UNIVERZITA PARDUBICE

FAKULTA EKONOMICKO-SPRÁVNÍ

# BAKALÁŘSKÁ PRÁCE

2025 ONDŘEJ RUML

Univerzita Pardubice
Fakulta ekonomicko-správní

# Porovnání vybraných kryptografických algoritmů a jejich využití v praxi

Bakalářská práce

2025 Ondřej Ruml

Univerzita Pardubice
Fakulta ekonomicko-správní
Akademický rok: 2024/2025

# ZADÁNÍ BAKALÁŘSKÉ PRÁCE
(projektu, uměleckého díla, uměleckého výkonu)

| | |
|---|---|
| Jméno a příjmení: | **Ondřej Ruml** |
| Osobní číslo: | **E22422** |
| Studijní program: | **B0688A140004 Informatika a systémové inženýrství** |
| Specializace: | **Informační a bezpečnostní systémy** |
| Téma práce: | **Porovnání vybraných kryptografických algoritmů a jejich využití v praxi** |
| Zadávající katedra: | **Ústav matematiky a kvantitativních metod** |

## Zásady pro vypracování

Cílem práce je popsat a porovnat vybrané kryptografické algoritmy a vyhodnotit jejich efektivitu a aplikovatelnost v kyberprostoru.
Osnova:
– Základní terminologie.
– Kryptografické algoritmy a jejich matematický popis.
– Porovnání kryptografických algoritmů.
– Aplikace kryptografických algoritmů.

Rozsah pracovní zprávy: **cca 35 stran**
Rozsah grafických prací:
Forma zpracování bakalářské práce: **tištěná/elektronická**

Seznam doporučené literatury:

BURDA, Karel. *Úvod do kryptografie*. Brno: Akademické nakladatelství CERM, 2015. ISBN 978-80-7204-925-7
LEPKA, Karel. *Základy elementární teorie čísel*. Brno: Munipress, 2023. ISBN 978-80-280-0423-1
PELZL, Jan, PAAR, Christof. *Understanding Cryptography: A Textbook for Students and Practitioners*. Berlin: Springer, 2010. ISBN 978-3-642-04101-3
STALLINGS, William. *Cryptography and Network Security, Eighth Edition, Global Edition*. Pearson, 2022. ISBN 978-1-292-43748-4

Vedoucí bakalářské práce: **Mgr. Libor Koudela, Ph.D.**
Ústav matematiky a kvantitativních metod

Datum zadání bakalářské práce: **1. září 2024**
Termín odevzdání bakalářské práce: **30. dubna 2025**

L.S.

**prof. Ing. Jan Stejskal, Ph.D.** v.r.
děkan

**prof. Ing. Jitka Komárková, Ph.D.** v.r.
garant studijního programu

V Pardubicích dne 1. září 2024

Prohlašuji:

Práci s názvem Porovnání vybraných kryptografických algoritmů a jejich využití v praxi jsem vypracoval samostatně. Veškeré literární prameny a informace, které jsem v práci využil, jsou uvedeny v seznamu použité literatury.

Byl jsem seznámen s tím, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, zejména se skutečností, že Univerzita Pardubice má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona, a s tím, že pokud dojde k užití této práce mnou nebo bude poskytnuta licence o užití jinému subjektu, je Univerzita Pardubice oprávněna ode mne požadovat přiměřený příspěvek na úhradu nákladů, které na vytvoření díla vynaložila, a to podle okolností až do jejich skutečné výše.

Beru na vědomí, že v souladu s § 47b zákona č. 111/1998 Sb., o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších předpisů, a směrnicí Univerzity Pardubice č. 7/2019 Pravidla pro odevzdávání, zveřejňování a formální úpravu závěrečných prací, ve znění pozdějších dodatků, bude práce zveřejněna prostřednictvím Digitální knihovny Univerzity Pardubice.

V Pardubicích dne 23. 04. 2025

Ondřej Ruml v.r.

# PODĚKOVÁNÍ

Mé poděkování patří především panu Mgr. Liboru Koudelovi, PhD. za odborné vedení, cenné rady a trpělivost při zpracování této práce. Dále bych chtěl poděkovat panu RNDr. Ing. Oldřichu Horákovi, PhD. za jeho čas a věcné připomínky při konzultacích. Nakonec bych chtěl poděkovat své rodině, která mi byla oporou během celého studia.

# ANOTACE

Cílem bakalářské práce je představit vybrané šifrovací algoritmy a popsat jejich vlastnosti pomocí matematického a grafického aparátu, následně doporučit jejich výběr pro běžné případy užití, čímž práce poslouží jako praktický návod pro čtenáře při výběru šifrovacích algoritmů. Dále je cílem práce demonstrovat implementovatelnost vybraných šifrovacích algoritmů ve vývojovém prostředí.

# KLÍČOVÁ SLOVA

kryptografie, informační bezpečnost, teorie čísel, řízení přístupu, RSA

# TITLE

Comparison of selected cryptographic algorithms and their use in practice

# ANNOTATION

The aim of the bachelor thesis is to present selected encryption algorithms and show their properties using mathematical and graphical apparatus, then recommend their selection for common use cases, thus serving as a practical guide for the reader in the selection of encryption algorithms. Furthermore, the thesis aims to demonstrate the implementability of the selected encryption algorithms in a development environment.

# KEYWORDS

cryptography, information security, number theory, access control, RSA

# Obsah

# Seznam ilustrací a tabulek

# Seznam zkratek

NÚKIB – Národní úřad pro kybernetickou a informační bezpečnost

NIST – National Institute of Standards and Technology

IS – informační systém

SW – software

DES – Data Encryption Standard

AES – Advanced Encryption Standard

RSA – Rivest, Shamir, Adleman

BFA – Brute Force Attack

ECC – Elliptic Curve Cryptography

NSD – největší společný dělitel

$\mathbb{P}$ – množina prvočísel

$\mathbb{N}$ – množina přirozených čísel

$\mathbb{Z}$ – množina celých čísel

$\mathbb{R}$ – množina reálných čísel

mod – operace modulo

$\wedge$ – logický operátor AND

$\vee$ – logický operátor OR

$\oplus$ – logický operátor XOR

# Úvod

Informace jsou významnou komoditou. Ať už se jedná o osobní či firemní informace, je důležité mít na paměti, že pokud se informace dostanou k neoprávněným osobám, může to mít závažné ekonomické, integritní nebo právní následky. Je tedy zapotřebí citlivé informace utajovat tak, aby k nim měly přístup pouze prověřené osoby. Toto platí jak pro reálný svět například v podobě státních prověrek nebo fyzických klíčů, tak pro svět digitální, kde se k zajištění bezpečnosti informací a zpráv využívají šifrovací algoritmy.

Práce je rozdělena do čtyř kapitol. První kapitola seznamuje čtenáře se základními výrazy a poznatky z oblastí kryptologie, matematiky a informačních systémů, které jsou nezbytné pochopení navazujících kapitol. Druhá kapitola pojednává o klasických a moderních šifrovacích algoritmech a jejich matematickém zápise. Ve třetí kapitole je věnován prostor pro porovnání vybraných šifrovacích algoritmů na základě stanovených kritérií. Poslední kapitola se věnuje aplikovatelností a implementovatelností vybraných šifrovacích algoritmů v kyberprostoru prostřednictvím ukázky případu užití a vývojového prostředí.

Cílem této práce je popsat a porovnat vybrané šifrovací algoritmy a vyhodnotit jejich efektivitu a aplikovatelnost v kyberprostoru.

# 1 Základní terminologie

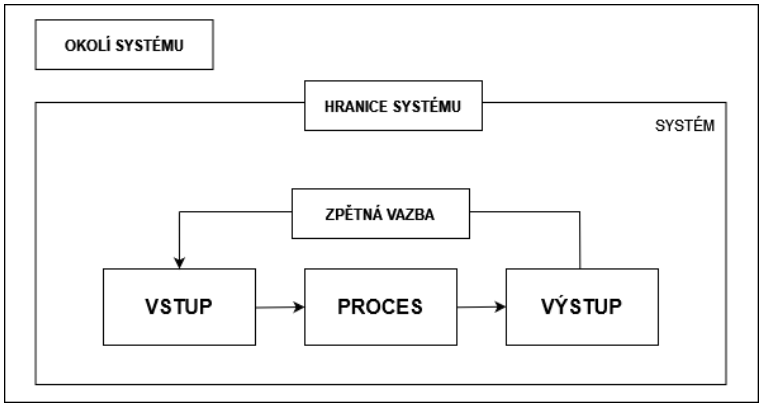
Nejprve je nutné si uvést základní oblasti, které budou doprovázet celou bakalářskou práci.

## 1.1 Informační a bezpečnostní systémy

Kryptografie a informační systémy jsou neoddělitelnou dvojicí, jelikož kryptografie je primární způsob zabezpečení informačních systémů. Pro pochopení způsobů zabezpečení dat a informací, je nutné vysvětlil základní principy informačních systémů a s tím související požadavky na jejich bezpečnost.

### 1.1.1 Systém

Systém je uspořádaná množina dále nedělitelných prvků a jejich vzájemných vazeb. Tyto vazby můžeme nazvat relacemi, které propojují jednotlivé prvky. Systém lze tedy vyjádřit jako množinu vstupních a výstupních veličin, prvků a vazeb. Systémy můžeme klasifikovat z hlediska vztahu k okolí, realitě, způsobu chování, atd. [5]




Obrázek 1: Základní schéma systému

*Zdroj: vlastní zpracování podle [26]*

### 1.1.2 Informační systém

Vymezit termín informační systém je poměrně obtížné, neboť existuje mnoho definic vymezujících tento pojem. Obecně se jedná o propojení uživatelů, hardwaru, softwaru, dat a případně metod, které dohromady tvoří funkční celek zabezpečující sběr, ukládání a zpracování dat pro potřeby uživatelů v systému. [9] Z hlediska informatiky jsou data, informace a znalosti klíčovými pojmy informačních systémů.

- **Data** jsou potenciální nosiče informace popisující nějaká fakta, které zatím nemusí mít konkrétní význam. Objemově jich je nejvíce.
- **Informace** jsou údaje, kterým byl přidán význam, či kontext.
- **Znalost** je ucelená soustava informací využitelná pro řešení konkrétních problémů. [7]

Informační systémy, které nedisponují žádnými bezpečnostními mechanismy jsou snadno napadnutelné, proto v rámci jejich zabezpečení hovoříme o požadavcích na bezpečnost IS, kterými se zabývá oblast informační bezpečnosti. [8]

Proniknutí neoprávněné osoby do informačního systému může ohrozit celý jeho životní cyklus. V případě napadení podnikového informačního systému může nastat únik firemních dat, což představuje ekonomickou, integritní a právní hrozbu pro společnost. Proto je zásadní, aby každý informační systém splňoval základní požadavky na informační bezpečnost.

### 1.1.3 Základní požadavky na informační bezpečnost

Základní pilíře informační bezpečnosti jsou požadavky na důvěrnost (anglicky confidentiality), dostupnost (availability) a integrity (integrity), které dohromady tvoří tzv. triádu CIA. [8]

- **Důvěrnost** je požadavek, aby informace byla dostupná pouze vybraným, ověřeným uživatelům. Narušením důvěrnosti je například odcizení a zneužití přihlašovacích údajů. K zajištění důvěrnosti informací využíváme šifrování, vícefázové ověření a školení.[8], [11]
- **Dostupnost** je požadavek zajišťující přístup ověřeným uživatelům k informacím v okamžiku jejich potřeby. Narušením dostupnosti často bývá výpadek serveru, selhání jiné výpočetní techniky, či výpadek elektrického proudu. K zajištění dostupnosti lze využít možnosti aktivního zálohování, přepojování na sekundární počítače nebo v případě výpadku elektřiny využívání záložních zdrojů. [8], [11]
- **Integrita** je požadavek na korektnost a celistvost informací, jehož hlavním účelem je předejít nežádoucím úpravám informací. Narušením integrity může být změna údajů v dokumentech, porušením sektorů pevného disku nebo narušitelem změněné přihlašovací údaje. [8]

V oblasti informační bezpečnosti lze však kromě základní triády CIA hovořit i o dalších nadstavbových požadavcích, jako jsou nepopíratelnost (non-repudiation), odpovědnost (accountability), hodnověrné popření (plausible deniability) a mnohá další. [10]

Při procesu dosažení informační bezpečnosti mluvíme takzvaně o aplikaci bezpečnostních mechanismů, které mohou být:

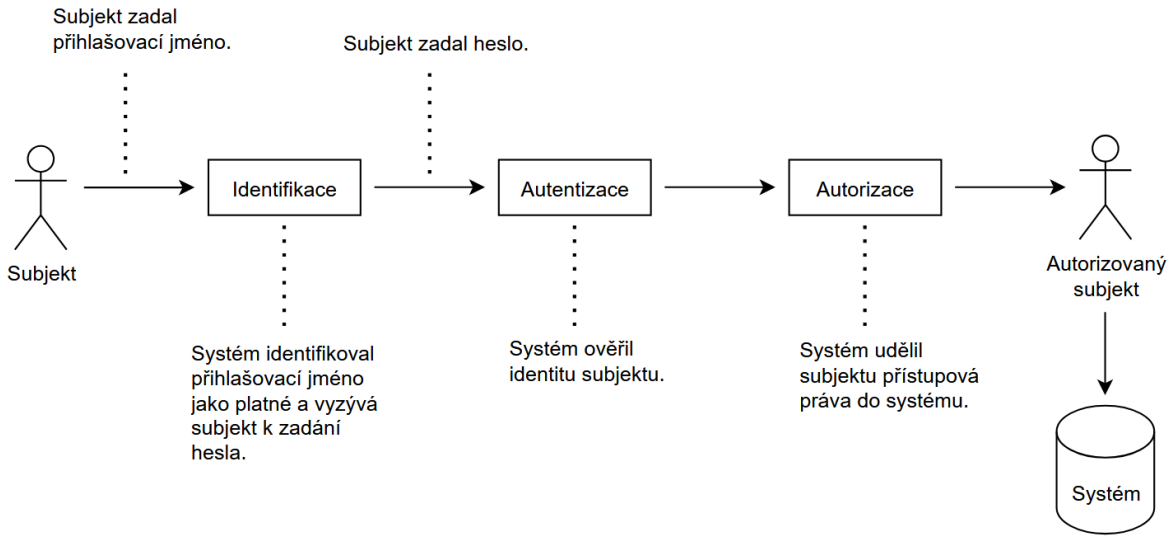
- Fyzického charakteru (fyzická ochrana)
- Logického charakteru (softwarová ochrana)
- Technického charakteru (hardwarová ochrana)
- Administrativního charakteru (školení, normy, zákony) [8]

## 1.2 Řízení přístupu k informacím

Abychom mohli splnit cíle informační bezpečnosti, musíme uživatelům systému nastavit patřičná oprávnění na základě kterých budou moci přistupovat k informacím v systému. [8], [4]

Při nastavování práv přístupu k informacím, je nejprve zapotřebí si identifikovat všechny uživatele systému a stanovit, ke kterým informacím budou moci přistupovat. Proces řízení přístupu zahrnuje tři klíčové fáze:

- **Identifikace**, při které se uživatel hlásí k identitě a odpovědnosti. Může se jednat o přihlašovací jméno, identifikační číslo, PIN nebo příslušnost ke skupině. Ve chvíli, kdy se subjekt systému identifikuje, začne mít odpovědnost za další provedené úkony. Počítačový systém rozlišuje jednotlivé identifikované subjekty podle unikátního identifikačního čísla nebo symbolu, na základě kterého monitoruje jejich aktivitu. [8]
- **Autentizace**, při které se ověřuje tvrzení subjektu o jeho identitě. Nejčastějšími formami autentizace bývají hesla, USB klíče nebo biometrické složky. [8] Samotná autentizace může být vícefázová, kdy je po uživateli požadováno předložit více autentizačních prvků, typicky to bývá kombinace a hesla a následného ověření v příslušné aplikaci v mobilním telefonu.
- **Autorizace**, při které je autentizovanému uživateli přidělen přístup do systému. [4]




Obrázek 2: Proces řízení přístupu

*Zdroj: vlastní zpracování podle [8]*

## 1.3 Kryptologie

Kryptologie je vědní obor zabývající se myšlenkou a studiem principů k zajišťování bezpečnosti zpráv, tvorbou šifrovacích metod a hledáním nástrojů k jejich prolomení. Obor kryptologie se dělí na kryptografii a kryptoanalýzu. [1]

Pro termínovou konzistenci vymezíme následující pojmy:

- **Kryptografie** je vědecký obor, který se zabývá utajováním zpráv přes přenosový kanál použitím primárně matematických metod. [1]
- **Kryptoanalýza** je komplementárním oborem ke kryptografii, který se zabývá překonáváním šifrovacích metod. [3]
- **Abeceda** je množina specifických znaků. [7]
- **Zpráva** je posloupnost rozlišitelných znaků, ve které je zakódovaná informace. [1]
- **Šifrování** je proces zabezpečení obsahu zprávy. Pojem šifrování je dobré nezaměňovat s pojmem kódování, které pouze znamená přenesení znaků do jiné abecedy. [1] [7]
- **Dešifrování** je převod šifrované zprávy do otevřeného textu. [1]
- **Klíč** je informace sloužící k šifrování a dešifrování zpráv. [1]
- **Otevřený text** je nezašifrovaný text. [1]
- **Algoritmus** je definovaný postup, jak převést počáteční stav na vyřešený. Jinými slovy, jedná se o podrobný postup, kterým lze vyřešit konkrétní problém. [6]

- **Kryptosystém** je systém, ve které lze provádět šifrovací operace. [1]
- **Hybridní kryptosystém** je kryptosystém, do něhož je zakompovaný symetrický, a zároveň asymetrický kryptografický algoritmus. [1], [8]
- **Veřejný a privátní (soukromý) klíče** jsou páry klíčů využívané v asymetrické kryptografii, kde jeden z nich je využívaný při šifrování a druhý při dešifrování. Samotná jejich transformace a způsob použití závisí na konkrétních šifrovacích algoritmech. [4]
- **Hashovací funkce** je nástoj (algoritmus), který převádí otevřený text na pevně dlouhý řetězec bitů (hash) [3]

## 1.4 Matematické principy kryptografie

Většina šifrovacích metod má kořeny v matematických disciplínách, jejichž osvojení je klíčové k pochopení principů kryptografie.

