

Aplikovaná moderní kryptologie v informačních technologiích

Applied Modern Cryptology in Information Technology

Silvia Panenková

Bakalářská práce
2013



Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

Univerzita Tomáše Bati ve Zlíně

Fakulta aplikované informatiky

akademický rok: 2012/2013

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Silvia Panenková**

Osobní číslo: **A10983**

Studijní program: **B3902 Inženýrská informatika**

Studijní obor: **Informační a řídicí technologie**

Forma studia: **kombinovaná**

Téma práce: **Aplikovaná moderní kryptologie v informačních technologiích**

Zásady pro vypracování:

1. Vypracujte literární rešerši na dané téma.
2. Provedte teoretický rozbor současných technik a metod moderní kryptologie, které se používají v informačních technologiích.
3. Vypracujte jednoduché demonstrativní ukázky šifrovacích metod a principů a porovnání různých metod zabezpečení.
4. Vytvořte multimediální pomůcku případně webovou prezentaci pro účely zkvalitnění výuky předmětu Kryptologie na FAI.

Rozsah bakalářské práce:

Rozsah příloh:

Forma zpracování bakalářské práce: tištěná/elektronická

Seznam odborné literatury:

1. ZELENKA, Josef. Ochrana dat: kryptologie. Vyd. 1. Hradec Králové: Gaudeamus, 2003, 198 s. ISBN 80-704-1737-4.
2. ČANDÍK, Marek. Základy informační bezpečnosti. Vyd. 1. Zlín: Univerzita Tomáše Bati ve Zlíně, 2004, 107 s. Učební texty vysokých škol (Univerzita Tomáše Bati ve Zlíně). ISBN 80-731-8218-1.
3. VONDRUŠKA, Pavel. Kryptologie, šifrování a tajná písma. 1. vyd. Praha: Albatros, 2006, 340 s. Oko. ISBN 80-000-1888-8.
4. KATZ, Jonathan a Yehuda LINDELL. Introduction to modern cryptography. Boca Raton: Chapman, 2008, xviii, 534 s. ISBN 978-1-58488-551-1.
5. PIPER, F a Sean MURPHY. Kryptografie. 1. vyd. v českém jazyce. Překlad Pavel Mondschein. Praha: Dokořán, 2006, 157 s. ISBN 80-736-3074-5.
6. BITTO, Ondřej. Šifrování a biometrika aneb tajemné bity a dotyky. Vyd. 1. Kralice na Hané: Computer Media, 2005, 168 s. ISBN 80-866-8648-5.
7. KLÍMA, Vlastimil; ROSA, Tomáš. Kryptologie pro praxi ?DSA, ECDSA. Dostupné z WWW: http://crypto-world.info/klima/2004/st_2004_04_21_21.pdf.

Vedoucí bakalářské práce:

Ing. Roman Šenkeřík, Ph.D.

Ústav informatiky a umělé inteligence

Datum zadání bakalářské práce:

24. února 2013

Termín odevzdání bakalářské práce:

14. června 2013

Ve Zlíně dne 24. února 2013

prof. Ing. Vladimír Vašek, CSc.
děkan



prof. Ing. Vladimír Vašek, CSc.
ředitel ústavu

ABSTRAKT

Táto bakalárska práca sa zaoberá aplikáciou moderných kryptografických systémov v informačných technológiách. V teoretickej časti sa práca zaoberá vybranými témami z oblasti kryptografie – od stručnej histórie, cez základné kryptografické štandardy, až po informačnú bezpečnosť. Praktická časť obsahuje výklad matematického aparátu využívaného v kryptografii, teoretický rozbor niektorých šifrovacích algoritmov a spôsoby ich aplikácie.

Súčasťou tejto práce je webová prezentácia, ktorá je určená na skvalitnenie výučby predmetu Kryptológia na FAI UTB v Zlíne.

Kľúčové slová: symetrická kryptografia, asymetrická kryptografia, hašovacie funkcie, aplikovaná kryptológia, informačná bezpečnosť, sieťová bezpečnosť

ABSTRACT

This thesis deals with the application of modern cryptographic systems in information technologies. The theoretical part deals with selected cryptographic topics— such as a brief history of cryptography, basic cryptographic standards and information security.

The practical part contains an interpretation of mathematical theories used in cryptography, theoretical analysis of some encryption algorithms and their applications.

This thesis also includes a web presentation for the improvement of teaching Cryptology at FAI UTB in Zlín.

Keywords: symmetric cryptography, asymmetric cryptography, hash functions, applied cryptology, information security, network security

Chcela by som sa poďakovať vedúcemu mojej bakalárskej práce Ing. Romanovi Šenkeříkovi, PhD. za jeho ochotu, ústretovosť a odborné rady.

Ďakujem aj svojim deťom, mame, manželovi a jeho rodičom za to, že mi vytvorili priaznivé podmienky, pomáhali a mali trpezlivosť nielen počas tvorby tejto práce, ale predovšetkým v čase celého môjho štúdia.

Prohlašuji, že

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

Prohlašuji,

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

Ve Zlíně

.....
podpis diplomanta

OBSAH

ÚVOD.....	10
I TEORETICKÁ ČASŤ.....	11
1 ZÁKLADNÉ POJMY	12
2 KRYPTOGRAFICKÉ ZÁKLADY	14
2.1 SYMETRICKÁ KRYPTOGRAFIA.....	14
2.2 ASYMETRICKÁ KRYPTOGRAFIA	14
2.3 HYBRIDNÁ KRYPTOGRAFIA	15
2.4 ZÁSADY A PRAVIDLÁ KRYPTOLÓGIE.....	15
2.5 TYPOLÓGIA ALGORITMOV	16
3 HISTÓRIA A VÝVIN.....	17
3.1 STAROVEKÁ KRYPTOLÓGIA	17
3.2 STREDOVEKÁ KRYPTOLÓGIA	18
3.3 20. STOROČIE.....	18
3.4 MODERNÁ KRYPTOLÓGIA.....	19
4 KRYPTOANALÝZA A ÚTOKY PROTI ŠIFRÁM	20
4.1 ÚTOKY PROTI ŠIFRÁM.....	20
4.2 ÚTOK HRUBOU SILOU (BRUTE FORCE ATTACK)	21
4.3 NIEKTORÉ ŠPECIÁLNE DRUHY ÚTOKOV	21
5 INFORMAČNÁ BEZPEČNOSŤ.....	22
5.1 HISTÓRIA POČÍTAČOVEJ KRIMINALITY	22
5.2 BEZPEČNOSŤ V INFORMAČNÝCH TECHNOLOGIÁCH.....	23
5.3 ZÁKLADNÉ PRINCÍPY INFORMAČNEJ BEZPEČNOSTI:	23
5.4 ÚTOKY NA INFORMAČNÉ SYSTÉMY:	23
5.5 PROSTRIEDKY SIEŤOVEJ BEZPEČNOSTI	23
6 KRYPTOGRAFICKÉ ŠTANDARDY	24
6.1 DES (DATA ENCRYPTION STANDARD).....	24
6.2 AES (ADVANCED ENCRYPTION STANDARD).....	24
6.3 PKCS (PUBLIC-KEY CRYPTOGRAPHY STANDARDS)	25
6.4 EESSI (EUROPEAN ELECTRONIC SIGNATURE STANDARDIZATION INITIATIVE)	26
6.5 DSS (DIGITAL SIGNATURE STANDARD)	27
7 PKI (PUBLIC KEY INFRASTRUCTURE)	28
8 ELEKTRONICKÝ PODPIS	29
II PRAKTICKÁ ČASŤ	30
9 MATEMATICKÉ ZÁKLADY	31

9.1	MODULÁRNA ARITMETIKA	31
9.2	PRVOČÍSLA	31
9.3	NAJVĚČŠÍ SPOLOČNÝ DELITEL (GREATEST COMMON DIVISOR)	32
9.4	EUKLIDOV ALGORITMUS	32
9.5	MODULÁRNA INVERZIA	33
9.6	FERMATOVA VETA	33
9.7	EULEROVA FUNKCIA	34
9.8	ČÍNSKA VETA O ZVÝŠKOC	34
9.9	KVADRATICKÉ REZÍDUÁ A NEREZÍDUÁ.....	35
9.10	LEGENDREOVA FUNKCIA	35
9.11	JACOBIHO FUNKCIA	35
9.12	BLUMOVE ČÍSLA.....	36
9.13	DISKRÉTNĚ LOGARITMY	36
9.14	GENEROVANIE PRVOČÍSEL.....	36
9.15	ÚLOHA FAKTORIZÁCIE	37
10	MODERNÉ ALGORITMY.....	38
10.1	SYMETRICKÁ KRYPTOGRAFIA.....	38
10.1.1	Základné režimy činnosti blokových šifier:	39
10.1.2	Algoritmy Feistelovho typu	42
10.1.3	DES (Data Encryption Standard)	44
10.1.4	Triple DES	46
10.1.5	IDEA (International Data Encryption Algorithm)	47
10.1.6	AES (Rijndael)	49
10.2	ASYMETRICKÁ KRYPTOGRAFIA	52
10.2.1	Batožinový algoritmus (knapsack).....	52
10.2.2	RSA	53
10.2.3	ElGamal.....	55
10.2.4	Eliptické krivky	56
10.3	HAŠOVACIE FUNKCIE	61
10.3.1	Narodeninový útok (Birthday Attack)	63
10.3.2	MD (Message-Digest Algorithm)	63
10.3.3	SHA (Secure Hash Algorithm)	64
10.3.4	RIPEMD (RACE Integrity Primitives Evaluation Message Digest)	66
10.4	MAC (MESSAGE AUTHENTICATION CODE)	66
11	APLIKOVANÁ KRYPTOLÓGIA	69
11.1	DIGITÁLNE PODPISY	69
11.1.1	RSA schéma	70
11.1.2	ElGamal schéma.....	70
11.1.3	DSA (Digital Signature Algorithm)	71
11.1.4	ECDSA (Elliptic Curve Digital Signature Algorithm)	73
11.2	KRYPTOGRAFICKÉ PROTOKOLY.....	74
11.2.1	Shamirov protokol (Shamir's three-pass protocol)	74
11.2.2	Diffie-Hellman schéma	75
11.2.3	ECDH.....	76

11.2.4	ECMVQ	77
11.3	PGP (PRETTY GOOD PRIVACY)	78
11.4	IDENTIFIKÁCIA A AUTENTIZÁCIA ENTÍT	82
11.4.1	Autentizácia heslom	83
11.4.2	Protokol typu výzva - odpoveď	85
11.4.3	Needham-Schroeder protokol	86
11.4.4	Kerberos	88
11.5	INFORMAČNÁ A SIEŤOVÁ BEZPEČNOSŤ	90
11.5.1	TLS/ SSL	90
11.5.2	HTTPS	94
11.5.3	SSH	95
11.5.4	WEP	99
11.5.5	WPA	100
ZÁVER		102
CONCLUSION		103
ZOZNAM POUŽITEJ LITERATÚRY		104
ZOZNAM POUŽITÝCH SYMBOLOV A SKRATIEK		108
ZOZNAM OBRÁZKOV		111
ZOZNAM TABULIEK		113
ZOZNAM PRÍLOH		114

ÚVOD

Ľudia mali od nepamäti potrebu ukrývať informácie pred nepovolanými osobami. Či už sa jednalo o tajnú informáciu v podobe vojenskej správy, alebo milostný list, šifrovanie dávalo ľuďom nádej, že ich dôverné informácie bude čítať len osoba, pre ktorú boli určené. Kryptológia teda predstavuje vedu, ktorou sa ľudia zaoberali už v staroveku.

Tak ako vývin ľudstva, aj spôsoby šifrovania prešli postupným zdokonaľovaním. Rýchlosť kryptografického vývoja priamo určovala schopnosť ľudí dešifrovať utajované správy. Kryptológia teda prešla od jednoduchých šifrovaných správ v podobe jednoduchých substitučných šifier až po moderné sofistikované spôsoby zabezpečenia, aké poskytuje napr. biometria.

Dnešné technológie majú však aj obrátenú stranu, akou je počítačová kriminalita, ktorá neraz spôsobila veľké škody v podobe úniku informácií, stratách na majetku, dokonca aj na životoch. Aby sme mohli predchádzať týmto nežiaducim situáciám, je potrebné vedieť, aké zabezpečenie nám poskytujú používané systémy. Podrobná kryptoanalýza dnes teda zohráva nemenej dôležitú úlohu, ako samotná kryptografia.

Táto práca je zameraná predovšetkým na kryptografiu, teda spôsoby šifrovania a ich využitie. Teoretická časť obsahuje niektoré vybrané kryptologické oblasti ako historický vývin, prvky informačnej bezpečnosti, spôsoby kryptoanalýzy, prehľad kryptografických štandardov a infraštruktúru.

Praktická časť objasňuje matematické kryptografické pozadie, podrobne rozoberá najpoužívanejšie kryptografické algoritmy a rôzne spôsoby ich využitia v moderných informačných systémoch.

Súčasťou práce je webová prezentácia, ktorá má za úlohu pomôcť študentom pochopiť základné kryptografické prostriedky v rozsahu tejto bakalárskej práce.

I. TEORETICKÁ ČASŤ

1 ZÁKLADNÉ POJMY

Kryptológia je samostatná vedná disciplína, ktorá sa rozdeľuje na *kryptografiu* a *kryptoanalýzu*, niekedy sa pridáva aj *steganografia*. Kryptológia zahŕňa tvorbu kryptografických techník, vymedzuje podmienky pre jej praktické využívanie a skúmanie odolnosti kryptografických algoritmov proti útokom. Využíva rozsiahly matematický aparát, teóriu informácie, teóriu zložitosti, teóriu čísel a teóriu pravdepodobnosti. [1]

Kryptografia je oblasť kryptológie, ktorá sa zaoberá matematickými metódami pre skúmanie a navrhovanie šifrovacích systémov, ktoré budú spĺňať podmienky ako je dôvernosť, integrita dát, autentizácia entít a nepopierateľnosť dát. [1]

Kryptoanalýza sa zaoberá analýzou odolnosti kryptografického systému a metódami, ktoré vedú k prelomeniu kryptografického systému. Je v podstate „opakom“ kryptografie a v klasickej podobe sa snaží o získanie pôvodnej informácie, alebo aspoň jej časti – teda prelomiť šifrovací algoritmus. [1]

Steganografia je tá časť kryptológie, ktorej úlohou je skryť existenciu správy, ale samotná správa nemusí byť šifrovaná. V súčasnosti steganografia využíva na ukrývanie správ obrázky, zvukové nahrávky, alebo rôzne texty. [1]

Steganografiu je vhodné kombinovať s kryptografiou pre väčšiu bezpečnosť prenášanej správy.

Kryptografický systém je akýkoľvek systém, pomocou ktorého je možné transformovať otvorený (nechránený) text na šifrovaný (chránený) text. Všetky transformácie sú vykonané pomocou príslušného kryptografického algoritmu s použitím kľúčov. [1]

Kryptografické protokoly určujú postup, ako využiť celý potenciál šifrovacieho algoritmu, teda ako najvhodnejšie previesť šifrovanie, vrátane prenosu správy a výmeny kľúčov.

Šifrovanie je spôsob spracovania správy, pri ktorom použijeme kryptografický systém na transformáciu tejto správy.

Dešifrovanie predstavuje rekonštrukciu pôvodného otvoreného textu zo šifrovanej správy pri ktorej použijeme vopred dohodnutý kryptografický algoritmus a šifrovací kľúč.

Šifrovaný (chránený) text je výsledok spracovania textu pomocou šifrovania.

Otvorený (nechránený) text je pôvodný nezašifrovaný text.

Abeceda (otvoreného) textu je akýkoľvek alfabetický, numerický, alebo interpunkčný znak vyskytujúci sa v otvorenom texte.

Šifrovaná abeceda je tvorená abecedou otvoreného textu alebo inými obrazcami. [2]

Šifry a kódy sú dva rôzne spôsoby úpravy textu. Pomocou šifier sa utajuje obsah správy, pomocou kódu sa správa upraví do takej podoby, aby ju bolo možné preniesť nejakým prenosným kanálom.

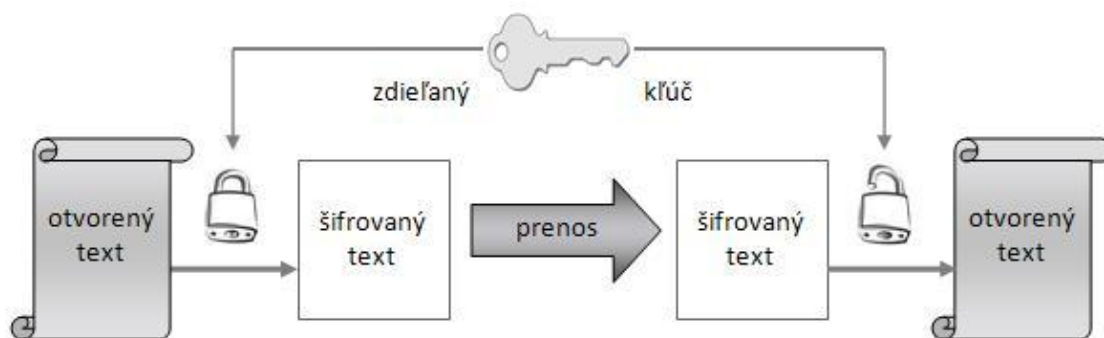
Kryptografický alebo šifrovací kľúč je voliteľný prvok kryptografického systému, ktorý sa používa na šifrovanie a dešifrovanie správy. Pomocou kľúča je šifrovaný text dešifrovateľný, preto musí byť kľúč dobre chránený.

Kľúčový priestor tvorí počet rôznych kľúčov pri určitom šifrovacom algoritme.

2 KRYPTOGRAFICKÉ ZÁKLADY

2.1 Symetrická kryptografia

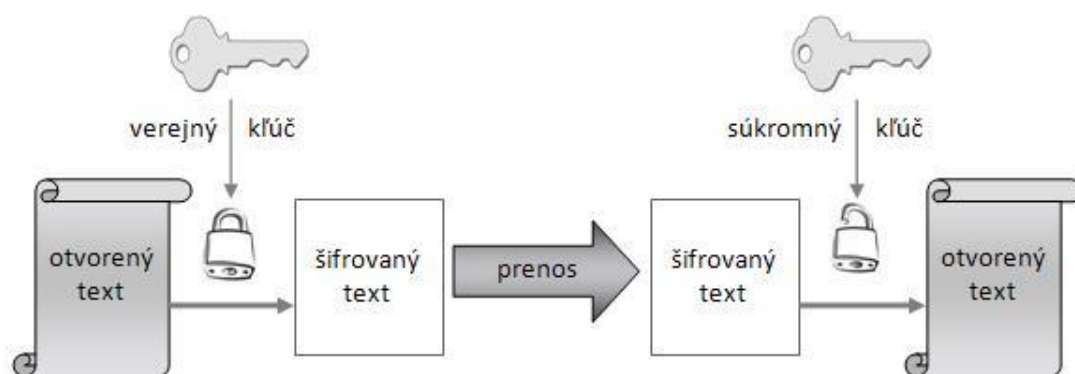
Pre šifrovanie aj dešifrovanie sa používa rovnaký kľúč. Proces šifrovania a dešifrovania je rovnaký, len vzájomne opačný. Na symetrickom šifrovaní bola založená väčšina klasických šifíer, ale patria sem aj moderné šifrovacie algoritmy – napr. DES, IDEA, AES.



Obr. 1. Princíp symetrického šifrovania

2.2 Asymetrická kryptografia

Základ asymetrickej kryptografie tvoria dve funkcie – jedna pre šifrovanie, druhá pre dešifrovanie. Najčastejšie sa využíva tzv. asymetrická kryptografia s verejným kľúčom. Je založená na používaní dvoch rôznych kľúčov, pričom nie je možné z jedného odvodiť druhý. Základom je jednosmerná funkcia, ktorou sa zašifruje správa za pomoci verejného kľúča, ale s rovnakým kľúčom sa nedá dešifrovať. Na dešifrovanie je potrebný súkromný kľúč, ktorý má príjemca správy. Jednotlivé algoritmy používajú na šifrovanie rozličný matematický aparát.



Obr. 2. Princíp asymetrického šifrovania

2.3 Hybridná kryptografia

Hybridná kryptografia spája symetrické a asymetrické kryptografické systémy, čo rieši nevýhody oboch systémov. Symetrické šifrovanie je niekoľkonásobne rýchlejšie, ale pri jeho použití vznikajú problémy pri kľúčovom manažmente, predovšetkým v rozsiahlejších sieťach. V hybridnom kryptosystéme sa využíva symetrický systém s náhodným kľúčom, ktorým sa zašifruje správa a asymetrický systém je použitý na zašifrovanie a prenos daného kľúča.

2.4 Zásady a pravidlá kryptológie

S vývojom ľudstva sa menili požiadavky na techniky šifrovania používané v kryptológii. Postupne sa vznikali nové pravidlá šifrovania a bol vyvíjaný tlak na kvalitu šifrovacieho systému. Medzi najčastejšie požiadavky patrí:

- spoľahlivosť kryptosystému
- čo najmenšia dĺžka kľúča
- primeraná zložitosť
- odolnosť kryptosystému voči šíreniu chyby
- primerané predlžovanie textu pri šifrovaní
- neprelomiteľnosť kryptosystému v reálnom čase
- množstvo vynaloženej práce pri šifrovaní a dešifrovaní musí zodpovedať požadovanému stupňu utajenia
- šifrovací algoritmus musí byť bez zbytočných obmedzení
- čo najjednoduchšia implementácia šifrovacieho algoritmu [1]

Bezpečnosť šifrovacích algoritmov je posudzovaná podľa:

- ceny potrebnej k prelomeniu kryptosystému ku cene dát
- doby aktuálnosti dát
- celkovému množstvu zašifrovaných dát pomocou jedného kľúča

Podmienky pre zabezpečenie silného kryptosystému:

- bezpečnosť je založená na utajení kľúča
- kryptosystém musí byť realizovateľný nad dostatočne veľkým priestorom kľúčov
- šifrovaný text musí mať náhodnú štruktúru voči všetkým statickým testom
- kryptosystém musí byť odolný voči všetkým známym kryptoanalytickým útokom
- kryptosystém musí mať dostatočnú rezervu odolnosti proti útoku hrubou silou [1]

2.5 Typológia algoritmov

Šifrovacie algoritmy majú rôzne charakteristické vlastnosti, pre niektoré sú dokonca typické viaceré rôzne vlastnosti. Základná typológia šifrovacích algoritmov:

- **podľa množstva šifrovaného textu:**
 - bloková šifra – spracováva celé bloky (reťazce) textu
 - prúdová šifra – spracováva jednotlivé znaky textu
- **podľa nutnosti utajenia kľúčov:**
 - privátny kľúč
 - verejný kľúč
- **podľa počtu kryptografických kľúčov:**
 - spoločný kľúč pre šifrovanie aj dešifrovanie
 - rôzny kľúč pre šifrovanie a dešifrovanie
- **podľa možnosti rekonštruovať pôvodný text:**
 - šifrovanie – otvorený text je možné rekonštruovať
 - hašovanie – otvorený text nie je možné rekonštruovať [1]

3 HISTÓRIA A VÝVIN

3.1 Staroveká kryptológia

Počiatky kryptológie siahajú až niekde do obdobia pred 4000 rokmi. Prvé záznamy pochádzajú zo starého Egypta, odkiaľ sa dochovali záznamy s atypickými hieroglyfmi. Pomocou nich bolo zaistené, že len vybraná skupina ľudí mala prístup k šifrovaným informáciám.

Okolo roku 1500 pred n. l. boli v starovekej Mezopotámii použité jednoduché kryptografické systémy založené na zámene jedného klinového písma za iné, s rovnakou zvukovou hodnotou.

Okolo roku 1000 pred n. l. používali starí Číňania kód, ktorý prirad'oval správam verše z určenej básne.

Hebrejci vynali a používali prvú šifru s názvom *Atbash* okolo roku 500 pred n. l. Je to reverzná substitučná šifra, kde sa prvé písmeno abecedy nahradí posledným, druhé predposledným atď. Zmienka o tejto šifre sa nachádza aj v Biblii.

V starovekom Grécku vznikli šifrovacie kľúče, ktoré boli rozdelené do dvoch skupín – transpozičné a substitučné. Plutarchos zdokumentoval prvý transpozičný šifrovací systém nazývaný *Skytalé*. Jeho princíp spočíval v tom, že sa na palicu navinula páska, na ktorú sa pozdĺžne vypisoval text. Dešifrovateľ musel mať rovnako hrubú palicu, inak bola text na páske len nečitateľným zhlukom znakov.



Obr. 3. Skytalé

Veľký objav učinil grécky historik Polybios svojou šifrovacou tabuľkou – tzv. *Polybiov štvorec*. Spočíva v umiestnení abecedy do štvorca, pričom každý znak reprezentuje dvojica čísel. Polybiov štvorec sa stal na dlhú dobu základom pre mnohé šifrovacie algoritmy. [1]

Medzi dôležité šifrovacie algoritmy staroveku patrí aj *Caesarova šifra*, ktorá reprezentuje klasický substitučný systém. Princíp spočíval v posunutí znakov abecedy o tri vpred, neskôr sa používal premenlivý počet posunutí v abecede.

3.2 Stredoveká kryptológia

Pod významné kryptologické systémy v stredoveku sa podpísali predovšetkým arabskí učitelia. Využívali rôzne substitučné šifrovacie metódy a položili prvé základy kryptoanalýzy, kde používali frekvenčnú analýzu.

V Európe sa začala kryptografia rozvíjať o niečo neskôr. Priekopníkom kryptografie bol benediktínsky mních Johannes Trithem, ktorý napísal približne v roku 1499 prvú významnú knihu o šifrovaní.

V tomto storočí sa začali objavovať prví kryptoanalytici a kryptoanalýza zohrávala čoraz dôležitejšiu úlohu v dejinách ľudstva. Medzi významné šifrovacie metódy stredovekej kryptológie patrí *Cardanova mriežka*, *bigramova substitúcia*, či *Morseova abeceda*.

3.3 20. storočie

Začiatkom prvej svetovej vojny nabrala kryptológia rýchly spád. Bezdrôtový prenos správ si vyžadoval dôkladné zabezpečenie proti prelomeniu šifier, kryptoanalýza preukázala svoju silu pri smerovaní udalostí. Samotný vstup USA do vojny bol následkom rozlúštenia obsahu šifrovaného telegramu, dnes známeho ako Zimmermannov telegram.

William Frederic Friedman sa stal jedným z veľikánov kryptológie dvadsiateho storočia. V roku 1923 vydal významné štvorzväzkové dielo *Základy kryptoanalýzy*. [1]

Gilbert S. Vernam vymyslel polyalfabetický šifrovací stroj, ktorý využíval náhodný neopakujúci sa kód. Vernamova šifra je dnes považovaná za jediný neprelomiteľný a teda absolútne bezpečný šifrovací systém.

Azda najznámejším šifrovacím strojom sa stala legendárna *Enigma*. Mala viacero prevedení a svoje využitie našla počas druhej svetovej vojny, keď ju používali nemeckí vojaci na šifrovanie tajných správ. Poľským matematikom sa neskôr podarilo túto šifru prelomiť, čo podľa historikov významne ovplyvnilo vývin udalostí druhej svetovej vojny.

Anglický inžinier Tommy Flowers zostrojil v roku 1943 jeden z prvých počítačov sveta Colossus, ktorý bol určený pre lúštenie nemeckých šifier. S ďalším vývojom informačných technológií bol priamo určovaný aj vývoj kryptológie.

3.4 Moderná kryptológia

Obdobie modernej kryptológie sa začína datovať v sedemdesiatych rokoch dvadsiateho storočia, keď bol objavený postup kryptosystému s verejným kľúčom, nazývaným aj ako asymetrická kryptografia. V roku 1976 publikovali W. Diffie a M. Hellman dielo s názvom *New directions in cryptography*, ktoré obsahuje myšlienku šifrovania s verejným kľúčom, ktoré je založené na existencii dvoch kľúčov – verejného a privátneho, pričom ani jeden sa nedá z druhého odvodiť.

Čoskoro sa objavil prvý šifrovací systém založený na tomto princípe šifrovania a dostal názov RSA podľa svojich objaviteľoch – R. L. Rivest, A. Shamir a L. M. Adelman. Systém RSA je založený na faktorizácii veľkých čísel, čo spôsobuje veľkú výpočtovú zložitosť, preto sa tento systém napriek všetkým svojim výhodám po svojom objave moc nepresadzoval. Využíval sa hybridný šifrovací systém, ktorý používal verejný kľúč len k distribúcii symetrických kľúčov.

