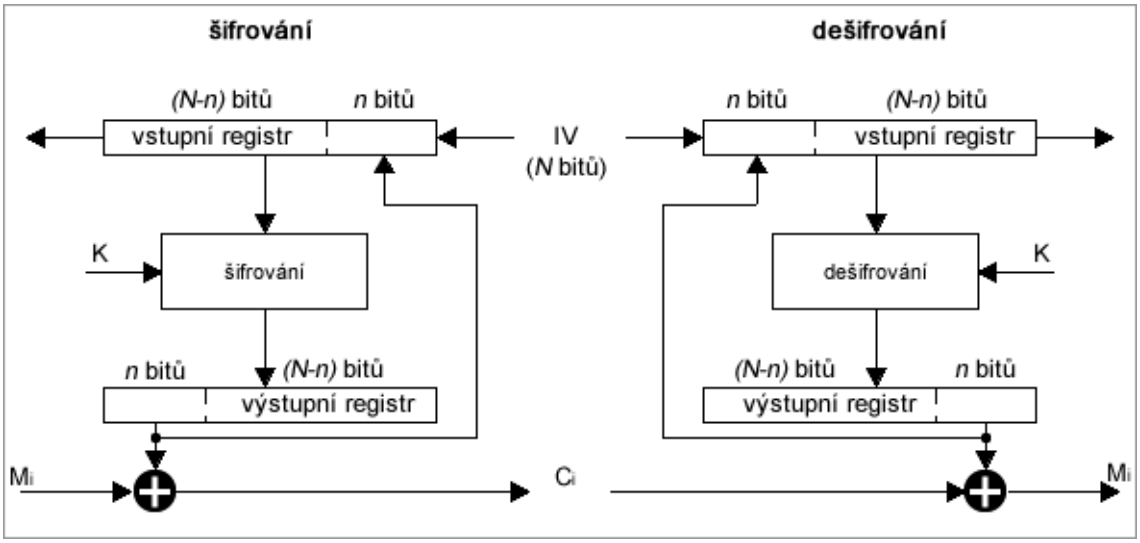
Obr. 2.5 Režim blokové šifry CFB, šifrování a dešifrování [17]

Na začátku šifrovacího procesu je do vstupního registru umístěn inicializační vektor, ten je v dalších krocích algoritmu nahrazen zašifrovaným subblokem textu. Vše je názorně zobrazeno na obrázku 2.5.

Proces dešifrování je analogicky přesně obrácený oproti procesu šifrování. Nevýhodou této realizace je, že pokud dojde při přenosu dat k chybě, jsou díky zpětné vazbě ovlivněny i ostatní bloky zprávy. Při dešifrování je tak ovlivněn nejen následující blok, ale dle nastavení velikosti bloků a subbloků i bloky další [17, 21].

Zpětná vazba z výstupu (režim OFB)

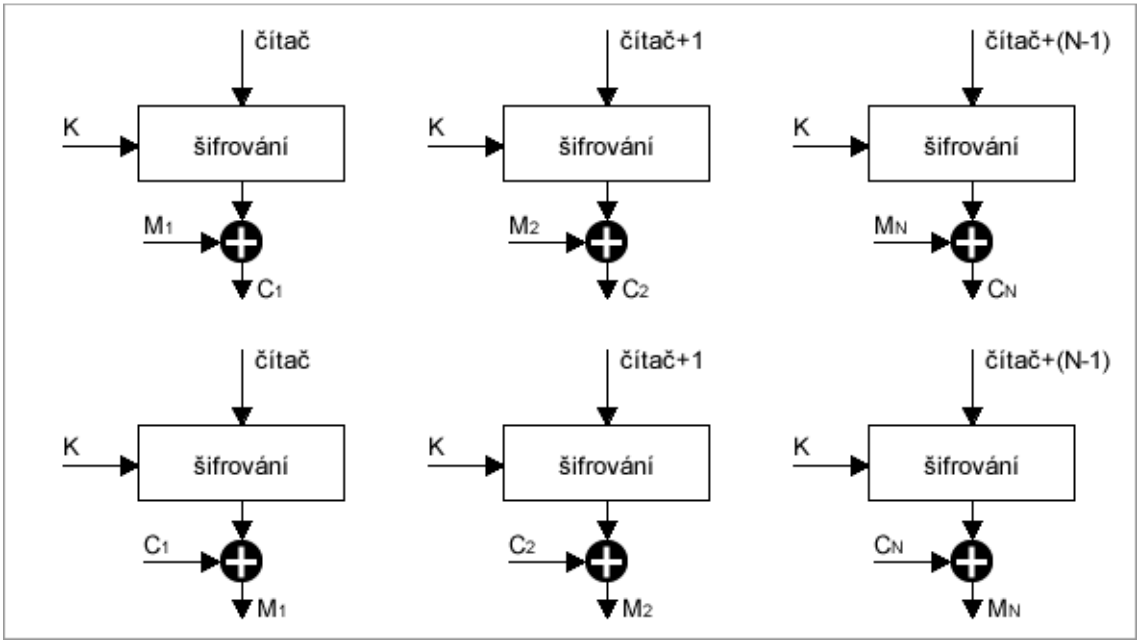
Režim blokových šifer OFB je téměř identický s režimem CFB. Jediným rozdílem v algoritmu je to, že zpětná vazba z výstupu je umístěna ještě před koncovou operaci XOR Jeho výhodou je, že chyba způsobená přenosem ovlivní jenom daný bit a nešíří se napříč všemi bloky [17, 21].



Obr. 2.6 Režim blokové šifry OFB, šifrování a dešifrování [17]

Čítačový režim (CTR)

Poslední režim blokových šifer kombinuje jednoduchost režimu ECB s bezpečností pokročilých režimů, jako jsou CFB a OFB. Velikost vstupního a výstupního bloku je opět stejný. Oproti ECB se však ke každému bloku přidává funkcí XOR číslo z externího čítače. Tento čítač se vždy před začátkem šifrování nastaví na počáteční hodnotu, která se s každým šifrovaným blokem inkrementuje.



Obr. 2.7 Režim blokové šifry CTR, šifrování a dešifrování [17]

Pro dešifrování je nutné použít stejné přednastavení čítače, jako při šifrování. Tento režim je velice snadný na implementaci a používá se zejména síťové bezpečnosti [17, 21].

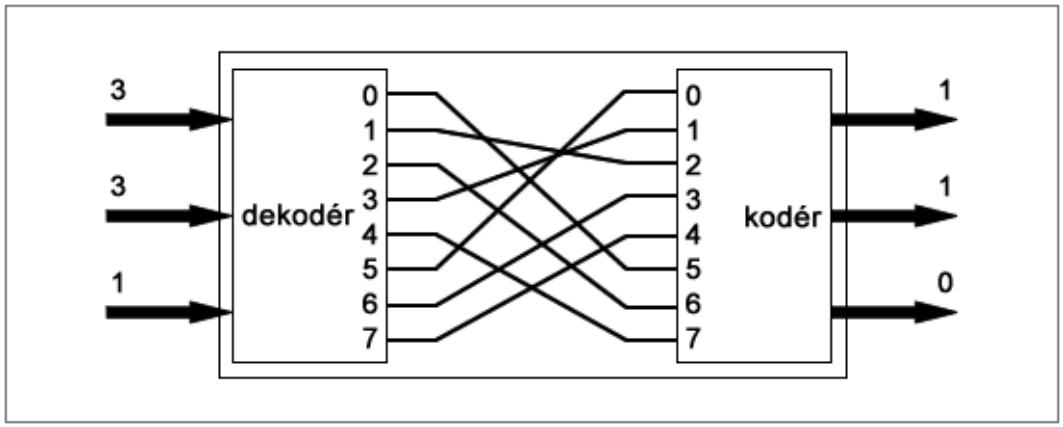
2.5.2 Algoritmus DES

Algoritmus DES byl patentován v polovině 70. let a byl šifrovacím standardem až do roku 1998. Šifrovací algoritmus DES vychází z původního interního šifrovacího algoritmu firmy IBM. Šlo o velice dobrý a bezpečný algoritmus s klíčem dlouhým 128 bitů. Při uvedení do veřejného provozu však došlo ke zkrácení bezpečnostního klíče na pouhých 56 bitů. Důvodem byly obavy společnosti NSA, že DES s dlouhým klíčem by byl příliš bezpečný a nezlomitelný a proto by byl vhodný pro nelegální činnost – což je paradox [25].

K tomu, aby byl šifrovací algoritmus DES schopen fungovat a zajistit základní podmínky popisované výše, používá 3 speciální bloky pro práci s daty.

S-Box

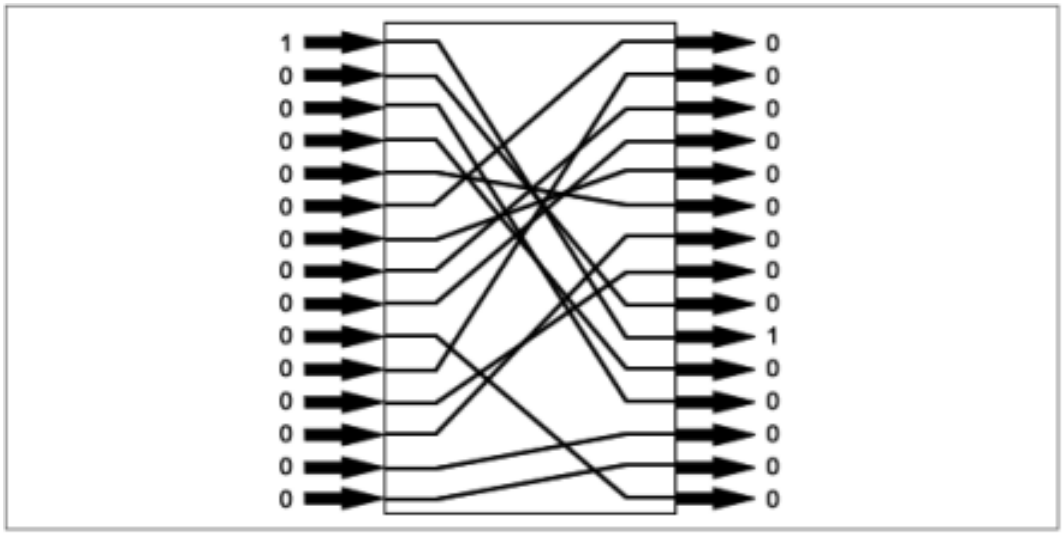
Jde v podstatě o substituční funkci. Podle dat na jejím vstupu určí výstupní kombinaci bitů. Tato závislost je dána vnitřní implementovanou funkcí S-Boxu. U algoritmu DES se využívá toho, že vstupní a výstupní počet bitů se liší [25].



Obr. 2.8 Příklad zapojení S-Boxu [25]

P-Box

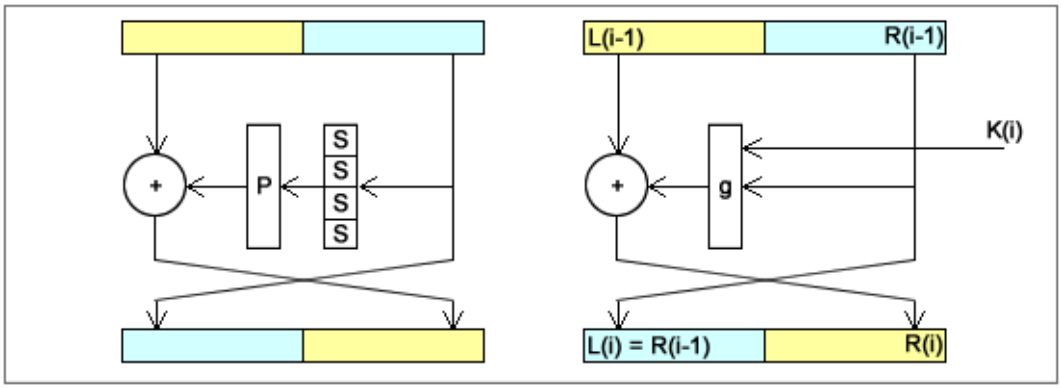
Plní obdobnou funkci jako S-Box. Svou funkci má však pevně danou. Úkolem P-Boxu je zpřeházení pořadí bitů v daném bloku [25].



Obr. 2.9 Příklad zapojení P-Boxu [25]

Feistelova síť

Znázorňuje vlastní průběh šifrovacího algoritmu včetně použití jednotlivých bloků, přesunů bitů a použití klíče. Slouží k lepšímu a názornějšímu pochopení funkce šifrovacího algoritmu [25].



Obr. 2.10 Částečná Feistelova síť pro algoritmus DES [25]

DES pracuje s bloky dat o délce 64 bitů při délce klíče 56 bitů. Úvodním krokem šifrování je počáteční permutace pomocí permutační tabulky. Poté následuje 16 kroků šifrování pomocí 16 podklíčů. Ty jsou vygenerovány ze základního klíče. Na závěr je provedena konečná permutace a výsledkem tohoto postupu je kryptogram.
Výhodou algoritmu DES je, že při dešifrování kryptogramu je postup přesně opačný než při šifrování. Není tedy nutné měnit funkce a programovat nové postupy. Stačí pouze, aby bylo vše provedeno pozpátku [25].

Úvodní permutace

Pro úvodní permutaci se používá následující tabulka. Na vstupu je 8 bitové číslo (část vstupních dat). To určuje indexy v permutační tabulce 4 bity sloupec a 4 bity řádek [25].

Tab. 2.4 Úvodní permutační tabulka pro algoritmus DES [25]

58	50	42	34	26	18	10	2
60	52	44	36	28	20	12	4
62	54	46	38	30	22	14	6
64	56	48	40	32	24	16	8
57	49	41	33	25	17	9	1
59	51	43	35	27	19	11	3
61	53	45	37	29	21	13	5
63	55	47	39	31	23	15	7

1/16 šifrovacího kroku

Vstupní blok dat o velikosti 64 bitů se rozdělí na levou a pravou polovinu o velikosti 32 bitů. Pravá strana je dále rozšířena pomocí expanzní permutace na 48 bitů tak, aby velikostně odpovídala velikosti subklíče pro daný šifrovací krok [25].

Tab. 2.5 Permutační tabulka pro 1 rundu algoritmu DES [25]

32	1	2	3	4	5
4	5	6	7	8	9
8	9	10	11	12	13
12	13	14	15	16	17
16	17	18	19	20	21
20	21	22	23	24	25
24	25	26	27	28	29
28	29	30	31	32	1

Dále se pomocí funkce XOR spojí s daným subklíčem. Výsledkem je tedy opět blok o velikosti 48 bitů. Tento blok je následně rozdělen na 8 částí, přičemž každá část vstupuje do jednoho S-Boxu. Výstupy z jednotlivých S-Boxů jsou opět spojeny dohromady a nyní tvoří blok velký 32 bitů. Následně se provede další permutace [25].

Tab. 2.6 Výstupní permutace algoritmu DES [25]

16	7	20	21
29	12	28	17
1	15	23	26
5	18	31	10
2	8	24	14
32	27	3	9
19	13	30	6
22	11	4	25

Po provedení předchozích kroků s pravou stranou zprávy se provede ještě funkce XOR s původní levou polovinou zprávy. Tímto získáme finální podobu výstupní pravé strany. Výstupní levou stranu zprávy získáme okopírováním vstupní pravé strany.

Tento postup se opakuje 16-krát po sobě, pokaždé s jiným subklíčem.

Výstupní permutace

Po provedení předchozí části algoritmu se na závěr provede permutace, která je inverzní k té vstupní. Výsledkem celého algoritmu je zašifrovaná zpráva, neboli kryptogram [25].

Nevýhody algoritmu DES

Existuje možnost, že při volbě určitého klíče nastane situace, kdy kryptogram vypadá stejně jako šifrovaný vstupní text. Tento problém se odvíjí od operací uvnitře algoritmu. Je tedy nutné se takovýmto klíčům vyhnout [12].

Tab. 2.7 Příklady slabých klíčů pro algoritmus DES [25]

Slabý klíč (hex)	C0	D0
0101 0101 0101 0101	$\{0\}^{28}$	$\{0\}^{28}$
FEFE FEFE FEFE FEFE	$\{1\}^{28}$	$\{1\}^{28}$
1F1F 1F1F 1F1F 1F1F	$\{0\}^{28}$	$\{1\}^{28}$
E0E0 E0E0 E0E0 E0E0	$\{1\}^{28}$	$\{0\}^{28}$

Tab. 2.8 Příklady poloslabých klíčů pro algoritmus DES [25]

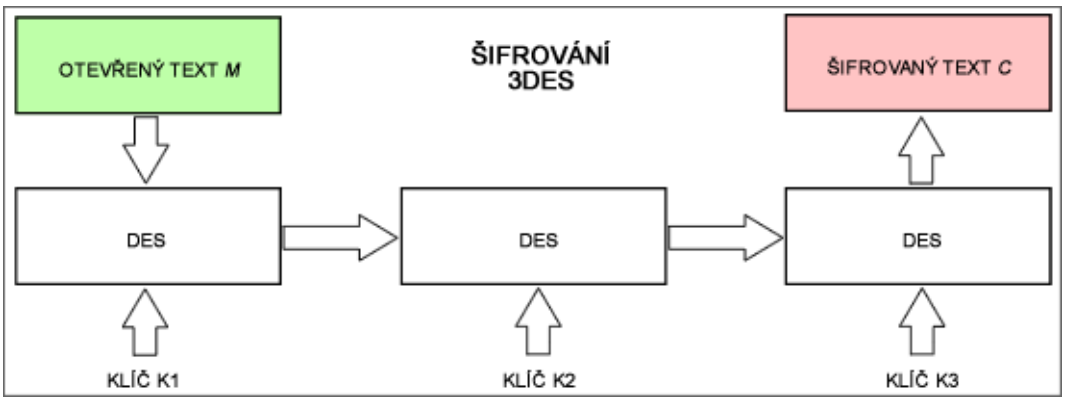
C0	D0	Pár poloslabých klíčů (hexa)	C0	D0
$\{01\}^{14}$	$\{01\}^{14}$	01FE 01FE 01FE 01FE, FE01 FE01 FE01 FE01	$\{10\}^{14}$	$\{10\}^{14}$
$\{01\}^{14}$	$\{10\}^{14}$	1FE0 1FE0 0EF1 0EF1, E01F E01F F01E F01E	$\{10\}^{14}$	$\{01\}^{14}$
$\{01\}^{14}$	$\{0\}^{28}$	01E0 01E0 01F1 01F1, E001 E001 F101 F101	$\{10\}^{14}$	$\{0\}^{28}$
$\{01\}^{14}$	$\{1\}^{28}$	1FFE 1FFE 0EFE 0EFE, FE1F FE1F FE0E FE0E	$\{10\}^{14}$	$\{1\}^{28}$
$\{0\}^{28}$	$\{01\}^{14}$	011F 011F 010E 010E, 1F01 1F01 0E01 0E01	$\{0\}^{28}$	$\{10\}^{14}$
$\{1\}^{28}$	$\{01\}^{14}$	E0FE E0FE F1FE F1FE, FEE0 FEE0 FEF1 FEF1	$\{1\}^{28}$	$\{10\}^{14}$

Bezpečnost algoritmu DES

V dnešní době (2010) je algoritmus považován za nespolehlivý a snadno prolomitelný. Mohou za to zejména chyby v návrhu a délka klíče pouhých 56 bitů. Dnešními prostředky je algoritmus prolomitelný hrubou silou za méně jak 24 hodin.

2.5.3 Algoritmus 3DES

Algoritmus 3DES je v podstatě rozšířenou verzí algoritmu DES. Jde o trojnásobné použití algoritmu za sebou. Pro každý šifrovací krok se může použít buď rozdílný klíč, nebo se využívá možnosti kombinace dvou klíčů. Poprvé se data zašifrují klíčem K_1, podruhé klíčem K_2 a potřetí opět klíčem K_1. Při použití tří různých klíčů tak dostáváme výsledný klíč o délce 168 bitů. Při použití dvou klíčů je to 112 bitů dlouhý klíč. Nevýhodou tohoto algoritmu je dnes již zastaralé a pomalé jádro. Proto se dnes téměř již nepoužívá [12].



Obr. 2.11 Šifrování pomocí algoritmu 3DES

Bezpečnost algoritmu 3DES

Algoritmus 3DES vznikl v době, když stále ještě nebyl k dispozici nástupce DESu a bylo třeba zvýšit bezpečnost. V reálu se nepoužívají tři různé klíče, ale pouze klíče dva, přičemž první a poslední šifrování se dělá se stejným klíčem. Délka klíče tedy nenaroste třikrát, ale pouze dvakrát, což opět degraduje bezpečnost algoritmu. Vzhledem k použití stejného algoritmu jsou v něm obsaženy také konstrukční chyby.

2.5.4 Algoritmus AES

Algoritmus AES, neboli Advanced Encryption Standard se stal nástupcem algoritmu DES, který byl na konci 90. let označen za slabý a prolomitelný. U algoritmu AES byla zvolena jiná strategie výběru. Mohly se do něj zapojit jakékoliv veřejné i soukromé subjekty se svými algoritmy. Vítězem soutěže, která trvala 4 roky, se stal algoritmus Rijndael, který ze všech algoritmů splňoval dané podmínky téměř ideálně.

AES disponuje volitelnou délkou kroku i klíče a to nevázaně na sobě. Velikosti jsou volitelné ve třech krocích 128, 192 a 256 bitů. Podobně jako v DESu se zde vnitřní operace násobně opakují. Počet opakování je závislý na zvolené velikosti vstupního bloku.