### 1.4.1 Výroková logika

Jak v informatických, tak matematických vědách se neobejdeme bez axiomatického odvození výrokových proměnných. V kryptografii provádíme logické operace při práci s bity (proměnnými, které mohou nabývat pouze hodnot 0 nebo 1).

- **Konjunkce (AND)** znamená „a zároveň“. Konjunkcí říkáme, že tvrzení je pravdivé, právě tehdy když jsou oba spojované výroky pravdivé. Konjunkci a,b symbolicky zapisujeme jako $a \wedge b$. [1]
- **Disjunkce (OR)** znamená „nebo“. Disjunkcí říkáme, že tvrzení je pravdivé, právě tehdy, když je alespoň jeden spojovaný výrok pravdivý. Disjunkci a,b symbolicky zapisujeme jako $a \vee b$. [1]
- **Exkluzivní disjunkce (XOR)** znamená „exkluzivní nebo“. Exkluzivní disjunkcí říkáme, že tvrzení je pravdivé, právě tehdy, když vstupní výrok nabývá jedinečné hodnoty. Exkluzivní disjunkci a,b symbolicky zapisujeme jako $a \oplus b$ . [1] Operace XOR je významná v hardwarově implementovaných šifrovacích algoritmech, díky její rychlé proveditelnosti. [4]
- **Negace** znamená „popření“. Negací vyjadřujeme pravdivostní opak původního výroku. Negaci $a$ symbolicky zapisujeme $\neg a$ . [1]

Tabulka 1: Operátory výrokové logiky

| $a$ | $b$ | $a \wedge b$ | $a \vee b$ | $a \oplus b$ | $\neg a$ | $\neg b$ |
|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 | 1 | 0 | 1 |
| 0 | 1 | 0 | 1 | 1 | 1 | 0 |
| 0 | 0 | 1 | 0 | 0 | 1 | 1 |

### 1.4.2 Substituce

Substituce je operace, kdy je vstupní proměnné $x$ přiřazené jiné číslo $y = S(x)$ dle definovaných pravidel. [1]

Tabulka 2: Substituční tabulka

| $x$ | 1 | 2 | 3 | 4 | 5 | $n$ |
|---|---|---|---|---|---|---|
| $y$ | 3 | 6 | 9 | 12 | 15 | $m$ |

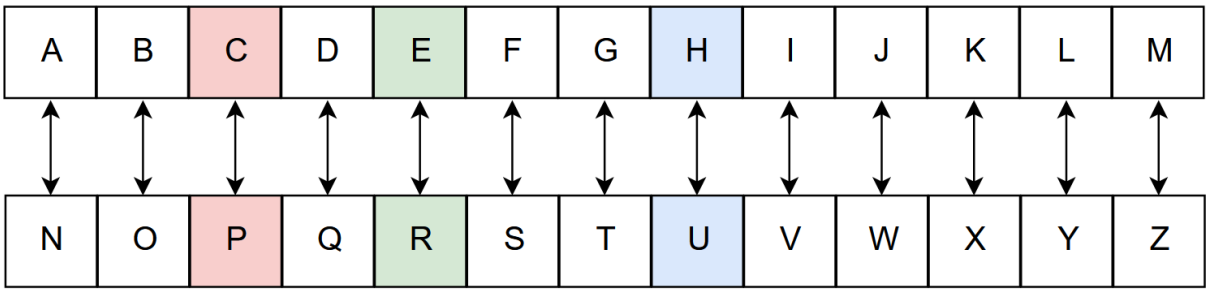
Vztah ovšem nemusí být vždy snadno odvoditelný a substituční tabulka může pro hodnotu $y$ vygenerovat pseudonáhodná čísla. V takovém případě bychom potřebovali znát všechny hodnoty, aby se mohl s jistotou určit vztah.

Za konkrétní případy užití substituce v šifrovacích algoritmech jsou posuny vlevo a vpravo (o počet $n$ bitů).

### 1.4.3 Rotace

Rotace je operaci, při které rotujeme blok $n$ čísel o $k$ pozic směrem vlevo nebo vpravo, za podmínky, že $0 \leq k < n$ . [1]

Příkladem elementárního šifrovacího algoritmu, který využívá rotaci je ROT13, kde ve 26 znakové abecedě rotujeme o 13 pozic [36], jak je znázorněno v Obrázku 4, kde je zpráva „UPCE“ zašifrovaná jako „HCPR“.




Obrázek 3: Operace rotace v šifrovacím algoritmu ROT13

*Zdroj: vlastní zpracování*

## 1.5 Vybrané okruhy z teorie čísel

Jednou z nejzásadnějších matematických disciplín, na které stojí část moderní kryptografie je teorie čísel, která se zabývá studiem celých čísel, prvočísel a jejich vlastností. [2].

1. **Dělitelnost**

**Definice**: Nechť $a, b \in \mathbb{Z}, b > 0$, pak $\exists\, q, r \in \mathbb{Z}$, pro které platí: $a = bq + r, 0 \leq r < b$ .

2. **Největší společný dělitel**

NSD je největší číslo, kterým můžeme beze zbytku vydělit 2 a více čísel.

V kryptografii se nalezení NSD často provádí pomocí Eukleidova algoritmu.

**Definice**: Nechť $a, b \in \mathbb{Z}$

1. Pokud $a = b$, potom $a = \text{NSD}$
2. Pokud $a \neq b$, potom od většího čísla odečteme menší.
3. Proces opakujeme, dokud $a = b$.

Př. $\frac{a}{b} = \frac{36}{16} \rightarrow \frac{36-16}{16} \rightarrow \frac{20}{16} \rightarrow \frac{20-16}{16} \rightarrow \frac{4}{16} \rightarrow \frac{4}{16-4} \rightarrow \frac{4}{12} \rightarrow \frac{4}{12-4} \rightarrow \frac{4}{8} \rightarrow \frac{4}{8-4} \rightarrow \frac{4}{4}$, $\text{NSD} = 4$ [2]

3. **Faktorizace**

Faktorizace je operace rozložení čísla na součin menších čísel, využívaná v kryptografii v souvislosti s prvočísly. [2]

## 4. Modulární aritmetika

Modulární aritmetika je matematická disciplína, která se zabývá zbytkovými třídy a kongruencí (algebraickou podobností) celých čísel.

Pro určení kongruence je nutné znát operaci modulo:

$$a \bmod n = a - n\left\lfloor\frac{a}{n}\right\rfloor$$

**Věta**: Nechť $a, b \in \mathbb{Z}, n \in \mathbb{N}$, pak

$$a \equiv b (\bmod n) \Leftrightarrow \frac{n}{(a-b)}$$

**Příklad**: $10 \equiv 14 \pmod 4 \Leftrightarrow \frac{4}{(10-14)} = -\frac{4}{4}$. Čísla 10 a 14 jsou kongruentní $mod$ 4, protože rozdíl $10 - 14 \ = \ -4$ je dělitelný číslem 4. [2]

## 5. Prvočísla

Prvočíslo je každé přirozené číslo větší než jedna, které lze dělit pouze jedničkou nebo samo sebou. Prvočísla hrají podstatnou roli v kryptografii díky jejich vlastnosti vytvářet matematicky složité problémy. Opakem prvočísel jsou čísla složená. [2]

Pro mnohé kryptografické algoritmy je nutností si zvolit více velkých prvočísel, což není úplně triviální úloha, protože je obtížné určit, zdali nějaké velké číslo nelze dělit jiným.

Formálně neexistuje žádný optimální vzorec pro určování prvočísel, jelikož jsou buď neúplné nebo výpočetně velmi náročné, proto při určování čísel s velkým počtem cifer budeme spoléhat na Millerův-Rabinův test prvočíselnosti, jehož princip vychází z malé Fermatovy věty: [37]

**Definice**: Nechť $a = 2k + 1,\ a, k \in \mathbb{Z},\ n \in \mathbb{P},\ 1 < a < n - 1$

Jestliže $\mathrm{NSD}(a, n) = 1$, pak:

$$a^{n-1} \equiv 1 \pmod n$$

Z tohoto vztahu kongruence lze odvodit následující výraz, který využijeme při Miller-Rabinově testu:

$$a^{n-1} - 1 \equiv 0 \ (mod\ n)$$

1. Nejprve faktorizujme $a^{n-1} - 1$:

$$\left(a^{\frac{n-1}{2}} - 1\right) \cdot \left(a^{\frac{n-1}{2}} + 1\right) \equiv 0 \pmod{n}$$

2. Faktorizujme do té doby, než bude exponent lichý:

$$\left(a^{\frac{n-1}{2^k}} - 1\right) \cdot \left(a^{\frac{n-1}{2^k}} + 1\right) \cdot \ldots \left(a^{\frac{n-1}{2}} + 1\right) \equiv 0 \pmod{n}$$

3. Nyní počítejme s tím, že exponent $\frac{n-1}{2^k}$ je lichý.

**Věta:** Pokud $n$ dělí alespoň jeden z výrazů faktorizace, potom $\mathrm{P}(n \in \mathbb{P}) \sim \frac{3}{4}$.

**Příklad**: Nechť $n = 23$. Potřebujeme dokázat, zdali $n \in \mathbb{P}$.

Zvolme takové $a$, které splňuje omezující podmínku: $1 < a < n - 1$.

Nechť $a = 2$ :

$$\left(2^{\frac{23-1}{2}} - 1\right) \cdot \left(2^{\frac{23-1}{2}} + 1\right) \equiv 0 \pmod{n}$$

…

$$(2^{11} - 1) \cdot (2^{11} + 1) \equiv 0 \pmod{n}.$$

Exponent je lichý, nyní vydělíme výrazy faktorizace hodnotou $n$. Číslo $(2^{11} - 1) = 2047$ je dělitelné $n$, $\Rightarrow n \in \mathbb{P}$.

Provedení testu manuálním výpočtem je u velkých čísel velmi časově náročné. Pro účely automatizace je v Příloze 1 uveden kód programu na výpočet prvočíselnosti v programovacím jazyce Python.

### 6. Eulerova funkce

**Definice**: $\varphi(n)$: $\mathbb{N} \to \mathbb{N}$, $n \in \mathbb{N}, k \in \mathbb{N}$, kde $1 \leq k \leq n \wedge \mathrm{NSD}(k,n) = 1$.

Pokud je znám rozklad argumentu $n$, tak dokážeme hodnotu Eulerovy funkce vypočítat pomocí kanonického rozkladu: [2]

$$\varphi(n) = n \cdot (1 - \frac{1}{p_1}) \cdot (1 - \frac{1}{p_2}) \cdot (1 - \frac{1}{p_3}) \cdot \ldots \cdot (1 - \frac{1}{p_m}) = n \cdot \prod_{\substack{p|n \\ p \in \mathbb{P}}} (1 - \frac{1}{p})$$

Výpočet rozkladu bez znalosti argumentu zatím není v rozumném čase možné, což je jedním z důvodů, proč je Eulerova funkce obsažena v některých šifrovacích algoritmech. [2], [3]

## 1.6 Vybrané pojmy z abstraktní algebry

Abstraktní algebra je matematická disciplína, která zkoumá algebraické struktury, jako jsou pole a grupy, které jsou významné při popisování šifrovacích algoritmů založených na eliptických křivkách.

### 1. Těleso

Těleso je množina T se dvěma operacemi sčítání a násobení, která splňuje následující axiomy:

1. $\forall a, b \in T: a + b = b + a$ …komutativita sčítání
2. $\forall a, b, c \in T: (a + b) + c = a + (b + c)$…asociativita sčítání
3. $\exists 0 \in T, \forall a \in T: a + 0 = a$…existence nulového prvku
4. $\forall a \in T, \exists -a \in T: a + (-a) = 0$ …existence opačného prvku
5. $\forall a, b, c \in T: (a \cdot b) \cdot c = a \cdot (b \cdot c)$ …asociativita násobení
6. $\exists 1 \in T, \forall a \in T: 1 \cdot a = a \cdot 1 = a$ …existence jednotkového prvku
7. $\forall a \in T, a \neq 0, \exists a^{-1} \in T: a \cdot a^{-1} = a^{-1} \cdot a = 1$…existence inverzního prvku
8. $\forall a, b, c \in T: (a + b) \cdot c = a \cdot c + b \cdot c$…distributivita
9. $\forall a, b, c \in T: a \cdot (b + c) = a \cdot b + a \cdot c$…distributivita

Poznámka: 8. axiom = 9. axiomu $\Leftrightarrow \forall a, b \in T: a \cdot b = b \cdot a$ [38]

## 2. Grupa

Grupa je množina prvků pouze s jednou binární operací sčítání nebo násobení.

