Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

Ing. Ivo Motýl

Výzkum využití možností fraktální geometrie pro zabezpečení informačních systémů

Research into the usage of the fractal geometry possibilities for the information systems security

Disertační práce

Studijní obor: Inženýrská informatika
Školitel: doc. Mgr. Roman Jašek, Ph.D.
Zlín, 2012

Poděkování:

Rád bych zde poděkoval mému vedoucímu doc. Mgr. Romanu Jaškovi, Ph.D. za jeho podmětné rady, podporu a přátelský přístup, kterého si velmi vážím, v průběhu mého studia a výzkumu.

Dále bych chtěl poděkovat mé skvělé rodině, která mne vždy podporovala, která je mým vzorem a motivátorem, o kterou jsem se mohl vždy opřít, a nikdy mne nezklamala.

Chtěl bych také poděkovat mé přítelkyni a všem mým dobrým přátelům, kterých si velmi vážím, za skvělé chvíle a zážitky strávené s nimi.

ABSTRAKT

Tato disertační práce se zabývá využitím principů fraktální geometrie využitelných v oblasti kryptografického zabezpečení komunikace v rámci informačních systémů. Teorie navrženého řešení vychází z oblasti iterativních fraktálů vytvořených pomocí algoritmu TEA.

V úvodní části je zpracována problematika volby vhodné kategorie fraktálů pro účel daný tématem disertační práce. Práce se v dalších krocích zabývá problematikou generování, analýzou fraktálních struktur, která je předpokladem pro provedení navrženého šifrovacího procesu. Výstupy z provedené analýzy fraktálu slouží také i pro dešifrovací proces. V závěru teoretické části je popsána metodika způsobu testování navrženého řešení vůči kryptoanalytickým metodám.

V experimentální části disertační práce byly jednotlivé prvky navrženého procesu realizovány pomocí naprogramovaného rozhraní v jazyce C#. Experimentální část dále pokračuje testováním odolnosti daného způsobu šifrování vůči kryptoanalytickým metodám. Pro zkoumání odolnosti byly použity statistické metody, analytické metody a útok hrubou silou. Získané poznatky prokázaly využitelnost navrženého řešení pro zvolenou oblast svého použití.

ABSTRACT

This thesis deals with the usage of fractal geometry principles for use in the cryptographic security of communications within the information systems. The theory of the proposed solution is based on the iterative fractals area generated by the algorithm TEA.

The first part analyzes the problem of choice of the appropriate category of fractals for the purpose of the dissertation topic. In next steps, this thesis deals with the generation and analysis of fractal structures which is a prerequisite for the implementation of the proposed encryption process. Outputs from the fractal analysis are also used for the decryption process. In the conclusion of the theoretical part, the methodology of testing the method of the proposed solution to cryptanalysis methods is described.

In the experimental part of the thesis the individual elements of the proposed process were implemented by the interface programmed in C #. Experimental part continues with testing of the resistance of the way to cryptanalysis encryption methods. In order to examine resistance statistical methods, analytical methods and brute force attack were used. The gained knowledge has shown the utility of the proposed solution for the selected area of its application.

OBSAH

SEZNAM OBRÁZKŮ

SEZNAM TABULEK

SEZNAM POUŽITÝCH ZKRATEK

Symbol	*Popis*
BFA	Brute Force Attack
COA	Ciphertext Only Atack
IS	Informační systém
TEA	Time Escape Algorithm

1 ÚVOD

Zajištění bezpečnosti přenášených informací od odesílatele k příjemci a snaha znemožnit nežádoucí osobě získání důvěrného obsahu, který nesou, je úkol provázející člověka již od starověku. S rozvojem lidského poznání a technického pokroku jsou šifrovací metody postupně vylepšovány a nahrazovány novými, které splňují nároky a požadavky na bezpečnost v čase svého nasazení. V době rozmachu výpočetní techniky narůstá rozvoj kryptografie do nebývalých rozměrů. Ta již není jen nástrojem vladařů, státníků, špionů a úzkého okruhu zainteresovaných lidí, kteří ji v minulosti využívali, ale stává se nyní běžným nástrojem člověka naší doby.

Existuje mnoho způsobů, jakými lze informaci chránit před nežádoucími zraky třetí osoby. V disertační práci je navržen a zpracován způsob šifrování informace založený na principech fraktální geometrie, konkrétně na jedné z jejich skupin – iterativních fraktálůvytvořeních pomocí algoritu TEA. Fraktální geometrie zasahuje svým rozsahem do mnoha oborů lidské činnosti. V tomto případě propůjčuje tato relativně mladá vědecká disciplína svůj potenciál v oblasti problematiky informační bezpečnosti.

Bezpečnost přenášených informací v rámci informačních systémů je jeden z mnoha klíčových faktorů úspěšného nasazení v praxi ať už ve vazbách člověk – člověk, člověk – stroj či stroj – stroj. Obsah disertační práce je na tuto problematiku úzce zaměřen.

Osnova kapitol teoretické i praktické části popisuje jednotlivé procesy a metodiky v takovém pořadí, v jakém se nacházely při dílčích úkonech řešení cílů disertační práce.

Úvodní část disertační práce popisuje oblast řešené problematiky, vytyčuje cíle disertační práce a dále pak metody zabezpečení informace používané v současnosti.

Kapitola 4 se zabývá principy fraktální geometrie a dále rozebírá jednotlivé skupiny fraktálu podle způsobu jejich konstrukce.

V kapitole 5 je popsán proces zabezpečení informace za použití iterativního fraktálu vytvořeném pomocí algoritmu TEA. Ve svých podkapitolách uvádí procesy generování fraktálu, analýzy fraktální struktury a dále průběh procesu zašifrování a dešifrování informace. V návaznosti na tyto body jsou zde popsány procesy rozšiřující navržený

algoritmus o šifrování dlouhých zpráv a dále je zde popsán navržený proces použití unikátního klíče, který zvyšuje odolnost navrženého řešení vůči kryptoanalytickým metodám.

Kapitola 6 sumarizuje metody kryptoanalýzy, které byly použity na testování odolnosti navrženého řešení.

Praktická část disertační práce začíná kapitolou 7, která popisuje řešení problematiky generování vhodné fraktální struktury u jednotlivých druhů fraktálů.

V návaznosti na tuto kapitolu jsou v kapitole 8 demonstrovány varianty šifrovacích a dešifrovacích procesů a dále popsány jejich vlastnosti a parametry.

Kapitola 9 shrnuje poznatky získané kryptoanalýzou navrženého řešení a formuluje podklady pro stanovení jeho odolnosti vůči kryptoanalytickým metodám.

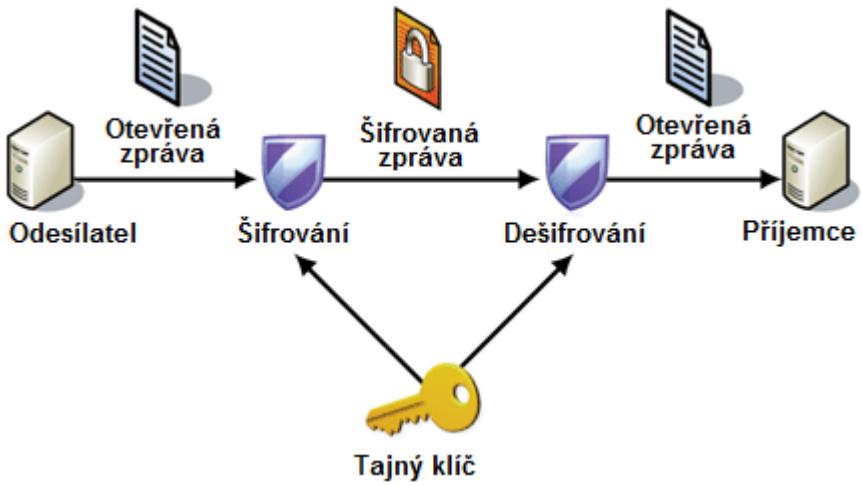
Kapitola 10 popisuje naprogramované rozhraní v jazyku C# použité pro demonstraci šifrovacích a dešifrovacích procesů a testování navrženého řešení. Dále jsou v kapitole 11 shrnuty hardwarové a softwarové prostředky použité v rámci výzkumu a všech činností spojených se zpracováním tématu disertační práce.

V závěrečné části práce popisuje kapitola 12 možnosti využití získaných poznatků obsažených v disertační práci pro vědu a praxi. Kapitola 13 shrnuje závěry výzkumu a jednotlivé části řešené problematiky.

2 SOUČASNÝ STAV ŘEŠENÉ PROBLEMATIKY

V dnešní době se setkáváme s rozmachem počítačových sítí [34] a internetu [36], kde putují přenosovým kanálem informace od odesílatele k příjemci. Zde vyvstává nutnost tyto informace ochránit před odposlechnutím. Většina technik odposlechu [32] neskrývá velkou složitost a útok na informaci nepředstavuje ve většině případů finančně náročné řešení. V současné době jsou pro zabezpečení informace [81] nejvíce rozšířeny základní dva druhy kryptografických metod [66], [83]. První z nich je Symetrická kryptografie [66], druhým Asymetrická kryptografie [66]. Každý ze způsobů má své klady i zápory a je nasazen tam, kde je jeho předností zapotřebí. Setkáváme se také jejich kombinacemi, jež nazýváme hybridní kryptografií.

Přednosti symetrické kryptografie [66], [41], jsou v její rychlosti ve srovnání s asymetrickou [66]. Pro její nasazení není třeba značné výpočetní rychlosti. Pro šifrování a dešifrování zprávy se používá stejný klíč. Před zahájením komunikace se musí vysílací a přijímací strana dohodnout na společném klíči.



Obr. 1: Symetrická kryptografie [57]

Asymetrické šifrování [83], [39], se vyznačuje různými klíči pro šifrování a dešifrování zprávy. Pro zašifrování zprávy se používá tzv. veřejný klíč, pro dešifrování

soukromý. Veřejný klíč uživatele je znám každém účastníkovi komunikace. Privátní klíč je unikátní pro každého účastníka. Zašifrování probíhá tak, že odesílatel zašifruje zprávu veřejným klíčem adresáta. Adresát zprávu přijme, použije svůj privátní klíč a zprávu přečte. Při procesu vytváření dvojice klíčů platí pravidlo neodvoditelnosti privátního klíče z klíče veřejného.

Obr. 2: Asymetrická kryptografie [57]

Hybridní šifrování [22] využívá výhody obou výše zmíněných šifrovacích metod [66]. Od symetrického šifrování [14] přebírá rychlost, od asymetrického [13] použitelnost a existenci dvojic klíčů. Kryptografický cyklus probíhá zašifrováním zprávy symetrickým klíčem, symetrický klíč je následně zašifrován veřejným klíčem příjemce a je poslán se zprávou. Příjemce obdrží zprávu, použije soukromý klíč na zjištění symetrického klíče a pomocí něj zprávu dešifruje.

Principy z oblasti fraktální geometrie byly využity v problematice generování veřejného klíče na základě privátního klíče s využitím principu vnitřního spojení fraktálů Mandelbrotovy množiny a Juliových množin. Tato problematika je více popsána v literatuře [3]. Problematika digitálního podpisu s využitím poznatků fraktální geometrie

je více rozvedena v kapitole [1]. Problematika fraktální geometrie v souvislosti s HASH [19], [43] funkcemi je vice rozvinuta v literatuře [59].

3 CÍLE DISERTAČNÍ PRÁCE

Za hlavní přínos disertační práce považuji nalezení způsobu pro zabezpečení dat proti nežádoucímu získání jejich obsahu. Zapojení odvětví fraktální geometrie do oblasti informační bezpečnosti otevírá nové možnosti, dané odlišným pojetí fraktálů, na rozdíl od objektů klasické euklidovské či jiné geometrie.

Navržený systém pracuje se složitými fraktálními strukturami, které lze popsat poměrně triviálními rovnicemi, což dovoluje používat tento systém s velkou rychlostí jak pro zakódování zprávy, tak i pro její zpětnou rekonstrukci. Tato skutečnost otevírá cestu využití navrženého systému pro zabezpečení informace i v zařízeních s omezenými výpočetními kapacitami.

Systém klade důraz na odolnost vůči kryptoanalytickým metodám, jako je například útok hrubou silou, statistické metody či analytické metody.

Cíle disertační práce jsou shrnuty do následujících bodů:

- Nalézt vhodnou kategorii fraktálů, využitelnou v oblasti zabezpečení informačních systémů
- Navrhnout a ověřit způsob vhodný pro fraktální zabezpečení informace
- Určit odolnost navrženého řešení vůči kryptoanalytickým metodám

TEORETICKÁ ČÁST

4 FRAKTÁLNÍ GEOMETRIE

Fraktální geometrie zkoumá objekty nazývané fraktály [54], [96]. Její prezence na poli seriózní vědy se datuje od 70. let, kdy matematik, Bernoit Mandelbrot definoval pojem fraktál, z latinského slova *fractus [70*], neboli rozbitý. První záchvěvy toho, co dnes nazýváme fraktály, však vznikaly dříve. Již v 19. století nacházela tehdejší matematika podivné obrazce a struktury fraktálního charakteru [54], avšak nebyla jim věnována dostatečná pozornost. Stále neexistuje matematická definice fraktálu. Nejlépe se k definici fraktálu blíží Mandelbrotovo tvrzení [54], které definuje fraktál jako takový útvar, jehož Hausdorfova dimenze[86] je větší než dimenze topologická [45]. Fraktální objekty neexistují jen ve světě matematiky. Fraktální struktury lze najít v okolním světě například ve formě oblaků, stromů, listů, skal, reliéfu terénu a mnoha dalších přírodních objektů.

Principy fraktální geometrie umožňují její využití zejména v oblastech kompresních algoritmů, artware, studia dynamických systémů, modelování chemických a fyzikálních procesů či v oblasti bezpečnostních technologií.

4.1 Klasifikace fraktálů a jejich konstrukce

Fraktály lze rozčlenit do kategorií na základě různých úhlů pohledu. Jedním z nich je klasifikace podle způsobu jejich konstrukce. Jedna z významných skupin fraktálů je založena na algoritmu IFS (Iteration Function System) [96]. Mezi další skupinu fraktálů patří skupina založená na algoritmu TEA [96]. Podle způsobu, jakým byl konkrétní fraktál vytvořen a jaké má vlastnosti, nachází každá skupina uplatnění pro konkrétní účel v praxi. Způsobů konstrukce existuje více. Pro zpracování cílů disertační práce bylo využito algoritmu TEA.

Skupina fraktálů vytvořených pomocí algoritmu TEA [96], patří mezi velmi rozšířenou. Příslušný fraktál této skupiny je definován svojí rovnicí a počátečními podmínkami. Rovnice je prováděna iteračně podle stanovených podmínek. Podmínky mohou být mezi sebou provázány a proces generování fraktálu tak závisí na jejich vzájemných kombinacích. Jedním z případů může být například stanovení daného počtu iterací a současně hodnota příslušného parametru rovnice, který se nesmí překročit.

Parametry, které figurovaly při řešení cílů disertační práce jsou uvedeny v kapitole 5.2.2. U této skupiny fraktálů je vyšetřován bod na ploše či v prostoru [54]. Po ukončení daného procesu je příslušný bod vykreslen odstínem úměrným počtu provedených iterací, které bylo třeba vykonat na opuštění zvolené hranice. Mezi nejznámější zástupce z fraktálů, kde lze použít TEA algoritmus patří Juliovy množiny [54], [96] Mandelbrotova množina [54], [96] a fraktály postavené na jejím základu.

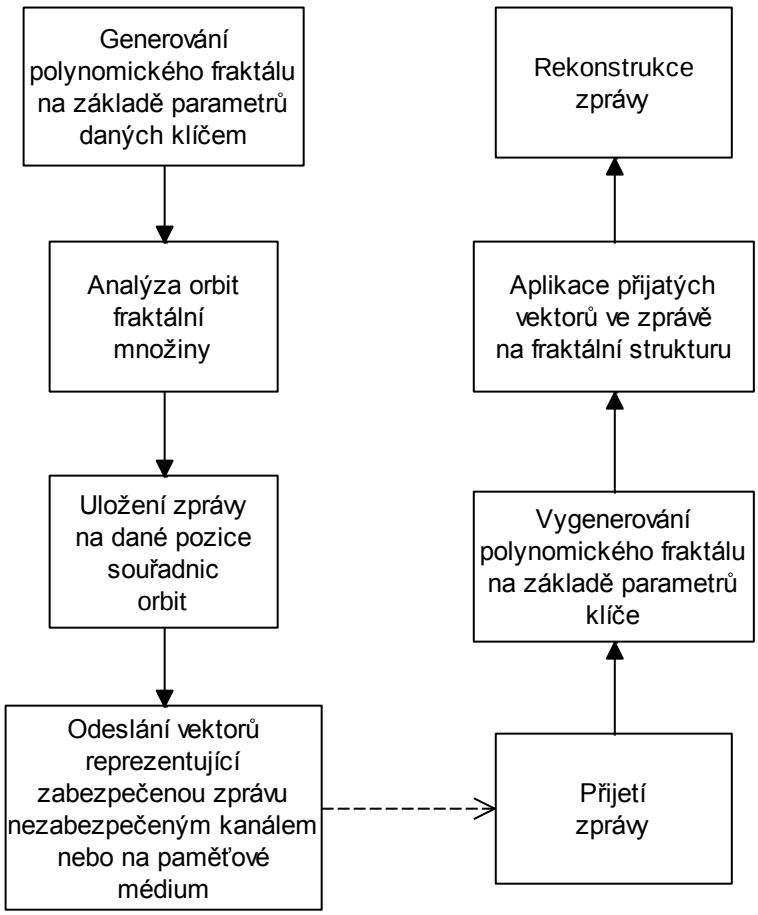
5 ZAŠIFROVÁNÍ INFORMACE

5.1 Proces využití bodů fraktální struktury pro zašifrování informace

Provedený výzkum fraktální geometrie, nasměroval postup v práci na oblast využití kategorie fraktálů vytvářených pomocí algoritu TEA. Tyto fraktály, jak se ukázalo, umožnily svými principy jejich aplikaci na řešení cílů disertační práce.