Koncom 20. storočia dochádza k masovému rozvoju v oblasti informačných technológií, čím sa stáva kryptológia nevyhnutnou súčasťou nášho života.

4 KRYPTOANALÝZA A ÚTOKY PROTI ŠIFRÁM

Okrem Veranamovej šifry sú všetky doposiaľ známe šifrovacie algoritmy za určitých podmienok prelomiteľné. Pomocou podrobnej kryptoanalýzy vieme odhadnúť možnosť prelomenia šifry pri známych spôsoboch útokov.

Kryptoanalýzu zjednodušujú tieto znalosti:

- znalosť otvoreného textu – napr. jazyk textu, pravdepodobnosť obsahu textu,...
- redundantné informácie v otvorenom texte – štruktúra slov, interpunkcia, redundancia jazyka
- znalosti použitého šifrovacieho algoritmu – dĺžka kľúča, jeho štruktúra, doba šifrovania, spôsob prenosu tajného kľúča a i.
- porušenie kryptografického protokolu – napr. nevhodný spôsob podpisovania súkromným kľúčom, opakované použitie hesla
- ďalšie znalosti o šifre – dostupnosť verejného kľúča a i. [1]

Podľa znalosti textu delíme útoky:

- znalosť zašifrovaného textu (ciphertext only attack) – útočník pozná iba šifrovaný text
- čiastočná znalosť otvoreného textu (probably plaintext attack) – útočník pozná okolnosti odosielania správy, ktoré mu môžu pomôcť k čiastočnému odhadu správy
- znalosť otvoreného textu (known plaintext attack) – útočník pozná k otvorenému textu aj šifrovaný text
- znalosť zvoleného otvoreného textu (chosen plaintext attack) – útočník si môže zvoliť k otvorenému textu šifrovaný text
- znalosť zvoleného šifrovaného textu (chosen ciphertext attack) – útočník si môže zvoliť šifrované texty, ku ktorým získa otvorené texty

4.1 Útoky proti šifráM

Tieto útoky sa od seba odlišujú počiatočnými podmienkami a znalosťami, ktorými útočník disponuje. Medzi metódy útoku proti klasickým šifráM patrí napr. frekvenčná analýza, metóda koeficientu koincidencie, Kasiskiho metóda.

Moderná kryptológia je oveľa odolnejšia voči útokom a stojí predovšetkým na silných matematických základoch, ktoré bez dodatočných znalostí o použitom kryptografickom systéme nie je väčšinou možné prelomiť inak než útokom hrubou silou.

4.2 Útok hrubou silou (Brute Force Attack)

Tento typ útoku spočíva v preskúšaní celého kľúčového priestoru, až kým sa nenájde ten správny kľúč. Dĺžka kľúča určuje realizovateľnosť útoku, pretože so zvyšujúcou sa dĺžkou kľúča rastie možnosť prelomenia šifry exponenciálne. Rýchlosť útoku sa zvyšuje pri využití nejakých známych skutočností, ktoré pomôžu útočníkovi zmenšiť kľúčový priestor.

Vývoj informačných technológií umožňuje prelomenie čoraz silnejších šifrovacích algoritmov. S nasadením veľkej výpočtovej techniky boli prelomené aj také algoritmy, ktorých dešifrovacia doba sa odhadovala na milióny rokov. V roku 1994 bola pomocou Internetu roznásobená šifra RSA-129 a o tri roky neskôr aj 56 bitový kľúč DES.

4.3 Niektoré špeciálne druhy útokov

- diferenciálna analýza
- kolízia kľúčov
- náhodné chyby (random faults)
- narodeninový útok (birthday attack)
- Man-in-the-Middle attack (MITM)
- časový útok (timing attack)
- útok postrannými kanálmi (side channel attack)

5 INFORMAČNÁ BEZPEČNOSŤ

„Informačná bezpečnosť sa definuje ako schopnosť siete alebo informačného systému ako celku odolať s určitou úrovňou spoľahlivosti náhodným udalostiam, alebo nezákonnému, alebo zákernému konaniu, ktoré ohrozuje dostupnosť, pravosť, integritu a dôvernosť uchovávaných alebo prenášaných údajov a súvisiacich služieb poskytovaných alebo prístupných prostredníctvom týchto sietí a systémov.“ [3]

5.1 História počítačovej kriminality

95% všetkých počítačových zločinov bolo odhalených náhodou, nie ako výsledok kontrolnej činnosti. [4]

60. a 70. roky

V USA od roku 1958 zaistoval Stanford Research Institute (SRI) zber údajov o zneužívaní počítačov. Tieto boli rozdelené do 4 kategórií:

- vandalizmus proti hardvéru
- krádež majetku alebo informácií
- podvod alebo krádež peňazí uskutočnený pomocou počítača
- neprístupné použitie počítača alebo krádež a predaj počítačového času [4]

Okrem peňažných podvodov, pri ktorých si páchatelia prevádzali peniaze na svoj účet, sa objavujú prvé popisy o technológii zničenia počítača a prípady magnetického vymazávania, či elektronického monitorovania.

80. roky

V tomto období dochádza okrem peňažných podvodov aj ku krádežiam databáz, šíreniu počítačových vírusov, rozširovaniu pirátskeho softwaru. Najrozšírenejším typom počítačovej kriminality sa stáva krádež softwaru nelegálnym kopírovaním. [4]

90. roky

Začiatkom 90. rokov dochádza k počítačovej kriminalite v rôznych podobách – krádeže peňazí, informácií a programov, krádeže služieb, zmena údajov, poškodenie softwaru a nepovolené vstupy. Pomocou Internetu sa začína celosvetovo rozširovať pornografia a rasizmus, dochádza k propagácii drog a výbušnín, či prezentácii extrémistov. [4]

5.2 Bezpečnosť v informačných technológiách

Pod informačnú bezpečnosť spadajú dve oblasti:

- *počítačová bezpečnosť* - zaoberá sa predovšetkým bezpečnou prevádzkou počítačov a ochranou dát pri ich spracovaní, pričom pojem počítač označuje zariadenie, ktoré obsahuje procesor a pamäť
- *sieťová bezpečnosť* - rieši otázku bezpečného pripojenia počítačov do telekomunikačných sietí

5.3 Základné princípy informačnej bezpečnosti:

- *dôvernosť* (confidentiality) – informácie sú prístupné iba oprávneným osobám
- *integrita* (integrity) – informácie môžu byť upravené iba predpísaným spôsobom a úpravy môže vykonávať len oprávnená osoba
- *dostupnosť* (availability) – informácie musia byť dostupné oprávneným osobám kedykoľvek v prípade potreby
- *autentickosť* (authenticity) – možnosť overenia identity oprávnených osôb
- *nepopierateľnosť* (non-repudiation) – nie je možné poprieť pôvod dát

5.4 Útoky na informačné systémy:

- *hardware* – prírodné katastrofy, úmyselné a neúmyselné deštruktívne činnosti, ...
- *software* – krádež či modifikácia programu, vymazanie softwaru, vírusy, trojské kone, počítačové červy, logické bomby...
- *údaje* – prerušenie, odpočúvanie, modifikácia, falšovanie údajov, ...[4]

5.5 Prostriedky sieťovej bezpečnosti

- šifrovanie – využitie kryptografických systémov a hašovacích funkcií
- digitálne podpisy a certifikáty – pre zaistenie autentifikácie, integrity a nepopierateľnosti dokumentov
- riadené prístupy – použitie šifrovacích aplikácií a zariadení na prístup do nechránenej siete
- bezpečnosť elektronickej pošty – PEM, PGP, S/MIME,...
- bezpečnosť Web systému – použitie bezpečnostných protokolov
- firewall – ochrana prístupu do siete [4]

6 KRYPTOGRAFICKÉ ŠTANDARDY

6.1 DES (Data Encryption Standard)

V roku 1976 bol v USA prijatý DES ako šifrovací štandard pre zabezpečovanie dát v civilnom a vládnom sektore organizáciou NBS, dnes NIST. Šifrovací algoritmus DES je zdokonalený algoritmus Lucifer od firmy IBM, jeho autorom je Horst Feistel. DES je blokový algoritmus, ktorý pracuje so 64 bitovými blokmi a na transformáciu využíva 64 bitový šifrovací kľúč.

Už v čase jeho štandardizácie boli vyslovené pochybnosti o bezpečnosti tohto algoritmu kvôli relatívne krátkemu šifrovaciemu kľúču a v roku 1997 bol pomocou Internetu prelomený. Namiesto DES sa používal 3DES a v roku 2001 bol nahradený novým štandardom pod názvom AES.

6.2 AES (Advanced Encryption Standard)

Pre nový šifrovací štandard bolo vyhlásené v roku 1997 verejné výberové konanie inštitúciou NIST. Nový štandard mal po dvadsiatich rokoch nahradiť nepostačujúci štandard DES. Víťazný algoritmus mal niesť názov – AES a spĺňať určité podmienky – flexibilitu, ľahkú implementáciu, má pracovať s rôznymi typmi procesorov, vrátane 8 bitových. AES má byť symetrický blokový algoritmus o veľkosti 128 bitov a pracovať s kľúčmi 128, 192 a 256 bitov. [1]

Do súťaže bolo prihlásených 15 kandidátov – CAST-256, DEAL, Crypton, DFC, E2, Frog, HPC, LOKI97, MAGENTA, MARS, Rijndael, RC6, SAFER+, Serpent a Twofish.

Do druhého kola postúpilo 5 algoritmov – MARS, RC6, Rijndael, Serpent a Twofish. Hodnotené boli predovšetkým tieto vlastnosti – bezpečnosť, rýchlosť šifrovania, platformová nezávislosť, jednoduchá softwarová i hardwarová implementácia a i. [1]

Víťazom výberového konania sa stal algoritmus Rijndael, ktorý vyvinuli dvaja belgickí kryptológovia Joan Daemen a Vincent Rijmen. Algoritmus podporuje rôzne veľkosti šifrovaných blokov a kľúčov a na výpočet používa 10, 12 alebo 14 šifrovacích cyklov v závislosti od veľkosti kľúča. Algoritmus AES je postavený na silnom kryptografickom základe a odhaduje sa, že bude bezpečne využívaný ešte ďalších 20 až 30 rokov.

Tab. 1. Výsledné hodnotenie finalistov AES [5]

Algoritmus	MARS	RC6	Rijndael	Serpent	Twofish
Bezpečnosť	1.6	1.2	2.0	3.2	2.0
Hardvérová účinnosť	0.6	1.2	2.4	1.8	1.2
Softvérová účinnosť	0.4	0.6	1.8	0.2	1.2
Flexibilita	0.4	0.7	0.7	0.6	0.5
Celkové skóre	3.0	3.7	6.9	5.8	4.9
Poradie	5	4	1	2	3

6.3 PKCS (Public-Key Cryptography Standards)

PKCS je skupina štandardov pre implementáciu algoritmu RSA a neskôr aj iných kryptografických algoritmov s verejným kľúčom, vytvorená spoločnosťou RSA Data Security Inc. Prvý raz boli publikované v roku 1991 a od tej doby sú používané a stali sa základom pre mnohé iné štandardy. Popis štandardov je v nasledujúcej tabuľke (Tab. 2). PKCS #2 a PKCS #4 boli zahrnuté do PKCS #1.

V súčasnosti sú štandardy upravované tak, aby našli čo najširšie využitie. Napriek mnohým iným štandardom, sú PKCS najviac dominantné a preto sú kompatibilné s inými štandardmi.

Tab. 2. Prehľad štandardov PKCS

Číslo normy	Popis normy
PKCS #1	RSA kryptografický štandard
PKCS #3	Diffie-Hellman štandard pre dohodu na kľúči
PKCS #5	PBES kryptografický štandard založený na hesle
PKCS #6	Štandard pre syntax rozšírených certifikátov, ktorý zahŕňa certifikát X.509
PKCS #7	Štandard pre syntax kryptografickej správy
PKCS #8	Štandard pre syntax súkromného kľúča
PKCS #9	Štandard definujúci vybrané atribúty pre štandardy PKCS #6, PKCS #7, PKCS #8 a PKCS #10
PKCS #10	Štandard definujúci syntax pre žiadosť o pridelenie certifikátu verejného kľúča
PKCS #11	Štandard definujúci rozhrania pre kryptografické tokeny
PKCS #12	Štandard pre syntax výmeny osobných informácií
PKCS #13	Štandard definujúci kryptografiu eliptických kriviek
PKCS #14	Štandard pre generovanie pseudonáhodných čísiel
PKCS #15	Štandard pre kryptografické tokeny

6.4 EESSI (European Electronic Signature Standardization Initiative)

EESSI bola pracovná skupina vytvorená organizáciou ICTSB v roku 1999 pre koordináciu štandardizačnej činnosti na vytvorenie normy o elektronickom podpise v Európskej únii. Tieto normy bližšie určovali podmienky pre vytvorenie a overovanie elektronických podpisov, technické a bezpečnostné požiadavky, syntax elektronických podpisov a programovú implementáciu. Výsledky boli zverejnené v oficiálnom dokumente v júli 2003 a v októbri 2004 ukončila EESSI svoju činnosť. Vytvorené štandardy prešli pod správu ETSI (European Telecommunications Standards Institute) – organizáciu pre tvorbu noriem v EÚ.

6.5 DSS (Digital Signature Standard)

DSS je štandard americkej vlády, ktorý určuje postupy a metódy pre digitálny podpis. Bol navrhnutý organizáciou NIST v roku 1991 a v súčasnosti je špecifikovaný v dokumente FIPS PUB 186-3 z roku 2002.

Táto norma bližšie špecifikuje požiadavky pre získanie záruky potrebnej na overovanie a schvaľovanie digitálnych podpisov. V DSS sú definované 3 základné typy algoritmov:

- DSA (Digital Signature Algorithm)
- RSA Digital Signature Algorithm
- ECDSA (Elliptic Curve Digital Signature Algorithm)

7 PKI (PUBLIC KEY INFRASTRUCTURE)

PKI (infraštruktúra verejných kľúčov) predstavuje súhrn organizačných a technických opatrení spojených so správou a používaním certifikátov verejných kľúčov. PKI zaručuje bezpečnosť informačných systémov, elektronickej komunikácie a transakcií. Je založená na asymetrickej kryptografii. Infraštruktúra je postavená na certifikačnej autorite (CA), ktorá vystavuje digitálne certifikáty (PKC – Public Key Certificate) a svojou autoritou sa zaručuje za dôveryhodnosť údajov v certifikáte.

PKI má nasledovné vlastnosti:

- **autentifikácia** – overenie totožnosti je zabezpečené pomocou digitálneho certifikátu
- **nepopierateľnosť** – autor nemôže poprieť podpísané informácie
- **dôvernoscť** – k informáciám sa dostane iba autorizovaná osoba
- **integrita** – nie je možné modifikovať správu pri prenose, čo je zabezpečené hašovacími funkciami
- **riadenie prístupu** – k informáciám majú prístup len autorizované osoby

Jednou z PKI noriem je internetový certifikát X.509 opísaný dokumentoch RFC. Ďalšie dôležité PKI štandardy sú opísané v ISO, ANSI, PKCS a ETSI štandardoch. Základné PKI protokoly sú IPSec, SSL, TLS a S/MIME.

8 ELEKTRONICKÝ PODPIS

Moderné telekomunikačné prostriedky prenikli do života mnohých ľudí a ich používanie sa stalo v súčasnosti každodennou záležitosťou. Elektronická forma dokumentov značne uľahčuje prácu a urýchľuje vzájomnú komunikáciu medzi účastníkmi, je však potrebné zaručiť ich autentifikáciu, integritu a nepopierateľnosť. Tento problém rieši elektronický podpis, ktorý je vo svojej podstate digitálnou formou vlastnoručného podpisu. Tvorí súčasť elektronického dokumentu, alebo je s ním logicky spojený. Nemôže byť použitý pre podpísanie iného dokumentu a vytvoriť ho môže iba autorizovaná osoba. Korektnosť podpisu musí byť kýmkoľvek jedného a spoľahlivo overiteľná.

Elektronický podpis je založený na PKI technológii, teda dôveryhodnosť je zaručená prostredníctvom certifikátu vydaným CA, ktorá vstupuje do elektronickej komunikácie ako tretí dôveryhodný subjekt.

Formy elektronického podpisu:

- digitálny podpis
- naskenovaný podpis
- podpis získaný pomocou biometrických metód [1]

Z kryptologického hľadiska je digitálny podpis kryptografickou konštrukciou, ktorá využíva predovšetkým systémy s verejnými kľúčmi a kryptografické hašovacie systémy.

II. PRAKTICKÁ ČASŤ

9 MATEMATICKÉ ZÁKLADY

Moderná kryptológia využíva rozsiahly matematický aparát, ktorý patrí do *teórie čísel*. Spomenuté budú tie časti oblasti teórie čísel, ktoré sa využívajú v kryptografickom systéme.

9.1 Modulárna aritmetika

Modulárna aritmetika je časť matematiky, v ktorej sa pracuje nad množinou celých čísel. Využíva sa na redukciu veľkých čísel používaných pri výpočtoch. Pre označenie sa používa výraz **mod** a platí:

$$a \equiv b \pmod{n} \quad (1)$$

teda že a je **kongruentné** (zhodné) s b modulo n a znamená to, že a aj b dávajú po delení s n rovnaký zvyšok. Obecne toto tvrdenie platí vtedy, ak

$$a = b + kn \quad (2)$$

pre nejaké celé číslo k , teda rozdiel $a - b$ je deliteľný číslom n .

V modulárnej aritmetike platia komutatívne, asociatívne aj distributívne zákony a teda:

$$(a \pm b) \pmod{n} = ((a \pmod{n}) \pm (b \pmod{n})) \pmod{n} \quad (3)$$

$$(a * b) \pmod{n} = ((a \pmod{n}) * (b \pmod{n})) \pmod{n} \quad (4)$$

$$(a * (b + c)) \pmod{n} = (((a * b) \pmod{n}) * ((a * c) \pmod{n})) \pmod{n} \quad (5)$$

9.2 Prvočísla

Prvočíslo je celé číslo, ktoré je väčšie ako 1 a je deliteľné iba jednotkou a sebou samým. Prvočísla zohrávajú v kryptológii významnú úlohu, predovšetkým v algoritmoch s verejným kľúčom. Prvočísel je nekonečne veľa a platí **základná veta aritmetiky**:

Každé prirodzené číslo $m > 1$ sa dá jednoznačne napísať v tvare:

$$m = p_1^{\alpha_1} * p_2^{\alpha_2} * \dots * p_k^{\alpha_k} \quad (6)$$

kde p_1, p_2, \dots, p_k sú navzájom rôzne prvočísla a $\alpha_1, \alpha_2, \dots, \alpha_k$ sú prirodzené čísla.

Toto je zápis čísel v **kanonickom tvare**. [6]

9.3 Najväčší spoločný deliteľ (Greatest Common Divisor)

Najväčší spoločný deliteľ (GCD) dvoch prirodzených čísel m a n je najväčšie nenulové prirodzené číslo, ktoré je deliteľné oboma číslami m a n . Ak je najväčší spoločný deliteľ čísel m a n číslo 1, tieto sa nazývajú **nesúdeliteľné** a zapisujú sa v tvare:

$$GCD(m, n) = 1 \quad (7)$$

Pre každé celé číslo n platí:

$$GCD(n, n + 1) = 1 \quad (8)$$

9.4 Euklidov algoritmus

Euklidov algoritmus je jeden zo spôsobov, ako určiť spoločného deliteľa dvoch čísel. Je to najstarší netriviálny algoritmus, ktorý pochádza približne z roku 300 pred n. l. a dodnes sa používa.

Najväčší spoločný deliteľ dvoch čísel a, b je $d \in N$ také, že platí:

d delí a aj b , teda d je spoločný deliteľ, potom d delí tiež $a - (k * b)$ pre každé k .

Pre $x = GCD(a, b)$, pričom $a > b$ platí:

$$a = m * b + r \quad (9)$$

(teda $m = a/b$ so zvyškom r). Potom x delí a, b aj r , pričom $GCD(a, b) = GCD(b, r)$ za podmienky, že $a > b > r \geq 0$. Môžeme teda a a b nahradiť b a r , až kým nie je zvyšok po delení 0.

Algoritmus:

Vstup: dve prirodzené čísla a, b , pričom $a \geq b$

Výstup: najväčší spoločný deliteľ čísel a a b

1. Pokiaľ $b \neq 0$, vykonaj:

$$a = b, \quad b = r, \quad r = a \bmod b$$

2. Vráť hodnotu a . [6]

9.5 Modulárna inverzia

Označme \odot ako číselnú operáciu, napríklad násobenie $*$ alebo sčítanie $+$. Číslo i bude **identickým prvkom** pre \odot , ak bude platiť $x \odot i = x$ a $i \odot x = x$ pre každé x . Napríklad 1 je identickým prvkom pre násobenie, pretože $x * 1 = x$ a $1 * x = x$, číslo 0 je zas identickým prvkom pre operáciu sčítania. [6]

Nech i je identickým prvkom pre \odot . Číslo b bude **inverzným prvkom** čísla a pre operáciu \odot vtedy, ak $a \odot b = i$. Platí:

$$1 = (a * x) \bmod n \quad (10)$$

alebo

$$a^{-1} \equiv x \pmod{n} \quad (11)$$

pričom obecné platí, že rovnica (11) má jediné riešenie vtedy, ak sú čísla a a n nesúdeliteľné. [6]

Na výpočet inverzného čísla modula n sa najčastejšie používa rozšírený Euklidov algoritmus.

9.6 Fermatova veta

Pre každé prvočíslo m a každý prvok $a < m$ platí:

$$a^m \bmod m = a \quad (12)$$

alebo

$$a^{m-1} \bmod m = 1 \quad (13)$$

Ak a nesmie byť násobkom m , potom platí **malá Fermatova veta**:

$$a^m \equiv a \pmod{m} \quad (14)$$

alebo

$$a^{m-1} \equiv 1 \pmod{m} \quad (15)$$

[6]

9.7 Eulerova funkcia

Eulerova funkcia udáva počet prirodzených čísel a menších ako n a čísla a , n sú nesúdeliteľné, platí:

$$\phi(n) = \sum_{\substack{a \leq n \\ \text{GCD}(a,n)=1}} 1 \quad (16)$$

Z tejto definície vyplýva:

- $\phi(1) = 1$
- $\phi(n) = n - 1$, ak je n prvočíslo [6]

Ak $n = p * q$, pričom p a q sú rôzne prvočísla, tak platí:

$$\phi(n) = \phi(p * q) = (p - 1) * (q - 1) \quad (17)$$

Tieto prvočísla sa používajú v šifrovacom algoritme s verejným kľúčom – RSA. [6]

Eulerovské zovšeobecnenie malej Fermatovej vety:

$$a^{\phi(n)} \bmod n = 1 \quad (18)$$

a pre modulárnu inverziu platí:

$$a^{-1} = a^{\phi(n)-1} \bmod n \quad (19)$$

[6]

9.8 Čínska veta o zvyškoch

Nech m_1, \dots, m_k sú po dvoch nesúdeliteľné celé čísla a nech a_1, \dots, a_k sú prirodzené celé čísla, potom sústava kongruencií

$$\begin{aligned} x &\equiv a_1 \pmod{m_1} \\ x &\equiv a_2 \pmod{m_2} \\ &\dots \dots \dots \\ x &\equiv a_k \pmod{m_k} \end{aligned} \quad (20)$$

má jediné riešenie m_1, \dots, m_k , teda existuje práve jedno $x \in \{0, 1, \dots, m_1, \dots, m_n - 1\}$.

Nech p, q sú navzájom nesúdeliteľné celé čísla, tak pre ľubovoľné $x \in Z_{p \cdot q}$ platí:

$$x \equiv a \pmod{p \cdot q} \Leftrightarrow x \equiv a \pmod{p}, x \equiv a \pmod{q} \quad (21)$$

Čínska veta o zvyškoch sa využíva napr. pre urýchlenie šifrovania RSA algoritmu. [7]

9.9 Kvadratické rezíduá a nerezíduá

Ak p je prvočíslo a a je číslo z intervalu $0 < a < p$, potom ak platí:

$$x^2 \equiv a \pmod{p} \quad (22)$$

bude číslo a **kvadratickým rezíduom** pre nejaké x . Čísla a , ktoré túto podmienku nespĺňajú, sa nazývajú **kvadratické nerezíduá**. Pre nepárne p existuje $(p-1)/2$ kvadratických rezíduí modulo p a rovnaký počet kvadratických nerezíduí modulo p . [6]

9.10 Legendreova funkcia

Legendreova funkcia (legendreov symbol) je funkcia $\left(\frac{a}{p}\right)$ definovaná pre akékoľvek celé číslo a a prvočíslo $p > 2$ takto:

- $\left(\frac{a}{p}\right) = 0$ ak je a deliteľné p
- $\left(\frac{a}{p}\right) = 1$ ak je a kvadratické rezíduo modulo p
- $\left(\frac{a}{p}\right) = -1$ ak je a kvadratické nerezíduo modulo p

Pre určenie Legendreovej funkcie platí:

$$\left(\frac{a}{p}\right) = a^{(p-1)/2} \pmod{p} \quad (23)$$

9.11 Jacobiho funkcia

Jacobiho funkcia (Jacobiho symbol) predstavuje zovšeobecnenie Legendreovej funkcie. Je to funkcia dvoch celých čísel a a n , značená $\left(\frac{a}{n}\right)$, ktorá je definovaná pre všetky $a \geq n$ a pre všetky nepárne celé čísla n nasledovne:

$$\left(\frac{a}{n}\right) = \left(\frac{a}{p_1}\right)^{\alpha_1} \left(\frac{a}{p_2}\right)^{\alpha_2} \dots \left(\frac{a}{p_k}\right)^{\alpha_k} \quad (24)$$

9.12 Blumove čísla

Prirodzené číslo n sa nazýva **Blumove (celé) číslo**, ak $n = p \cdot q$, kde p a q sú prvočísla kongruentné s $3 \bmod 4$, teda $p \equiv 3 \bmod 4$ a $q \equiv 3 \bmod 4$.

Blumove čísla sa využívajú v generátoroch pseudonáhodných čísel. [6]

9.13 Diskrétny logaritmus

Úlohou diskrétného logaritmu je nájsť také celé číslo pre neznámu hodnotu x , aby platilo:

$$a^x \equiv y \pmod{n} \quad (25)$$

Riešenie diskrétného logaritmu je výpočtovo ťažká úloha, preto sa často využíva v asymetrickej kryptografii.

9.14 Generovanie prvočísel

Veľké prvočísla sa využívajú v mnohých kryptografických algoritmoch a protokoloch a je veľmi dôležité ich náhodné generovanie. V prvom kroku pri získavaní takéhoto čísla sa náhodne vyberie nepárne číslo n a potom sa overuje jeho prvočíselnosť niektorým so známych testov. Ak sa zistí, že n nie je prvočíslo, je možné zvoliť ďalšie náhodné číslo, alebo zvoliť najbližšie nepárne číslo a test zopakovať.

Na overovanie prvočíselnosti existuje niekoľko známych testov – Lehmanov test, Rabin-Millerov test, Solovay-Strassenov test, a i.)

Lehmanov test na prvočíselnosť čísla p :

1. náhodne vyberieme číslo $a < p$
2. vypočítame $a^{(p-1)/2} \pmod{p}$
3. ak $a^{(p-1)/2} \not\equiv \pm 1 \pmod{p}$, tak p určite nie je prvočíslo
4. ak $a^{(p-1)/2} \equiv \pm 1 \pmod{p}$, potom pravdepodobnosť, že p nie je prvočíslo, je vyššia ako 50%
5. ak tento test zopakujeme t -krát a test je vždy úspešný, potom pravdepodobnosť, že p nie je prvočíslo, je $1:2^t$ [6]

Rabin-Millerov test na prvočíselnosť čísla p :

1. vypočítame číslo b , ktoré určuje počet delení čísla $(p - 1)$ číslom 2
2. určíme číslo m , pre ktoré platí $p = 1 + 2^b \cdot m$

3. náhodne vyberieme číslo $a < p$
4. položíme $j = 0$ a $z = a^m \bmod p$
5. ak $z = 0$ alebo $z = p - 1$, p môže byť prvočíslo
6. pre $j > 0$ a $z = 1$, p nie je prvočíslo
7. pre $j = j + 1$, ak $j < b$ a $z \neq p - 3$, položí sa $z = z^2 \bmod p$ a urobíme návrat k predchádzajúcemu kroku. Ak $z = p - 1$, p môže byť prvočíslo.
8. ak $j = b$, $z \neq p - 1$, p nie je prvočíslo
9. ak tento test zopakujeme t -krát, pravdepodobnosť, že p nie je prvočíslo, je iba $1:4^t$ [6]

Silné prvočísla sú také prvočísla p a q , ktoré sú odolné voči faktorizácii ich súčinu a spĺňajú tieto podmienky:

- najväčší spoločný deliteľ čísel $(p - 1)$ a $(q - 1)$ musí byť malý
- čísla $(p - 1)$ a $(q - 1)$ musia mať veľké prvočinitele p' a q'
- čísla $(p' - 1)$ a $(q' - 1)$ by mali mať veľké prvočinitele
- čísla $(p + 1)$ a $(q + 1)$ by mali mať veľké prvočinitele
- čísla $(p - 1)/2$ a $(q - 1)/2$ majú byť prvočísla [1]

9.15 Úloha faktorizácie

V modernej kryptografii sa využíva šifrovanie kľúčov pomocou súčinu dvojice veľmi vysokých prvočísel. Opačnou úlohou je **faktorizácia** – teda rozklad celého čísla na súčin prvočísel. Je to veľmi ťažká úloha s exponenciálnou zložitosťou a na jej náročnosti je založená bezpečnosť mnohých šifrovacích algoritmov, napr. algoritmus RSA.