**Definice**: Množina G s pouze jednou binární operací sčítání je grupa, právě tehdy, když:

1. $\forall a, b, c \in G: (a + b) + c = a + (b + c)$
2. $\exists 0 \in G, \forall a \in G: a + 0 = 0 + a = a$
3. $\forall a \in G, \exists -a \in G: a + (-a) = (-a) + a = 0$

**Definice**: Množina G s pouze jednou binární operací násobení je grupa, právě tehdy, když:

1. $\forall a, b, c \in G: (a \cdot b) \cdot c = a \cdot (b \cdot c)$
2. $\exists 1 \in G, \forall a \in G: a \cdot 1 = 1 \cdot a = a$
3. $\forall a \in G, \exists a^{-1} \in G: a \cdot a^{-1} = a^{-1} \cdot a = 1$

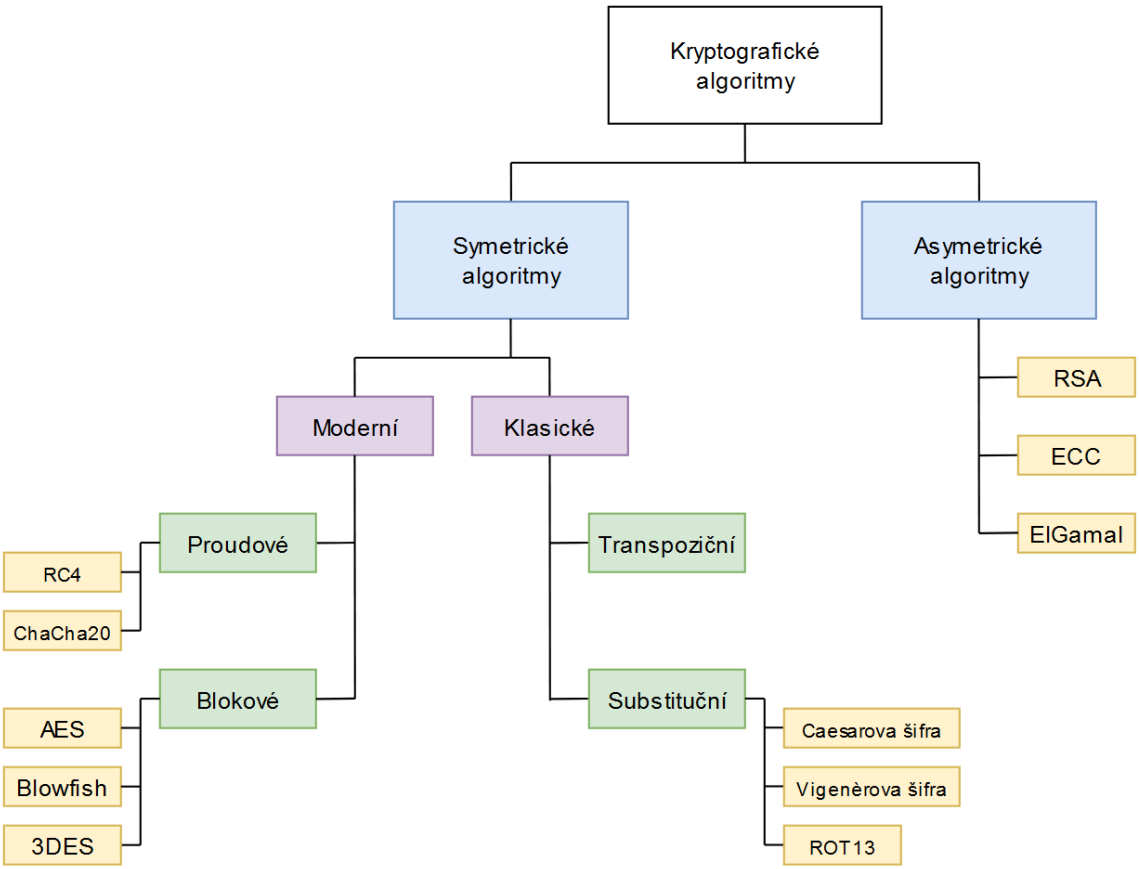
Poznámka: Pokud platí, že $\forall a, b \in G: a + b = b + a$, případně $\forall a, b \in G: a \cdot b = b \cdot a$, pak budeme strukturu nazývat Abelova (komutativní) grupa. [38]

# 2 Kryptografické algoritmy a jejich matematický popis

Tato kapitola se zabývá popisem významných kryptografických algoritmů, jejichž výběr byl určen na základě historické významnosti a relevantnosti v moderní kryptografii dle [8], [13], [22].

## 2.1 Klasifikace kryptografických algoritmů

Způsobů, jak klasifikovat kryptografické algoritmy je nespočet. Tato práce se bude držet obecného rozdělení na symetrické a asymetrické algoritmy, tedy podle způsobu rozdělení klíčů. [3]




Obrázek 4: Klasifikace kryptografických algoritmů

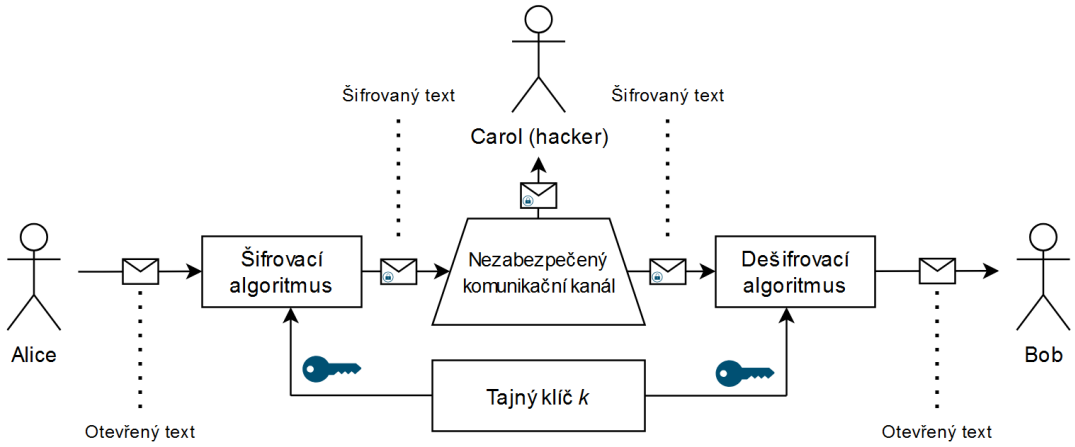
*Zdroj: vlastní zpracování podle [3] a [14]*

## 2.2 Symetrické kryptografické algoritmy

Až do 70. let 20. století byly všechny kryptosystémy založeny na symetrických algoritmech, u nichž lze jejich princip fungování ilustrovat na příkladu:

Existují dva uživatelé, Alice a Bob, kteří si chtějí navzájem poslat zprávu skrze nezabezpečený komunikační kanál. Nastal ale problém, kdy se do nezabezpečeného komunikačního kanálu naboural neautorizovaný uživatel Carol, který chce číst zprávy, které si mezi sebou posílají Alice a Bob. Otázka zní: „Jak si mohou Alice a Bob bezpečně posílat zprávy, aniž by si komunikaci mohl číst Carol?“ Symetrická kryptografie je zde možným řešením. Alice zašifruje svou zprávu dohodnutým tajným klíčem (vytvořený například náhodným generováním), kterou následně zašle přes komunikační kanál Bobovi, který zprávu pomocí stejného klíče dešifruje. Pokud je algoritmus dostatečně silný, tak bude pro narušitele komunikačního kanálu zpráva vypadat pouze jako spleť náhodných bitů. [3]

Hlavní charakteristikou symetrických kryptosystémů je tedy využití stejného klíče pro šifrování a dešifrování zpráv, což má za výhodu nižší výpočetní nároky na systém, avšak za cenu existence problému s distribucí klíčů. [4]




Obrázek 5: Symetrická kryptografie

*Zdroj: vlastní zpracování podle [3]*

Samotnou symetrickou kryptografii lze rozdělit do dvou etap. První z nich budeme nazývat „klasická symetrická kryptografie“, jejíž hlavním znakem je, že k zašifrování stačily jednoduché pomůcky (papír a tužka, primitivní mechanické stroje, …). [8]

Do tohoto období lze zařadit:

- **Substituční šifry** – kde principem fungování je přiřazení každému znaku $x$ otevřeného textu jiný znak $y$, tedy využití operace substituce. Substituční šifry rozlišujeme podle způsobu přiřazování znaků na: [1]

1. **Monoalfabetické** – u kterých je každému znaku otevřeného textu přiřazen konkrétní znak právě jedné abecedy. Například „A“ vždy zašifrujeme jako „D“.
2. **Polyalfabetické** – které umožňují při šifrování využívat více abeced. To znamená, že každý znak otevřeného textu může být zašifrován jinou abecedou.
3. **Homofonní** – u kterých je každému znaku otevřeného textu přiřazeno více možných znaků. Například „A“ můžeme zašifrovat jako „01“ nebo „21“.
4. **Polygramové** – u kterých se šifrují skupiny znaků. Například „AD“ můžeme zašifrovat jako „CX“. [8]

- **Transpoziční šifry** – kde se mění uspořádání znaků otevřeného textu. [8]

Druhou etapu budeme nazývat „moderní symetrická kryptografie“, která se odlišovala její výraznější výpočetní složitostí. V moderní symetrické kryptografii rozlišujeme 2 typy šifer:

- **Blokové šifry** – které šifrují celý blok otevřeného textu o délce $2^n$ bitů. To znamená, že zašifrování jakéhokoliv bitu otevřeného textu závisí na každém dalším bitu otevřeného textu ve stejném bloku. Většina blokových šifer má nastavenou velikost bloku 64 bitů (3DES) nebo 128 bitů (AES). [3], [4]
- **Proudové šifry** – kde se šifruje každý bit otevřeného textu samostatně. Většinou s využitím operace XOR. [15]

### 2.2.1 Caesarova šifra

Caesarova šifra je elementární monoalfabetická substituční šifra. Původně se jednalo o posun znaků o tři místa v abecedě, kde se „A“ se zašifrovalo jako „D“, „B“ se zašifrovalo jako „E“, „Z“ se zašifrovalo jako „C“. Novodobé pojetí je však zobecněné a říká, že se jedná o posun o $n$ znaků. Číslo $n$ je zde tajným klíčem. [4]

Předpokládejme anglickou abecedu, zašifrování znaku $x$ o $n$ míst pomocí Caesarovy šifry se dá vyjádřit vzorcem:

$$E_n(x) = (x + n) \bmod 26$$

Jelikož se jedná o symetrické šifrování, tak se znalostí čísla $n$ lze zprávu dešifrovat opačným procesem k šifrování:

$$D_n(x) = (x - n) \bmod 26$$

Caesarova šifra se stala vzorem pro vznik dalších substitučních šifer jako je Vigenèrova šifra nebo ROT13.

Dnes již nepovažujeme Caesarovu šifru za bezpečnou z důvodu jejího snadného pokoření útokem hrubou silou nebo za použití frekvenční analýzy [8]. Pro ilustraci si zašifrujeme zprávu „UNIVERZITA PARDUBICE“ pomocí Caesarovy šifry, kde klíčem $n$ bude posun o 10 znaků.

Tabulka 3: Klíč k Caesarově šifře

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

Tabulka 4: Šifrování pomocí Caesarovy šifry

| $x$ | U | N | I | V | E | R | Z | I | T | A | | P | A | R | D | U | B | I | C | E |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $y$ | E | X | S | F | O | B | J | S | D | K | | Z | K | B | N | E | L | S | M | O |

Otevřený text „UNIVERZITA PARDUBICE“ se nám zašifroval na „EXSFOBJSDK ZKBNELSMO“. Nyní v roli narušitele prolomíme šifru útokem hrubou silou. Jelikož jsou nám známy informace, že text obsahuje 20 znaků včetně mezery a mohutnost zprávy je 26 znaků, tak dokážeme odvodit, že celkový počet klíčů je 25.

Tabulka 5: Prolomení Caesarovy šifry

| Dešifrovaný text | Klíč |
|---|---|
| DWRENAIRCJ YJAMDKRLN | 1 |
| CVQDMZHQBI XIZLCJQKM | 2 |
| BUPCLYGPAH WHYKBIPJL | 3 |
| ATOBKXFOZG VGXJAHOIK | 4 |
| ZSNAVJENYF UFWIZGHNI | 5 |
| YRMZIVDMXE TEVHYFMGI | 6 |
| XQLYHUCLWD SDUGXELFH | 7 |
| WPKYGTKBVC RCTFWDKEG | 8 |
| VOJWFSAJUB QBSEVCJDF | 9 |
| UNIVERZITA PARDUBICE | 10 |
| TMHUDQYHSZ OZQCTAHBD | 11 |
| SLGTCPXGRY NYPBSZGAC | 12 |
| RKFSBOWFQX MXOARYFZB | 13 |

| Dešifrovaný text | Klíč |
|---|---|
| QJERANVEPW LWNZQXEYA | 14 |
| PIDQZMUDOV KVMVPWDXZ | 15 |
| OHCPYLTGNU JULXOVCWY | 16 |
| NGBXOKSBMT ITKWNUBVX | 17 |
| MFANWJURALS HSUVMTAUW | 18 |
| LEZMVJQZKR GRIULSZTV | 19 |
| KDYLUHPYJQ FQHTKRYSU | 20 |
| JCXKTGOXIP EPGSJQXRR | 21 |
| LBWJSFNVHO DOFRIPWQS | 22 |
| HAVIREMNWG CNEQHOVTQ | 23 |
| GQUDHQLUFM BMDPGNUNO | 24 |
| FYTGPCKTEL ALCOFMTNP | 25 |

Po útoku hrubou silou lze odvodit, že klíč $n = 10$. Je tedy zcela zřejmé, že dnes již Caesarova šifra nemá v praxi bezpečnostní využití. Lze ji však představovat jako ucelený základ, ze kterého se postupem času vyvíjely důmyslnější kryptografické algoritmy. Implementace Caesarovy šifry v programovacím jazyce Python je uvedena v Příloze 2.

### 2.2.2 Vigenèrova šifra

Vigenèrova šifra je zásadní šifrou v období klasické symetrické kryptografie, která přímo vychází z Caesarovy šifry. Jedná se o polyalfabetickou substituční šifru, kde šifrujeme s využitím Vigenèrova čtverce (známého též jako tabula recta), který nám umožňuje použít až 26 abeced pro šifrování. [8], [14].

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

Obrázek 6: Vigenèrův čtverec

*Zdroj: [39]*

Při šifrování se pro každé písmeno abecedy otevřeného textu vytvoří další šifrovací abeceda jejíž znaky jsou posunuté o $n$ pozic. [14], [39]

Mějme otevřený text „KRYPTOGRAFIE“, pro který nastavíme klíč „UPCE“. Jelikož počet znaků klíče neodpovídá počtu znaků otevřeného textu, tak budeme znaky klíče opakovat do té doby, dokud nebudou rovny počtu znaků otevřeného textu. Algoritmus pro zašifrování bude vypadat následovně: Vezmeme první znak otevřeného textu, který nalezneme v řádku $i$ Vigenèrova čtverce, následně vezmeme příslušný znak klíče a ten nalezneme v sloupci $j$ Vigenèrova čtverce. V místě průniku $i$ a $j$ se nachází náš substituční znak:

Tabulka 6: Šifrování pomocí Vigenèrovy šifry

| $x$ | K | R | Y | P | T | O | G | R | A | F | I | E |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| klíč $n$ | U | P | C | E | U | P | C | E | U | P | C | E |
| $y$ | E | G | A | T | N | D | I | V | U | U | K | I |

Přestože náš otevřený text obsahoval dva stejné znaky („R“), tak se nám díky využití více abeced zašifrovaly zcela unikátně.

Algebraicky se dá vzorec pro zašifrování otevřeného textu pomocí Vigenèrovy šifry formulovat: [14]

$$E_i = (P_i + K_i) \bmod 26$$

Kde $P_i = i$-tý znak otevřeného textu, $K_i = i$-tý znak klíče. A jehož přeformulováním získáme vzorec pro dešifrování: [14]

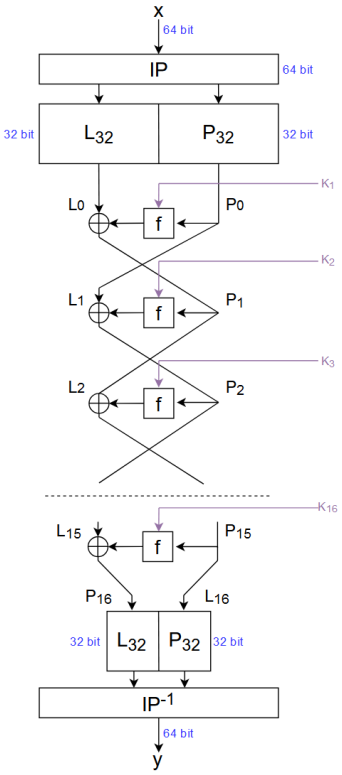
$$P_i = (E_i - K_i) \bmod 26$$

Vigenèrova šifra má však značné bezpečnostní vady spojené s tvorbou klíče, který je délkou omezen samotnou délkou zprávy a skutečností, že klíč je tvořen opakujícími se polygramy. Pokud bychom chtěli Vigenèrovu šifru rozluštit, bylo by zapotřebí odhadnout nebo vyvrátit délku klíče, čehož lze docílit nalezením frekventovaně opakujících se polygramů v zašifrovaném textu, kde vzdálenost mezi jednotlivými bigramy a trigramy představují násobky délky klíče. Tento jev se dá testovat použitím Kasiského metody. [12]

### 2.2.3 Data Encryption Standard

Data Encryption Standard je blokový kryptografický algoritmus fungující na principu Feistelovy sítě vyvinutý v 70. letech 20. století společností IBM. Algoritmus DES šifruje data

po blocích s délkou 64 bitů, kde je každý osmý bit vyhrazen ke kontrole parity, což nastavuje čistou délku klíče na 56 bitů. DES byl vyvinut explicitně pro hardwarovou realizaci, proto využívá výhradně operaci XOR. [16], [17]




Obrázek 7: Algoritmus DES

*Zdroj: vlastní zpracování podle [3], [4]*

V případě šifrování na Obrázku 7 algoritmus DES pracuje se 64bitovým vstupním blokem X, se kterým se provádí 16 kol šiforvání, kde pro každé kolo šifrování (dale jen runda) je nastaven jiný klíč $K_n$. Výstupem je 64bitový zašifrovaný blok Y.