Proces začíná generováním některého z fraktálů algoritmem TEA. Proces generování fraktálu probíhá na základě předem daných podmínek vztažených jak ke konstrukčním vlastnostem fraktálu, tak k účelu použití výstupní fraktální struktury. Více informací o problematice *generování fraktálních struktur*, je obsaženo v kapitole 5.2. Fraktály, které byly použity, jsou uvedeny v příloze Příloha B. Po operaci vygenerování fraktálu, lze zahájit proces šifrování zprávy. Vizualizované body, jež reprezentují vygenerovaný fraktál, mají různé odstíny barev. Každý odstín barvy je reprezentován číslem. Toto číslo reprezentuje počet iterací provedeného algoritmu při výpočtu daného bodu fraktální množiny. O každém takto vykresleném bodu známe informaci o jeho souřadnici *x,* o souřadnici *y* a informaci o barevném odstínu. Na základě těchto hodnot můžeme provést zakódování zprávy do tzv. *vektoru*. Tento vektor může být uložen, či bezpečně převeden nezabezpečeným kanálem k příjemci. Proces zakódování zprávy probíhá tak, že číslo barevného odstínu definuje hodnotu znaku. Jako znak lze chápat písmena, číslice a další speciální znaky. Při zakódování budou na zobrazeném fraktálu vyhledány body, které mají hodnoty barevného odstínu rovnu hodnotám zakódovaným znakům reprezentovaných v tabulce vstupních znaků, které lze zabezpečit. Poté budou zjištěny jejich souřadnice *x, y* a následně zapsány do výstupního souboru. Aby bylo zabezpečení ještě účinnější a odolalo například frekvenční analýze, bylo stanoveno pravidlo, že žádná souřadnice nesmí být použita dvakrát a souřadnice budou hledány na ploše fraktální množiny pomocí generátoru náhodných čísel. Po přijetí zprávy příjemcem a dekódování dat dojde k zadání klíče pro vygenerování stejného fraktálu jako při procesu kódování. Následně budou „přiloženy“ přijaté vektory souřadnic na vygenerovaný fraktál a odečteny hodnoty bodů, které následně vydají obsah zprávy. Před prvním použitím mechanismu je nutná vzájemná dohoda na klíči použitým pro šifrování.

Lze tedy říci, že podobně jako u jiných šifrovacích mechanismů, síla zašifrování nespočívá v neznalosti, jaké transformace byly použity, ale ve značné šíři možných řešení a tudíž značné výpočetní náročnosti na rozluštění a následné analýze vzniklých datových struktur. Tento princip popisuje tzv. *Kerckhoffův předpoklad* [56]. Celý popsaný postup šifrování a dešifrování ukazuje blokové schéma na *Obr. 3*.



Obr. 3: Blokové schéma procesu zabezpečení a rekonstrukce zprávy

5.2 Popis implementovaného algoritmu pro zašifrování informace

Pro účely testování a vývoje algoritmu byl vytvořen program ve vývojovém prostředí Microsoft Visual Studio 2010. Více informací o naprogramovaném prostředí je uvedeno v 10. kapitole *Rozhraní pro šifrování, dešifrování a testování navrženého* řešení. Proces

zašifrování informace postavený na principech fraktální geometrie se skládá z trojice po sobě jdoucích operací. Tyto operace byly nazvány: *Generování fraktálu, Analýza fraktálu* a *Zašifrování informace*. Generování fraktálu bylo implementováno do testovacího programu manuálně i automaticky. Součástí automatického generování fraktálu bylo implementováno podrobné nastavení generujícího procesu.

5.2.1 Manuální generování fraktálu

Manuální generování fraktálu lze provést dvěma způsoby. Pomocí myši nebo zapsáním jeho parametrů do příslušných textových polí v programu.

Prvním způsobem byla fraktální struktura vygenerována pomocí kliknutí kurzoru myši do příslušné oblasti fraktální množiny. Po provedení kliku byl tento bod zaznamenán, zvolen za nový střed zobrazení a proveden dvojnásobný zoom v této oblasti. Parametry provedeného kliknutí jsou zaznamenány v podobě souřadnic zvoleného bodu. Dále zde figuruje parametr rozsah a počet iterací. Informaci o maximální délce otevřeného textu udává parametr *Maximální délka zprávy*. Tento postup lze opakovat a procházet tak do větší hloubky dané fraktální množiny, nebo proces ukončit a zvolit tak výsledek jako výsledný fraktál pro využití v dalším kroku navrženého procesu.

Druhý způsob spočívá v zadání parametrů středu zobrazení *x* a *y*, *Počáteční rozsah* a *Počet iterací.* Po stisknutí tlačítka pro generování dojde k vygenerování fraktální struktury s danými parametry. Tento postup může být aplikován při procesu šifrování, ale je zejména využíván při dešifrování. Může být prováděn manuálně nebo automatizován.

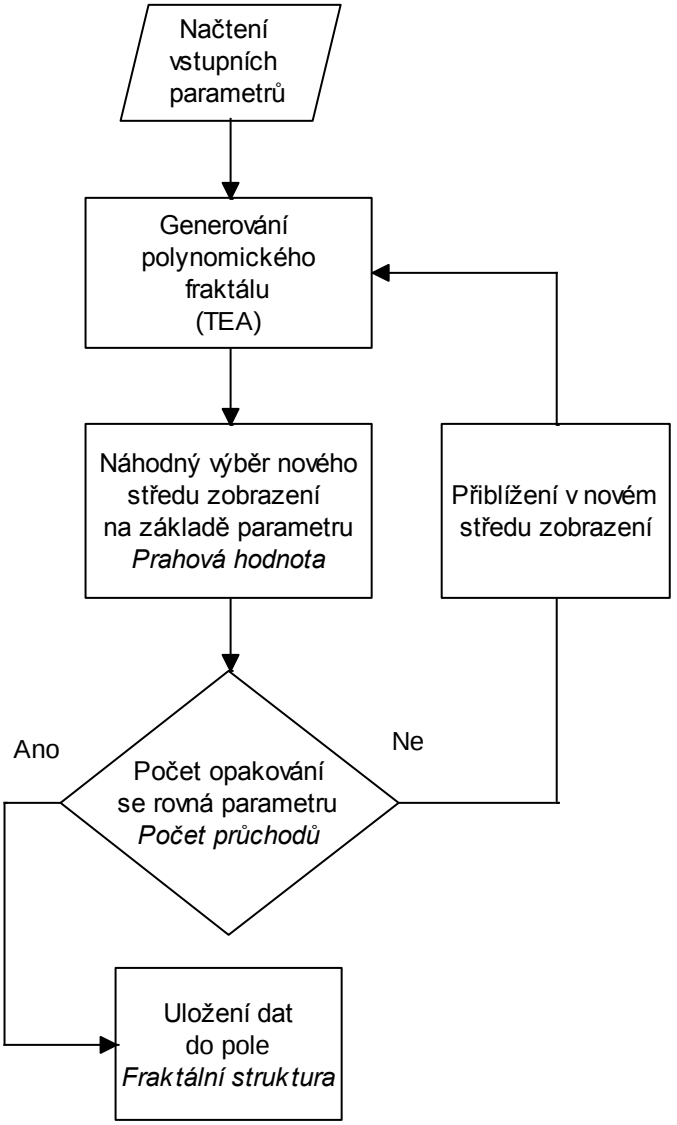
5.2.2 Automatické generování fraktálu

Proces automatického generování pracuje podobně jako proces manuálního generování. Proces manuálního generování je popsán v kapitole 5.2.1. Rozdíl spočívá ve způsobu výběru nového bodu pro další iteraci generovaného fraktálu.

Proces začíná načtením vstupních parametrů. Tyto parametry obsahují hodnoty: *Počet průchodů, Prahová hodnota bodu, Počáteční rozsah a Počet iterací.* Význam jednotlivých parametrů popisuje kapitola 5.2.6. Před procesem generování se výchozí souřadnice souřadného systému nachází ve svých počátcích. Z parametru *Počáteční rozsah* je určena

oblast *kartézských souřadnic*, na které bude započato vygenerování fraktálu. Po této operaci dochází k první iteraci generování fraktální množiny. Ve vnořených cyklech jsou vyšetřovány jednotlivé body souřadného systému a testovány na podmínky, které odpovídají konstrukci příslušného fraktálu. Popis použitých fraktálů obsahuje příloha Příloha B. Způsob generování fraktálů metodou TEA je detailně popsán v literatuře [*96*]. Na konci tohoto kroku je vytvořeno dvourozměrné pole, které nese informaci o každém vyšetřovaném bodu. Tato informace říká, ve které iteraci fraktální rovnice opustil vyšetřovaný bod hranici definovanou podmínkami konstrukce fraktálu. Množství těchto iterací udává vstupní veličina *Počet iterací*, kterou lze modifikovat a výrazně tak měnit podobu výsledného fraktálu. Parametr *Počet průchodů* představuje počet simulovaných kliknutí kurzoru myši na plochu fraktálu. Výběr nového středu zobrazení fraktálu je vykonán pomocí generátoru náhodných čísel. Tento výběr je však dále ovlivněn parametrem *Prahová hodnota bodu*. Tato hodnota udává minimální počet iterací, který musí obsahovat náhodně vybíraný bod, aby byl zahrnut do výběru pro nový střed zobrazení. Tento parametr zajišťuje nalezení struktury, která obsahuje široký rozsah hodnot bodů fraktální struktury, což je výhodné pro daný problém. Vylučuje body, u nichž klesá pravděpodobnost vygenerování složité struktury v další iteraci. Po nalezení nového bodu dochází k další iteraci procesu. Počet iterací je dán parametrem *Počet iterací*. Po ukončení iterací je vygenerována výsledná fraktální struktura.

Vývojový diagram procesu automatického generování je zobrazen na Obr. 4



Obr. 4: Schéma procesu automatického generování fraktální struktury

5.2.3 Kombinované generování

Kombinované generování fraktální množiny je možné provést kombinací manuálního generování, popsaného v kapitole 5.2.1 a automatického generování, popsaném v kapitole 5.2.2. Po provedení automatického generování lze vygenerovanou fraktální strukturu dále modifikovat pomocí kurzoru myši.

Po provedení této části je provedena fáze *analýzy fraktálu*.

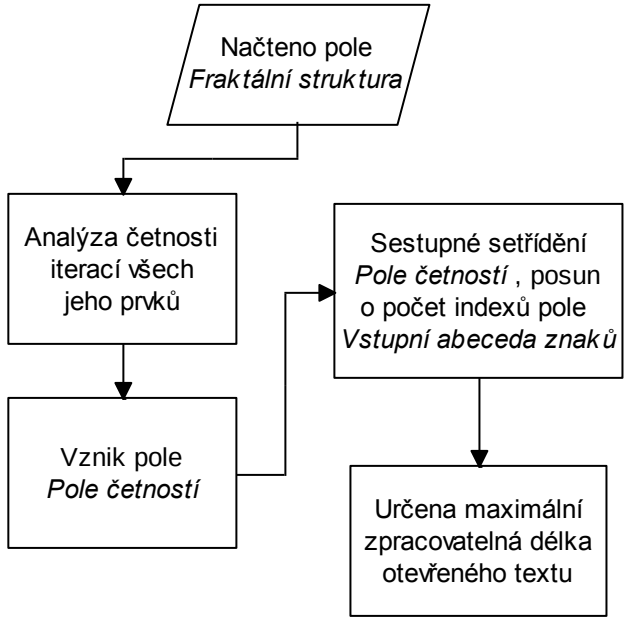
5.2.4 Analýza fraktálu

V této části popisovaného algoritmu je provedena analýza vygenerovaného pole bodů fraktální struktury a příprava pro zabezpečení zprávy. Před uložením informace je nutné podrobit fraktální strukturu analýze a zjistit její schopnost pojmout informaci, která má být zabezpečena.

Po vygenerování fraktálu je zobrazena jeho vizualizace na hlavním okně programu. Jak již bylo zmíněno v části 5.2.2, fraktální struktura je uložena ve dvourozměrném poli, nazvaném *Fraktální struktura*, kde je každý vykreslený pixel reprezentován svými souřadnicemi a počtem iterací. V dalším kroku je toto pole analyzováno a spočítána četnost iterací všech jednotlivých prvků, které obsahuje. Vznikne tím datové pole, nazvané *Pole četností*, kde jsou tyto četnosti zaznamenány. V prvním indexu tohoto pole je zaznamenán počet bodů, které splnily konstrukční podmínky fraktálu po první iteraci, ve druhém indexu jsou zahrnuty četnosti bodů, které tyto podmínky splnily ve druhé iteraci, atd. Maximální počet indexů tohoto pole souvisí s nastaveným parametrem *Počet iterací*.

Vstupní abeceda obsahuje znaky abecedy A-Z, číslice 0-9 a další znaky, jako jsou desetinná čárka, znak mínus nebo znak mezery či speciální znak oddělovače klíče. Více o vstupních znacích pojednává kapitola 5.2.8. Tyto informace jsou uložené v jednorozměrném poli, nazvané *Vstupní abeceda znaků*, kde každý index obsahuje jeden ze znaků vstupní abecedy. Po operaci sestupného setřídění *Pole četností* a posunu o celkový počet indexů, které má pole *vstupní abeceda znaků*, lze určit maximální délku informace, která může být zašifrována.

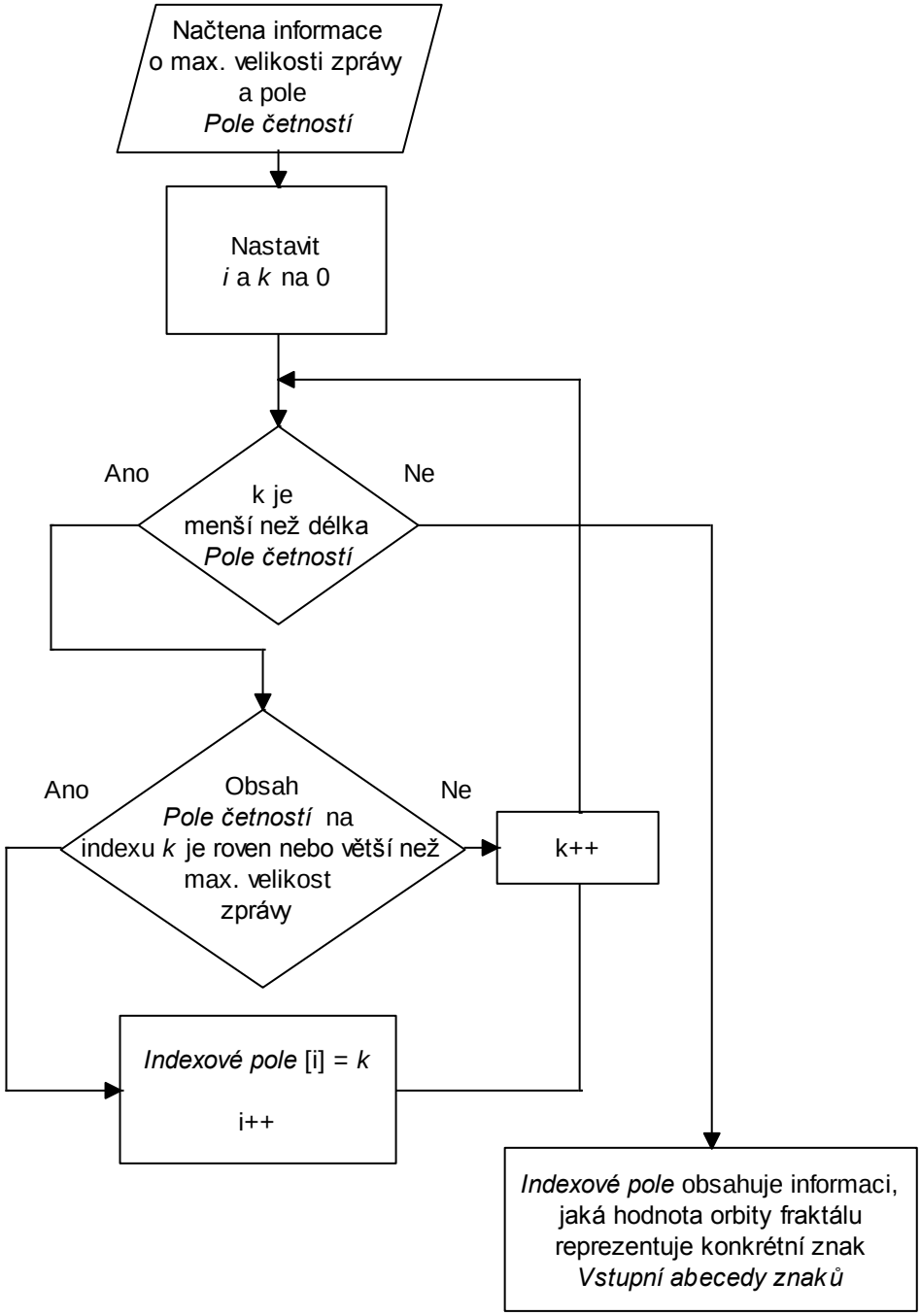
Schéma procesu analýzy fraktálu je zobrazeno na Obr. 5.



Obr. 5: Zjištění maximální délky zpracování otevřeného textu na základě vygenerované fraktální struktury

V dalším kroku dochází k tzv. mapování znaků. Zde je určeno, jaká hodnota bodu z fraktální struktury bude zastupovat konkrétní znak vstupní abecedy. Pro tuto operaci je vytvořeno nové pole, zvané *Indexové pole*. V cyklu je procházeno *Pole četností*. Pokud má jeho index hodnotu obsahu větší nebo rovnu, než je určená maximální velikost zprávy, je mu přiřazeno číslo, které odpovídá indexu ve vstupní abecedě znaků. V další iteraci cyklu je inkrementován ukazatel *Indexového pole*. Při splnění výše popsané podmínky je do *Indexového pole* opět přiřazen index odpovídající písmenu vstupní abecedy. Tato operace proběhne tolikrát, kolik indexů obsahuje pole vstupní abecedy znaků.

