Katedra informatiky
Přírodovědecká fakulta
Univerzita Palackého v Olomouci

# BAKALÁŘSKÁ PRÁCE

Enigma

2019

Tomáš Orlík

Vedoucí práce: doc. RNDr. Miroslav Kolařík, Ph.D.

Studijní obor: Informatika, prezenční forma

**Bibliografické údaje**

| | |
|---|---|
| Autor: | Tomáš Orlík |
| Název práce: | Enigma |
| Typ práce: | bakalářská práce |
| Pracoviště: | Katedra informatiky, Přírodovědecká fakulta, Univerzita Palackého v Olomouci |
| Rok obhajoby: | 2019 |
| Studijní obor: | Informatika, prezenční forma |
| Vedoucí práce: | doc. RNDr. Miroslav Kolařík, Ph.D. |
| Počet stran: | 34 |
| Přílohy: | 1 DVD |
| Jazyk práce: | český |

**Bibliograhic info**

| | |
|---|---|
| Author: | Tomáš Orlík |
| Title: | Enigma |
| Thesis type: | bachelor thesis |
| Department: | Department of Computer Science, Faculty of Science, Palacký University Olomouc |
| Year of defense: | 2019 |
| Study field: | Computer Science, full-time form |
| Supervisor: | doc. RNDr. Miroslav Kolařík, Ph.D. |
| Page count: | 34 |
| Supplements: | 1 DVD |
| Thesis language: | Czech |

## Anotace

*Práce popisuje šifrovací nástroj Enigma z pohledu informatiky, popisuje její nedostatky, řeší jejich odstranění a případné zlepšení síly šifry. Kryptografická složitost přístroje závisí na pěti klíčových komponentách, které jsou jednotlivě rozebrány v samostatných podkapitolách. Šifra s eliminovanými nedostatky byla využita v desktopové aplikaci Enigma, které se věnuje praktická část práce.*

## Synopsis

*Current thesis describes the Enigma cipher machine with an informatics approach. It presents machine's weak points and potential solutions for their improvement, namely improvement of cipher. The Enigma machine's cryptographic complexity depends on five key components, each of which is analyzed in separate subchapter. The cipher with eliminated shortcomings has been used in the desktop application Enigma, which is described in the practical part of the thesis.*

**Klíčová slova:** Enigma; šifra; .NET aplikace; kryptografie; substituce

**Keywords:** Enigma; cipher; .NET app; cryptography; substitution

Děkuji svému vedoucímu doc. RNDr. Miroslavu Kolaříkovi, Ph.D. za vedení mé práce, podporu a odborné rady. Dále bych chtěl poděkovat své rodině za oporu po celou dobu studia.

*Místopřísežně prohlašuji, že jsem celou práci včetně příloh vypracoval samostatně a za použití pouze zdrojů citovaných v textu práce a uvedených v seznamu literatury.*

datum odevzdání práce podpis autora

# Obsah

# Seznam obrázků



# Seznam tabulek



# Seznam zdrojových kódů

# 1 Úvod

První zmínky o kryptografii nalezneme již v materiálech ze starověkého Řecka. Tehdy bylo zapotřebí skrýt utajovanou komunikaci a varovat Řeky před Peršanem Xerxem. Z toho důvodu vznikla první nám známá šifra, která byla steganografická (z řeckého steganós – schovaný a gráphein – psát; jedná se tedy o ukrytí zprávy do předmětu, mluveného slova, nebo psaní zprávy na první pohled neviditelným materiálem). Současně se steganografií se rozvíjela také kryptografie (z řeckého kryptós – skrytý; v tomto případě již jde o šifrování, tedy změnu smyslu zprávy na základě změny její podoby), u které nebylo potřeba utajit existenci zprávy, ale její význam. Šifrování zpráv se dále dělí na transpoziční a substituční. Transpoziční šifra vznikne přeuspořádáním znaků, kdežto substituční šifra nahrazuje písmena jinými písmeny. První zmínku o substituční šifře překvapivě nenajdeme zmíněnou ve spojitosti s válkou, ale v Kámasútře ze 4. stol. př. n. l., kde bráhman Vátsjájana vybízí ženy ke studiu umění tajného písma, aby mohly ukrýt informace o svých vztazích. Pro vojenské účely byla substituční šifra poprvé využita Gaiem Juliem Caesarem v době galské války, kdy v dopise Ciceronovi, který byl v obklíčení a hrozila mu kapitulace, použil substituci římských písmen za řecké, kterým nikdo nerozuměl. Tak vznikly teoretické základy pro tvorbu Enigmy.

Enigmě se ve druhé světové válce důvěřovalo natolik, že jí byly šifrovány i ty nejcitlivější zprávy. Krásně to ilustruje výrok německého kontradmirála Ludwiga Stummela: „*Na základě důkladného a opakovaného vyšetřování se ukázalo, že je Enigma neprolomitelná.*“ [1] Historikové tvrdí, že prolomení Enigmy pomohlo zkrátit druhou světovou válku o zhruba dva až tři roky.

V práci chceme objasnit, jak silná je Enigma jako symetrická substituční šifra, na jakém principu pracuje a jak je ji možné z matematického pohledu prolomit. V teoretické části, zaměřené na Enigmu jako na šifrovací nástroj, objasňujeme, v čem tkvěla největší kryptografická síla Enigmy, pokusíme se najít její slabiny a vysvětlit, jak mohlo dojít k prolomení šifry.

V praktické části práce navrhujeme, jak odstranit slabé stránky Enigmy, případně jak řádově zlepšit jejich šifrovací sílu. Silnější řešení, založené na stejném konceptu šifrování, jako má přístroj Enigma, pak zapracováváme do aplikace, která je součástí této závěrečné práce. Klademe si tedy za cíl vytvořit díky opraveným chybám silnější šifrovací nástroj, který je možné využívat v praxi. Od aplikace chceme, aby byla uživatelsky přívětivá, a proto k ní dodáváme uživatelskou příručku.

# 2 Enigma

## 2.1 Vnější charakteristika stroje Enigma

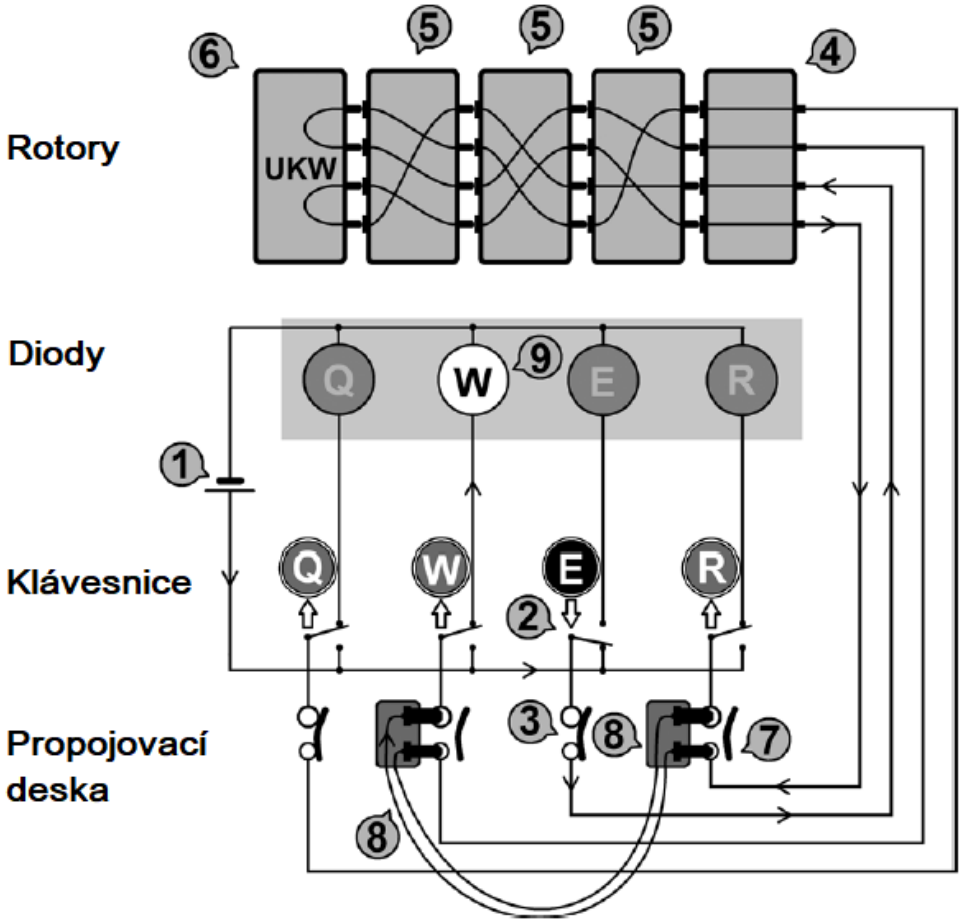
Stroj Enigma nápadně připomíná psací stroj, jehož hlavní částí je QWERTZ klávesnice, pomocí které je zadáván otevřený text. Otevřeným textem rozumíme nezašifrovaný text, který dostal operátor k zašifrování. Dále je zde sada diod s jednotlivými písmeny abecedy. Ty slouží pro zobrazení zašifrovaného textu. V horní části zařízení lze nalézt rotory, které jsou hlavními prvky Enigmy. Rotory jsou umístěny vedle sebe, mají na okraji vyražené znaky abecedy a při stisku klávesy na klávesnici se rotor, který je umístěn nejvíce vpravo, otočí o jednu pozici ve směru hodinových ručiček. Každý rotor má na sobě zarážku. Zarážka slouží k tomu, aby se následující rotor pootočil o jednu polohu. Princip je podobný jako u tachometru v automobilu. Vnitřní propojení znaků je závislé na konkrétním rotoru. Zde je využito první substituce, tzn. že každý znak je zaměněn za jiný znak. Na konci rotorů se nachází reflektor. I reflektor má pevně dané propojení jednotlivých písmen. Narozdíl od rotorů je jeho pozice pevná, neotáčí se a signál nepokračuje do dalších rotorů, ale zpět do původních rotorů. Signál se vrací stejnými rotory, ale jinou cestou vzhledem k principu fungování reflektoru. Než signál dojde k diodě zobrazující zašifrovaný znak, projde propojovací deskou (Steckerbrett). Ta umožňuje propojení dvou znaků, čímž jsou tyto dva znaky prohozeny. Propojení je provedeno zasunutím dvojitého konektoru do jednoho znaku a na opačné straně vodiče dvojitým konektorem do znaku druhého. Dvojitým konektorem je zajištěno, aby se znak substituoval za jiný a pokaždé stejný, když jde signál směrem z klávesnice k rotorům, ale také zpět od rotorů k diodám. Nepropojené znaky zůstávají v této fázi nezměněny. Běžně se za války používalo deset vodičů, deset substitucí.