**Princip fungování:**

1. Algoritmus provede počáteční permutaci IP, ta rozdělí 64bitový blok X na levou část (L) a pravou část (P) po 32 bitech.
2. V každé rundě se nejprve použije na pravou část a funkci $f$ (vzniklou expanzí klíče a substitucí) unikátní klíč $K_n$.

3. Funkce $f$ je spojena s levou částí pomocí logického operátoru XOR. Následně se levá část prohodí s pravou a tím se ukončuje 1. runda.

$$L_{i+1} = P_i$$

$$P_{i+1} = L_i \oplus f(P_i, k_{i+1})$$

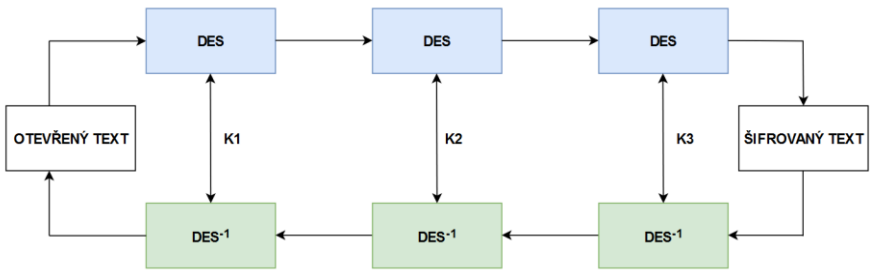
4. Tento postup proběhne celkem 16krát. Na konci 16. rundy se levá část s pravou neprohazují, ale opět se sloučí pomocí inverzní permutace $IP^{-1}$, čímž vznikne výstupní zašifrovaný blok dat Y. [3]

Dešifrování DES probíhá inverzním procesem se zachováním struktury rund, kdy se začíná rundou 16.

Dnes již není algoritmus DES považován za bezpečný, vzhledem k jeho krátké délce klíče a zranitelnosti vůči moderním útokům na šifrovací algoritmy. Kdybychom dnes i tak chtěli algoritmus fungující na tomto principu využít v praxi, máme možnost využít algoritmus Triple Data Encryption Standard (3DES), který je však od roku 2024 též považován za nevhodný pro šifrování nových dat dle NIST. [40]

Algoritmus 3DES vychází z klasického DES s tím rozdílem, že se na každý blok dat aplikuje algoritmus DES celkem třikrát (ať už stejnými, nebo rozdílnými klíči). [40]

$$E_{3DES} = DES_{k3}(DES_{k2}(DES_{k1}(X)))$$




Obrázek 8: Kryptosystém 3DES

*Zdroj: vlastní zpracování podle [3], [18]*

### 2.2.4 Advanced Encryption Standard

Advanced Encryption Standard je blokový šifrovací algoritmus, který je považován NÚKIB a NIST za šifrovací bezpečnostní standard [22], [35]. Algoritmus AES disponuje pevnou velikostí bloku 128 bitů s možnými délkami klíčů 128, 192 nebo 256 bitů.

Šifrování probíhá na více vrstvách, kdy je nejprve vykonána expanze klíče, kde se z klíče $K$ vytvoří 11 odvozených klíčů $K_n$. Poté se z otevřeného textu se odejme 128 bitový blok, který je následně převeden na 16 bytový řetězec ve formě stavové matice 4x4. [1]

$$S = \begin{bmatrix} 63 & 7C & 77 & 7B \\ CA & 82 & C9 & 7D \\ B7 & FD & 93 & 26 \\ 04 & C7 & 23 & C3 \end{bmatrix}$$

V algoritmu AES jsou pro stavovou matici definovány operace substituce bytů, rotace řádků, substituce sloupců a přidání iteračního klíče.

1. **Substituce bytů** se provádí pomocí substituční tabulky zvané S-box.

| x \ y | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | a | b | c | d | e | f |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 63 | 7c | 77 | 7b | f2 | 6b | 6f | c5 | 30 | 01 | 67 | 2b | fe | d7 | ab | 76 |
| 1 | ca | 82 | c9 | 7d | fa | 59 | 47 | f0 | ad | d4 | a2 | af | 9c | a4 | 72 | c0 |
| 2 | b7 | fd | 93 | 26 | 36 | 3f | f7 | cc | 34 | a5 | e5 | f1 | 71 | d8 | 31 | 15 |
| 3 | 04 | c7 | 23 | c3 | 18 | 96 | 05 | 9a | 07 | 12 | 80 | e2 | eb | 27 | b2 | 75 |
| 4 | 09 | 83 | 2c | 1a | 1b | 6e | 5a | a0 | 52 | 3b | d6 | b3 | 29 | e3 | 2f | 84 |
| 5 | 53 | d1 | 00 | ed | 20 | fc | b1 | 5b | 6a | cb | be | 39 | 4a | 4c | 58 | cf |
| 6 | d0 | ef | aa | fb | 43 | 4d | 33 | 85 | 45 | f9 | 02 | 7f | 50 | 3c | 9f | a8 |
| 7 | 51 | a3 | 40 | 8f | 92 | 9d | 38 | f5 | bc | b6 | da | 21 | 10 | ff | f3 | d2 |
| 8 | cd | 0c | 13 | ec | 5f | 97 | 44 | 17 | c4 | a7 | 7e | 3d | 64 | 5d | 19 | 73 |
| 9 | 60 | 81 | 4f | dc | 22 | 2a | 90 | 88 | 46 | ee | b8 | 14 | de | 5e | 0b | db |
| a | e0 | 32 | 3a | 0a | 49 | 06 | 24 | 5c | c2 | d3 | ac | 62 | 91 | 95 | e4 | 79 |
| b | e7 | c8 | 37 | 6d | 8d | d5 | 4e | a9 | 6c | 56 | f4 | ea | 65 | 7a | ae | 08 |
| c | ba | 78 | 25 | 2e | 1c | a6 | b4 | c6 | e8 | dd | 74 | 1f | 4b | bd | 8b | 8a |
| d | 70 | 3e | b5 | 66 | 48 | 03 | f6 | 0e | 61 | 35 | 57 | b9 | 86 | c1 | 1d | 9e |
| e | e1 | f8 | 98 | 11 | 69 | d9 | 8e | 94 | 9b | 1e | 87 | e9 | ce | 55 | 28 | df |
| f | 8c | a1 | 89 | 0d | bf | e6 | 42 | 68 | 41 | 99 | 2d | 0f | b0 | 54 | bb | 16 |

Obrázek 9: S-box v hexadecimální číselné soustavě

*Zdroj: [41]*

Při této operaci se ke každé hodnotně vstupního bytu přiřadí jedinečná hodnota výstupního bytu, kde první čtyři bity vstupního bytu se rovnají proměnné $x$, která vytváří řádek tabulky a poslední čtyři bity vstupního bytu se rovnají proměnné $y$, která vytváří sloupec tabulky. Souřadnice $x, y$ udávají hodnotu substituce.

$$A = \begin{bmatrix} 1A & 1B & 1C & 1D \\ 2A & 2B & 2C & 2D \\ 3A & 3B & 3C & 3D \\ 4A & 4B & 4C & 4D \end{bmatrix} \rightarrow Substituce \rightarrow A' = \begin{bmatrix} 1A' & 1B' & 1C' & 1D' \\ 2B' & 2C' & 2D' & 2A' \\ 3C' & 3D' & 3A' & 3B' \\ 4D' & 4A' & 4B' & 4C' \end{bmatrix}$$

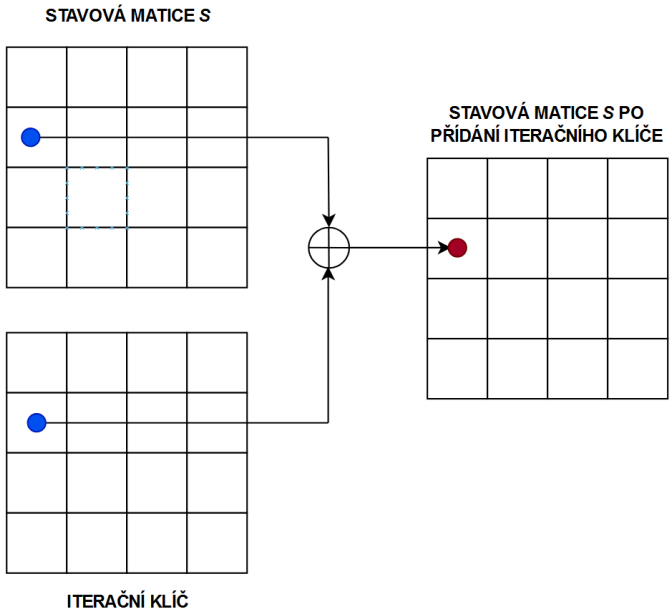
2. **Rotace řádků** posune první řádek o nula pozic, druhý řádek o jednu pozici vlevo, třetí řádek o dvě pozice vlevo a poslední řádek o tři pozice vlevo.

$$\begin{bmatrix} 1A & 1B & 1C & 1D \\ 2A & 2B & 2C & 2D \\ 3A & 3B & 3C & 3D \\ 4A & 4B & 4C & 4D \end{bmatrix} \rightarrow Rotace \rightarrow \begin{bmatrix} 1A & 1B & 1C & 1D \\ 2B & 2C & 2D & 2A \\ 3C & 3D & 3A & 3B \\ 4D & 4A & 4B & 4C \end{bmatrix}$$

3. **Substituce sloupců** spočívá ve vynásobení stavové matice s předdefinovanou mixovací maticí.

$$\begin{bmatrix} 2 & 3 & 1 & 1 \\ 1 & 2 & 3 & 1 \\ 1 & 1 & 2 & 3 \\ 3 & 1 & 1 & 2 \end{bmatrix} \cdot \begin{bmatrix} A \\ B \\ C \\ D \end{bmatrix} = \begin{bmatrix} 2 \cdot A + 3 \cdot B + 1 \cdot C + 1 \cdot D \\ 1 \cdot A + 2 \cdot B + 3 \cdot C + 1 \cdot D \\ 1 \cdot A + 1 \cdot B + 2 \cdot C + 3 \cdot D \\ 3 \cdot A + 1 \cdot B + 1 \cdot C + 2 \cdot D \end{bmatrix}$$

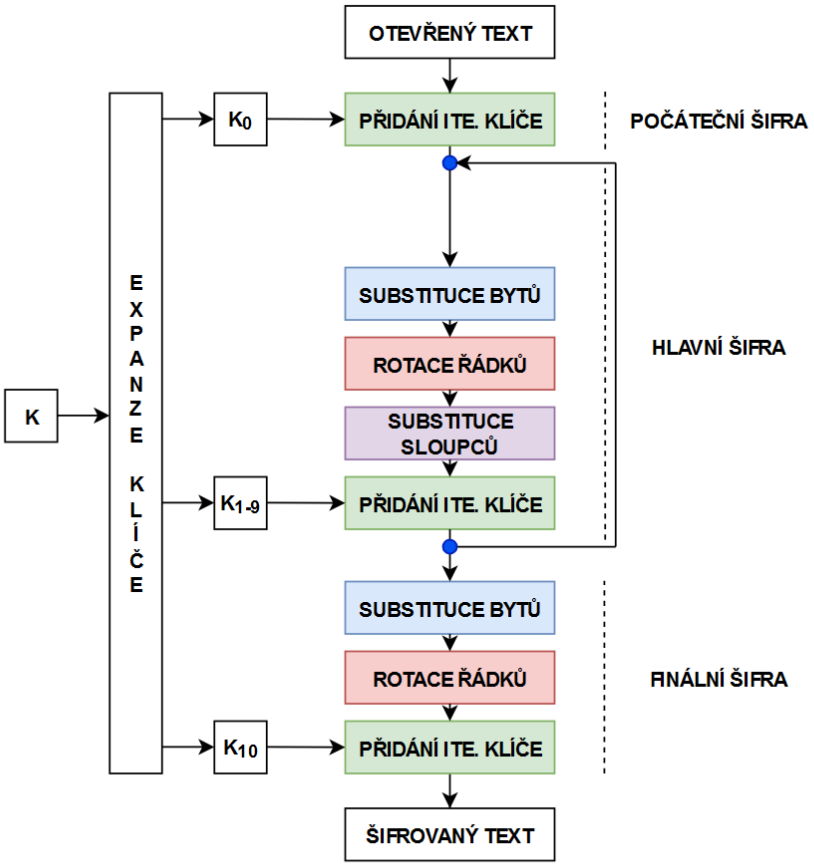
4. **Přidání iteračního klíče** je poslední z hlavních operací v algoritmu AES. Jedná se o přičtení klíče (který je ve stejném formátu jako stavová matice) ke stavové matici pomocí operace XOR.




Obrázek 10: Přičtení iteračního klíče

*Zdroj: vlastní zpracování podle [1]*

Jednotlivá posloupnost kroků je schematicky znázorněna v Obrázku 11.




Obrázek 11: Algoritmus AES

*Zdroj: vlastní zpracování podle [1]*

### 2.2.5 Rivest Cipher 4

Rivest Cipher 4 (RC4) je symetrická proudová šifra disponující proměnlivou délkou klíče, fungující na principu vygenerovaného proudu náhodně vypadávajících bytů, které jsou následně spojeny s otevřeným textem pomocí operace XOR, čímž vzniká šifrovaný text. RC4 používá k šifrování 256 bytový S-box, kde záznamy jsou permutace čísel 0 až 255, které jsou funkcí proměnné délky klíče. [43] Ačkoliv je algoritmus RC4 10x rychlejší, než algoritmus DES, tak je vzhledem k jeho potenciální napadnutelnosti moderní kryptoanalýzou považován za nedoporučovaný bezpečnostními institucemi NIST a NÚKIB. [22], [42]

### 2.2.6 Blowfish

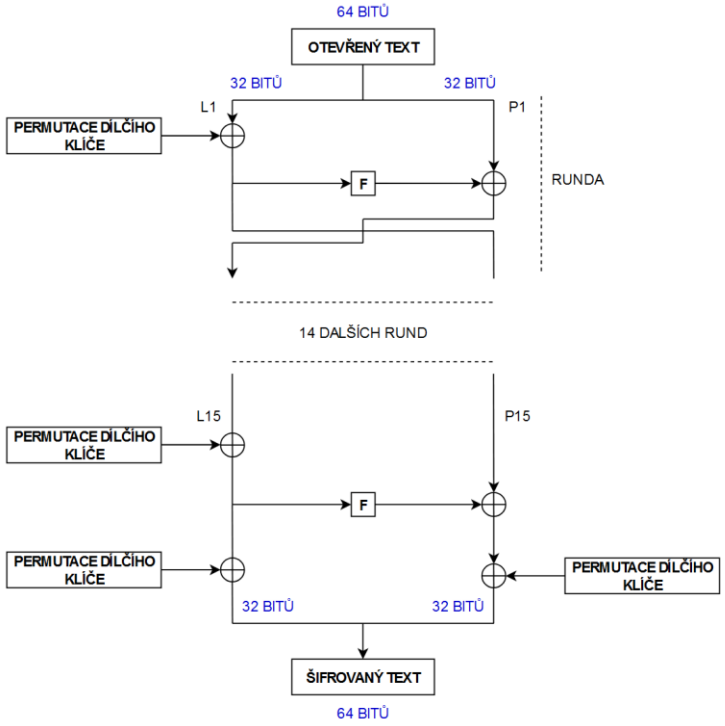
Blowfish je symetrická bloková šifra, původně zamýšlená jako alternativa k AES, která využívá 64-bitové bloky s proměnlivou délkou klíče 32-448 bitů. Podobně jako algoritmus DES, využívá k šifrování Feistelovu síť.