SubBytes

Prvním krokem při zpracování jedné rundy alegoritmu AES je operace nelineární substituce. Ta se aplikuje nezávisle na každý blok matice podle substituční tabulky S-box, zmíněné již v algoritmu DES

ShiftRows

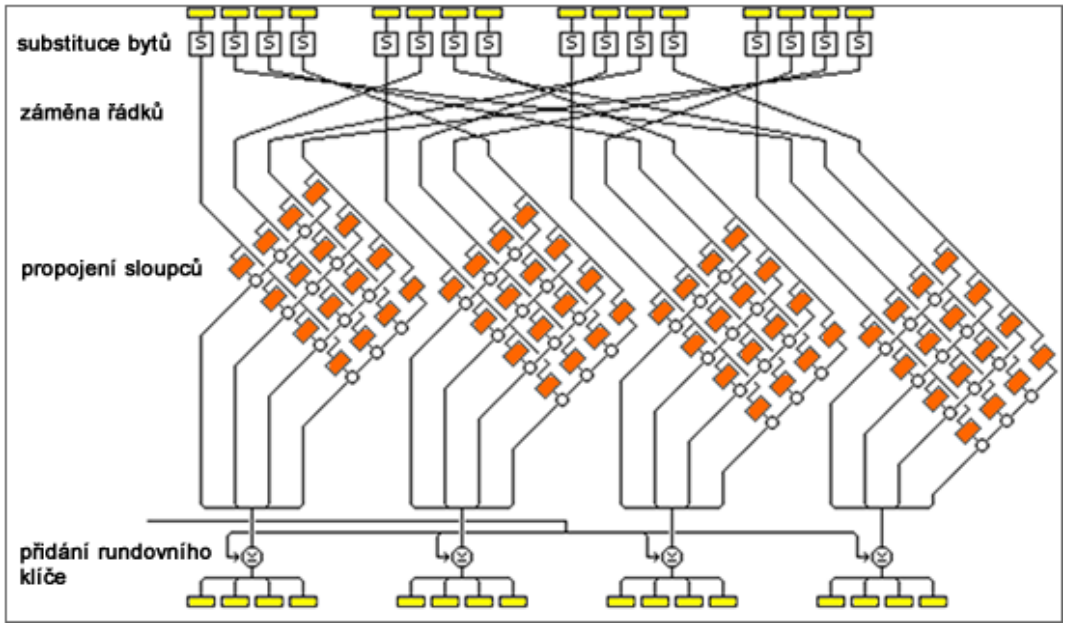
Provede posun prvků v řádku o n-1 pozic doleva. Pokud jsme na prvním řádku, výsledek zůstává stejný. Na druhém řádku se provede posun o jeden prvek doleva a tak dále.

MixColumns

Operace se sloupci vzniklé matice, které se násobí jako polynom s polnomem a po modulo funkcí $x^4 + 1$ obdržíme opět polynom 3 stupně.

AddRoundKey

Rundovní klíč projde funkcí XOR postupně se všemi sloupci matice State. Rundovní klíč se generuje za pomoci nelineární expanze z primárního klíče [12].



Obr. 2.12 Šifrování pomocí algoritmu AES [22]

Bezpečnost algoritmu AES

Prozatím není znám žádný realizovatelný útok, který by bylo možno aplikovat na prolomení bezpečnosti algoritmu AES. Při útoku na základní verzi algoritmu s klíčem 128bitů bychom museli vyjádřit 8000 rovnic o 1600 neznámých. Není znám útok s menší časovou náročností než 2^{128}.

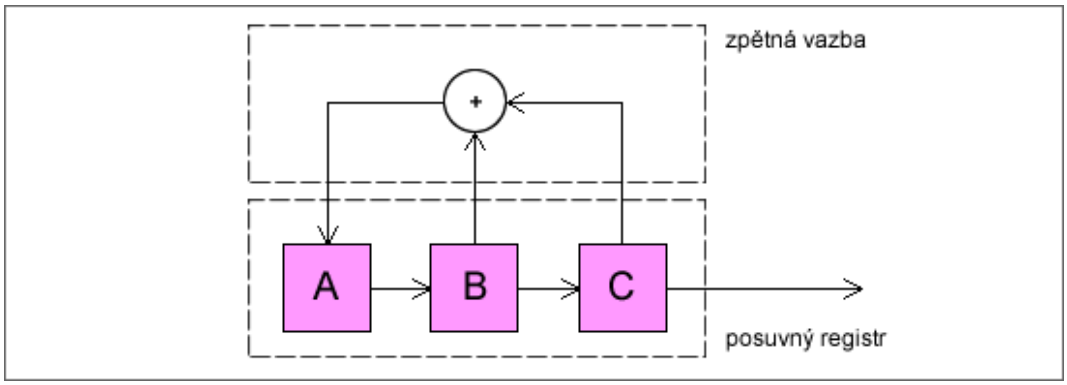
2.5.5 Generátory náhodných čísel

Náhodná posloupnost

Posloupnost bitů, která je v celé své délce dokonale náhodná. To znamená, že na základě známých prvků nejsme schopni vypočíst následující prvek. Pro generování náhodných posloupností se většinou používají náhodné fyzikální děje, jako je například teplotní šum. Právě výstup z generátoru, založeném na fyzikálních dějích lze jako jediný považovat za naprosto náhodný. Ve většině případů však tyto jevy nejde použít v praxi, a proto jsou konstruovány generátory pseudonáhodných posloupností. Ty sice negenerují naprosto náhodnou posloupnost, ale jejich výstup lze ze statistického hlediska považovat za náhodný.

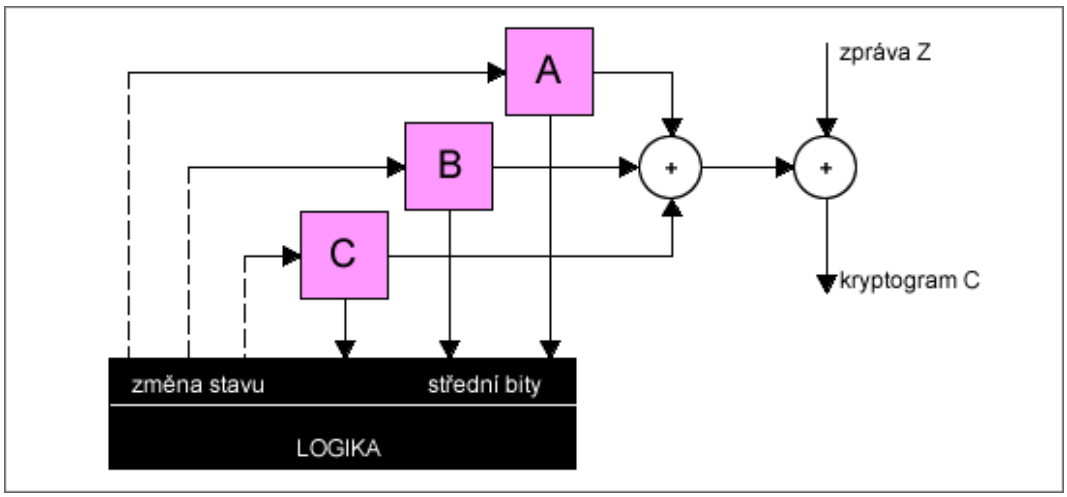
Pseudonáhodná posloupnost

Posloupnost bitů, která se jeví jako zcela náhodná, avšak generuje se na základě předem stanovených pravidel. Tato posloupnost se generuje v generátoru pseudonáhodné posloupnosti. Jako úvodní se do generátoru pseudonáhodné posloupnosti nastavují počáteční stavy, které jsou v podstatě heslem. Právě tento počáteční stav jednoznačně určuje další běh generátoru. Generátory pseudonáhodných posloupností mohou být lineární a nelineární [1].



Obr. 2.13 Lineární generátor PNP [1]

Funkce, která generuje pseudonáhodnou posloupnost, musí být jednosměrná. Právě její jednosměrnost nám zaručuje bezpečnost algoritmu.
V praxi se lineární generátor PNP jako takový příliš nepoužívá, protože je z hlediska kryptoanalýzy málo bezpečná. Bezpečnější je verze, kdy se jednotlivé lineární generátory nelineárně kombinují.

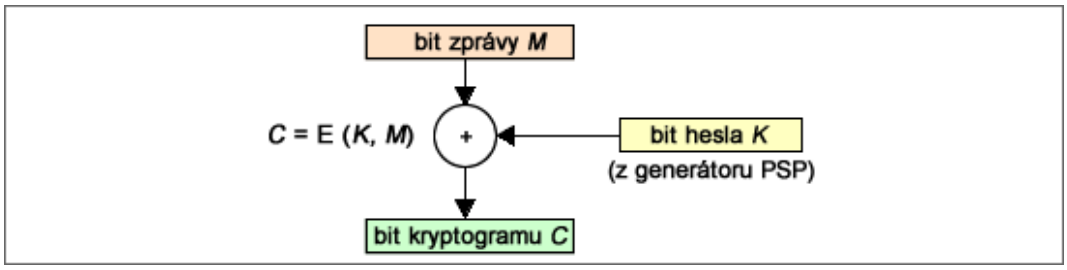


Obr. 2.14 Nelineární generátor PNP [1]

Na obrázcích 2.13 a 2.14 jsou znázorněny generátory PNP. Na počátku generování pseudonáhodné posloupnosti je třeba registr (paměťová místa A, B, C) naplnit počátečními hodnotami – nejčastěji vlastním heslem [1].

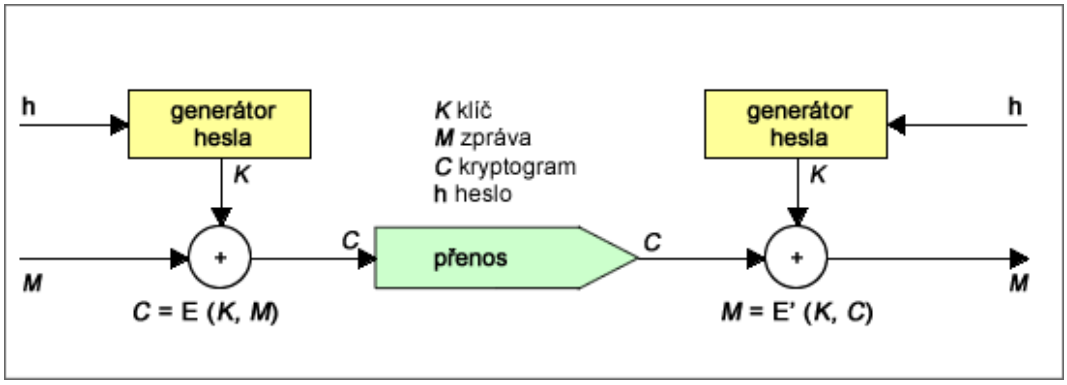
2.5.6 Algoritmus RC4

Tento algoritmus patří mezi proudové šifry. Byl vyvinut v roce 1987, zveřejněn v roce 1994. Jejich hlavním rozdílem oproti blokovým šifrám je zpracování vstupních dat po jednotlivých bitech, nikoliv po blocích. Z toho se odvíjí také jejich hlavní výhoda, tou je real-time zpracování dat. To je nutnost například v telekomunikacích nebo online přenosech. Výhodou algoritmu RC4 je jeho snadná softwarová implementace. Tento algoritmus již není dnes považován za bezpečný, přesto je použit v protokolu SSL pro zabezpečení webu a protokolech WEP a WPA pro zabezpečení bezdrátových sítí [12].



Obr. 2.15 Šifrování pomocí algoritmu RC4

Nevýhodou algoritmu je nutnost na obou stranách (šifrovací i dešifrovací) generovat stejný náhodný klíč, nebo po zašifrování klíč předat. Předání klíče však snižuje bezpečnost algoritmu a degraduje výhodu zpracování v reálném čase. Proto se využívají tzv. generátory pseudonáhodného kódu. Ty si po dohodě základních koeficientů dokáží generovat náhodnou řadu čísel (heslo), které je stejné na obou stranách. K sestavení těchto koeficientů se používá například algoritmus Diffie-Hellman.



Obr. 2.16 Šifrování pomocí algoritmu RC4 s generátory PNP

Bezpečnost algoritm RC4

Algoritmus RC4 již není považován za bezpečný. I přes to je používán k šifrování například GSM přenosů (v podobě šifry A5). Ta je luštitelná dokonce v reálném čase na dnes běžném PC. Doba potřebná pro odhalení klíče je řádově měřena na minuty. Druhou možností je nabourání tzv. „man in the middle“ jelikož algoritmus neověřuje s kým komunikuje. Vstupem mezi komunikující strany můžeme podvrhnout identitu jedné z komunikujících stran.

2.6 Asymetrické šifry

Na rozdíl od symetrické kryptografie používá asymetrická kryptografie pro šifrování a dešifrování dvojici klíčů, které jsou různé. Klíče se nesmějí ani rovnat, ani být jeden od druhého odvoditelný. Pro klíče K_1 a K_2 tedy musí platit:

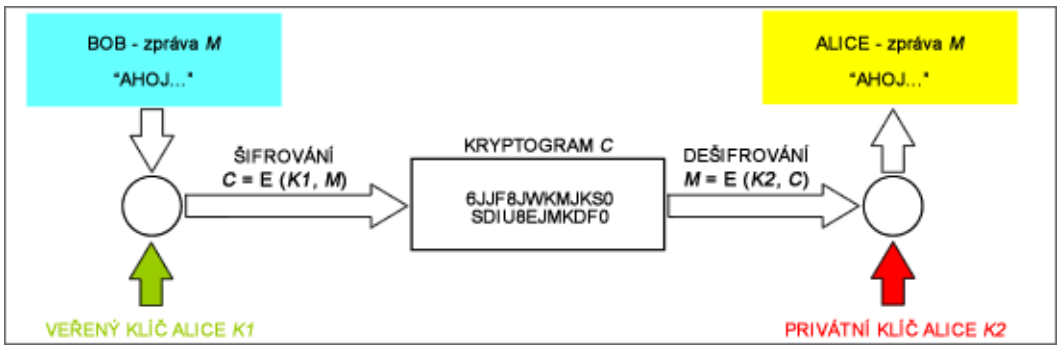
$$K_1 \neq K_2 \tag{2.21}$$

U asymetrické kryptografie odpadá složité sdílení klíčů jako u symetrické kryptografie. Klíč je zde rozdělen na 2 části K_1 a K_2 přičemž jeden je označován jako veřejný a druhý

jako privátní. K šifrování nebo dešifrování nám vždy stačí pouze jeden klíč. Proto odpadá již zmíněné sdílení klíčů.

Pokud uživatel zašifruje zprávu svým privátním klíčem, lze to považovat za ověření původu dat – tedy jejich autora.

Jak jsem zmínil výše, oba klíče jsou spolu v podstatě svázány. Důležitou podmínkou ale je, aby byl jeden klíč od druhého neodvoditelný (dostupnými prostředky). V podstatě lze říci, že asymetrické šifrovací algoritmy používají k šifrování funkce, které lze jednoduše provést pouze v jednom směru. Výpočet počáteční hodnoty z výsledku je pak téměř neproveditelný a výpočetně velice náročný [12].



Obr. 2.17 Průběh šifrování a dešifrování v asymetrické kryptografii

Použití asymetrické kryptografie

Oproti symetrické kryptografii odpadá složitá distribuce klíčů. Privátní klíč je třeba si pečlivě uschovat. Jeho druhou polovinu – veřejný klíč, je možno volně distribuovat. Na rozdíl od symetrické kryptografie se tedy hodí pro vzdálené použití. Kdy se nemusí obě strany komunikace navzájem znát.

Zpravidla lze asymetrickou kryptografii použít dvěma různými způsoby, přičemž záleží na pořadí použitých klíčů. Je důležité zmínit, že oba klíče jsou schopny jak šifrovat, tak dešifrovat. Není tedy pevně stanoveno pořadí klíčů.

Pokud data zašifruje kdokoliv veřejným klíčem určitého uživatele, má jistotu, že si je bude moci přečíst pouze daný uživatel za pomoci svého tajného soukromého klíče. Zde je tedy ověřena identita příjemce neboli čtenáře, přičemž odesílatelem může být kdokoliv a pravost dat tedy není ověřena.

Druhým způsobem je zašifrování odchozí zprávy soukromým klíčem. Tím je ověřena identita odesílatele zprávy. Tuto zprávu pak může dešifrovat příjemce (a nejen on), pomocí veřejného klíče odesílatele. V případě potřeby se dá tento způsob použití asymetrické kryptografie kombinovat s takzvanýma hashovacími funkcemi pro zajištění integrity dat, o tom se však zmíním dále [12].

Ve skutečnosti je použití asymetrické kryptografie velice náročné, co se týče výpočtů kryptogramů. Není tedy vhodné pro dlouhé komunikace s velkým objemem. Pokud bychom chtěli tento jev optimalizovat, zdá se jako nejlepší řešení použití symetrické a asymetrické kryptografie najednou. Pomocí asymetrické kryptografie dochází pouze k distribuci klíče, který je použit v symetrické kryptografii. Následná komunikace se pak šifruje symetricky a je velice rychlá. Tento způsob je v praxi velice často a úspěšně využíván [7, 11, 12].

2.6.1 Algoritmus RSA

Název tohoto algoritmu vznikl z počátečních písmen jmen jeho objevitelů (Rivest, Shamir, Adleman). Spočívá v náročnosti výpočtu rozkladu čísla na součin dvou prvočísel. Pokud je číslo dostatečně velké, je při dnešních výpočetních možnostech algoritmus bezpečný. Za minimální hranici bezpečného klíče je dnes považováno 1024 bitů, přičemž doporučená délka klíče je 2048 bitů. Tento algoritmus není vhodný pouze pro šifrování dat, ale například také pro elektronické podepisování dokumentů.

Tvorba klíčů algoritmu RSA

Jelikož jde o asymetrický algoritmus, tak se šifrovací a dešifrovací klíč neshodují. Jde tedy o klíčový pár, který je na sobě závislý, ale jeden od druhého odvodit nejde. Generování klíčového páru probíhá v celkem šesti základních krocích.

1. Uživatel si volí dvě prvočísla $\boldsymbol{p}$ a $\boldsymbol{q}$, která musí být náhodná, vysoká a dostatečně vzdálená
2. Algoritmus vypočte jejich vzájemný součin $\boldsymbol{n = p*q}$
3. Dále je vypočtena Eulerovy funkce $\boldsymbol{r = (p - 1) * (q - 1)}$
4. Zvolí se celé číslo $\boldsymbol{e}$ menší než $\boldsymbol{r}$, číslo $\boldsymbol{e}$ musí splňovat podmínku nesoudělnosti s $\boldsymbol{r}$
5. Algoritmus najde číslo $\boldsymbol{d}$ podle zákonitosti $\boldsymbol{(d*e) \bmod r \equiv 1}$
6. Veřejnými parametry jsou $\boldsymbol{e}$ (veřejný klíč) a $\boldsymbol{n}$ (modulo)

Šifrování a dešifrování algoritmem RSA

Kryptogram C je vypočten pomocí vzorce $\boldsymbol{C = Z^e \bmod n}$, k jeho dešifrování je pak nutno použít druhý klíč, v našem případě soukromý $\boldsymbol{M = C^d \bmod n}$. Pořadí klíčů je možno prohodit v závislosti na tom, co potřebujeme v rámci komunikace zaručit [4].

Bezpečnost algoritmu RSA

Základ bezpečnosti algoritmu RSA je založen na problému faktorizace velkých čísel. Tedy rozkladu na součin čísel, nebo prvočísel. Pokud bude číslo n menší než 256 bitů, lze jej faktorizovat i na osobním počítači během několika málo hodin. Na n o velikosti 512 bitů bychom potřebovali několik set osobních počítačů. Teoreticky je v dnešní době možno dešifrovat heslo o velikosti n 1024 bitů, prakticky to však prozatím provedeno nebylo. Za bezpečnou délku hesla se aktuálně považuje délka hesla 2048 bitů.

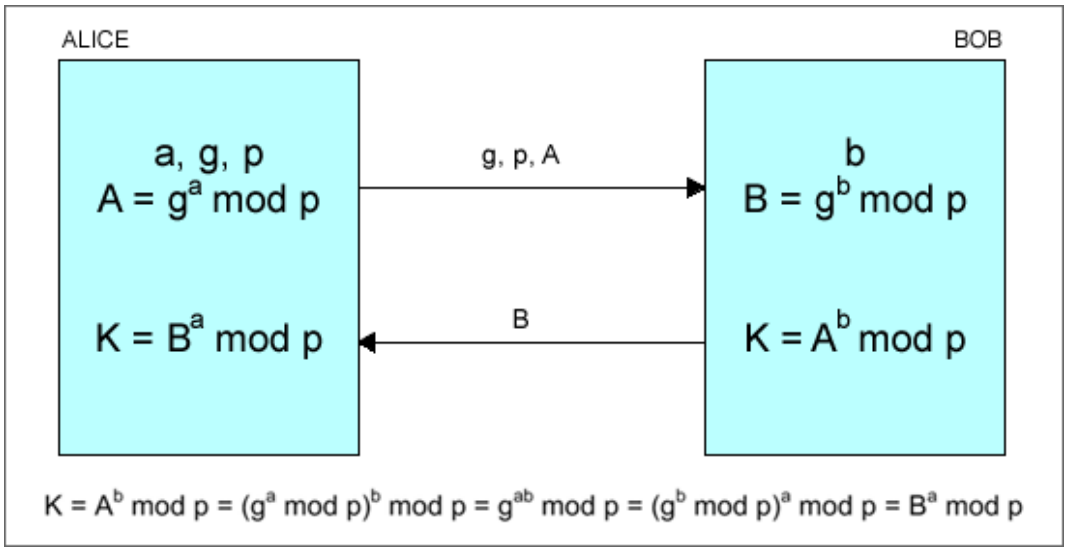
2.6.2 Algoritmus Diffie-Hellman

Tento algoritmus se nepoužívá pro přímé šifrování dat, nýbrž jako protokol pro bezpečný přenos klíče pro symetrické šifrování. Algoritmus je založen na matematické složitosti výpočtu diskrétního logaritmu. Odpadá tak víceméně složitý přenos klíče na velké vzdálenosti bez rizika [4].