Enigma je šifrovací stroj, který byl používán v první polovině dvacátého století. Své největší využití našel během druhé světové války u nacistického Německa. Z kryptologického pohledu se jedná o nástroj polyalfabetické substituční šifry, která spadá do symetrických šifer. Enigma je také elektromechanické zařízení, které po stisknutí klávesy na klávesnici spojí elektrický obvod, kde je elektrický proud veden přes spojovací desku, rotory a reflektor až do lamp, které značí zašifrovaný znak. V rámci práce se budeme zmiňovat o přístroji a aplikaci, vždy se bude jednat o Enigmu a to v podobě přístroje a software. Obecné informace týkající se přístroje, byly čerpány z webu Crypto musea. [2]

## 2.2 Popis mechanického procesu šifrování

Pro lepší představu, jak probíhá mechanický proces šifrování, poslouží schéma na obrázku 1. Abychom se v něm lépe orientovali, obsahuje pouze 3 rotory a pouze 4 vybrané znaky Q, W, E a R.

Při stisku tlačítka teče proud z baterie (1) přes stisknuté dvoucestné tlačítko (2) do propojovací desky (3), odkud se dostane do pevného vstupního kola (4). Pokračuje do tří rotorů (5) a po nich do reflektoru (6). Schéma znázorňuje, jak




Obrázek 1: Schéma elektrického obvodu Enigmy upraveno ze zdroje [3]

pohyblivé části, kterými jsou rotory, určují, které znaky budou jak substituovány. Reflektor má vždy pevné propojení na pozicích x s y, v našem schématu 1. s 2. pozicí a 3. se 4. pozicí. A proto záleží, který znak bude zrovna natočen na třetím rotoru, který znak bude spojen se svým protějškem. Z reflektoru proud teče zpět jinou cestou přes rotory, vstupní kolo, propojovací desku, kde narazí na propojenou zástrčku (7), a přes vodič (8), kde proběhne prohození znaků W, R. Následně se dostane až do diody (9) zobrazující zašifrovaný znak.

Schéma také vysvětluje, proč Enigma nemůže šifrovat znak sám na sebe, tzn. že například znak E nikdy nezašifruje na E. Stisknuté dvoucestné tlačítko nemůže rozsvítit diodu, protože jím neprochází proud směrem k diodě. Zároveň reflektor nedokáže vrátit proud stejným směrem. A pokud by to dokázal, došlo by k elektrickému zkratu.

## 2.3 Postup při použití

Při šifrování zprávy operátor napřed nastavil přístroj podle denního klíče, který získal v tajné kódové knize. Jak vypadal denní klíč, můžeme vidět v tabulce 1. V prvním sloupci najdeme označení dne, kdy má být klíč použit, další dva sloupce popisují, který reflektor a které rotory má operátor vložit do přístroje a v jakém pořadí. Další sloupce označují umístění zarážky na rotorech, výchozí nastavení rotorů a v druhé polovině tabulky nalezneme prohození jednotlivých písmen na propojovací desce.

| Den | Reflektor | Rotory | Zarážky | Výchozí pozice |
|---|---|---|---|---|
| 1 | B | I II III | D R U | G R J |
| Propojovací deska | | | | |
| AV HU IC JG KD NR OT QP SL WF | | | | |

Tabulka 1: Příklad denního kódu

Poté operátor zvolil náhodně trojici písmen, která měla tvořit klíč pro konkrétní zprávu. Tuto trojici písmen napsal dvakrát po sobě na přístroji nastaveném podle denního klíče. Pak natočil jednotlivé rotory do výchozího nastavení tak, aby v otvorech v horní desce přístroje byla vidět tři zvolená písmena, a pokračoval v šifrování pomocí zvoleného klíče pro tuto konkrétní zprávu.

## 2.4 Slabiny Enigmy

Šifrovací přístroj Enigma má několik nedostatků. Jedním z nich je již zmíněná klíčová vlastnost, že znak nikdy nemůže být zakódován sám na sebe. Když zmáčkneme znak „A“, rozsvítí se kterákoliv dioda kromě samotného znaku „A“. Tato vlastnost je způsobena použitím reflektoru.

Další nedostatek spočíval v pravidelném otáčení rotorů. V mnoha verzích přístroje Enigma dochází k pootočení druhého rotoru až poté, co se celý první rotor otočí, tj. po 26 znacích. U třetího rotoru dojde k pootočení jednou za $26 \times 26$ znaků. To dělá systém predikovatelný a jednodušší. Proto bylo u některých typů Enigmy (např. Enigma T) použito více zarážek.

U některých typů Enigmy dochází k dvojitému otočení druhého rotoru. Například pokud uvažujeme Enigmu typu M3, u druhého rotoru dochází ke dvěma otočením při dvou za sebou stisknutých klávesách, pokud se při prvním stisknutí dostane druhý rotor do polohy se zarážkou. Tím se zmenšuje síla šifry (viz tabulka 2). [4]

U Enigmy typu M4, námořní, byl přidán čtvrtý rotor na čtvrtou pozici, ale tento rotor trpěl nedostatkem, během šifrování se neotáčel. Tento rotor byl speciální, pro tuto pozici totiž byly vytvořeny dva nové typy rotorů Beta a Gamma. Proto nebylo možné jej nahradit jedním ze stávajících rotorů.

| III | II | I | komentář |
|---|---|---|---|
| A | D | O | |
| A | D | P | |
| A | D | Q | |
| A | E | R | ← 1. krok prostředního rotoru |
| B | F | S | ← 2. krok prostředního rotoru a 1. krok levého rotoru |
| B | F | T | |
| B | F | U | |

Tabulka 2: Dvojité otočení druhého rotoru (double step sequence)

Námořní Enigma měla ještě jednu slabinu. Dostala navíc další druhy standardních rotorů. Jednalo se o rotory s označením VI, VII a VIII. Tyto rotory obsahovaly dvě zarážky, které ležely naproti sobě, což způsobilo, že síla šifry byla poloviční.

Při nastavování Enigmy si můžeme vybrat tři z osmi různých možných rotorů. Matematicky vyjádřeno: variace $V(3, 8) = 336$ možností uspořádání. Dle instrukcí ale bylo nutné použít alespoň jeden rotor z námořní Enigmy. Jednalo se o rotory s označením VI, VII a VIII a vybraný rotor se nesměl znovu použít následující den na stejném místě.

Propojovací deska má 26 portů, do kterých lze zasunout vodič a tím prohodit dvě jakákoliv písmena. Pokud se do portu vodič nezasunul, písmeno mělo svou původní hodnotu. Maximálně tedy bylo možno použít 13 vodičů. V praxi se nejčastěji používalo pouze 10 vodičů, a to i přesto, že maximálního možného počtu kombinací je dosaženo při použití 11 vodičů.