Šifrovací klíč se zde hraje roli k nastavení počátečních hodnot pro dílčí klíče, pomocí kterých se v 16 rundách šifruje.

Dílčí klíče tedy vychází z hlavního klíče, a to pomocí řady operací, které mixují bity hlavního klíče. Princip šifrování zde spočívá v rozdělení otevřeného textu (64 bitů) na levou a pravou část bloku (32 bitů) a poté se provede 16 rund základních operací: [44]

1. XOR dílčího klíče
2. XOR levé části bloku
3. XOR pravá části bloku

Šifrování otevřeného textu pomocí algoritmu Blowfish je ilustrován na Obrázku 12:




Obrázek 12: Algoritmus Blowfish

*Zdroj: vlastní zpracování podle [44]*

## 2.3 ChaCha20

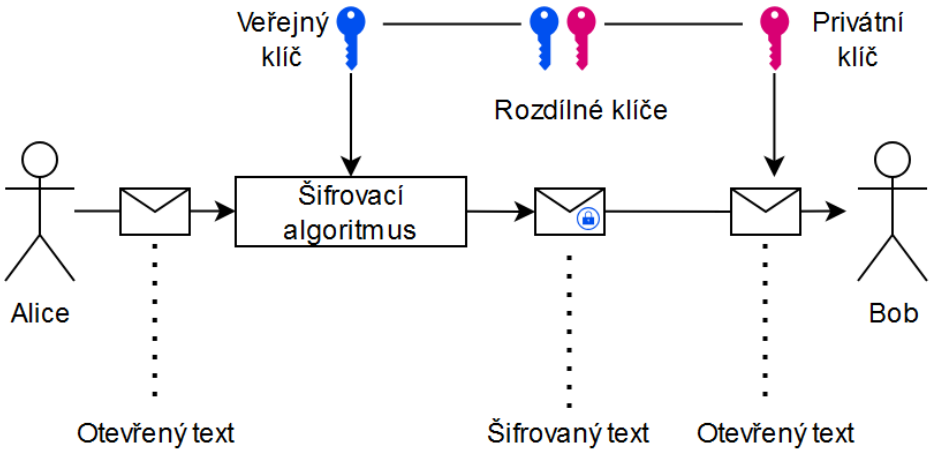
ChaCha20 je proudový šifrovací algoritmus, který využívá 256-bitový klíč. Jedná se o speciální případ proudové šifry zpočátku fungující jako bloková šifra, která vyvolá blokovou funkci, která využívá stejný klíč a k němu náhodně vygenerovanou hodnotu s postupně se zvyšujícím se počítadlem bloku, čímž se vytvářejí bloky šifrovacího proudu, které jsou poté sloučeny

dohromady do jedno mohutného proudu. Nakonec provede operace XOR mezi šifrovaný proudem a otevřeným textem. Šifra ChaCha20 je díky své funkcionalitě jedním z nejefektivnějších proudových šifrovacích algoritmů, který však může být omezen délkou klíče v závislosti na konkrétním protokolu, jehož je součástí. [45]

## 2.4 Asymetrické kryptografické algoritmy

Asymetrická kryptografie neboli kryptografie s veřejným klíčem vznikla v 70. letech 20. století a setkáme s ní hlavně u digitálních podpisů a bezpečnostních protokolů. Dalo by se konstatovat, že se jednalo o jedinou obří revoluci v kryptografii, jelikož až do jejího vzniku veškeré šifrování záviselo na elementárních operacích jako jsou substituce, rotace a permutace. [1] Další takovou revoluci lze očekávat v budoucnu s nástupem kvantové kryptografie. [23]

Hlavním rozdílem mezi symetrickou a asymetrickou kryptografií je, že v asymetrické kryptografii šifrujeme a dešifrujeme pomocí separátních klíčů. Tyto klíče dělíme na veřejný a privátní (soukromý), kde každý uživatel, který je součástí kryptosystému má svůj vlastní pár. Je to právě díky této funkcionalitě, která nám řeší problém s distribucí klíčů, avšak za cenu vyšší výpočetní náročnosti. [8]




Obrázek 13: Asymetrický kryptosystém

*Zdroj: vlastní zpracování podle [8]*

Princip fungování asymetrické kryptografie lze ilustrovat na následujícím příkladu, kdy si Alice a Bob chtějí bezpečně poslat zprávu bez nutnosti řešení problému distribuce klíčů:

1. Nejprve si Alice s Bobem každý vygenerují svůj vlastní veřejný a privátní klíč.
2. Alice zašifruje zprávu veřejným klíčem Boba a zašle zprávu přes komunikační kanál.
3. Bob po přijetí zprávy využije svůj privátní klíč, aby zprávu dešifroval. [4]

Nicméně, i zde se vyskytuje významný problém, a to s ověřením původu klíčů.

Aby se problému o původu klíčů předešlo a Bob měl jistotu, že komunikuje s Alicí, je zapotřebí, aby Alice svou zprávu opatřila digitálním podpisem a tím zajistila její autenticitu. [3]

## 2.5 Digitální podpis

Digitální podpis je elektronická obdoba ověřeného vlastnoručního podpisu. Jedná se o způsob zajištění autenticity, integrity a nepopiratelnosti zprávy.

**Princip fungování:**

1. Alice ze zprávy vytvoří hash (například pomocí hashovací funkce SHA-256).
2. Alice zašifruje hash svým privátním klíčem, čímž prokáže svou jedinečnou identitu a vytvoří tak digitální podpis.
3. Alice digitální podpis připojí k původní zprávě a pošle ji přes komunikační kanál Bobovi.
4. Bob nyní musí ověřit digitální podpis. Pomocí veřejného klíče Alice dešifruje digitální podpis a získá hash.
5. Bob vytvoří hash ze zprávy od Alice (pomocí stejné hashovací funkce).
6. Bob porovná svůj hash s hashem Alice. Pokud se shodují, pak je digitální podpis platný a ověřen. [4]

Existuje mnoho způsobů, jak zprávu zašifrovat pomocí asymetrické kryptografie. Jedním z hojně používaných metod je šifrování prostřednictvím algoritmu RSA.

## 2.6 RSA

RSA (Rivest, Shamir, Adleman) je asymetrický šifrovací algoritmus, pomocí kterého lze vytvořit digitální podpis.

Při šifrování algoritmem RSA se předpokládá, že faktorizace velkých čísel na součin prvočísel je obtížná, výpočetně a časově velmi náročná úloha. Princip je takový, že pokud máme součin

prvočísel $n = p \cdot q$, pak nelze v čase životnosti informace zjistit hodnoty $p$ a $q$, jestliže je nám známo číslo $n$ (neboli nedokážeme zjistit jeho činitele). [2], [3]

**Matematický princip algoritmu RSA:**

1. Zvolíme si dvě velká prvočísla $p$ a $q$.
2. Proveďme operaci $n = (p \cdot q)$.
3. Vypočítejme hodnotu Eulerovy funkce: $\varphi(n)$ .
4. Zvolíme exponent $e$, takový aby $1 < e < \varphi(n)$, který je s $\varphi(n)$ nesoudělný.
5. Nalezněme číslo $d$ takové, pro které platí, že: $d \cdot e \equiv 1 \bmod \varphi(n)$.
6. Nyní známe dvojice klíčů $VK = (e, n)$ a $PK = (d, n)$.
7. Zprávu $Z$ nyní převedeme na $\mathbb{Z} \in [0, n-1]$.
8. Zprávu zašifrujeme: $C = Z^e \bmod n$.
9. Zprávu dešifrujeme: $Z = C^e \bmod n$. [3]

Zdali zvolené $p$ a $q$ jsou skutečně prvočísla nebo čísla složená lze ověřit již zmíněným Miller-Rabinovým testem prvočíselnosti. Implementace algoritmu RSA ve vývojovém prostředí je uvedena v Příloze 4.

## 2.7 Kryptografie nad eliptickými křivkami

Kryptografie nad eliptickými křivkami (ECC) je asymetrický šifrovací algoritmus, který využívá při šifrování algebraické vlastnosti eliptických křivek, díky kterým lze vytvářet kryptosystémy o síle RSA s daleko menšími délkami klíčů, čímž snižuje výpočetní náročnost. [33]

### 2.7.1 Eliptická křivka

Eliptická křivka je algebraická křivka, kterou lze vyjádřit kubickou rovnicí $y^2 = x^3 + ax + b$, kde $a, b, x, y \in \mathbb{R}$. [27]

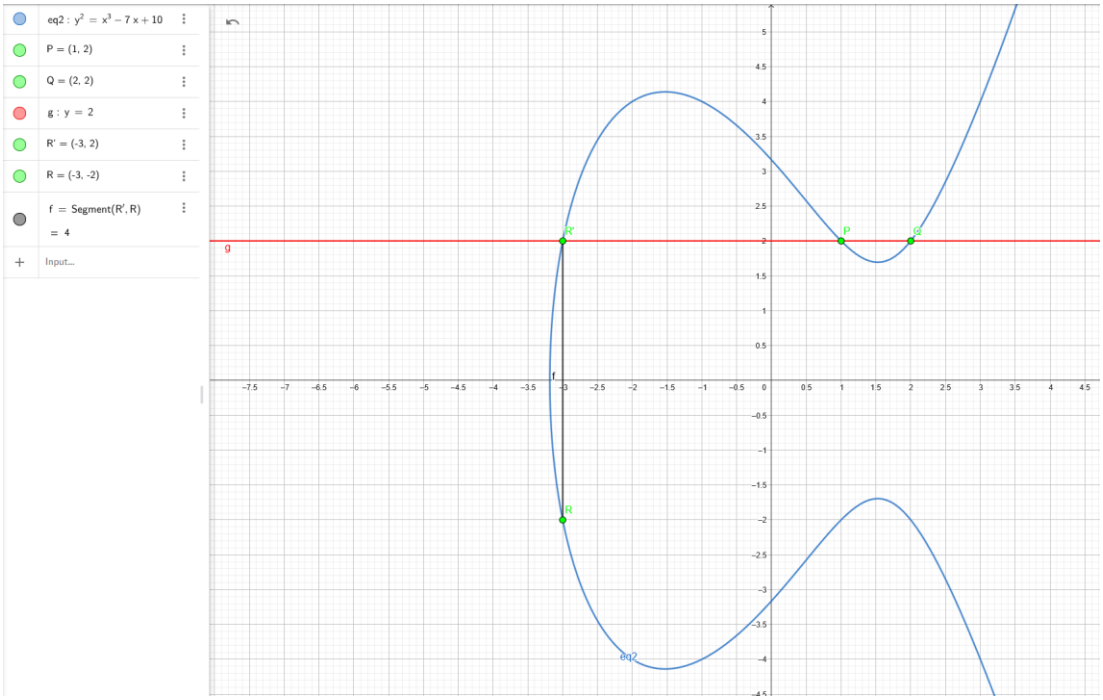
Jednou z nejzásadnějších vlastností eliptických křivek je, že pokud na křivce zavedeme množinu bodů, tak geometrickým součtem těchto bodů dostaneme nový bod ležící na stejné křivce. Z toho vyplývá, že eliptické křivky nesou vlastnosti aditivní algebraické grupy.

Př. Nechť na křivce $y^2 = x^3 - 7x + 10$ leží body $P(1,2)$ a $Q(2,2)$. Tyto body nejprve proložíme přímkou $y = 2$. Následně nalezneme průsečíky přímky s křivkou.

$$2^2 = x^3 - 7x + 10$$

$$4 = x^3 - 7x + 10$$
$$x^3 - 7x + 6 = 0$$
$$x^3 - 7x + 6 = (x - 1)(x - 2)(x + 3)$$

Z výpočtu lze odvodit, že průsečíkem $P(1,2)$ a $Q(2,2)$ je bod $R'(-3,2)$ jehož zrcadlovým bodem přes osu $x$ je $R(-3,-2)$, tedy geometrickým součtem $P(1,2) + Q(2,2) = R(-3,-2)$. [27]




Obrázek 14: Eliptická křivka

*Zdroj: vlastní zpracování*

### 2.7.2 Šifrování pomocí eliptických křivek

Premisou bezpečnosti ECC je složitost řešení speciálního případu problému diskrétního logaritmu:

Nechť $E$ je eliptická křivka definována nad konečným polem $\mathbb{F}_q$, bod $P \in E(\mathbb{F}_q)$ n-tého řádu. a bod $Q \in [P]$. Úkolem je nalézt číslo $l$, kde $0 \leq l \leq n-1$, pro které platí $Q = l \cdot P$. [28]

ECC je základem mnoha šifrovacích protokolů, jako jsou Elliptic-curve Diffie-Hellman pro bezpečnou výměnu klíče nebo Elliptic Curve Digital Signature Algorithm nebo Elliptic Curve Schnorr Signature Algorithm pro digitální podpis. [33]

## 2.8 Shrnutí symetrické a asymetrické kryptografie

Symetrická kryptografie je systém se 2 hlavními vlastnostmi:

1. Stejný klíč je používán pro proces šifrování a dešifrování
2. Funkce šifrování a dešifrování jsou si identické nebo velmi podobné

Moderní symetrické šifrovací algoritmy jsou bezpečné a velmi rychlé. Avšak disponují několika problémy, které je nutné si při implementaci uvědomit: [3], [8]

- **Problém distribuce klíče** – není možné užitím samotné symetrické kryptografie zajistit bezpečné předání klíče mezi uživateli, jelikož jim není dostupný bezpečný komunikační kanál [3]
- **Problém velkého počtu klíčů** – v případech, kdy je v systému velký počet uživatelů se vyskytuje problém s uchování velkého počtu klíčů. Příkladem může být firma, která zaměstnává 1000 pracovníků, kde každý pár pracovníků musí mít svůj vlastní klíč. To lze vypočítat vzorcem: [3]

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

  Kde $n$ je počet pracovníků ve firmě. Dosazením zjistíme, že by bylo potřeba bezpečně uchovávat 499 500 klíčů, což by bylo velmi náročné.

- **Žádné zabezpečení proti podvodům (Non-repudation problem)** – v symetrické kryptografii neexistuje způsob, který by dokázal ověřit, zdali v komunikaci někdo nelže o zaslání nebo nezaslání zprávy. [3]

Asymetrická kryptografie je systém, který pro šifrování a dešifrování využívá jiné klíče, což dělá asymetrické šifrovací algoritmy výpočetně náročnější, zdrojově méně efektivní a pomalejší než symetrické. Praktické zabezpečení asymetrických šifrovacích algoritmů spočívá (podle rodiny algoritmů) neřešitelností matematických problémů (faktorizace velkých čísel, diskrétní logaritmus) v rozumném čase. [3]

Samotnou symetrickou a asymetrickou kryptografii lze kombinovat tvorbou hybridních kryptosystémů v rámci bezpečnostních protokolů. [8]

# 3 Porovnání kryptografických algoritmů

Určit, který algoritmus je pro řešení bezpečnostních problémů optimální nelze, proto při porovnání je zásadní informací popis problému v určitém kontextu.

Jak již bylo zmíněno, hlavní účel kryptografických algoritmů, jakožto logických bezpečnostních mechanismů je ochránit informaci před narušitelem v kyberprostoru. Každý typ kryptografického algoritmu má odlišnou míru odolnosti vůči různým způsobům útoků.