Schéma procesu mapování znaků je zobrazeno na Obr. 6.



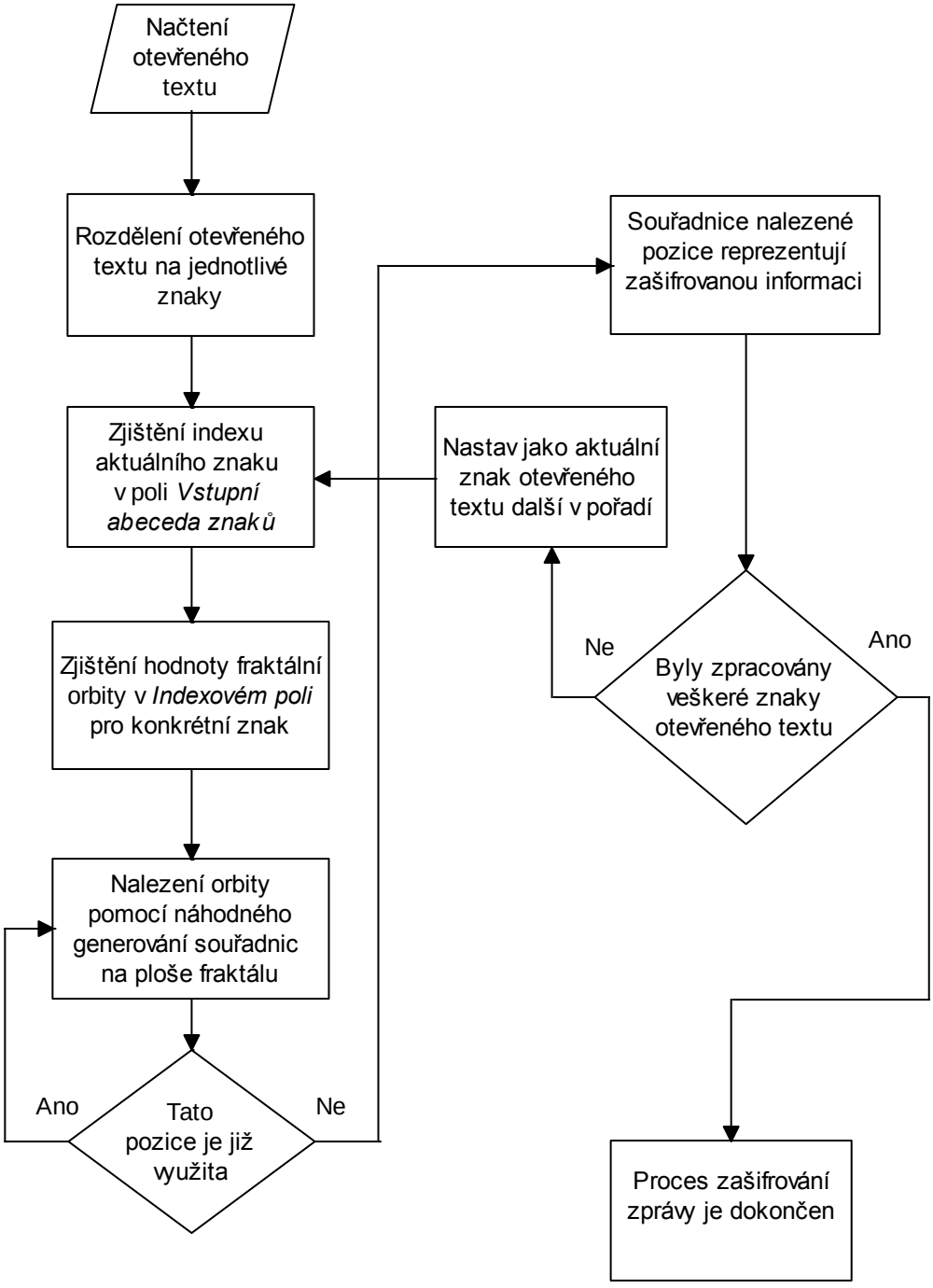
Obr. 6: Schéma procesu mapování znaků

5.2.5 Zašifrování informace

Po úspěšném provedení operací vygenerování fraktálu a jeho analýzy lze přistoupit k procesu zašifrování informace. Vstupní informace je přečtena z textového pole na kartě s názvem *Zašifrovat zprávu*, nacházející se na hlavním okně programu. Z vygenerovaného fraktálu je vypočítána maximální možná délka vstupní informace a stanoveno *Indexové pole*, které určuje body fraktální struktury, budoucí reprezentanty použitých znaků. Během vývoje způsobu zápisu byl brán ohled na odolnost vůči kryptoanalytickým metodám, více rozvinutých v kapitole 6. Z tohoto důvodu byl určen postup popsaný níže.

Zpráva je rozdělena na jednotlivé znaky. V každém cyklu je zpracován každý znak samostatně. V první fázi je zjištěn index znaku v poli *Vstupní abeceda znaků* na základě aktuálně zpracovaného znaku. Následně je vygenerována dvojice čísel v rozsahu *Fraktálního pole*, popsaném v kapitole 5.2.4. Protože je toto pole dvourozměrné, je třeba vygenerovat dvojici hodnot. Tyto hodnoty slouží jako indexy pro jeho procházení. Pokud je tímto náhodným generováním ve *Fraktálním poli* nalezena hodnota odpovídající indexu daného písmena v *Indexovém poli* a není tato buňka dosud využita, je do výstupního vektoru zaznamenána tato dvojice vygenerovaných hodnot. Tato dvojice reprezentuje zašifrovanou informaci. Pokud jsou již tyto hodnoty využity, proces se opakuje do doby, kdy budou unikátní. Tento cyklus probíhá pro všechny znaky vstupní informace. Proces je ukončen zpracováním posledního znaku zprávy.

Schéma procesu zašifrování informace je zobrazeno na Obr. 7.



Obr. 7: Proces zašifrování informace

5.2.6 *Parametry klíče*

Tabulka Tab. 1 ukazuje parametry, ze kterých se skládá klíč. Pomocí klíče je provedeno dešifrování zprávy a zobrazení šifrovaného textu v čitelné podobě. První dva parametry slouží pro lokaci na ploše, třetí definuje rozsah, rozměr hrany čtverce, na kterém bude generována fraktální struktura. Poslední parametr stanovuje počet iterací sloužící algoritmu TEA při konstrukci fraktálu. V případě použití více typů fraktálu lze parametry rozšířit o jejich identifikátory.

Tab. 1: Parametry klíče

x parametr	Vertikální parametr fraktální struktury
y parametr	Horizontální parametr fraktální struktury
Rozsah	Délka hrany čtverce definující rozsah zobrazení na ploše fraktální struktury
Počet iterací	Počet iterací při konstrukci fraktálu algoritmem TEA

5.2.7 *Parametry automatického generování fraktálu*

Tabulka Tab. 2 ukazuje parametry, které jsou vkládány do algoritmu pro automatické generování fraktální struktury.

Tab. 2: Parametry automatického generování fraktálu

Rozlišení	Počet bodů v horizontální a vertikální ose, které vykreslují fraktál
Počet průchodů	Počet iterací, při kterých je simulováno kliknutí uživatele a tím zoom uvnitř fraktální struktury
Práh	Hranice hodnoty bodu fraktální struktury, od které je volen tento bod jako nový střed zobrazení příštího průchodu
Počet iterací	Počet iterací při konstrukci fraktálu algoritmem TEA
Rozsah	Délka hrany čtverce při prvním průchodu definující rozsah zobrazení na ploše fraktální struktury

5.2.8 Vstupní znaky

Tabulka Tab. 3 sumarizuje vstupní znaky, kterými lze vyjádřit vstupní informaci – otevřený text. Tyto skupiny tvoří alfanumerické znaky, numerické a speciální. Speciální jsou vždy svázány s alfanumerickými či numerickými. Mezi speciální patří mezera – *Space,* znaménko mínus, tečka, čárka a znak oddělovače (ASCII 254). Tento znak od sebe odlišuje znaky klíče a dat. Více informací o funkci znaku oddělovače je popsáno v kapitole 8.

Tab. 3: Vstupní znaky

Alfanumerické	A – Z, 0 – 9
Numerické	0 – 9
Speciální	„Space“, „-“, „.“, „■“ , „ ,“

5.3 Popis implementovaného algoritmu pro dešifrování informace

5.3.1 Generování fraktálu

Před samotnou rekonstrukcí zašifrované zprávy je nutné provést nejdříve proces vygenerování fraktálu, pomocí kterého bylo provedeno zašifrování informace. Generování je provedeno klíčem. Pomocí parametrů, které jsou klíčem neseny, je vygenerována fraktální struktura. Generování je provedeno postupem popsaným v kapitole 5.2.1 ve třetím odstavci. Parametry klíče jsou popsány v kapitole 5.2.6. Po vygenerování fraktální struktury algoritmem TEA [96], je provedeno její uložení do dvourozměrného pole, tzv. *Fraktálního pole*. V dalším kroku je toto pole analyzováno a následně dojde k procesu rekonstrukce zašifrované informace.

5.3.2 Analýza fraktálu

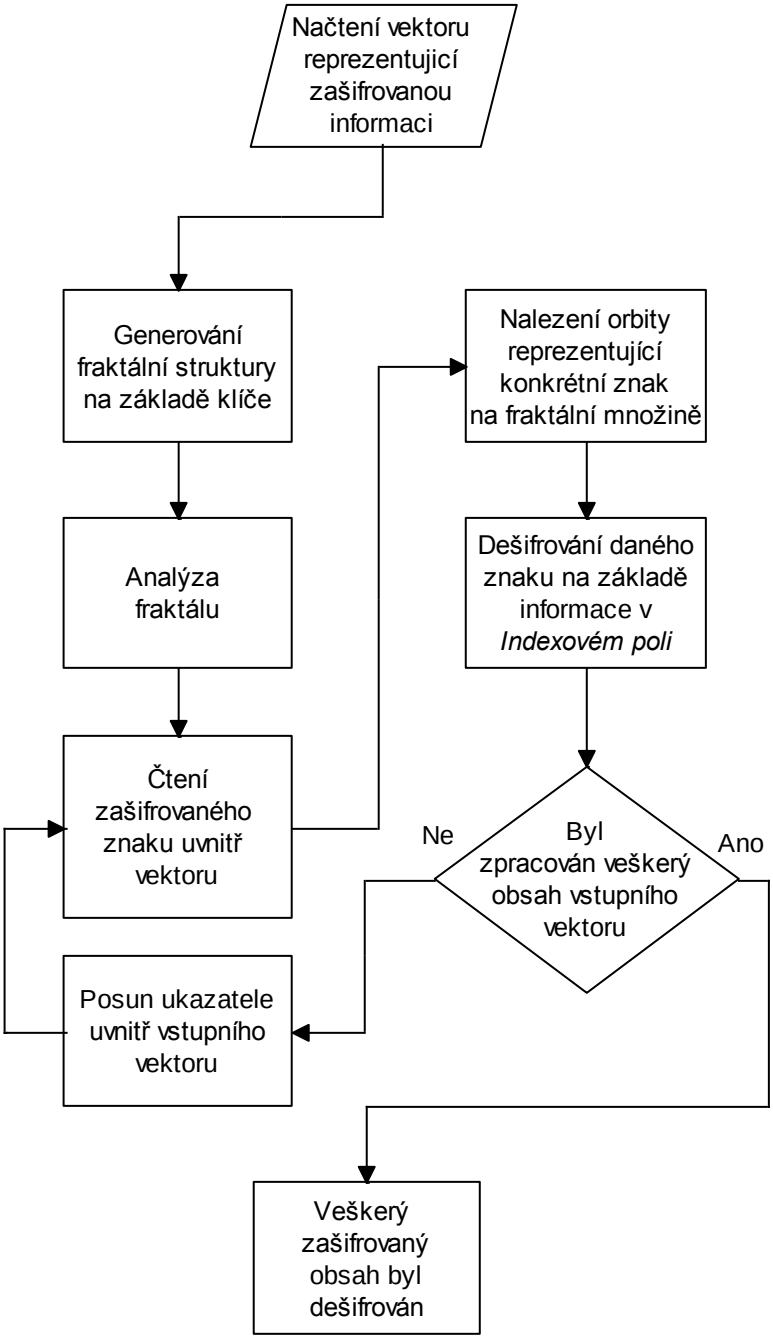
Podobně jako při procesu zašifrování informace, figuruje také zde, při dešifrování informace, proces analýzy fraktálu. Hlavním důvodem tohoto kroku je zjištění četnosti

iterací bodů fraktální struktury, uložené do dvourozměrného *Fraktálního pole* a poté určení maximální délky dešifrované zprávy. Tato informace je poté použita pro mapování *Indexového pole* a tím určení bodu fraktální struktury, který bude reprezentovat konkrétní dešifrovaný znak. Informace o množství a počtu iterací bodů fraktální struktury na fraktálu jsou uloženy v jednorozměrném poli s názvem *Pole četností*. Stejně jako při procesu zašifrování informace, popsaném v kapitole 5.2.5, je vytvořeno pole zvané *Indexové pole*, které udává informaci, jaká hodnota bodu z fraktální struktury bude zastupovat konkrétní znak *Vstupní abecedy*. *Pole četností* je postupně procházeno v cyklu. Jakmile je splněna podmínka, kdy konkrétní buňka v tomto poli obsahuje vyšší nebo rovnu hodnotu, než je maximální délka přenášené zprávy, je jeho obsah zaznamenán do příslušného indexu *Indexového pole*. Tato operace je provedena cyklicky. Po úspěšném ukončení tohoto cyklu je získána jednoznačná informace, která v *Indexovém poli* určí, jaká hodnota bodu fraktální struktury na ploše fraktálu zastupuje konkrétní znak pro přijaté zprávy.

5.3.3 Dešifrování informace

Po úspěšném provedení operací generování fraktálu a jeho analýzy lze zahájit krok rekonstrukce zašifrované informace. Zašifrovaná zpráva má tvar vektoru čísel, který obsahuje souřadnice směřující na konkrétní body fraktální struktury. Analýza fraktálu provedená v předcházejícím kroku určila v *Indexovém poli*, jaké písmeno zastupuje konkrétní hodnota bodu fraktální struktury.

Vektor obsahující zašifrovanou informaci, je čten postupně a na odpovídajících souřadnicích ve fraktálu, jsou nalézány body s čísly odpovídající hodnotě bodu fraktální struktury. Tato hodnota je nalezena v *Indexovém poli* a index tohoto pole odpovídá zašifrovanému znaku. Tento proces je ukončen zpracováním posledního znaku zašifrované zprávy. Po jeho ukončení je dešifrovaná informace zobrazena v příslušném textovém poli na kartě *Dešifrovat zprávu* v hlavním rozhraní programu popsaném v kapitole 10.



Obr. 8: Proces dešifrování informace

5.4 Rozšíření algoritmu pro zpracování dlouhých zpráv

Informace z kapitoly 5.2 uvádí, že maximální délka otevřeného textu, který lze zašifrovat, je určena na základě parametrů vygenerovaného fraktálu. Zvýšení této délky lze provést několika způsoby:

- Opětovným vygenerováním fraktálu s lepšími parametry
- Zvýšením rozlišení generovaného fraktálu
- Použití vícenásobného generování fraktálů

O možnosti použití vícenásobného generování fraktálů pojednává kapitola 5.4.1. První dva způsoby jsou možné, ale jak se ukázalo, technicky nepraktické. U prvního způsobu není zaručeno, že bude nalezen fraktál, který zcela splní požadavky na zašifrování dlouhé zprávy a u druhého způsobu navíc mohou narůstat nároky na výpočetní výkon hardware.

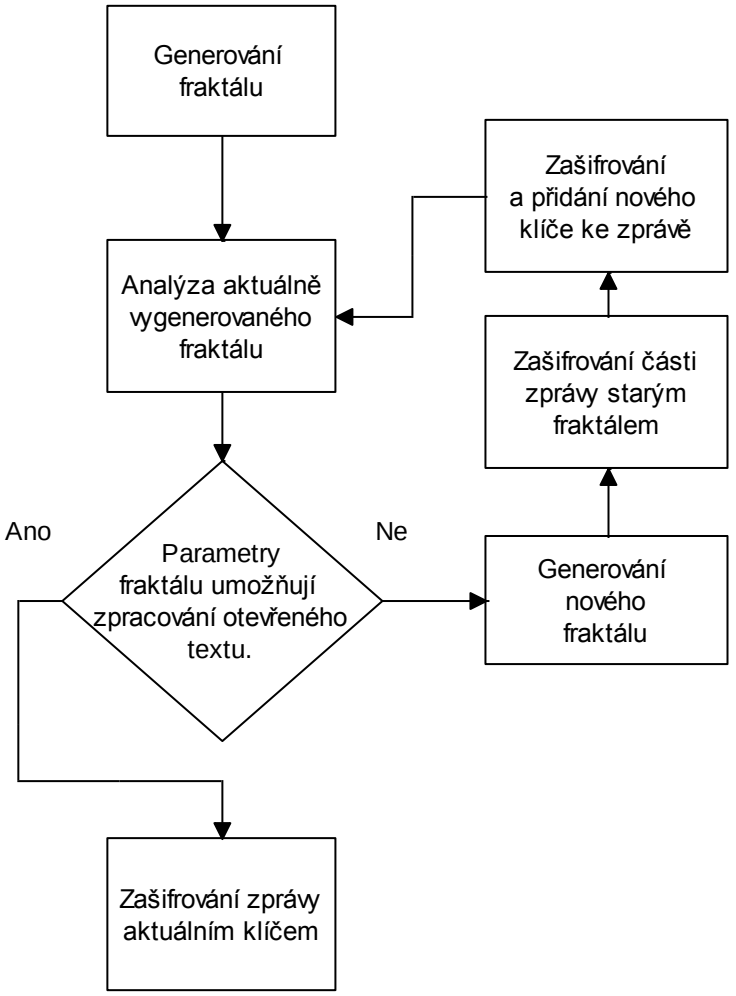
5.4.1 Proces vícenásobného generování při zašifrování zprávy

Na Obr. 9 je znázorněn vývojový diagram popisující proces vícenásobného generování fraktálu. Tento způsob slouží v případě zpracování otevřeného textu zprávy, kdy parametry vygenerovaného fraktálu nesplňují podmínky pro zpracování zprávy v jednom kroku, tzn. u dlouhých zpráv.