Prohození znaků v páru opět dramaticky snižuje počet kombinací, kterých lze dosáhnout, oproti tzv. jednostranné propojovací desce, při jejímž použití lze dosáhnout až 26! kombinací ($26! = 403\,291\,461\,126\,605\,635\,584\,000\,000$). Pro velké množství chyb, které vznikaly při zapojování vodičů, bylo od tohoto typu propojovací desky upuštěno, ačkoliv poskytoval nejvíce možností zapojení. Rozložení jednostranné propojovací desky vypadalo následovně: padesát dva portů bylo rozděleno do čtyř řad po třinácti portech. V prvních dvou řadách byly porty označeny arabskými číslicemi a ve třetí a čtvrté řadě byly označeny znaky abecedy. Pro zprovoznění Enigmy bylo zapotřebí připojit všech 52 konektorů a všechny mít správně zapojené. Tato verze propojovací desky nesla označení Mark 2.

Každý den byl používán jiný denní klíč, pomocí kterého byly zašifrovány všechny klíče všech zpráv daného dne. I tento jeden klíč byl slabým místem při používání Enigmy, ale Němci si byli tak jisti složitostí šifry, že toto riziko podstoupili. Každá zpráva, jak již bylo zmíněno, byla zašifrována klíčem zprávy.

Tento klíč tvořily tři náhodné znaky, které se zašifrovaly právě dvakrát po sobě. Bylo to z bezpečnostních důvodů, aby bylo možné detekovat chybu při přenosu a tu později opravit. Dříve se totiž k přenosu používal telegraf. Tento opravný mechanismus dostal do Enigmy další slabinu. Němci o tomto problému věděli, ale opět spoléhali na složitost šifry. Slabina tkví v tom, že se na problém lze podívat jako na matematickou rovnici. Pokud máme kód zprávy o šesti znacích, tak při šifrování existuje šest neznámých permutací, kde první a čtvrtá permutace musí při dešifrování dávat stejný znak.

Poslední slabina Enigmy spočívá v lidském faktoru, konkrétně v lenosti operátorů. Enigma byla považována za neprolomitelnou. Síla šifry měla být natolik silná, aby v případě kompromitace knihy kódů s denními kódy, byla zpráva i přesto považována za nerozluštitelnou, protože útočník nezná kód zprávy. Operátoři se však často dopouštěli chyby, kdy jako kód zprávy zvolili kombinaci znaků ležících na klávesnici vedle sebe (např. “QWERTZ”). Nebo ještě horší podobu kódu – trojici stejných písmen. Každá zpráva měla mít svůj unikátní kód, aby nebylo možné vypozorovat opakující se vzor na začátku zpráv, ale došlo k tomu, že jeden den měly zprávy stejné kódy zpráv, protože si operátoři šifrování zjednodušovali a kódy opakovali.

## 2.5 Složitost Enigmy

Pro výpočet složitosti Enigmy použijeme Enigmu typu M3 námořní (Navy). Tento model vycházel z modelů Enigma M2 a M2a a spatřil světlo světa v letech 1939–1940. Zaměříme se tedy na verzi, kterou používaly pouze námořní síly. Od ostatních se lišila počtem rotorů, které měl operátor k dispozici, protože k původní verzi byly přidány tři nové rotory. Ostatní vojenské složky, jako armáda (Heer) a letectvo (Luftwaffe), používaly verzi Enigma I. Enigma M3 byla zpětně kompatibilní s Enigmou I, pokud operátor zvolil správné rotory. Pro matematickou složitost Enigmy je klíčových pět komponent, díky kterým se Enigma bezesporu může řadit mezi polyalfabetické substituční šifry. Jako komponenty uvažujeme tři rotory, výchozí nastavení rotorů, zarážky na rotorech, propojovací desku a reflektor. V této kapitole budeme vycházet z časopisu Cryptologia z článku A. Ray Millera o Kryptografii Enigmy [5] a z webu Crypto musea [2].

### 2.5.1 Možnosti vložení rotorů do přístroje

Rotory Enigmy typu M3 můžeme rozdělit do dvou sad. První sadou jsou rotory I–V, které pocházejí z předchozího typu Enigmy. Druhou sadu rotorů s označením VI–VIII využívaly pouze námořní síly. Enigma M3 námořní má jistá pravidla a omezení ve volbě rotorů. Jeden z rotorů musí být z kolekce VI–VIII. Tento rotor pak nesmí být použit na stejném místě ve dvou po sobě jdoucích dnech. Počítáme variace, záleží nám totiž na pořadí. Bez omezení bychom mohli spočítat složitost tímto způsobem:

$$V(3, 8) = 336.$$

Uvažujeme-li pravidla pro použití M3, vychází nám nižší hodnoty, protože dva rotory jsou z větší kolekce a jeden rotor vybíráme z menší kolekce:

$$V(2,5) \times V(1,3) = 60.$$

Rotor z kolekce námořních sil můžeme umístit na kterékoliv ze tří míst, proto musíme výsledek vynásobit třemi:

$$(V(2,5) \times V(1,3)) \times 3 = 60 \times 3 = 180.$$

Nakonec nesmíme zapomenout, že musíme odečíst $V(2,5)$, protože nemůžeme použít uspořádání z předchozího dne:

$$(V(2,5) \times V(1,3)) \times 3 - V(2,5) = 180 - 20 = 160.$$

Kvůli pravidlům používání Enigmy M3 přišli Němci zhruba o polovinu možností uložení rotorů. Stále však byla šifra silnější než v době, kdy měla Enigma pouze 5 rotorů:

$$V(3,5) = 60.$$

Na začátku kryptoanalýzy však nebylo známo zapojení rotorů, proto se na složitost můžeme podívat z jiného pohledu. Máme rotor, který má 26 vstupů a 26 výstupů, a jelikož neznáme jejich propojení, možností je:

$$26! = 403\,291\,461\,126\,605\,635\,584\,000\,000.$$

Předpokládejme, že nebyly použity dva stejné rotory, proto následující rotor bude mít $26!-1$ možností a následující $26!-2$ možností propojení. Takže výsledný výpočet by mohl vypadat:

$$26! \times (26! - 1) \times (26! - 2) = 6,56 \times 10^{79}.$$

Ve skutečnosti bylo známo, že německá vojska využívala pouze některé z možných propojení, proto $6,56 \times 10^{79}$ je pouze teoretická hodnota.

### 2.5.2 Výchozí nastavení rotorů

Každý z rotorů měl na sobě 26 znaků, tj. celou anglickou abecedu, a operátor si mohl vybrat, jak dané rotory nastaví. Odtud výpočet:

$$26^3 = 17\,576.$$

Jak je nám již známo, Enigma trpěla dvojitým otočením druhého rotoru (tzv. Double stepping sequence). [6] Kvůli této chybě je nutno opravit sílu šifry na hodnotu:

$$26 \times 25 \times 26 = 16\,900.$$

### 2.5.3 Nastavení zarážek na rotoru

Každý ze tří rotorů obsahoval zarážku, která mohla být umístěna kdekoliv. Zarážka měla za úkol otočit dalším rotorem při jejím průchodu počátkem. Rotor, který se nacházel nejvíce vlevo, neměl již čím otočit, proto byla jeho zarážka bezvýznamná, reflektor měl pevnou pozici. Po umístění dvou zarážek dostáváme:

$$26^2 = 676.$$

Námořní Enigma M3, jak jsme se již dozvěděli, měla dvě sady rotorů. První sada rotorů, která byla kompatibilní s Enigmou I, měla pouze jednu zarážku. Druhá sada měla právě dvě zarážky, které byly umístěny naproti sobě. Mezi každou zarážkou v druhé sadě bylo třináct znaků. Kryptografickou sílu dostaneme studiem případů. Rotor umístěný vlevo neuvažujeme pro jeho bezvýznamnost v tomto případě.

Uvažujeme tři možnosti, kde může být umístěn rotor z druhé sady (nový rotor). Ve třetím případě je nový rotor úplně vlevo.

1. nový rotor a původní rotor $\rightarrow 13 \times 26 +$
2. původní rotor a nový rotor $\rightarrow 26 \times 13 +$
3. původní a původní rotor $\rightarrow 26 \times 26 +$

celkem 1352.