10 MODERNÉ ALGORITMY

V súčasnosti sa využívajú šifrovacie algoritmy s rôznymi matematickými postupmi. V tejto časti budú priblížené základné princípy modernej kryptografie a dôkladnejší popis niektorých moderných šifrovacích algoritmov.

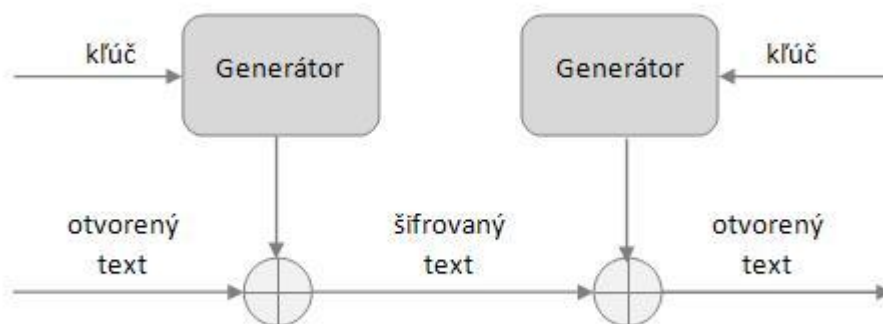
10.1 Symetrická kryptografia

Podľa spôsobu šifrovania sa delí symetrická kryptografia na *prúdové* a *blokové* šifry.

Prúdové šifry spracúvajú otvorený text po jednotlivých bitoch, prípadne po veľmi malých blokoch. Sú založené na aplikácii časovo premenlivej šifrovacej transformácie v závislosti od použitého prúdového kľúča. Tento môže byť generovaný náhodne, alebo pomocou algoritmu - generátora prúdového kľúča. [8]

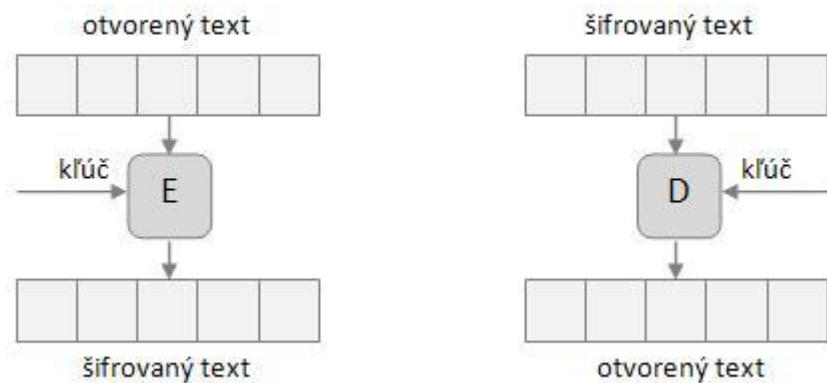
Prúdové šifry sú rýchlejšie než blokové a využívajú sa všade tam, kde je potrebné šifrovať dáta okamžite, teda nie je možné čakať na celý blok. V prípade straty alebo chybnéj rekonštrukcii jedného bitu sa táto chyba prejaví len v jednom znaku otvoreného textu, čo patrí medzi ďalšie výhody prúdových šifier.

Medzi algoritmy, ktoré využívajú prúdové šifry, patria: RC4, FISH, SEAL, Helix a i.



Obr. 4. Prúdová šifra

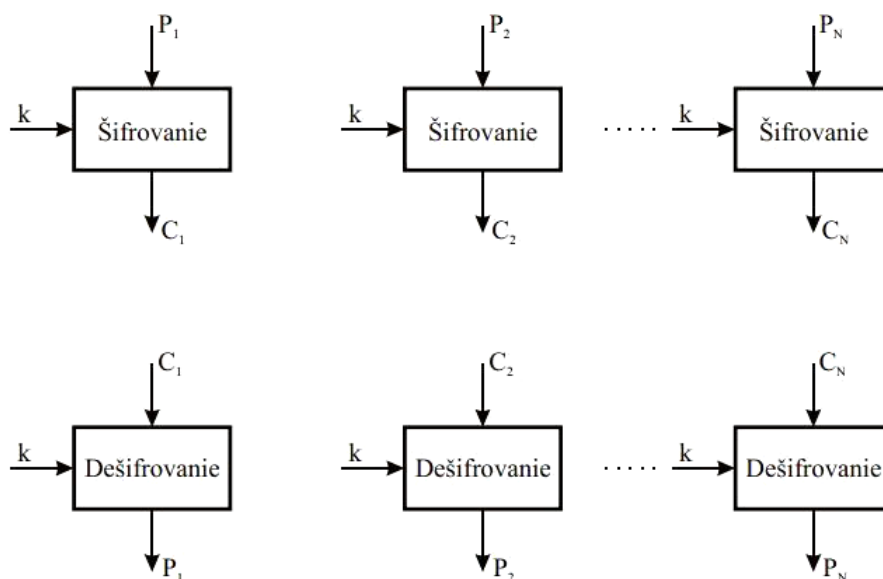
Blokové šifry rozdeľujú otvorený text na bloky s pevnou dĺžkou a spracovávajú jeden blok v jednom čase. Pomocou určeného algoritmu a šifrovacieho kľúča sú tieto bloky transformované na šifrovaný text. Dešifrovanie prebieha rovnako ako šifrovanie, iba sa zmení poradie kľúčov, prípadne šifrovacích operácií. [8]



Obr. 5. Bloková šifra

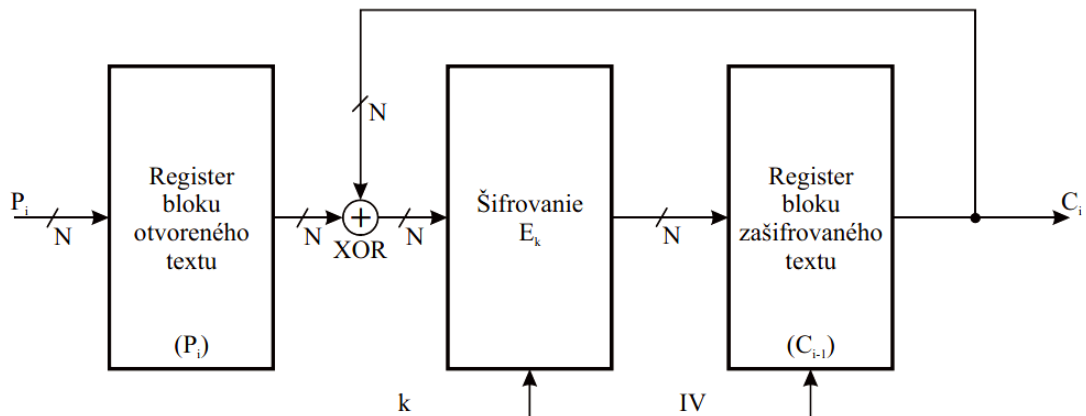
10.1.1 Základné režimy činnosti blokových šifier:

- **ECB (Electronic Code Book)** sa používa v prípade, ak je vstupný blok rozdelený na viac blokov a každý z nich je šifrovaný nezávisle od ostatných blokov. Tento režim podporuje využitie rozdielneho šifrovacieho kľúča pre každý blok a jednotlivé bloky je možné šifrovať v ľubovoľnom poradí. Medzi ďalšie výhody ECB režimu patrí možnosť paralelizácie pri šifrovaní a dešifrovaní. Pre bezpečnosť tohto režimu je potrebné zvoliť dostatočne veľkú dĺžku blokov. Nevýhodou tohto režimu je možnosť modifikácie obsahu správy bez znalosti kľúčov, ktoré nemusia byť na strane prijímateľa odhalené.



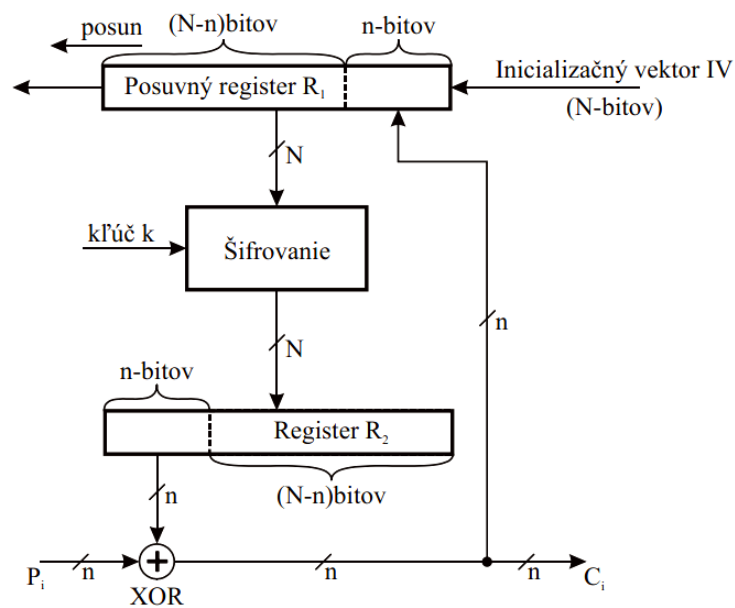
Obr. 6. Režim ECB [9]

- CBC (Cipher Block Chaining)** režim je založený na tom, že každý blok otvoreného textu je upravený pomocou operácie XOR s predchádzajúcim šifrovaným textom a až potom je zašifrovaný pre výstup. Pre jedinečnosť každého šifrovaného textu sa používa v prvom bloku inicializačný vektor. Dešifrovanie prebieha analogicky – dešifrovaný text je upravený operáciou XOR s predchádzajúcim. Medzi nevýhody tohto režimu patrí sekvenčnosť šifrovania, teda nie je možné využiť paralelizmus. Zmena jedného bitu v otvorenom texte ovplyvní všetky nasledujúce šifrované bloky. Pri dešifrovaní takáto chyba ovplyvní aktuálny blok a zodpovedajúci bit v ďalšom bloku otvoreného textu. Nakoľko blok otvoreného textu je možné získať z dvoch susedných blokov šifrovaného textu, je možné využiť paralelné spracovanie pri dešifrovaní.



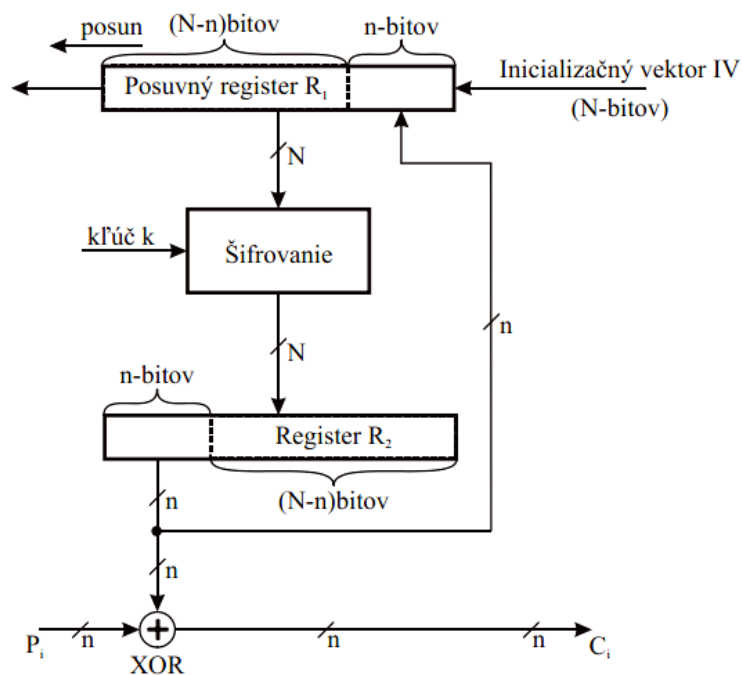
Obr. 7. Režim CBC [9]

- CFB (Cipher Feedback)** prevádza blokovú šifru na samo-synchronizačnú prúdovú šifru. Tento režim je podobný CBC, tiež využíva inicializačný vektor a operácie XOR. V tomto režime je predchádzajúci šifrovaný text najskôr zašifrovaný a potom XORovaný s blokom otvoreného textu. Pri dešifrovaní je možné využiť paralelizmus.



Obr. 9. Režim CFB [9]

- **OFB (Output Feedback)** sa podobá režimu CFB. Šifrovaný blok vznikne pomocou operácie XOR prevedenej medzi generátorom pseudonáhodných čísel a otvoreným textom, pričom spätná väzba je pred operáciou XOR. V tomto režime sa využíva inicializačný vektor, ktorého zmena pri rovnakom vstupnom bloku spôsobí zmenu zašifrovaného bloku.



Obr. 8. Režim OFB [9]

10.1.2 Algoritmy Feistelovho typu

Algoritmy Feistelovho typu umožňujú pomocou jednoduchých transformácií vytvoriť zložitý šifrovací systém. Používajú sa v mnohých kryptografických algoritmoch.

Feistelova šifra (Feistelova sieť) spočíva v tom, že sa vstupný blok otvoreného textu veľkosti n bitov rozdelí na dve rovnaké časti L_0 a R_0 . Potom nasleduje niekoľko rúnd, pričom pre každú rundu $i = 0, \dots, n$ platí:

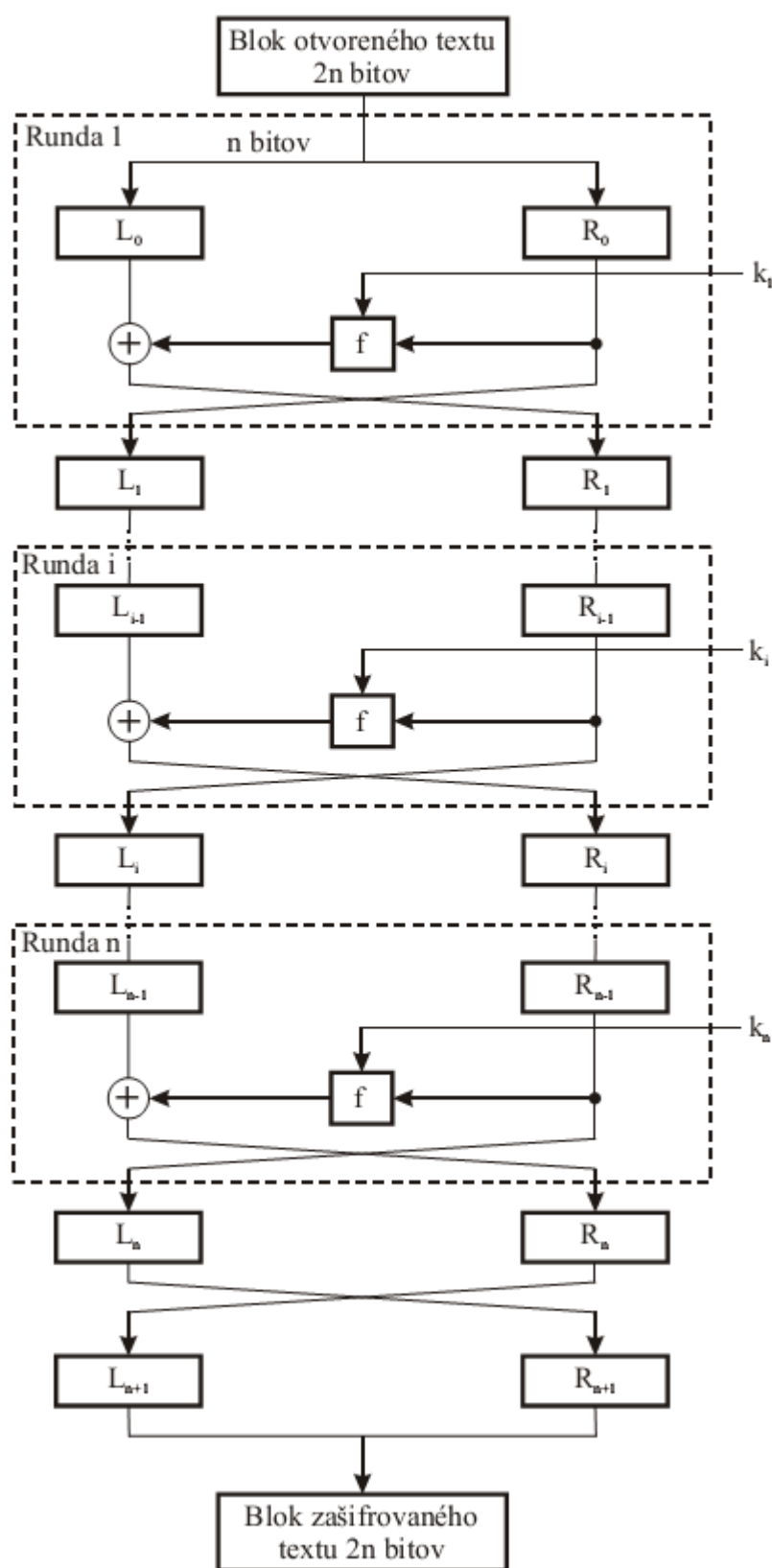
$$\begin{aligned} L_{i+1} &= R_i \\ R_{i+1} &= L_i \oplus f(R_i, K_i) \end{aligned} \tag{26}$$

kde K_i je subkľúč použitý v i -tej runde, f je rundová funkcia a \oplus je operácia XOR. Výsledný šifrovaný text je R_{n+1}, L_{n+1} . [1]

Na dešifrovanie sa používa rovnaká funkcia, len sa použijú subkľúče v opačnom poradí. Takže pre $i = n, \dots, 0$ platí:

$$\begin{aligned} R_i &= L_{i+1} \\ L_i &= R_{i+1} \oplus f(L_{i+1}, K_i) \end{aligned} \tag{27}$$

Existujú aj rôzne modifikácie Feistelovej šifry, napr. nepravidelné rozdelenie vstupného bloku, alebo sa mení rundová funkcia, ale využíva sa aj v asymetrickej kryptografii.



Obr. 10. Feistelova bloková šifra [9]

10.1.3 DES (Data Encryption Standard)

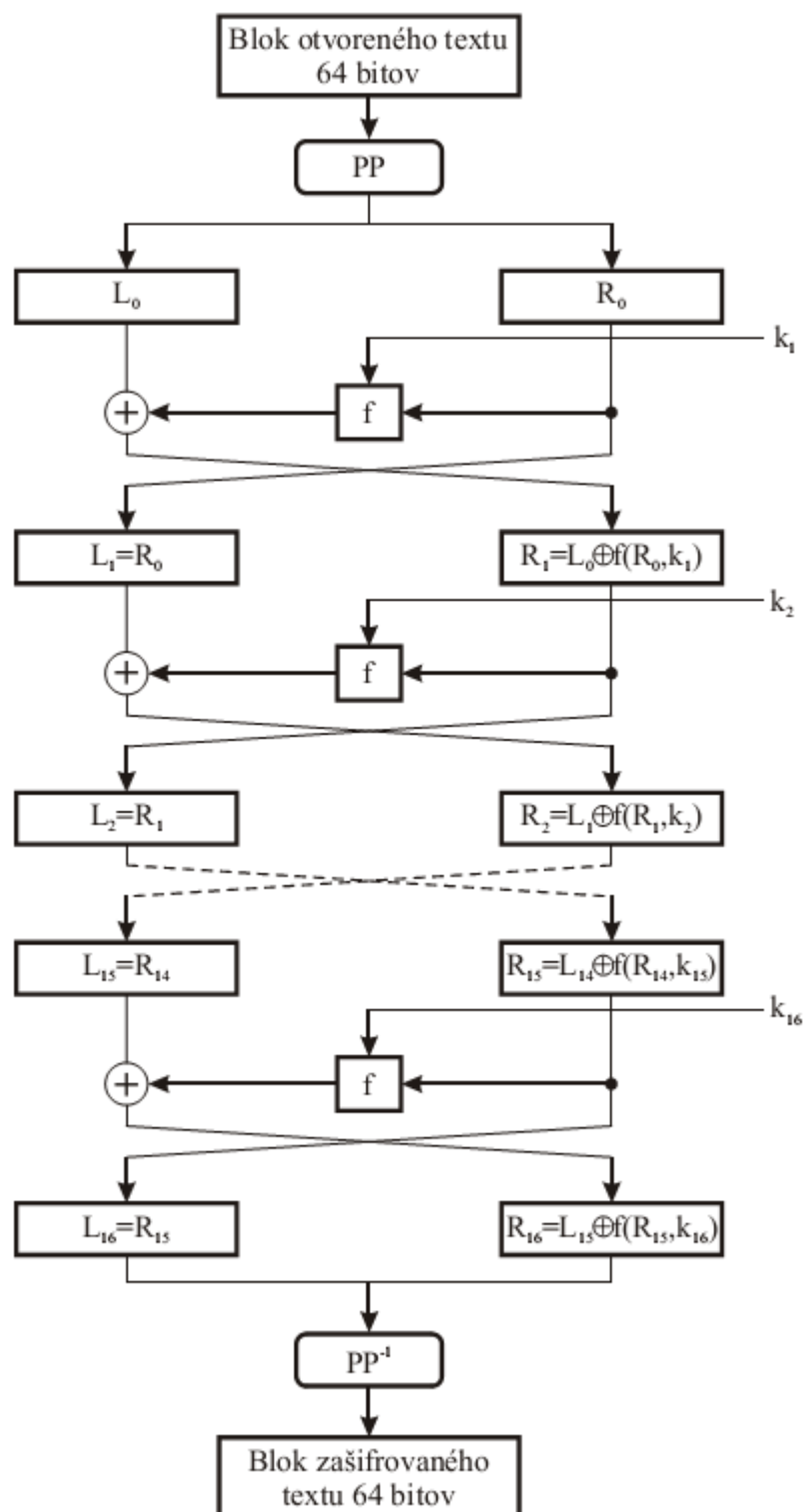
DES je blokový šifrovací algoritmus, ktorý bol v polovici 70-tych rokov vyvinutý organizáciou NBS a stal sa na viac ako 20 rokov štandardom pre blokové symetrické algoritmy. Je modifikáciou algoritmu Lucifer od firmy IBM a niekedy sa označuje aj ako DEA (Data Encryption Algorithm). Stal sa celosvetovo najrozšírenejším šifrovacím algoritmom.

Algoritmus DES pracuje s blokmi veľkosti 64 bitov – 56 bitov pre šifrovací kľúč a 8 bitov pre kontrolu parity. Pracuje v 16-tich cykloch podľa Feistelovej šifry a používa vstupnú a výstupnú permutáciu, ktoré súvisia s hardvérovou implementáciou. Algoritmus podporuje 4 režimy činnosti - ECB, CBC, CFB a OFB. Dešifrovanie je identický proces ako šifrovanie, len sa otočí poradie subkľúčov. [1]

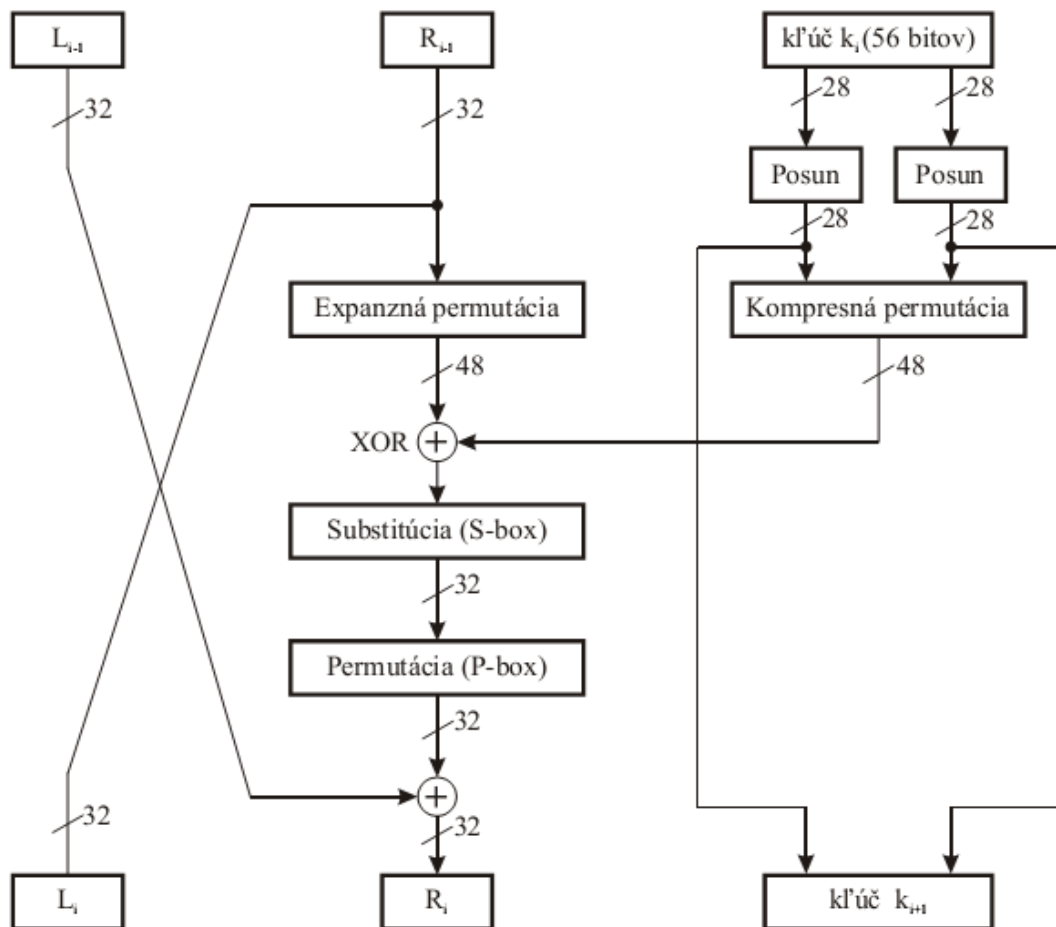
Postup šifrovania:

1. vstupná permutácia otvoreného textu
2. rozdelenie vstupného bloku na dve časti L_0 a R_0
3. 16 opakovaní rundovej funkcie
4. zlúčenie oboch subblokov a výstupná permutácia

Rundová funkcia pozostáva zo 4 krokov. V prvom kroku dochádza k expanzii subbloku na 48 bitov pomocou permutácie. V druhom kroku je výsledok spojený s podkľúčom pomocou operácie XOR. V ďalšom kroku je blok spracovaný v tzv. *S-boxe*, ktorý predstavuje základné jadro bezpečnosti algoritmu DES. V poslednom kroku dochádza k spracovaniu pomocou permutácie v tzv. *P-boxe*.



Obr. 11. Schéma šifrovania DES [9]



Obr. 12. Jedna runda DES [9]

Bezpečnosť šifry

Na prelomenie šifry sa úspešne používa lineárna aj diferenciálna kryptoanalýza a vzhľadom na to, že algoritmus využíva pomerne krátky šifrovací kľúč, je možné použiť aj útok hrubou silou. V roku 1997 bola šifra prelomená hrubou silou pomocou Internetu a v roku 1998 bol zostrojený DES cracker – stroj, ktorý dokázal odhaliť správny kľúč za 56 hodín skúšaním všetkých možných kľúčov. Na zvýšenie bezpečnosti šifry DES je možné použiť jej viacnásobnú aplikáciu.