Tvorba klíčů obou stran v algoritmu Diffie-Hellman probíhá následujícím způsobem:

1. Alice náhodně zvolí čísla ***a***, ***g***, ***p***
2. Alice vypočte číslo $\boldsymbol{A = g^a \bmod p}$ a zašle Bobovi čísla ***g***, ***p***, ***A***
3. Bob náhodně vybere číslo ***b*** a obdobně vypočte $\boldsymbol{B = g^b \bmod p}$
4. Bob pomocí A vypočte klíč $\boldsymbol{K = A^b \bmod p}$ a zašle Alici číslo ***B***
5. Alice na základě ***B*** vypočte klíč $\boldsymbol{K = B^a \bmod p}$

Výsledné klíče *K* musí být shodné a jsou společným klíčem obou stran, použitelným pro symetrickou kryptografii. Čísla *a*, *b* jsou soukromými klíči Alice a Boba a v otevřené formě se nesmí dostat k potencionálnímu útočníkovi. Graficky je průběh sestavení klíče popsán na obrázku 2.18.



Obr. 2.18 Sestavení klíčů algoritmem Diffie-Hellman

Bezpečnost algoritmu Diffie-Hellman

Případný útočník, který je schopen odposlouchávat komunikaci mezi oběma stranami není schopen ze zjištěných informací zjistit heslo žádné z obou stran. Nevýhodou tohoto algoritmu je bezbrannost proti útoku „man-in-the-middle“, protože algoritmus neumožňuje vzájemnou autentizaci obou komunikujících stran. Případný útočník je tedy schopen se, při odchycení komunikace, vydávat za jednu z komunikujících stran a podvrhnout tak její identitu. Sestavený klíč je poté útočníkovi znám [1].

2.6.3 Algoritmus El Gamal

Tento algoritmus je podobný algoritmu Diffie-Hellman a jeho bezpečnost je také založena na složitosti výpočtu diskrétního logaritmu. Algoritmus El Gamal tvoří základní stavební prvek pro digitální podepisovací algoritmy, jako je například algoritmus DSA, o kterých se zmíním později. Na rozdíl od algoritmu Diffie-Hellman, který je použitelný pouze pro bezpečné sestavení klíče je El Gamal vhodný také pro šifrování a dešifrování zpráv. Sestavení klíče probíhá v následujících třech krocích:

1. Uživatel si zvolí prvočíslo $\boldsymbol{p}$ a dvě náhodná čísla $\boldsymbol{g}$ a $\boldsymbol{x}$, která musí být menší než $\boldsymbol{p}$
2. Uživatel následně vypočte hodnotu $\boldsymbol{y = g^x \bmod p}$
3. Čísla $\boldsymbol{y}$, $\boldsymbol{g}$, $\boldsymbol{p}$ jsou veřejným klíčem, číslo $\boldsymbol{x}$ je soukromým klíčem uživatele

Z výše uvedeného zápisu je vidět nutnost výpočtu diskrétního logaritmu pro odhalení soukromého klíče. Abychom mohli pomocí získaného klíče zašifrovat zprávu, musíme provést následující kroky:

1. Uživatel zvolí číslo $\boldsymbol{k}$, které splňuje podmínku $\boldsymbol{1 \leq k \leq p - 2}$
2. Zašifrovaný text je rozdělen mezi dvě proměnné, nejčastěji $\boldsymbol{a}$, $\boldsymbol{b}$. Ty se vypočtou podle vztahů $\boldsymbol{a = g^k \bmod p}$, $\boldsymbol{b = y^k M \bmod p}$

Výsledná šifrovaná zpráva má tedy dvojnásobnou délku zprávy původní. Pro dešifrování je nutné udělat pouze jeden krok:

1. Adresát vypočte zprávu $\boldsymbol{M}$ podle vztahu $\boldsymbol{M = b / a^x \bmod p}$

Výhodou algoritmu El Gamal je, že volba koeficientů p, g, x, k, ovlivní výsledný kryptogram. To znamená, že když pro stejnou vstupní zprávu zvolíme jiné parametry, bude výsledná šifra odlišná [17].

2.6.4 Kryptografie eliptickými křivkami

Problematika eliptických křivek v kryptografii je aktuální již téměř 30 let. Kryptografie nad eliptickými křivkami (ECC) se využívá zejména pro výhodu použití kratšího klíče než v klasické kryptografii. Kratší klíče se v praxi odrážejí na rychlejších výpočtech kryptogramů a úsporách paměti. Agentura NSA doporučuje využívat ECC ve veškeré vládní komunikaci.

ECC je alternativou ke kryptografickým algoritmům, jako jsou RSA nebo DSA. Velmi dobře se dají využít také při sestavování klíčů algoritmem Diffie-Hellman. Vzhledem k potřebě kratších klíčů lze ECC provozovat na slabším hardwaru. Proto se hodí zejména pro implementaci do čipových karet a podobných zařízení. Budoucnost této metody je tedy velice slibná. Momentálně se EC tolik nevyužívají vzhledem k tomu, že před jejich uvedením do praxe vzniklo mnoho algoritmů, které se již dokázaly uplatnit a uživatelé nejsou přesvědčeni o nutnosti přechodu [26].

Za eliptickou křivku můžeme prohlásit skupinu bodů, která splňuje Weierstrassovu rovnici:

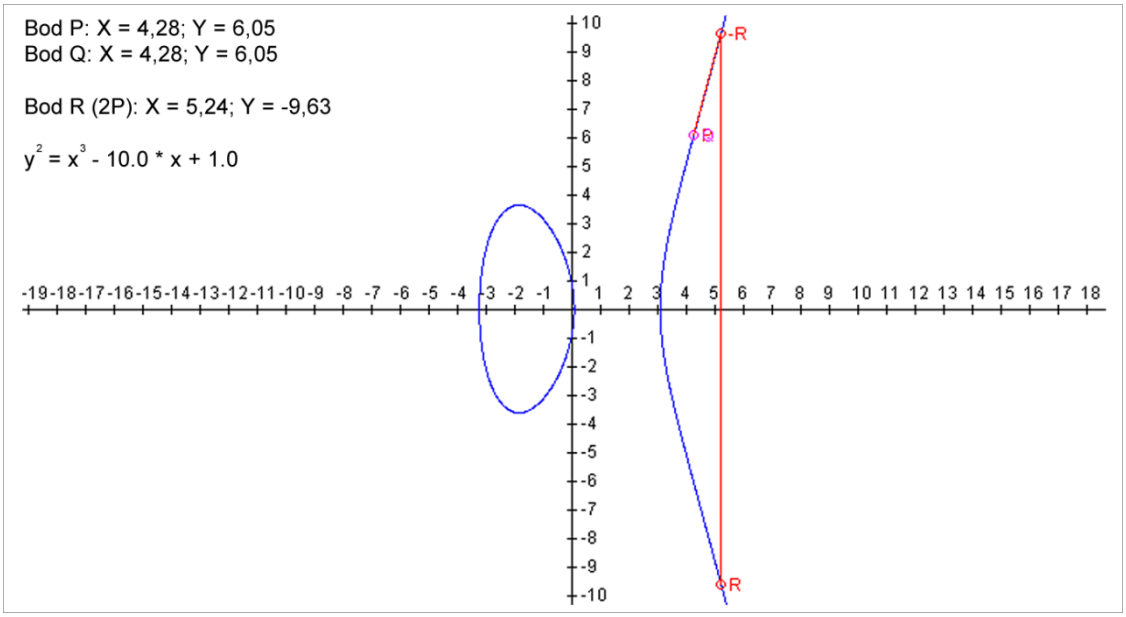
$$y^2 = x^3 + ax + b \tag{2.22}$$

Princip šifrování pomocí ECC

Máme-li eliptickou křivku, můžeme na ní zvolit jakýkoliv bod *P*. Dále zvolíme libovolné přirozené číslo, a vypočteme jeho součin s bodem *P*. Tím dostaneme obraz dalšího bodu na eliptické křivce – bodu *Q*. Pokud bychom však ze známých bodů *P* a *Q* chtěli zpětně zjistit zvolené přirozené číslo, bude to velice výpočetně náročné a v reálném čase s aktuálními prostředky nemožné [26].

Průběh šifrování pomocí ECC

1. Na začátku komunikace je nutné, aby si obě komunikující strany domluvily bod ***P*** a zároveň také příslušnou eliptickou křivku.
2. Obě komunikující strany si zvolí svoje přirozené číslo ***SK***, které je jejich soukromým klíčem.
3. Následně obě strany pomocí svého soukromého klíče vypočítají veřejný klíč ***VK=n*P***
4. Výsledkem je rovnost, která nám potvrzuje vzájemné ustanovení klíčů
   $\boldsymbol{M = SK_1 * VK_2 = SK_1 * SK_2 * P = SK_2 * VK_1 = M}$



Obr. 2.19 Grafická interpretace principu součtu dvou bodů v rovině

2.7 Hashovací funkce

Jde o matematickou funkci, která slouží pro ověření integrity dat. Hashovací funkce dokáže převést velké množství vstupních dat na poměrně malé výstupní číslo. Výstupem hashovací funkce je tzv. otisk.
Pro porovnávání dat je důležité, aby hashovací funkce dokázala vytvořit z jakýchkoliv vstupních dat stejně dlouhý otisk.

Základním nedostatkem hashování je, že počet vstupních dat mnohonásobně převyšuje počet vytvořitelných otisků. Proto neznamená, že pokud se rovnají výstupní otisky, pak se rovnají také vstupní data.
Snahou nejrůznějších hashovacích funkcí je minimalizace této náhodné shody otisků na minimum.

Kvalitní hashovací funkce musí splňovat 3 základní podmínky k tomu, aby byla bezpečná. Tyto podmínky jsou považovány za splněné, pokud jsou splněny po dostatečně dlouhou dobu – jejich nabourání je tedy velmi výpočetně a časově náročné. [8, 12]

Jednosměrnost
Podmínka toho, že ze vstupního řetězce dat lze vypočíst jednoznačný otisk, avšak z otisku nelze zpětně vypočíst původní data.

Slabá bezkoliznost
Není možné ke vstupním datům a jejich otisku vypočíst jiná data se stejným otiskem.

Silná bezkoliznost
Není možné najít dva různé texty se stejným otiskem [8, 12].

2.7.1 Algoritmus MD5

Jeden z neznámějších a nejpoužívanějších hashovacích alegoritmů. Byl vynalezen v roce 1991 jako nástupce již zastaralé MD4. Základní charakteristika tohoto hashovacího algoritmu je hash o konstantní délce 128 bitů. Výhodou je, že malá změna v původních datech vyvolá zdánlivě velkou a viditelnou změnu v hashi. V roce 2004 byly objeveny závažné bezpečnosti v hashovacím algoritmu MD5, a proto se od něj začalo postupně upouštět. Základní úlohou algoritmu MD5 je hashování hesel z důvodu, aby nebyla na disku počítače uložena v čistém textu ale pouze jako hash. Stejně tak probíhá i ověřování přístupového hesla – porovnávají se pouze hashe.

Text: Toto je pokusny text pro hashovaci funkci
MD5 hash: b50b5926ed9ffac4d244cbb39e0e6ebd

Text: toto je pokusny text pro hashovaci funkci
MD5 hash: c0d03208fba65a0d4ef48d566e2263ff

Z uvedeného příkladu je opravdu poznat, že i malá změna ve vstupním stringu vyvolá velkou změnu na výstupním hashi [13].

Bezpečnost algoritmu

V devadesátých letech přišly první nástiny možností, že hashování MD5 není bezpečné, vzhledem k chybě v návrhu. Tato teorie se potvrdila až téměř o deset let později, když byly objeveny závažné chyby v návrhu a od funkce MD5 se začalo upouštět. Zejména pokud byla hashovací funke použita k bezpečnému ukládání hesel [13].

2.7.2 Algoritmy rodiny SHA

Hashovací algoritmy SHA jsou nástupci hashovacího algoritmu MD5. Prvním zástupcem tohoto algoritmu byl SHA-0, který však byl po krátké době stažen americkou agenturou pro bezpečnost NSA, byly na něm provedeny dosud nezveřejněné změny a byl vrácen do oběhu pod názvem SHA-1. Tento hashovací algoritmus vytváří ze vstupních dat hash o délce 160 bitů. Postupem času a zvyšováním bezpečnostních nároků byl tento algoritmus nahrazen nástupcem SHA-2, který zahrnuje alegoritmy SHA-224, SHA-256, SHA-384 a SHA-512. Čísla zde určují délku hashe v bitech. Přesto že algoritmus dnes není považován za zcela bezpečný, je použit například v SSL nebo SSH. Může být použit také jako algoritmus nahrazující CRC a zvyšující bezpečnost, nebo jako hashovací funkce pro ukládání hesel [13].

Text: Toto je pokusny text pro hashovaci funkci
SHA-1 hash: 8542e59f91c2c221c1ba98fe2f8155ed24567d25

Text: toto je pokusny text pro hashovaci funkci
SHA-1 hash: 23d9b2a7c0031f6a96ab060b1b002121ce2786d3

Bezpečnost algoritmu

Vzhledem k momentální rychlosti rozvoje výpočetní techniky byl vyhlášen konkurz na nástupce tohoto hashovacího algoritmu, tedy SHA-3. Tento způsob se již ověřil při hledání algoritmu AES popsaného výše. Jako rok uvedení nového algoritmu SHA-3 je plánován rok 2012. Aktuálně již není používání hashovacího algoritmu SHA-1 považováno za bezpečné a je doporučeno používat jeho nástupce z rodiny SHA-2 [13].

2.7.3 Algoritmus HMAC

Tento algoritmus slouží pro ověření autenticity zprávy. Tvoří základ pro ověření elektronického podpisu. Nejde tedy čistě o hashovaní algoritmus jako takový, ale o standardní postup, jakým zajistit, aby byly všechny neoprávněné změny provedené v přijaté zprávě snadno detekovány. Postup pro tvorbu HMAC hashe je shrnut do tří základních kroků [13, 17]:

- ke zprávě, kterou chceme ověřit, přidáme náhodná data, které nezná nikdo jiný
- z kombinace zprávy a náhodných dat je vypočten hash, pomocí téměř jakékoliv dostupné hashovaní funkce (např. MD5, SHA, atd.)
- druhé straně je tento hash zaslán společně s textem původní zprávy

Ověření autenticity zprávy pak probíhá následovně:

- k přijaté zprávě je opět přidán náhodný kód
- z kombinace zprávy a náhodného obsahu je vypočten hash pomocí stejné funkce
- pokud tento hash souhlasí s hashem přijatým se zprávou, je zpráva platná

2.7.4 Algoritmus CRC

Tato hashovací funkce slouží pro kontrolu integrity dat. Zkratka CRC po překladu znamená „cyklický redundantní součet“. Používá se pro kontrolu dat při jejich zasílání elektronickou cestou. Před odesláním je CRC spočten a připojen k odesílaným datům, příjemce po přijetí dat spočte opět CRC a porovná jej s původním. Pokud jsou hodnoty stejné, pak nedošlo k narušení dat. Tato metoda je využívána například při komprimaci souborů, například pomocí algoritmů RAR a ZIP. Slouží k zajištění správnosti a bezztrátovosti komprese, která mohla vzniknout například chybou softwaru nebo hardwaru. Tyto komprimační algoritmy jsou v některých případech schopny, na základě hodnot CRC, poškozený soubor opravit. Tato hashovací funkce není vhodná pro ochranu integrity dat před případným útočníkem [13].

Výpočet cyklického redundantního součtu

Mechanikou výpočtu je převod posloupnosti dat na polynomy a jejich následný součet s náhodně zvoleným kontrolním heslem převedeným na polynomy. Tento součet polynomů odpovídá XOR operaci mezi binárními vstupy. Pokud je zvolené heslo dobré, tak i malá změna ve vstupní posloupnosti vytvoří velkou změnu na výstupu

100101 převedeme na $x^5 + x^2 + 1$ vstupní data
110011 převedeme na $x^5 + x^4 + x + 1$ heslo

100101 XOR 110011 = 010110
$(x^5 + x^2 + 1) + (x^5 + x^4 + x + 1) = x^4 + x^2 + x$

$010110 = x^4 + x^2 + x$

Text: Toto je pokusny text pro hashovaci funkci
CRC32: 4f3190ce (hex), 41 data bytes

Bezpečnost algoritmu CRC

Pro správnou funkci algoritmu je nutno správně zvolit klíč tak, aby malé změny na vstupu vyvolaly velké změny na výstupu. Stupeň řídícího polynomu (hesla) se nejčastěji uvádí za zkratkou CRC, například CRC16. Jelikož však existuje několik algoritmů pro

výpočet kontrolního součtu, nestačí nám na ověření integrity souboru znát pouze řídící polynom. Jde o velice oblíbenou funkci používanou v hojném množství, vzhledem k její jednoduchosti, rychlosti a přehlednosti [13].

2.8 Digitální podepisování

Algoritmy digitálního podpisu slouží k ověření identifikace podepisujícího subjektu a kontrolu integrity podepisované zprávy. Jsou tedy analogické ke klasickému ručnímu podpisu například papírové smlouvy, faktury a podobně. Algoritmy pro digitální podpis musí zaručovat následující podmínky [17]:

- **nefalšovatelnost** – podpis by neměl být napodobitelný, případně zfalšovaný podpis by měl být jednoznačně rozpoznatelný
- **autentizace** – podpis je jednoznačně přidělen k jeho vlastníkovi
- **nepřenosnost** – podpis je nepřenosný jak mezi uživateli, tak mezi dokumenty
- **integrita dat** – podepsaný dokument není po podpisu možno změnit, pokud ke změně dojde, je podpis neplatný
- **nepopiratelnost** – pokud uživatel podepíše dokument svým podpisem a podpis je ověřen, nemůže uživatel popřít podepsání dokumentu

Standardní podepisování může probíhat různými způsoby.

- Prvním způsobem je, že podpis je připojen externě k podepisovanému textu, který je nešifrovaný. Text zprávy si tak může kdokoliv přečíst a porozumět mu.
- Druhým způsobem je, že podpis je integrován do souboru s podepisovaným textem. Ten je opět nešifrován a lze volně číst.
- Třetím a nejbezpečnější způsobem je, že podpis je připojen do souboru s podepisovaným textem, který je však zašifrován. Text tedy nelze volně číst bez znalosti klíče.

2.8.1 Algoritmus DSA

Jde o algoritmus, který je používán pro digitální podepisování dat. Jeho první verze byla vyvinuta v roce 1991 a byla používána výhradně pro americkou vládu. Pro veřejnost byl algoritmus uvolněn v roce 1993. Jeho bezpečnost spočívá podobně jako u algoritmu Diffie-Hellman na matematické složitosti výpočtu diskrétního logaritmu [4].

Tvorba podpisového páru klíčů

1. V aktuální verzi algoritmu DSA se používá hashovací funkce **SHA-2**
2. Zvolí se délka parametrů ***L*** a ***N***, které určují délku klíčů. Aktuálně je doporučeno používat délky (**1024**,**160**), (**2048**,**224**), (**2048**,**256**) a (**3072**,**256**)
3. Zvolíme číslo ***q***, jehož velikost je dána číslem ***N*** zvoleném v předchozím kroku. Délka ***N*** musí být minimálně tak velká, jako je délka výstupu použité hashovací funkce

4. Volba prvočísla p o velikosti L bitů. Prvočíslo musí splňovat podmínku, že $p-1$ je násobek q
5. Stanovení g tak, že jeho násobky modulo p jsou rovny q
6. Výběr čísla x v rozsahu $0 < x < q$
7. Výpočet $y = g^x \bmod p$
8. Veřejný klíč je (p, q, g, y), soukromý klíč je x

Podepisování
Průběh podepisování začíná výběrem náhodné proměnné k, která musí být v rozsahu $0 < k < q$. Následně je vypočten $r = (g^k \bmod p) \bmod q$ a $s = (k^{-1}(H(z) + x * r)) \bmod q$.

Podmínkou je, že ani r ani s se nesmí rovnat nule, pokud tento případ nastane, je nutno celý podpis opakovat. Pokud jsou obě hodnoty větší než nula, tvoří podpis dvojice (r, s) [4].

Ověření podepsaného dokumentu
Na začátku ověřování podpisu je nutno ověřit dvě základní podmínky $0 < r < q$ a $0 < s < q$. Pokud tyto podmínky nejsou splněny, je podpis zamítnut a neprovádí se již žádné další kontroly [4].

Pro ověření podpisu je zapotřebí vypočítat následující 3 parametry $w = (s)^{-1} \bmod q$, $u_1 = (H(z) * w) \bmod q$ a $u_2 = (r * w) \bmod q$. Finální ověření probíhá výpočtem $v = ((g^{u1} * y^{u2}) \bmod p) \bmod q$. Pokud se obě strany rovnice rovnají, je dokument považován za podepsaný. Pokud se strany rovnice nerovnají, podpis není platný.

Bezpečnost algoritmu
Tento široce využívaný podepisovací algoritmus je považován za bezpečný. Od té roku 1993, kdy byl uvolněn pro veřejnost je využíván i v OpenSSL a openSSH, tedy volně šiřitelných verzích protokolů SSL a SSH [4].

2.8.2 Algoritmus ECDSA
Jde o podepisovací algoritmus fungující na základě použití eliptických křivek. Algoritmus je principem velice podobný algoritmu DSA a jeho bezpečnost se opírá o výpočet diskrétního logaritmu. Výhodou oproti algoritmu DSA je menší bezpečná velikost klíče a rychlost zpracování [17].