Nyní je nutné výsledek vydělit počtem možností. Předpokládejme, že pravděpodobnost vložení nového typu rotoru je pro všechny 3 pozice stejná.

$$1352 \div 3 \approx 451.$$

### 2.5.4 Složitost propojovací desky

Propojovací deska procházela vývojem jako samotná Enigma. Nejprve existovaly návrhy, kde byly znaky uspořádány do dvou kruhů po třinácti znacích. Propojení bylo zajištěno napevno připevněnými třinácti kabely v každém kruhu. Propojit a tím prohodit znaky šlo pouze v rámci těchto dvou skupin. Tím byla zajištěna složitost:

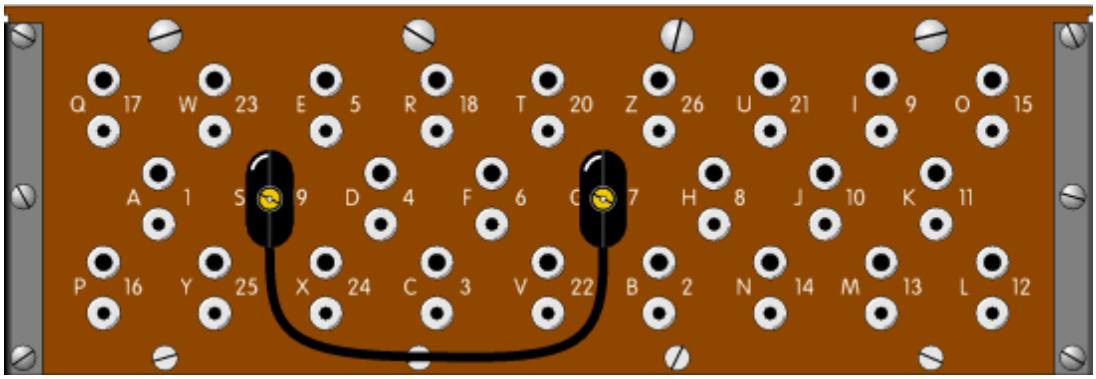
$$13! \times 13! = 6\,227\,020\,800 \times 6\,227\,020\,800 = 3,88 \times 10^{19}.$$

Verze s označením Mark 2, o které jsme se zmiňovali v kapitole o slabinách Enigmy, byla kryptograficky nejsilnější ze všech možných návrhů. Bylo však zapotřebí propojit všech 26 propojovacích kabelů. Její složitost byla:

$$26! = 4,03 \times 10^{26}.$$

My se však podrobněji zaměříme na propojovací desku s označením Mark 3, která je součástí Enigmy M3. Uspořádání portů znaků je QWERTZ a s obdobným rozložením, jaké dnes známe z české klávesnice. U každého znaku je arabská

číslovka, jež značí pořadí v abecedě. Každý port má 2 piny, to proto, aby když dojde k zasunutí propojovacího kabelu, mohly být znaky jednoduše substituovány jeden druhým. Pro lepší představu, jak propojovací deska funguje, nám poslouží schéma na obrázku 2.




Obrázek 2: Schéma propojovací desky Enigmy M3 [7]

Přístroj se zmíněnou propojovací deskou obsahoval pouze dvanáct propojovacích kabelů a bylo možné zaměnit znaky pouze v párech. Odtud vzorec pro výpočet složitosti:

$$\frac{26!}{(26-2n)! \times n! \times 2^n},$$

kde $n$ je počet propojovacích kabelů.

V tabulce 3 lze vidět řádově narůstající množství kombinací od prvního do jedenáctého vodiče. Připomeňme si, že Němci používali pouze deset vodičů a samotný přístroj obsahoval jen dvanáct vodičů, tudíž celková hodnota je jenom teoretická.

Matematicky zapsáno:

$$\sum_{n=0}^{n=13} \frac{26!}{(26-2n)! \times n! \times 2^n} = 532\,985\,208\,200\,576.$$

### 2.5.5 Reflektor

Reflektor je komponenta, která je velmi podobná rotoru s tím rozdílem, že kontakty jsou pouze na jedné straně a 26 znaků (pinů) je interně propojeno do 13 párů. Když bychom opět uvažovali, že neznáme vnitřní propojení, dostaneme:

$$25!! = 7\,905\,853\,580\,550,\,[1]$$

[1]Dvojitý faktoriál jsme zavedli podle [9] takto:

$$n!! \equiv \begin{cases} n \cdot (n-2) \ldots 5 \cdot 3 \cdot 1 & n > 0 \, \textit{liché} \\ n \cdot (n-2) \ldots 6 \cdot 4 \cdot 2 & n > 0 \, \textit{sudé} \\ 1 & n = -1,\, 0. \end{cases}$$

| $n$ | počet kombinací |
|---|---|
| 0 | 1 |
| 1 | 325 |
| 2 | 44 850 |
| 3 | 3 453 450 |
| 4 | 164 038 875 |
| 5 | 5 019 589 575 |
| 6 | 100 391 791 500 |
| 7 | 1 305 093 289 500 |
| 8 | 10 767 019 638 375 |
| 9 | 53 835 098 191 875 |
| 10 | 150 738 274 937 250 |
| 11 | 205 552 193 096 250 |
| 12 | 102 776 096 548 125 |
| 13 | 7 905 853 580 625 |
| Celkem | 532 985 208 200 576 |

Tabulka 3: Složitost propojovací desky [8]

stejně jako když na propojovací desce použijeme 13 vodičů:

$$\frac{26!}{(26-2\times 13)!\times 13!\times 2^{13}} = 7\,905\,853\,580\,550.$$

K výsledku jsme dospěli následovně: jeden znak chceme propojit s jiným, a proto vybíráme mezi dalšími 25 znaky. Druhý znak chceme opět propojit, ale už nám zůstává jen 23 znaků na výběr a tak dále. V lednu roku 1944 vznikl nový typ reflektoru UKW-D, který nahrazoval reflektory B a C. U tohoto reflektoru bylo možné nastavit vnitřní propojení jednotlivých znaků a tím zvýšit sílu šifry. Operátory nebyl oblíben pro své složité nastavování. V Bletchey Park v Anglii, kde bylo hlavní dešifrovací centrum, rotoru UKW-D říkali „Uncle Dick“.

Standardní reflektory byly označeny UKW-A, UKW-B a UKW-C. Reflektor s označením UKW-A se používal před druhou světovou válkou v Enigmě I. Začátkem války byl použit reflektor UKW-B a později byl přidán reflektor UKW-C.

Pro naše účely je důležitý fakt, že do Enigmy M3 je možné vložit jeden ze dvou reflektorů.

### 2.5.6 Celková složitost

V předchozí kapitole jsme si vysvětlili, jakým způsobem byla zaručena složitost Enigmy a jaké komponenty se na tom podílely. Zjistili jsme, že teoretická složitost je mnohonásobně vyšší než reálná. V této kapitole se na rozdíl mezi nimi podíváme podrobněji.

**Teoretickou složitost** rozepsanou po jednotlivých komponentech nalezneme v tabulce 4.

| Komponenta | Složitost |
|---|---:|
| Rotory | $26!^3$ |
| Výchozí nastavení rotorů | $26^3$ |
| Nastavení zarážek | $26^2$ |
| Propojovací deska | $26!$ |
| Reflektor | $25!!$ |

Tabulka 4: Teoretická složitost Enigmy

$26!^3 \times 26^3 \times 26^2 \times 26! \times 25!! = 2,48 \times 10^{126} \approx 2^{420}$

**Reálnou složitost** šifry a komponenty, ze kterých je složena nalezneme v tabulce 5.

| Komponenta | Složitost |
|---|---:|
| Rotory | 160 |
| Výchozí nastavení rotorů | $26^2 \times 25$ |
| Nastavení zarážek | 451 |
| Propojovací deska | $\frac{26!}{10! \times 6! \times 2^{10}}$ |
| Reflektor | 2 |

Tabulka 5: Reálná složitost Enigmy

$160 \times 26^2 \times 25 \times 451 \times \frac{26!}{10! \times 6! \times 2^{10}} \times 2 = 3,68 \times 10^{23} \approx 2^{79}$

Reálná kryptografická síla Enigmy je $2^{341}$*krát* menší než její teoretický model.

Jednou z největších slabin, na kterou jsme při psaní této práce narazili a která měla podíl na odhalení principu šifry, byla lidská chyba. Operátoři, kteří Enigmu

obsluhovali, netušili, co mohou způsobit neopatrným zacházením s ní a následným vysíláním pomocí telegrafu. Další lidskou chybou byl únik materiálů, ať už se jedná o knihu denních kódů, jednotlivé rotory či schémata popisující princip šifrování. Tyto uniklé materiály pomohly polským kryptoanalytikům a později Britům neřešit dešifrování hrubou silou, ale matematicky a na základě zkoumání principů fungování přístroje.

## 2.6 Vybrané verze Enigmy

V letech 1923–1941 bylo využíváno přes 24 druhů Enigmy. Lišily se nejen počtem rotorů, ale také speciálními znaky národních abeced. Například Enigma typu B' měla 28 znaků a mezi speciální znaky patřily å, ä, ö. Dále byly odstraňovány slabiny předchozích verzí, přidávaly se další rotory, zdokonalovala se propojovací deska apod.