## 3.1 Způsoby prolomení kryptosystémů

Existuje mnoho způsobů útoků na šifrovací algoritmy. Tyto způsoby se převážně liší jejich výpočetní náročností a efektivitou. [3]

### 3.1.1 Útok hrubou silou

Útok hrubou silou (Brute-Force Attack, BFA) je jedním z nejrozšířenějších způsobů prolomování šifer. Princip tohoto útoku spočívá v systematickém zkoušení kombinací znaků zvolené abecedy, dokud není nalezen klíč, většinou s využitím SW nástroje. [8]

### 3.1.2 Frekvenční analýza

Frekvenční analýza (Frequency analysis) je metoda prolomování šifer, jejíž premisa spočívá ve kvantitativním měření výskytu znaků vybrané abecedy. Pokud získáme informaci o relativní četnost znaků vybrané abecedy, lze ji porovnat s relativní četností znaků šifrovaného textu. Frekvenční analýza též může sloužit jako nástroj k odhalení, zdali je šifrovaný text zašifrovaný substitučním nebo transpozičním šifrovacím algoritmem. [11]

### 3.1.3 Útok postranním kanálem

Útok postranním kanálem (Side Channel Attack) je metoda prolomování šifer s využitím informací, které neúmyslně unikly během šifrování nebo dešifrování. Útok postranním kanálem není přímo zaměřen na šifrovací algoritmus nebo klíč, ale na odhalení a zneužití suportivních informací, jak jsou spotřeba energie či elektromagnetické záření. [20]

### 3.1.4 Útok na vybraný otevřený text

Útok na vybraný otevřený text (Chosen-Plaintext Attack) je způsob útoku na kryptosystémy s využitím znalostí otevřeného textu a k němu odpovídajícímu zašifrovanému textu, kde si útočník vybere dvojice otevřeného a zašifrovaného text a snaží se na základě vztahu mezi těmito dvojicemi textů (vzory, polygramy, apod.) určit šifrovací klíč. [20]

### 3.1.5 Útok při znalosti otevřeného textu

Útok při znalosti otevřeného text (Known-Plaintext attack) je způsob útoku, kdy má útočník k dispozici některé páry otevřeného a zašifrovaného textu a snaží se na základě vztahu mezi těmito útočníkem nezvolenými dvojicemi určit šifrovací klíč. [20]

### 3.1.6 Diferenční kryptoanalýza

Diferenciální kryptoanalýza je způsob útoku určený na prolomení blokových kryptosystémů založený na statistické analýze, kdy se stanovuje pravděpodobnost klíče na základě analýzy párů otevřených textů a k nim odpovídajících zašifrovaných textů. [4]

### 3.1.7 Kvantové útoky

Kvantové útoky představují situaci, kdy s využitím výkonu kvantových počítačů bude možné ve velmi rychlém čase vyřešit matematické problémy soudobé kryptografie, jejichž řešení lze využít k prolomení moderních kryptosystémů. Mezi tyto problémy patří faktorizace velkých čísel a rychlé řešení úloh hledání diskrétních logaritmů nad klasickými tělesy a eliptickými křivkami. [21], [23]

## 3.2 Kritéria výběru kryptografického algoritmu

Předtím, než budeme posuzovat využitelnost kryptografických algoritmů pro konkrétní případy užití, zavedeme kritéria, podle kterých budeme algoritmy hodnotit.

- **Způsob rozdělení klíče** – rozdělení na symetrické a asymetrické algoritmy
- **Délka klíče** – odolnost vůči útokům (bude uváděn v bitech)
- **Počet možných klíčů**
- **Rychlost prolomení útokem hrubou silou (BFA)** – bude vypočítáno dle vzorce:

$$\mathrm{BFA} = \frac{\text{počet možných klíčů}}{\text{počet kombinací za sekundu}}$$

- **Odolnost vůči kvantovým útokům** (post-kvantová kryptografie)
- **Splnění omezujících kritérií dle NÚKIB** [22]

Tato kritéria byla zvolena na základě klíčových vlastností vybraných kryptografických algoritmů. Vždy je ovšem nutné mít na paměti, že omezujícími bezpečnostními kritérii zařazení algoritmů do porovnání jsou:

1. Náklady na prolomení šifry převyšují hodnotu zašifrované informace.

2. Čas potřebný na prolomení šifry převyšuje životnost informace. [3]

Pokud je alespoň jedno z těchto omezujících kritérií splněno, poté lze konstatovat, že šifrování je výpočetně bezpečné. [3]

## 3.3 Výběr kryptografických algoritmů

Informace při vytváření následujících tabulek byly zpracovány primárně z pramenů a dokumentací národních bezpečnostních institucí NIST a NÚKIB. Nutno dále podotknout, že při výpočtu atributu „BFA (ve sekundách)“ byla ve vzorci při určování počtu kombinací za sekundu užita hodnota $10^{16}$, která na teoretické rovině odpovídá počítači o výkonu 10 petaflop (floating point operation per second).

Tabulka 7: Výběr vhodných šifrovacích algoritmů na základě stanovených kritérií

| Algoritmus | Rozdělení klíče | Délka klíče | Počet možných klíčů |
|---|---|---|---|
| Caesarova šifra | Symetrické | pouze posun | 25 |
| Vigenèrova šifra | Symetrické | roven délce textu | $26^n$ |
| ROT13 | Symetrické | pouze posun | 1 |
| DES | Symetrické | 56 bitů | $2^{56}$ |
| 3DES | Symetrické | 168 bitů | $2^{168}$ |
| Blowfish | Symetrické | 32-448 bitů | $2^{32}$ až $2^{448}$ |
| AES | Symetrické | 128, 192 nebo 256 bitů | $2^{128}, 2^{192}, 2^{256}$ |
| RC4 | Symetrické | 40-256 bitů | $2^{40}$ až $2^{256}$ |
| ChaCha20 | Symetrické | 256 bitů | $2^{256}$ |
| RSA | Asymetircké | >4096 bitů | $> 2^{4096}$ |
| ElGamal | Asymetircké | >256 bitů | $> 2^{256}$ |
| ECC | Asymetircké | 521 bitů | $2^{521}$ |

| Algoritmus | BFA (ve sekundách) | Požadavky dle NÚKIB | Post-kvantové |
|---|---|---|---|
| Caesarova šifra | 0 | Ne | Ne |
| Vigenèrova šifra | 0 | Ne | Ne |
| ROT13 | 0 | Ne | Ne |
| DES (56 bit) | 3.6 | Ne | Ne |
| 3DES (168 bit) | $\sim 3.741 \times 10^{34}$ | - | Ne |
| Blowfish (256 bit) | $\sim 1.158 \times 10^{61}$ | Ano, pro délku klíče 128-256 bitů | Ano |
| AES (256 bit) | $\sim 1.158 \times 10^{61}$ | Ano | Ano |
| RC4 (256 bit) | $\sim 1.158 \times 10^{61}$ | Ne | Ne |
| ChaCha20 (256 bit) | $\sim 1.158 \times 10^{61}$ | Ano | Ano |
| RSA (4096 bit) | $\sim 1.044 \times 10^{1217}$ | Ano | Ne |
| ElGamal (2048 bit) | $\sim 3.232 \times 10^{600}$ | Ano | Ne |
| ECC (521 bit) | $\sim 6.865 \times 10^{140}$ | Ano | Ne |
| | | | |
| Nevhodná šifra | | | |
| Vhodná šifra | | | |

*Zdroj: zpracováno na základě dat z: [19], [22], [23], [24], [27], [34], [35], [40]*

# 4 Aplikace kryptografických algoritmů

Případů užití kryptografických algoritmů je nespočet. Od běžného šifrování otevřeného textu, přes zabezpečení internetových stránek, až po zajištění anonymity v distribuovaných decentralizovaných databázích typů Blockchain.

Ovšem výběr toho správného kryptografického algoritmu představuje složitý úkol, zejména kvůli široké škále dostupných, stále rozšiřujících se možností.

Je tedy potřeba se na základě stanovených kritérií umět rozhodnout mezi více alternativami řešení, což z úkolu činí rozhodovací problém. Existuje mnoho metod, kterými lze řešit problémy týkající se rozhodování, za zmínku stojí metoda Fullerova trojúhelníku, Saatyho metoda, či metoda PROMETHEE [26].

Rozhodovací procesy ovšem nejsou zaměřením této práce a následující text bude primárně sloužit jako stručný a praktický návod pro čtenáře, který si na jeho základě bude moci sám ohodnocovat kritéria výběru, vyčleňovat omezující kritéria a volit alternativy řešení. [26]

## 4.1 Podpora výběru kryptografického algoritmu

Výběr vhodného kryptografického algoritmu je zásadní při vytváření logického bezpečnostního řešení. Různé algoritmy disponují různými vlastnostmi, jak kladnými, tak zápornými pro vybrané případy užití.

Při rozhodování je zapotřebí nejprve stanovit omezující kritéria, která nám redukují konečný soubor potenciálních alternativ řešení.

### 4.1.1 Požadavek na bezpečnost

Nejzásadnějším požadavkem pro výběr šifrovacího algoritmů je jeho samotná bezpečnost. Tím je myšleno jeho odolnost vůči známým typům útoků a kvantovým útokům.

Bezpečný algoritmus by měl být zahrnut v seznamu doporučených šifrovacích algoritmů národní bezpečnostní autoritou, jako je NIST nebo NÚKIB. Pro obecný standard budeme považovat algoritmus za bezpečný, jsou-li splněna omezující bezpečnostní kritéria [3], a zároveň je uveden (nebo jeho příbuzná alternativa) v seznamu „Minimální požadavky na kryptografické algoritmy“ od NÚKIB. [22]

### 4.1.2 Požadavek na rychlost

V mnoha případech je rychlost šifrování nejvíce ohodnoceným kritériem, obzvlášť když se jedná o zařízení s nízkým výpočetním výkonem (například zařízení IoT) bez hardwarové akcelerace nebo při práci s velkým objemem dat. Pokud je rychlost a výpočetní náročnost při šifrování prioritizována před požadavkem na bezpečnost, volíme symetrické šifrovací algoritmy před asymetrickými. V případě, že nastane situace, kdy je omezující podmínkou použití asymetrického šifrovacího algoritmu, pak se musí nahlédnout do specifických vlastností jednotlivých algoritmů, případně problém řešit kombinací různých algoritmů v rámci hybridního kryptosystému. [8]

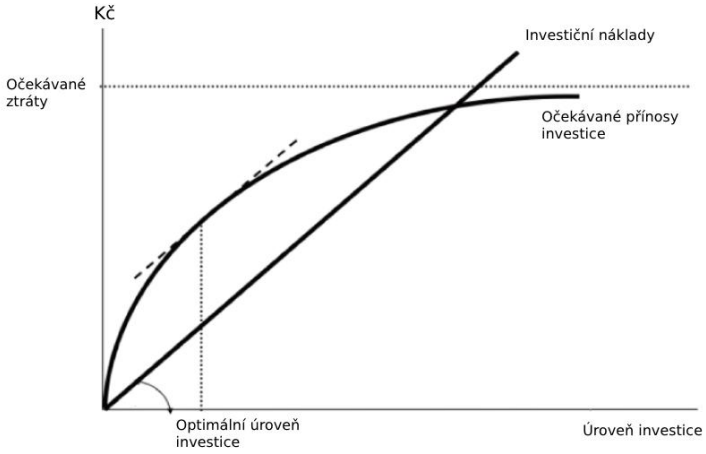
V situaci, kdy máme do méně výkonného zařízení implementovat asymetrický šifrovací algoritmus se ohlížíme na výpočetní náročnost daných alternativ řešení. Algoritmus ECC je méně výpočetně náročný než algoritmus RSA při stejné délce klíče. Ovšem algoritmus RSA je pro mnoho zavedených systémů více kompatibilní a snadno implementovatelný. [4]

### 4.1.3 Ekonomické aspekty a životnost

Zavádění kryptografického algoritmu do firemního prostředí může s sebou nést významné ekonomické dopady.

Ekonomický aspekt kryptosystému můžeme brát jako investici do bezpečnosti podnikových informací, které by při úniku mohly způsobit ekonomické a právní problémy. Příkladem může být odcizení firemní databáze o zaměstnancích (to znamená i jejich údajů spadající pod GDPR) narušitelem a její následné zveřejnění. V takovém případě je nutné neprodleně podat hlášení na Úřad pro ochranu osobních údajů. Dalšími příklad ohrožující ekonomiku podniku je únik utajovaných dat týkající se produktů a služeb.

Samotné určení optimální finanční investice do informační bezpečnosti podniku lze vyjádřit pomocí Gordon-Loebova modelu, znázorňující důležitost tohoto kritéria. Nutno ale konstatovat, že není možné dosáhnout 100% bezpečnosti. [29]




Obrázek 15: Gordon-Loebův model investice

*Zdroj: upraveno podle [29]*

## 4.2 Případy užití kryptografických algoritmů

Nyní si představíme konkrétní případy užití vybraných kryptografických algoritmů.

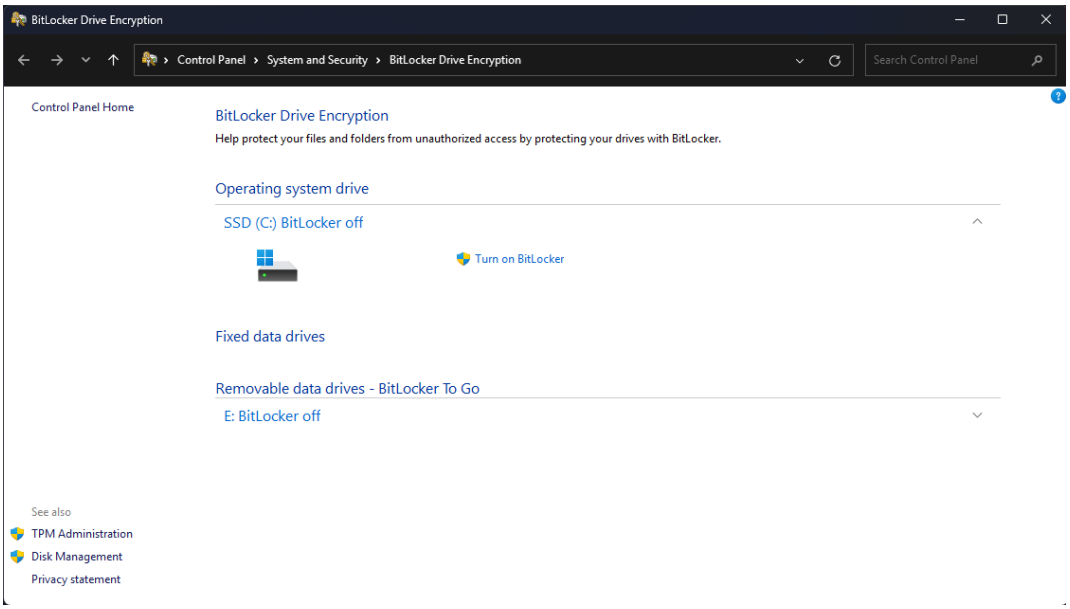
Jak již bylo zmíněno v kapitole 3.2, při rozhodování jsou stanoveny omezující kritéria výběru. Dále bude rozhodování ovlivněno na základě těchto pravidel:

1. Symetrické šifry jsou výpočetně méně náročné než asymetrické [8]
2. Preferujeme blokové šifry před proudovými [22]
3. Zohledňujeme vždy doporučenou délku šifrovacího klíče dle NÚKIB, případně NIST

### 4.2.1 Šifrování dat na disku

Při výběru kryptografického algoritmu pro šifrování dat na disku je vhodné nastavit rychlost šifrování jako velmi významné kritérium. Při šifrování dat na disku probíhá šifrování v podstatě neustále. A to při práci s velkým objemem dat vyžaduje vysoký výkon systému. Zvolením výpočetně náročného algoritmu by se celý proces načítání, zápisu a ukládání souborů zpomalil, což by velmi negativně ovlivnilo uživatelský zážitek. Vzhledem k tomu, že časová a výkonnostní nenáročnost je doménou symetrických kryptografických algoritmů, a jelikož preferujeme blokové šifry před proudovými, nabízí se využití šifrovacího algoritmu algoritmu AES.

Možnost šifrování dat na disku pomocí algoritmu AES lze přímo v operačním systému Windows 11 Pro pomocí integrovaného SW nástroje BitLocker.