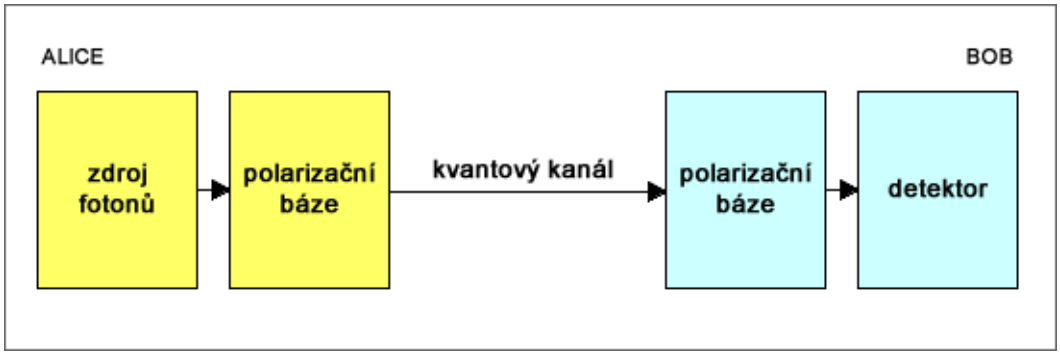
2.9 Kvantová kryptografie
Kvantová kryptografie je poměrně mladý obor, přestože první zmínky a návrhy algoritmů byly navrženy v roce 1984 fyziky Bennetem a Brassardem. Jejich algoritmus se jmenuje BB84 a je založen na principech kvantové mechaniky. Hlavním rozdílem oproti standardním algoritmům založených na matematických zákonech je podloženost fyzikálními zákony.

Hlavním fyzikálním poznatkem, díky němuž může kvantová kryptografie existovat, je existence fotonu. Ten jen elementární částicí záření a zároveň nejdůležitějším prvkem kvantové kryptografie. Všechny protokoly, které jsou ve kvantové kryptografii používány, pracují na principu polarizace fotonů. Ty mohou mít dvě roviny, které jsou na sebe navzájem kolmé. Polarizace fotonu je vždy vztažena k dané bázi. Pokud chceme polarizaci fotonu měřit na straně příjemce, musíme naprosto stejně zvolit i polarizační bázi. Pokud bychom polarizační bázi zvolili jinou, nejsme v podstatě schopni říci, které z naměřených fotonů jsou natočeny správně a které špatně [24].

Při použití matematiky jsme odkázání na výpočetní rychlost dnešních počítačů. Pokud výkon vzroste, algoritmus založený na matematických operacích je oslaben. U fyzikálních zákonů to tak není, ty jsou neměnné a jasně dané. Teoreticky tak jejich bezpečnost nezávisí na výkonu výpočetních strojů a v časové linii by měly být stále stejně bezpečné.

Bezpečnostní výhodu tvoří především ta vlastnost fotonu, že nelze měřit jednu jeho vlastnost, aniž bychom tím ovlivnili vlastnost jinou. Foton nelze duplikovat, aniž by byly změněny vlastnosti zdrojového fotonu. Jakékoliv odposlouchávání komunikace je tedy snadno odhalitelné.

Kvantová kryptografie se nepoužívá k šifrování dat, které chceme přenést, ale pouze pro bezpečnou tvorbu a distribuci klíče. Teorie je taková, že pokud jsme schopni bezpečně vytvořit a distribuovat náhodný klíč o požadované velikosti, jsme schopni pak data zašifrovat pomocí Vernamovy šifry a poslat je adresátovi nezabezpečenou cestou [13].



Obr. 2.20 Přenosová soustava v kvantové kryptografii

V kvantové kryptografii je přenosová soustava klíčů mírně odlišná například od jednoduchého sestavení klíčů protokolem Diffie-Hellman. Celá soustava se skládá z pěti základních částí, z nichž 2 jsou na straně odesílatele, 2 na straně příjemce a jednu část tvoří přenosový kanál mezi nimi. Celá soustava je názorně zobrazena na obrázku 2.20.

2.9.1 Protokol BB84

Tento čtyřstavový protokol nejčastěji využívá lineární a diagonální polarizaci fotonů. Pro komunikaci lze využít jednu nebo dvě polarizační báze. Fotony tedy lze polarizovat buď lineárně, nebo diagonálně vzhledem k základním osám x, y.

Pro základní pochopení lze uvést komunikaci mezi Alicí a Bobem, kteří používají pouze jednu polarizační bázi. Předem se domluví, která polarizace bude odpovídat logické hodnotě 1 a která logické hodnotě 0. Alice pak přes polarizační bázi vysílá generovanou posloupnost a Bob ji detekuje pomocí své polarizační báze a detektorů. Tento typ komunikace se nedoporučuje, protože pokud by útočník, který zachytává fotony v kvantovém kanálu, uhodl typ báze, ani jedna z komunikujících stran nemá možnost útočníka odhalit.

Z tohoto důvodu se používají polarizační báze 2 a tím pádem má každý foton 4 polarizace. Je na domluvě obou stran, které dvě polarizace přiřadí které logické hodnotě. Jednotlivé kroky komunikace jsou uvedeny v tabulce 2.9 a očíslovány v následujícím popisu:

Alice vygeneruje náhodnou posloupnost bitů, která by měla být klíčem (krok 1). Tato posloupnost by měla být dvojnásobné délky oproti požadované délce klíče, protože 50% fotonů bude podle pravděpodobnosti detekováno špatnou bází. Obě strany náhodně volí polarizační báze (krok 2 a 4). Alice následně pomocí polarizačních bází posílá Bobovi fotony (krok 3), který je pomocí svých polarizačních bází přijímá (krok 5) a následně detekuje přijatou posloupnost bitů (krok 6). Po komunikaci si může Alice s Bobem vyměnit nezabezpečené informace o použitých bázích a fotony, na kterých se báze neshodují, jsou zahozeny (krok 7 a 8). To statisticky nastane v 50% případů.

Odhalení útočníka je v tomto případě velice jednoduché. I ten totiž musí náhodně volit polarizační báze. Pokud se trefí do stejné báze jako Alice, dostane Bob foton nezměněný, pokud ne, dostane Bob foton změněný. Alice si s Bobem tedy vymění konkrétní informace o některých bitech (krok 9). Pokud informace na obou stranách souhlasí, komunikace s vysokou pravděpodobností nebyla odposlouchávána (krok 10). Ostatní bity, které nebyly obětovány tvoří klíč (krok 11) [24].

Tab. 2.9 Průběh sestavení klíče a odhalení útočníka v protokolu BB84 [24]

fáze	krok	průběh															
Fáze 1 sestavení	1	0	0	1	0	1	0	1	0	0	0	1	0	0	1	1	1
	2	+	×	×	+	×	+	+	×	+	×	×	×	+	×	+	+
	3	↔	↗	↖	↔	↖	↔	↕	↗	↔	↗	↖	↗	↔	↖	↕	↕
	4	×	×	+	+	×	+	×	+	×	×	+	×	+	+	+	×
	5	↖	↗	↕	↔	↖	↔	↗	↔	↖	↗	↔	↗	↔	↕	↕	↗
	6	1	0	1	0	1	0	0	0	1	0	0	0	0	1	1	0
Fáze 2 domluva	7	×	✓	×	✓	✓	✓	×	×	×	✓	×	✓	✓	×	✓	×
	8		0		0	1	0				0		0	0		1	
Fáze 3 odhalení	9		0				0										
	10		✓				✓										
	11				0	1					0		0	0		1	

Pokud některé z polarizací nesouhlasí, komunikace byla odposlouchávána a celý proces je nutno z důvodu bezpečnosti zopakovat. Pravděpodobnost zjištění útočníka se zvyšuje s počtem obětovaných fotonů (bitů klíče). Podle následujícího vzorce, kde *n* je počet obětovaných bitů klíče, lze tuto pravděpodobnost *P* vypočítat [24]:

$$P = 1-(^{3}/_{4})^{n} \tag{2.23}$$

Bezpečnost protokolu
Bezpečnostní překážku tvoří fakt, že Alice neví, jestli komunikuje právě s Bobem, nebo útočníkem, který se za něj vydává. To lze však vyřešit pomocí jiných autentizačních protokolů.

2.9.2 Protokol B92
Tento protokol je velice podobný původnímu BB84. Jediným a podstatným rozdílem v jeho fungování je to, že používá pouze 2 polarizační stavy fotonu. Jejich polarizační osy jsou od sebe navíc odchýleny pouze 45°. To snižuje celkovou pravděpodobnost zachycení správné polarizace pomocí dvou různých polarizačních bází na polovinu. Je to způsobeno tím, že pokud detekujeme daný foton špatnou bází, dostaneme hned dva výsledky, z nichž není správný ani jeden a tato hodnota tedy není akceptovaná. Bob tedy na jeden polarizovaný foton od Alice může na své straně získat 3 různé výsledky, z nichž pouze jeden je správně.

Pro správnou detekci fotonu v protokolu B92 musí Bob použít přesně opačnou polarizační bázi pro daný foton než Alice.

Ostatní průběh celé komunikace je identický s protokolem BB84. Rozdílný je pouze vzorec, podle kterého počítáme pravděpodobnost *P*, se kterou jsme schopni detekovat útočníka v závislosti na počtu obětovaných bitů *n*. Pravděpodobnost se vypočte podle vzorce [24]:

$$P = 1-(^{7}/_{8})^{n} \tag{2.24}$$

2.9.3 Šestistavový protokol
Protokol je opět postaven na základě protokolu BB84. Oproti němu však, dle svého názvu, nabízí dvě polarizace fotonů navíc. Jde o polarizaci kruhovou, která může být jak vlevo, tak vpravo. Komunikace principielně probíhá opět stejně. Rozdíl je v menší pravděpodobnosti, že Alice i Bob vyberou stejnou bázi a tím snižuje i účinnost algoritmu. Tento problém však zároveň zvyšuje bezpečnost, protože stejný problém s výběrem polarizační báze má i případný útočník.

Schopnost detekce útočníka v šestistavovém protokolu lze opět vypočítat podle jednoduchého vzorce, kde *P* značí pravděpodobnost odhalení útočníka a *n* značí počet obětovaných bitů [24]:

$$P = 1-(^2/_3)^n \qquad (2.25)$$

Z výše uvedeného vzorce můžeme vidět, že ze všech tří protokolů vycházejících z původního BB84 má šestistavový protokol nejlepší poměr obětovaných bitů k dobrému procentu odhalení útočníka. Jinak řečeno nám stačí na odhalení útočníka obětovat méně bitů, než je tomu v případě protokolů BB84 a B92. [24]

2.9.4 Protokol EPR (E91)

Jde o třístavový protokol, který již není svým základem postaven na původním BB84. Hlavním rozdílem od předešlých protokolů je počet fotonů, které jsou použity pro přenos jednoho bitu. U protokolu EPR jsou pro přenos bitu použity dva fotony v rámci jedné částice, které jsou vždy opačně polarizované.

V rámci protokolu EPR probíhá komunikace odlišně oproti protokolům rodiny BB84. První krok komunikace dělá třetí strana, která zašle pár fotonů, po jednom každé straně. Alice si zvolí svou polarizační bázi a přijme svůj foton. Dle provázanosti tohoto páru fotonů víme, že pokud Bob zvolí stejnou bázi, naměří hodnotu přesně opačnou než Alice.

Bob následně volí hodnotu své báze a následná komunikace již probíhá velmi podobně jako v předešlých algoritmech. Rozdíl je pouze v detekování útočníka. K jeho detekci se nepoužívají platné bity, ale bity, které jsou zamítnuté z důvodu rozdílných bází. Pokud se útočník bude snažit do systému nabourat, bude to detekováno právě díky tomu, že komunikace probíhá po dvojicích fotonů a jakýmkoliv narušením jednoho z nich dojde i k narušení rovnovážného stavu mezi nimi [24].

2.10 Útoky na kryptografické algoritmy

Hlavním problémem všech kryptografických algoritmů, založených na matematických zákonech, je stále rostoucí výpočetní výkon a cena dnešních běžně dostupných počítačů. Pokud tedy mluvíme o útocích hrubou silou, tak algoritmy, které jsou dnes bezpečné a běžně se v komunikaci používají, již za několik let bezpečné být nemusí.
I v dnešní době je možno sestrojit stroje, které luští klíče běžně používaných kryptografických algoritmů téměř v reálném čase. Tyto stroje jsou však velmi drahé a nejsou tedy dostupné pro běžného uživatele.

Základním typem útoku hned po útoku hrubou silou je klasická kryptoanalýza šifrovaného textu, průběhu komunikace atd. Pokud se útočníkovi povede zachytit dostatečné množství šifrovaných zpráv, může se mu podle nich povést odhalit použitý

algoritmus. Pak už chybí jen krůček k tomu, aby na daný algoritmus provedl nějaký příslušný útok, nebo ze zachycených zpráv vypočetl klíč.

Dalším klasickým typovým útokem je tzv. „man-in-the-middle“. Tento způsob napadení spočívá v tom, že se útočník vydává za jednu z komunikujících stran a je schopen nejen číst tajnou komunikaci, ale také zprávy měnit a posílat dále skutečnému adresátovi. Tento útok je možno provést, pokud nejsme schopni kvalitně zaručit, že na druhé straně komunikace se opravdu nachází ten, kdo tam být má. Je čistě na uživatelích, jakou implementaci algoritmů zvolí a jestli ji chtějí zabezpečit i proti takovémuto typu útoku.

Existuje mnoho typů útoků, které lze na kryptografické algoritmy aplikovat. Některé z nich využívají interních chyb v algoritmech a pokouší se nabourat do jejich matematických výpočtů. Takovými můžou být různé analytické programy, které řeší v krátkých dobách výpočty diskrétních logaritmů, nebo faktorizaci vysokých čísel. Tedy dva základní stavební kameny bezpečnosti dnešní kryptografie.

Kromě interních útoků jsou popsány také útoky, které se nezabývají vnitřní stavbou algoritmů a vnitřními výpočty, ale zajímají se o to, jaká data jdou vyčíst zvenku algoritmu. Útočníka v takovém případě nemusí zajímat pouze zašifrovaná zpráva, kterou si mezi sebou uživatelé posílají, ale zaměřují se především na to, jaké informace z algoritmu unikají neplánovaně. Příkladem tohoto typu útoky mohou být například tzv. „útoky postranními kanály“. Ty se dělí do několika skupin podle toho, na kterou oblast úniku informací z algoritmu se zaměřují [27]:

- **časová analýza** – útok zaměřující se na dobu vykonávání šifrování. Pomocí této doby lze určit šifrovací algoritmus

- **jednoduchá odběrová analýza** – útočník sleduje proudový odběr například čtečky čipových karet. Ze získaných proudových špiček při čtení z karty a provádění kryptografických operací, může útočník odhalit například typ použitého algoritmu.

- **diferenciální odběrová analýza** – útočník sleduje odběr zařízení a testuje jej při provádění několika tisíc operací s náhodnými daty. Statisticky pak může odhadnout pravou podobu klíče.

- **elektromagnetická analýza** – každé elektronické zařízení vydává el.-mg. záření při svém provozu. Útočník se jej snaží zjistit pomocí např. cívky a dále jej analyzovat za účelem zjištění algoritmu, nebo klíče.

- **zavádění chyb** – útočník může vyvoláním dočasných extrémních podmínek způsobit změny v algoritmech, nebo v jejich hardwarových implementacích a

tím se do nich nabourat. Jde zejména o změnu napájecích napětí, jejich frekvence, teploty, ozařování zařízení apod.

Ve výše uvedeném článku jsem zmínil časté typy útoků. Ve skutečnosti je jich však mnohem více. Téměř každým dnem vznikají nové a nové metody, jak se do kryptografických algoritmů nabourat.

Pokud se na celou věc podíváme z nadhledu, tak je v dnešní době k dispozici dostatek prostředků na to, aby uživatelé mohli svá data uchovávat a přenášet bezpečně. I přes to většina komunikace probíhá nezabezpečenou cestou a dle známého tvrzení je celý systém tak bezpečný, jako jeho nejslabší článek. Je tedy jen na uživatelích, jakým rizikům chtějí sebe a svoje data vystavovat.

3. Frontend portálu

Plánovaný portál o kryptografii by měl umožňovat široké veřejnosti co nejsnadnější přístup k informacím týkajícím se této problematiky. Veškerá teorie, která bude v rámci portálu zpracována, bude vysvětlena od největších základů tak, aby byla pochopitelná i laiky v této oblasti.

3.1 Webdesign

Webdesign je soubor komplexních činností, které zajišťují tvorbu webových prezentací. Webdesign zahrnuje celou tvorbu, od grafického návrhu, přes tvorbu kaskádových stylů, až po programování webu jako takového. Ten je nejčastěji tvořen jazykem HTML. Doménou posledních let jsou takzvané dynamické stránky, které jsou psány pomocí jazyku PHP, který je založen na bázi jazyka C. Jelikož jazyk PHP sám o sobě je příliš úzce vyhrazen právě pro psaní webů, neumožňuje používat programování typické pro standardní počítačové aplikace. Proto je možné do webů vkládat skripty například v jazyce Java, který je multiplatformní.

Hlavním úkolem webdesignu při tvorbě webové prezentace je zajištění toho, aby byla prezentace snadno dostupná své hlavní cílové skupině zákazníků, aby byla jednoduchá a přehledná, ale aby v ní nechyběly žádné zásadní informace.

Zdálo by se, že při tvorbě výukového portálu není třeba tolik hledět na jeho optimalizaci. Je to však naopak. Pokud má portál sloužit široké veřejnosti k pochopení problematiky kryptografie, je téměř nutností, aby byl snadno vyhledatelný a byl na internetu vidět. Větší návštěvnost totiž přiláká i odbornou veřejnost, která přispívá ke zkvalitnění obsahu webu. Právě přilákání odborné veřejnosti, která by přispívala vlastními odbornými články a zkušenostmi je cílem tohoto portálu. Z historického hlediska je ověřeno, že kryptografie je právě jedním z oborů, které jsou závislé na uživatelské interakci. Ta je jedinečným testerem jakýchkoliv např. kryptografických algoritmů – viz. AES.

3.1.1 Optimalizace pro prohlížeče

S problémy se webdesign setkává téměř na každém kroku. V dobách dávno minulých, kdy trh obsazoval z 99% internetový prohlížeč Internet Explorer od firmy Microsoft, stačilo, aby byl návrh optimalizován právě pouze pro tento prohlížeč. Webdesignér pak měl jistotu, že se stránka zobrazí korektně všem uživatelům. Dnes je problém v tom, že na trhu je prohlížečů více a webové prezentace je nutno optimalizovat tak, aby se korektně zobrazily ve všech prohlížečích. To by standardně měla zajistit validita stránky, bohužel tomu tak však není. Problémy se zobrazování jsou způsobeny vlastními jádry jednotlivých prohlížečů. Každý prohlížeč totiž používá pro vykreslování jiné metody. Je tedy jen na webdesignerovi, jak si s danou situací poradí. Často je však nutné volit mnoho kompromisů.

3.1.2 Optimalizace pro vyhledávače (SEO)

Filozofie přístupu SEO se zabývá optimalizací webových stránek pro vyhledávací roboty. V dnešní době jsou informace na webu indexovány a katalogizovány pomocí tzv. vyhledávacích serverů. Mezi nejznámější patří například Google, Bing nebo Yahoo.

Z technického hlediska není možné, aby se pokaždé při zadání hledacího dotazu do formuláře vyhledávače prohledal celý internet. Proto vyhledávače obsah internetu indexují a ukládají si do databáze pouze klíčová slova, zvýrazněné nadpisy a obsahy článků. Pro tuto indexaci využívají takzvané roboty. To jsou ve skutečnosti skripty, které procházejí internetové stránky a podle předem určených pravidel provádějí ukládání výše zmíněného obsahu.

Zde se dostává ke slovu právě SEO. Indexační robot je totiž „naučen“ hledat určité věci na určitých místech v dané stránce. Pokud na předem řečených pozicích nic nenajde, stránku zavře a pokračuje dál. SEO je právě soubor pravidel, podle kterých robot stránku indexuje. Platí, že čím jednodušší prostředí indexačnímu robotovi připravíme, tím vyšší hodnocení naše stránka ve vyhledávači dostane [9].

Optimalizace pro vyhledávače zahrnuje jak etické, tak také neetické způsoby, jak zlepšit svoji pozici ve vyhledávačích. Mezi etické patří tyto metody:

- Kvalitní a unikátní obsah webové prezentace
- Validita stránek (HTML kód bez chyb a se všemi náležitostmi)
- Používání CSS stylů v externích souborech
- Používání nadpisů <H> ke zdůraznění důležitých částí webu
- Krátká webová adresa neobsahující „nečitelný kód“ – použití modu rewrite
- Uvedení a zobrazení mapy webu
- Vyplnění autora, jazyka, titulku a klíčových slov charakterizujících prezentaci
- Usnadnění indexace pro roboty pomocí příkazů v souboru robots.txt
- Tvorba zpětných odkazů na naše stránky z jiných stránek (například výměna odkazů). Množství těchto odkazů charakterizuje jeden z nejznámějších ukazatelů – PageRank, ten tvoří roboti vyhledávače Google a je aktualizován cca. 1x za tři měsíce [9]

Za každou z těchto podmínek přidělují roboti webové stránce body, za každou chybějící část jsou body strženy. Výsledek ze všech prohledaných stránek tvoří žebříček ve vyhledávači. Je tedy v zájmu každého webdesignera tyto zásady dodržovat a snažit se je plnit [9].