### 2.6.1 Ruská Enigma

Takto bývá někdy označována ruská Fialka M-125. Jedná se o elektromechanický šifrovací přístroj, který byl vyvinut v Sovětském svazu krátce po 2. světové válce. Poprvé byl představen v roce 1956 a brzy se stal oblíbeným v zemích, které podepsaly Varšavskou smlouvu. Ruská Fialka neměla tolik slabin jako Enigma. Používala deset rotorů oproti Enigmě, která využívala tři nebo čtyři. Rotory se otáčely častěji, měly více zarážek. Sousední rotory se otáčely do protisměru, propojení a nastavení rotoru mohlo být od roku 1978 měněno. Místo propojovací desky byl použit děrný štítek. Jednou z velkých změn byla možnost zakódovat písmeno samo na sebe, což v Enigmě nebylo možné. Ve verzi M-125-3 bylo možno zapisovat také čísla a interpunkční znaménka. Fialkou bylo možné zakódovat jak cyrilici, tak latinku.

### 2.6.2 Hagelin B-21

Boris Hagelin, švédský inženýr a zakladatel společnosti A. B. Cryptoteknik, již v roce 1925 vytvořil elektromechanický šifrovací přístroj. Zvnějšku se B-21 podobal Enigmě, uvnitř nevyužil substituci znaků, ale postavil stroj na programovatelných rotorech, které ovládaly nepravidelné otáčení šifrovacích rotorů. Signál $+/-$ se dále kódoval každý jinou programovatelnou maticí $m \times n$. Na konci se signál spojil a rozsvítil příslušnou diodu. Pro dešifrování bylo nutné přístroj rozebrat, později byl dodán přepínač. Časem dostala inovovaná verze B-211 tiskárnu a oblíbila si ji francouzská armáda.

# 3 Praktická část

## 3.1 Programátorská příručka

V rámci praktické části bakalářské práce jsme naprogramovali aplikaci, která pracuje na podobných principech jako Enigma, ale netrpí některými slabinami jako přístroj, který byl používán Němci za války. Pro tvorbu aplikace jsme použili programovací jazyk C# a využili .NET frameworku, kde jsme použili grafickou knihovnu Windows Forms Application. Při psaní kódu jsme využívali oficiální dokumentaci Microsoftu pro .NET framework dostupnou online.[2]

V kódu využíváme jmenných prostředí .NETu: System.Collections.Generic, System.ComponentModel, System.Data, System.Drawing, System.Linq, System.Text, System.Threading.Tasks, System.Windows.Forms a System.IO, kde využíváme vybraných tříd, metod a funkcionalit pro čtení a zápis do souboru.

Program jsme vložili do jmenného prostředí (namespace) Enigma a dále jsme jej rozčlenili na 6 tříd.

První a hlavní třídou je *Program*. Třída nastaví renderování textu, povolí vizuální styly potřebné pro Windows Forms a spustí aplikaci, instanci třídy Eni.

Následuje třída nazývající se *Eni*, která je hlavním oknem aplikace. Třída je parciální a to z toho důvodu, abychom oddělili grafickou část okna od části, kde se zpracovávají obsluhy událostí. Události jsou vyvolány při stisku kláves, při psaní jednotlivých znaků, které budou zašifrovány, výběru rotorů, editaci nastavení rotorů a zarážky a výběru reflektoru. Třída je tvořena zejména prvky typu textového pole. V těchto prvcích jde vidět pozice rotorů, nastavení zarážky na jednotlivých rotorech, zvolené rotory a těmi nejdůležitějšími okny jsou vstup (input) a výstup (output). Reflektor si uživatel volí jako jednu z položek objektu rozbalovacího seznamu. Tlačítko Setup spustí další okno, což je instance třídy Settings. Důležitou funkcionalitou je tlačítko Random, které náhodně nastaví reflektor, rotory a všechny atributy rotorů.

Další třídu jsme nazvali *Settings*. Třída je taktéž parciální, protože obsahuje okno, kde si uživatel může nastavit propojovací desku (plugboard). Uživatel mění propojení jednotlivých znaků a zároveň vidí současné nastavení ve dvou objektech třídy listBox. Pro změnu výchozího propojení znaků je možnost znaky přesunout pomocí tlačítek. Při jejich stisku je událost příslušnou metodou zpracována. Znaky jsou položky jednotlivých seznamů. Prohození znaků je vyřešeno pomocí změny indexů. V prvním seznamu jsou znaky uspořádány abecedně a každý z nich má svůj pevný index, v druhém seznamu je pak substituovaný znak, který lze nastavit. I do této třídy jsme zakomponovali funkcionalitu náhodného nastavení pro vyšší komfort uživatele.

Čtvrtou třídou je *Rotor*. Třídu jsme nazvali Rotor, protože v ní probíhá samotné šifrování pomocí rotorů. Pro každý rotor jsme vytvořili strukturu Rotors. Struktura Rotors se skládá z pole znaků, kterými je každý rotor charakteristický, dále pak z dalších dvou znaků, a to z výchozího nastavení, tzv. ground value nebo

[2] https://docs.microsoft.com/en-us/dotnet/

settings, a z písmene, kde je umístěna zarážka, díky které dojde k otočení dalšího rotoru. Pro všechny znaky jsme použili datový typ char, respektive pole charů u pole znaků.

Poslední dvě třídy se nazývají *About* a *HowToUse* a jsou opět okny aplikace, které slouží jako komentář a nápověda k aplikaci.

### 3.1.1 Proměnné třídy Rotor

- rotor
  - pole o velikosti 8
  - datovým typem je struktura Rotors
  - pole využito pro uložení jednotlivých možných rotorů
- alphabet
  - proměnná struktury Rotors
  - slouží k uložení abecedy
  - využili jsme datový typ Rotors, aby byl zajištěn jednotný přístup ke znakům rotoru a ke znakům abecedy
- reflectors
  - dvourozměrné pole o velikosti $8 \times 26$
  - pole datového typu integer
  - slouží k uložení všech reflektorů a jejich znaků
- chosRefle
  - proměnná datového typu integer
  - slouží pro uložení vybraného reflektoru
  - název vznikl jako zkratka chosen reflector
- chosRot
  - pole o velikosti závisející na počtu rotorů
  - datový typ integer
  - slouží pro uložení vybraných rotorů k šifrování
  - hodnoty jsou v poli uložené pozpátku, tak jako Enigma šifruje zprava doleva
  - název proměnné odvozen od chosen rotors

### 3.1.2 Metody třídy Rotor

- InitializeRotors

  Metoda je vyvolána při každém spuštění aplikace a slouží k definici proměnných rotor, alphabet a reflectors.

- getNumberOfRotors

  Metoda vrací počet aktuálně použitých rotorů. V aplikaci je možné použít 2–5 rotorů. S počtem rotorů se pracuje například při šifrování.

- shiftLetter

  Metoda shiftLetter má za úkol posunout vybraný rotor o jednu pozici dopředu. Pokud je rotor na posledním znaku, vrátí jej na začátek.

- Encode

  Klíčová metoda Encode slouží k šifrování znaků. Na vstupu přijme znak k zašifrování, pak otočí prvním rotorem a kontroluje, jestli je potřeba otočit i se zbývajícími rotory. Znak poté projde všemi vybranými rotory, reflektorem a zpět vybranými rotory. Vrácen je zašifrovaný znak. Metoda je vyvolána z metody textBoxInputKeyUp, respektive při stisknutí klávesy ve vstupním boxu.

### 3.1.3 Metody třídy Eni

- EniFormLoad

  Metoda vyvolána při načtení okna Eni. V metodě dochází k definici jednotlivých položek reflektorů, ze kterých bude uživatel vybírat. Dále jsou zde aktualizována textová pole s nastavením rotorů a nastaven výchozí reflektor.

- refreshRotors

  Metoda refreshRotors slouží k načtení vybraných rotorů z proměnné chosRot a zobrazení v okně aplikace.

- refreshRings

  Pro zobrazení zarážek jednotlivých rotorů je zapotřebí je načíst z vybraných rotorů, k tomu slouží metoda refreshRings.

- refreshGroundValues

  Při každém zašifrování znaku se musí rotory otočit. Pro načtení jejich interní reprezentace slouží metoda refreshGroundValues.

- textBoxInputKeyUp

  Metoda, jež reaguje na událost při stisknutí znaku na vstupu. V metodě je nejprve ověřeno, zda je se jedná o validní znak abecedy. Dále je odeslán

do propojovací desky, aby bylo zjištěno, zda má být znak substituován za jiný a případně nahrazen. Následuje hlavní zašifrování pomocí rotorů a nakonec opět substituce pomocí propojovací desky. Výsledek je zobrazen v okně s výstupem (output).

- ostatní metody reagující na události

  Další metody mají podobnou funkcionalitu. Po stisku klávesy ověří zadaný vstup a pokud je validní, modifikují příslušnou interní proměnnou, kterou pak nechají překreslit v aplikaci. Takto je zpracován výběr rotorů, jejich výchozí nastavení, zarážky a výběr reflektoru.