Samotný proces šifrování zprávy začíná generováním fraktálu na základě klíče. Proces generování je detailně popsán v kapitole 5.2. V dalším kroku je fraktální struktura podrobena analýze, popsané detailně v kapitole 5.3.2. Analýzou je primárně určeno, kolik znaků otevřeného textu daná fraktální struktura pojme. Na základě provedené analýzy je v dalším kroku stanoveno, zda parametry fraktálu umožňují zašifrování celého otevřeného textu najednou, nebo zda bude nutné použít metodu vícenásobného generování.

V prvním případě je proveden krok zašifrování zprávy, popsaný detailně v kapitole 5.2.5. Po provedení tohoto kroku je zpráva zašifrována standardním způsobem.

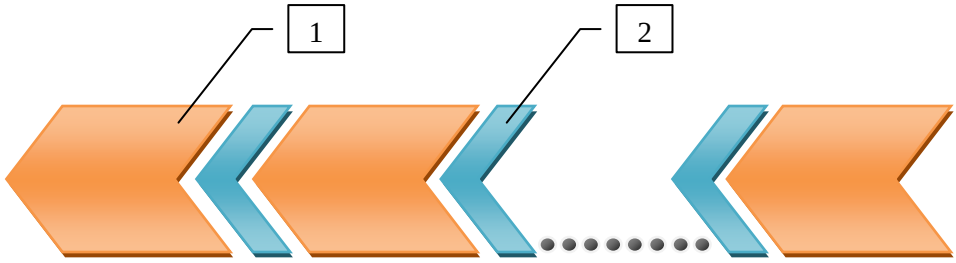
Ve druhém případě, kdy parametry fraktálu neumožňují zpracování otevřeného textu je proveden zmíněný proces vícenásobné generování fraktálu. Po zjištění této skutečnosti je vygenerován nový fraktál a poté provedeno zašifrování části zprávy na základě parametrů prvního fraktálu. Generování probíhá před zašifrováním z toho důvodu, že generovaný klíč nedosahuje konstantní délky, ale je proměnný. Za první část zprávy, jsou poté zašifrovány parametry nového fraktálu, sloužící jako nový klíč pro budoucí dešifrování dat. V dalším procesu probíhá analýza aktuálně vygenerovaného fraktálu (na Obr. 9 druhý blok shora vlevo). Pokud již parametry fraktálu umožňují zpracování otevřeného textu, proběhne zašifrování zprávy aktuálním klíčem a ukončení procesu vícenásobného generování fraktálu. V opačném případě probíhá cyklus dále do splnění tohoto kritéria. Mezi šifrovanými daty a klíči ve výstupním vektoru jsou vloženy speciální znaky oddělovače označující sektory s klíči a sektory s daty.



Obr. 9: Metoda vícenásobného generování fraktálu při procesu šifrování zprávy

Schéma zprávy po procesu vícenásobného generování popisuje Obr. 10. Výstupní vektor obsahuje opakující se části dat (1) a klíčů (2). Každá oblast dat je zašifrována pomocí jiného klíče, který definuje konstrukční parametry použitého fraktálu v každém cyklu. Princip uspořádání výstupního vektoru vychází z vývojového diagramu na Obr. 9. V první fázi je zašifrována datová část. Poté, co jsou v aktuálně použité fraktální struktuře vyčerpány veškeré prostředky pro účel zabezpečení informace, následuje fáze generování nového fraktálu. Klíč, který vzniknul vygenerováním nové fraktální struktury, byl v zašifrované podobě přidán k první části dat a následuje fáze šifrování zbytku zprávy

pomocí nově vygenerované fraktální struktury. Tyto fáze jsou opakovány do zpracování všech znaků otevřeného textu.



Obr. 10: Schéma zprávy po provedení procesu vícenásobného generování (1-data, 2-klíče)

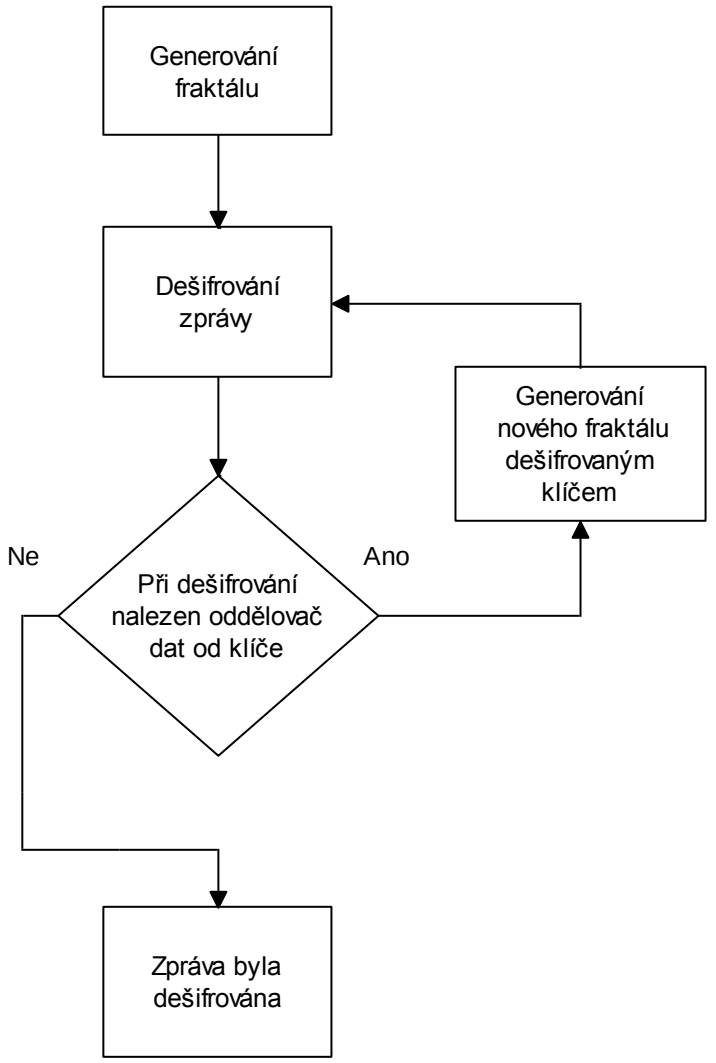
5.4.2 Proces vícenásobného generování při dešifrování zprávy

Na Obr. 11 je zobrazen vývojový diagram popisující proces vícenásobného generování fraktálu při dešifrování zprávy.

Proces dešifrování zprávy začíná generováním fraktálu algoritmem TEA na základě znalosti klíče. Detailní popis generování fraktálu je popsán v kapitole 5.2. V dalším kroku probíhá proces *dešifrování zprávy*. Ten je detailně popsán v kapitole 5.3.3. V procesu mohou nastat dva případy.

V prvním případě není ve zprávě nalezen znak oddělovače klíče od dat a zpráva je tak dešifrována v jednom kroku. Tento případ nastane, pokud nebyla při procesu zašifrování použita metoda vícenásobného generování.

V případě, kdy je v dešifrované zprávě nalezen znak oddělovače, definující oblast informací přidělené pro klíč, dochází ke generování fraktální struktury na základě informací z klíče. Proces se navrací opět k dešifrování zprávy. V případě výskytu dalších oddělovacích znaků dochází k opakování cyklu generování fraktálu. V případě, kdy se již v dešifrovaném textu nenachází znak oddělovače, je zpráva dešifrována až do konce bez nutnosti generování nové fraktální struktury.



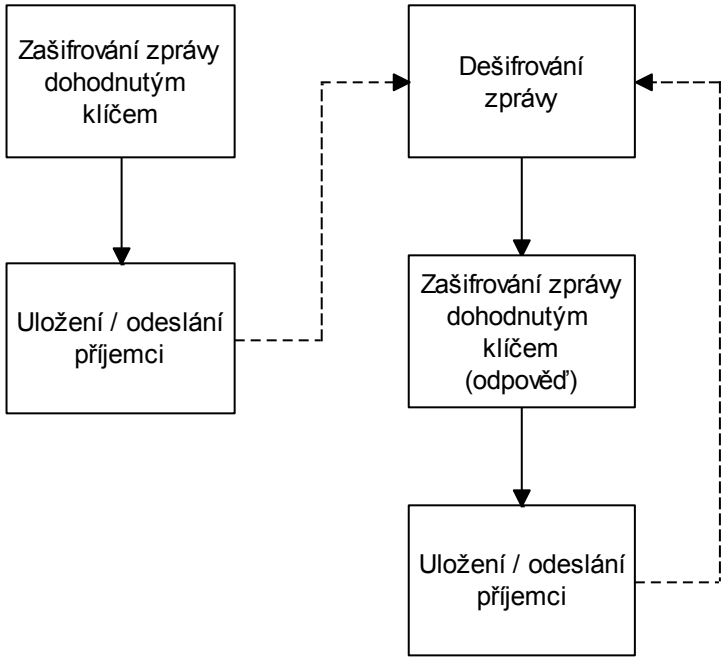
Obr. 11: Metoda vícenásobného generování fraktálu při procesu dešifrování zprávy

5.5 Proces aplikace unikátního klíče při šifrování informací

Použití procesu umožňujícího použití unikátního klíče vede ke zvýšení bezpečnosti navrženého algoritmu zašifrování informace pomocí metod fraktální geometrie. Absence tohoto procesu by neznamenala nepoužitelnost algoritmu. Způsob, kdy šifrovací systém používá delší čas stejný klíč, není ničím neobvyklým. Daná implementace však zvyšuje

odolnost algoritmu vůči kryptoanalytickým metodám. V následujících odstavcích jsou popsány a porovnány způsoby použití neunikátního a unikátního klíče pro daný algoritmus.

V případě, kdy je aplikován způsob použití neunikátního klíče, probíhá kryptografický cyklus následovně. Zpráva je zašifrována dohodnutým klíčem a uložena na paměťové médium nebo odeslána komunikačním kanálem k příjemci. Příjemce zná dohodnutý klíč a pomocí něj zprávu dešifruje. Pokud bude chtít na zprávu zareagovat nebo zahájit novou komunikaci, zašifruje zprávu dohodnutým klíčem a uloží na paměťové médium nebo ji odešle příslušným komunikačním kanálem. Celou situaci popisuje schéma na Obr. 12. Proces použití unikátního klíče a jeho rozdíly proti použití neunikátního klíče jsou popsány v dalším odstavci této kapitoly.

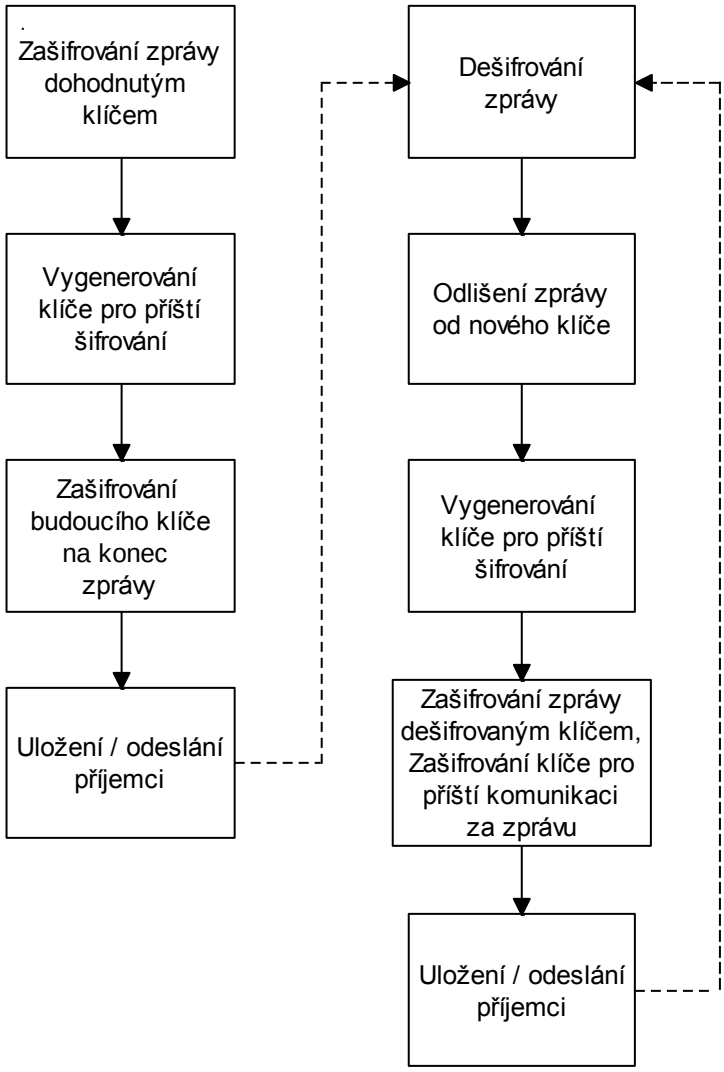


Obr. 12: Použití neunikátního klíče

V situaci, kdy je aplikován způsob použití unikátního klíče, probíhá cyklus o něco složitěji než v prvním případě. Začátek probíhá shodně jako u předešlého případu. Před zahájením prvotní komunikace je nutná dohoda o tzv. *startovním klíči*. Informace je po této

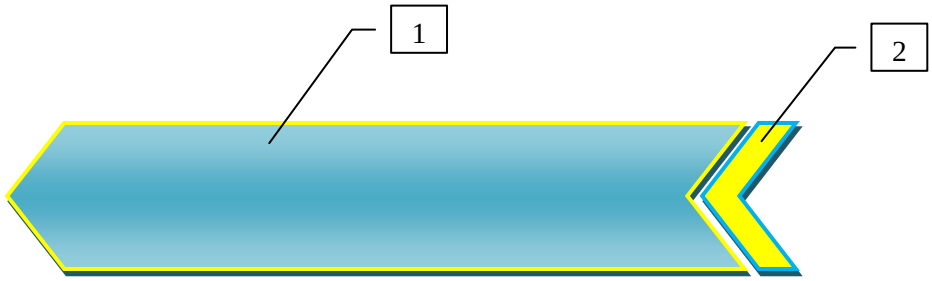
operaci zašifrována pomocí *startovního klíče*. V další části je vygenerován klíč pro příští komunikaci. Tento klíč je zašifrován původním klíčem a přiložen na konec zprávy. Mezi zprávou a klíčem je speciální znak, který tyto sektory odlišuje. Tento speciální znak je obsažen ve *Vstupní abecedě znaků*, popsané v kapitole 5.2.4. Tím je zaručeno, že nijak nevybočuje z výstupního vektoru zprávy, čímž je anonymní pro případného útočníka. Po této operaci následuje uložení na paměťové médium nebo odeslání zprávy příjemci. Zde dochází k dešifrování zprávy *startovním klíčem*. Nyní je okamžik, kdy již nebude v budoucnu použit. Jedinou výjimkou by představovalo jeho náhodné opětovné vygenerování. Vzhledem k jeho parametrům je tato pravděpodobnost mizivá. Více informací o kombinacích klíče je popsáno v kapitole 9.3. Při operaci dešifrování je odlišena datová část od části s klíčem. Tímto klíčem dojde později k zašifrování budoucí zprávy. V další části je vygenerován nový klíč pro příští komunikaci, zašifrován, přiložen ke zprávě a uložen na paměťové médium nebo odeslán komunikačním kanálem. Od tohoto kroku se celý proces opakuje. Použitím této metody je zajištěno, že každá budoucí zpráva je zašifrována jiným klíčem. Jak již bylo zmíněno, tato skutečnost zvyšuje odolnost tohoto algoritmu proti kryptoanalýze ze strany útočníka. Schéma tohoto procesu popisuje Obr. 13.

Po srovnání těchto způsobů je již z diagramů na Obr. 12 a Obr. 13. patrné, že metoda postavená na použití unikátního klíče obsahuje více operací nutných pro její úspěšnou aplikaci. Z této skutečnosti vyplývá větší časová náročnost na její průběh. Také zde narůstá délka zprávy o znak oddělovače a obsah zašifrovaného klíče na konci každé zprávy. Metoda unikátního klíče však zvyšuje bezpečnost daného řešení proti způsobu, kdy je používán k šifrování stále identický klíč. Tato vlastnost však zvyšuje robustnost řešení. Procesy jednotlivých způsobů šifrování jsou více popsány a demonstrovány v kapitole 8.



Obr. 13: Schéma procesu použití unikátního klíče pro šifrování

Na Obr. 14 je zobrazeno schéma zprávy, kterou produkuje a se kterou pracuje algoritmus unikátního klíče pro šifrování zpráv. Výstupní vektor reprezentující zprávu, obsahuje datovou část (1) a část s klíčem (2). Při použití algoritmu nezahrnujícího unikátní klíč, popsaný schématem na Obr. 12, výstupní vektor reprezentující zašifrovanou zprávu neobsahuje část s klíčem budoucí komunikace.



Obr. 14: Schéma zprávy nesoucí klíč budoucí komunikace (1-data, 2-klíč)

5.6 Typy výstupních dat

Tabulka Tab. 4 ukazuje čtyři typy výstupních dat, které je možné obdržet použitím algoritmu, popsaném v kapitolách výše tohoto oddílu.