Kromě výše uvedených etických možností zviditelnění webu existují bohužel také metody neetické, které zneužívají umělou inteligenci indexačních robotů. S postupem času a vývojem robotů jsou však tyto metody eliminovány na minimum. V případě

odhalení jakékoliv neetické metody použité na propagaci webové stránky, je tato stránka ve vyhledávači degradována. Mezi nejznámější neetické metody patří:

- Metoda „spamování“ spočívá v rozesílání odkazů na stránky pomocí mailu, nebo ukládání odkazů na stránky do volně přístupných diskuzí na internetu a do blogů. O toto ukládání se starají předem naprogramovaní roboti
- Klamání indexačních robotů nepravým obsahem. Robotovi je předložena stránka s jiným obsahem, než vidí uživatelé internetu
- Skrývání textů (zejména v tagu <H>) tak, že jej normální uživatel internetu nevidí. Vidí jej pouze indexační robot a je tím oklamán [9].

3.2 Webová adresa

Aby byl portál dobře přístupný všeobecné veřejnosti, je třeba, aby byl dobře vyhledatelný pomocí základních vyhledávacích služeb, jako jsou například Google a Bing. Základem této dostupnosti je webová adresa, která je srozumitelná v co nejvíce jazycích tak, aby jednoznačně říkala, co je obsahem internetové prezentace. Pro portál s obsahem o kryptografii, šifrách, jejich slabinách a podobně bych navrhoval adresu, která bude obsah portálu dostatečně charakterizovat.

3.3 Podoba portálu

Portál by měl být uživatelsky příjemný jak na pohled, tak na ovládání. Důraz je kladen na moderní a technický design, který bude plně odpovídat obsahu webu. Web bude mít 2 základní části. Jedna bude viditelná pro nezaregistrované návštěvníky a druhá část se zpřístupní pouze po registraci, která bude zdarma. Registrace bude sloužit pouze k oddělení pasivních návštěvníků a aktivních uživatelů, kteří budou chtít přispívat svými články k růstu webu.

V případě, že bude web zpřístupněn na veřejné webové adrese, bude nutnost, aby se o jeho obsah starali takzvaní „vydavatelé“. Ti mají za úkol číst a zveřejňovat články, kterými přispívají ostatní registrovaní uživatelé. Nekontrolují jen spisovnost a stylistiku, ale jsou zodpovědní také za to, že článek bude umístěn v tematicky příhodné kategorii na webu.

Web by měl mít pro veřejnost následující strukturu:

- O webu
- Aktuality
- CryptoWiki
- Výukové skripty
- Výukové animace
- Ke stažení

Pro redaktory by dále bylo umožněno:

- Přidávání, publikace a zveřejňování článků
- Tvorba a přidávání appletů
- Tvorba a přidávání animací

3.4 Grafická podoba portálu

Logo

Jde v podstatě o zjednodušenou grafickou prezentaci celého portálu. Od loga se následně odvíjí řada prvků přítomných na webové stránce, včetně jejích barev. Návrh a styl loga je proto velice důležitý a je třeba mu věnovat velkou pozornost. Logo musí být výmluvné, chytlavé a velice snadno zapamatovatelné.

Obr. 3.1 Návrh loga portálu.

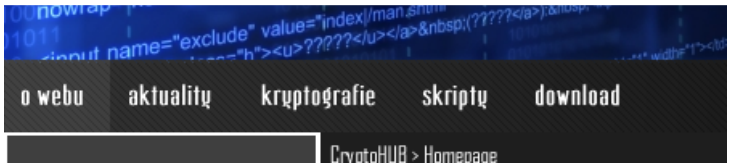
Hlavička

V hlavičce celého portálu bude umístěno právě již zmíněné logo. Vedle loga bude možnost zobrazit část obsahu webu, doplňkové menu nebo například nejaktuálnější článek. Toto místo bývá ve velké části webů zneužíváno pro umístění nejrůznějších reklam nebo odkazů do speciálních částí webového portálu, na které chceme uživatele upozornit. Tento postup bych však nedoporučoval, protože právě na většině těchto webů působí špatný dojem ihned při vstupu na stránku.

Obr. 3.2 Návrh hlavičky portálu.

Menu

Pod hlavičkou portálu bude umístěno vodorovné menu s velkými úvodními tlačítky. Těmi bude portál rozdělen na několik základních částí podle popisu výše. Menu bude zvoleno s ohledem na potřebnou rozšiřitelnost a přehlednost. Nejlepší je, co se týče přehlednosti a dostupnosti pro uživatele internetu, dvouúrovňové menu.

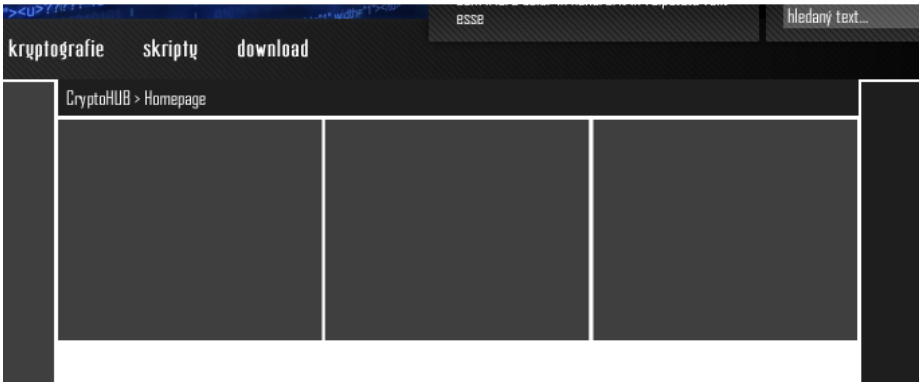


Obr. 3.3 Detail návrhu menu.

Oblast prezentace novinek

Vzhledem k zaměření portálu není vhodné pronajímat na něm reklamní plochy komerčním společnostem. To však neznamená, že by tyto strategické pozice měly zůstat nevyužity. Právě pod menu je ideální a nejlukrativnější místo, které je vhodné pro zdůraznění čehokoliv, co se týká problematiky kryptografie. Může jít o nejaktuálnější článek, pozvánku na konferenci, nejnovější objev v oblasti kryptografie, nebo upozornění na nově objevenou závažnou chybu v nějakém bezpečnostním algoritmu.

Pokud umístíme obsah do tohoto místa, můžeme si být jisti tím, že si jej (nebo alespoň jeho část) přečte 99% návštěvníků portálu – tedy cílová skupina pro důležitá sdělení.



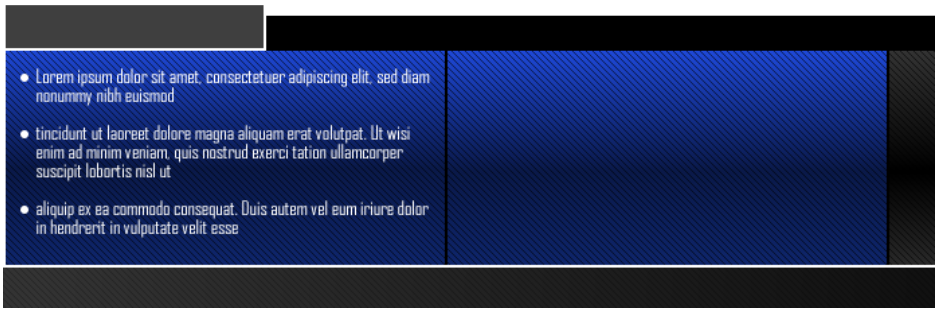
Obr. 3.4 Detail pozic pro umístění reklamy.

Obsah

Obsah portálu by mohl být rozdělen až do 3 sloupců. Prostřední sloupec by byl hlavním nositelem obsahu – tedy článků, obrázků, skriptů atd. Boční sloupce by mohly být naplněny doplňkovými informacemi, jako například seznamy nejčtenějších článků, anketami, doplňkovým menu atp. Nastavení hlavního obsahu by bylo volitelné pro každou sekci portálu zvlášť tak, aby jeho nastavení odpovídalo vždy koncepci dané sekce.

Oblast externího obsahu

CMS Joomla! obsahuje integrovanou čtečku rss kanálů. Je tedy možnost do portálu načítat hlavičky článků (ale i celé články) z partnerských webů, se kterými by byla dohodnuta spolupráce. Je možno načítat i články v cizích jazycích.



Obr. 3.5 Návrh místa pro umístění externího obsahu.

Stálé odkazy

Tato oblast webu již není tak strategicky výhodná a je navštěvována pouze uživateli, kteří vědí, co zde hledají – tedy konkrétní odkaz. Proto se do těchto míst umisťují odkazy na technickou podporu, faq, lokální wiki atd. Mohl by zde být navíc umístěn odkaz na rejstřík základních pojmů v kryptografii. Oblast je také dobře využitelná na zobrazení odkazů na weby s podobnou tématikou – možnost vzájemné výměny odkazů.

Obr. 3.6 Návrh místa pro umístění stálých odkazů.

Patička webu

V patičce bývají standardně umístěny informace o právech k zobrazenému obsahu, o autorovi a odkaz do administrace webu. Vzhledem k tomu, že obsah webu bude autorský, tedy nepůjde o články druhých stran, bude v patičce zapsán jak copyrigt ©, tak informace o výhradních právech.

Obr. 3.7 Návrh patičky webu a autorských informací.

Při grafickém návrhu webu jsem přihlížel k celkové koncepci. Důraz je kladen na přehlednost, originalitu a modernost. Jelikož kryptografie je dnes svým způsobem hlavně počítačová problematika, podřídil jsem tomuto směru i grafický návrh. Jde o prvotní návrh, který se dá samozřejmě upravovat podle toho, jak se bude web vyvíjet a jaké budou požadavky, co se týče přehlednosti. Jeho layout by však měl být zachován tak, aby nebylo nutné měnit CSS šablonu.

3.5 Šablona portálu

K tomu, abychom byli schopni v portálu adresovat data a umisťovat je na platné pozice v návrhu, je třeba vytvořit šablonu, která bude obsahovat pozice jednotlivých částí webu popsaných výše. K přípravě šablony pro CMS Joomla! se používá CSS kaskádové styly a XML.

CSS

Základním úkolem kaskádových stylů je oddělení obsahu webu od jeho grafické podoby. Jejich použití nejen zpřehledňuje zdrojový kód pro případné úpravy, ale také jejich úpravu usnadňuje. Například pro změnu barvy a velikosti všech nadpisů na webu nemusíme tento krok opakovat pro všechny html značky, ve kterých jsou nadpisy umístěny, ale stačí nám upravit pouze styl, který odpovídá všem nadpisům. Je tomu tedy podobně jako například ve Wordu. Výhodou je, že pokud připravíme kaskádové styly pro jednu stránku webu, můžeme je pak použít na stránce jiné, což nám velmi usnadňuje práci. Další výhodou je cachování souboru s kaskádovými styly v prohlížeči. Ten je načten pouze jednou pro celý web, což velmi urychluje načítání designu jednotlivých stránek celého webu.

XML

Jde o značkovací jazyk, který bývá při designovém návrhu webových stránek použit na popis struktury a obsahu jednotlivých stránek portálu. Na rozdíl od kaskádových stylů tedy nepopisuje vzhled a designovou stránku, ale věcnou stránku – tedy co a kde bude umístěno a jak to bude pojmenováno pro případnou adresaci obsahu.

3.6 Členění obsahu

Hlavní sílou portálu by měla být schopnost návštěvníkům demonstrovat základní kryptografické techniky. Není to však jediný obsah, který bude portál poskytovat. Jeho obsah bude rozčleněn do několika sekcí, z nich každá bude zaměřena na něco jiného. Dalo by se říci, že základní dělení bude na teorii a praktické ukázky.

3.6.1 Teoretická sekce

V teorii budou postupně popsány všechny metody kryptografie. Budou rozděleny podle typu a obsah popisů jednotlivých metod bude samozřejmě možno měnit a rozvíjet. Součástí teorie bude také popis historického vývoje kryptografie tak, jak je tomu v teoretické části této práce. Doplněny budou také znalosti nutné pro pochopení kryptografických algoritmů, jako jsou matematické základy výpočtu kryptogramů, ale také důvody, proč jsou algoritmy bezpečné. Teoretická část této práce by tak měla tvořit svým členěním a obsahem jakousi obsahovou kostru celého portálu.

3.6.2 Praktická sekce

V sekci praktických ukázek bude mít uživatel možnost seznámit se se základními principy kryptografie. Nepůjde pouze o ukázky, ale veškeré napsané skripty budou vyžadovat uživatelovu interakci. Uživatel si tak bude moci vyzkoušet například zašifrování a dešifrování svého vlastního kryptogramu. Získá tím přehled nejen o způsobu šifrování a jeho průběhu, ale také si ověří časovou náročnost jednotlivých operací.

4. Administrace portálu

V dnešní době se rozmáhá takový přístup k tvorbě internetových portálů, že jeho obsah je vlastně plněn samotnými uživateli (v tomto případě jejich odbornou částí). Na začátku je vytvořen základ portálu včetně rozvržení, designu a implementace dostupných appletů. Portál je nadále plněn aktuálním obsahem pomocí jeho autorů a redaktorů. Tak dokáže držet krok s aktuálními technologiemi a děním v oblasti kryptografie.

V rámci úspory času, který bude potřeba na plnění portálu kvalitními daty a na programování demonstračních appletů, jsem zvolil použití již hotového CMS systému, který je ověřen uživateli, je šířen pod licencí GNU/GPL a mám s ním vlastní zkušenosti.

4.1 CMS systém Joomla

Joomla je webový CMS systém, neboli systém pro správu obsahu webu. Její hlavní výhodou je otevřenost systému a neustálý vývoj reagující na nové trendy v internetu.

Obr. 4.1 Oficiální logo CMS Joomla! [10]

Stálá aktualizace je důležitá také z pohledu bezpečnosti systému. Joomla je napsána v jazyce PHP a podporuje databáze MySQL.

Kromě toho, že je Joomla systémem pro správu obsahu, umožňuje její stavba také jednoduché rozšíření o nejrůznější komponenty a moduly. Těch oficiálně existuje kolem 3500 a většinou jsou také volně šiřitelné. Umožňují tak Joomlu kompletně předělat podle potřeb uživatele.

Základní kompilace Joomly umožňuje základní věci jako:

- Správa uživatelů a přidělování práv
- Správa článků, blogů z frontendu internetové prezentace
- Indexaci a vyhledávání v prezentaci
- Tvorba anket
- Tiskové a pdf náhledy článků
- Přizpůsobení vzhledu, tvorba vlastních šablon

Jde pouze o zevrubný pohled na celý CMS. Kromě možnosti editace obsahu z frontendu prezentace, obsahuje Joomla také velice dobře propracované administrační rozhraní. Do něj se mohou přihlásit všichni, kteří mají administrační práva k danému portálu – není tedy volně přístupná pro registrované uživatele [10].

4.1.1 Správa uživatelů a přidělování práv

Základní prostředí pro přidělování práv a registraci uživatelů v CMS Joomla umožňuje pouze základní správu uživatelů s předem přednastavenými poli pro vyplňování při registraci. Tento přístup není příliš ideální, pokud chceme o uživateli získat více dat, než jen přihlašovací jméno, heslo a emailovou adresu. Základní rozhraní také neumožňuje přidávat volitelné uživatelské úrovně přístupu. Systém je však v tomto ohledu vstřícný a nabízí hned 4 administrační úrovně pro správu webu z frontendu stránky a 3 úrovně administrátorské, které mají přístup také do administrační části webu.
Základní uživatelská oprávnění jsou následující, přičemž platí, že vyšší úroveň oprávnění dědí oprávnění nižší úrovně:

- **registrovaný uživatel** – umožňuje pouze správu vlastního profilu a údajů
- **autor** – umožňuje přidávat články, neumožňuje jejich zveřejňování
- **editor** – navíc umožňuje korektury jiným autorům
- **publisher** – zveřejňuje články, třídí je do kategorií, hlavní dozor nad obsahem webu

- **manažer** – základní správcovská práva, tvorba struktury webu
- **správce** – plný přístup ke všem volbám, nemůže editovat pouze jiné správce
- **supersprávce** – komplexní dozor nad webem, umožňuje tvořit a mazat správce

Protože základní komponenta umožňuje pouze výše uvedené volby, rozhodl jsem se pro portál využít instalovatelnou komponentu, která tyto volby rozšiřuje o velké množství dalších prvků a voleb. Zvolil jsem komponentu Comunity Builder, která je také šířena pod licencí GNU/GPL [10].

Comunity Builder

Tato komponenta umožňuje kromě standardních procedur ověřování uživatelských účtů pomocí tzv. aktivačních emailů. To je důležité zejména jako ochrana proti spamovacími roboty. Druhým důvodem pro aktivaci účtu je také potvrzení jeho majitele. Na portálu bude totiž moct zaregistrovaný uživatel přispívat vlastními články. Z tohoto důvodu je důležité, aby v souvislosti s článkem byla uvedena platná emailová adresa – jak z hlediska autorství, tak případného porušení autorství třetí straně.

Comunity Builder umožňuje přesně specifikovat položky profilu. Administrátor také může nastavit, které položky budou vyžadovány při registraci a které jsou pouze doplňkové. Zároveň lze také nastavit všeobecné podmínky, které musí uživatel při registraci odsouhlasit.

Pomocí komponenty lze také přidávat další úrovně oprávnění, kromě těch, které jsou primárně nastaveny a nejdou měnit, tak můžeme přidávat i uživatelská práva specifická pro daný web. To se může hodit, pokud například chceme uživatele rozdělit podle skupin v rámci jedné úrovně oprávnění.

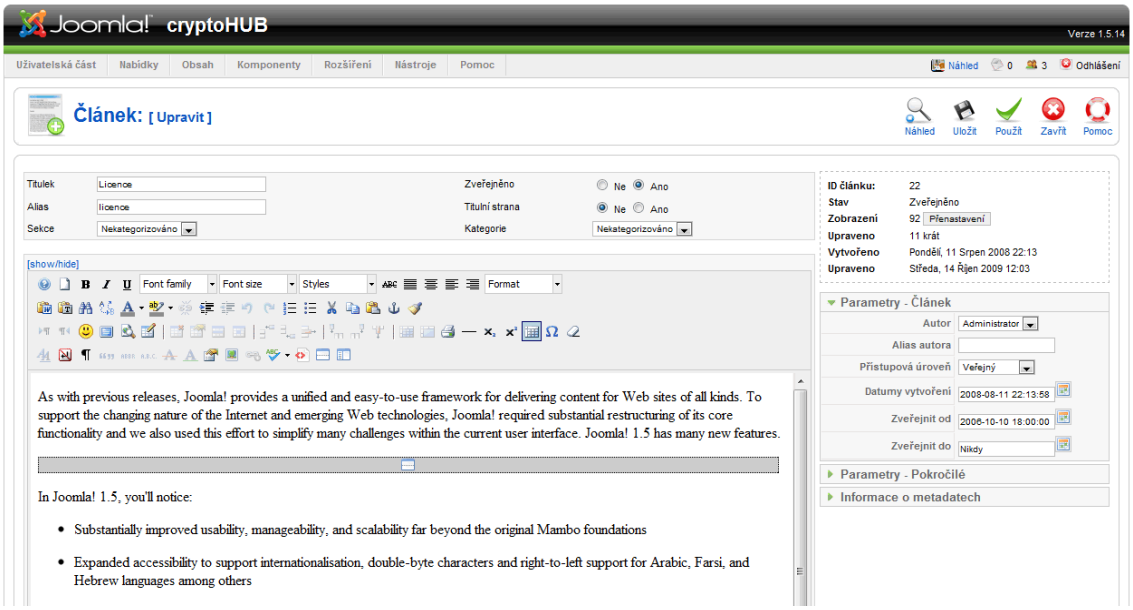
Poslední z důležitých funkcí je synchronizace uživatelských účtů s původním enginem Joomly. To nám umožňuje nejen přejít na Comunity Builder i v průběhu tvorby webu, ale také využití výhodných vlastností obou komponent dohromady. Tou je například rozesílání hromadných zpráv všem uživatelům webu, nebo vybraným skupinám.

4.1.2 Správa článků a blogů

Základní filozofie dělení článků pomocí redakčního systému Joomla spočívá v dělení do sekcí a kategorií. Každá sekce může obsahovat několik kategorií, které umožňují třídit články do příslušných skupin. Ty pak lze pomocí odkazů zobrazovat přesně podle jejich rozřazení. Lze tedy zobrazit jak celou sekci, tak i vybranou kategorie, nebo dokonce i samotný článek.

Při psaní článku můžeme také zvolit, zda chceme článek zobrazit na úvodní straně, která je tak jakousi speciální nadřazenou sekcí. Článek je tak zobrazen jak na hlavní stránce, tak v místě, kam patří svým zařazením. Každý článek nese meta značku svého autora a datum vytvoření. Ke každému článku je také umožněno vložit klíčová slova [10].

Psaní článku je v Joomle vyřešeno pomoví tzv. wysiwyg editoru. Ten umožňuje editaci textu podobným způsobem, jako je tomu například v textovém editoru Word od firmy Microsoft. Základní komponenta Joomly umožňuje pouze základní editaci, jako je volba fontu, barvy textu a zarovnání. Pro naše potřeby jsou třeba i složitější funkce, jako je například vkládání obrázků do článků, popřípadě vložení externího zdrojového kódu. Z tohoto důvodu jsem ke standardnímu editoru doinstaloval editor třetí strany JCE (Joomla Content Editor), který je opět šířen pod licencí GNU/GPL.