### 3.1.4 Proces šifrování

Ve chvíli, kdy je zapsán znak do vstupního pole, je tímto polem vyvolána událost, na kterou reaguje metoda InputBoxKeyUp. Metoda nejprve ověří, zda se je jedná o validní vstup, tzn. jestli se jedná o písmeno anglické abecedy A–Z. Dále je znak předán jako argument instanci třídy Settings metodě Encode. Tím simulujeme činnost propojovací desky. Výsledkem je argument pro další metodu, která se také nazývá Encode a je instancí třídy Rotor. Tento krok rozebereme podrobněji.

Metoda Encode dostane na vstup znak, který má zakódovat podle aktuálního nastavení. Nejdřív však musí otočit prvním rotorem a vyhodnotit, zda je potřeba otočit také dalšími rotory. V tomto bodě se nachází podmínka, která dělí otáčení rotorů na dvě větve. Na větev, kde je aplikováno dvojité otočení druhého rotoru (Double Step Sequence) a na větev bez této funkcionality. Následně se větve opět spojí. Poté dochází k samotnému šifrování. Nejprve zjistíme pozici vstupního znaku v abecedě. Totéž provedeme pro aktuální znak z nastavení konkrétního rotoru (Ground Settings). Tyto dva indexy sečteme, zkontrolujeme, jestli není součet větší než počet znaků v anglické abecedě, a pokud ano, aplikujeme modulo 26. Načteme znak z aktuálního rotoru pomocí indexu, který jsme si právě spočítali. A u tohoto nového znaku si opět spočítáme pozici v abecedě. Tento postup provedeme pro každý rotor, poté načteme odpovídající hodnotu z reflektoru a opakujeme předchozí postup v obráceném pořadí.

Samotné šifrování je plné indexů a polí, se kterými pracujeme po celou dobu a bez kterých by aplikace Enigma nemohla fungovat. Významnou roli zde také hraje funkce modulo. V aplikaci jsme se zaměřili na možné následné rozšíření a to zejména v počtech rotorů. Jako první jsme zmínili proměnnou chosRot a metodu getNumberOfRotors, ze kterých je patrné, že počet rotorů je proměnlivý. V aplikaci využíváme 2–5 rotorů, ale je připravena na libovolný požadovaný počet rotorů.

V následující ukázce kódu 1 metody Encode třídy Rotor prezentujeme algoritmus zašifrování jednoho znaku na jednom rotoru.

Pro lepší přehlednost a čitelnost jsme zakódování rozdělili do čtyř řádků. V prvním řádku spočítáme index výchozího nastavení daného rotoru, dále přičteme index znaku ze vstupu a v případě, že se jedná o dva znaky z druhé půlky abecedy, se aplikuje funkce modulo. Díky tomuto indexu načteme substituovaný

```
1 indexGroundLetter = (int)rotor[chosRot[i]].ground - 65;
2 indexInputAndGround = (pos + indexGroundLetter) % 26;
3 indexEncodedLetter =
4   (int)rotor[chosRot[i]].letters[indexInputAndGround] - 65;
5 pos = (indexEncodedLetter + 26 - indexGroundLetter) % 26;
```

Zdrojový kód 1: Ukázka algoritmu zašifrování znaku skrze rotor

znak z pole daného rotoru. Nakonec musíme odečíst index výchozího nastavení rotoru.

Jakmile je znak zakódován pomocí rotorů a reflektoru, projde opět propojovací deskou, tj. metodou Decode. Dále se pošle znak do výstupního pole a aktualizují se hodnoty rotorů.

## 3.2 Odstraněné slabiny z Enigmy

Naším cílem bylo naprogramovat aplikaci, která bude fungovat na podobných principech jako Enigma a kterou bude v praxi možno využívat s tím rozdílem, že budou odstraněny nebo vylepšeny slabiny přístroje. Nejprve popíšeme slabiny, které se nám podařilo odstranit a poté v další kapitole spočítáme kryptografickou sílu aplikace.

První nedostatek, nemožnost zašifrování znaku sám na sebe, jsme odstranili přidáním upraveného reflektoru B, kde jsme prohodili znaky B a R.

Dvojité otáčení druhého rotoru jako charakteristický jev Enigmy, který snižuje sílu, jsme si dovolili ve výchozím nastavení ponechat. Dali jsme však operátorovi možnost tuto vlastnost deaktivovat zrušením zaškrtnutí položky dvojitého otáčení (Double Step Sequence).

Během programování jsme vycházeli z konstrukce Enigmy M3, která má 3 rotory. Do naší aplikace jsme přidali další dva rotory. Domníváme se, že pět rotorů poskytuje dostatečnou sílu při šifrování zpráv do deseti milionu znaků v rámci jednoho dne, a tudíž tento počet není omezující. Šifrovací algoritmus je však připraven pracovat s jakýmkoliv počtem rotorů.

Na rozdíl od námořní Enigmy M4 jsme u rotorů VI, VII a VIII použili jednu zarážku a zrušili jsme pravidla pro jejich použití. Je tedy možné je použít na jakémkoliv místě a v jakémkoliv počtu.

Maximální využití propojovací desky limitovalo pravidlo používání deseti vodičů a jejich propojení v páru. V naší aplikaci se nám podařilo oba nedostatky odstranit.

Jednou z posledních slabin používání Enigmy byla lenost operátorů vymýšlet kódy zpráv. Tato slabina se sice netýká samotného přístroje, ale jednoduchou pomůckou můžeme tento problém eliminovat. V aplikaci je možnost nechat nastavení náhodně vygenerovat, čímž se práce operátora zjednoduší a nebude docházet ke generování stejných či obdobných nastavení přístroje v rámci jednoho dne.

## 3.3 Matematická složitost aplikace

Stejně jako v teoretické části projdeme všechny klíčové komponenty pro kryptografickou složitost aplikace, zapíšeme jejich hodnoty do tabulky 6 a poté spočítáme složitost všech komponent.

| Komponenta | Složitost |
|---|---|
| Rotory | V(5,8) = 6720 |
| Výchozí nastavení rotorů | $26^5$ |
| Nastavení zarážek | $26^4$ |
| Propojovací deska | 26! |
| Reflektor | 8 |

Tabulka 6: Kryptografická složitost aplikace

$26^5 \times 26^4 \times 26! \times 6720 \times 8 = 1,18 \times 10^{44} \approx 2^{147}$

V porovnání s přístrojem Enigma jsme dosáhli $2^{68}$*krát* silnější šifry. Domníváme se tedy, že se nám v tomto ohledu podařilo zadání práce splnit.

## 3.4 Nastavení aplikace

Pro jednodušší použití v praxi existuje v aplikaci možnost nastavení načíst ze souboru nebo stávající nastavení do souboru uložit. K tomuto účelu slouží položka načíst nastavení Enigma (Load Enigma settings) v nabídce soubor (File). Obdobně pro uložení nastavení. Při ukládání nastavení se uživateli otevře okno s možností výběru adresáře a názvu souboru. Ve výchozím nastavení se otevře uživatelův adresář s dokumenty a soubor je uložen pod názvem nastavení (Settings) s příponou txt.

Ve struktuře souboru lze přečíst interní reprezentaci nastavení. V prvním řádku se nachází ID reflektoru. Na dalším řádku jsou uloženy zvolené rotory v pořadí zprava doleva. Na následujících dvou řádcích můžeme nalézt nastavení zarážek a výchozího nastavení opět v pořadí zprava doleva. Na posledním řádku najdeme nastavení propojovací desky. Znaky nejsou zapsány ve dvojicích tak, jak to známe z kódové knihy. Zapisujeme je podle abecedního pořadí jejich protějšků. Pořadí zapisovaných atributů jsme převzali z kódové knihy Enigmy.

Při načítání nastavení ze souboru ověřujeme, zda se jedná o validní nastavení. V prvním kroku ověřujeme počet argumentů a dále procházíme veškeré argumenty, jestli jsou platnými hodnotami aplikace. Pokud není hodnota v platném rozsahu, je vypsána chybová hláška a další načítání nastavení ukončeno.

## 3.5 Použití v praxi

Pro bezchybné fungování aplikace ji bylo nutné vytvořit tak, aby byla rezistentní vůči uživatelským chybám, nevalidním vstupům a aby byla intuitivní. Dále jsme aplikaci doplnili o bublinové nápovědy, které poslouží zejména při prvním použití. Pro komfortnější práci jsme implementovali import a export nastavení a dále pak export zašifrovaného textu do souboru. Při vývoji pro nás bylo klíčové zesílení šifry mechanické Enigmy odstraněním jejích slabin. Tím je zaručena vyšší bezpečnost.

Jediným dnes limitujícím faktorem je přítomnost počítače pro použití v terénu. Aplikace by se však dala přepracovat tak, aby se dala použít například jako webová aplikace a byla dostupná online v telefonu nebo tabletu.