10.1.4 Triple DES

Triple DES (3DES) je zosilnený variant algoritmu DES založený na jeho trojnásobnej aplikácii. Tento algoritmus využíva dvojnásobne alebo trojnásobne dlhší šifrovací kľúč, teda 112 bitov alebo 168 bitov. Pre zosilnenie šifrovania sa používa variant EDE

(Encrypt-Decrypt-Encrypt), kde sa v prvom a treťom kroku šifruje, v druhom kroku sa dešifruje odlišným kľúčom. [1]

Algoritmus, ktorý sa skladá z troch DES kľúčov k_1 , k_2 a k_3 , každý má veľkosť 56 bitov, šifruje nasledovne:

$$C = E_{k_3}(D_{k_2}(E_{k_1}(P))) \quad (28)$$

kde C predstavuje šifrovaný text, P je otvorený text, E je šifrovanie, D dešifrovanie.

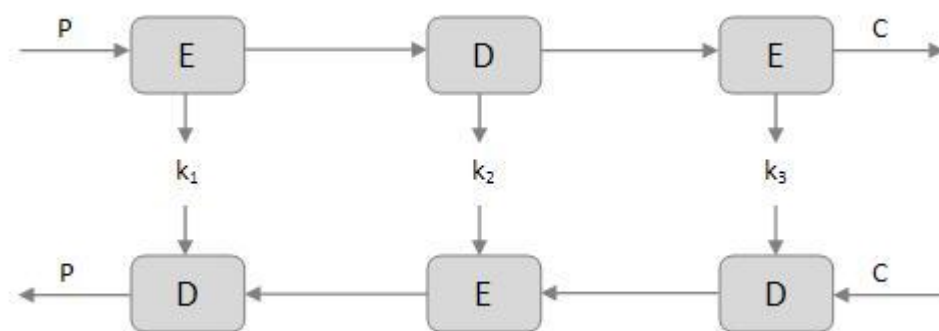
Na dešifrovanie sa používa opačný postup:

$$P = D_{k_1}(E_{k_2}(D_{k_3}(C))) \quad (29)$$

Norma definuje tri základné možnosti využívania kľúčov:

1. všetky kľúče sú nezávislé, teda $k_1 \neq k_2 \neq k_3$
2. dva kľúče sú nezávislé, $k_1 \neq k_2$ a $k_1 = k_3$
3. všetky kľúče sú zhodné, $k_1 = k_2 = k_3$

Vzhľadom na dĺžku šifrovacieho kľúča sa považuje šifra Tripple DES za spoľahlivú a dodnes sa používa.

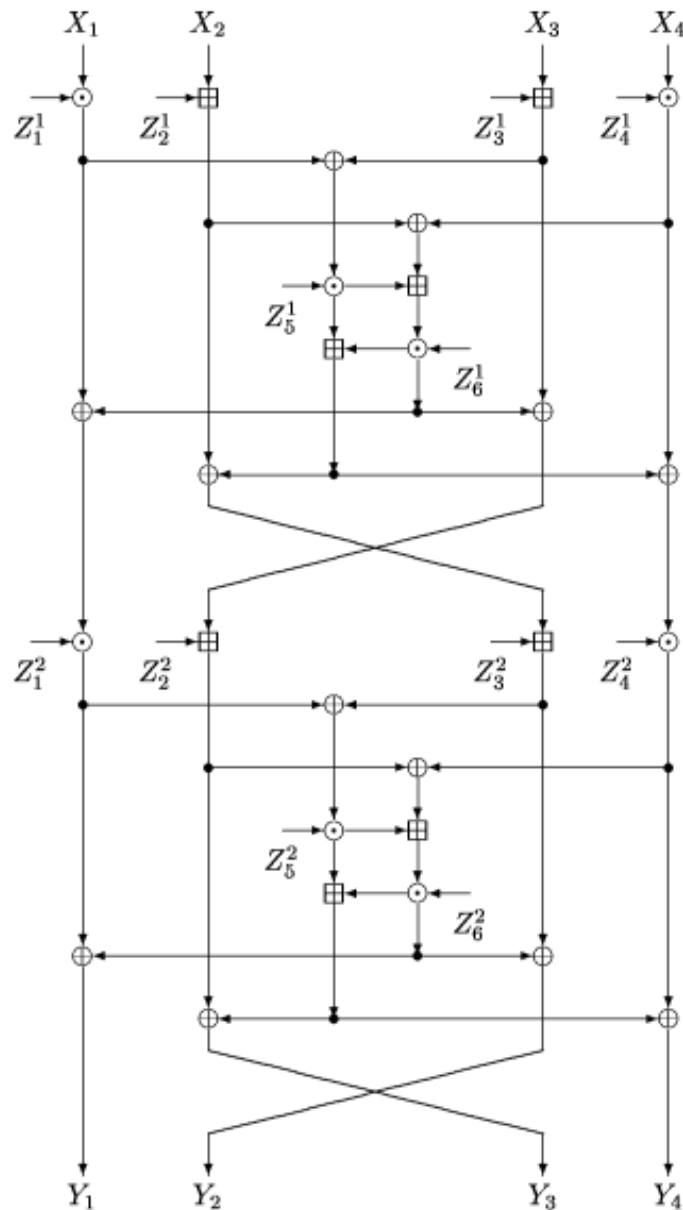


Obr. 13. 3DES s tromi kľúčmi

10.1.5 IDEA (International Data Encryption Algorithm)

IDEA je symetrický blokový algoritmus, prvý raz sa opísaný v roku 1991 vo Švajčiarsku. Je patentovaný v mnohých krajinách, ale môže sa bezplatne využívať na nekomerčné účely. Je to rýchly a bezpečný algoritmus, ktorý poskytuje vysoký stupeň ochrany a nachádza uplatnenie v mnohých aplikáciách.

IDEA algoritmus pracuje po 64 bitových blokoch a používa 128 bitový kľúč. Šifrovanie sa skladá z ôsmich identických cyklov výpočtu a výstupnej transformácie. Proces šifrovania a dešifrovania je podobný. IDEA v sebe spája viaceré algebrické operácie, v čom spočíva jej bezpečnosť. Obvykle sa využíva v režimoch CBC a CFB. [1]



Obr. 14. Schéma šifry IDEA

Postup šifrovania:

- rozdelenie vstupného 64 bitového bloku X na štyri subbloky X_1, X_2, X_3 a X_4
- 8 cyklov výpočtu, pričom jeden cyklus je tvorený týmito operáciami:

\oplus bitová operácia XOR

\boxplus sčítanie modulo 2^{16}

\odot násobenie modulo $2^{16} + 1$

- výstupná transformácia, tzv. polovičná runda, výsledkom ktorej je zašifrovaný blok pozostávajúci zo štyroch subblokov Y_1 , Y_2 , Y_3 a Y_4

V každom cykle algoritmus využíva šesť unikátnych subkľúčov veľkosti 16 bitov. Výstupná transformácia využíva ešte štyri subkľúče. Je teda vygenerovaných 52 unikátnych subkľúčov z primárneho šifrovacieho kľúča.

Generovanie kľúčov:

- 128 bitový primárny kľúč je rozdelený na osem 16 bitových subkľúčov
- nasleduje bitový posun primárneho kľúča o 25 bitov vľavo a opätovné rozdelenie na 8 subkľúčov
- tieto kroky sa opakujú až do vygenerovania všetkých 52 kľúčov

Bezpečnosť algoritmu

IDEA je odolná voči diferenciálnej kryptoanalýze a doposiaľ nebol zaznamenaný lepší spôsob prelomenia, než útokom hrubou silou. V takom prípade je však potrebné vyskúšať 2^{18} možných kombinácií k objaveniu kľúča, čo nie je momentálne možné splniť v reálnom čase.

10.1.6 AES (Rijndael)

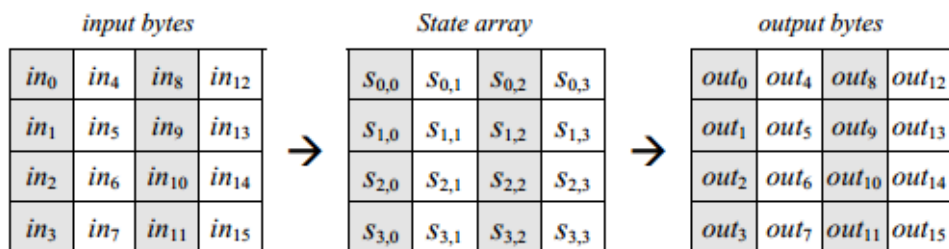
AES (Advanced Encryption Standard) je šifrovací štandard schválený v roku 2001 americkým inštitútom NIST, ako odozva na prelomenie dovtedy platného šifrovacieho štandardu DES. Je založený na šifrovacom algoritme Rijndael, ktorý vyvinuli belgickí kryptológovia J. Daemen a V. Rijmen. Životnosť AES sa odhaduje na 20 – 30 rokov.

AES je iteračná symetrická bloková šifra so stanovenou veľkosťou bloku 128 bitov a veľkosť kľúča 128, 192 a 256 bitov, hoci samotná šifra Rijndael podporuje rôzne veľkosti bloku N_b a veľkosti kľúča N_k .

Veľkosť kľúča závisí od počtu cyklov, ktoré prevádzajú otvorený text na šifrovaný:

- 10 cyklov – 128 bitový kľúč
- 12 cyklov – 192 bitový kľúč
- 14 cyklov – 256 bitový kľúč

Jednotlivé operácie algoritmu pracujú na základe medzivýsledkov, nazývaných *Stav*, reprezentovaných obdĺžnikovou maticou bajtov. Táto matica pozostáva zo štyroch riadkov obsahujúcich N_b stĺpcov, v AES je štandardne $N_b = 4$.



Obr. 15. Zmena stavu bloku AES [10]

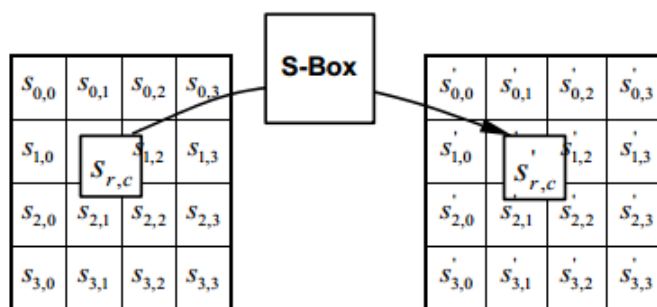
Popis šifrovania:

- Expanzia kľúča - výpočet 32 bitových rundových kľúčov z primárneho kľúča, ktorých je $N_b + N_b N_r$, kde N_r je počet cyklov
- Počiatočná inicializácia – vstupné bity sa spoja s prvými N_b rundovými kľúčmi pomocou operácie XOR a uložia sa do premennej *Stav* (*State*).
- Rundy – každá runda pozostáva zo štyroch operácií podľa predpisu:

```
Round (State, RoundKey)
{
    ByteSub (State);
    ShiftRow (State);
    MixColumn (State);
    AddRoundKey (State, RoundKey);
}
```

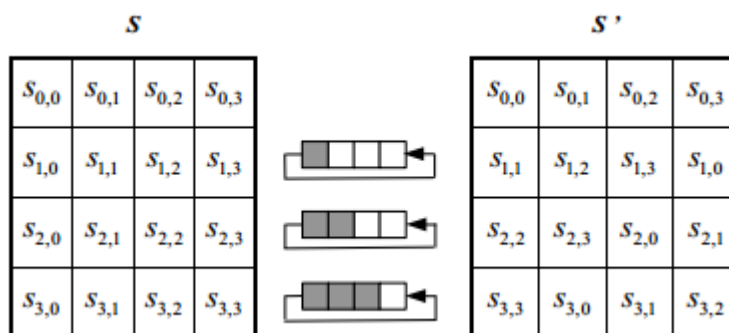
- Záverečná runda (bez MixColumn)

Operácia ByteSub je nelineárna bajtová substitúcia, realizovaná na každom bajte matice *A*, pričom využíva substitučnú tabuľku (*S-box*).



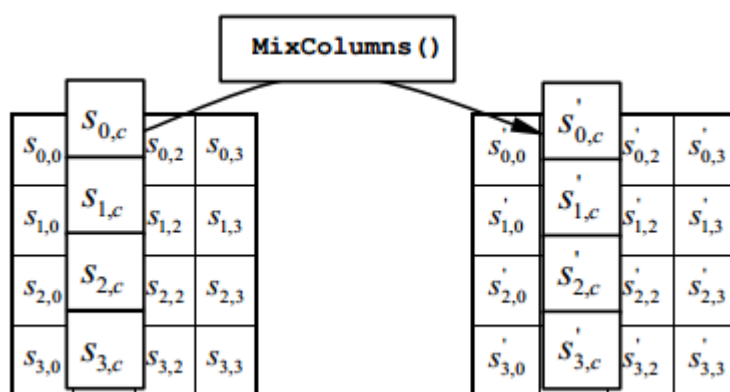
Obr. 16. Operácia ByteSub [10]

Operácia ShiftRow vykonáva cyklický posun riadkov matice A. Riadok 0 ostáva bez zmeny, riadok 1 sa posúva o C1 bajtov, riadok 2 o C2 bajtov, riadok 3 o C3 bajtov. Posuny C1, C2 a C3 závisia od veľkosti vstupného bloku N_b .



Obr. 17. Operácia ShiftRow() [10]

Operácia MixColumn pracuje s jednotlivými stĺpcami matice A. S jednotlivými prvkami stĺpcov sa pracuje ako s polynómami z $GF(2^8)$. Táto operácia zabezpečí premiešanie stĺpcov a zároveň ich vynásobí polynómom $C(x)$.



Obr. 18. Operácia MixColumn() [10]

Operácia AddRoundKey zabezpečí pridanie rundového kľúča do matice A pomocou bitovej operácie XOR.

Proces dešifrovania prebieha podobne ako samotné šifrovanie. Využívajú sa operácie inverzné k operáciám použitým na šifrovanie, mimo AddRoundKey.

Bezpečnosť algoritmu

Doposiaľ nebol zaznamenaný žiadny úspešný pokus o prelomenie plnej šifry. Útok hrubou silou vyžaduje 2^{200} operácií pri AES-259, čo nie je nateraz realizovateľné v reálnom čase.

Jediný úspešný útok na plnú AES šifru bol v roku 2009 pomocou postranných kanálov, teda vďaka úniku informácií. Životnosť AES sa odhaduje na 20 – 30 rokov.

10.2 Asymetrická kryptografia

10.2.1 Batožinový algoritmus (knapsack)

Tento algoritmus bol jeden z prvých algoritmov využívajúcich verejný kľúč. Jeho autormi sú R. Merkle a M. Hellman, ktorí ho vynali v roku 1978.

Princíp algoritmu spočíva vo výbere prvkov z množiny predmetov tak, aby mala batožina po naplnení týmito predmetmi určenú hmotnosť. Platí:

$$S = b_1M_1 + b_2M_2 + \dots + b_nM_n \quad (30)$$

kde S je celková hmotnosť, M_i je hmotnosť i -teho predmetu, n je počet predmetov a b_i koeficienty, ktoré môžu nadobúdať hodnotu 0 alebo 1 (podľa toho, či predmet je alebo nie je v batožine). [11]

Postup šifrovania:

1. určenie superrastúcej postupnosti, ktorá tvorí súkromný kľúč (ľahká batožina), pričom pre prvky množiny $\{M_i\}$ platí:

$$\sum_{k=0}^i M_k < M_{i+1} \quad (31)$$

2. vytvorenie verejného kľúča (ťažká batožina) transformáciou zo súkromného vynásobením číslom n modulo m , pričom platí:

$$\begin{aligned} GCD(n, m) &= 1 \\ m &> \sum_{i=1}^n M_i \end{aligned} \quad (32)$$

3. šifrovaná správa sa rozdelí na bloky s rovnakým počtom prvkov ako je prvkov batožiny
4. jednotkové prvky správy sa nahradia príslušným prvkom verejného kľúča na rovnakom mieste a stanovia celkové váhy batožín pre jednotlivé bloky – šifrovaný text

Dešifrovanie sa realizuje pomocou vzťahu:

$$n(n^{-1}) \equiv 1(\text{mod } m) \quad (33)$$

Bezpečnosť algoritmu

Batožinový algoritmus bol prelomený a napriek rôznym pokusom o zosilnenie, nepodarila sa zvýšiť jeho bezpečnosť.

10.2.2 RSA

Algoritmus RSA dostal názov podľa mien svojich objaviteľov Rivest-Shamir-Adelman, ktorí ho objavili v roku 1977. Tento pôvodne patentovaný algoritmus sa dnes s obľubou využíva v mnohých oblastiach bezpečnej komunikácie.

Inicializácia

Inicializácia je proces, pri ktorom sa vytvorí verejný a súkromný kľúč šifry RSA nasledovne:

1. najskôr sa náhodne zvolia dve dostatočne veľké prvočísla p a q pomocou generátora náhodných čísel, pričom ($p \neq q$)
2. spočítame modul $n = p \cdot q$, jeho dĺžka predstavuje dĺžku kľúča
3. vyberieme náhodné prirodzené číslo e – šifrovací kľúč tak, aby platilo:

$$\text{GCD}(e, (p-1)(q-1)) = 1 \quad (34)$$

4. dešifrovací kľúč d sa vypočíta tak, aby platilo:

$$e \cdot d \equiv 1 \pmod{(p-1)(q-1)} \quad (35)$$

a platí

$$d = e^{-1} \pmod{(p-1)(q-1)} \quad (36)$$

5. verejný kľúč je dvojica čísel (n, e) , číslo d tvorí súkromný kľúč [1]

Postup šifrovania a dešifrovania

- šifrovanie prebieha pomocou vzťahu

$$c = m^e \bmod n \quad (37)$$

- dešifrovanie je podľa vzťahu

$$m = c^d \bmod n \quad (38)$$

Použitím Čínskej vety o zvyškoch možno dosiahnuť niekoľkonásobné zrýchlenie dešifrovania.

Útoky na šifru

- **problém faktorizácie modulu n** , t.j. útok hrubou silou. Faktorizácia predstavuje veľký výpočtový problém v reálnom čase pri voľbe dostatočne veľkých prvočísel p a q .
- **RSA problém** nám umožňuje vypočítať e -tu odmocninu modulu n bez znalosti rozkladu n na prvočísla - teda zo vzťahu (34) musíme nájsť také celé číslo m , aby platilo:

$$m^e \equiv c \pmod{n} \quad (39)$$

Takýto algoritmus doposiaľ nebol nájdený, avšak ani dôkaz, že by nemohol existovať.

- **chybné generovanie kľúčov** – pre bezpečnosť algoritmu je potrebné dodržať niekoľko zásad generovania kľúčov:
 - prvočísla p a q nesmú byť príliš blízke
 - rozklad na prvočinitele musí obsahovať aspoň jedno veľké prvočíсло
 - podiel p/q nesmie byť jednoduchým zlomkom
- **spoločný modul** – pomocou znalostí verejného a súkromného kľúča, je možné faktorizovať modul n . V prípade, ak sa použije rovnaký modul n , môže teda dôjsť k prelomeniu šifry.
- **malá hodnota súkromného exponentu** - umožňuje tzv. Wienerov útok, teda súkromný exponent sa dá vypočítať aj zo znalosti verejného kľúča
- **malá hodnota verejného exponentu** – nie je taký nebezpečný ako predchádzajúci útok, ale je potrebné voliť exponent vyšší ako 3
- **čiastočné odhalenie súkromného kľúča** – ak platí, že $e < \sqrt{n}$, je možné zrekonštruovať celé d len zo zlomku jeho dĺžky
- **časový útok** - pomocou presne meraného času potrebného na dešifrovanie je možné získať súkromný exponent. Vhodným spôsobom ochrany je zabezpečenie konštantného časového množstva na dešifrovanie.

- **útok postranným kanálom** – je založený na nežiaducom úniku informácií postrannými kanálmi – napr. spotreba elektrického prúdu, elektromagnetické vyžarovanie, alebo chybové hlásenia
- **útok pomocou kvantového počítača** – využitím kvantového počítača by bola možná faktorizácia n v polynomiálnom čase

Bezpečnosť algoritmu

Na prelomenie šifry RSA prebiehajú verejné súťaže s finančnými odmenami pod názvom RSA Factoring Challenge. Súťaž je založená na probléme faktorizácie modulu. K prvému prelomeniu algoritmu došlo v roku 1991. Doposiaľ najdlhší známy prelomený algoritmus pomocou faktorizácie je RSA-768.

V roku 2010 sa trom vedcom z Michigenskej univerzity podarilo získať 1024 bitový kľúč RSA pomocou modifikácie napájania procesora, ktorý je držiteľom kľúča. Tento proces opisujú v dokumente *Fault-Based Attack of RSA Authentication*. [12] Aj keď útok nespôsobuje žiadne škody, ani nezanecháva stopy na zariadení, vyžaduje neustále napojenie na dané zariadenie.

Napriek všetkým popísaným útokom je algoritmus stále považovaný za bezpečný a využíva sa na poskytovanie bezpečnosti v informačných technológiách.

10.2.3 ElGamal

ElGamal je asymetrický algoritmus, ktorý nesie meno svojho objaviteľa Tahera El Gamala od roku 1985. Je založený na probléme diskretného algoritmu. Nevýhodou tohto algoritmu je, že veľkosť šifrovaného textu je dvojnásobná oproti otvorenému textu. ElGamal algoritmus je možné použiť pre šifrovanie aj digitálny podpis.

Postup šifrovania a dešifrovania:

- voľba prvočísla p a dvoch náhodných čísel g a x , ktoré sú menšie než p
- výpočet:

$$y = g^x \pmod{p} \quad (40)$$

- číslo x je súkromným kľúčom, čísla y , g , p predstavujú verejný kľúč
- voľba náhodného čísla k , pre ktoré platí:

$$\text{GCD}(k, (p - 1)) = 1 \quad 1 \leq k \leq (p - 2)$$

- pre šifrovanie otvoreného textu M platia vzťahy:

$$a = g^k \pmod{p} \quad (41)$$

$$b = y^k M \pmod{p}$$

kde čísla a a b predstavujú šifrovaný text.

- pre dešifrovanie platí:

$$M = b \cdot a^{(p-1-x)} \pmod{p} \quad (42)$$

[13]

Pre rôznych užívateľov, ktorým chceme zaslať tú istú správu, je nevyhnutné použiť rozdielne číslo k . Pre bezpečnosť šifrovania je potrebné číslo k vygenerovať skutočne náhodne a zabezpečiť, aby sa útočníkovi nedostalo do rúk, rovnako ani hodnota y^k .

Bezpečnosť algoritmu

Bezpečnosť ElGamal algoritmu sa opiera o vysokú výpočtovú zložitosť diskretného logaritmu. Doposiaľ nepoznáme algoritmus, ktorý by ho vedel riešiť problém diskretného algoritmu v reálnom čase.

10.2.4 Eliptické krivky

ECC (Elliptic Curve Cryptosystems) predstavuje asymetrický kryptografický systém založený na báze eliptických kriviek nad konečnými telesami. Kryptografické využitie eliptických kriviek navrhli v roku 1985 nezávisle na sebe N. Koblitz a V. S. Miller.

Konečné teleso

V kryptografii využívajúcej eliptické prvky sa pracuje s konečnými poľami. Toto pole sa skladá z konečnej množiny prvkov, pre ktorú sú definované algebrické operácie - sčítanie a násobenie. Operácia odčítania je definovaná ako pričítanie opačného prvku a delenie je definované ako násobenie inverzným prvkom.

Štruktúra, pre ktorú platia nasledovné axiómy, sa nazýva **konečné teleso**.

Pre každé $(F, +, \cdot)$, kde pre každé $a, b, c \in F$ platí:

Tab. 3. Základné axiómy pre konečné teleso F [14]

Algebrická uzavretosť poľa	$a + b = c \Rightarrow c \in F$ $a \cdot b = c \Rightarrow c \in F$	(43)
Asociatívny zákon	$(a + b) + c = a + (b + c)$ $(a \cdot b) \cdot c = a \cdot (b \cdot c)$	(44)
Komutatívny zákon	$a + b = b + a$ $a \cdot b = b \cdot a$	(45)
Distributívny zákon	$a \cdot (b + c) = a \cdot b + a \cdot c$	(46)
Existencia neutrálneho prvku	$\forall a \in F \exists 0 \in F: a + 0 = a$ $\forall a \in F \exists 1 \in F: a \cdot 1 = a$	(47)
Existencia opačného prvku	$\forall a \in F \exists (-a) \in F: a + (-a) = 0$	(48)
Existencia inverzného prvku	$\forall a \in F \exists a^{-1} \in F: a \cdot a^{-1} = 1$	(49)

Prvočíselné teleso F_p

Nech p je prvočíslo, pre ktoré platí $p > 3$, potom množina celých čísel $\{0, 1, 2, \dots, p-1\}$ spolu s operáciami sčítanie modulo p a násobenie modulo p tvorí konečné prvočíselné teleso F_p . [14]

Pre inverzný prvok a^{-1} platí:

$$a \cdot a^{-1} \equiv 1(\text{mod } p) \quad (50)$$

a pre opačný prvok $-a$ platí:

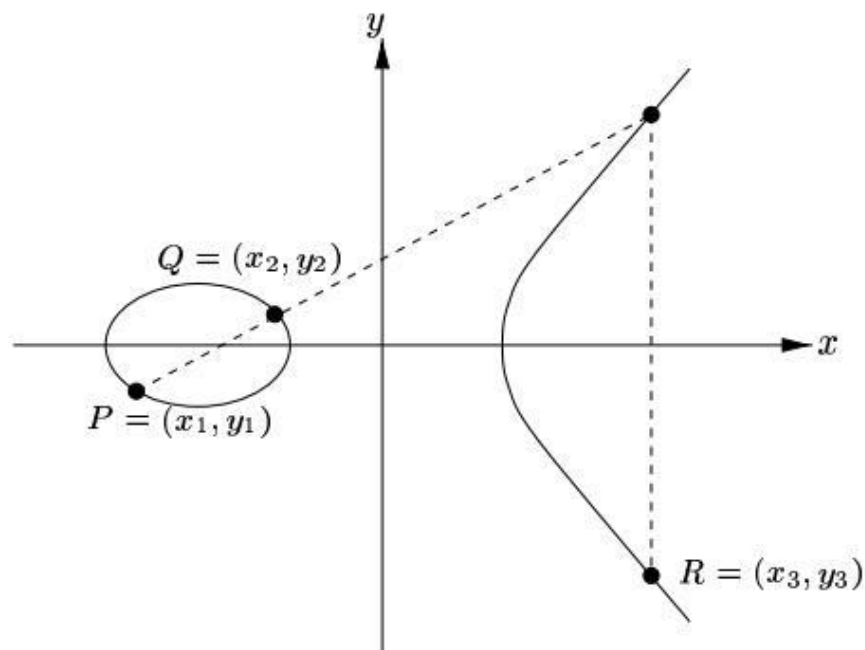
$$a + (-a) \equiv 0(\text{mod } p) \quad (51)$$

Pomocou týchto prvkov sa zbavíme zlomkov, ktoré spôsobovali zaokrúhľovacie chyby.

Galoisovo teleso $GF(2^m)$

Galoisovo teleso (Galois Field) s označením $GF(2^m)$ je konečné teleso Fp^m , kde $p = 2$. Galoisovo teleso tvoria polynómy s maximálnym stupňom $m - 1$, pričom jednotlivé koeficienty nadobúdajú hodnoty $\{0, 1\}$.

Nakoľko sa jedná o binárne postupnosti, počítačové spracovanie je jednoduché a efektívne. [14]



Obr. 19. Geometrická interpretácia sčítania dvoch bodov krivky E

$$P + Q = R \quad [15]$$

Eliptická krivka

Eliptická krivka E definovaná nad konečným poľom F je definovaná pomocou Weierstrassovej rovnice nasledovne:

$$y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6 \quad (52)$$

kde $a_1, a_2, a_3, a_4, a_6 \in F$. [14]

V kryptografii eliptických kriviek sa pracuje s eliptickými krivkami definovanými nad prvočíselným telesom Fp a s Galoisovým telesom $GF(2^m)$.

Eliptická krivka nad Fp

Nech Fp je konečné teleso a $p > 3$ je nepárne prvočíslo, potom eliptická krivka E nad telesom Fp je bod v nekonečne O spolu s množinou bodov $P = (x, y)$ pre $x, y \in Fp$ a spĺňa rovnicu:

$$y^2 = x^3 + ax + b \quad (53)$$

kde pre $a, b \in Fp$ platí:

$$4a^3 + 27b^2 \pmod{p} \neq 0 \quad (54)$$

[16]

Eliptická krivka nad $GF(2^m)$

Eliptická krivka E nad konečným telesom $GF(2^m)$ je bod v nekonečne O spolu s množinou bodov $P = (x, y)$ pre $x, y \in GF(2^m)$ a spĺňa rovnicu::

$$y^2 + xy = x^3 + ax^2 + b \quad (55)$$

kde $a, b \in GF(2^m)$ a $b \neq 0$.