Prvním typem výstupních dat jsou *Krátké zprávy s neunikátním klíčem*. Za takové lze označit otevřený text, kdy je jeho délka kratší, než maximální délka zprávy odvozená z použité fraktální struktury určené pro její zašifrování. Zprávy, které využívají principů neunikátního klíče, jsou takové, kdy je klíč pro jejich zašifrování použitý více než jednou.

Druhým typem výstupních dat jsou *Dlouhé zprávy s neunikátním klíčem.* Za dlouhou zprávu lze označit takovou, kdy je její délka větší, než maximální délka zprávy odvozená z použité fraktální struktury určené pro její zašifrování. Pro úspěšné zpracování zprávy tohoto typu je třeba použít více, než jednu fraktální strukturu. Počet fraktálních struktur závisí na jejich schopnosti pojmout znaky *otevřeného textu* a také znaky představující klíč pro generování využitých fraktálů. Princip neunikátního klíče je shodný jako u předešlého typu zpráv.

Třetím typem jsou *Krátké zprávy s unikátním klíčem*. Princip krátkých zpráv je shodný jako u prvního typu. Zpráva, která používá principu unikátního klíče, může být tímto klíčem zašifrována pouze jednou. Na jejím konci je zašifrován klíč pro budoucí komunikaci. Pokud by byla informace zašifrována stejným klíčem několikrát, nemohla by být klasifikována jako *zpráva s unikátním klíčem.*

Čtvrtým typem jsou *Dlouhé zprávy s unikátním klíčem.* Princip aplikace myšlenky dlouhých zpráv je shodný s druhou skupinou, princip využití unikátního klíče je shodný se třetí skupinou.

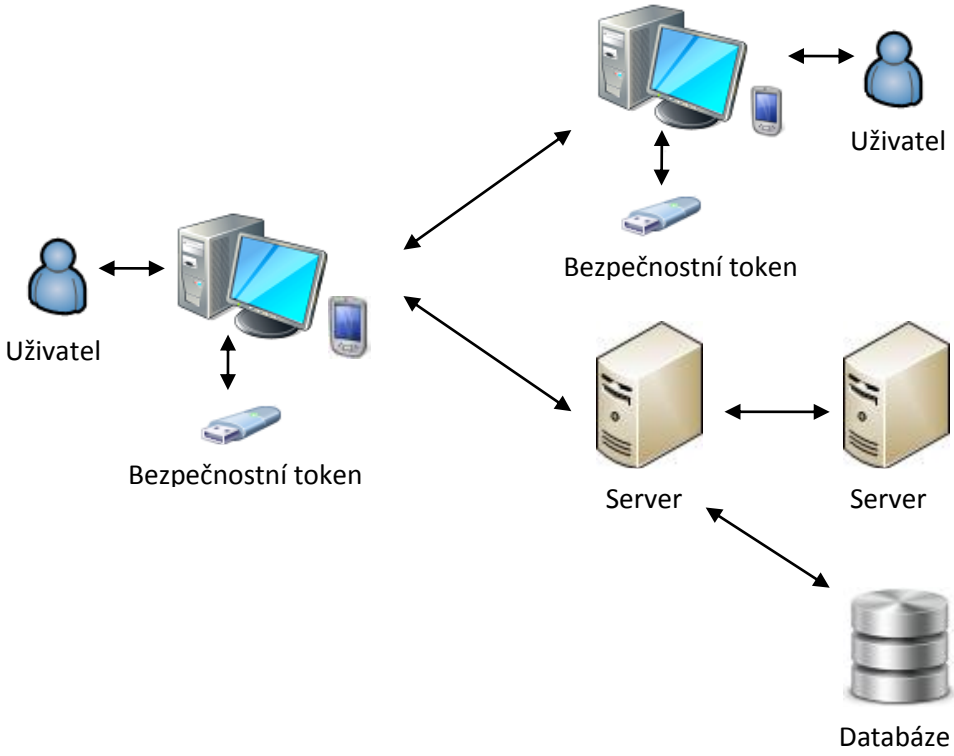
Tab. 4: Typy výstupních dat

Typ	*Vlastnost*
Krátké zprávy s neunikátním klíčem	Identický klíč, 1 fraktál
Dlouhé zprávy s neunikátním klíčem	Identický klíč, více fraktálů
Krátké zprávy s unikátním klíčem	Různé klíče, 1 fraktál
Dlouhé zprávy s unikátním klíčem	Různé klíče, více fraktálů

5.7 Návrh modelu nasazení šifrovacího algoritmu

Na Obr. 15 je zobrazeno navržené schéma znázorňující možnosti využití daného řešení v jednotlivých vazbách. Schéma se skládá z vazeb člověk – člověk, člověk – stroj *[15*] a stroj – stroj. Bezpečnost lidského faktoru lze zvýšit použitím bezpečnostních tokenů pro uložení klíče.

Jednotlivé vazby zahrnují komunikaci osob s požadavkem vysoké priority bezpečnosti, dále interakci osob s prvky informačního systému, které slouží pro získání požadovaného výsledku, výstupu, zpracování požadavků, uložení dat do databázového systému, apod. Další vazba zahrnuje vzájemnou komunikaci hardware informačních systémů, kde dochází opět k předávání informací důvěrného charakteru, podobně jako v předchozích vazbách.



Obr. 15: Schéma modelu použití navrženého řešení

6 KRYPTOANALÝZA

Tato kapitola sumarizuje statistické [46] a analytické metody [26], [27] kryptoanalýzy. Podrobnější informace získané pro účel hodnocení odolnosti navrženého algoritmu popisuje kapitola 9. Více o problematice kryptoanalýzy popisuje literatura [9], [67], [80], [93].

6.1 Statistické metody

Pro účel testování z hlediska *Statistických metod* byla použita metoda frekvenční analýzy a Kasiskiho metoda popsaná níže. Základ těchto metod je popsán v následujících podkapitolách. Více informací lze čerpat z literatury uvedené u každé z nich. Aplikace těchto metod popisují kapitoly 9.1 a 9.2.

6.1.1 Frekvenční analýza

Úloha frekvenční analýzy [93] spočívá ve stanovení četností výskytu jednotlivých znaků šifrovaného textu a využití této skutečnosti v dalším postupu při kryptoanalýze. Každý jazyk obsahuje určitou frekvenci výskytu znaků abecedy. Na základě informací o použitém jazyku, frekvenci výskytu jednotlivých znaků uvnitř šifrovaného textu a logické úvaze, lze provést v některých případech úspěšnou kryptoanalýzu.

Po provedené frekvenční analýze, kryptoanalytika v prvním případě zajímá, která písmena se vyskytují nejčastěji. Kryptoanalytik také ví, které znaky se nejvíce vyskytují v konkrétním jazyce. Proběhne substituce mezi nalezenými a odhadovanými znaky. Dále proces pokračuje podrobnější analýzou znaků v okolí nejvíce se vyskytujících znaků. Tato analýza vychází ze zákonitostí konkrétního jazyka, kdy se konkrétní znaky nachází s určitou pravděpodobností vedle jiných znaků na základě znalosti, zda se například jedná o samohlásku či souhlásku. Další analýzou a substitucemi konkrétních znaků na základě znalostí pravidel jazyka lze úspěšně dokončit kryptoanalýzu šifrovaného textu.

6.1.2 Kasiskiho metoda

Kasiskiho metoda [67] je postavena na myšlence počtu výskytu stejných řetězců v otevřeném textu a použití několika substitucí k šifrování. Tyto substituce jsou cyklicky střídány. Tato metoda říká, že řetězec bude zašifrován tolikrát, kolikrát odpovídá podíl výskytu stejných řetězců v otevřeném textu k použitému množství cyklicky se střídajících substitucí.

Proces začíná získáním četností výskytu opakujících se řetězců v zašifrovaném textu určité délky. Poté je zjištěna diference jednotlivých řetězců a určen seznam všech dělitelů z diference. Počet substitucí, které byly použity, odpovídá jednomu z nejvíce se vyskytujících dělitelů. Pomocí Kasiskiho metody je zjišťováno, z kolika šifrovacích abeced se zašifrovaný text skládá. Pokud tato metoda nezodpoví otázku počtu šifrovacích abeced, šifrovací algoritmus nebude s největší pravděpodobností realizován polyalfabetickou substitucí [46].

6.2 Útok hrubou silou

Princip útoku hrubou silou (BFA) [9] je poměrně jednoduchý. Představuje postupné zkoušení kombinací jednotlivých kombinací znaků do doby nalezení správného řešení. Tato metoda je často úspěšná pro velmi krátké klíče. U dlouhých klíčů vzrůstá jeho časová náročnost v řádu let, desetiletí, staletí i tisíciletí.

V souvislosti s útoky hrubou silou jsou velmi často používané tzv. slovníkové metody. Podobně jako u klasického útoku hrubou silou jsou cyklicky zkoušeny slova, které lidé obvykle používají jako své heslo. Tyto slova mohou být uložena v datovém souboru nebo ve specializované databázi. Pro zvýšení účinnosti lze slovníkový útok zkombinovat s klasickým útokem hrubou silou. V tomto případě jsou ke slovům ze slovníku přidávány generované znaky vhodným generátorem a výsledná kombinace je použita pro nalezení hesla.

Lze říci, že úspěšnost útoku hrubou silou závisí na délce a složení použitého klíče. U krátkých klíčů je vyšší pravděpodobnost prolomení než u delších. Dalším parametrem

úspěšnosti prolomení je také rychlost, s jakou lze jednotlivé klíče zkoušet a také rychlost rozpoznání správně nalezeného klíče.

6.3 Analytické metody

Mezi analytické metody kryptoanalýzy patří metoda *Ciphertext Only Attack* a její podskupiny, popsané níže. Základ této metody je popsán v následující kapitole. Více informací lze čerpat z uvedené literatury. Praktická aplikace je popsána v kapitole 9.4.

6.3.1 Chosen Plaintext Attack

Základem typu útoku Chosen Plaintext Attack [29] je možnost volby otevřeného textu a následná analýza šifrovaného textu. Kryptoanalytik se snaží vhodnou volbou a modifikacemi otevřeného textu analyzovat šifrový text a nalézt zákonitosti, které jej dovedou k odhalení slabých míst šifrovacího mechanismu.

6.4 Kerckhoffův předpoklad

Kerckhoffův předpoklad [56] je postaven na principu, kdy má útočník plnou znalost všech principů šifrovacího algoritmu a výjimku tvoří pouze použitý klíč.

PRAKTICKÁ ČÁST

7 GENEROVÁNÍ FRAKTÁLNÍ STRUKTURY

Generování fraktální struktury je jedním z důležitých kroků v procesu zašifrování i dešifrování informace. Teorie tohoto procesu je detailně popsána v kapitole 5 a jejích podkapitolách. Kapitola 7 popisuje použité fraktály generované algoritmem TEA [96] a dále popisuje úkony, při kterých byly hledány vhodné parametry pro jejich generování. Jak již bylo popsáno v kapitole 5.2.4 v procesu šifrování a dále zmíněno v kapitole 5.3.2 při procesu dešifrování, množství informace, kterou lze danou strukturou zašifrovat závisí na členitosti fraktální struktury. Vhodným nastavením parametrů generátoru lze docílit toho, že členitost fraktální struktury bude na vysoké úrovni. Následující podkapitoly dokumentují provedený výzkum v této oblasti. Výsledky byly dosaženy pomocí testovacího prostředí implementovaného do rozhraní programu vytvořeného za účelem výzkumu procesů generování, šifrování a dešifrování. Více informací nabízí kapitola 10. Rozhraní pro testování je detailně popsáno v kapitole 10.4.

7.1 Vhodné parametry použité pro generování fraktálních struktur

V následujících kapitolách jsou uvedeny parametry, při kterých vykazovaly jednotlivé fraktální struktury nejlepší vlastnosti pro účel zašifrování informace. Tyto parametry byly zadány do testovacího rozhraní popsaného v kapitole 10.1. Další informace o použitých fraktálech jsou uvedeny v příloze Příloha B.

7.1.1 Mandelbrotova množina

Tabulka Tab. 5 sumarizuje vhodné parametry generátoru u Mandelbrotovy množiny. Grafické porovnání výsledků je zobrazeno v grafu na obrázku Obr. 16.

Tab. 5: Vhodné parametry generátoru u Mandelbrotovy množiny

Rozlišení	200 x 200	
Typ dat	Alfanumerické	Numerické
Počet průchodů	12	10
Práh	125	50
Počet iterací	145	55
Rozsah	4	4
Průměrná rychlost generování	1,74s	0,52s
Průměrná max. délka	310	989

7.1.2 Juliovy množiny

Tabulka Tab. 6 sumarizuje vhodné parametry generátoru u Juliových množin. Grafické porovnání výsledků je zobrazeno v grafu na obrázku Obr. 16.

Tab. 6: Vhodné parametry generátoru u Juliových množin

Rozlišení	200 x 200	
Typ dat	Alfanumerické	Numerické
Počet průchodů	9	10
Práh	55	115
Počet iterací	295	315
Rozsah	4	4
Průměrná rychlost generování	1,87s	2,09s
Průměrná max. délka	281	779

7.1.3 Burning Ship

Tabulka Tab. 7 sumarizuje vhodné parametry generátoru u fraktálu Burning Ship. Grafické porovnání výsledků je zobrazeno v grafu na obrázku Obr. 16.

Tab. 7: Vhodné parametry generátoru u fraktálu Burning Ship

Rozlišení	200 x 200	
Typ dat	Alfanumerické	Numerické
Počet průchodů	15	17
Práh	130	130
Počet iterací	235	215
Rozsah	4	4
Průměrná rychlost generování	3,07s	3,18s
Průměrná max. délka	362	894

7.1.4 Bird of Prey

Tabulka Tab. 8 sumarizuje vhodné parametry generátoru u fraktálu Bird of Prey. Grafické porovnání výsledků je zobrazeno v grafu na obrázku Obr. 16.

Tab. 8: Vhodné parametry generátoru u fraktálu Bird of Prey

Rozlišení	200 x 200	
Typ dat	Alfanumerické	Numerické
Počet průchodů	16	15
Práh	130	90
Počet iterací	240	240
Rozsah	4	4
Průměrná rychlost generování	3,46s	2,72s
Průměrná max. délka	368	947

7.1.5 Water Plane

Tabulka Tab. 9 sumarizuje vhodné parametry generátoru u fraktálu Water plane. Grafické porovnání výsledků je zobrazeno v grafu na obrázku Obr. 16.

Tab. 9: Vhodné parametry generátoru u fraktálu Water plane

Rozlišení	200 x 200	
Typ dat	Alfanumerické	Numerické
Počet průchodů	19	8
Práh	90	90
Počet iterací	150	210
Rozsah	4	4
Průměrná rychlost generování	6,75s	3,63s
Průměrná max. délka	203	334

7.1.6 4th Degree Multibrot

Tabulka Tab. 10 sumarizuje vhodné parametry generátoru u fraktálu 4th Degree Multibrot. Grafické porovnání výsledků je zobrazeno v grafu na obrázku Obr. 16.

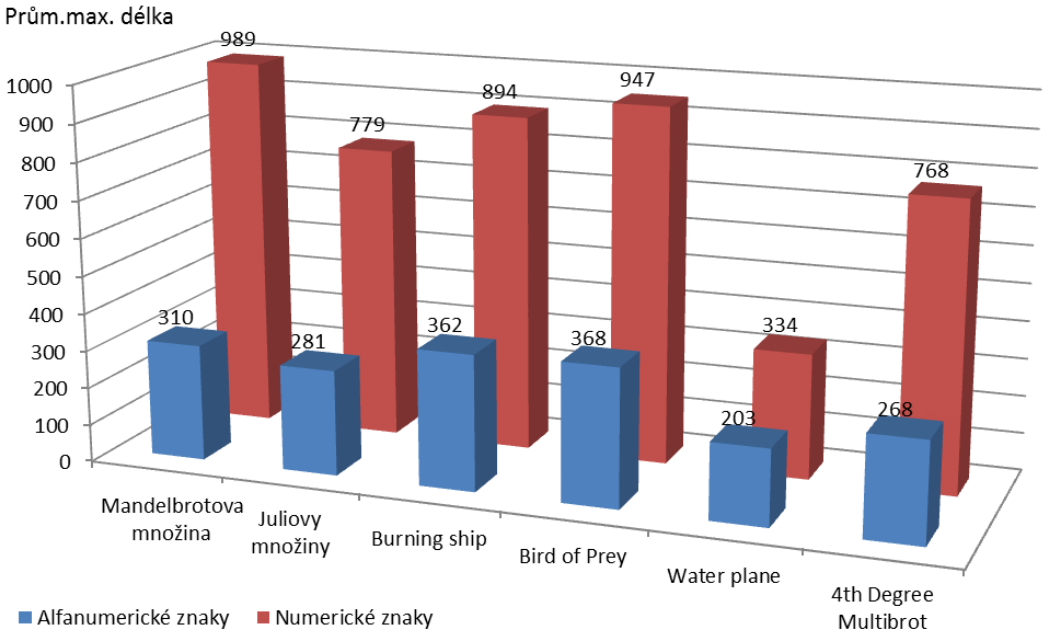
Tab. 10: Vhodné parametry generátoru u fraktálu 4th Degree Multibrot

Rozlišení	200 x 200	
Typ dat	Alfanumerické	Numerické
Počet průchodů	18	15
Práh	115	115
Počet iterací	135	295
Rozsah	4	4
Průměrná rychlost generování	3,52s	5,91s
Průměrná max. délka	268	768

7.2 Porovnání jednotlivých fraktálních struktur

V kapitole 7.1 byly určeny vhodné parametry pro generování fraktálních struktur. Rozhraní, které bylo k tomuto procesu použito, je popsáno v kapitole 10 a dále rozvinuto v podkapitolách 10.1 a 10.4. Hodnoty byly určeny po provedení deseti měření. Tento parametr byl nastaven na rozhraní na Obr. 37 nad tlačítkem 13. Získané parametry jsou sumarizovány v tabulkách *Tab. 5, Tab. 6, Tab. 7, Tab. 8, Tab. 9* a *Tab. 10*. Na Obr. 16 jsou provedena tato srovnání graficky. Modré charakteristiky značí alfanumerickou znakovou sadu, červené numerickou znakovou sadu. Rozlišení fraktálu bylo u všech typů 200 x 200 bodů. Vyšší hodnoty výrazně prodlužovaly proces generování, při použití výpočetní techniky popsané v kapitole 11.1.