Obrázek 16: BitLocker

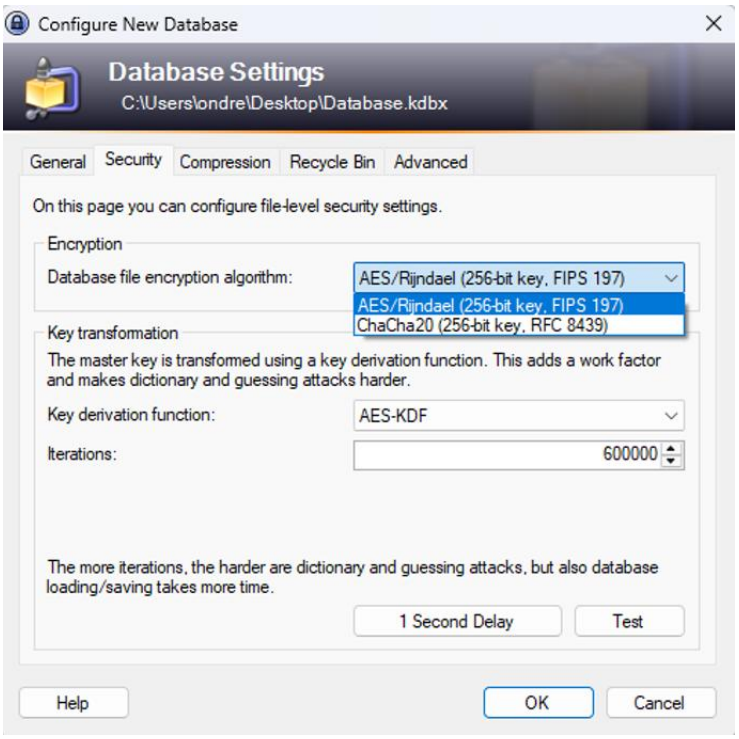
*Zdroj: vlastní zpracování*

### 4.2.2 Ochrana hesel

Bezpečné ukládání hesel je klíčovou součástí řízení přístupu. [8] Jelikož s rostoucími požadavky na hesla roste i obtížnost jejich zapamatování, tak stále více uživatelů inklinuje k alternativní SW řešením zvanými jako „správci hesel“, do kterých lze ukládat přihlašovací údaje, které jsou SW šifrovány. Mezi populární správce hesel patří KeePass, LastPass a Bitwarden.

Správce hesel KeePass je obzvláště oblíbený z důvodu, že se jedná o open-source SW s otevřenou licencí. Funguje na principu lokální databáze, která je chráněna primárním heslem, což má za důsledek, že uživatel namísto toho, aby si musel do každého systému pamatovat unikátní heslo, si musí pamatovat pouze ono primární heslo k databázi. [25]

Podobně jako je to u šifrování dat na disku se k ochraně hesel prostřednictvím správců hesel využívají symetrické šifrovací algoritmy, které nabízejí akceptovatelný poměr rychlosti a bezpečnosti. Správce hesel KeePass od verze 2.X využívá kryptografické algoritmy AES a ChaCha20. [25]




Obrázek 17: Zabezpečení hesel v SW KeePass

*Zdroj: vlastní zpracování*

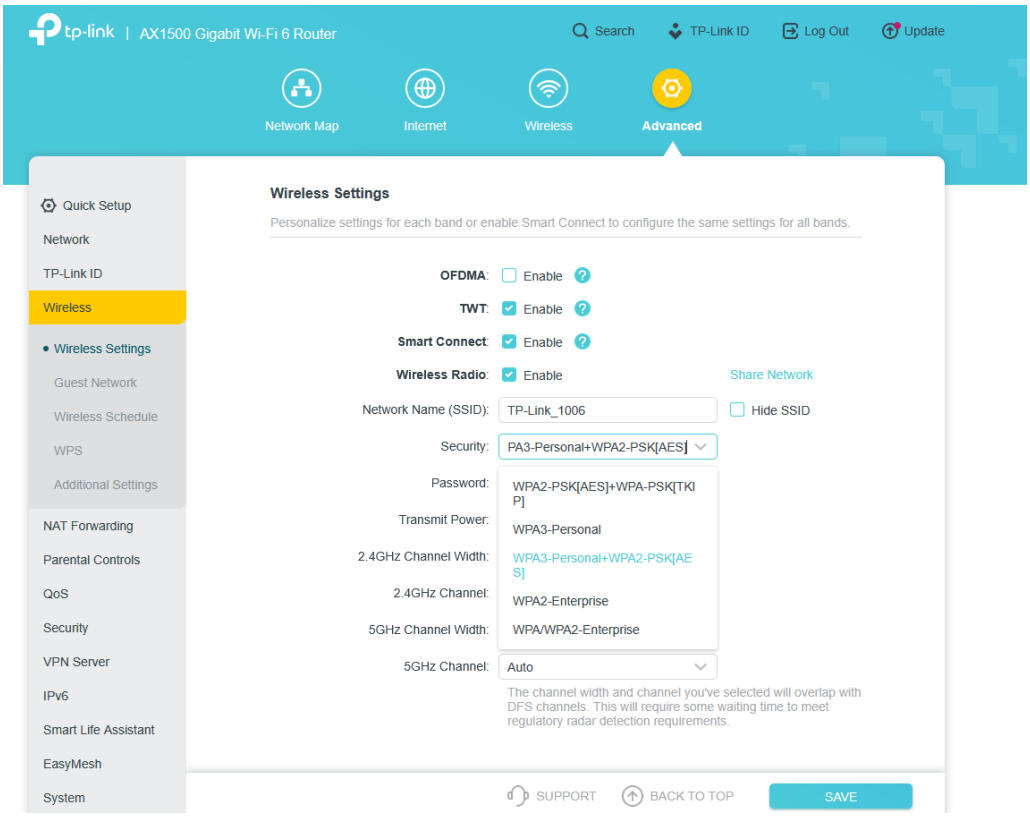
### 4.2.3 Zabezpečení Wi-Fi sítě

S přibývajícím počtem uživatelů na internetu přibývá i počet bezdrátových sítí [31], pro které je zapotřebí poskytnou dostatečnou úroveň zabezpečení. Tato potřeba vyvolala požadavek na vývoj jednoho z bezpečnostních protokolů WPA (Wi-Fi Protected Access), jejíž třetí iterace WPA3 vydaná v roce 2018 společností Wi-Fi Alliance je jedním z nynějších bezpečnostních standardů pro zabezpečení Wi-Fi sítě v rámci síťového směrovače (routeru). [30], [32]

Při výběru vhodného zabezpečení Wi-Fi sítě je pro nás zásadní informace, že samotný router zpracovává velký objem dat v reálném čase, tedy je pro nás významným kritériem rychlost šifrování. Dalším kritériem je výpočetní náročnost šifrovacího řešení, jelikož router má (i při hardwarové akceleraci) velmi omezený výkon.

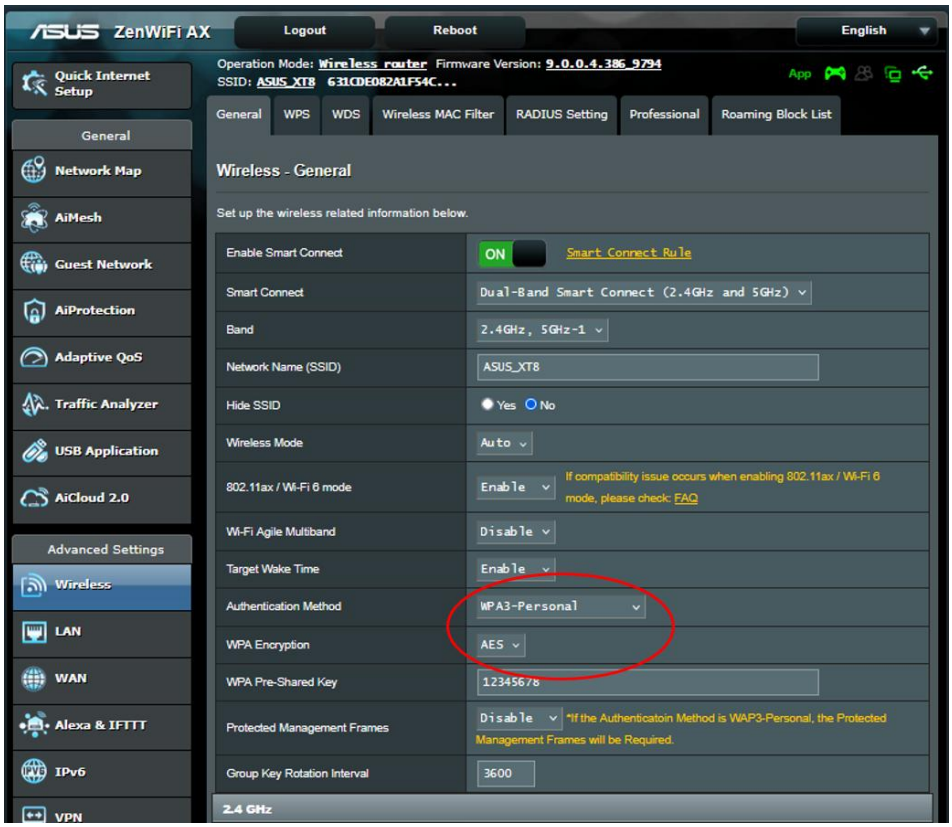
Samotná aktuálnost informací, se kterými směrovač pracuje je velmi krátká, z čehož vyplývá, že bezpečnostní výhody, které by přinesl asymetrický šifrovací algoritmus by nedokázal efektivní využít, proto pro zabezpečení Wi-Fi sítě je nejvhodnější zvolit symetrický šifrovací algoritmus, který je součástí certifikovaného bezpečnostního protokolu. [4]

Nastavení zabezpečení Wi-Fi sítě se liší v závislosti na konkrétním modelu routeru. Moderní routery umožňující nové technologie, jako je Wi-Fi 6 většinou disponují více možnostmi zabezpečení. Obvykle se pro zabezpečení Wi-Fi sítě využívá druhé, či třetí iterace protokolu WPA, nebo CCMP se šifrovacím algoritmem AES. [32]




Obrázek 18: Zabezpečení Wi-Fi sítě v routeru TP-LINK AX1500

*Zdroj: snímek obrazovky z: https://emulator.tp-link.com*




Obrázek 19: Zabezpečení Wi-Fi sítě v routeru ASUS RT-AX95Q

*Zdroj: snímek obrazovky z: https://demoui.asus.com*

### 4.2.4 Ověření identity

Jedním ze způsobů, jak v kyberprostoru ověřit identitu odesílatele zprávy je skrze digitální podpis, kterým lze zajistit autenticitu a integritu přijaté zprávy.

Při volbě šifrovacího řešení pro ověření identity odesílatele je nezbytné brát ohled na problém distribuce klíčů (viz. kapitola 2.4), se kterým se potýkají všechny symetrické kryptosystémy. Tento problém nám omezí množinu řešení pouze na asymetrické šifrovací algoritmy.

Samotné řešení může je ovlivněno faktorem kompatibility, kdy pro starší, již zavedené systémy, může z hlediska snadnější implementace být vhodnější zvolit šifrovací algoritmus RSA. Ovšem z hlediska budoucího vývoje kvantové kryptografie je nutné zohlednit faktor potenciálních kvantových útoků, proti kterým není algoritmus RSA odolný. [22] V takovém případě se nabízí využít rychlejší, zdrojově efektivnější šifrovací algoritmus ECC (ECDSA), který dle NÚKIB by měl být v budoucnu odolný vůči kvantovým útokům. [22], [23]

### 4.2.5 Šifrování textu ve vývojovém prostředí

Následující podkapitola primárně slouží jako ukázka implementace vybraných algoritmů pro šifrování textu ve vývojovém prostředí Microsoft Visual Studio Code a programovacím jazyce Python. Samotné kódy programů jsou uvedeny v Příloze 2, 3, 4.

### 4.2.6 Zabezpečení webové stránky

Klíčovým prvkem ochrany dat na webových stránkách mezi uživateli a serverem je nasazením bezpečnostního protokolu obsahující silné šifrovací algoritmy. Kritérium bezpečnosti je zde nejvýznamnější, což tíhne k využití asymetrické šifrovacího algoritmu, jakým je RSA nebo ECC. Avšak pořád je zde velmi významný požadavek na rychlost, jelikož chceme, aby se nám obsah zpráv šifroval co nejrychleji a byla tak komunikace se serverem svižná. V praxi je efektivním řešení nasazení hybridního kryptosystému [8], jakým je v případě webové komunikace protokol Hypertext Transfer Protocol Secure (HTTPS), u kterého je asymetrické šifrování (RSA nebo ECC) delegováno na úlohu výměny šifrovacích klíčů, pomocí kterých lze ověřit totožnost a tím zajistit autenticitu. Následný obsah zprávy je však šifrován pomocí symetrického šifrovacího algoritmu AES [3]. Tímto způsobem se dá zajistit bezpečnost rychlého přenosu dat mezi klientem a serverem bez nutnosti řešení problému s distribucí klíčů.

# Závěr

Tato bakalářská práce se zaměřovala na oblast kryptografie, kryptografických algoritmů a s nimi spojenou problematiku informační bezpečnosti, která je v současné době čím dál aktuálnějším tématem, vzhledem k narůstajícímu počtu kybernetických útoků. Hlavním tématem práce bylo porovnání vybraných kryptografických algoritmů na základě stanovených kritérií, vycházejících z obecného popisu jednotlivých způsobů šifrování a dokumentace národních bezpečnostních institucí.

Jednotlivé kryptografické algoritmy byly posuzovány a popisovány od triviálních až po komplexní, v současné době používané, jakými jsou AES, RSA či ECC pomocí zjednodušeného popisu a matematického, případně grafického aparátu, což sloužilo jako teoretický základ pro pochopení principu jejich fungování.

Praktická část práce byla zaměřena na vyhodnocení efektivnosti vybraných kryptografických algoritmů v kyberprostoru. To bylo realizováno analýzou jednotlivých algoritmů, popisem jejich vlastností, i když v mnoha případech jen teoretických a následné porovnání, na jehož základě byl sestaven „návod“ sloužící pro čtenáře jako podpora při ohodnocování kritérií výběru a určování alternativ řešení v rámci rozhodovacího procesu.

Dále byly v práci popsány běžné případy užití kryptografických algoritmů a následné doporučení. Součástí práce je také implementace vybraných kryptografických algoritmů prostřednictvím vývojového prostředí a programovacího jazyka Python.

Zpracování této práce mi dalo teoretický i praktický úvod do kryptografie, včetně osvojení matematických disciplín, jako jsou teorie čísel nebo abstraktní algebra a donutilo mě to se zamyslet nad budoucností implementace logických bezpečnostních mechanismů vzhledem k čím dál více se přibližující době kvantových počítačů.

# Použitá literatura

[1] BURDA, Karel. *Úvod do kryptografie*. Brno: Akademické nakladatelství CERM, 2015. ISBN 978-80-7204-925-7

[2] LEPKA, Karel. *Základy elementární teorie čísel*. Brno: Munipress, 2023. ISBN 978-80-280-0423-1

[3] PELZL, Jan, PAAR, Christof. *Understanding Cryptography: A Textbook for Students and Practitioners*. Berlin: Springer, 2010. ISBN 978-3-642-04101-3

[4] STALLINGS, William. *Cryptography and Network Security, Eighth Edition, Global Edition*. Pearson, 2022. ISBN 978-1-292-43748-4

[5] JANČÍKOVÁ, Zora. *Teorie systémů*. Ostrava: VŠB – Technická univerzita Ostrava, 2010. ISBN 978-80-248-2561-8

[6] CORMEN, Thomas, LEISERSON, Charles, RIVEST, Ronald, STEIN, Clifford. *Introduction to algorithms, Fourth Edition*. Cambridge: The MIT Press, 2022. ISBN 978-0262046305

[7] ČAPEK, Jan, MÁCHOVÁ, Renáta. *Teoretické základy informatiky: distanční opora. Vyd. 3., upr., rozš*. Pardubice: Univerzita Pardubice, 2013. ISBN 978-80-7395-574-8.