Problém diskretného logaritmu eliptických kriviek (ECDLP)

Ak máme eliptickú krivku E definovanú nad konečným telesom Fp , potom konečný počet bodov na tejto krivke určuje rád krivky a značí sa $\#E$. Na výpočet približného rádu krivky $\#E$ slúži *Hassov teorém*:

$$p + 1 - 2\sqrt{p} \leq \#E \leq p + 1 + 2\sqrt{p} \quad (56)$$

Určiť rád krivky je výpočtovo náročná úloha, preto sa využíva Schoofov algoritmus.

Pre bod P na krivke E vypočítame postupnosť bodov $2P, 3P, 4P, \dots$, teda rôzne body xP . Vzhľadom na konečný počet bodov krivky, sa po určitom kroku m sa bude postupnosť opakovať.

Nech mP je bod opakovania, potom existuje nejaké n , $1 \leq n \leq m$, pre ktoré platí $mP = nP$, kde nP je predchádzajúci bod postupnosti. Ak $r = m - n$, potom $rP = mP - nP = O$, pričom najmenšie r , pre ktoré platí $rP = O$ sa nazýva rád bodu P . [17]

Kofaktor h je celočíselná hodnota podielu rádu eliptickej krivky a rádu bodu:

$$h = \frac{\#E}{r} \quad (57)$$

Kvôli bezpečnosti kryptografického systému sa snažíme, aby bola jeho hodnota čo najnižšia, ideálne ak $h = 1$, teda rád krivky je prvočíslo. [17]

Pri určenej eliptickej krivke $E(Fp)$ a bodoch $P, Q \in E(Fp)$, výpočet najmenšieho prirodzeného čísla k , $1 \leq k \leq r$, pre ktoré platí $Q = kP$, predstavuje *problém diskrétného logaritmu*. Pre veľké r je úloha neriešiteľná v polynomiálnom čase, v čom spočíva bezpečnosť kryptosystému eliptických kriviek.

Šifrovanie a dešifrovanie

Princíp šifrovania spočíva v prevode otvoreného textu na číselný formát, napr. pomocou binárnych čísel ASCII tabuľkou. Potom sa správa transformuje na jednotlivé bloky s pevnou dĺžkou, ktorých veľkosť nesmie presiahnuť veľkosť rádu bodu P .

Algoritmus šifrovania:

- vygenerujeme eliptickú krivku E nad telesom Fp s prvočíselným rádom r
- zvolíme súkromný kľúč $k \in [1, r-1]$
- vypočítame verejný kľúč $Q = kP$
- prevedieme správu m na bod M eliptickej krivky E
- vyberieme $a \in [1, r-1]$
- vypočítame $C1 = aP$
- vypočítame $C2 = M + aQ$
- výstup je šifrovaná správa $C1, C2$

Algoritmus dešifrovania:

- spočítame $M = C2 - k \cdot C1$
- prevedieme bod M na správu m
- výstup je otvorený text m

Kryptografické systémy založené na eliptických krivkách:

ECDH (Elliptic Curve Diffie-Hellman) – je protokol určený na výmenu súkromných kľúčov pomocou nezabezpečeného verejného kanálu, založený na Diffie-Hellmanovej schéme.

ECMVQ (Elliptic Curve Menezes-Qu-Vanstone) – je protokol založený na Diffie-Hellmanovej schéme určený na výmenu kľúčov, ktorý je odolný voči útoku Man-in-the-Middle.

ECIES (Elliptic Curve Integrated Encryption Scheme) – je najrozšírenejší šifrovací systém založený na eliptických krivkách. Využíva ElGamal schému.

ECDSA (Elliptic Curve Digital Signature Algorithm) – schéma pre digitálny podpis založená na DSA algoritme.

Bezpečnosť

ECC majú veľkú budúcnosť, pretože disponujú veľkou výhodou oproti ostatným kryptografickým systémom – poskytujú vysokú bezpečnosť vzhľadom na veľkosť použitého kľúča. Ďalšou veľkou výhodou je, že prakticky všetky systémy využívajúce problém diskretného logaritmu možno previesť na systém eliptických kriviek. ECC sa vďaka šifrovaniu s kratšími kľúčmi používa predovšetkým pre čipové karty, mobilné telefóny a v rôznych oblastiach bezkontaktného styku.

Tab. 4. Porovnanie dĺžky kľúčov RSA a ECC [18]

Bezpečnostná úroveň [b]	RSA dĺžka kľúča [b]	ECC dĺžka kľúča [b]	Približný pomer
80	1024	160-223	5-6:1
112	2048	224-255	8-9:1
128	3072	256-283	11-12:1
192	7680	384-511	15-20:1
256	15360	512-571	27-30:1

10.3 Hašovacie funkcie

Hašovacia funkcia je jednosmerná matematická funkcia, ktorá na vstupe dostane správu ľubovoľnej dĺžky, z ktorej vytvorí *haš* – jednoznačnú hodnotu (textový alebo binárny reťazec) s pevnou dĺžkou o veľkosti niekoľko bitov.

Hašovacie funkcie sa využívajú pri elektronických podpisoch, na overenie integrity dát, v kryptografických protokoloch, certifikátoch, pre kontrolu hesiel a mnohých iných kryptografických systémoch.

Definícia a vlastnosti

Hašovacia funkcia h je zobrazenie $h : X \rightarrow Y$, kde Y je konečná množina a X môže ale nemusí byť konečná množina. Potom $x \in X$ predstavuje vstup a $h(x)$ výsledný haš. Odtlačok $h(x)$ teda reprezentuje pôvodnú správu x . [19]

Hašovacie funkcie musia mať nasledujúce vlastnosti:

- **jednosmernosť**

Hašovacia funkcia $h : X \rightarrow Y$ je jednosmerná, ak pre dané $y \in Y$ nie je možné efektívne nájsť $x \in X$, aby platilo $h(x) = y$. [19]

Jednosmernosť predstavuje odolnosť systému, teda ak z výsledného odtlačku nevieme spätne určiť pôvodnú správu v efektívnom čase.

- **slabá bezkolíznosť**

Hašovacia funkcia $h : X \rightarrow Y$ je slabo odolá voči kolíziám, ak pre dané $x \in X$ nie je možné efektívne nájsť také $x' \in X$, aby platilo $h(x) = h(x')$. [19]

Slabá bezkolíznosť teda hovorí, že k danému dokumentu nájsť iný dokument s rovnakým výsledným odtlačkom.

- **silná bezkolíznosť**

Hašovacia funkcia $h : X \rightarrow Y$ je silne odolá voči kolíziám, ak nie je možné efektívne nájsť takú dvojicu vstupov $x, x' \in X$, aby platilo $x \neq x'$ a $h(x) = h(x')$. [19]

Silná bezkolíznosť hovorí, že nie je možné nájsť dva rozdielne vstupné dokumenty s rovnakým výstupným odtlačkom v efektívnom čase. Toho môžeme docieľiť zväčšením množiny Y . Naopak priveľká množina Y môže spôsobiť, že pri krátkych dokumentoch je výsledný odtlačok väčší, než vstupný dokument.

Ďalej musia hašovacie funkcie spĺňať nasledovné podmienky:

- **jednoduchosť výpočtu** – pre danú funkciu h a vstup x je jednoduché vypočítať $h(x)$
- **kompresia dát** – funkcia h prevádza ľubovoľne konečne veľký vstup x na výstup $h(x)$ s pevnou dĺžkou

10.3.1 Narodeninový útok (Birthday Attack)

Narodeninový útok je univerzálny typ útoku hrubou silou, ktorý sa využíva na hľadanie kolízií a je aplikovateľný na všetky hašovacie algoritmy. Je založený na tzv. narodeninovom paradoxe známom z teórie pravdepodobnosti.

Pre hašovaciu funkciu $h : X \rightarrow Y$ hľadáme také dva vstupy $x, x' \in X$, aby platilo $h(x) = h(x')$. [19]

Pre n -bitovú hašovaciu funkciu h pri útoku na vyhľadávanie kolízií stačí vyskúšať iba $2^{n/2}$ operácií, namiesto očakávaných 2^n . Pre 128 bitovú hašovaciu funkciu stačí teda iba 2^{64} náhodných pokusov, aby sme našli kolíziu. Úspešnosť narodeninového útoku teda závisí od dĺžky výstupu danej funkcie.

10.3.2 MD (Message-Digest Algorithm)

MD alebo digitálny odtlačok je kryptografický hašovací algoritmus vyvinutý profesorom Ronaldom Rivestom. MD dostáva na vstupe ľubovoľne veľkú správu, výstupom je haš s veľkosťou 128 bitov.

MD2

Algoritmus bol vyvinutý v roku 1989 a pozostáva z troch krokov. V prvom kroku je správa rozšírená na dĺžku deliteľnú v bajtoch 16timi, v druhom kroku je ku správe pripojený 16 bajtový kontrolný súčet, v poslednom kroku sa opakovane vykonáva kompresná funkcia, ktorej výsledkom je 128 bitový odtlačok. [20] V súčasnosti sa algoritmus nepoužíva, nakoľko nie je považovaný za bezpečný, pretože nespĺňa podmienku bezkolíznosti.

MD4

Algoritmus bol vyvinutý v roku 1990 a stal sa základom pre ďalšie známe algoritmy. V prvej časti je vstupná správa rozšírená na bloky s veľkosťou 512 bitov. V druhom kroku je k predchádzajúcemu výsledku pripojená 64 bitová reprezentácia pôvodnej správy. V nasledujúcom kroku sú spracovávané 512 bitové bloky v troch cykloch. [20] MD4 sa kvôli kolíznym útokom dnes nepoužíva a v roku 2011 prešiel do historického statusu.

MD5

Tento algoritmus bol navrhnutý v roku 1991 a je rozšírením MD4 algoritmu. Veľkosť vstupných a výstupných dát je rovnaká, ale zmenil sa vnútorný matematický výpočet funkcie.

Algoritmus MD5 nie je v súčasnosti považovaný za bezpečný. V roku 2006 publikoval český kryptológ Vlastimil Klíma práce, v ktorých popisuje program schopný vykonať kolízie MD5 do jednej minúty na osobnom počítači. Túto metódu nazval „tunelovanie“.

10.3.3 SHA (Secure Hash Algorithm)

Je skupina kryptografických algoritmov, ktoré sú súčasťou štandardu SHS americkej inštitúcie NIST. Tento štandard bližšie špecifikuje bezpečnosť algoritmov SHA-1, SHA-224, SHA-256, SHA-384, SHA-512, SHA-512/224 a SHA-512/256.

Tab. 5. Porovnanie vlastností algoritmov SHA [21]

Algoritmus	Veľkosť správy [b]	Veľkosť bloku [b]	Veľkosť slova [b]	Výstupná správa [b]
SHA-1	$<2^{64}$	512	32	160
SHA-224	$<2^{64}$	512	32	224
SHA-256	$<2^{64}$	512	32	256
SHA-384	$<2^{128}$	1024	64	384
SHA-512	$<2^{128}$	1024	64	512
SHA-512/224	$<2^{128}$	1024	64	224
SHA-512/256	$<2^{128}$	1024	64	256

SHA-1

SHA-1 je iteračný hašovací algoritmus, ktorý vychádza z algoritmu MD4. Bol predstavený v roku 1995 ako štandard FIPS PUB 180-1.

SHA-1 pracuje so správami s maximálnou veľkosťou 2^{64-1} bitov. Vstup je rozdelený na bloky s veľkosťou 512 bitov a doplnený tak, aby bola dĺžka deliteľná 512. Za správu sa pridá bit "1", k tomu potrebný počet núl tak, aby posledných 64 bitov ostalo na údaj o veľkosti správy. Týmto spôsobom sa sťažuje možnosť výskytu kolízií. Nasleduje samotná iteračná operácia, ktorá pozostáva z rozdelenia bloku na 16 x 32 bitových slov, ktoré sú rozšírené na 80 bitov a spracované v štyroch 20-krokových rundách pozostávajúcich z nelineárnych funkcií, bitových rotácií a exkluzívneho súčtu. Výstupom je 160 bitový haš. [22]

Podstatou SHA algoritmov je, že na začiatku každej operácie dostávajú na vstup dve hodnoty – výstup z predchádzajúcej operácie a nový blok správy určený na hašovanie. Na začiatku celého iteračného procesu je k prvému bloku pripojený tzv. inicializačný vektor.

SHA-1 dnes nie je považovaný za bezpečný algoritmus a samotná organizácia NIST odporúča od roku 2010 ukončiť jeho používanie.

SHA-2

Pod názvom SHA-2 sú označované ďalšie modifikácie SHA algoritmu. Patria sem SHA-224, SHA-256, SHA-384 a SHA- 512.

SHA-224 a SHA-256 pracujú s rovnakým vstupom ako SHA-1, používajú rovnaký spôsob rozdelenia vstupnej správy, líšia sa len vo vnútorných operáciách.

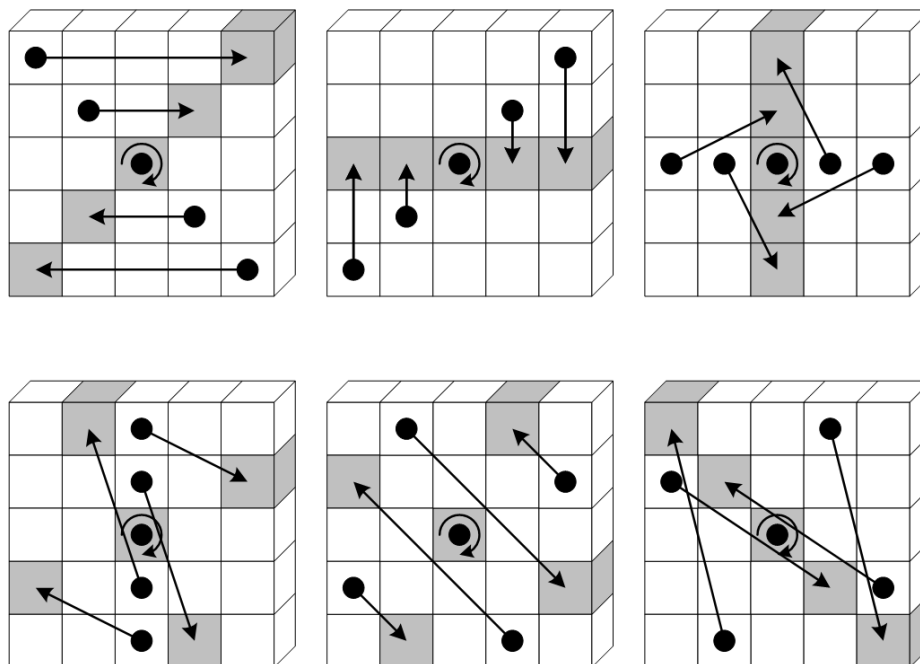
SHA-384 a SHA-512 môžu spracovať správy s maximálnou veľkosťou 2^{128} , vstupné bloky sú rozdeľované a dopĺňané podobne ako SHA-1, rozdiel je len vo veľkosti blokov – 1024 bitov a posledných 128 bitov je určených pre údaj o veľkosti správy. Tieto algoritmy pracujú so 64-bitovými slovami. [22]

NA SHA-2 nie sú zatiaľ známe žiadne útoky, ktoré by významným spôsobom narušili jeho bezpečnosť.

SHA-3

V roku 2007 vyhlásila organizácia NIST súťaž pre nový kryptografický hašovací algoritmus pod názvom SHA-3. Do súťaže bolo prihlásených 64 kryptografických algoritmov z celého sveta a 2. októbra 2012 bol vyhlásený víťazná funkcia s názvom KECCAK, ktorú navrhol tím belgických a talianskych kryptológov. SHA-3 má zatiaľ plniť len akúsi poistku v prípade prelomenia štandardu SHA-2.

Jadro KECCAK-u je založené na silných permutáciách a výstup môže byť menený podľa potreby. Medzi hlavné prednosti tohto algoritmu patrí vysoká úroveň paralelizmu, flexibilita vo výstupnej veľkosti, vysoká rýchlosť na hardwarovej úrovni a ochrana proti útoku postrannými kanálmi.



Obr. 20. Keccak Pi funkcia [23]

10.3.4 RIPEMD (RACE Integrity Primitives Evaluation Message Digest)

RIPEMD bol navrhnutý v roku 1992 v Belgicku v rámci projektu Race Integrity Primitives Evaluation pre európsku štandardizáciu kryptografických systémov. Algoritmus je založený na MD4.

V roku 1996 bola navrhnutá posilnená verzia RIPEMD-160 so 160 bitovým výstupným odtlačkom, RIPEMD-128 pre spätnú kompatibilitu a rozšírené verzie RIPEMD-256 a RIPEMD-320. Uvedené čísla v názve algoritmu udávajú veľkosť výstupného odtlačku v bitoch.

Všetky verzie RIPEMD algoritmu spracúvajú správy s maximálnou dĺžkou $2^{64}-1$. Vstup je doplnený a rozdelený na 512 bitové bloky. Funkcie sú rozdelené na dve časti, ktoré sú paralelne spracúvané a ich výsledky skombinované v závere spracovania každého bloku. [24]

V roku 2004 bolo ohlásené definitívne prelomenie pôvodného RIPEMD algoritmu.

10.4 MAC (Message Authentication Code)

MAC je autentizačný kód správy, ktorý je založený na použití hašovacej funkcie a súkromného kľúča. Je určený pre overenie správy, poskytuje integritu a autentizáciu

pôvodcu správy. Odtlačok správy teda nie je závislý len na samotnej správe, ale aj na kľúči, ktorým disponujú komunikujúci účastníci. MAC kód dokáže detekovať akúkoľvek úmyselnú alebo neúmyselnú zmenu v správe. K výpočtu MAC sa používajú blokové symetrické šifry (CBC-MAC, OMAC, PMAC) alebo hašovacie funkcie (HMAC).

Označme správu M , hašovaciu funkciu H a kľúč K , potom pre autentizačný kód platí:

$$\begin{aligned} MAC(M) &= H(K||M) \\ MAC(M) &= H(M||K) \end{aligned} \quad (58)$$

kde $||$ značí zretazenie dát. Kľúč sa pridáva k správe ako prefix alebo postfix. [24]

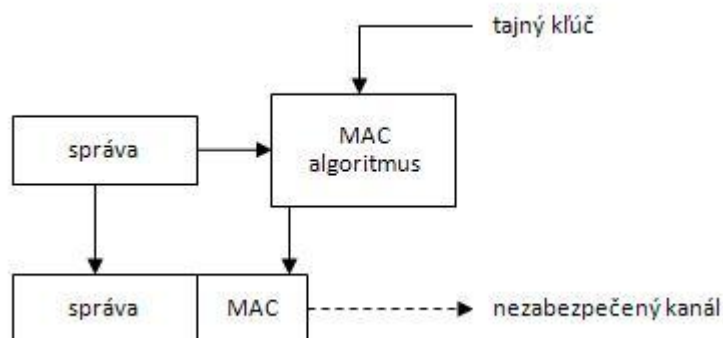
CBC-MAC (Cipher Block Chaining MAC) je založená na blokových symetrických šifrách. Správa je šifrovaná pomocou niektorého blokového algoritmu v režime CBC (2.1.1). Kvôli bezpečnosti je posledný blok ešte spracovaný trojitým šifrovaním. [19]

Výpočet MAC:

Nech $m = m_1, m_2, \dots, m_t$ je správa rozdelená do blokov, E šifrovacia transformácia, D dešifrovacia transformácia a IV je inicializačný vektor, potom:

1. $H_0 = IV$
2. $H_i = E_k(H_{i-1} \oplus m_i)$, pre $i = 1, \dots, t$
3. $MAC = E_k(D_{k'}(H_t))$

predstavuje postup pre výpočet MAC kódu. Pri výpočte sú použité dva symetrické kľúče k a k' , pre ktoré platí $k \neq k'$. [19]



Obr. 21. MAC algoritmus [28]

HMAC jedná sa o jednu z najpoužívanějších konštrukcií MAC, ktorá bola publikovaná v roku 1996 a je súčasťou štandardu RFC 2104.

Výpočet HMAC:

Nech M je správa, K tajný kľúč a H hašovacia funkcia, ktorá rozdeľuje správu do blokov s veľkosťou B bajtov. [24] Najskôr sa kľúč doplní nulovými bitmi na dĺžku bloku B a určia sa dva pevné reťazce *ipad* a *opad* nasledovne:

- *ipad* = bajt 0x36 opakované B -krát
- *opad* = bajt 0x5C opakované B -krát [25]

Výsledný HMAC bude vypočítaný pomocou vzťahu:

$$H((K \oplus opad) || H(K \oplus ipad) || M) \quad (59)$$

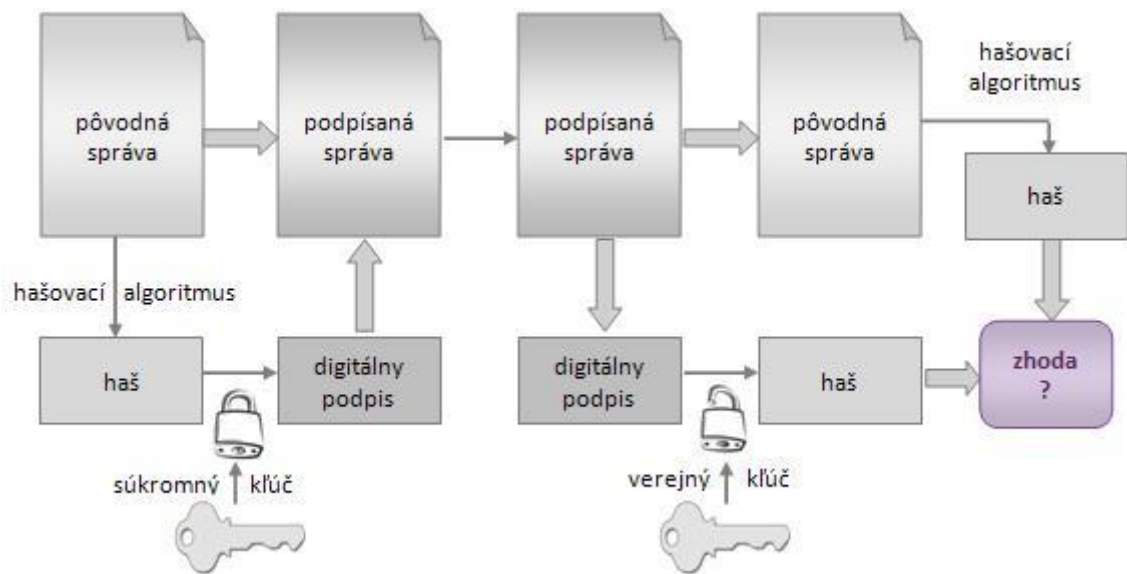
kde \oplus predstavuje operáciu XOR a $||$ operáciu zret'azenia. [24]

Takto vypočítaný HMAC má označenie v tvare HMAC-H, podľa použitej hašovacej funkcie.

11 APLIKOVANÁ KRYPTOLÓGIA

11.1 Digitálne podpisy

Digitálny podpis predstavuje číselný reťazec, ktorý je pevne spojený s elektronickým dokumentom. Schéma digitálneho podpisu pozostáva z podpisového a overovacieho algoritmu, pri čom sa využívajú asymetrické kryptografické systémy s hašovacími funkciami.



Obr. 22. Princíp digitálneho podpisu

Digitálne podpisové schémy sa delia do dvoch skupín:

- *digitálne podpisové schémy s dodatkom* – pri overovaní vyžadujú originálnu správu ako vstup – napr. DSA, ElGamal a i.
- *digitálne podpisové schémy s obnovou správy* – originálna správa sa získa zo samotného podpisu, nie je potrebná pri overovaní – napr. RSA, Rabínov systém, a i.

11.1.1 RSA schéma

RSA podpisová schéma je založená na asymetrickom kryptografickom algoritme RSA. Označme verejný kľúč dvojicou čísel (n, e) , súkromný kľúč d . Nech s je digitálny podpis, m je správa a H je hašovacia funkcia. [19]

Vytvorenie podpisu:

$$s = H(m)^d \pmod{n} \quad (60)$$

Overenie podpisu:

$$H(m) = s^e \pmod{n} \quad (61)$$

Bezpečnosť

Bezpečnosť RSA schémy závisí od bezpečnosti samotného RSA algoritmu, ako aj hašovacej funkcie. Problém vzniká pri dlhších správach a ich rozdelením na viac blokov. Už z dvoch podpísaných dokumentov sa dá zostrojiť tretí podpísaný dokument bez toho, aby sa o tom dozvedel užívateľ. Je to kvôli multiplaktívnym vlastnostiam algoritmu RSA:

Ak $s_1 = m_1^d \pmod{n}$ a $s_2 = m_2^d \pmod{n}$ sú podpisy správ m_1 a m_2 , potom

$$s = (m_1 m_2)^d \pmod{n} \quad (62)$$

a ak platí, že $m = m_1 m_2$, potom s je platným podpisom. [19]

Použitím hašovacej funkcie sa bezpečnosť zvyšuje, pretože hašovacie funkcie zvyčajne nemajú multiplaktívne vlastnosti.

11.1.2 ElGamal schéma

Postup generovania inicializačných kľúčov je rovnaký ako pri ElGamal algoritme, z ktorého vychádza:

1. zvolíme si veľké prvočíslo p a náhodné číslo g , ktoré je menšie než p
2. zvolíme náhodné celé číslo x , pre ktoré platí $1 \leq x \leq (p - 2)$
3. vypočítame: $y = g^x \pmod{p}$
4. čísla (y, g, p) tvoria verejný kľúč a číslo x súkromný kľúč [19]

Vytvorenie podpisu:

1. zvolíme náhodné celé číslo k , pre ktoré platí:

$$1 \leq k \leq (p - 2) \quad \text{a} \quad GCD(k, (p - 1)) = 1$$

2. vypočítame: $r = g^k \pmod{p}$
3. vypočítame: $k^{-1} \pmod{p - 1}$
4. vypočítame: $s = k^{-1}\{H(m) - xr\} \pmod{p - 1}$
5. podpis pre správu m tvorí dvojica (r, s)

Overenie podpisu:

Podpis môže overiť ktokoľvek zo znalosti verejného kľúča (y, g, p) a správy m .

1. overíme platnosť $1 \leq r \leq (p - 1)$
2. vypočítame: $v_1 = y^r r^s \pmod{p}$
3. vypočítame: $H(m)$ a $v_2 = g^{H(m)} \pmod{p}$
4. podpis je korektný, ak $v_1 = v_2$

Dôkaz o korektnosti podpisu vyplýva zo vzťahu:

$$y^r r^s \equiv g^{xr+ks} \equiv g^{H(m)} \pmod{p} \quad (63)$$

[19]

Bezpečnosť

ElGamal systém sa opiera o bezpečnosť diskretného logaritmu a použitia bezpečnej hašovacej funkcie. Pre každú správu treba vygenerovať novú hodnotu k a túto treba držať v tajnosti, lebo s jej znalosťou, či viacnásobným použitím si útočník ľahko vypočíta súkromný kľúč.

ElGamal je digitálna podpisová schéma s dodatkom, ale môže byť použitá aj ako schéma s obnovou správy.

11.1.3 DSA (Digital Signature Algorithm)

DSA algoritmus tvorí základ štandardu pre digitálny podpis DSS. Je variantom ElGamal schémy a patrí do skupiny digitálnych podpisov s dodatkom. DSA algoritmus sa používa s hašovacou funkciou SHA-1.