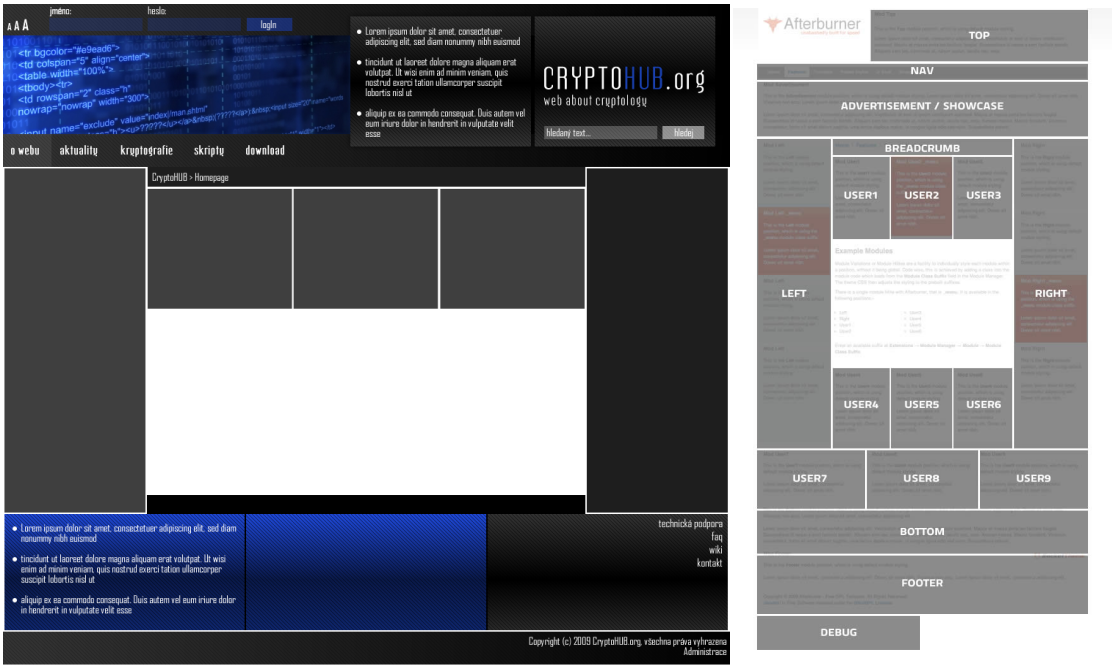
Obr. 4.2 WYSIWYG editor článků [10]

4.1.3 Tvorba anket

Správa anket je v CMS Joomla velice jednoduchá. Základní komponenta zakomponovaná v systému obsahuje dostatečné možnosti pro zbudování jakékoliv jednoduché ankety. Pomocí tohoto systému bohužel nejdou budovat složitější dotazníky. Tato funkce však není pro naši potřebu nutná. Pokud bychom přece jenom některou z pokročilejších funkcí chtěli aplikovat, museli bychom opět použít některou z komponent třetích stran. V této oblasti je však bohužel většina kvalitních komponent placených [10].

4.1.4 Přizpůsobení vzhledu

Joomla umožňuje aplikovat šablony na přizpůsobení vzhledu frontendu webové stránky. Pomocí administračního panelu můžeme do systému nainstalovat hned několik šablon naráz a následně mezi nimi přepínat. Volba stylu je tak velice variabilní a během několika vteřin jsme schopní změnit vzhled celého webu. Na internetu je volně dostupné velké množství předpřipravených šablon, které jsou téměř okamžitě po nainstalování připraveny k použití. Další možností je stáhnout si layout šablony, tedy jakýsi základ s rozmístěnými prvky pro zobrazování a následně si šablonu dotvořit, jak graficky, tak programově. Tuto cestu jsem zvolil i já pro portál Cryptohub [10].



Obr. 4.3 Kompletní návrh portálu s rozmístěním prvků

Po portál jsem zvolil jednu z nejrychlejších šablon, které se na internetu vyskytují. Důvodem je nejen její rychlost, ale také jednoduchost a mnoho pozicovatelných polí rozmístěných po celém webu. Rozmístění polí v šabloně zhruba odpovídá také rozmístění prvků v grafickém návrhu.

V CMS Joomla se pozicují grafické a funkční prvky zvlášť. Na Obr. 4.3 vlevo můžeme vidět grafický návrh webu, na který je třeba napsat css šablonu. Na obrázku vpravo pak vidíme popis adresovatelných bloků z původní šablony.

4.1.5 Externí skripty

Joomla v základním nastavení nepodporuje vkládání uživatelských kódů v jiném jazyce než html. Uživatel může vkládat svůj html kód do jakéhokoliv článku a překladač jej uloží, jako by použil vstupní wysiwyg editor. Výstupem z tohoto editoru je totiž také html kód [10].

Abychom mohli do článků vkládat php skripty, java skripty, nebo flash videa, je potřeba doinstalovat do Joomly příslušná rozšíření. Ty vesměs fungují na stejném principu, jako když chceme do článku vložit vlastní html kód – tedy podle postupu uvedeného výše. Jediný rozdíl je v uvození a zakončení vloženého skriptu speciální značkou, která není zaměnitelná s jakoukoliv značkou jazyka html.

V mém případě je použita následující syntaxe pro vložení appletu do webové stránky:

```
{source}
<APPLET CODE="SafePassword1.class" width="540" height="280"
border="1"></APPLET>
{/source}
```

Zde právě úvodní značka *source* ve složených závorkách uvozuje kód, který je předán komponentě, která jej umí obsloužit (přeložit). Podobným způsobem můžeme do článků vkládat jak php skripty, například pomocí syntaxe *include*, tak videa flash, pomocí syntaxe *object*.

4.1.6 RSS kanály

Jde o formát informací, který se používá na rychlé a jednoduché sdílení novinek a článků pomocí tzv. kanálů. Jde o zjednodušený obsah článku zapsaný v XML formátu, který je sdílen na webu, společně s ostatními metadaty, jako je jméno autora a ostatní iniciály zdroje. Pokud webová stránka podporuje poskytování RSS zpráv, může se kdokoliv přihlásit k jejich odběru. To lze realizovat buď pomocí RSS čtečky zabudované například v internetovém prohlížeči, nebo může být odběr realizován přímo pomocí jiného webového rozhraní.

CMS Joomla ve výchozím nastavení podporuje tvorbu vlastního RSS kanálu, takže se kdokoliv, nejen zaregistrovaný uživatel, bude moci přihlásit k jejich odběru. Tento kanál může být použit i jinými weby zabývajícími se tématem kryptografie na čerpání obsahu.

Pro čerpání novinek opačným směrem, tedy publikace článků z ostatních webů o kryptografii, je nutno do Joomly doinstalovat komponentu třetí strany, která tuto funkci

čtení bude zajišťovat. Moderní komponenty si poradí nejen se čtením, ale přímo také se zobrazováním odkazů na dané články. Ty buď odkazují na web autora, nebo jsou celé články přeformátovány do designu klientského webu společně s uvedením autora a zdroje. Pro novinky z ostatních webů je v designu vyhrazeno hned několik míst, která jdou variabilně plnit [10].

4.1.7 Správa sdíleného obsahu

CMS systém Joomla umožňuje veškerý textový obsah webu exportovat do formátu pdf nebo odesílat rovnou na tiskárnu. Kromě toho bude na portálu uloženo mnoho obrázků, animací a skriptů s tématikou kryptografie. Ten bude navíc v souladu s autorskými právy k jednotlivým prvkům nabídnut registrovaným uživatelům ke stažení. Tuto službu však standardní instalace systému Joomla nenabízí, proto jsem po zanalyzování potřeb webu přistoupil k instalaci externí komponenty PhocaDownload [10].

PhocaDownload

Jde o velice jednoduchou a přizpůsobivou českou komponentu, která umožňuje právě nahrávání a sdílení souborů. Jediná nevýhoda této komponenty je, že soubory může nahrávat a sdílet pouze registrovaný uživatel, který svými práv patří do skupiny správců webu. Kromě jiných dostupných komponent má však tato velkou výhodu v možnosti třízení souborů do sekcí a kategorií a ty pak zpřístupňovat pouze jednotlivým skupinám uživatelů. Další nespornou výhodou a v tomto případě nutnou je možnost volby licenční smlouvy. Každý uživatel tak bude muset při stahování sdílených souborů souhlasit s licenčními podmínkami stanovenými k danému souboru. Vzhledem k tomu, že se daná práce po odevzdání stane majetkem VUT, bude nutno právě toto v licenční smlouvě ošetřit.

4.1.8 Webové fórum

V dnešní době se bez něj neobejde žádný větší web. Většina začínajících webů začíná pouze na návštěvní knize a teprve podle jejího využívání a zvyšování nároků na ni se případně zavádí diskusní fórum. Nejinak tomu je i v případě portálu Cryptohub. Nainstalování návštěvní knihy místo fóra jsem zvolil i kvůli její menší paměťové a diskové náročnosti. Webová fóra jsou často velice rozsáhlá a náročná již po instalaci a Joomle toto platí dvojnásob. Dalším důvodem pro instalaci návštěvní knihy místo fóra je nedostupnost kvalitních fór právě pro CMS Joomla [10].

Většina výrobců se snaží napodobit téměř jediného ideálního zástupce na poli diskuzních fór, kterým je phpBB. V poslední době se však situace obrací k lepšímu a začínají se objevovat komponenty, které tyto dva systémy umí propojit. Do budoucna bych tedy doporučoval v případě nutnosti instalace diskuzního fóra použít právě phpBB a komponentu pro Joomlu, která tyto dva systémy propojí. Věřím, že se v nejbližší době tyto komponenty dostanou ze svých vývojových betaverzí a budou k dispozici plně funkční.

PhocaGuestbook
Tato komponenta nabízí jednoduchou tvorbu a zprávu návštěvních knih. Umožňuje vytvořit více knih a umístit je na různé části webu, popřípadě pro různé uživatele. Tato komponenta neumožňuje žádná speciální nastavení. Pro potřeby začínajícího webu však jistě dostačuje.

4.1.9 Multijazyčnost
Vzhledem k tomu, že většina materiálů ke kryptografii je k dispozici v anglickém jazyce, je také větší komunita anglicky mluvících uživatelů. Také pokud by portál měl sloužit čistě jenom škole, tak tu také navštěvují zahraniční studenti. Z těchto důvodů bych doporučoval do portálu doinstalovat multijazyčnou podporu pro webový obsah. Existuje mnoho komponent, které tuto službu méně či více kvalitně zajišťují. Já osobně mám nejlepší zkušenosti s komponentou Joom!Fish [10].

Joom!Fish
Tato komponenta jednoduše a nenásilně implementuje multijazyčnost s počtem jazyků dle potřeby. Komponenta funguje na principu duplikování obsahu. To znamená, že výchozí obsahy článků jsou přiřazeny k jazyku, který zvolíme jako výchozí. K ostatním jazykům a všem článkům jsou vytvořeny duplikáty, do kterých umístíme texty v příslušných jazycích. Není nutno vytvářet překlady pro všechny články, pokud chceme mít například pouze část webu vícejazyčnou. Komponenta má navíc hlídací systém, který nás upozorní na změny provedené v primárním článku a nutnost aktualizace článků v cizím jazyce. Vzhledem k tomu, že si komponenta dělá duplikáty hlavních článků, není problém tuto komponentu v případě potřeby odinstalovat.

4.2 Správa databází
Většina základních uživatelů se nedostane při správě webu do takových potíží, že by museli umět spravovat databázový server po svůj web. S rostoucí velikosti webu a tím i jeho databází je však třeba, se o databáze starat. Právě databáze a její „uklizenost“ určuje rychlost webu, která samozřejmě zajímá hlavně koncové uživatele.

Webový systém Joomla je známý tím, že nepatří zrovna k nejrychlejším i s čistou instalací. Je tedy na správci, jestli tento stav dokáže udržet. Špatný stav databází v naprosté většině způsobuje testování komponent a jejich odinstalování ze systému. Odinstalátor většinou odstraní základní komponenty systému z disku, dočasné soubory a databáze však ponechává pro případ znovuinstalace dané komponenty.

Existuje mnoho nástrojů, které umožní pročištění databází automaticky. S těmito nástroji však nemám nejlepší zkušenosti. Proto je nejlepším řešením ruční pročištění celé databáze pomocí systému PhpMyAdmin.

4.3 Správa obsahu webového disku

Webový diskový prostor je místo, kam se ukládají veškeré systémové části webu. Prostor je v naprosté většině případu navázán na doménové jméno. Platí zde podobná pravidla jako pro úklid databází. Zde je však situace mnohem jednodušší, protože orientace v diskovém prostoru je všem jistě bližší, nežli orientace v databázovém systému. Z hlediska bezpečnosti je třeba dávat pozor na oprávnění k jednotlivým složkám a zejména konfiguračním souborům systému. Tyto práva se nedoporučují měnit. Pokud jsme k tomu však donuceni okolnostmi a musíme například editovat konfigurační soubory, je nutné nezapomenout vrátit jim po ukončení editace původní přístupová práva.

Bezpečnostním problémem může v tomto případě být nezabezpečený přístup k webovému prostoru. Naprostá většina hostingů totiž poskytuje přístupy pouze přes naprosto nezabezpečené FTP spojení, které může kdokoliv odposlouchávat. Může se tak velice snadno dostat nejen k obsahu webového prostoru, ale také k obsahům konfiguračních souborů a tím pádem i k heslům do databáze, což už znamená velké bezpečnostní riziko. Pro přístup k webovému prostoru je tedy doporučeno využívat přímo osobní návštěvu u fyzického serveru, nebo použití některého ze zabezpečených přenosů dat.

4.4 Aktualizace a zálohování CMS systému

Jde o velice důležitou proceduru, kterou je třeba pravidelně provádět. Každý rozsáhlejší systém obsahuje bezpečnostní mezery. CMS Joomla není výjimkou, je tomu spíše naopak. Je to způsobeno zejména instalováním komponent od ne zcela důvěryhodných autorů. Tyto komponenty často obsahují bezpečnostní mezery, které mohou využít útočníci k napadení celého systémů. Administrátor by měl pravidelně kontrolovat aktualizační balíčky nejen CMS Joomla, ale také jednotlivých použitých komponent.

Druhou důležitou částí je zálohování systému. Zde je důležité věnovat pozornost hlavně zálohám databází, protože právě ty obsahují veškerá data a tedy duševní vlastnictví celého webu. Perioda zálohování se samozřejmě odvíjí od návštěvnosti webu a je třeba tomu zálohovací plán přizpůsobit. Záloha systému z disku jako takového je také důležitá, ale v případě ztráty těchto dat lze vše nainstalovat znovu.

Před každou aktualizací systému a komponent se doporučuje zálohovat jak databázi, tak systém z webového disku.

5. Výukové applety a animace

5.1 Programovací technologie

Vzhledem k tomu, že teorie se nejlépe učí na praktických příkladech a ukázkách, rozhodl jsem se základní principy demonstrovat graficky. Pro implementaci do webového portálu jsem po analýze jako nejvhodnější zvolil jazyk Java. Ten sice nedisponuje takovými grafickými a názornými možnostmi jako například Flash, na rozdíl od Flashe je však co se týče kompatibility na daleko lepší úrovni. Použití vyššího programovacího jazyka nad klasickým webovým jazykem PHP je v tomto případě téměř nutností, protože PHP ve spojení s HTML neumožňuje téměř žádnou uživatelskou interakci. Ta lze sice složitě implementovat i v PHP, jde však o velice složitou a neefektivní práci, kterou se nedosáhne tak dobrých výsledků, jako v případě použití Javy.

Ne vždy je však třeba uživatelská interakce pro názornou představu o tom, jak které algoritmy fungují. Pokud zůstane uživatel pouze pozorovatelem, je mnohem výhodnější použít k demonstraci názorné animace. Pomocí těch je také uživatel schopen danou problematiku pochopit a mnohdy mnohem názorněji. V tomto případě není třeba téměř žádná základní znalost problematiky – tu si uživatel vytváří sledování animace.

Obr. 5.1 Oficiální logo JAVA a FLASH aplikací

Java applety

Java applety se programují pomocí objektově orientovaného programovacího jazyka Java. Jeho hlavní výhodou je vysoká přenositelnost, což znamená, že jeden program je možno spustit na nejrůznějších platformách a přístrojích.

Multiplatformnost je docílena tím, že zdrojový kód je znovu kompilován při každém spuštění programu napsaného v Javě. K tomu je využíváno tzv. virtuálních strojů JVM, které jsou známé i pro uživatele internetu. Velkou výhodou programovacího jazyku Java je jeho otevřenost – vývoj probíhá otevřeně. Tím je minimalizována jeho chybovost a redundantní kód, který je vždy zavčas odhalen.

Hlavní nevýhodou Javy je rychlost spouštění jejich aplikací. Vzhledem k použití již zmíněných virtuálních strojů je třeba kód kompilovat při každém spuštění a toje hlavní důvod zpoždění při spouštění. Po spuštění je však již vše rychlé a svižné jako v klasických aplikacích.

Pro spuštění Java appletů v rámci internetového prohlížeče je nutné mít nainstalovanou podporu Java Virtual Machine (JVM) která se stará o kompilaci zdrojových kódů. V případě, že JVM nainstalováno nemáme, nejsme schopni applet spustit. V dnešní době to však již není problémem jako před cca. 3 lety, kdy JVM nebyla standardní součástí operačních systémů ani prohlížečů. Nyní je JVM v operačních systémech většinou implementována, což umožnilo právě otevření kódu firmou Sun Microsystems v roce 2007.

Flash animace
Flash je programovací jazyk, využívající se pro tvorbu interaktivních animací. Tyto animace je možno použít jak pro webové prezentace, tak pro prezentace v PC. Na interentu se s flashovými animacemi můžeme setkat nejčastěji v podobě reklamních bannerů, kde flash nahradil původní obrázky gif. Jedním z důvodů úspěchu flashe je malá velikost jeho výstupních souborů. Důvodem je, že si veškeré objekty uvnitř animací uchovává ve vektorové (matematické) podobě, nikoliv v podobě bitmapové. Ve flash animaci sice lza použít také bitmapové obrázky, to však degraduje výhodu malých výsledných animací.

Součástí nástrojů pro tvorbu flash animací je jazyk ActionScript, který nám umožňuje do animací vložit určitou část interaktivity. Toho využívají nejčastější aplikace dosažitelné na internetu – flashové hry. K tomu, aby flashové video bylo v počítači, nebo internetovém prohlížeči spustitelné, je nutné mít nainstalovaný doplněk od firmy Adobe – Flash Player. Flash umožňuje exportovat videa nejen do formátu pro Flash Player, ale také do samospustitelného souboru .exe. K jeho přehrání tak není jřeba žádná jiná aplikace. Tento formát lze použít pouze pro desktopové aplikace, nikoliv pro web.

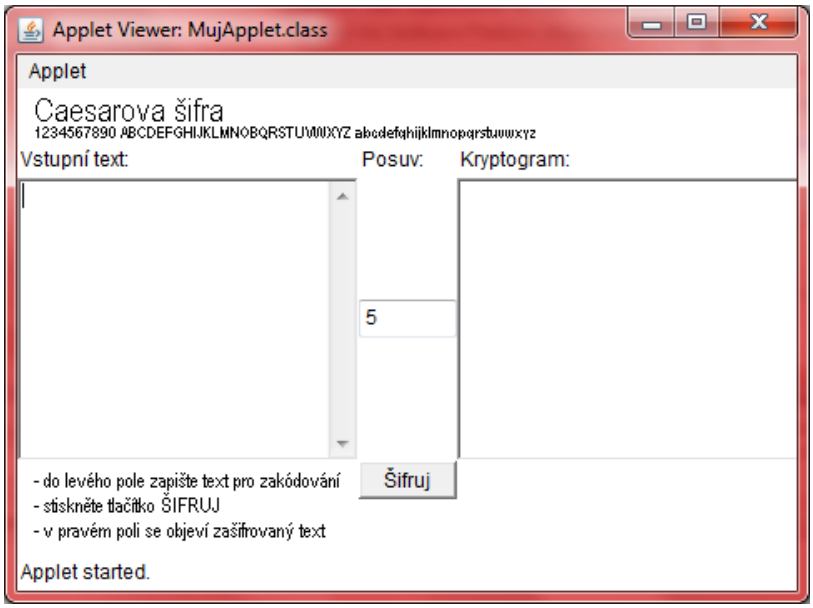
5.2 Applety pro portál
Pro vývoj java appletů existuje mnoho vývojových prostředí a jelikož jde o velice rozšířený jazyk, je mnoho vývojových prostředí volně šiřitelných. Mezi nejznámější patří prostředí Eclipse a NetBeans. Prostředí Eclipse je zaměřeno více na java aplikace pro desktopové počítače. NetBeans je všeobecný nástroj pro mnoho jazyků a syntaxi. Jdou v něm vyvíjet nejen java aplikace a applety, ale například celé webové stránky. Umožňuje rychlo práci s php skripty a umí dokonce tvořit aplikace v jazyce C++.

Pro portál pro podporu kryptografie je důležitá názornost a tak jsem se zaměřil na demonstraci aplikace základních algoritmů a procedur. K appletům, které jsem vytvořil, jsem dostal k dispozici i applety a desktopové aplikace, které v javě tvořili i jiní studenti v rámci svých diplomových prací a prací pro školu. Ty bylo nutno předělat tak, aby byly funkční i v prostředí internetu a webových prohlížečů.

5.2.1 Caesarova šifra

Applet demonstruje základní funkci Caesarovy šifry, neboli posun znaků vstupního textu o předepsaný počet znaků v abecedě. Applet je konstruován tak, že abecedu posouvá v pořadí, jaké je uvedeno v jeho hlavičce. Tedy od číslic, přes velká písmena, až po písmena velká. Stejné znaky jsou tedy povoleny i na vstupu appletu.

V programu jsou pro lepší bezpečnost šifry smazány mezery z původního textu. Pomocí detekování mezer by totiž bylo možno snadno zjistit velikost posuvu znaků.