[8] HUB, Miloslav. *Bezpečnost a ochrana informací v prostředí internetu*. Pardubice: Univerzita Pardubice, 2013. ISBN 978-80-7395-701-8

[9] DANEL, Roman, 2011. *Informační systémy* [online]. Katedra systémového inženýrství a informatiky VŠB-TUO [cit. 2025-04-16]. Dostupné z: https://homel.vsb.cz/~dan11/rd_is_skripta.htm

[10] *Metodika stanovení požadavků na bezpečnost IS - příloha č. 4 souhrnné analytické zprávy* [online], 2018. Ministerstvo vnitra [cit. 2025-04-16]. Dostupné z: https://mv.gov.cz/

[11] KER, Andrew, 2014. *Computer Security* [online]. Department of Computer Science, Oxford University [cit. 2025-04-16]. Dostupné z: https://www.cs.ox.ac.uk/andrew.ker/docs/computersecurity-lecture-notes-mt2014.pdf

[12] CHRISTENSEN, Chris, 2019. *Cryptography of the Vigenère Cipher* [online]. Northern Kentucky University [cit. 2025-04-16]. Dostupné z:

https://websites.nku.edu/~christensen/1901csscmat483%20section%2013%20vigenere%20cryptography.pdf

[13] MALDONADO, Roberto, MOREANO Patricio, CARRERA Pablo, INTURRALDE, Mauricio, 2015. *An Hybrid Encryption Mechanism for short Text messaging in Mobile Devices* [online]. Universidad San Francisco de Quito [cit. 2025-04-16]. Dostupné z: https://www.researchgate.net/publication/280578941_An_Hybrid_Encryption_Mechanism_for_short_Text_messaging_in_Mobile_Devices

[14] KHANDURI, Ayush, 2024. *Vigenère Cipher* [online]. GeeksforGeeks [cit. 2025-04-16]. Dostupné z: https://www.geeksforgeeks.org/vigenere-cipher/

[15] AMIROVÁ, Kamilla. *Úvod do kryptografie - Proudové šifry* [online]. 2007. ČVUT [cit. 2025-04-16]. Dostupné z: https://sifrovani.fd.cvut.cz/prou_sifr.html

[16] KOZLÍK, Andrew. *Data Encryption Standard (DES)* [online]. Katedra algebry, Matematicko-fyzikální fakulta, Univerzita Karlova [cit. 2025-04-16]. Dostupné z: https://www.karlin.mff.cuni.cz/~kozlik/udk_mat/des.pdf

[17] KAMILLA, Amirová, 2007. *Úvod do kryptografie - Algoritmus DES* [online]. ČVUT [cit. 2025-04-16]. Dostupné z: https://sifrovani.fd.cvut.cz/des.html

[18] RAZA, Muhammad, 2023. *The Triple DES Intro: Triple Data Encryption Standard* [online]. Splunk Technology [cit. 2025-04-16]. Dostupné z: https://www.splunk.com/en_us/blog/learn/triple-des-data-encryption-standard.html

[19] BARKER, Elaine, MOUHA Nicky. *Recommendation for the Triple Data Encryption Standard (TDEA) Block Cipher* [online]. Computer Security Division, Information Technology Laboratory, NIST [cit. 2025-04-16]. Dostupné z: https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-67r2.pdf

[20] BURGE, Simon, 2024. *8 Types of Attack in Cryptography* [online]. International Security Journal [cit. 2025-04-16]. Dostupné z: https://internationalsecurityjournal.com/types-of-attack-in-cryptography/#8_Types_of_Attack_in_Cryptography

[21] CAMPAGNA, Matthew, et al, 2015. *Quantum Safe Cryptography and Security - An introduction, benefits, enablers and challenges* [online]. European Telecommunications Standards Institute. [cit. 2025-04-16]. ISBN 979-10-92620-03-0. Dostupné z: https://www.etsi.org/images/files/ETSIWhitePapers/QuantumSafeWhitepaper.pdf

[22] *Minimální požadavky na kryptografické algoritmy - doporučení v oblasti kryptografické bezpečnosti* [online], 2025. NÚKIB [cit. 2025-04-16]. Dostupné z: https://nukib.gov.cz/download/publikace/podpurne_materialy/Minimalni_pozadavky_v4_FINAL.pdf

[23] *Kvanotvá hrozba a kvantově odolná kryptografie* [online], 2025. NÚKIB [cit. 2025-04-16]. Dostupné z: https://nukib.gov.cz/cs/infoservis/dokumenty-a-publikace/podpurne-materialy/

[24] BARKER, Elaine a ROGINSKY, Allen, 2019. *Transitioning the Use of Cryptographic Algorithms and Key Lengths* [online]. Computer Security Division, Information Technology Laboratory, NIST [cit. 2025-04-16]. Dostupné z: https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-131Ar2.pdf

[25] *Security - KeePass* [online]. KeePass Help Center [cit. 2025-04-16]. Dostupné z: https://keepass.info/help/base/security.html

[26] KŘUPKA, Jiří, KAŠPAROVÁ Miloslava, MÁCHOVÁ Renáta, 2012. *Rozhodovací procesy*. Ústav systémového inženýrství a informatiky, Fakulta ekonomicko-správní, Univerzita Pardubice. ISBN 978-80-7395-478-9.

[27] HU, Vincent, 2023. *Overview and Considerations of Access Control Based on Attribute Encryption* [online]. Computer Security Division, Information Technology Laboratory, NIST [cit. 2025-04-16]. Dostupné z: https://nvlpubs.nist.gov/nistpubs/ir/2023/NIST.IR.8450-upd1.pdf

[28] HANKERSON, Darrel, MENEZES Alfred, 2011. *Elliptic Curve Discrete Logarithm Problem* [online]. Springer [cit. 2025-04-16]. ISBN 978-1-4419-5905-8. Dostupné z: https://link.springer.com/referenceworkentry/10.1007/978-1-4419-5906-5_246#citeas

[29] GORDON, Lawrence, et al, 2016. *Investing in Cybersecurity: Insights from the Gordon-Loeb Model. Journal of Information Security* [online]. Scientific Research - An Academic

Publisher [cit. 2025-04-16]. Dostupné z: https://www.scirp.org/journal/paperinformation?paperid=64892

[30] WRIGHT, Gavin, GILLIS Alexander. *What is WPA3 (Wi-Fi Protected Access 3)?* [online]. TechTarget [cit. 2025-04-16]. Dostupné z: https://www.techtarget.com/searchsecurity/definition/WPA3

[31] PETROSYAN, Ani, 2024. *Number of internet users worldwide from 2005 to 2024* [online]. Statista [cit. 2025-04-16]. Dostupné z: https://www.statista.com/statistics/273018/number-of-internet-users-worldwide/

[32] *WPA3™ Specification* [online], 2024. Wi-Fi Alliance [cit. 2025-04-16]. Dostupné z: https://www.wi-fi.org/system/files/WPA3%20Specification%20v3.3.pdf

[33] *Comparing ECDSA vs RSA: A Simple Guide* [online], 2024. SSL [cit. 2025-04-16]. Dostupné z: https://www.ssl.com/article/comparing-ecdsa-vs-rsa-a-simple-guide/

[34] ROMINE, Charles, 2023. *DIGITAL SIGNATURE STANDARD (DSS)* [online]. Information Technology Laboratory, NIST [cit. 2025-04-16]. Dostupné z: https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.186-5.pdf

[35] DWORKIN, Morris, et al, 2001. *Advanced Encryption Standard (AES)* [online]. NIST [cit. 2025-04-16]. Dostupné z: https://doi.org/10.6028/NIST.FIPS.197

[36] PERTIWI, Ayu, FAUZI, SYAHPUTRA, Siswan, 2023. *Application Of Super Encryption Using Rot 13 Algorithm Method and Algorithm Beaufort Cipher For Image Security Digital* [online]. Journal of Artificial Intelligence and Engineering Applications [cit. 2025-04-19]. Dostupné z: http://dx.doi.org/10.59934/jaiea.v3i1.263

[37] MILLER, Gary, 1975. *Riemann's Hypothesis and Tests for Primality* [online]. Department of Mathematics, University of California: Journal of Computer and System Sciences [cit. 2025-04-19]. Dostupné z: https://dl.acm.org/doi/10.1145/800116.803773

[38] BEČVÁŘ, Jindřich, 2005. *Lineární algebra* [online]. Matematicko-fyzikální fakulta, Univerzita Karlova: Matfyzpress [cit. 2025-04-19]. ISBN 80-86732-57-6. Dostupné z: https://www.karlin.mff.cuni.cz/~halas/becvar_-_linearni_algebra.pdf

[39] GIBSON, Taylor, 2021. *The Tabula Recta* [online]. NCSSM [cit. 2025-04-19]. Dostupné z: https://macs4200.org/chapters/07/1/tabula-recta#the-tabula-recta

[40] BARKER, Elaine, MOUHA, Nicky, 2024. *Recommendation for the Triple Data Encryption Algorithm (TDEA) Block Cipher* [online]. Computer Security Division, Information Technology Laboratory, NIST [cit. 2025-04-19]. Dostupné z: https://doi.org/10.6028/NIST.SP.800-67r2

[41] SELIMIS, Georgios, 2007. *A Low Power Design for Sbox Cryptographic Primitive of Advanced Encryption Standard for Mobile End-Users* [online]. Journal of Low Power Electronics [cit. 2025-04-19]. Dostupné z: https://www.researchgate.net/publication/220091765_A_Low_Power_Design_for_Sbox_Cryptographic_Primitive_of_Advanced_Encryption_Standard_for_Mobile_End-Users

[42] FRANKEL, Sheila, HOFFMAN, Paul, OREBAUGH, Angela, PARK Richard, 2008. *Guide to SSL VPNs - Recommendations of the National Institute of Standards and Technology* [online]. Computer Security Division, Information Technology Laboratory, NIST [cit. 2025-04-20]. Dostupné z: https://doi.org/10.6028/NIST.SP.800-113

[43] SCHNEIER, Bruce, 1996. *Applied Cryptography, Second Edition: Protocols, Algorithms, and Source Code in C (cloth)* [online]. John Wiley & Sons [cit. 2025-04-20]. ISBN 0471128457.

[44] HEMEIDA, Farah, ALEXAN, Wassim, SADEK, Salma, 2019. *Blowfish-Secured Audio Steganography* [online]. Novel Intelligent and Leading Emerging Sciences conference [cit. 2025-04-20]. Dostupné z: http://dx.doi.org/10.1109/NILES.2019.8909206

[45] NIR, Y., LANGLEY, A., 2018. *ChaCha20 and Poly1305 for IETF Protocols* [online]. RFC 8439. [cit. 2025-04-20]. Dostupné z: https://doi.org/10.17487/RFC8439

# Přílohy

## Příloha č.1 – Miller-Rabinův test prvočíselnosti v programovacím jazyce Python

```python
def miller_rabin(a, n):

    print(f"n = {n}, a = {a}")
    print(f"Faktorizace a^(n-1) - 1")

    k = 0 #horní index mocniny ve vzorci
    exp = (n - 1) // (2 ** k) #exponent koeficientu a

    while (n - 1) % (2 ** k) == 0:
        exp = (n - 1) // (2 ** k)
        value = pow(a, exp, n)

        sign = '-' if k == 0 else '+'
        print(f"k = {k}: (a^({n-1}/2^{k}) {sign} 1) = (a^{exp} {sign} 1) ≡ {value} mod {n}")

        if (value - 1) % n == 0 or (value + 1) % n == 0:
            print(f" {n} dělí alespoň jeden z výrazů, test prošel pro k = {k}")
            return True

        k += 1

    print(f"Ani jeden výraz není dělitelný {n}, číslo není prvočíslo.")
    return False

# Příklad
a = 2 #hodnota koeficientu, z definice musí být splněna podmínka: 1 < a < n-1
n = 1000000009 #celé číslo, u kterého chceme dokázat, že náleží množině prvočísel

is_prime = miller_rabin(a, n)

if is_prime:
    print(f"\nVýsledek: {n} podle testu náleží množině prvočísel s P(3/4)")
else:
    print(f"\nVýsledek: {n} podle testu nenáleží množině prvočísel")
```

## Příloha č.2 – Implementace Caesarovy šifry v programovacím jazyce Python

```
alphabet = [
    "a", "á", "b", "c", "č", "d", "ď", "e", "é", "ě",
    "f", "g", "h", "ch", "i", "í", "j", "k", "l", "m", "n", "ň",        # Množina znaků, které budeme posouvat
    "o", "ó", "p", "q", "r", "ř", "s", "š", "t", "ť",
    "u", "ú", "ů", "v", "w", "x", "y", "ý", "z", "ž"
]

def encrypt(plaintext, n):
    ans = ""
    plaintext = plaintext.lower()

    for char in plaintext:
        if char in alphabet:
            index = alphabet.index(char)
            n_index = (index + n) % len(alphabet)   #Cyklus, který posune otevřený o n počet znaků vpravo
            ans += alphabet[n_index]
        else:
            ans += char

    return ans

plaintext = input("Zadejte text: ")
n = int(input("Zadejte klíč: "))

print("Otevřený text:", plaintext)
print("Klíč (posun vpravo):", n)
print("Zašifrovaný text:", encrypt(plaintext, n))
```

**Příloha č.3 – Implementace algoritmu AES v programovacím jazyce Python skrze knihovnu PyCryptodome**

```python
from Crypto.Cipher import AES
from secrets import token_bytes

KEY = token_bytes(16)

def encrypt(msg: str) -> tuple:
    cipher = AES.new(KEY, AES.MODE_EAX)
    nonce = cipher.nonce
    ciphertext, tag = cipher.encrypt_and_digest(msg.encode('utf-8'))
    return nonce, ciphertext, tag

def decrypt(nonce: bytes, ciphertext: bytes, tag: bytes) -> str | None:
    try:
        cipher = AES.new(KEY, AES.MODE_EAX, nonce=nonce)
        plaintext = cipher.decrypt(ciphertext)
        cipher.verify(tag)
        return plaintext.decode('utf-8')
    except (ValueError, KeyError):
        return False

def main():
    msg = input('Zadejte text: ')
    nonce, ciphertext, tag = encrypt(msg)
    print(f'Šifrovaný text: {ciphertext}')

    plaintext = decrypt(nonce, ciphertext, tag)
    if plaintext is None:
        print('Zpráva je poškozena')
    else:
        print(f'Otevřený text: {plaintext}')

if __name__ == '__main__':
    main()
```

## Příloha č.4 – Implementace RSA v programovacím jazyce Python

```python
def nsd(a, b):
    while b != 0:           #Definování největšího společného dělitele pomocí Eukleidova algoritmu
        a, b = b, a % b
    return a

def euklides_inverse_modulo(a, m):      #Rozšířený Eukleidův algoritmus
    m0, x0, x1 = m, 0, 1
    while a > 1:
        q = a // m
        m, a = a % m, m
        x0, x1 = x1 - q * x0, x0
    return x1 + m0 if x1 < 0 else x1

def primes(p, q):                       #Deklarace parametrů p, q
    n = p * q
    phi = (p - 1) * (q - 1)             #Výpočet Eulerovy funkce phi
    exp = 65537                           #Nejčastěji používaný exponent v RSA
    if nsd(exp, phi) != 1:
        raise ValueError("Není splněno pravidlo nesoudělnosti, zvol jiné parametry p a q.")
    d = euklides_inverse_modulo(exp, phi)
    return (exp, n), (d, n)

def encrypt(msg, public_key):
    exp, n = public_key
    return [pow(ord(char), exp, n) for char in msg]

def decrypt(cipher, private_key):
    d, n = private_key
    return ''.join([chr(pow(char, d, n)) for char in cipher])

p = 1000000009          # Hodnoty parametrů p a q, které musí být prvočísla (lze navázat na Miller-Rabinův test provčíselnosti)
q = 7727

public_key, private_key = primes(p, q)
print("Veřejný klíč:", public_key)
print("Soukromý klíč:", private_key)

message = input("Zadejte text: ")

encrypted = encrypt(message, public_key)
print("Zašifrovaný text:", encrypted)

decrypted = decrypt(encrypted, private_key)
print("Otevřený text:", decrypted)
```