Základné parametre:

- p – veľké prvočíslo s bitovou dĺžkou L pre ktoré platí $2^{L-1} < p < 2^L$
- q – prvočíslo s dĺžkou N , ktoré je deliteľom $(p - 1)$ a platí $2^{N-1} < q < 2^N$
- g – číslo z intervalu $[2, p - 2]$, pre ktoré platí $g^q \pmod{p} = 1$ [19]

Štandard DSS [27] špecifikuje tieto dĺžky:

$$L = 1024, N = 160$$

$$L = 2048, N = 224$$

$$L = 2048, N = 256$$

$$L = 3072, N = 256$$

Generovanie kľúčov:

1. náhodne vygenerujeme privátny kľúč x v intervale $[1, q - 1]$
2. vypočítame verejný kľúč: $y = g^x \pmod{p}$
3. trojica (p, q, g) a verejný kľúč y tvoria verejné parametre

Vytvorenie podpisu:

1. náhodne generujeme číslo k , pre ktoré platí $0 < k < q$
2. vypočítame $r = (g^k \pmod{p}) \pmod{q}$
3. vypočítame

$$s = (k^{-1}(H(m) + xr) \pmod{q}) \tag{64}$$

kde H je hašovacia funkcia správy m , a $kk^{-1} \pmod{q} = 1$

4. dvojica čísel (r, s) tvorí podpis správy m [19]

Overenie podpisu:

Máme správu m s podpisom (r, s) a verejné parametre (p, q, g, y) :

1. overíme $0 < r < q$ a $0 < s < q$, inak je podpis neplatný
2. vypočítame $w = s^{-1} \pmod{q}$ a $H(m)$
3. vypočítame $u_1 = w \cdot H(m) \pmod{q}$
 $u_2 = rw \pmod{q}$
4. vypočítame $v = (g^{u_1} y^{u_2} \pmod{p}) \pmod{q}$
5. ak $v = r$, podpis je korektný [26]

Bezpečnosť

Číslo k sa nazýva kľúčom správy a musí byť kvôli bezpečnosti vygenerované pre každú správu zvlášť. Toto číslo musí byť utajené, v opačnom prípade je možné pomocou verejných hodnôt a podpísanej správy jednoducho vypočítať súkromný kľúč.

Bezpečnosť DSA schémy je založená na výpočtovej zložitosti diskretného logaritmu. Bez použitia hašovacej funkcie sa dá náhodná správa falšovať, podobne ako v RSA alebo ElGamal schéme.

11.1.4 ECDSA (Elliptic Curve Digital Signature Algorithm)

ECDSA patrí do rodiny štandardizovaných postupov pre digitálne podpisy definovaných v DSS. Schéma je analogická ku DSA, ale pracuje nad konečnými telesami eliptických kriviek. Tento systém poskytuje rovnakú bezpečnosť pri použití kratšieho kľúča.

Tab. 6. ECDSA bezpečnostné parametre [27]

Bitová dĺžka n [$\log_2 n$]	Maximálny kofaktor (h)
160 - 223	2^{10}
224 - 255	2^{14}
256 - 383	2^{16}
384 - 511	2^{24}
≥ 512	2^{32}

Generovanie kľúčov:

- zvolíme bod $P \in E$ s rádom n
- vygenerujeme náhodný privátny kľúč $d \in [1, n - 1]$
- vypočítame verejný kľúč $Q = dP$
- štvorica $[E, P, n, Q]$ tvorí verejný kľúč [15]

Vytvorenie podpisu:

- náhodne vygenerujeme tajné číslo $k \in [1, n - 1]$
- vypočítame bod $kP = (x_1, y_1)$ a číslo $r = x_1 \bmod n$
- ak $r = 0$, vygenerujeme nové číslo k

- vypočítame $k^{-1} \bmod n$
- vypočítame $s = k^{-1}\{H(m) + dr\} \bmod n$, kde $H(m)$ predstavuje haš správy m
- ak $s = 0$, vygenerujeme nové číslo k
- dvojica (r, s) predstavuje podpis správy m [15]

Overenie podpisu:

Máme správu m s podpisom (r, s) a verejný kľúč $[E, P, n, Q]$:

- overíme, že $r, s \in [1, n - 1]$, inak je podpis neplatný
- vypočítame $w = s^{-1} \bmod n$ a $H(m)$
- vypočítame $u_1 = w \cdot H(m) \bmod n$

$$u_2 = rw \bmod n$$

- vypočítame $u_1P + u_2Q = (x_0, y_0)$

$$v = x_0 \bmod n$$

- ak $v = r$, podpis je korektný [15]

Bezpečnosť

ECDSA bezpečnosť je založená na probléme diskretného logaritmu eliptických kriviek ECDLP a bezkolíznosti vybranej hašovacej funkcie.

11.2 Kryptografické protokoly

Kryptografický protokol je zložený algoritmus, definovaný presnou postupnosťou krokov, ktoré stanovujú opatrenia vyžadované z dvoch alebo viac strán, aby bolo možné dosiahnuť stanovený cieľ.

11.2.1 Shamirov protokol (Shamir's three-pass protocol)

Shamirov protokol (niekde nazývaný aj *Shamirov no-key protocol*) je spôsob na bezpečnú výmenu akejkoľvek správy bez nutnosti použitia šifrovacích kľúčov. Bol vyvinutý v približne roku 1980 A. Shamirom. [28]

Počas šifrovacej aj dešifrovacej fázy nie je potrebná žiadna kľúčová dohoda, ani výmena kľúčov, každá strana používa len svoj lokálny symetrický kľúč. Tento spôsob výmeny správ však vyžaduje použitie komutatívnej šifrovacej funkcie, ktorá povoľuje dešifrovanie správy bez ohľadu na poradie použitých kľúčov.

Postup:

Nech je odosielateľ A, jeho súkromný symetrický kľúč k_a , príjemca B so symetrickým súkromným kľúčom k_b a správa M . Prenos správy potom pozostáva z týchto krokov:

1. $A \rightarrow B: \{M\}_{k_a}$
2. $A \leftarrow B: \{\{M\}_{k_a}\}_{k_b}$
ak platí $\{\{M\}_{k_a}\}_{k_b} = \{\{M\}_{k_b}\}_{k_a}$, potom:
3. $A \rightarrow B: \{M\}_{k_b}$
4. $B: \{M\}$ [29]

Shamirov algoritmus využíva problém diskretného logaritmu a veľké prvočíslo:

- šifrovanie: $E(e, m) = m^e \bmod p$
- dešifrovanie: $D(d, m) = m^d \bmod p$, kde p je veľké prvočíslo.

Pre každý exponent e v intervale $1 \dots p - 1$, platí:

$$\text{GCD}(e, p - 1) = 1$$

a pre exponent d musí platiť:

$$de \equiv 1 \pmod{(p - 1)}$$

Shamirov protokol vyžaduje komutatívne vlastnosti, aby platilo:

$$E(a, E(b, m)) = m^{ab} \bmod p = m^{ba} \bmod p = E(b, E(a, m))$$

Bezpečnosť

Shamirov protokol poskytuje ochranu len pred pasívnymi útokmi, nakoľko nevyžaduje autentifikáciu. Je preto citlivý na útoky typu Man-In-The-Middle.

11.2.2 Diffie-Hellman schéma

Diffie-Hellman (D-H) schéma umožňuje účastníkom dohodnúť si verejným kanálom spoločný kľúč určený na následné šifrovanie komunikácie. Túto schému prvý raz publikovali W. Diffie a M. Hellman v roku 1976, aj keď ju objavil skôr Malcolm J. Williamson. V roku 2002 Hellman navrhol pomenovanie algoritmu Diffie-Hellman-Malcolm.

Postup:

Nech p je dostatočne veľké prvočíslo a g je základ, na ktorom sa vopred dohodnú účastníci komunikácie A a B:

- $A \rightarrow B: X$, kde $X = g^x \bmod p$ a x je náhodne zvolené číslo odosielateľom A
- $A \leftarrow B: Y$, kde $Y = g^y \bmod p$ a y je náhodne zvolené číslo príjemcom B
- $A: K = Y^x \bmod p$
- $B: K = X^y \bmod p$

pričom K je spoločný výsledný kľúč, pretože platí:

$$Y^x \bmod p = g^{xy} \bmod p = X^y \bmod p \quad [19]$$

Bezpečnosť

Bezpečnosť D-H protokolu je založená na probléme diskretného logaritmu a tzv. Diffie-Hellman probléme:

Pre X a Y je potrebné vypočítať také K , aby platilo:

$$K = g^{xy}$$

kde $X = g^x$ a $Y = g^y$. [19]

D-H protokol je odolný voči pasívnym útokom, ale možno ho napadnúť aktívnym útokom MITM, nakoľko protokol v základnej forme nevyžaduje autentifikáciu.

11.2.3 ECDH

ECDH (Elliptic Curve Diffie-Hellman) je protokol pre verejnú dohodu na kľúči, ktorý umožňuje dvom stranám vymeniť spoločný tajný kľúč v nechránenom komunikačnom kanáli pre následné použitie v symetrickom šifrovaní. Je založený na D-H protokole s využitím kryptografie eliptických kriviek.

Postup:

Generácia doménových parametrov nad konečným poľom Fp :

- veľkosť poľa $q = p$, pričom $p > 3$ musí byť prvočíslo
- dva elementy $a, b \in Fp$, ktoré sú definované na eliptickej krivke E
- dva elementy $xG, yG \in Fp$, ktoré určujú bod $G = (xG, yG)$ na krivke E , pričom $G \neq 0$

- n je rád bodu G , pričom $n > 2^{160}$
- kofaktor $h = \#E(F_p)/n$
- doménové parametre sú (q, a, b, G, n, h) [30]

Generácia kľúčového páru:

- výber náhodného čísla d na intervale $[1, n - 1]$
- výpočet bodu $Q = (x_Q, y_Q) = dG$
- kľúčový pár predstavuje dvojica (d, Q) , pričom d je súkromný a Q verejný kľúč [30]

Algoritmus:

Predpokladom sú spoločné doménové parametre (q, a, b, G, n, h) , na ktorých sa dohodli účastníci A a B.

- obaja účastníci si vypočítajú svoj kľúčový pár, teda (d_A, Q_A) pre účastníka A a (d_B, Q_B) pre účastníka B
- účastník A si vypočíta bod $Z = (x_Z, y_Z) = d_A Q_B$
- účastník B vypočíta bod $Z' = (x_{Z'}, y_{Z'}) = d_B Q_A$
- $Z = Z'$ pretože $d_A Q_B = d_A d_B G = d_B d_A G = d_B Q_A$
- x_Z je spoločný tajný symetrický kľúč [30]

Bezpečnosť

Bezpečnosť ECDH sa zakladá na probléme diskretného logaritmu a Diffie-Hellman probléme pre eliptické krivky.

11.2.4 ECMVQ

ECMVQ (Elliptic Curve Menezes-Qu-Vanstone) je protokol pre kľúčovú dohodu založený na D-H schéme s použitím kryptografie na báze eliptických kriviek.

Postup:

Základným predpokladom sú rovnaké doménové parametre (q, a, b, G, n, h) , ktoré boli generované obdobne, ako pri schéme ECDH. [30]

Algoritmus má tieto vstupy:

- dva páry súkromných kľúčov (d_{1A}, Q_{1A}) a (d_{2A}, Q_{2A}) , ktoré vlastní účastník A
- dva verejné kľúče Q_{1B} a Q_{2B} , ktoré vlastní účastník B [30]

Algoritmický výpočet:

- výpočet celého čísla $\text{implicitsig}_A \equiv d_{2A} + (\text{avf}(Q_{2A}) d_{1A}) \pmod{n}$
- výpočet bodu $P = h(\text{implicitsig}_A)(Q_{2B} + (\text{avf}(Q_{2B})Q_{1B}))$
- kontrola $P \neq 0$, inak STOP
- x_P je x -ová súradnica bodu P a predstavuje spoločný tajný symetrický kľúč

Výpočet pomocných funkcií:

- výpočet $x_p' = \overline{x_p} \pmod{2^{\lceil f/2 \rceil}}$

kde $\overline{x_p}$ je celočíselná reprezentácia x -ovej súradnice bodu G a $f = \lceil \log_2 n \rceil$

- výpočet $\text{avf}(Q) = x_p' + 2^{\lceil f/2 \rceil} [30]$

Bezpečnosť

ECMVQ protokol obsahuje násobenie kofaktoru, čo predstavuje účinný spôsob ako predchádzať bezpečnostným problémom, napr. útokom typu MITM.

11.3 PGP (Pretty Good Privacy)

PGP je hybridný kryptografický program, primárne určený na ochranu a bezpečnosť mailových správ v elektronickej komunikácii, neskôr aj na zabezpečenie telefonických hovorov a šifrovanie diskových oddielov. PGP zabezpečuje šifrovanie, dešifrovanie, podpisovanie, kľúčové hospodárstvo a certifikačné služby. Vznikol v roku 1991 a jeho autorom je Phil R. Zimmermann. Vďaka vysokým kvalitám bol PGP prijatý ako štandard a podrobne popísaný v dokumente RFC4880 s názvom OpenPGP. PGP je dostupný pre mnohé platformy – napr. Windows, Unix, MacOS, MS-DOS,...

PGP systém v sebe zahŕňa digitálny podpis, dôvernosť, kompresie a konverzie správ a dátových súborov. Symetrický systém sa používa na šifrovanie a asymetrický systém na ochranu daného symetrického kľúča. [31]

Postup šifrovania a dešifrovania:

1. odosielateľ vytvorí dokument
2. dokument sa vo väčšine prípadov skomprimuje
3. vygeneruje sa jedinečný kľúč pre konkrétny dokument, ktorý je následne zakódovaný pomocou symetrického algoritmu
4. symetrický kľúč je zakódovaný verejným kľúčom príjemcu

5. obe časti sa spoja a tvoria PGP dáta [31]

Dešifrovanie prebieha analogicky:

1. príjemca oddelí zašifrovaný text a zašifrovaný symetrický kľúč
2. symetrický kľúč príjemca dešifruje pomocou svojho súkromného asymetrického kľúča
3. symetrický kľúč príjemca použije na dešifrovanie textu
4. ak bol dokument skomprimovaný, tak ho príjemca dekomprimuje [31]

Autentifikácia pomocou digitálneho podpisu:

Pri digitálnom podpise sa využíva hašovacia funkcia a podpisový algoritmus s verejným kľúčom. Postup je nasledovný:

1. odosielateľ vytvorí správu
2. vygeneruje sa odtlačok správy
3. vygeneruje sa podpis z hašovacieho odtlačku pomocou súkromného kľúča odosielateľa
4. podpis sa pripojí ku správe
5. príjemca uloží kópiu podpisu prijatej správy
6. príjemca zo správy vygeneruje nový odtlačok a porovná ho s uloženým podpisom – ak sa zhodujú, podpis je považovaný za korektný [31]

Algoritmy vhodné na použitie v PGP odporúčané v dokumente RFC 4880. [31]

Algoritmy s verejným kľúčom:

- RSA – šifrovanie a podpis
- ElGamal – šifrovanie
- DSA - podpis
- ECDH - šifrovanie
- ECDSA – podpis

Symetrické algoritmy:

- IDEA – 128 bitový kľúč
- TrippleDES - 168 bitový kľúč
- CAST5 - 128 bitový kľúč
- Blowfish - 128 bitový kľúč

- AES – 128, 192 a 256 bitový kľúč
- Twofish - 256 bitový kľúč
- Camellia - 128, 192 a 256 bitový kľúč

Hašovacie algoritmy:

- MD5
- SHA-1
- RIPEMD-160
- SHA256, SHA384, SHA512, SHA 224

Eliptické krivky v OpenPGP

V dokumente RFC 6637 je podrobne popísaný štandard pre použitie ECC v systéme OpenPGP. Kryptografia založená na eliptických krivkách patrí medzi asymetrické kryptografické systémy.

Štandardom podporované algoritmy sú ECDSA a ECDH s krivkami P-256, P-384 a P-521, ktoré sú opísané v štandarde DSS (FIPS PUB 186-3). Odporúčané hašovacie funkcie k jednotlivým krivkám sú zobrazené v nasledujúcej tabuľke (Tab. 7). Hašovacia funkcia SHA-1 sa nesmie používať kvôli bezpečnosti.

Pre distribúciu súkromného kľúča sa odporúča používať algoritmus AES vďaka jeho spätnej kompatibilite s ECDH.

Tab. 7. Odporúčané algoritmy k eliptickým krivkám [27]

Názov krivky	Hash algoritmus	Šifrovací algoritmus
NIST krivka P-256	SHA2-256	AES-128
NIST krivka P-384	SHA2-384	AES-192
NIST krivka P-521	SHA2-512	AES-256

Bezpečnosť

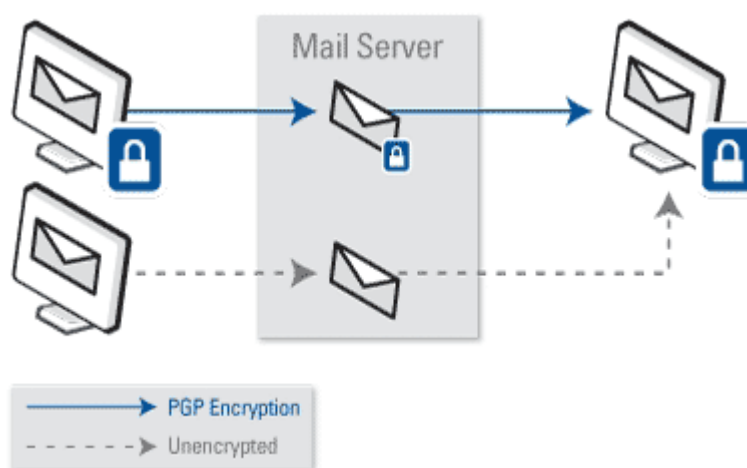
ECDH a ECDSA sú citlivé na útoky postrannými kanálmi. Celý systém je však považovaný za bezpečný a doteraz sa používa.

Použitie PGP:

PGP je veľmi obľúbený a rozšírený na internete. Poskytuje komplexné šifrovacie služby na ochranu dát bez toho, aby zaťažoval užívateľov. V roku 2010 bola spoločnosť PGP akvizovaná spoločnosťou Symantec, ktorá zaradila poskytované produkty do svojho portfólia.

PGP Desktop Email

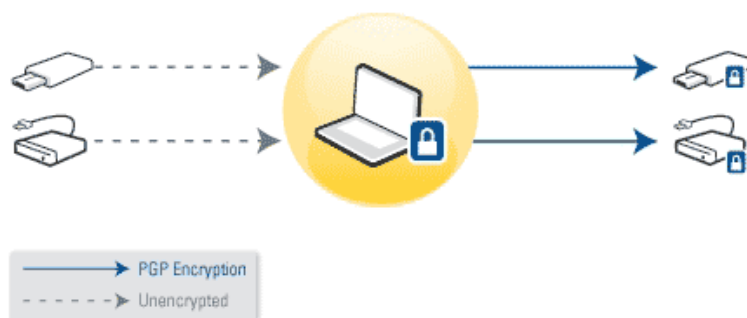
Zabezpečuje automatické šifrovanie, dešifrovanie, digitálne podpisy a overenie emailov. Ochrana dát je zabezpečená na systémovej úrovni. Medzi podporovaných klientov patria napr. Microsoft Outlook, Mozilla Thunderbird, Apple Mail a i. [32]



Obr. 23. PGP Desktop Email [32]

PGP Whole Disk Encryption

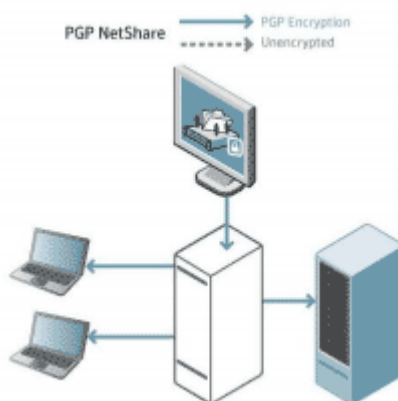
Tento nástroj zabezpečuje komplexné šifrovanie pre všetky dáta (užívateľské súbory, systémové súbory, skryté súbory, atď.) pevných aj externých diskov, vrátane USB flash jednotiek a vymeniteľných zariadení. Užívateľ nie je zaťažovaný šifrovaním, pracuje ako na nezašifrovanom počítači. [32]



Obr. 24. PGP Whole Disk Encryption [32]

PGP Net Share

Technológia Net Share umožňuje kontrolu prístupu k zdieľaným zložkám a súborovým serverom. Dáta prechádzajú v sieti chránené šifrovaním, pretože k šifrovaniu a dešifrovaniu dochádza až v koncových staniciach. [32]



Obr. 25. PGP Net Share [32]

11.4 Identifikácia a autentizácia entít

Entita – predstavuje ľubovoľný objekt (osoba, vec, udalosť, jav), ktorý má svoju identitu a môže byť jednoznačne identifikovateľný.

Identifikácia – predstavuje proces, pri ktorom je neznámej entite pridelená hodnota - identita, vďaka ktorej sa stane známou.

Autentizácia – je proces, pri ktorom je overená identita entity. Tento proces môže byť rozdelený do troch kategórií:

- *čo viem* – heslo, identifikačné číslo, PIN kód, ...

- *čo vlastním* – osobný preukaz, identifikačná karta, GRID karta - spoločne nazývané **tokeny**
- *čo som* – odtlačok prsta, štruktúra očnej sietnice, DNA, ... [29]

11.4.1 Autentizácia heslom

Predstavuje tzv. slabú autentizáciu a patrí k najpoužívanejším spôsobom autentizácie. Jednotlivé schémy sa odlišujú spôsobom ukladania hesiel a spôsobom ich overovania.

Heslo v nešifrovanom tvare – heslo sa odosiela vo otvorenej podobe a následne je overené. Tento systém patrí k najviac zraniteľným, pretože ktokoľvek môže heslo získať. Administrátor má prístup ku všetkým heslám a v prípade nedostatočného zabezpečenia môže byť zneužitá celá databáza hesiel.

Nech je systém hesiel tvorený množinou usporiadaných dvojíc:

$$S = (userID, password),$$

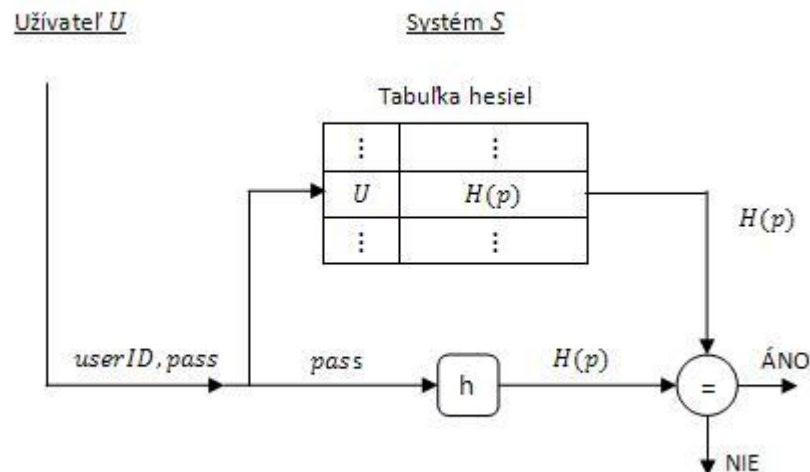
potom pre každú dvojicu $(U, p) \in S$, systém povolí prístup, v opačnom prípade prístup odmietne.

Heslo s použitím jednosmernej hašovacej funkcie – heslo sa odosiela v otvorenej podobe a systém si ukladá jeho haš - administrátor teda vidí iba odtlačky hesiel. Druhý spôsob spočíva v tom, že sa do systému odosiela iba haš hesla.

Nech je systém hesiel tvorený množinou usporiadaných dvojíc:

$$S = (userID, hash) \text{ a } hash = H(p),$$

kde H je jednosmerná hašovacia funkcia a p je heslo užívateľa, pričom pre každú dvojicu $(U, p) \in S$ systém povolí prístup.



Obr. 26. Použitie jednosmernej funkcie pri autentizácii [28]

Bezpečnosť

Tieto heslá patria do skupiny fixných hesiel a sú náchylné na množstvo útokov – útok hrubou silou, útok opakovaním, slovníkový útok. Fixné heslá sú vhodné iba pri prenose zabezpečeným komunikačným kanálom.

Ochrana spočíva vo voľbe silnejšieho hesla, použitím pomalšej hašovacej funkcie, určením doby platnosti hesla alebo pridaním tzv. soli – náhodným reťazcom znakov.

Jednorazové heslo

Jednorazové heslá (OTP - One-Time Password) predstavujú tzv. silnú autentizáciu a poskytujú ochranu systému pred pasívnymi útočníkmi. Každé heslo sa dá použiť len raz a po použití sa stáva neplatným.

Táto schéma zabezpečenia môže byť uskutočnená tromi spôsobmi [28]:

- *zdieľané zoznamy hesiel* – užívateľ a systém používajú súbor t tajných hesiel, pričom každé môže byť použité iba raz. Nevýhodou je obtiažna údržba zoznamu.
- *sekvenčná aktualizácia hesiel* – táto metóda začína spoločným heslom. Počas autentizácie pomocou hesla i vytvorí užívateľ nové heslo $i + 1$, ktoré odošle do systému zašifrované s použitím hesla i .
- schéma založená na jednosmernej funkcii – *Lamportova schéma*

Postup Lamportovej schémy:

- užívateľ A začína s heslom w
- nech H je jednosmerná funkcia a konštanta t určuje počet povolených identifikácií, po ich uplynutí je vygenerované nové heslo w
- A zašle inicializované heslo $w_0 = H^t(w)$ do systému B
- B inicializuje počítadlo $i_A = 1$
- priebeh i -tej identifikácie:
 - A vypočíta $w_i = H^{t-i}(w)$ a pošle B
 - B overí, že $i = i_A$ a prijaté heslo spĺňa podmienku $H(w_i) = w_{i-1}$
 - ak obe podmienky súhlasia, tak systém B prijaté heslo akceptuje nastaví $i_A \rightarrow i_A + 1$ a uloží w_i pre ďalšiu autentifikáciu [28]

Bezpečnosť

Lamportova schéma je odolná voči útoku opakovaním. Avšak môže byť napadnutá aktívnym útočníkom, ktorý odchyť nepoužité heslo, ktoré využije vo svoj prospech. Túto hrozbu odstraňuje schéma výzva - odpoveď.

11.4.2 Protokol typu výzva - odpoveď

Výzva - odpoveď (challenge - response) protokol - patrí medzi schémy silnej autentizácie. Je založená na overovaní identity účastníka A bez vyzradenia jeho tajomstva, čo je zabezpečené tým, že A reaguje na **časovo-premenné výzvy - parametre**. Tie sú rozdelené do niekoľkých základných skupín:

- náhodné čísla
- sekvenčné čísla
- časové pečiatky

Časovo premenné parametre často nazývané ako *nonce*, slúžia na odlišenie jednotlivých protokolov - použité môžu byť najviac jeden raz pre rovnaký účel. [28]

1. **Náhodné čísla** – systém B vygeneruje náhodné číslo, ktoré zašle účastníkovi A ako výzvu (challenge). A toto číslo pripojí k svojej odpovedi, s ktorou sa stane neoddeliteľne zviazaná a odošle ju naspäť systému B.

Problémy môžu vzniknúť s opakovaním hodnoty pri generovaní čísiel (narodeninový paradox) a pri nedostatočnej entropii v generovaní pseudonáhodných čísiel.

2. **Sekvenčné čísla** – slúžia ako jedinečné čísla, ktoré identifikujú správu. Obe stany, ktoré sa zúčastňujú komunikácie disponujú rovnakým zoznamom vygenerovaných čísiel a podľa vopred dohodnutého predpisu ich používajú v komunikácii výzva-odpoveď.

Nevýhodou tohto postupu je nutnosť udržiavať informácie o aktuálnom stave poradia čísiel po dlhú dobu. Ďalšou nevýhodou je nutnosť špeciálnych postupov pri narušení postupnosti čísiel, napr. pri zlyhaní systému.

3. **Časové pečiatky** – zaručujú jedinečnosť každej správy pridaním časovej hodnoty. Môžu byť použité aj pre časové obmedzenie prístupu. Obaja účastníci komunikácie majú možnosť skontrolovať správnosť časovej pečiatky pomocou svojho aktuálneho času. Je nutné vopred dohodnúť akceptovaný časový rozdiel prijatia správy.

Nevýhodou je nutná časová synchronizácia na oboch stranách prostredníctvom zabezpečeného kanálu a potreba udržiavať zoznam použitých časových pečiatok pre aktuálne okno.

Bezpečnosť

V protokoloch typu výzva - odpoveď sa využívajú techniky symetrickej aj asymetrickej kryptografie, kryptografické postupy založené na jednosmerných funkciách a digitálnych podpisoch. Bezpečnosť týchto schém závisí predovšetkým na bezpečnosti použitých kryptografických techník.

11.4.3 Needham-Schroeder protokol

Tento protokol bol navrhnutý v roku 1978 R. Needhamom a M. Schroederom, po ktorých dostal názov. Má dve verzie – jedna využíva symetrickú a druhá asymetrickú kryptografiu a stal sa základom pre mnohé ďalšie protokoly – napr. Kerberos.