Nejlepších parametrů v alfanumerické části dosahovaly fraktály Bird of Prey, Burning Ship a Mandelbrotova množina. V numerické části se jednalo o fraktály Mandelbrotova množina, Bird of Prey a Burning Ship. Se vzrůstajícím počtem průchodů vzrůstala i časová náročnost generování fraktálních struktur. V kapitole 9.4.2 lze porovnat odolnost klíče v závislosti na zvoleném počtu průchodů při procesu generování.



Obr. 16: Srovnání průměrných max. délek zpráv u jednotlivých fraktálů

8 PROCES ŠIFROVÁNÍ A DEŠIFROVÁNÍ

V následujících podkapitolách jsou demonstrovány šifrovací a dešifrovací procesy pro navržené kategorie zpráv. V závěru kapitoly je provedeno shrnutí získaných poznatků.

8.1 Šifrování jednotlivých kategorií zpráv

Kapitola nazvaná *Šifrování jednotlivých kategorií zpráv*, pojednává o šifrování dat, členěných z hledisek principu a specifikace popsaných v kapitole 5.6.

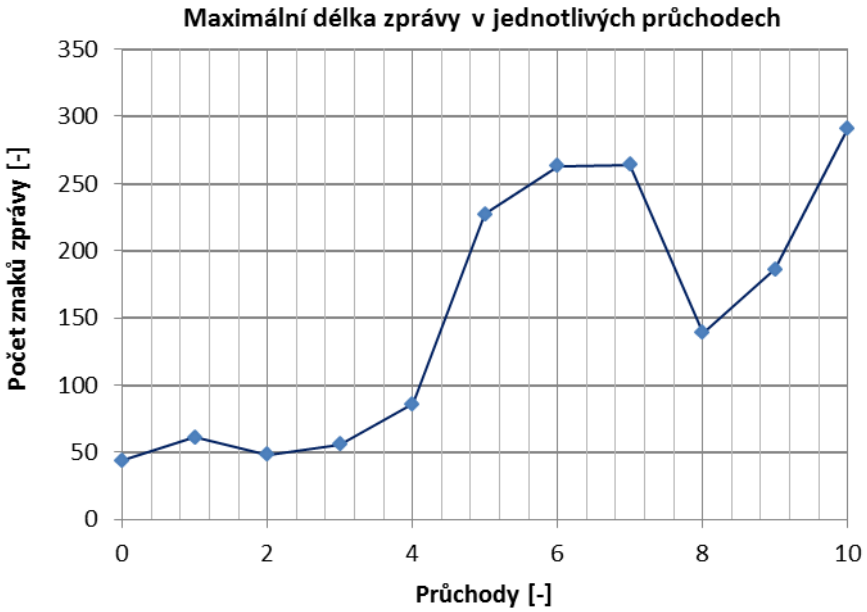
8.1.1 Krátké zprávy s neunikátním klíčem

V této části bylo provedeno a vyhodnoceno šifrování krátké zprávy s neunikátním klíčem. Tabulka Tab. 11 popisuje parametry sloužící jako klíč pro vygenerování fraktální struktury. V daném případě byly pro zašifrování zprávy použity Juliovy množiny s parametry popsanými v tabulce. Další informace o použitých fraktálech jsou uvedeny v kapitole 7 a v příloze Příloha B. U fraktálu Juliových množin se v jeho klíči nachází ještě konstanta C. Daný fraktál je zkonstruován s parametry rozlišení 200 x 200 bodů.

Tab. 11: Parametry algoritmu

Typ použitého fraktálu	Juliovy množiny (200x200)
x parametr	-1,40375 (Cx = -0,771)
y parametr	0,0753125 (Cy = 0,115)
Rozsah	0,00390625
Počet iterací	250
Max. délka zprávy	291

Graf na Obr. 17 ukazuje průběh hodnot *maximální délky vstupní informace* během generování fraktální struktury použité v procesu zašifrování krátké zprávy s neunikátním klíčem. Parametry pro konstrukci tohoto fraktálu jsou uvedeny v tabulce Tab. 11.



Obr. 17: Zobrazení maximální délky zprávy v jednotlivých průchodech

V tabulce Tab. 12 jsou zobrazeny vstupní a výstupní data daného algoritmu. Znaky, které mohou obsahovat vstupní data, jsou popsány v kapitole 5.2.8. Výstupní data reprezentují vektor hodnot představující zašifrovaný text.

Tab. 12: Transformace textového řetězce zprávy do zašifrovaného tvaru

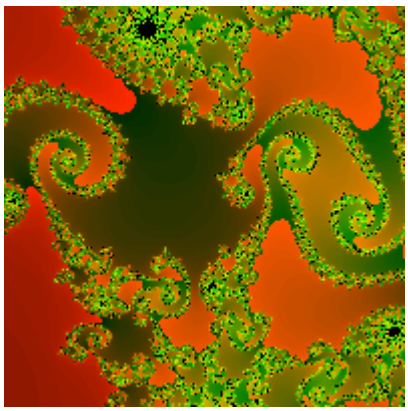
Vstupní data
informatika
Výstupní data
33, 141, 105, 62, 39, 179, 30, 118, 116, 114, 61, 73, 0, 186, 51, 74, 49, 159, 99, 62, 7, 169

Tabulka Tab. 13 popisuje strukturu vstupních dat šifrovacího algoritmu. Vstupní informace otevřeného textu je obsažena od počátku do pozice 22. V tomto případě se jedná o proces šifrování krátké zprávy s neunikátním klíčem, tudíž za blokem vstupních informačních dat nenásleduje znak oddělovače pro odlišení informačních dat od klíče. Opačná situace je patrná u jiných typů zpráv v dalších podkapitolách.

Tab. 13: Analýza a složení vstupních dat krátké zprávy s neunikátním klíčem

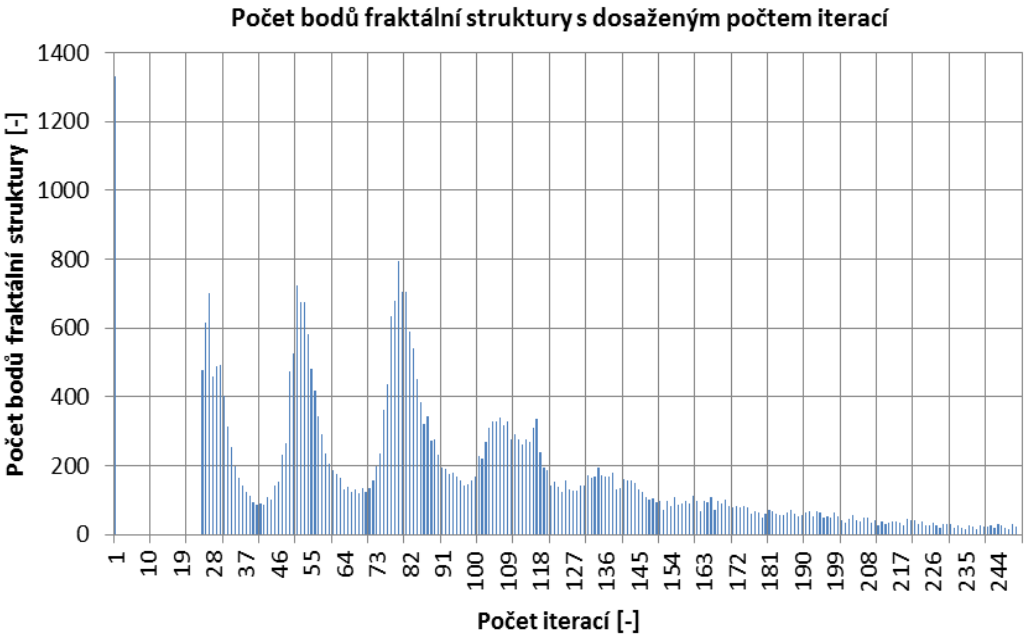
Pozice	*Délka*	*Obsah*
0	22	vstupní informace – otevřený text

Na Obr. 18 je zobrazena použitá fraktální struktura, vygenerovaná pomocí parametrů v z tabulky Tab. 11. Jedná se o Juliovu množinu vytvořenou algoritmem TEA [96].



Obr. 18: Použitá fraktální struktura

Graf na Obr. 19 zobrazuje charakteristiku počtu bodů, které dosáhly příslušného počtu iterací při generování fraktálu v posledním průchodu. Na základě těchto dat byla stanovena maximální možná délka vstupní informace na 291 alfanumerických znaků. Tento graf odpovídá fraktální struktuře na Obr. 18. Parametry konstrukce této struktury jsou popsány v tabulce Tab. 11.



Obr. 19: Počet bodů fraktální struktury s dosaženým počtem iterací

8.1.2 Dlouhé zprávy s neunikátním klíčem

V této části bylo provedeno a vyhodnoceno šifrování dlouhé zprávy s neunikátním klíčem. Tabulky Tab. 14 a Tab. 15 popisují parametry sloužící jako klíč pro vygenerování fraktálních struktur. V daném případě byla pro zašifrování zprávy použita dvojice fraktálů typu Mandelbrotova množina. Parametry jejího generování jsou popsány v tabulce. Fraktál byl konstruován s rozlišením 200 x 200 bodů.

Tab. 14: Parametry algoritmu – první fraktál

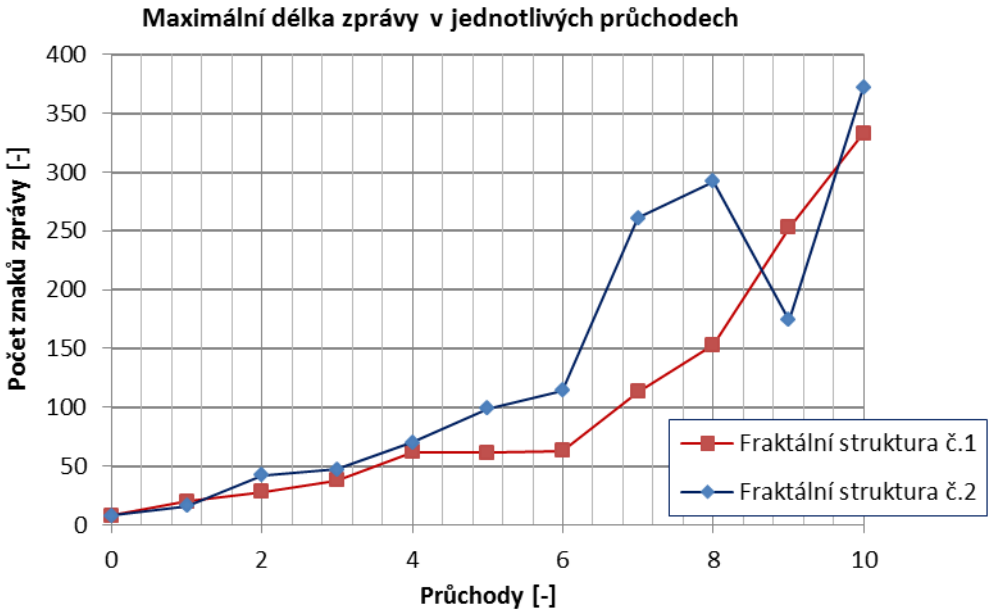
Typ použitého fraktálu	Mandelbrotova množina (200x200)

x parametr	-0,0478515625
y parametr	0,67578125
Rozsah	0,00390625
Počet iterací	250
Max. délka zprávy	333

Tab. 15: Parametry algoritmu – druhý fraktál

Typ použitého fraktálu	Mandelbrotova množina (200x200)
x parametr	-1,2232421875
y parametr	-0,1548828125
Rozsah	0,00390625
Počet iterací	250
Max. délka zprávy	372

Graf na Obr. 20 ukazuje průběh hodnot *maximálních délek vstupní informace* během generování fraktálních struktur použitých v procesu zašifrování dlouhé zprávy s neunikátním klíčem. Červená charakteristika představuje první použitý fraktál, modrá fraktál použitý jako druhý. Parametry pro konstrukci těchto fraktálů jsou uvedeny v tabulkách Tab. 14 a Tab. 15.



Obr. 20: Zobrazení maximální délky zprávy v jednotlivých průchodech při generování dvou fraktálních struktur

V tabulce Tab. 16 jsou zobrazeny vstupní a výstupní data daného algoritmu. Znaky, které mohou obsahovat vstupní data, jsou popsány v kapitole 5.2.8. Výstupní data reprezentují vektor hodnot představující zašifrovaný text.

Tab. 16: Transformace textového řetězce zprávy do zašifrovaného tvaru

Vstupní data
po uspesnem provedeni operaci vygenerovani fraktalu a jeho analyzy lze pristoupit k procesu zabezpeceni informace vstupni informace je prectena z textoveho pole na karte s nazvem zasifrovat zpravu, nachazejici se na hlavnim okne programu z vygenerovaneho fraktalu je vypocitana maximalni mozna delka vstupni informace a stanoveno indexove pole
Výstupní data
23, 177, 165, 85, 105, 149, 112, 60, 162, 75, 189, 20, 125, 170, 20, 167, 173, 96, 125, 171, 101, 158, 196, 5, 49, 168, 113, 34, 187, 72, 97, 61, 144, 193, 139, 174, 125, 172, 14, 185, 117, 154, 98, 117, 148, 138, 161, 83, 166, 108, 30, 179, 190, 144, 193, 108, 74, 196, 7, 101, 152, 117, 91, 87, 125, 147, 125, 173, 1, 191, 125, 174, 162, 78, 59, 166, 115, 144, 190, 145, 190, 98, 97, 165, 19, 96, 114, 179, 94, 148, 190, 146, 44, 199, 40, 174, 190, 147, 110, 163, 49, 164, 55, 85, 191, 140, 78, 58, 160, 102, 125, 175, 86, 183, 53, 167, 147, 55, 190, 148, 154, 110, 190, 149, 33, 198, 128, 68, 163, 62, 75, 166, 158, 102, 100, 158, 117, 88, 125, 176, 115, 149, 164, 83, 149, 129, 74, 197, 77, 166, 45, 174, 150, 129, 42, 174, 116, 144, 183, 23, 77, 172, 198, 97, 69, 180, 194, 77, 138, 135, 11, 170, 51, 169, 157, 159, 181, 107, 118, 36, 15, 167, 21, 96, 194, 94, 193, 133, 171, 138, 187, 105, 182, 92, 1, 180, 125, 177, 157, 160, 126, 165, 188, 46, 76, 188, 101, 150, 124, 141, 17, 184, 165, 106, 89, 155, 87, 156, 16, 199, 190, 150, 157, 161, 126, 166, 65, 162, 164, 66, 24, 169, 180, 55, 182, 99, 108, 159, 36, 186, 97, 166, 58, 20, 125, 140, 179, 17, 114, 180, 135, 131, 130, 122, 90, 156, 190, 151, 157, 162, 126, 167, 34, 94, 150, 142, 126, 168, 83, 156, 143, 140, 182, 16, 171, 106, 188, 188, 104, 46, 126, 169, 119, 139, 190, 152, 10, 113, 24, 161, 70, 128, 157, 80, 138, 151, 62, 156, 102, 160, 167, 13, 46, 166, 126, 170, 199, 73, 180, 102, 63, 23, 121, 139, 50, 169, 86, 160, 126, 171, 171, 87, 198, 22, 190, 153, 45, 132, 167, 93, 190, 157, 152, 120, 158, 73, 126, 172, 127, 78, 23, 168, 47, 132, 125, 133, 190, 154, 199, 103, 85, 154, 175, 106, 124, 137, 111, 135, 150, 135, 191, 141, 151, 119, 94, 166, 114, 181, 117, 30, 8, 181, 80, 160, 198, 122, 24, 166, 121, 132, 85, 153, 80, 166, 17, 168, 190, 155, 129, 118, 198, 71, 120, 137, 23, 96, 68, 174, 199, 129, 157, 163, 159, 112, 191, 142, 65, 153, 126, 173, 78, 178, 126, 139, 157, 164, 74, 198, 171, 94, 131, 120, 126, 174, 190, 21, 61, 169, 191, 143, 56, 17, 125, 142, 59, 176, 190, 156, 148, 129, 111, 163, 162, 99, 175, 29, 25, 126, 61, 167, 69, 181, 179, 35, 126, 175, 79, 120, 163, 83, 181, 100, 76, 175, 98, 191, 185, 67, 190, 158, 45, 179, 7, 171, 93, 112, 83, 91, 57, 124, 151, 121, 84, 77, 191, 52, 191, 31, 172, 15, 156, 141, 72, 175, 91, 153, 97, 62, 190, 159, 186, 70, 126, 176, 165, 98, 117, 142, 188, 14, 156, 130, 123, 137, 190, 160, 45, 195, 8, 169, 190, 161, 79, 170, 51, 165, 31, 96, 117, 152, 188, 105, 11, 98, 169, 69, 164, 61, 7, 176, 169, 90, 157, 165, 123, 142, 187, 89, 191, 144, 196, 90, 191, 145, 184, 48, 125, 134, 191, 146, 53, 165, 75, 191, 101, 159, 191, 147, 138, 136, 127, 131, 78, 183, 48, 132, 145, 139, 160, 22, 47, 134, 182, 63, 185, 83, 58, 98, 176, 40, 44, 172, 101, 106, 83, 114, 56, 104, 23, 162, 59, 89, 149, 53, 140, 104, 173, 31, 120, 109, 174, 72, 104, 149, 46, 165, 77, 120, 150, 125, 40, 95, 105, 150, 106, 97, 78, 124, 102, 101, 23, 161, 111, 136, 2, 158, 90, 104, 70, 129, 108, 95, 141, 108, 191, 75, 141, 126, 103, 142, 163, 24, 89, 60, 44, 100, 103, 158, 89, 110, 144, 105, 146, 129, 65, 165, 93, 66, 82, 27, 79, 48, 8, 144, 108, 112, 24, 96, 8, 124, 10, 129, 49, 137, 12, 154, 108, 140, 54, 130, 70, 89, 111, 56, 25, 185, 76, 42, 79, 49, 37, 95, 130, 162, 36, 103, 40, 160, 26, 59, 6, 198, 115, 60, 60, 96, 12, 155, 24, 157, 15, 121, 109, 129, 12, 156, 145, 72, 35, 184, 79, 98, 5, 139, 92, 67, 24, 174, 42, 97, 0, 72, 86, 65, 81, 91, 113, 145, 17, 132, 3, 199, 29, 143, 33, 138, 17, 199, 76, 44, 29, 124, 1, 72, 31, 123, 96, 63, 15, 126, 45, 136, 25, 166

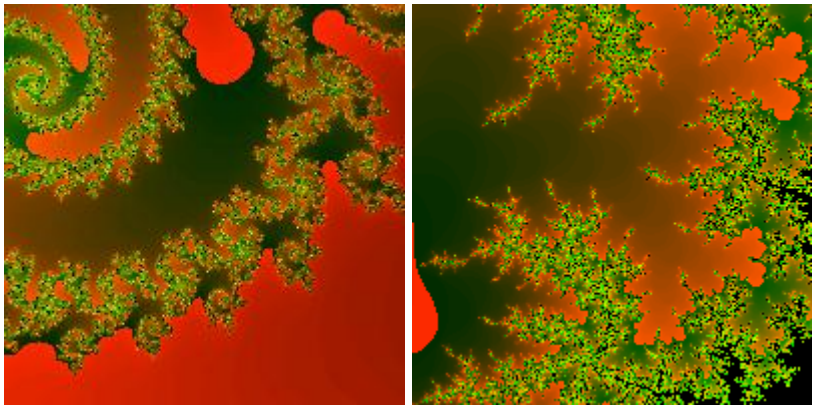
Tabulka Tab. 17 popisuje detailní strukturu vstupních dat šifrovacího algoritmu.