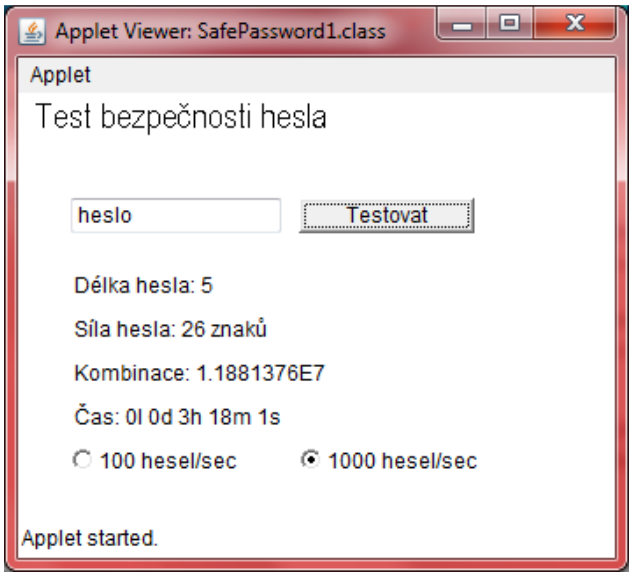
Obr. 5.2 Vzhled appletu Caesarova šifra.

Applet lze využít i na vyzkoušení opačné funkce, tedy rozluštění zašifrovaného textu a to jednoduše tak, že místo přičtení určitého počtu znaků, znaky odečteme.

5.2.2 Ověření bezpečnosti hesla

Vzhledem že bezpečnost uživatelských dat na internetu není dánapouze kvalitou jejich uložení, ale také sílou hesla, které je střeží, rozhodl jsem se vypracovat applet na ověření jejich bezpečnosti. To umožní komukoliv ověřit si jákékoliv svoje heslo. Uživatel tak zjistí, jestli je jeho heslo bezpečné a za jak dlouhou dobu je prolomitelné hrubou silou.

Z hlediska uživatele stačí zadat pouze heslo a stisknout tlačítko „Testovat“. Applet následně provede analýzu na základě použitých znaků. Podle znaků určí abecedu a počet jejich znaků, které jsou potřeba použít pro uhodnutí hesla. Z počtu znaků abecedy pak algoritmus vypočte počet všech kombinací, které jsou pomocí této abecedy a daného počtu znaků hesla vytvořitelné. Počet kombinací je přepočten na čas, krerý by počítač strávil nad nabouráním hesla, pokud by měl vyzkoušet všechny kombinace. Vzhledem k tomu, že osobní počítače nejsou stejně rychlé, umožňuje applet přepínání rychlosti testování hesel. Nižší hodnota, tedy 100hesel za sekundu by měl odpovídat starším počítačům. Vyšší hodnota pak počítačům novějším.



Obr. 5.3 Vzhled appletu Test bezpečnosti hesla.

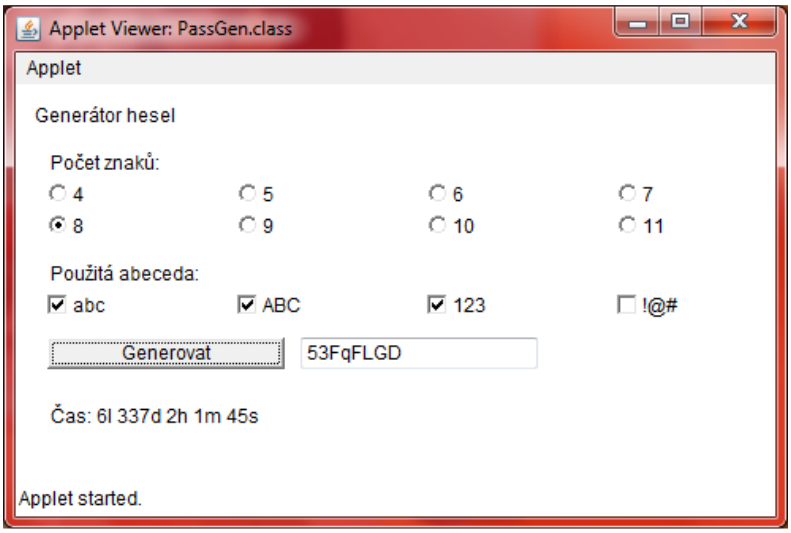
Applet testuje heslo pouze vzhledem k rychlostem osobních počítačů a neuvažuje rychlost stojů přímo sestavených na lámání hesel. Ty jsou však velmi drahé a nepředpokládá se, že by byly použity na prolamování zabezpečení dat u standardních uživatelů.

Applet také nepočítá se slovníkovými útoky, které velmi zkracují dobu luštění hesel hrubou silou. Princip spočívá v databázi nejčastěji používaných slov v heslech, které přednostně zkouší. Pokud uživatel toto takovéto slovníkové heslo použil, vystavuje se nebezpečí nezabezpečení svých dat. V dnešní době je mnoho takovýchto slovníků k dispozici jako freeware na internetu. Pro případného útočníka tedy není problém se k tomuto slovníku dostat. Existují také velice obsáhlé a rychlé databáze všech možných kombinací hesel, zjištěných útočníky po celém světě. Ty jsou však obyčejnému uživateli zapovězeny.

5.2.3 Generátor bezpečných hesel

Vzhledem k důležitosti této tématiky jsem vytvořil generátor bezpečných hesel, který je schopen podle předvoleb generovat bezpečná hesla. Uživatel tak může zvolit rozumný kompromis mezi dobou prolomení hesla a jednoduchostí jeho zapamatování.

Uživatel musí jako vstup zadat délku požadovaného hesla. Volbu má v rozmezí od 4 do 11 znaků. Dále je nutno zadat abecedy, které má generátor pro tvorbu hesla použít. Uživatel má na výběr ze čtyř různých abeced, přičemž je doporučeno vybrat minimálně dvě až tři. Po stisku tlačítka „Generovat“ applet vytvoří heslo podle zadaných parametrů. Pokud se uživateli heslo nelíbí, je možno stisk tlačítka opětovat.



Obr. 5.4 Vzhled appletu Generátor hesla.

Znaky a abecedy jsou voleny na základě generátoru náhodných čísel, zabudovaném přímo v kompilátoru jazyka Java. V editačním poli se ukáže heslo, které applet vygeneroval a uživatel si jej může zkopírovat nebi zapamatovat. Vzhledem k úspoře času se ve spodní části appletu zároveň opět vypočte časová náročnost prolomení hesla hrubou silou.

Tab. 5.1 Porovnání bezpečnosti hesel vzhledem k délce a abecedě [28].

\ délka hesla použité znaky		4 Kombinací 100hesel/s	5 Kombinací 100hesel/s	6 Kombinací 100hesel/s	7 Kombinací 100hesel/s	8 Kombinací 100hesel/s
0-9	10 znaků	10.000 2 minuty	100.000 16 minut	1.000.000 3 hodiny	10.000.000 1 den	100.000.000 11 dní
a-z, 0-9	36 znaků	7.311.616 5 hodin	380.204.032 7 dní	2×10^9 8 měsíců	8×10^{10} 25 let	3×10^{12} 900 let
a-z, A-Z, 0-9	62 znaků	14.776.336 2 dny	916.132.832 3 měsíce	5×10^{10} 18 let	4×10^{12} 1000 let	2×10^{14} 70.000 let
a-z, A-Z, 0-9 ščěř..., @#$%^	85 znaků	52.200.625 6 dní	443.705.312 1 rok	3×10^{11} 120 let	3×10^{13} 10.000 let	3×10^{15} 800.000 let

V tabulce 5.1 si můžeme porovnat časové náročnosti bezpečnosti různých hesel, vzhledem k počtu použitých znaků a typu znaků. Je zde názorně poznat, že pokud budeme mít heslo dlouhé 8 znaků, stačí nám k poměrně dobrému zajištění bezpečí znaky malé abecedy a číslice. Všeobecně se doporučuje používat minimálně znaky velké a malé abecedy a číslice v hesle o minimální délce 8 znaků. Rychlý vývoj výpočetních technologií totiž znamená, že dnes bezpečné heslo již za rok bezpečné být nemusí.

5.2.4 Ostatní applety

Pro rozšíření výuky usnadnění práce s applety, které již škola vlastní v rámci bývalých diplomových prací jsem dostal k dispozici applety studentů, kteří zpracovávali podobná témata přede mnou. Některé z appletů bylo třeba upravit pro zobrazení na webových stránkách a bylo také nutno překompilovat jejich strukturu tak, aby byly webovými stránkami zobrazitelné. Po hodinách strávených testováním jsem dospěl k závěru

neimplementovat do webových stránek přímo třídy (poznají se podle koncovky .class) vzhledem k tomu, že jde o poměrně složité applety. K implementaci jsem použil netradiční metodu, používanou spíše pro desktopové aplikace – spustitelné balíčky (končící koncovkou .jar). V desktopové aplikaci umí tyto balíčky javový virtuální stroj přímo skompilovat a spustit. Nevýhodou je, že pro webové aplikace je nutné znát vnitřní uspořádání balíčků tohoto souboru a to následně uvést při jeho spouštění.

```
<applet code="podpis.Podpis" archive="Podpis.jar"
width="770" height="680"></applet>
```

Podle výše uvedeného příkladu přistupujeme ve spustitelném archivu **Podpis.jar** k balíku **podpis**, ve kterém je implementována třída Podpis.class.

Výpočet inverzního prvku
Jednoduchý applet, využívající se při výpočtech součtů a násobení bodů v eliptických křivkách [14].

Zobrazení a výpočet eliptické křivky
Applet, který umožňuje na základě zadání bodů *a, b* vygenerovat a zakreslit příslušnou eliptickou křivku. Kliknutím na zobrazenou můžeme přidávat body, se kterými lze provádět operace součtu, popřípadě násobení příslušného bodu konstantou [14].

Zobrazení množiny prvků náležících zadané eliptické křivce
V appletu je opět po zadání základních parametrů vykreslena eliptická křivka, která je tentokrát zobrazena jako množina bodů. Nad tou lze opět provádět operace jako v předchozím případě. Applet navíc dokáže vypočítat řád bodu křivky, na který klikneme [14].

Šifrování pomocí eliptických křivek
Po spuštění appletu nastavíme parametry eliptické křivky a text, který chceme šifrovat. Jednotlivá písmena textu jsou pak přiřazena bodům eliptické křivky [14].

Digitální podpis pomocí eliptických křivek
Opět je zde možnost vybrat eliptickou křivku dle parametrů. Po zadání textu je možno vygenerovat elektronický podpis. Pokud následně v ověřované zprávě změníme jakýkoliv znak a spustíme ověření podpisu, zjistíme, že podpis již není platný. Pokud ověření spustím, aniž bychom zprávu změnili, dojde k ověření podpisu [14].

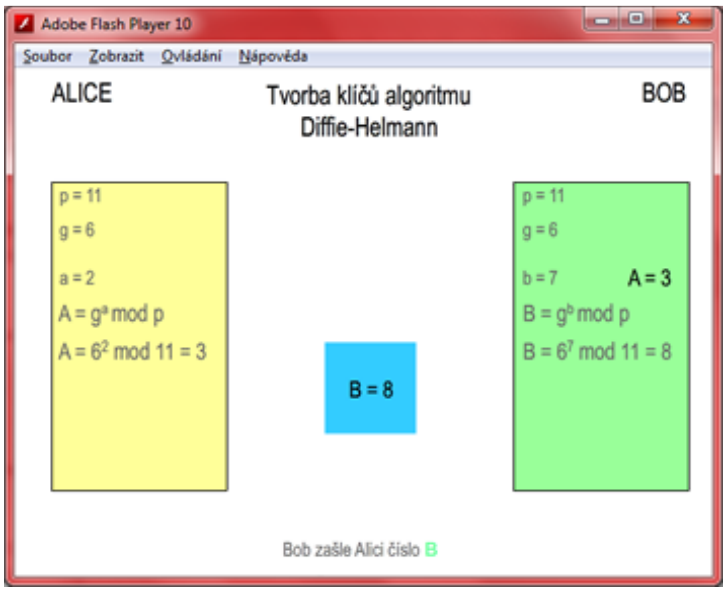
5.3 Animace pro portál

Pro vývoj flashových animací lze v dnešní době využít v podstatě pouze jednu rodinu programů a to od firmy Adobe. Vzhledem k tomu, že jde o komerční formát, jsou všechny tyto programy placené, a tudíž nelze k tvorbě flash animací a bannerů použít

žádný vhodný a volně šiřitelný program. Naštěstí jsou nástroje volné k vyzkoušení pro nekomerční využití, čehož jsem využil i pro tvorbu animací pro portál Cryptohub.

5.3.1 Animace protokolu Diffie-Hellman

Flash animace názorně předvádí sestavování klíčů pomocí protokolu Diffie-Helmann tak, jak to bylo popsáno v teoretickém rozboru této práce. V horní části animace jsou postupně zobrazovány výpočty a generované hodnoty v pořadí, v jakém algoritmus postupuje. Názorně je také zobrazena výměna hodnot mezi Alicí a Bobem.



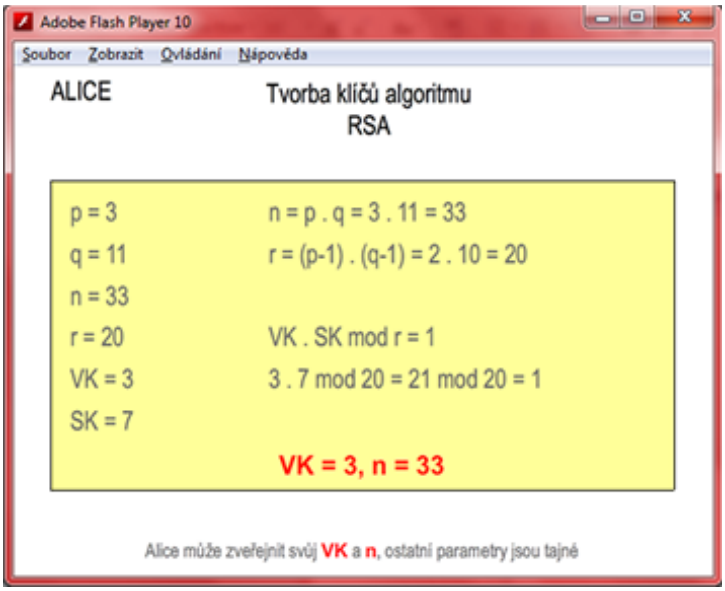
Obr. 5.5 Vzhled animace protokolu Diffie-Helmann.

Ve spodní části animace se průběžně mění popis právě vykonávaných činností tak, aby byla animace co nejvíce pochopitelná i pro úplného začátečníka.

5.3.2 Animace tvorby RSA klíčů

Vzhledem k tomu, že výpočet klíčů algoritmu RSA je poměrně jednoduchým postupem, je v tomto appletu výpočet demonstrován rovnou na příslušném příkladě, kdy Alice generuje svůj klíčový pár na základě náhodně vybraných čísel. Ve skutečnosti nejsou čísla použitá v animaci reálná, protože ve skutečnosti se pracuje s čísly mnohonásobně většími. Pro základní demonstraci a představu o výpočtu klíčů tento příklad dostačuje.

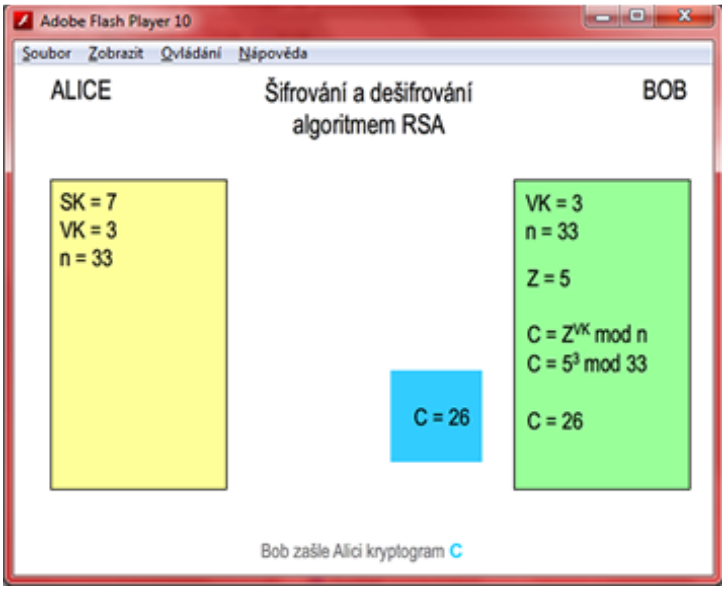
V hlavní části appletu jsou opět zobrazeny základní parametry klíčového páru, ty jsou soustředěny na levou stranu animace. Na pravé straně jsou pak znázorněny výpočty postupně tak, jak jsou za sebou.



Obr. 5.6 Vzhled animace tvorby klíčového páru RSA.

5.3.3 Animace šifrování a dešifrování pomocí RSA

V návaznosti na předchozí animaci je zde zobrazeno použití vygenerovaných klíčů. Pro větší názornost klíče i číslo *n* souhlasí s předchozím příkladem, abychom si tak mohli utvořit ucelenější představu o fungování celého algoritmu.



Obr. 5.7 Vzhled animace šifrování algoritmem RSA.

V animaci je opět vše vykreslováno v pořadí, v jakém ve skutečnosti jednotlivé kroky probíhají. Vše je doprovázeno slovním vysvětlením právě probíhajícího úkonu ve spodní části animace.

5.3.4 Ostatní animace

Stejně jako tomu bylo v případě appletů, i co se týká animací, mi byly poskytnuty podklady, které byly zpracovávány jinými studenty v rámci svých diplomových prací v dřívějších letech. Vzhledem k tomu, že principy v kryptografii zůstávají stále stejné, lze tyto materiály použít i pro portál Cryptohub. V tomto případě nebylo nutno dělat do animací žádné zásahy a ani to nebylo možné, protože jsem k dispozici dostal finální zkompilovaná videa, u kterých již není možno měnit zdrojové kódy.

Sestavení spojení pomocí 802.1x

Animace názorně zobrazuje kroky, potřebné k sestavení spojení na základě standardu 802.1x. Tento standard značně rozšiřuje zabezpečení v rámci sítě. K autentizaci uživatele je zde využíváno autentizačního serveru (Radius) přes zprostředkovatele, nejčastěji přístupového bodu Wi-Fi sítě [15].

Ověření uživatele podle standardu GSM

V animaci je názorně zobrazen průběh autentizace uživatele, který se připojuje k GSM síti. Je zde také vidět, které šifry se v jaké části autentizace používají. Podle animace je vidět, že téměř stejný postup se odehrává jak v cílovém uživatelském zařízení, tak v základové stanici operátora. To je důležité pro to, aby mohla být komunikace mezi koncovým zařízením a základovou stanicí šifrována [15].

Sestavení zabezpečeného spojení pomocí TLS

Tento protokol se stal nástupcem SSL a poskytuje velmi rozšířené možnosti pro zabezpečení transportní síťové vrstvy. Tímto protokolem lze zabezpečit standardní komunikační protokoly, jako jsou www, pošta, bankovnictví a podobně. Na animaci je názorně znázorněna výměna všech potřebných informací a klíčů pro sestavení zabezpečeného spojení TLS [15].

Autentizační systém Kerberos

Protokol umožňující bezpečnou autentizaci na základě důvěře třetí straně, neboli tzv. autentizačnímu serveru. Na základě potvrzení autentizačním serverem si může klient požádat o poukázku na službu, kterou má zájem využívat. Po předložení poukázky serveru poskytujícímu požadovanou službu, je tato služba klientovi poskytnuta. V animace je znázorněna výměna informací postupně mezi klientem a všemi třemi servery tak, jak následuje za sebou [15].

Demonstrace kvantových kryptografických systémů

Animace demonstruje protokoly, jejichž funkce je vysvětlena v kapitole 2.9 a podtrhuje tak pochopení jejich funkce. Na počátku si uživatel může vybrat z daných kvantových kryptografických algoritmů a po zadání všech parametrů komunikace může přehledně sledovat její průběh, který je názorně animován a vkreslován [24].

6. Závěr

V rámci semestrálního projektu jsem vytvořil nejzákladnější návrh obsahu webu a grafický návrh portálu. Pro realizaci portálu jsem z dostupných GNU/GPL CMS systémů vybral redakční systém CMS Joomla a to z toho důvodu, že jsem s ním měl ještě před realizací diplomové práce zkušenosti.

Na takto vytvořený základ jsem navázal v diplomové práci. Vhledem k tomu, že obor kryptografie je velice široký a zahrnuje v sobě více oborů, bylo nutno teoretickou část přestrukturovat tak, aby to vyhovovalo právě tomuto pojetí. Teoretická část byla doplněna o analýzu stávajícího stavu webových portálů zabývajících se kryptografií, na základě které byla upravena struktura dokumentu tak, jak by měl vypadat obsah výsledného portálu. Výsledná teorie tedy zahrnuje téměř vše od historie kryptografie, přes matematické základy potřebné pro pochopení současných metod, až po vysvětlení dnes používaných algoritmů včetně stručných popisů funkce. Pro veškeré popisy jsem se snažil čerpat informace z co největšího počtu zdrojů tak, aby byly snadno pochopitelné.
Pro praktickou část portálu jsem vytvořil tři výukové applety v jazyce Java a tři výukové animace v jazyce Flash. Doplnil jsem tak spektrum appletů a animací, které mi byly poskytnuty vedoucím práce.
Na základě vytvořené databáze znalostí jsem vytvořil webovou prezentaci právě s pomocí CMS systému Joomla. Do systému jsem nainstaloval komponenty, které by takový portál měl mít, a které zjednodušují práci jak administrátorům webu, tak samotným uživatelům. Zároveň jsem do vytvořeného systému nahrál většinu článků prezentovaných v této diplomové práci, společně s upravenými obrázky, animacemi a java applety. Aktuální podoba rozčlenění webu je velice variabilní a dá se podle potřeby měnit. Záleží pouze na požadavcích.