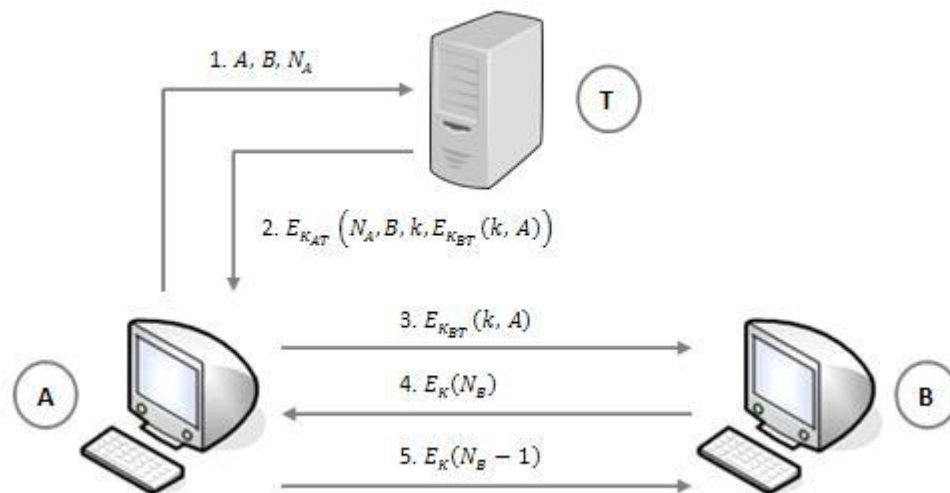
Needham-Schroeder protokol s použitím symetrickej kryptografie

Tento protokol predpokladá existenciu dôveryhodnej tretej strany T a jeho úlohou je vzájomná autentizácia oboch strán A a B a ustanovenie symetrického kľúča. [28]

E predstavuje symetrický šifrovací algoritmus, N_A a N_B sú náhodné čísla vybrané stranami A a B, k je zdieľaný kľúč ustanovený dôveryhodným serverom T pre A a B, K_{AT} je symetrický kľúč zdieľaný medzi A a T, K_{BT} je kľúč zdieľaný medzi B a T. [28]

Algoritmus

1. $A \rightarrow T: A, B, N_A$
2. $A \leftarrow T: E_{K_{AT}}(N_A, B, k, E_{K_{BT}}(k, A))$
3. $A \rightarrow B: E_{K_{BT}}(k, A)$
4. $A \leftarrow B: E_K(N_B)$
5. $A \rightarrow B: E_K(N_B - 1)$ [28]



Obr. 27. Needham-Schroeder protokol so symetrickými kľúčmi

V prvých krokoch dochádza k ustanoveniu zdieľaného symetrického kľúča pre A a B, kroky (4.) a (5.) predstavujú autentizáciu užívateľa A. Náhodne vygenerované čísla N_A a N_B zabezpečujú kontrolu čerstvosti kľúča k . V treťom kroku má protokol slabinu, nakoľko B nemá ako overiť čerstvosť kľúča, je teda náchylný na útok opakovaním.

Algoritmus útoku:

3. $A' \rightarrow B: E_{K_{BT}}(k, A)$
4. $A' \leftarrow B: E_K(N_B')$
5. $A' \rightarrow B: E_K(N_B' - 1)$ [28]

Needham-Schroeder protokol s použitím asymetrickej kryptografie

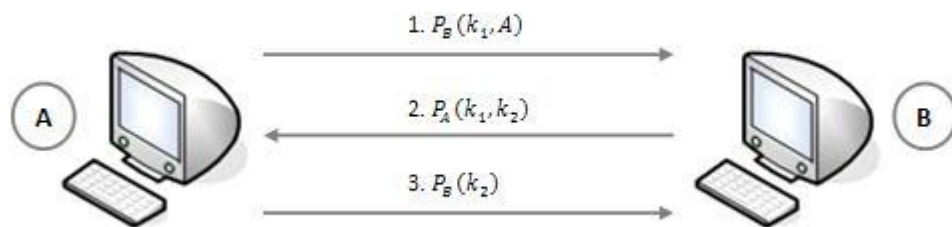
Tento protokol nevyžaduje, aby bola prítomná dôveryhodná tretia strana. Jeho úlohou je vzájomná autentizácia entít a kľúčov a vzájomná výmena kľúčov, čo pozostáva z výmeny troch správ.

$P_X(Y)$ predstavuje asymetrické šifrovanie správy Y použitím verejného kľúča X , $P_X(Y_1, Y_2)$ je zreťazené šifrovanie, k_1 a k_2 sú tajné symetrické kľúče zvolené účastníkmi A a B. [28]

Algoritmus

1. $A \rightarrow B: P_B(k_1, A)$
2. $A \leftarrow B: P_A(k_1, k_2)$
3. $A \rightarrow B: P_B(k_2)$ [28]

Po výmene týchto troch správ majú obe strany kľúče k_1 aj k_2 . Spoločný relačný kľúč môže byť vypočítaný pomocou nejakej verejnej funkcie $f(k_1, k_2)$. [28]



Obr. 28. Needham-Schroeder protokol s asymetrickými kľúčmi

11.4.4 Kerberos

Kerberos je sieťový autentizačný protokol založený na Needham-Schroeder protokole so symetrickými kľúčmi. Bol vyvinutý na MIT (Massachusetts Institute of Technology) pre projekt Athena. Protokol zabezpečuje autentizáciu entít A a B a výmenu kľúčov pomocou techník symetrickej kryptografie a zabezpečenia dôveryhodnou treťou stranou T. Dôveryhodnú tretiu stranu predstavuje zabezpečený centrálny server, ktorý plní úlohu kľúčového distribučného centra (KDC) a zdieľa tajomstvá s oboma stranami A a B.

E je symetrický šifrovací algoritmus, N_A je náhodné číslo účastníka A a T_A je jeho časová pečiatka, k predstavuje relačný kľúč určený serverom T pre účastníkov A a B, L určuje dobu životnosti tiketu. K_{AT} je spoločný kľúč medzi A a T, K_{BT} je kľúč zdieľaný B a T. [28]

Algoritmus

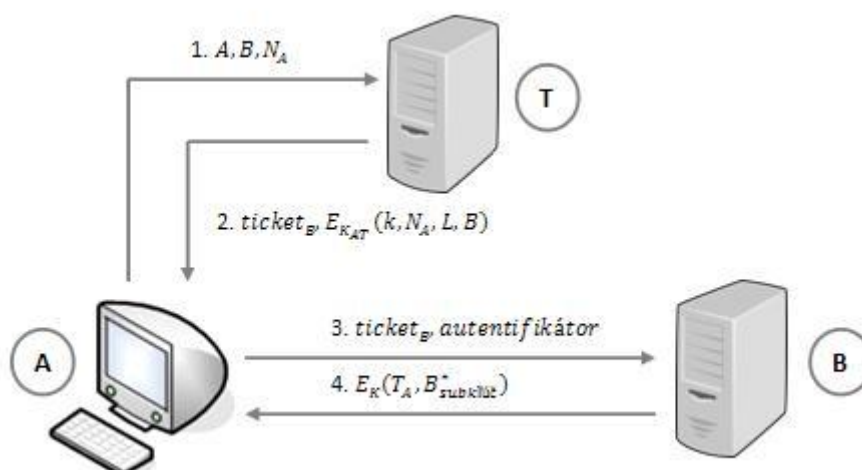
1. $A \rightarrow T: A, B, N_A$
2. $A \leftarrow T: ticket_B, E_{K_{AT}}(k, N_A, L, B)$
3. $A \rightarrow B: ticket_B, autentifikátor$
4. $A \leftarrow B: E_K(T_A, B_{subklíč})$

kde $ticket_B = E_{K_{BT}}(k, A, L)$ a $autentifikátor = E_K(A, T_A, A_{subklíč}^*)$ [28]

Postup

1. A pošle T požiadavku, ktorá obsahuje jeho identifikátor A, identifikátor B a náhodne vygenerované číslo N_A .
2. T vytvorí nový relačný kľúč k , určí životnosť tiketu L a zašle ich účastníkovi A spolu s tiketom, ktorý je zašifrovaný kľúčom K_{BT} a tiež obsahuje kľúč k a určenú životnosť L .
3. A zašle B získaný tiket a svoj autentifikátor zašifrovaný spoločným relačným kľúčom k , ktorý obsahuje identifikátor A, časovú pečiatku T_A a subkľúč.
4. B pošle A správu šifrovanú kľúčom k pre overenie správnosti, ktorá obsahuje časovú pečiatku T_A a subkľúč. [28]

V druhom kroku je vygenerovaný relačný kľúč k a kroky (3.) a (4.) predstavujú autentizačný proces.



Obr. 29. Kerberos protokol

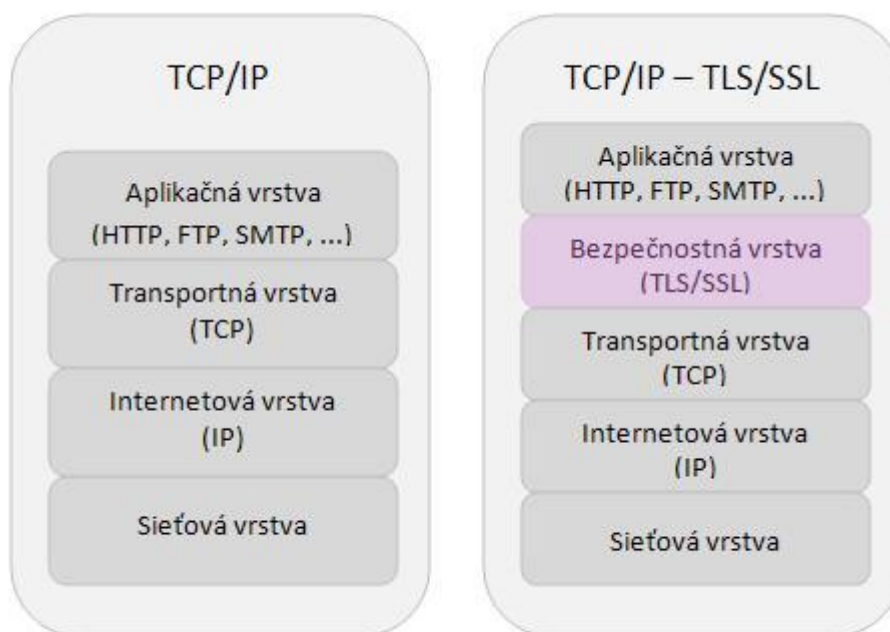
11.5 Informačná a sieťová bezpečnosť

Bezpečnosť sieťových služieb

11.5.1 TLS/SSL

SSL (Secure Sockets Layer) a **TLS (Transport Layer Security)** sú protokoly určené na bezpečnú komunikáciu na Internete. Protokoly zabezpečujú výmenu kľúčov, autentizáciu entít a poskytujú ochranu dôvernosti a integrity prenášaných údajov. Využívajú sa na bezpečné prehliadanie webových stránok, internetovú komunikáciu, výmenu správ a mnoho iných internetových aplikácií.

SSL protokol bol prvý raz definovaný v roku 1994 firmou Netscape Communications a jeho verzia 3.0 je špecifikovaná v dokumente RFC 6101. Protokol TLS bol prvý raz navrhnutý v roku 1999 a je v podstate nástupcom protokolu SSL. Verzia TLS 1.2 je štandardizovaná v dokumente RFC 5246.



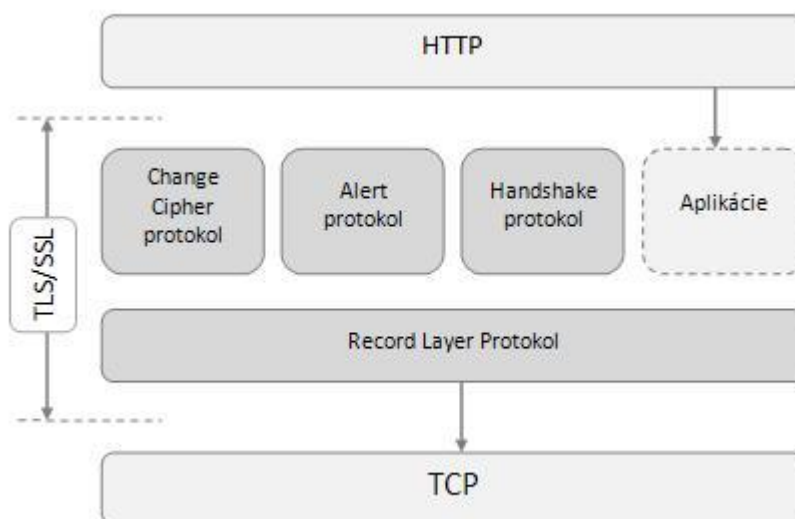
Obr. 30. Model protokolu SSL/TLS

SSL protokol predstavuje bezpečnostnú vrstvu, ktorá pracuje medzi transportnou a aplikačnou vrstvou. Skladá sa z dvoch vrstiev:

- *Record Layer protokol* pracuje nad spoľahlivým transportným protokolom TCP a používa sa na zapuzdrenie dát vyšších vrstiev
- štyri subprotokoly:
 - *Change Cipher protokol*
 - *Alert protokol*
 - *Handshake protokol*
 - *Application data protokol* [33]

Record Layer protokol poskytuje dôvernosť a integritu dát. Prijíma dáta z vyšších vrstiev a spracováva ich:

- Fragmentácia – rozdeľuje dáta na bloky s maximálnou veľkosťou 2^{14} bajtov
- Komprimácia – musí byť bezstratová, nesmie zvýšiť dĺžku o viac ako 1024 bajtov a presiahnuť maximálnu veľkosť bloku
- Výpočet MAC – pomocou nejakej hašovacej funkcie (MD5, SHA)
- Šifrovanie – pomocou dohodnutého algoritmu



Obr. 31. Vrstvový systém TLS/SSL

Change Cipher protokol zabezpečuje prechod v šifrovacej stratégii. Obsahuje jednu správu, ktorá je zaslaná klientovi aj serveru, aby ich informovala, že nasledujúce správy budú zašifrované novou zjednanou šifrou a kľúčom.

Alert protokol umožňuje odosielanie chybových správ. Ak má správa úroveň – *fatal*, tak je spojenie okamžite ukončené.

Handshake protokol predstavuje celé jadro protokolu SSL. Zabezpečuje autentizáciu účastníkov, kľúčovú výmenu a dohodu na kryptografickom algoritme.

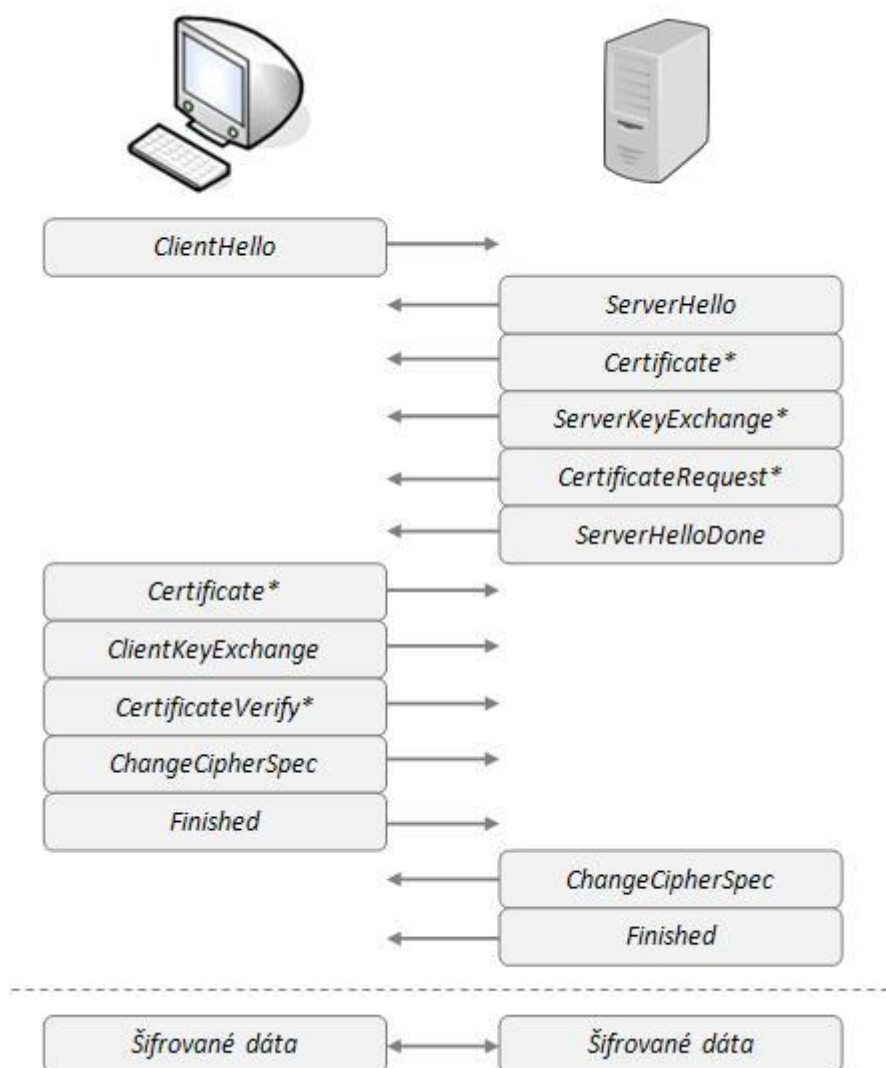
Postup:

- *ClientHello* – klient pošle v správe záznam o podporovanej verzii SSL, náhodné číslo, ID spojenia, zoznam kompresných metód a zoznam podporujúcich šifrovacích algoritmov.
- *ServerHello* – server zašle správu s údajmi o zvolenej verzii SSL, náhodné číslo, ID spojenia a zvolené šifrovacie algoritmy.
- *Certificate* – server posiela certifikát.
- *ServerKeyExchange** – posiela server iba ak nemá certifikát, prípadne má certifikát len pre digitálny podpis, umožňuje kľúčovú výmenu.
- *CertificateRequest** – server môže poslať žiadosť o certifikát.
- *ServerHelloDone* – server ohlasuje ukončenie inicializačných správ a čaká na odpoveď od klienta.
- *Certificate** – klient odosiela v prípade, ak server požaduje certifikát.
- *ClientKeyExchange* – obsahuje *PreMasterSecret*, z ktorého sa spolu s náhodnými číslami vypočíta *MasterSecret* - zdieľané tajomstvo, pomocou ktorého sa počítajú všetky ostatné kľúče.
- *CertificateVerify** – používa sa na overenie klientovho certifikátu.
- *ChangeCipherSpec* – klient oznamuje serveru, že budú nasledujúce správy šifrované.
- *Finished* - klient zašle šifrovanú správu.
- Server sa snaží správu rozšifrovať, v prípade neúspechu je komunikácia ukončená.
- *ChangeCipherSpec* a *Finished* - rovnako server zasiela klientovi oznámenie a šifrovanú správu.
- Inicializačná fáza je ukončená a obe strany sú pripravené na bezpečnú komunikáciu na aplikačnej úrovni, ktorá je šifrovaná pomocou kľúčov vygenerovaných v handshake fáze. [33]

SSL podporuje tieto kryptografické algoritmy:

- výmena kľúčov - D-H, RSA, FORTEZZA
- podpisové schémy - DSA, RSA
- MAC algoritmus - MD5, SHA
- symetrické šifry - RC2, RC4, DES, 3DES, DES40, FORTEZZA [33]

Dokument RFC 6101 podrobne špecifikuje podmienky použitia a vzájomné kombinácie šifier určených pre vzájomnú autentifikáciu, výmenu kľúčov a bezpečnú komunikáciu.



Obr. 32. Jednotlivé fázy Handshake protokolu

TLS protokol je založený na rovnakých postupoch komunikácie medzi klientom a serverom ako protokol SSL. Rozdiely sa týkajú predovšetkým kryptografických šifrier a spôsobov ich použitia:

- podpora AES, ECDH a ECDSA
- nové módy blokových šifrier - okrem CBC aj CCM a GCM (Galois/Counter Mode)
- nové hašovacie algoritmy - SHA-224, SHA-256, SHA-384 a SHA-512
- odstránené rôzne chyby pri šifrovaní, vylepšené spôsoby autentizácie, zmenené kombinácie šifrier, a i.

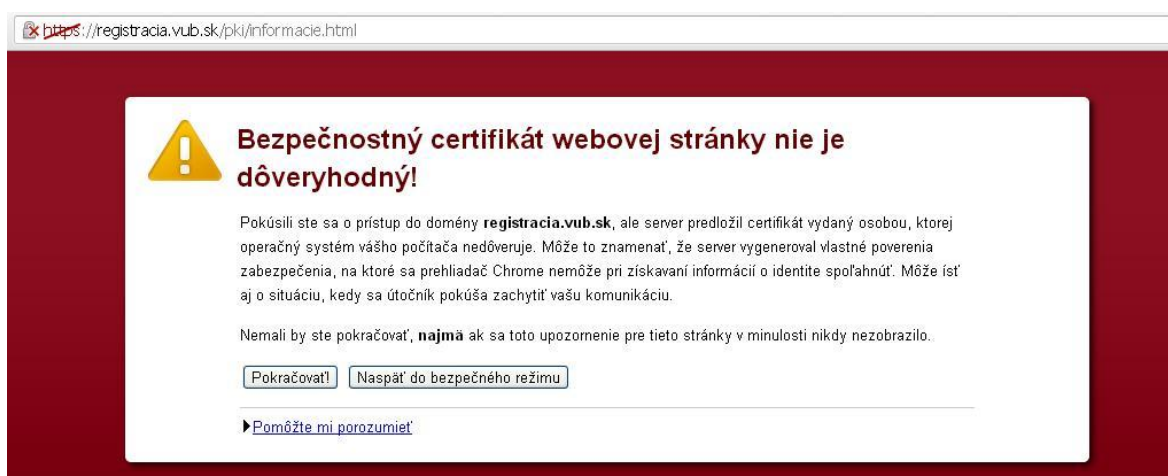
11.5.2 HTTPS

HTTPS (Hypertext Transfer Protocol Secure) je zabezpečený internetový protokol určený na komunikáciu medzi webovým prehliadačom a serverom. HTTPS je nadstavba štandardného protokolu http rozšírená o šifrovanie pomocou protokolu SSL/TLS. Dokument RFC 2818 popisuje použitie HTTP nad protokolom TLS.

Inicializácia - najskôr prebehne SSL/TLS handshake protokol, potom HTTP požiadavka, pričom všetky údaje sú posielané ako TLS aplikačné dáta. [35]

Uzatvorenie spojenia - TLS poskytuje bezpečné ukončenie spojenia, po prijatí výstrahy pre ukončenie už nie sú vymenené žiadne dáta. TLS môže ukončiť spojenie aj bez varovnej správy, ale nesmie byť znovu použitá rovnaká relácia. [35]

HTTP/TLS používa na spojenie port 443, aby bolo hneď jasné, o aký typ spojenia sa jedná. URI formát sa líši pomocou identifikátora „https“. [35]



Obr. 33. Bezpečnostné upozornenie v prehliadači Google Chrome

Bezpečnosť

V prípade nezabezpečeného spojenia väčšina prehliadačov informuje užívateľov vo forme správy alebo dialógového okna a v adresovom riadku sa zobrazujú informácie o zabezpečení servera.

Bezpečnosť HTTPS závisí predovšetkým od spôsobu implementácie v prehliadači a od použitých kryptografických techník.

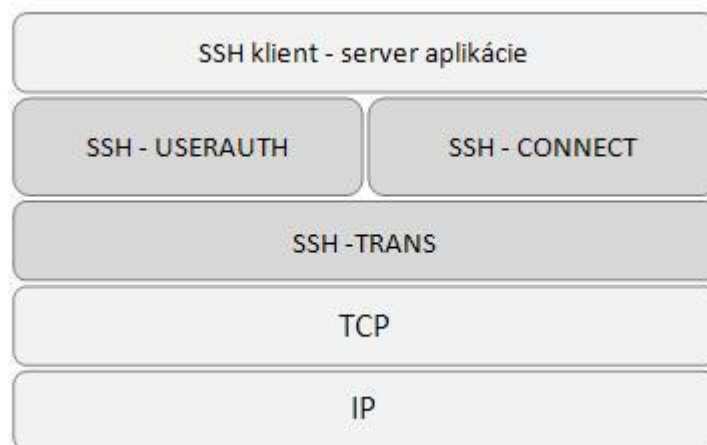
11.5.3 SSH

SSH (Secure Shell) je sieťový protokol určený pre zabezpečené vzdialené pripojenie a iné sieťové služby cez nezabezpečený komunikačný kanál. Používa sa ako náhrada protokolu Telnet a ostatných nezabezpečených protokolov.

Bol navrhnutý v roku 1995 a má dve verzie – SSH-1, ktorá je už dnes zastaraná a SSH-2 verzia, ktorá je štandardizovaná a popísaná v dokumentoch RFC 4250 – 4254.

SSH je vrstvomý protokol, ktorý sa skladá z troch častí:

- SSH-TRANS - protokol poskytuje overovanie servera, dôvernosť a integritu
- SSH-USERAUTH - protokol vykonáva autentizáciu klienta
- SSH-CONNECT - protokol spojenia



Obr. 34. SSH vrstvomý model

SSH-TRANS - je bezpečný protokol transportnej vrstvy, ktorý zvyčajne beží nad TCP/IP protokolom na porte 22. Na tejto vrstve sú dohodnuté metódy na výmenu kľúčov, symetrické kryptografické algoritmy, autentizačné a hašovacie algoritmy, prípadne kompresia dát. Protokol nevykonáva autentizáciu klienta a je navrhnutý tak, aby všetky procesy prebehli v dvoch, maximálne troch kolách vzájomnej výmeny správ.

1. Klient nadväzuje spojenie.
2. Po nadviazaní spojenia posielajú obe strany identifikačné reťazce vo formáte:
SSH-protoversion-softwareversion SP comments CR LF,
kde sa vzájomne predstavia a v prípade, že ich verzie nie sú kompatibilné, spojenie ukončia.
3. Nasleduje vzájomná výmena paketu *SSH_MSG_KEXINIT*, pomocou ktorého sa vyberie vhodný algoritmus pre výmenu kľúčov. Paket obsahuje informácie o použitých šifrovacích algoritmoch, algoritmoch pre kľúčovú výmenu, kompresné metódy a náhodný prvok „cookie“.
4. Pre výpočet kľúčov je použitá haš funkcia v tvare:

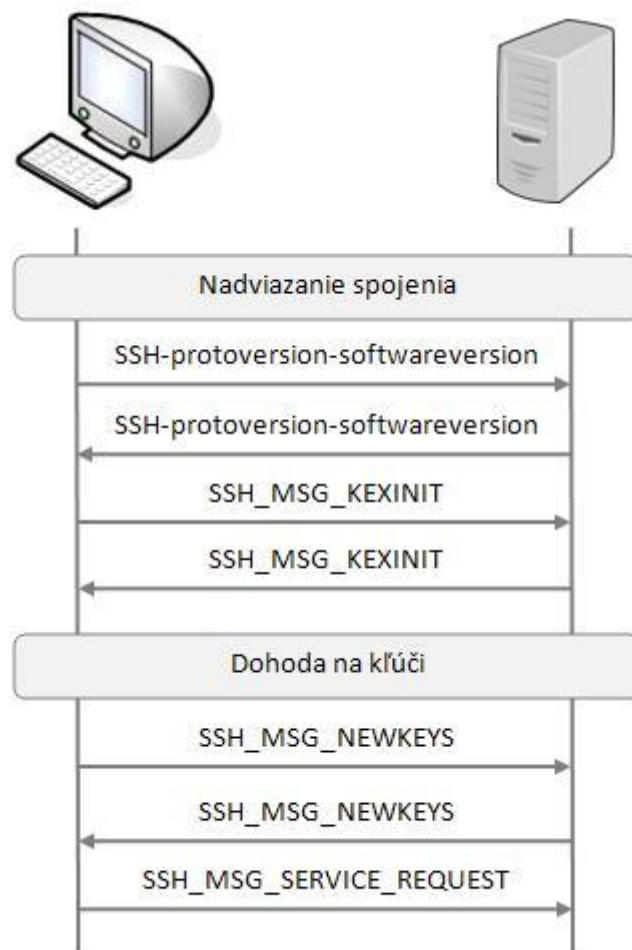
$$HASH(K || H || "X" || session_id) \quad (65)$$

kde *HASH*, *K*, *H*, *session_id* sú dohodnuté parametre počas prvej vzájomnej výmeny a *X* predstavuje ASCII znak, napr. „A“ s hodnotou 65. Na výmenu kľúčov je použitý obvykle D-H algoritmus a výsledkom je zdieľaný kľúč *K* a haš *H*.