Vstupní informace vyplňuje počátek datového souboru do pozice 289. Na pozici 289 jsou informační data přerušena znakem oddělovače. Znak oddělovače má tvar černého čtverečku a reprezentuje jej ASCII znak č. 254. Protože se jedná o typ dlouhé zprávy, následují po tomto znaku parametry klíče pro generování dalšího fraktálu. Jak je uvedeno v tabulce Tab. 14, tato fraktální struktura pojme celkem 333 znaků. Proto je nutné vygenerovat fraktál nový, zašifrovat jeho klíč k první části dat a použít jeho strukturu k dokončení veškerých operací algoritmu. Parametry jeho klíče se nachází na pozicích 290, 304, 318 a 329. Na pozici 333 se nachází zbytek otevřeného textu o délce 54 znaků. Parametry druhého fraktálu jsou uvedeny v tabulce a Tab. 15.

Tab. 17: Analýza a složení vstupních dat dlouhé zprávy s neunikátním klíčem

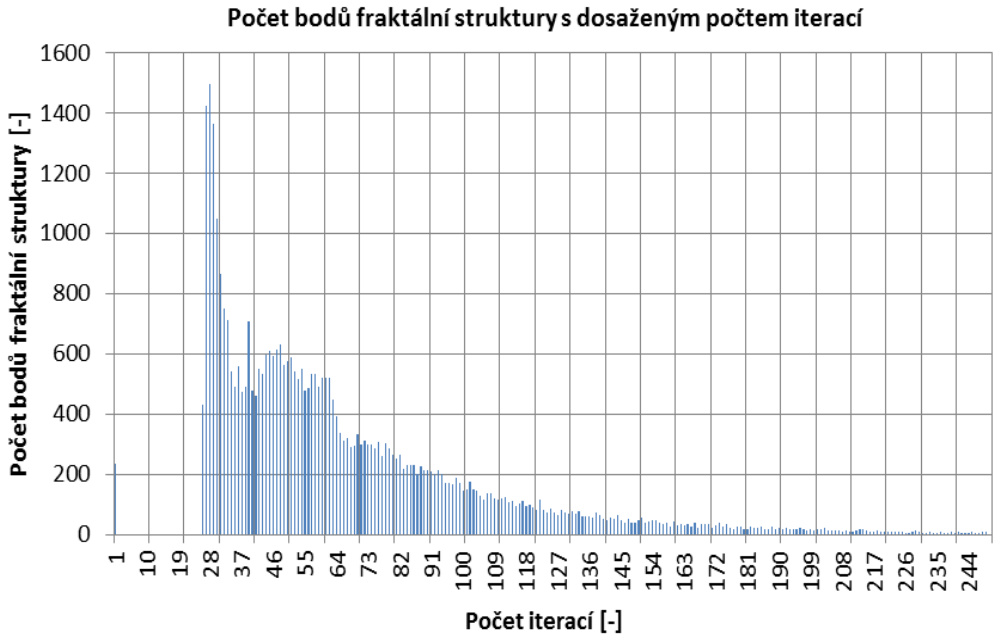
Pozice	*Délka*	*Obsah*
0	289	vstupní informace – otevřený text
289	1	znak oddělovače klíče *(■)*
290	13	klíč (*x parametr: -1,2232421875*)
303	1	znak oddělovače klíče *(■)*
304	13	klíč (*y parametr: -0,1548828125*)
317	1	znak oddělovače klíče *(■)*
318	10	klíč (*Rozsah: 0,00390625*)
328	1	znak oddělovače klíče *(■)*
329	3	klíč (*Počet iterací: 250*)
332	1	znak oddělovače klíče *(■)*
333	54	vstupní informace – otevřený text

Na Obr. 21 jsou zobrazeny použité fraktální struktury, vygenerované pomocí parametrů v z tabulky Tab. 14 a Tab. 15. Jedná se o dvojici fraktálů – Mandelbrotových množin vytvořený algoritmem TEA [96]. Jako první byl využit fraktál v levé části, pro druhou část zprávy byl použit fraktál v pravé části.



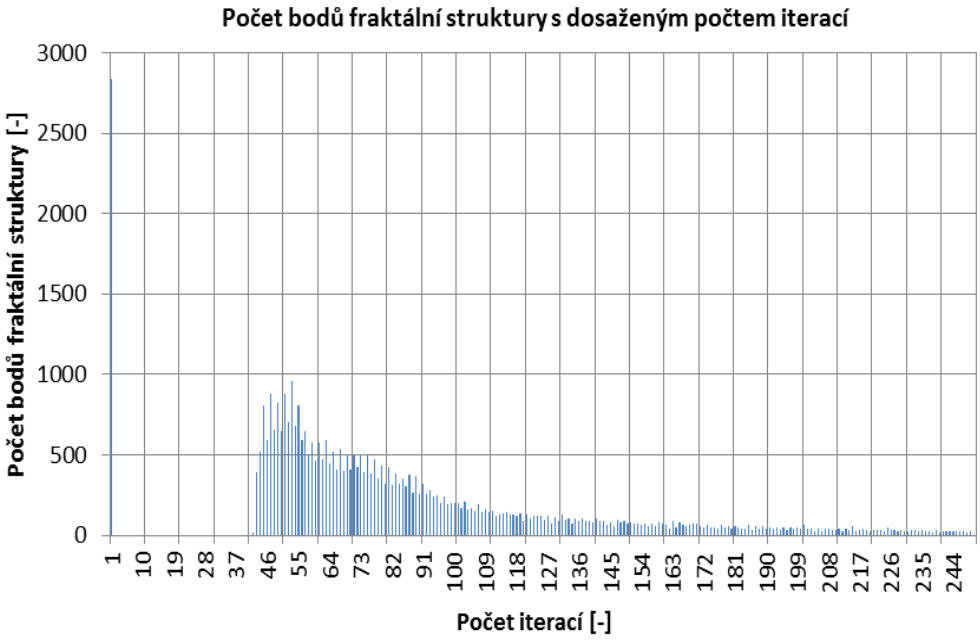
Obr. 21: Použité fraktální struktury

Graf na Obr. 22 zobrazuje charakteristiku počtu bodů, které dosáhly příslušný počet iterací při generování fraktálu v posledním průchodu. Tento fraktál byl použit jako první při zašifrování informace. Na základě zobrazených dat byla stanovena maximální možná délka vstupní informace 333 alfanumerických znaků. Tento graf odpovídá levé fraktální struktuře na Obr. 21. Parametry konstrukce této struktury jsou popsány v tabulce Tab. 14.



Obr. 22: Počet bodů fraktální struktury č. 1 s dosaženým počtem iterací

Graf na Obr. 23 zobrazuje charakteristiku počtu bodů, které dosáhly příslušný počet iterací při generování fraktálu v posledním průchodu. Tento fraktál byl použit jako druhý při procesu zašifrování informace. Na základě zobrazených dat byla stanovena maximální možná délka vstupní informace na 372 alfanumerických znaků. Tento graf odpovídá pravé fraktální struktuře na Obr. 21. Parametry konstrukce této struktury jsou popsány v tabulce Tab. 15.



Obr. 23: Počet bodů fraktální struktury č. 2 s dosaženým počtem iterací

8.1.3 Krátké zprávy s unikátním klíčem

V této části bylo provedeno a vyhodnoceno šifrování krátké zprávy s unikátním klíčem. Tabulka Tab. 18 popisuje parametry sloužící jako klíč pro vygenerování fraktální struktury. Tabulka Tab. 19 popisuje parametry klíče pro vygenerování fraktálu pro budoucí komunikaci a zajištění režimu unikátního klíče. V daném případě byla použita pro zašifrování zprávy dvojice Mandelbrotových množin. Použité fraktály byly konstruovány s rozlišením 200 x 200 bodů.

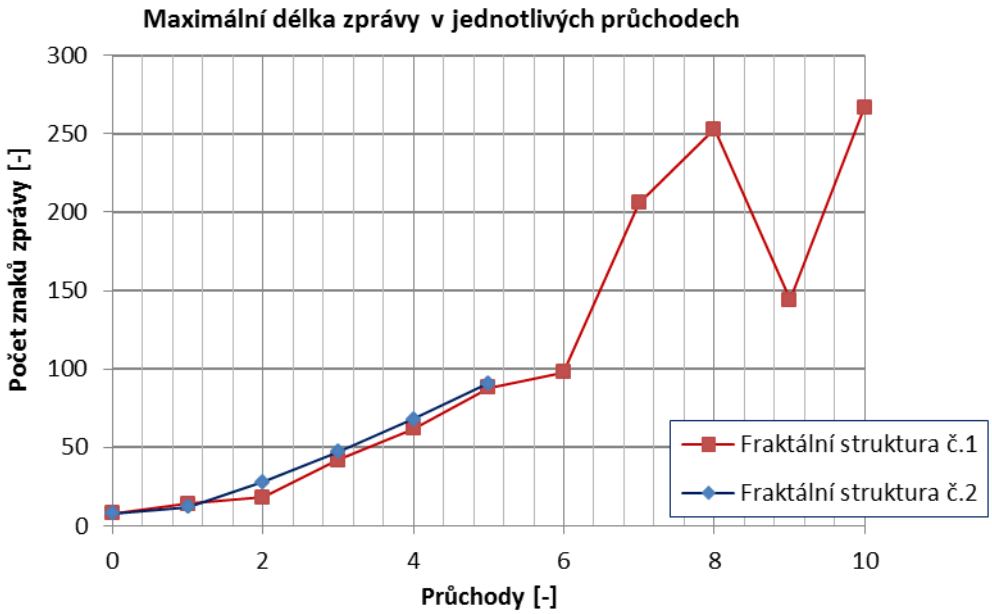
Tab. 18: Parametry algoritmu – první fraktál

Typ použitého fraktálu	Mandelbrotova množina (200x200)
x parametr	0,112265625
y parametr	-0,6311328125
Rozsah	0,00390625
Počet iterací	250
Max. délka zprávy	267

Tab. 19: Parametry algoritmu – budoucí fraktál

Typ použitého fraktálu	Mandelbrotova množina (200x200)
x parametr	-0,255
y parametr	-0,65125
Rozsah	0,125
Počet iterací	250
Max. délka zprávy	91

Graf na Obr. 24 ukazuje průběh hodnot *maximálních délek vstupní informace* během generování fraktálních struktur použitých v procesu zašifrování krátké zprávy s unikátním klíčem. Červená charakteristika představuje první použitý fraktál, modrá fraktál použitý jako druhý. Parametry pro konstrukci těchto fraktálů jsou uvedeny v tabulkách Tab. 18 a Tab. 19. Jak je z grafu patrné, první fraktální struktura byla generována s deseti průchody, pro druhou bylo použito průchodů pět. Tato druhá struktura je též označována jako budoucí. Ve zprávě není určena pro nesení informace, slouží zde pouze jako klíč pro budoucí komunikaci.



Obr. 24: Zobrazení maximální délky zprávy v jednotlivých průchodech při generování dvou fraktálních struktur

V tabulce Tab. 20 jsou zobrazeny vstupní a výstupní data použité pro algoritmus pracující s krátkými zprávami s unikátním klíčem. Znaky, které mohou obsahovat vstupní data, jsou popsány v kapitole 5.2.8. Výstupní data reprezentují vektor hodnot představující zašifrovaný text.

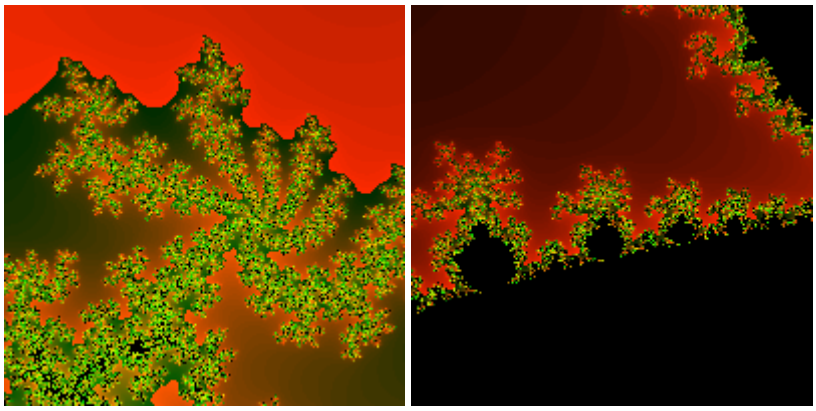
Tab. 20: Transformace textového řetězce zprávy do zašifrovaného tvaru

Vstupní data
univerzita
Výstupní data
180, 94, 65, 106, 138, 77, 163, 35, 68, 56, 6, 132, 81, 105, 53, 79, 177, 95, 3, 10, 187, 59, 173, 103, 154, 98, 176, 41, 74, 105, 111, 115, 74, 100, 101, 88, 18, 77, 122, 106, 137, 18, 161, 52, 62, 123, 122, 105, 169, 103, 145, 109, 93, 84, 144, 14, 55, 128, 111, 113, 163, 39, 128, 95, 168, 108, 80, 114, 139, 21, 73, 117

Tabulka Tab. 21 popisuje strukturu vstupních dat šifrovacího algoritmu pro aplikaci na krátký typ zprávy s unikátním klíčem. Ze struktury je patrné, že je použita dvojice fraktálů. Tyto fraktály jsou zobrazeny na Obr. 25. První z nich je fraktál použitý pro zašifrování zprávy. Druhý představuje fraktál určený pro budoucí komunikaci a zajištění režimu unikátního klíče. Datový soubor obsahující vstupní informaci je obsažen od počátku do desáté pozice. Za desátou pozicí se vyskytuje znak oddělovače, určující sektor pro klíč. Jednotlivé parametry klíče jsou vždy odděleny znakem oddělovače na pozicích 10, 17, 26 a 32. Tento klíč představuje informace nutné pro vygenerování fraktálu použitého v budoucí komunikaci. Parametry tohoto fraktálu jsou obsaženy také v tabulce Tab. 19.

Tab. 21: Analýza a složení vstupních dat krátké zprávy s unikátním klíčem

Pozice	*Délka*	*Obsah*
0	10	vstupní informace – otevřený text
10	1	znak oddělovače klíče *(■)*
11	6	klíč (*x parametr: -0,255*)
17	1	znak oddělovače klíče *(■)*
18	8	klíč (*y parametr: -0,65125*)
26	1	znak oddělovače klíče *(■)*
27	5	klíč (*Rozsah: 0,125*)
32	1	znak oddělovače klíče *(■)*
33	3	klíč (*Počet iterací: 250*)

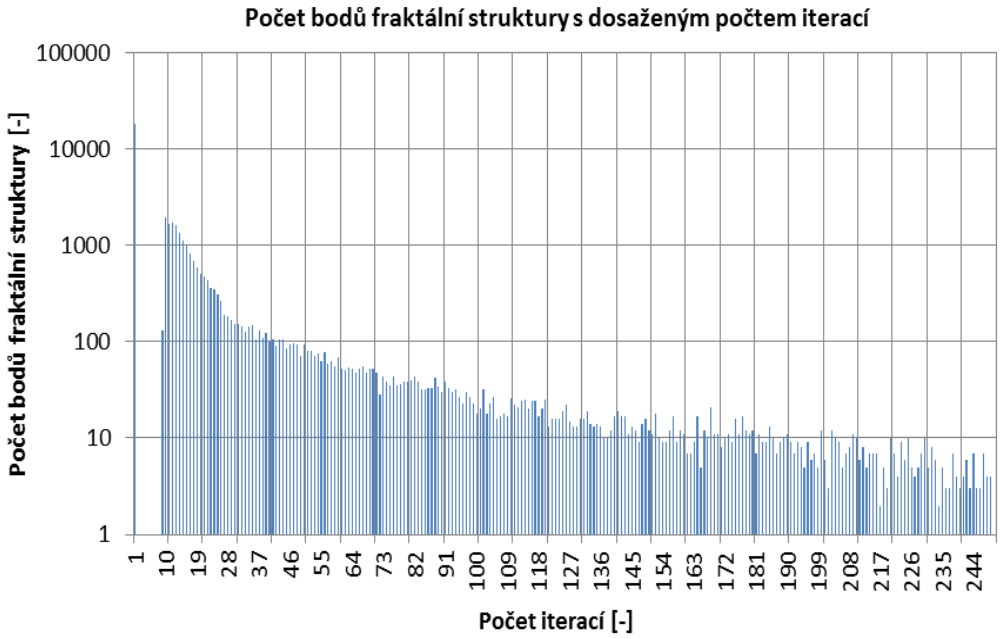


Obr. 25: Použité fraktální struktury (vlevo – aktuálně použitá, vpravo - budoucí)

Graf na Obr. 26 zobrazuje charakteristiku počtu bodů, které dosáhly příslušný počet iterací při generování fraktálu v posledním průchodu. Tento fraktál byl použit při procesu zašifrování informace. Na základě zobrazených dat byla stanovena maximální možná délka vstupní informace na 267 alfanumerických znaků. Tento graf odpovídá levé fraktální struktuře na Obr. 25. Parametry konstrukce této struktury jsou popsány v tabulce Tab. 18.