Součástí diplomové práce je také jednoduchý manuál, ve kterém jsem popsal nutné kroky pro přesun celého portálu na jiný hosting a jinou doménu.

Výsledkem diplomové práce je tedy nejen výukový portál, ale také poměrně snadno pochopitelná databáze znalostí, která se velice dobře dá použít k výukovým účelům.

Seznam použité literatury

[1] BURDA K. *Bezpečnost informačních systémů.* [Skriptum VUT v Brně.], Brno, 2005. 104s.

[2] Security portal. *Praktické základy kryptografie a steganografie*. [online]. 2004 - [cit. 8. prosince 2009]. Dostupné na WWW: http://www.security-portal.cz/clanky/praktické-základy-kryptologie-steganografie

[3] Wikipedia Foundation. *Autentizace*. [online]. 2009 - [cit. 3. prosince 2009]. Dostupné na WWW: http://cs.wikipedia.org/wiki/Autentizace

[4] Wikipedia Foundation. *Kryptografie*. [online]. 2009 - [cit. 17. listopadu 2009]. Dostupné na WWW: http://cs.wikipedia.org/wiki/Kryptografie

[5] Zive.cz. *Steganografie - ukryjte, že máte tajemství*. [online]. 2008 - [cit. 17. listopadu 2009]. Dostupné na WWW: http://uzivatel.blog.zive.cz/2008/10/steganografie-ukryjte-ze-mate-tajemstvi/

[6] Wikipedia Foundation. *Symetrická kryptografie*. [online]. 2009 - [cit. 21. listopadu 2009]. Dostupné na WWW: http://cs.wikipedia.org/wiki/Symetrická_kryptografie

[7] Svět sítí & Infinity, a.s.. *Pravdy o elektronickém podpisu a šifrování*. [online]. 2003 - [cit. 22. listopadu 2009]. Dostupné na WWW: http://www.svetsiti.cz/view.asp?rubrika=Tutorialy&clanekID=245

[8] BĚHÁLEK M. *Pravdy Bezpečnost a zabezpečení*. [online]. 2007 - [cit. 27. listopadu 2009]. Dostupné na WWW: http://www.cs.vsb.cz/behalek/vyuka/pcsharp/text/ch09s01.html

[9] Wikipedia Foundation. *Search Engine Optimalization*. [online]. 2009 - [cit. 28. listopadu 2009]. Dostupné na WWW: http://cs.wikipedia.org/wiki/Search_Engine_Optimization

[10] Joomla!. *Joomla! Official Documentation*. [online]. 2009 - [cit. 2. prosince 2009]. Dostupné na WWW: http://docs.joomla.org/

[11] ZEMAN V. *Kryptografie v informatice.* [Skriptum VUT v Brně.], Brno, 2009.

[12] Wikipedia Foundation. *Portal:Cryptography*. [online]. 2009 - [cit. 13. prosince 2009]. Dostupné na WWW: http://en.wikipedia.org/wiki/Portal:Cryptography

[13] Wikipedia Foundation. *Hašovací funkce*. [online]. 2009 - [cit. 1. prosince 2009]. Dostupné na WWW: http://cs.wikipedia.org/wiki/Hašovací_funkce

[14] SZTURC, J. *Softwarová podpora výuky kryptosystémů založených na eliptických křivkách.* Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2009. 65 s. Vedoucí diplomové práce doc. Ing. Karel Burda, CSc.

[15] MAREK, T. *Softwarová podpora výuky kryptografických protokolů.* Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2009. 69 s. Vedoucí práce doc. Ing. Karel Burda, CSc.

[16] BITTO, O. *Historie kryptologie*. [online]. 2009 – [cit. 22.května 2010].
Dostupné na WWW: http://www.fi.muni.cz/usr/jkucera/pv109/2003/xbitto.htm

[17] LEVICKÝ, D. *Kryptografia v informačnej bezpečnosti*. Košice : Elfa, s.r.o., 2005. 274 s.

[18] WEISSTEIN, W. *"Euclidean Algorithm." From MathWorld.* [online]. 2009 – [cit. 22.května 2010].
Dostupné na WWW: http://mathworld.wolfram.com/EuclideanAlgorithm.html

[19] Wikipedia Foundation. *Grupa*. [online]. 2010 - [cit. 23. května 2010].
Dostupné na WWW: http://cs.wikipedia.org/wiki/Grupa

[20] Ústav matematiky FSI VUT Brno. *Matematika online.* [online]. 2005 – [cit. 22.května 2010]. Dostupné na WWW:
http://mathonline.fme.vutbr.cz/download.aspx?id_file=796

[21] VANĚK, T. *Režimy činnosti blokových šifer.* [online]. 2010 – [cit. 24.května 2010].
Dostupné na WWW: http://www.comtel.cz/files/download.php?id=4852

[22] SAVARD, J. *The Advanced Encryption Standard (Rijndael).* [online]. 1998 – [cit. 10.dubna 2010]. Dostupné na WWW: http://www.eng.tau.ac.il/~yash/crypto-netsec/rijndael.htm

[23] WEISSTEIN, W. *" Discrete Logarithm." From MathWorld.* [online]. 2009 – [cit. 22.května 2010]. : http://mathworld.wolfram.com/DiscreteLogarithm.html

[24] PAJTINOVÁ, M. *Metody kvantové kryptografie*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2009. 49 s. Vedoucí bakalářské práce doc. Ing. Václav Zeman, Ph.D.

[25] Wikipedia Foundation. *Data Encryption Standard*. [online]. 2010 - [cit. 23. května 2010].
Dostupné na WWW: http://cs.wikipedia.org/wiki/Data_Encryption_Standard, http://homel.vsb.cz/~mor196/mal338.pdf

[26] OCHODKOVÁ, E. *Přínos teorie eliptických křivek k řešení moderních kryptografických systémů.* [online]. 2003 - [cit. 16. května 2010]. Dostupné na WWW: http://www.cs.vsb.cz/arg/workshop/files/ecc_eli.pdf

[27] Wikipedia Foundation. *Útoky postranními kanály*. [online]. 2009 - [cit. 17. května 2010]. Dostupné na WWW: http://cs.wikipedia.org/wiki/Útoky_postranními_kanály

[28] Wikipedia Foundation. *Bezpečné heslo*. [online]. 2010 - [cit. 18. května 2010]. Dostupné na WWW: http://cs.wikipedia.org/wiki/Bezpečné_heslo

Seznam použitých zkratek

A5		- šifra využíváná pro zabezpečení GSM
BB84		- protokol kvantové kryptografie
C		- symbol pro označení kryptogramu
C++		- objektově orientovaný program. jazyk
CMS	- Content Management System	- systém pro správu obsahu
CSS	- Cascading Style Sheets	- jazyk pro popis zobrazení web. stránek
D-H	- Diffie-Helmann	- protokol vytvoření šifrovaného spojení
DSA	- Digital Signature Algorythm	- standard pro tvorbu digitálního podpisu
RSA	- Rivest, Shamir, Adleman	- asymetrický šifrovací protokol
RC4		- proudová šifrovací metoda
DES	- Data Encryption Standard	- nepoužívaný protokol sym. krypt.
AES	- Advanced Encryption standard	- standard pro bezpečnou komunikaci
3DES	- Tripple DES	- DES se zvýšenou bezpečností
ECC	- Eliptic Curve Cryptography	- kryptografie eliptickými křivkami
FISH		- proudová šifrovací metoda
GNU/GPL	- GNU General Public License	- licence pro svobodný software
GSM	- Groupe Spécial Mobile	- standard pro mobilní telefony
HTML	- Hypertex Markup Language	- jazyk pro vytváření www stránek
IBM		- přední světová společnost v oboru IT
MD5	- Message-Digest Algorithm	- rozšířená rodina hašovacích funkcí
M	- Message	- zkratka pro označení zprávy
MySQL		- databázový systém
NIST		- institut standardů a technologie USA
NSA	- National security agency	- národní bezpečnostní agentura USA
PHP	- Hypertext Preprocessor	- skriptovací programovací jazyk
PNP		- pseudonáhodná posloupnost
RSS	- Really Simple Syndication	- formát pro sdílení novinek na webu
SEO	- Search Engine Optimization	- optimalizace stránek pro vyhledávače
SHA	- Secure Hash Algorythm	- rozšířená hašovací funkce
SK		- zkratka pro označení soukromého klíče
SSL	- Secure Sockets Layer	- vrstva pro zabezpečení komunikace
VK		- zkratka pro označení veřejného klíče
WEP	- Wired equivalent privacy	- původní zabezpečení Wi-Fi sítí z 1999
WPA	- Wi-Fi protected access	- zabezpečení Wi-Fi sítí, nástupce WEPu
WYSIWYG	- What you see is what you get	- obrazově názorný editor obsahu
XML	- Extensible Markup Language	- značkovací jazyk pro různé typy dat
XOR	- Exclusive Disjunction	- logická/matematická bitová operace
WWW	- World Wide Web	- aplikace internetového protokolu
Z		- česká zkratka pro označení zprávy

Rejstřík pojmů

Administrace	- správa a udržování, v našem případě webového portálu
Algoritmus	- přesný návod, nebo postup na vyřešení dané úlohy
Applet	- softwarová komponenta jednoho programu běžící v jiném
Autentizace	- ověření identity subjektu
Backend	- systém na pozadí webu, nejčastěji slouží ke správě obsahu
Eliptická křivka	- algebraická struktura konstruovaná nad tělesem
Frontend	- část internetových stránek viditelná pro uživatele
Hash (haš, heš)	- zakódovaný otisk zprávy přesně dané délky
Integrita	- zajištění komletnosti a neporušenosti dat
Korelace	- vzájemný vztah mezi dvěma veličinami, závislost mezi nimi
Kryptoanalýza	- věda zabývající se získáváním obsahu šifrovaných informací
Kryptografie	- věda zabývající se utajením obsahu zprávy pomocí šifrování
Kvantová fyzika	- fyzikální teorie, popisující stav na základě pravděpodobnosti
Optimalizace	- metoda, vedoucí k zvětšení evektivity výkonu systému
Permutace	- skupina všech prvků, uspořádaná v jakémkoliv možném pořadí
Steganografie	- věda, zabývající se ukrýváním zprávy
Šifra	- „nerozluštitelný“ zašifrovaný otisk původní zprávy
Webdesign	- činnost návrhu webových stránek a aplikací

Seznam příloh

A.Návod na zprovoznění portálu a jeho nastavení

Celý obsah portálu je rozdělen na 2 základní části:

1. obsah webového prostoru
2. obsah databáze

Pro zprovoznění portálu na doménové adrese je zapotřebí nastavit následující části webu přesně v daném pořadí:

1. bitová kopie obsahu doménového prostoru
2. import databáze a jejího obsahu
3. přenastavení konfiguračního souboru

1. Bitová kopie obsahu doménového prostoru

Pro přesun portálu postaveného na CMS Joomla do nového umístění se nedoporučuje vytvářet na doménovém prostoru novou čistou instalaci. Vhodnějším postupem je přímá bitová kopie celého obsahu z jednoho doménového prostoru do druhého. Vyhneme se tím nepříjemnostem způsobeným chybějícími a nekompatibilními komponentami. Nejvhodnější je pro tuto operaci využít jeden z dostupných FTP klientů s ověřováním úspěšnosti přenosu.

Webový CMS systém Joomla obsahuje v základní konfiguraci okolo 3500 souborů, s nadstavbami použitými v našem portále může jejich hodnota dosahovat až 4000. Protože přenos přes FTP není stavěný na přenos velkého počtu malých souborů, bude tvorba bitové kopie časově náročnější, je proto třeba částečná trpělivost.

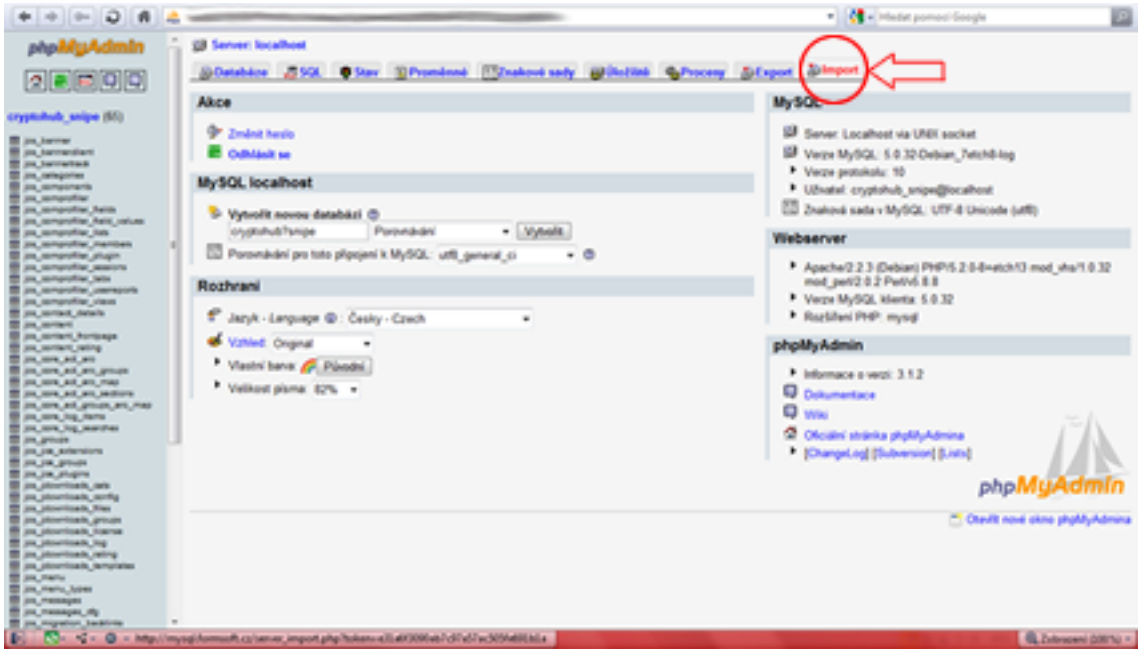
Po tomto přenosu portálu do nového úložiště bude portál prozatím nefunkční!

2. Import databáze a jejího obsahu

Pro zprovoznění portálu je nutno importovat i jeho obsah do databáze, která přísluší danému webovému prostoru. Vzhledem k tomu, že portál byl instalován na databázovém serveru MySql, je vhodné použít tento databázový server i novém doménovém prostoru. Pokud by daný prostor nepodporoval databáze MySql, bude nutné soubor s daty převést na daný typ databáze. Toto by však v naprosté většině případů neměl být problém, protože 99% webových serverů podporuje databáze MySql.

Import se provádí velice jednoduše. Po přihlášení do PhpMyAdmin přiděleným přístupovým jménem a heslem zvolíme **import** v pravém horním rohu (viz. obrázek A1). Následně vybereme v adresářové struktuře soubor s databází a ten nahrajeme na server. Veškerý import databází a tabulek včetně obsahu by měl proběhnout automaticky.

Pokud máme na hostingu přidělené jméno databáze a nemůžeme tak vytvářet databázi další, je pro tuto situaci připraven i soubor s importem pouze tabulek a dat.



Obr. A1.: Import databáze pomocí PhpMyAmin.

Po tomto importu databáze do nového úložiště bude portál prozatím nefunkční!

3. Přenastavení konfiguračního souboru

Vzhledem k tomu, že byl portál při instalaci umístěn na jiném úložišti, nesouhlasí údaje v konfiguračním souboru, který se tvoří při instalaci a od té doby je do něj zakázán přístup i zápis. Tento soubor je umístěn v kořenovém adresáři webového prostoru a jeho název je **configuration.php**. Pomocí volně dostupných prostředků si jej otevřeme přímo na přes FTP, nebo si jej uložíme do počítače. Po otevření uvidíme konfigurační příkazy, z nichž následující je třeba upravit:

```
var $log_path = '/var/www/domains/snipers.cz/cryptohub/webdata/logs';
var $tmp_path = '/var/www/domains/snipers.cz/cryptohub/webdata/tmp';
```

- **nastavení cest k složkám /logs a /tmp**
- **obě složky musí mít povoleny zápis**

```
var $ftp_enable = '0';
var $ftp_host = '127.0.0.1';
var $ftp_port = '21';
var $ftp_user = '';
var $ftp_pass = '';
var $ftp_root = '';
```

- **nastavení ftp serveru Joomly**
- **v naší konfiguraci zakázán z důvodu zabezpečení**

var $dbtype = 'mysql';
var $host = 'localhost';
var $user = 'cryptohub_snipe';
var $db = 'cryptohub_snipe';

- **nastavení typu databáze (dbtype)**
- **nastavení přístupové adresy k databázovému serveru, ta je většinou na stejném stroji jako data (host)**
- **nastavení přístupového jména k databázi (user)**
- **nastavení jména databáze, buď importované ze souboru nastavené od poskytovatele (db)**

var $dbprefix = 'jos_';
var $mailfrom = 'tf2001@seznam.cz';
var $fromname = 'CryptoHUB';
var $password = 'cryptohub';
prefix tabulek s daty CMS Joomla (dbprefix)

- **e-mailová adresa administrátora, jsou z ní zasílány e-maily (mailfrom)**
- **jméno uvedené v e-mailu (fromname)**
- **heslo do databáze (password)**

Po provedení výše uvedených změn by měl být webový portál plně funkční!

4. Přístupy a nastavení CMS Joomla

CMS systém Joomla se dělí na dvě základní části – frontend a backend. Frontend je přístupný přímo zadáním webové adresy:

http://www.mojedomena.cz

přístup na tuto adresu je veřejný a nejsou k němu třeba žádné přístupové údaje. I přes toto rozhraní se však mohou uživatelé webu přihlašovat, spravovat svoje profily, přidávat články atd., vše podle nastavených přístupových oprávnění.

Druhou částí webového portálu je backend, neboli administrátorská část. Do ní mají přístup osoby s vyšším oprávněním než pouze standardní registrovaný uživatel. Administrační část je přístupná po zadání adresy:

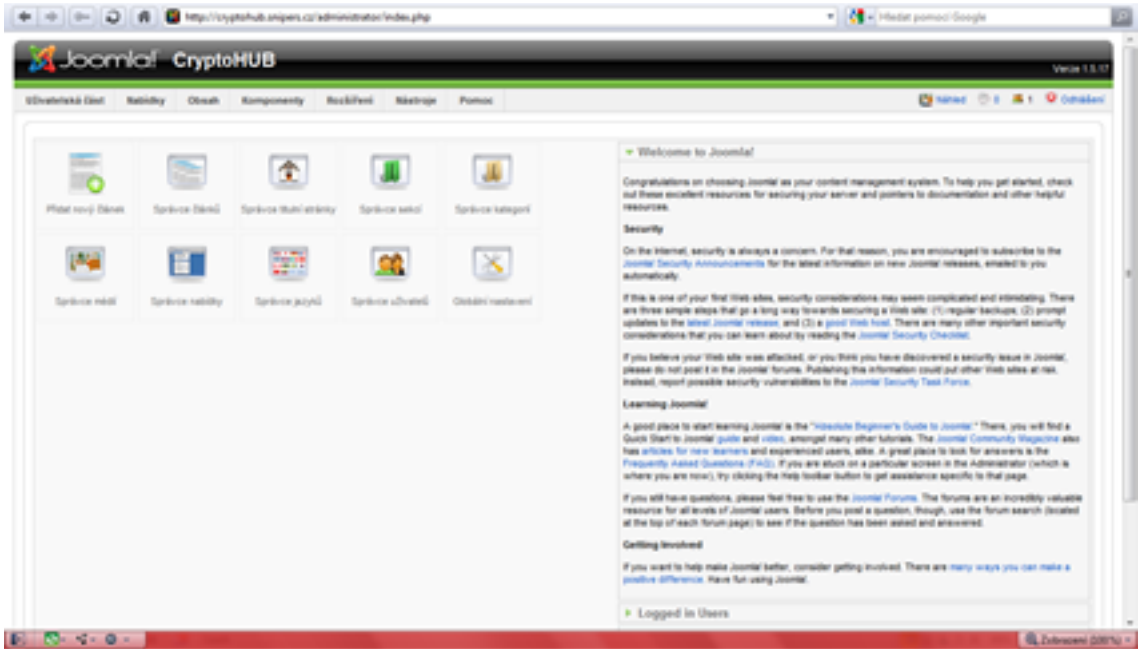
http://www.mojedomena.cz/administrator

Přístupové údaje pro superadministrátora jsou:

jméno: **admin**
heslo: **cryptohub**

B. Výsledný vzhled portálu CryptoHUB

Obr. B1.: Vzhled úvodní stránky portálu.



Obr. B2.: Vzhled úvodní stránky administrace portálu.

C. Obsah přiloženého DVD disku

- elektronická verze zadání diplomové práce ve formátu pdf
- elektronická verze desek diplomové práce ve formátu pdf
- elektronická verze vypracování diplomové práce ve formátu pdf
- elektronická verze vypracování diplomové práce ve formátu docx
- elektronická verze licenční smlouvy ve formátu pdf
- veškeré autorské obrázky použité pro diplomovou práci ve formátu png
- veškeré autorské obrázky použité pro webovou prezentaci ve formátu png
- obrázky použité pro design portálu Cryptohub
- šablona pro CMS Joomla použitá pro tvorbu designu
- zkomprimovaný obsah webového disku pro instalaci na jinou doménu
- obsah webové databáze se skripty pro její vytvoření ve formátu sql
- obsah webové databáze ve formátu sql
- návod na instalaci portálu Cryptohub na jinou doménu
- screenshoty finální verze webového portálu ve formátu png
- dokumentace CMS systému Joomla!
- zdrojové kódy appletů
- zdrojové kódy flash animací