5. Výmena kľúčov končí výmenou správy *SSH_MSG_NEWKEYS*, po ktorej sú všetky údaje šifrované novými kľúčmi. Následne klient posiela svoju požiadavku na službu pomocou správy *SSH_MSG_SERVICE_REQUEST*. [36]

SSH protokol podporuje tieto kryptografické algoritmy:

- šifrovacie algoritmy – 3DES, Blowfish, Twofish, AES, Serpent, RC4, IDEA, Cast 128
- MAC algoritmy – SHA-1, MD5
- Asymetrické algoritmy – DSS, RSA, D-H pre výmenu kľúčov



Obr. 35. SSH transportný protokol

SSH-USERAUTH – je autentizačný protokol, ktorého cieľom je overenie užívateľskej identity. Protokol pracuje nad zabezpečeným protokolom transportnej vrstvy a poskytuje jeden tunel pre spojovací protokol.

1. Pre žiadosť o autentizáciu zasiela klient správu `SSH_MSG_USERAUTH_REQUEST`, ktorá musí obsahovať meno klienta, požadovanú službu a podporované metódy autentizácie.
2. Server ponúkne klientovi rôzne autentizačné metódy, z ktorých si klient môže vybrať. Server podporuje tri rôzne situácie po autentizácii:
 - `SSH_MSG_USERAUTH_FAILURE` – neúspešné overenie
 - `SSH_MSG_USERAUTH_SUCCESS` – úspešné overenie
 - `SSH_MSG_USERAUTH_BANNER` – napr. po prekročení maximálneho limitu neúspešných pokusov o overenie

3. SSH poskytuje niekoľko prístupových metód:

- „publickey“ je jediná doporučená metóda, ktorú musia obsahovať všetky implementácie. Klient pošle požiadavku obsahujúcu podpis a podporovaný algoritmus pre autentizáciu, server overí kľúč a správnosť podpisu. Ak obe súhlasia, autentizácia prebehla úspešne.
- „password“ je metóda overovania pomocou hesla. Heslo je zasielané ako textová správa, kódované pomocou ISO-10646 UTF-8. Server môže požadovať zmenu hesla a záleží na implementácii, v akej podobe si ukladá a vedie databázu hesiel.
- „hostbased“ nie je odporúčaná metóda, nakoľko nie je dostatočne zabezpečená [37]

SSH-CONNECT – je protokol spojenia, ktorý slúži na vzdialené prihlasovanie a spúšťanie príkazov a pracuje nad transportnou vrstvou. Všetky spojenia sú uskutočňované pomocou číslovaných kanálov.

1. Každá strana môže otvoriť kanál zaslaním správy `SSH_MSG_CHANNEL_OPEN`, pričom jedno spojenie môže obsahovať viacero kanálov. Proti strana reaguje zaslaním správy o prijatí alebo odmietnutí kanála.
2. Prenos dát sa uskutočňuje pomocou správ `SSH_MSG_CHANNEL_DATA`.
3. Keď účastník nechce odosielať ďalšie dáta kanálom, pošle správu `SSH_MSG_CHANNEL_EOF`, pričom kanál ostáva ešte otvorený pre dáta odosielané v opačnom smere.
4. Ak chce niektorý z účastníkov ukončiť kanál, odošle druhej strane správu `SSH_MSG_CHANNEL_CLOSE`, ktorá tiež odpovie ukončovacou správou, po ktorej bude kanál uzavretý. [38]

Bezpečnosť

V SSH-1 verzii boli zistené bezpečnostné nedostatky, ktoré dávali priestor pre MITM útoky. Vzhľadom na tieto chyby bola verzia SSH-1 označená za zastaranú a dnes už väčšina serverov podporuje bezpečnejšiu verziu SSH-2.

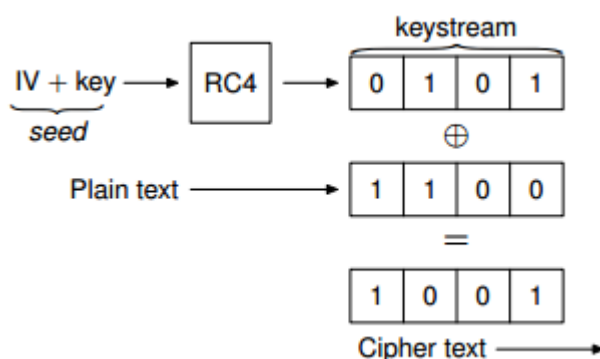
Bezpečnosť bezdrôtových sietí

Bezdrôtové siete WLAN sú založené na štandardoch IEEE 802.11, súčasťou ktorých sú odporúčené metódy zabezpečenia.

11.5.4 WEP

WEP (Wired Equivalent Privacy) je bezpečnostný algoritmus pre zabezpečenie bezdrôtových sietí, ktorý tvoril súčasť pôvodného štandardu IEEE 802.11. Algoritmus bol navrhnutý pre ochranu dát pred nechceným odpočúvaním.

WEP je symetrická prúdová šifra, ktorá používa pre šifrovanie algoritmus RC4 a kľúč s veľkosťou 64 alebo 128 bitov. Kľúč sa skladá zo 40-bitového alebo 104-bitového užívateľského kľúča a z dynamicky sa meniaceho inicializačného vektora (IV), ktorý sa pripája k heslu a spolu majú veľkosť 24 bitov. Výstup tvorí pseudonáhodný algoritmus (tzv. „keystream“), ktorý sa pomocou operácie XOR spočíta s otvoreným textom, rozšíreným o kontrolný súčet CRC32. Výsledkom je šifrovaný text. [39] Dešifrovanie pozostáva z analogických operácií.



Obr. 36. Princíp šifrovania [39]

Bezpečnosť

Algoritmus WEP bol prelomený v roku 2001 a kvôli viacerým slabším sa viac neodporúča používať. S použitím bežného počítača a dešifrovacieho programu je možné WEP prelomiť v priebehu niekoľkých sekúnd. WEP mal niekoľkých nástupcov ako WEP2 alebo WEPplus, pri ktorých boli odstránené niektoré nedostatky, ale ani tie nie sú dostatočne bezpečné.

11.5.5 WPA

WPA (Wi-Fi Protected Access) je bezpečnostný protokol, ktorý bol vydaný ako náhrada slabého WEP protokolu a predstavuje akýsi prechodný produkt k pripravovanému protokolu WPA2.

WPA pracuje vnútorne ako WEP, ale pre zvýšenie bezpečnosti je doplnený o protokol TKIP a autentizačný protokol IEEE 802.1x.

TKIP (Temporal Key Integrity Protocol) je protokol, ktorý bol vyvinutý na odstránenie nedostatkov protokolu WEP:

- per-packet-key – dynamické generovanie kľúčov
- MIC (Message Integrity Code) – integrita dát
- predĺžený IV [40]

TKIP využíva šifrovací algoritmus RC4, aby bolo možné previesť aktualizáciu WEP pomocou softvéru na bezpečnejší protokol. Na šifrovanie používa 128-bitový kľúč dynamicky sa meniaci pre každý paket. Inicializačný vektor má dĺžku 48 bitov, čím sa znížil počet odosielaných paketov. MIC algoritmus Michael nahrádza cyklický kód CRC a zabezpečuje ochranu pred MITM útokmi. [40]

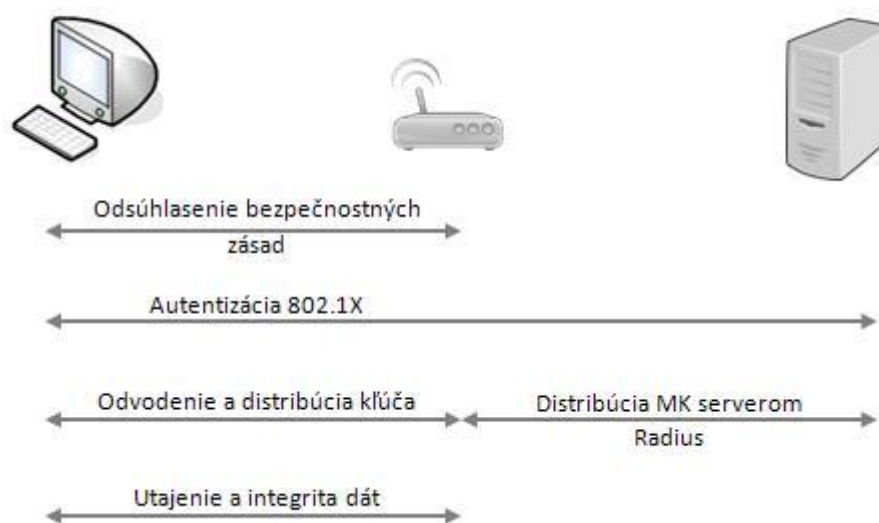
Autentizačný protokol IEEE 802.1X sa používa pre vzájomnú autentizáciu medzi klientom a prístupovým bodom (access point). Jeho architektúru tvoria tri funkčné entity:

- supplicant pripojenia k sieti
- autentizátor zaisťujúci riadenie prístupu - EAP protokol
- autentizačný server, ktorý prevádza autorizačné rozhodnutie - Radius

Pre domácnosti a jednoduchšie siete sa používa PSK (Pre-shared key).

Bezpečnosť

Napriek pridaným bezpečnostným opatreniam sa WPA neodporúča používať, pretože sú známe možné útoky proti nemu a je ľahké prelomiť ho.



Obr. 37. Jednotlivé fázy štandardu 802.11i [40]

WPA2

WPA2 označovaný tiež ako 802.11i je dodatok ku štandardu 802.11 pre bezdrôtové siete Wi-Fi. Bol schválený v roku 2004, čím nahradil predchádzajúci štandard WEP.

WPA2 má oproti svojim predchodcom novú architektúru postavenú na silnom zabezpečení šifrovacími algoritmami AES a CCMP.

Protokol CCMP (Counter-Mode/CBC-MAC Protocol) zabezpečuje autentizáciu a integritu dát a nahrádza TKIP protokol. Používa šifru AES v režime CCM so 128-bitovým kľúčom a 128-bitovými blokmi na zabezpečenie utajenia a MIC na integritu. [40]

Bezpečnosť

WPA2 sa považuje za bezpečný vďaka silnej šifre AES.

ZÁVER

Novodobé informačné technológie prenikli takmer do každej oblasti nášho života. Stretávame sa s nimi denne, či už v podobe mobilných telefónov, výberov finančných hotovostí alebo digitálnej televíznej stanice. Počítače už tiež nie sú iba výsadou veľkých organizácií a podnikov, ale zaujali svoje miesto v mnohých domácnostiach. S nimi zavítal do nášho súkromia veľmi silný novodobý komunikačný nástroj – Internet. Prostredníctvom Internetu vykonávame bankové transakcie, nakupujeme v rôznych internetových obchodoch alebo len tak komunikujeme s priateľmi. Preto je potrebné zabezpečiť tieto formy elektronickej komunikácie pred nežiaducimi vplyvmi, akými sú napr. krádež peňazí alebo súkromia.

Moderná kryptológia sa preto stala neoddeliteľnou súčasťou informačných technológií. Je nám dôverne známy spôsob, ktorým si môžeme napr. vybrať finančnú hotovosť z bankomatu a spoliehame sa na skutočnosť, že bez krádeže karty a PIN kódu je táto metóda absolútne bezpečná. Málokto však vie, že za všetkým stoja v prvom rade silné kryptografické systémy. Šifrovanie dnes nadobudlo omnoho silnejšie postavenie než v minulosti.

Táto práca má za úlohu nielen pomôcť študentom získať prehľad o moderných kryptologických systémoch a ich využití v niektorých oblastiach informačných technológií, ale aj poukazuje na potrebu dostatočného zabezpečenia v rôznych sférach elektronickej komunikácie. Rovnakú funkciu má plniť aj webová prezentácia, ktorá vznikla ako výstup tejto bakalárskej práce.

CONCLUSION

Modern information technologies have spread into almost every area of our lives. We meet them every day, in the form of mobilephones, withdraw cash from the automated teller machine or digital television. Computers are no longer the privilege of large organizations and companies but they have taken place in many households. This trend has introduced a very powerful tool, the internet, into our lives. We use the internet for banking, shopping or just chatting with our friends. That is why it is necessary to protect these forms of electronic communication against unwanted events, such as loss of money or privacy.

Modern cryptology has become an inseparable part of information technologies. Everybody knows the way how to withdraw cash from the automated teller machine and trusts that without stealing the credit card and PIN this method is absolutely safe. Not many people know that this safety is ensured by strong cryptographic systems. Nowadays encryption has obtained much stronger position than it was in the past.

This thesis aims not only to help students obtain an overview of modern cryptologic systems and their use in some areas of information technologies, but also it emphasizes the need of sufficient security in different areas of electronic communication. This is also the main aim of the web presentation which was created as the output of the thesis.

ZOZNAM POUŽITEJ LITERATÚRY

- [1] ZELENKA, Josef. *Ochrana dat: kryptologie*. Vyd. 1. Hradec Králové: Gaudeamus, 2003, 198 s. ISBN 80-704-1737-4.
- [2] VONDRUŠKA, Pavel. *Kryptologie, šifrování a tajná písma*. 1. vyd. Praha: Albatros, 2006, 340 s. Oko. ISBN 80-000-1888-8.
- [3] Informačná bezpečnosť: Čo to je informačná bezpečnosť. BÍRO, Peter. *Ministerstvo financií SR* [online]. 27.03.2008, 06.12.2012 [cit. 2013-05-31]. Dostupné z: <http://www.informatizacia.sk/informacna-bezpecnost/>
- [4] ČANDÍK, Marek. *Základy informační bezpečnosti*. Vyd. 1. Zlín: Univerzita Tomáše Bati ve Zlíně, 2004, 107 s. Učební texty vysokých škol (Univerzita Tomáše Bati ve Zlíně). ISBN 80-731-8218-1.
- [5] HAGSTRÖM, Åsa. *Ranking the AES Finalists* [online]. February 8, 2001 [cit. 2013-05-31]. Dostupné z: <http://teal.gmu.edu/courses/ECE636/AES/AsaHagstrom.PDF>
- [6] ZÁKLADY MODULÁRNEJ ARITMETIKY, TEÓRIA ČÍSEL. In: *Aplikovaná kryptografia* [online]. 2003 [cit. 2013-05-31]. Dostupné z: http://www.kemt.fei.tuke.sk/predmety/KEMT414_AK/_materialy/Cvicenia/kryp_1_2.pdf
- [7] OCHODKOVÁ, Eliška. *Matematické základy kryptografických algoritmov* [online]. 2011 [cit. 2013-05-31]. Dostupné z: http://mi21.vsb.cz/sites/mi21.vsb.cz/files/unit/mat_zaklady_kryptografickych_algoritmu.pdf
- [8] PIPER, F a Sean MURPHY. *Kryptografie*. 1. vyd. v českém jazyce. Překlad Pavel Mondschein. Praha: Dokořán, 2006, 157 s. ISBN 80-736-3074-5.
- [9] SYMETRICKÉ ŠIFRY. In: *Aplikovaná kryptografia* [online]. 2003 [cit. 2013-05-31]. Dostupné z: http://www.kemt.fei.tuke.sk/predmety/KEMT414_AK/_materialy/Prednasky/Obr4.pdf
- [10] FIPS PUB 197: Advanced Encryption Standard (AES). In: *Federal Information Processing Standards Publications (FIPS PUBS)*. November 26, 2001. Dostupné z: <http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf>

- [11] ŠIFROVANIE S VEREJNÝM KLÚČOM. In: *Aplikovaná kryptografia* [online]. 2003 [cit. 2013-05-31]. Dostupné z: http://www.kemt.fei.tuke.sk/predmety/KEMT414_AK/_materialy/Cvicenia/kryp_7.pdf
- [12] PELLEGRINI, A., V. BERTACCO a T. AUSTIN. *Fault-based attack of RSA authentication: Proceedings / Design, Automation and Test in Europe* [online]. 2010 [cit. 2013-05-31]. ISBN 978-1-4244-7054-9. Dostupné z: <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5456933&isnumber=5456897>
- [13] KLÍMA, V. a T. ROSA. Kryptologie pro praxi – schémata ElGamal. *Sdělovací technika* [online]. 2004, ST 4/2004 [cit. 2013-05-31]. Dostupné z: http://crypto-world.info/klima/2004/st_2004_06_12_12.pdf
- [14] GEYER, Lukáš. *Eliptické křivky v kryptografii: Elliptic curves in cryptography* [online]. Brno: Vysoké učení technické, Fakulta elektrotechniky a komunikačních technologií, 2010 [cit. 2013-05-31]. 1 elektronický optický disk [CD-ROM / DVD]. Dostupné z: https://www.vutbr.cz/www_base/zav_prace_soubor_verejne.php?file_id=31002. Bakalářská práce.
- [15] ANSI X9.62. *The Elliptic Curve Digital Signature Algorithm (ECDSA)*. ASC X9 Secretariat - American Bankers Association, December 1997. Dostupné z: <http://cs.ucsb.edu/~koc/ccs130h/notes/ecdsa-cert.pdf>
- [16] KLÍMA, V. a T. ROSA. Kryptologie pro praxi - principy ECC. *Sdělovací technika* [online]. 2005, ST 10/2005 [cit. 2013-05-31]. Dostupné z: http://crypto-world.info/klima/2005/ST_2005_10_12_13.pdf
- [17] KLÍMA, V. a T. ROSA. Kryptologie pro praxi – použití ECC. *Sdělovací technika* [online]. 2005, ST 11/2005 [cit. 2013-05-31]. Dostupné z: http://crypto.hyperlink.cz/files/ST_2005_11_x_y.pdf
- [18] *A Survey of the Elliptic Curve Integrated Encryption Scheme* [online]. Journal of Computer Science and Engineering, August 2010 [cit. 2013-05-31]. ISSN 2043-9091. Dostupné z: <http://digital.csic.es/bitstream/10261/32671/1/V2-I2-P7-13.pdf>
- [19] STANEK, Martin. *Základy kryptologie*. verzia 0.16. 12. decembra 2004. Dostupné z: <http://www.dcs.fmph.uniba.sk/~stanek/crypto/main2.pdf>

- [20] PINKAVA, J. Hashovací funkce v roce 2004. *Crypto-World* [online]. 15. září 2004, roč. 6, 9/2004, s. 15-18 [cit. 2013-05-31]. Dostupné z: http://crypto-world.info/pinkava/clanky/hash_2004.pdf
- [21] FIPS PUB 180-4: Secure Hash Standard (SHS). In: *Federal Information Processing Standards Publications (FIPS PUBS)*. March 2012. Dostupné z: <http://csrc.nist.gov/publications/fips/fips180-4/fips-180-4.pdf>
- [22] FIPS PUB 180-1: SECURE HASH STANDARD. In: *Federal Information Processing Standards Publications (FIPS PUBS)*. 1995 April 17. Dostupné z: <http://www.itl.nist.gov/fipspubs/fip180-1.htm>
- [23] BERTONI, G., J. DAEMEN a M. PEETERS. The Keccak sponge function family. [online]. © 2008-2013 [cit. 2013-05-31]. Dostupné z: <http://keccak.noekeon.org/>
- [24] KLÍMA, V. Hašovací funkce a kódy: Jak se melou data. *Počítačová bezpečnost* [online]. duben 1999, s. 44-46 [cit. 2013-05-31]. Dostupné z: <http://crypto-world.info/klima/1999/chip-1999-04-44-46.pdf>
- [25] RFC 2104: HMAC: Keyed-Hashing for Message Authentication. In: *The Internet Engineering Task Force (IETF)*. February 1997. Dostupné z: <http://tools.ietf.org/html/rfc2104#section-3>
- [26] KLÍMA, V. a T. ROSA. Kryptologie pro praxi – DSA, ECDSA. *Sdělovací technika* [online]. 2004, ST 4/2004 [cit. 2013-05-31]. Dostupné z: http://crypto.hyperlink.cz/files/ST_2004_04_17_17.pdf
- [27] FIPS PUB 186-3: Digital Signature Standard (DSS). In: *Federal Information Processing Standards Publications (FIPS PUBS)*. June 2009. Dostupné z: http://csrc.nist.gov/publications/fips/fips186-3/fips_186-3.pdf
- [28] MENEZES, Alfred J. *Handbook of applied cryptography*. Vyd. 1. Boca Raton: CRC Press, 1997, 780 s. ISBN 08-493-8523-7. Dostupné z: <http://cacr.uwaterloo.ca/hac/>
- [29] BITTO, Ondřej. *Šifrování a biometrika aneb tajemné bity a dotyky*. Vyd. 1. Kralice na Hané: Computer Media, 2005, 168 s. ISBN 80-866-8648-5.
- [30] ANSI X9.63: Elliptic Curve Key Agreement and Key Transport Schemes. In: *Public Key Cryptography for the Financial Services Industry*. July 5, 1998. Dostupné z: <ftp://ftp.iks-jena.de/mitarb/lutz/standards/ansi/X9/x963-7-5-98.pdf>

- [31] RFC 4880: OpenPGP Message Format. In: *The Internet Engineering Task Force (IETF)*. November 2007. Dostupné z: <http://tools.ietf.org/html/rfc4880>
- [32] PGP Desktop. *Symantec (PGP)* [online]. SkyNet, 1997 [cit. 2013-05-31]. Dostupné z: <http://www.pgp.cz/pgp-desktop>
- [33] RFC 6101: The Secure Sockets Layer (SSL) Protocol Version 3.0. In: *Internet Engineering Task Force (IETF)*. August 2011. Dostupné z: <http://tools.ietf.org/html/rfc6101>
- [34] RFC 5246: The Transport Layer Security (TLS) Protocol Version 1.2. In: *Internet Engineering Task Force (IETF)*. August 2008. Dostupné z: <http://tools.ietf.org/html/rfc5246>
- [35] RFC 2818: HTTP Over TLS. In: *Internet Engineering Task Force (IETF)*. May 2000. Dostupné z: <http://tools.ietf.org/html/rfc2818>
- [36] RFC 4253: The Secure Shell (SSH) Transport Layer Protocol. In: *Internet Engineering Task Force (IETF)*. January 2006. Dostupné z: <http://tools.ietf.org/html/rfc4253>
- [37] RFC 4252: The Secure Shell (SSH) Authentication Protocol. In: *Internet Engineering Task Force (IETF)*. January 2006. Dostupné z: <http://tools.ietf.org/html/rfc4252>
- [38] RFC 4254: The Secure Shell (SSH) Connection Protocol. In: *Internet Engineering Task Force (IETF)*. January 2006. Dostupné z: <http://tools.ietf.org/html/rfc4254>
- [39] BITTAU, A., M. HANDLEY a J. LACKEY. The final nail in WEP's coffin. 2006 *IEEE Symposium on Security and Privacy (S)* [online]. IEEE, 2006, s. 386-400 [cit. 2013-06-01]. DOI: 10.1109/SP.2006.40. Dostupné z: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1624028>
- [40] LEHEMBRE, Guillaume. Bezpečnost Wi-Fi – WEP, WPA a WPA2. *Hakin9* [online]. 2006, 1/2006 [cit. 2013-06-01]. Dostupné z: http://www.hsc.fr/ressources/articles/hakin9_wifi/hakin9_wifi_CZ.pdf

ZOZNAM POUŽITÝCH SYMBOLOV A SKRATIEK

AES	Advanced Encryption Standard
ASCII	American Standard Code for Information Interchange
CA	Certificate Authority
CBC	Cipher Block Chaining
CCMP	Counter Cipher Mode with Block Chaining Message Authentication Code Protocol
CFB	Cipher Feedback
CRC	Cyclic Redundancy Check
DES	Data Encryption Standard
D-H	Diffie–Hellman
DSA	Digital Signature Algorithm
DSS	Digital Signature Standard
ECB	Electronic Code Book
ECC	Elliptic Curve Cryptosystems
ECDH	Elliptic Curve Diffie-Hellman
ECDLP	Elliptic Curve Discrete Logarithm Problem
ECDSA	Elliptic Curve Digital Signature Algorithm
ECIES	Elliptic Curve Integrated Encryption Scheme
ECMVQ	Elliptic Curve Menezes-Qu-Vanstone
EDE	Encrypt-Decrypt-Encrypt
EESSI	European Electronic Signature Standardization Initiative
ETSI	European Telecommunications Standards Institute
FIPS PUB	Federal Information Processing Standards Publications
GCD	Greatest Common Divisor
HMAC	keyed-Hash Message Authentication Code

HTTPS	Hypertext Transfer Protocol Secure
ID	Identification
IDEA	International Data Encryption Algorithm
IEEE	Institute of Electrical and Electronics Engineers
IETF	Internet Engineering Task Force
IP	Internet Protocol
IV	Initialization Vector
KDC	Key Distribution Center
MAC	Message Authentication Code
MD	Message-Digest Algorithm
MIC	Message Integrity Code
MIT	Massachusetts Institute of Technology
MITM	Man In The Middle útok
NIST	National Institute of Standards and Technology
OFB	Output Feedback
OTP	One-Time Password
PGP	Pretty Good Privacy
PKCS	Public-Key Cryptography Standards
PKI	Public Key Infrastructure
RFC	Request for Comments
RIPEMD	RACE Integrity Primitives Evaluation Message Digest
RSA	(Rivest-Shamir-Adelman) šifrovací algoritmus
SHA	Secure Hash Algorithm
SSH	Secure Shell
SSL	Secure Sockets Layer
TCP	Transmission Control Protocol

TKIP	Temporal Key Integrity Protocol
TLS	Transport Layer Security
URI	Uniform Resource Identifier
WEP	Wired Equivalent Privacy
Wi-Fi	Wireless Fidelity
WLAN	Wireless Local Area Network
WPA	Wi-Fi Protected Access
XOR	Exclusive OR

ZOZNAM OBRÁZKOV

Obr. 1. Princíp symetrického šifrovania	14
Obr. 2. Princíp asymetrického šifrovania	15
Obr. 3. Skytalé	17
Obr. 4. Prúdová šifra.....	38
Obr. 5. Bloková šifra.....	39
Obr. 6. Režim ECB [9]	39
Obr. 7. Režim CBC [9]	40
Obr. 8. Režim OFB [9]	41
Obr. 9. Režim CFB [9].....	41
Obr. 10. Feistelova bloková šifra [9]	43
Obr. 11. Schéma šifrovania DES [9]	45
Obr. 12. Jedna runda DES [9].....	46
Obr. 13. 3DES s tromi kľúčmi.....	47
Obr. 14. Schéma šifry IDEA.....	48
Obr. 15. Zmena stavu bloku AES [10]	50
Obr. 16. Operácia ByteSub [10]	50
Obr. 17. Operácia ShiftRow() [10]	51
Obr. 18. Operácia MixColumn() [10].....	51
Obr. 19. Geometrická interpretácia sčítania dvoch bodov krivky $E P + Q = R$ [15].....	58
Obr. 20. Keccak Pi funkcia [23]	66
Obr. 21. MAC algoritmus [28]	67
Obr. 22. Princíp digitálneho podpisu	69
Obr. 23. PGP Desktop Email [32]	81
Obr. 25. PGP Net Share [32]	82
Obr. 24. PGP Whole Disk Encryption [32]	82
Obr. 26. Použitie jednosmernej funkcie pri autentizácii [28].....	84
Obr. 27. Needham-Schroeder protokol so symetrickými kľúčmi.....	87
Obr. 28. Needham-Schroeder protokol s asymetrickými kľúčmi.....	88
Obr. 29. Kerberos protokol.....	89
Obr. 30. Model protokolu SSL/TLS	90
Obr. 31. Vrstvový systém TLS/SSL	91
Obr. 32. Jednotlivé fázy Handshake protokolu.....	93

Obr. 33. Bezpečnostné upozornenie v prehliadači Google Chrome.....	94
Obr. 34. SSH vrstvový model.....	95
Obr. 35. SSH transportný protokol	97
Obr. 36. Princíp šifrovania [39]	99
Obr. 37. Jednotlivé fázy štandardu 802.11i [40].....	101

ZOZNAM TABULIEK

Tab. 1. Výsledné hodnotenie finalistov AES [5]	25
Tab. 2. Prehľad štandardov PKCS	26
Tab. 3. Základné axiómy pre konečné teleso F [14]	57
Tab. 4. Porovnanie dĺžky kľúčov RSA a ECC [18].....	61
Tab. 5. Porovnanie vlastností algoritmov SHA [21]	64
Tab. 6. ECSDA bezpečnostné parametre [27].....	73
Tab. 7. Odporúčané algoritmy k eliptickým krivkám [27]	80

ZOZNAM PRÍLOH

P I: CD s textom práce a zdrojovými kódmi