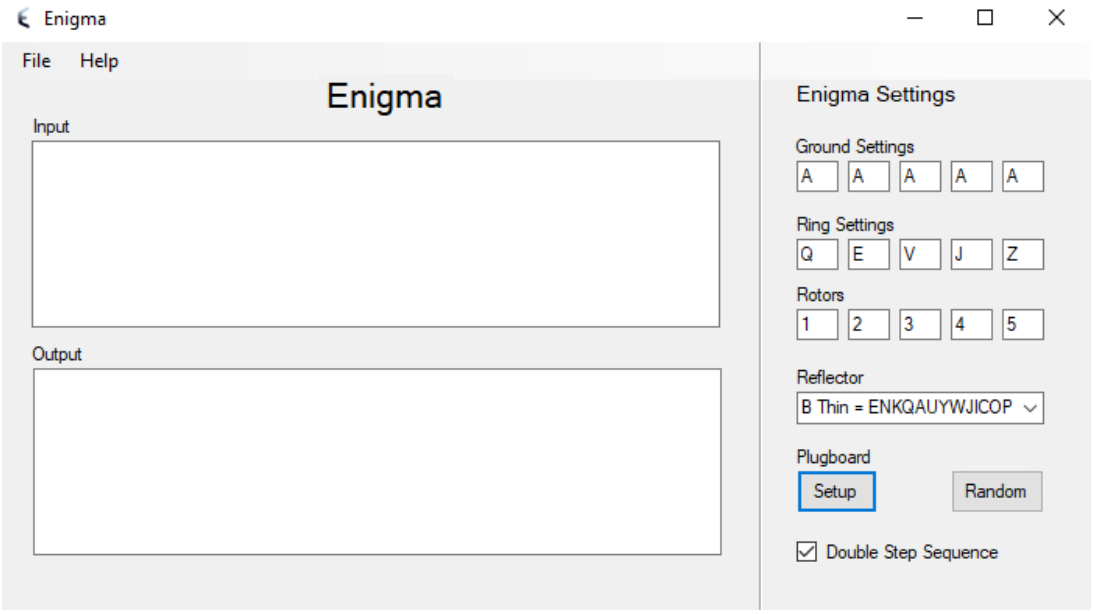
# 4 Uživatelská příručka

## 4.1 Požadavky aplikace

Software je naprogramován, kompilován a optimalizován pro platformu .NET 4.6.1. Pro spuštění je tedy zapotřebí mít nainstalován .NET framework 4.6.1 a vyšší. Program není potřeba instalovat. Software nemá speciální hardwarové požadavky, pro spuštění stačí běžný kancelářský či domácí počítač.

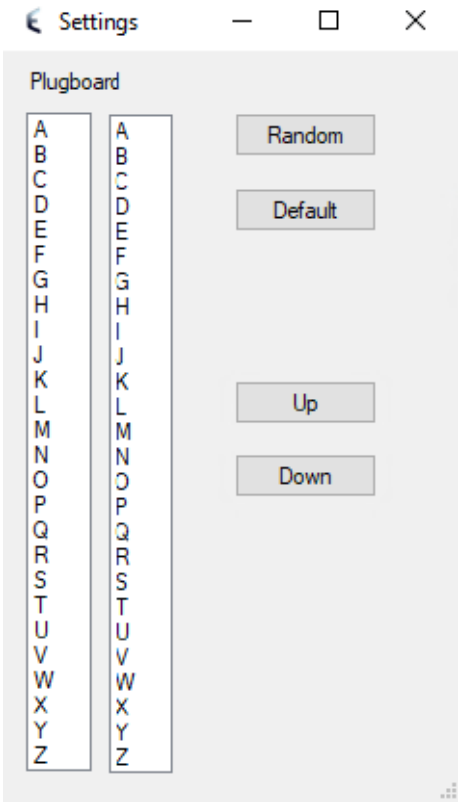
## 4.2 Grafika

Po spuštění aplikace Tom's Enigma.exe se vám otevře okno s programem Enigma. Okno bychom mohli rozdělit do dvou částí. V levé části můžete najít textové pole pro vstup – otevřený text a pod ním další textové pole pro výstup – šifrovaný text. V pravé části lze nalézt nastavení aplikace. V prvních pěti polích můžete nastavit výchozí stav rotorů, v dalším řádku nadefinujete, kde bude na rotoru umístěna zarážka, a ve třetím řádku zvolíte, jaký rotor bude použit. V neposlední řadě je zde rozbalovací nabídka, kde můžete zvolit reflektor.




Obrázek 3: Ukázka hlavního okna aplikace

Tlačítko Setup slouží pro nastavení propojovací desky. Po kliknutí na něj se otevře nové okno s možností jejího nastavení. Konfigurace propojovací desky je skrytá, tak jako je na první pohled skrytá na zařízení Enigma. V okně propojovací desky (Settings) v prvním sloupci je neměnná abeceda, v druhém sloupci je 26 znaků, které lze přeskupit a tím jednotlivé znaky zaměnit. Pro posun znaků v druhém sloupci slouží tlačítka Up/Down. Tlačítko Default ruší všechny substituce a navrací znakům jejich původní hodnotu. Tlačítko Random náhodně přeskupí všechny znaky.




Obrázek 4: Ukázka okna s propojovací deskou

## 4.3 Fungování aplikace

Po spuštění aplikace je zapotřebí si ji nastavit. Samotné šifrování funguje i s výchozím nastavením, ale bylo by pro útočníka velmi snadné brzy dešifrovat text po tomto základním nastavení přístroje. Případně je zde možnost si nechat náhodně zvolit nastavení.

### 4.3.1 Rotor

Jako první doporučujeme nastavit si rotory. Rotory volíme zprava doleva, tak jako tomu je u opravdové Enigmy. První rotor leží nejvíce vpravo. Máte možnost si zvolit z osmi základních rotorů, jejichž rozložení můžeme nalézt v tabulce 7.

Číslo rotoru zapisujte arabskými číslicemi. Minimální počet rotorů pro šifrování jsou dva, toho můžete docílit tím, že do prostředního, třetího rotoru, napíšete 0. Tím se doplní nuly i do zbývajících rotorů nalevo a jsou využívány jenom dva rotory. Pokud zvolíte neexistující rotor, zobrazí se Vám dialogové okno s chybovou hláškou, že je potřeba zvolit jeden z rotorů I–VIII.

| Rotor | Propojení |
|---|---|
| I | EKMFLGDQVZNTOWYHXUSPAIBRCJ |
| II | AJDKSIRUXBLHWTMCQGZNPYFVOE |
| III | BDFHJLCPRTXVZNYEIWGAKMUSQO |
| IV | ESOVPZJAYQUIRHXLNFTGKDCMWB |
| V | VZBRGITYUPSDNHLXAWMJQOFECK |
| VI | JPGVOUMFYQBENHZRDKASXLICTW |
| VII | NZJHGRCXMYSWBOUFAIVLPEKQDT |
| VIII | FKQHTLXOCBJSPDZRAMEWNIUYGV |

Tabulka 7: Vnitřní propojení rotorů

### 4.3.2 Zarážka

Každému rotoru můžete nastavit, v jaké poloze bude umístěna zarážka. Zarážku lze nastavit i pro pátý rotor, který leží nejvíce vlevo, tam však postrádá smysl, protože reflektor má pevnou pozici a neotáčí se, zarážka tedy nemá čím otáčet. Jako zarážku můžete zvolit jakékoliv písmeno anglické abecedy.

### 4.3.3 Základní nastavení

Nakonec je potřeba rotorům zvolit výchozí polohu, od které se budou dále otáčet, a jak budou zašifrovány první znaky. Polohu určete vložením písmena anglické abecedy.

### 4.3.4 Reflektor

Ve výchozím nastavení je používán reflektor B. Ten můžete vyměnit za jeden z osmi reflektorů z tabulky 8. Reflektory substituují znaky z abecedy za znak z reflektoru, který leží na stejné pozici. Pro využití funkcionality šifrování znaku na sebe sama vyberte reflektor s označením Ba.

### 4.3.5 Propojovací deska

Záměnu znaků pomocí propojovací desky provedete kliknutím na tlačítko Setup v hlavním okně. Po otevření nového okna vyberte znak z druhého sloupce a pomocí tlačítek Up/Down jej umístěte na vybranou pozici. Následně bude znak z prvního sloupce $x - té$ pozice substituovaný za znak z druhého sloupce $x - té$ pozice. Pro náhodnou substituci všech znaků stiskněte tlačítko Random, naopak pro navrácení znakům jejich původní hodnoty stiskněte tlačítko Default.

| Rotor | Propojení |
|---|---|
| Beta | LEYJVCNIXWPBQMDRTAKZGFUHOS |
| Gamma | LEYJVCNIXWPBQMDRTAKZGFUHOS |
| A | EJMZALYXVBWFCRQUONTSPIKHGD |
| B | YRUHQSLDPXNGOKMIEBFZCWVJAT |
| C | FVPJIAOYEDRZXWGCTKUQSBNMHL |
| B Thin | ENKQAUYWJICOPBLMDXZVFTHRGS |
| C Thin | RDOBJNTKVEHMLFCWZAXGYIPSUQ |
| Ba | YBUHQSLDPXNGOKMIERFZCWVJAT |

Tabulka 8: Vnitřní propojení reflektorů

Změny jsou ukládány ihned a automaticky. Po ukončení nastavení se vraťte zpět do původního okna zavřením okna s nastavením propojovací desky.

### 4.3.6 Stav

Průběžný stav se mění a lze sledovat. Při každém stisku znaku ve vstupním poli se mění výchozí nastavení prvního rotoru, případně dalších rotorů. Tuto změnu můžete sledovat v okénkách vpravo v části Ground Settings.

### 4.3.7 Praktické tipy

- Během psaní textu můžete průběžně měnit nastavení a tím dosáhnout silnější šifry. Domluvte se s příjemcem, že každé slovo či věta budou mít jiné nastavení přístroje.
- Text ve vstupním poli můžete mazat, nemá vliv na změnu nastavení.
- Při překlepu se nelze vrátit o krok zpět. Příjemce si musí s překlepem poradit, nebo si přenastavte nastavení o krok zpět a smažte jeden znak. Pozor na přítomnost zarážky, která otočí další rotory.
- Využijte import a export nastavení přístroje do souboru. Zjednoduší to zdlouhavé nastavování.

# Závěr

Naším cílem bylo popsat Enigmu jako šifrovací nástroj z pohledu informatiky, popsat její fungování, kryptografickou sílu symetrické substituční šifry a nalézt slabé stránky přístroje. V praktické části jsme si kladli za cíl odstranit slabiny Enigmy, případně je vylepšit, a poté získané poznatky implementovat do aplikace, která bude fungovat na stejných principech jako Enigma.

V teoretické části jsme popsali mechanickou stránku přístroje a jednotlivé fáze substituce znaku. Odhalili jsme slabiny Enigmy, spočítali jsme matematickou složitost jednotlivých komponent a přístroje jako celku. Nakonec jsme se zmínili o alternativách tohoto zařízení.

V praktické části práce jsme se zaměřili na odstranění chyb v Enigmě, zvyšování síly šifry a na následné naprogramování aplikace, ve které jsme využili teoretického modelu silnější šifry. Jednou z hlavních slabin Enigmy byla nemožnost šifrovat písmena sama na sebe. Tento problém se nám podařilo odstranit přidáním speciálního rotoru, který toto šifrování umožňuje. Omezení praktického a mechanického rázu, která měla důsledek na sílu šifry, se nám podařila taktéž vyřešit. Po odstranění slabých stránek původní šifry, která byla použita za druhé světové války, jsme tak dostali šifru, která má dvojnásobně silnou kryptografickou sílu.

Další kapitola praktické části obsahuje programátorskou příručku. V ní nalezneme ukázku kódu, popis jednotlivých tříd a informace o tom, jak jsou jednotlivé substituce implementovány.

V poslední kapitole přibližujeme aplikaci koncovému uživateli v rámci uživatelské příručky. Uvedli jsme v ní softwarové požadavky aplikace, popsali jsme její grafické prostředí, vysvětlili základy práce s touto aplikací a nakonec jsme přidali praktické tipy pro její užívání.

Domníváme se, že jsme cíle stanovené v úvodu práce splnili. Při studiu tohoto tématu nás nejvíce překvapovala a zarážela bezmezná důvěra německých hodnostářů i kryptoanalytiků v neprolomitelnost šifry, ačkoliv už tehdy znali její slabé stránky.

# Conclusions

Our research goal was to describe the Enigma machine as a cipher instrument from the perspective of informatics, to describe its working principles, the key space of symmetric substitution cipher and find out weak points of the machine. The practical part was focused on finding and possible improving of Enigma's weaknesses with the following implementation of found solutions in the application, which works on the same principles as the Enigma machine does.

In the theoretical part we describe the mechanical side of the machine and different phases of letter substitution. We discover weaknesses of the Enigma, we calculate key space of each components and whole Enigma. Finally, we mention alternatives to the machine.

In the practical part of the current paper we focus on removing of Enigma's mistakes, namely, increasing a key space of the cipher and programming an application, in which a theoretical model of stronger cipher is used. One of Enigma's weakest points is an impossibility to encode a letter to itself. We have removed this issue by adding a new special rotor, which has this feature. We have also solved practical and mechanical issues, which limited and influenced the key space cipher. By removing weaknesses of the original cipher, which was used during World War II, we got the cipher, which keyspace is twice stronger.

The next chapter of the practical part presents programmer's manual. It contains an example of the code from the app, descriptions of each class and information how each substitution is implemented.

In the final chapter created app is presented to the user by the user manual. We have mentioned software requirements in the manual, described graphical user interface and explained app's working principles and added a few practical tips.

We believe that research goals were met. While studying the topic we have been surprised how German dignitaries and cryptanalysts absolutely believed in an unbreakability of the cipher despite they knew its weak side.

# A Obsah přiloženého DVD

Struktura přiloženého DVD se skládá ze 3 složek, bin, doc a src, ve kterých je uložena práce a aplikace spolu se zdrojovými kódy. Disk dále obsahuje soubor readme.txt s instrukcemi pro spuštění aplikace.

**`bin/`**
: Aplikace Enigmy se nachází přímo v kořenou složky bin pod názvem Tom's Enigma.exe.

**`doc/`**
: Text práce ve formátu PDF, vytvořený s použitím závazného stylu KI PřF UP v Olomouci pro závěrečné práce, včetně všech příloh, a všechny soubory potřebné pro bezproblémové vygenerování PDF dokumentu textu (v ZIP archivu), tj. zdrojový text textu, vložené obrázky, apod.

**`src/`**
: Kompletní zdrojové kódy programu Enigma se všemi potřebnými soubory pro bezproblémové vytvoření spustitelných verzí programu se nacházejí v ZIP archivu.

**`readme.txt`**
: Instrukce pro instalaci a spuštění programu, včetně všech požadavků pro jeho bezproblémový provoz.

# Literatura


[1] KAHN, David. *Seizing the enigma: The Race to Break the German U-Boat Codes, 1939–1943*. Revised edition. Boston: Houghton Mifflin Co., 1991. ISBN 978-1-84832-636-1 US.

[2] REUVERS, Paul; SIMONS, Marc. *Crypto Museum* [online]. 2009-08-11 [cit. 2019-4-10]. Dostupný z: ⟨https://www.cryptomuseum.com⟩.

[3] OSTWALD, Olaf; WEIERUD, Frode. History and Modern Cryptanalysis of Enigma's Pluggable Reflector. *Cryptologia*. 2015-08-07, roč. 40, č. 1, s. 71. Dostupný také z: ⟨http://www.tandfonline.com/doi/full/10.1080/01611194.2015.1028682⟩. ISSN 0161-1194.

[4] HAMER, David H. ENIGMA: ACTIONS INVOLVED IN THE 'DOUBLE STEPPING' OF THE MIDDLE ROTOR. *Cryptologia*. 1997, roč. 21, č. 1, s. 47–50. Dostupný také z: ⟨http://www.tandfonline.com/doi/abs/10.1080/0161-119791885779⟩. ISSN 0161-1194.

[5] MILLER, A. Ray. THE CRYPTOGRAPHIC MATHEMATICS OF ENIGMA. *Cryptologia*. 1995, roč. 19, č. 1, s. 65–80. Dostupný také z: ⟨http://www.tandfonline.com/doi/abs/10.1080/0161-119591883773⟩. ISSN 0161-1194.

[6] OSTWALD, Olaf; WEIERUD, Frode. History and Modern Cryptanalysis of Enigma's Pluggable Reflector. *Cryptologia*. 2015-08-07, roč. 40, č. 1, s. 80. Dostupný také z: ⟨http://www.tandfonline.com/doi/full/10.1080/01611194.2015.1028682⟩. ISSN 0161-1194.

[7] REUVERS, Paul; SIMONS, Marc. *Plugboard Mark 3* [online]. [Cit. 2019-5-11]. Dostupný z: ⟨https://www.cryptomuseum.com/crypto/enigma/i/img/sb_mk3.png⟩.

[8] SALE, Anthony Edgar. *Counting the Possible Plugboard Settings* [online]. [cit. 2019-5-15]. Dostupný z: ⟨https://www.codesandciphers.org.uk/enigma/steckercount.htm⟩.

[9] WEISSTEIN, Eric W. *Double Factorial* [online]. 1999 [cit. 2019-5-15]. Dostupný z: ⟨http://mathworld.wolfram.com/DoubleFactorial.html⟩.

[10] SINGH, Simon. *Kniha kódů a šifer: tajná komunikace od starého Egypta po kvantovou kryptografii*. Praha: Dokořán, 2003. ISBN 80-720-3499-5.