Obr. 26: Počet bodů fraktální struktury č. 1 s dosaženým počtem iterací

Graf na Obr. 27 zobrazuje charakteristiku počtu bodů, které dosáhly příslušný počet iterací při generování fraktálu v posledním průchodu. Tento fraktál byl použit při procesu zajištění unikátnosti klíče pro budoucí komunikaci. Tato fraktální struktura bude použita v budoucí komunikaci s příjemcem. Na základě zobrazených dat lze stanovit maximální možnou délku vstupní informace na 91 alfanumerických znaků. Znaky, které mohou obsahovat vstupní data, jsou popsány v kapitole 5.2.8. Tento graf odpovídá fraktální struktuře na Obr. 25 vpravo. Parametry konstrukce této struktury jsou popsány v tabulce Tab. 19. Pro větší názornost byla vertikální osa grafu zobrazena v logaritmickém měřítku.



Obr. 27: Počet bodů fraktální struktury č. 2 s dosaženým počtem iterací

8.1.4 Dlouhé zprávy s unikátním klíčem

V této části bylo provedeno a vyhodnoceno šifrování dlouhé zprávy s unikátním klíčem. Tabulka Tab. 22 popisuje parametry sloužící jako klíč pro vygenerování úvodní fraktální struktury. Tabulka Tab. 23 popisuje parametry klíče pro vygenerování druhého fraktálu pro zašifrování druhé části zprávy. Tabulka Tab. 24 obsahuje parametry vygenerování fraktální struktury pro budoucí komunikaci a zajištění tak funkce unikátního klíče. V daném případě byla použita pro zašifrování zprávy dvojice Mandelbrotových množin. Použité fraktály byly konstruovány s rozlišením 200 x 200 bodů. Maximální délka počtu zpracovatelných alfanumerických znaků vstupních dat u prvního vygenerovaného fraktálu byla 322. U druhého 328. U fraktálu určeného pro budoucí komunikaci byla tato délka určena na 265 znaků.

Tab. 22: Parametry algoritmu – první fraktál

Typ použitého fraktálu	Mandelbrotova množina (200x200)
x parametr	-0,380625
y parametr	-0,6544921875
Rozsah	0,00390625
Počet iterací	350
Max. délka zprávy	352

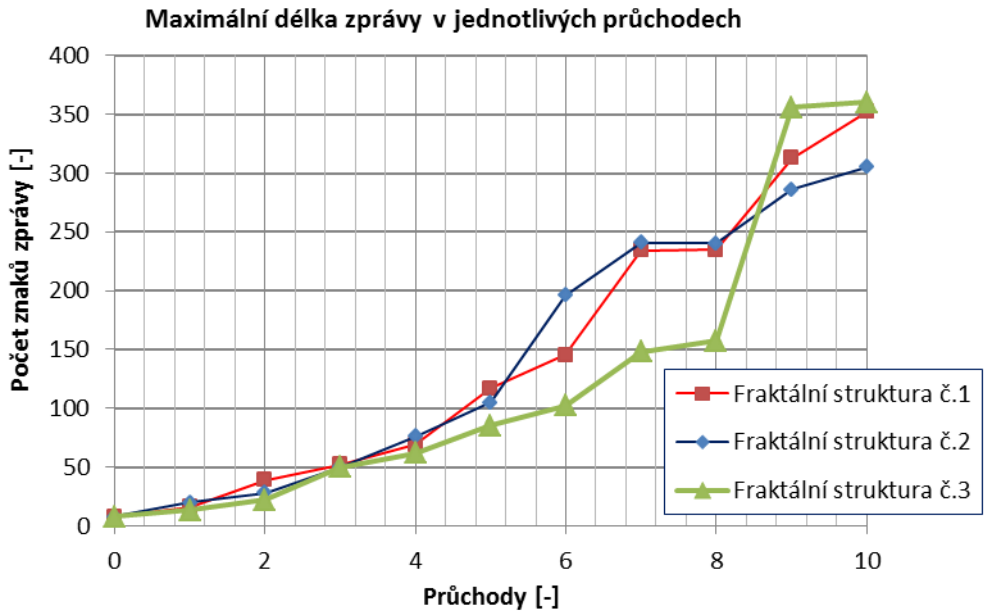
Tab. 23: Parametry algoritmu – druhý fraktál

Typ použitého fraktálu	Mandelbrotova množina (200x200)
x parametr	-0,54609375
y parametr	-0,5787109375
Rozsah	0,00390625
Počet iterací	250
Max. délka zprávy	305

Tab. 24: Parametry algoritmu – budoucí fraktál

Typ použitého fraktálu	Mandelbrotova množina (200x200)
x parametr	0,0733984375
y parametr	0,6482421875
Rozsah	0,00390625
Počet iterací	600
Max. délka zprávy	360

Graf na Obr. 28 ukazuje průběh hodnot *maximálních délek vstupní informace* během generování fraktálních struktur použitých v procesu zašifrování dlouhé zprávy s unikátním klíčem. Červená charakteristika představuje první použitý fraktál, modrá fraktál použitý jako druhý. Třetí charakteristika, označena zeleně, reprezentuje fraktál určený pro budoucí komunikaci. Parametry pro konstrukci těchto fraktálů jsou uvedeny v tabulkách Tab. 22, Tab. 23 a Tab. 24.



Obr. 28: Zobrazení maximální délky zprávy v jednotlivých průchodech při generování tří fraktálních struktur

Tab. 25: Transformace textového řetězce zprávy do zašifrovaného tvaru

Vstupní data
Prvnim zpusobem byla fraktalni struktura vygenerovana pomoci kliknuti kurzoru mysi do prislusne oblasti fraktalni mnoziny. Po provedni kliku byl tento bod zaznamenan, zvolen za novy stred zobrazeni a proveden dvojnasobny zoom v teto oblasti. Parametry provedeneho ukonu jsou zaznamenany v podobe souradnic zvoleneho bodu. Dale zde figuruje parametr rozsah a pocet iteraci
Výstupní data
7, 120, 19, 120, 170, 26, 52, 87, 21, 69, 108, 98, 36, 115, 41, 128, 121, 100, 97, 114, 57, 118, 150, 46, 29, 38, 18, 61, 97, 110, 184, 47, 13, 48, 112, 114, 109, 96, 12, 34, 186, 47, 96, 59, 69, 156, 11, 10, 50, 70, 154, 59, 25, 3, 91, 117, 94, 116, 29, 64, 161, 134, 101, 144, 15, 131, 18, 119, 24, 120, 52, 67, 132, 197, 115, 101, 84, 147, 10, 25, 115, 176, 78, 153, 39, 129, 139, 23, 61, 52, 60, 127, 1, 79, 149, 61, 126, 73, 53, 108, 20, 31, 163, 31, 9, 44, 6, 167, 93, 145, 135, 74, 129, 50, 107, 126, 57, 7, 50, 59, 115, 197, 124, 74, 121, 62, 142, 45, 149, 44, 16, 112, 102, 118, 197, 30, 14, 79, 18, 177, 67, 112, 38, 147, 176, 12, 155, 67, 53, 87, 122, 105, 152, 64, 143, 163, 18, 78, 53, 77, 41, 87, 132, 47, 143, 173, 86, 9, 40, 68, 125, 63, 32, 66, 73, 154, 136, 46, 193, 28, 102, 97, 194, 32, 198, 27, 51, 87, 82, 49, 24, 150, 192, 17, 25, 34, 108, 97, 16, 15, 188, 25, 84, 148, 7, 98, 174, 56, 38, 60, 29, 87, 21, 27, 191, 1, 152, 54, 22, 9, 11, 109, 96, 113, 118, 62, 155, 143, 51, 74, 186, 16, 117, 99, 170, 24, 27, 62, 130, 73, 112, 187, 128, 185, 15, 119, 186, 20, 116, 179, 136, 54, 41, 89, 67, 145, 136, 191, 26, 59, 108, 4, 101, 98, 148, 41, 81, 166, 153, 43, 104, 98, 112, 84, 90, 113, 48, 65, 62, 196, 34, 39, 5, 127, 32, 64, 37, 195, 77, 154, 43, 54, 142, 58, 42, 136, 132, 71, 136, 71, 46, 13, 160, 15, 92, 19, 109, 197, 4, 132, 14, 36, 5, 128, 57, 107, 11, 27, 117, 96, 60, 58, 146, 57, 14, 0, 115, 95, 146, 119, 83, 162, 20, 129, 164, 44, 141, 59, 1, 99, 8, 79, 114, 95, 83, 168, 138, 186, 24, 28, 98, 154, 43, 64, 73, 145, 28, 71, 108, 150, 191, 47, 72, 157, 42, 62, 194, 26, 24, 59, 36, 55, 10, 151, 27, 158, 162, 46, 11, 49, 53, 154, 3, 41, 97, 150, 117, 1, 9, 116, 147, 42, 129, 182, 25, 30, 116, 199, 106, 123, 190, 25, 173, 10, 96, 150, 88, 50, 14, 63, 12, 75, 97, 111, 117, 177, 65, 39, 102, 103, 51, 67, 55, 73, 102, 129, 30, 19, 192, 28, 7, 119, 32, 34, 129, 73, 67, 159, 134, 157, 30, 93, 55, 75, 121, 61, 121, 98, 20, 61, 130, 139, 195, 57, 88, 150, 8, 80, 170, 29, 90, 143, 61, 160, 8, 120, 21, 44, 66, 121, 21, 3, 76, 150, 164, 23, 131, 48, 62, 162, 84, 146, 5, 29, 67, 155, 8, 37, 75, 135, 46, 54, 192, 30, 57, 125, 198, 41, 91, 155, 130, 91, 43, 63, 57, 117, 23, 100, 58, 52, 8, 69, 17, 62, 126, 98, 41, 55, 43, 60, 161, 22, 37, 196, 142, 103, 15, 88, 120, 99, 35, 66, 105, 150, 89, 158, 53, 65, 30, 86, 106, 125, 192, 31, 156, 112, 46, 174, 3, 37, 13, 143, 73, 142, 9, 47, 123, 56, 20, 60, 51, 88, 14, 43, 174, 10, 112, 96, 148, 117, 73, 160, 181, 40, 145, 60, 170, 33, 8, 70, 83, 144, 9, 51, 64, 52, 144, 155, 92, 150, 41, 68, 195, 33, 140, 61, 16, 30, 26, 55, 84, 141, 149, 39, 46, 26, 130, 123, 147, 142, 65, 157, 9, 119, 44, 61, 25, 59, 58, 85, 184, 199, 78, 164, 148, 142, 156, 78, 6, 152, 14, 171, 106, 179, 7, 146, 100, 176, 115, 124, 167, 157, 194, 49, 38, 193, 91, 192, 156, 51, 164, 136, 132, 95, 96, 193, 193, 44, 19, 160, 128, 109, 19, 136, 111, 191, 149, 99, 52, 107, 141, 118, 82, 162, 75, 163, 194, 58, 133, 131, 148, 191, 87, 156, 88, 194, 54, 177, 196, 53, 40, 76, 31, 116, 75, 186, 95, 197, 24, 96, 35, 149, 10, 156, 157, 2, 140, 41, 104, 48, 160, 118, 180, 8, 127, 70, 162, 9, 144, 85, 159, 147, 171, 24, 5, 45, 115, 49, 3, 72, 121, 142, 0, 111, 162, 17, 4, 73, 70, 83, 9, 82, 142, 66, 143, 38, 129, 71, 106, 69, 28, 133, 190, 44, 160, 23, 24, 153, 186, 66, 11, 26, 4, 116, 45, 4, 15, 110, 152, 22, 61, 61, 99, 44, 92, 58, 101, 44, 104, 49, 199, 72, 122, 33, 44, 5, 83, 4, 12, 114, 2, 48, 95, 77, 100, 10, 105, 46, 199, 11, 9, 75, 70, 57, 136, 89, 161, 65, 169, 110, 55, 5, 54, 38, 44, 7, 163, 9, 132, 46, 29, 151, 113, 36, 106, 78, 199, 86, 109, 116, 50, 113, 51, 120, 143, 158, 76, 57, 42, 110, 156, 137, 184, 127, 60, 83, 91, 71, 114, 73, 50, 148, 164, 132, 36,

141, 112, 121, 155, 137, 160, 133, 49, 114, 189, 71, 149, 159, 119, 89, 38, 92, 41, 30, 178, 81, 113, 96, 161, 117, 114, 38, 141, 87, 69, 72, 7, 120, 111, 119, 121, 93, 44, 127, 50, 26, 18, 149, 46, 120, 13, 117

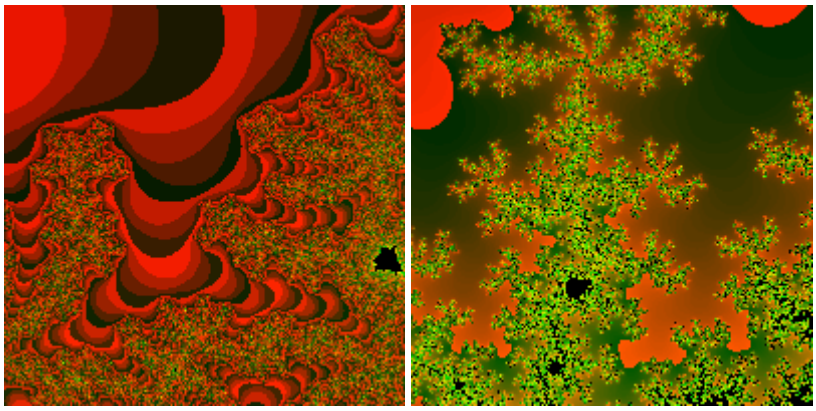
Tabulka Tab. 26 popisuje strukturu vstupních dat šifrovacího algoritmu pro aplikaci na dlouhý typ zpráv s unikátním klíčem. Detailní pohled na strukturu napovídá použití dvojice fraktálů pro zašifrování vstupních dat a klíčů nutných pro tento proces. Třetí fraktál byl určen pro budoucí komunikaci a zajištění unikátnosti použitého klíče. Vstupní informace se nachází v sektoru mezi indexy 0 a 310. Poté co bylo algoritmem zjištěno, že parametry první vygenerované fraktální struktury nejsou schopny pojmout veškerou vstupní informaci, došlo k vygenerování dalšího fraktálu. Mezi indexy 310 a 311 byl vložen znak oddělovače a parametry nového fraktálu byly připojeny mezi oddělovače na pozicích 322, 336, 347 a 351. Na indexu 352 pokračuje další část vstupních informací po index 411. Na této pozici jsou vyčerpány veškeré vstupní informační data. Aby byla zajištěna podmínka unikátního klíče, nachází se na indexu 411 opět znak oddělovače, který uvozuje parametry klíče pro příští komunikaci. Tyto parametry se nachází mezi oddělovači na pozicích 411, 423, 437 a 448.

Tab. 26: Analýza a složení vstupních dat dlouhé zprávy s unikátním klíčem

Pozice	*Délka*	*Obsah*
0	310	vstupní informace – otevřený text
310	1	znak oddělovače klíče
311	11	klíč (*x parametr: -0,54609375*)
322	1	znak oddělovače klíče *(■)*
323	13	klíč (*y parametr: -0,5787109375*)
336	1	znak oddělovače klíče *(■)*
337	10	klíč (*Rozsah: 0,00390625*)

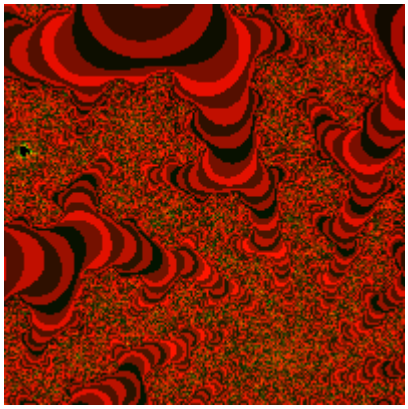
347	1	znak oddělovače klíče *(■)*
314	3	klíč (*Počet iterací: 250*)
351	1	znak oddělovače klíče *(■)*
352	59	vstupní informace – otevřený text
411	1	znak oddělovače klíče *(■)*
412	12	klíč (*x parametr: 0,0733984375*)
423	1	znak oddělovače klíče *(■)*
424	12	klíč (*y parametr: 0,6482421875*)
437	1	znak oddělovače klíče *(■)*
438	10	klíč (*Rozsah: 0,00390625*)
448	1	znak oddělovače klíče *(■)*
449	3	klíč (*Počet iterací: 600*)

Na Obr. 29 jsou zobrazeny použité fraktální struktury pro zašifrování informace. Jejich parametry pro konstrukci jsou uvedeny v tabulkách Tab. 22 a Tab. 23. Levý obrázek byl použitý pro zpracování první části, pravý pro druhou část vstupních informačních dat.



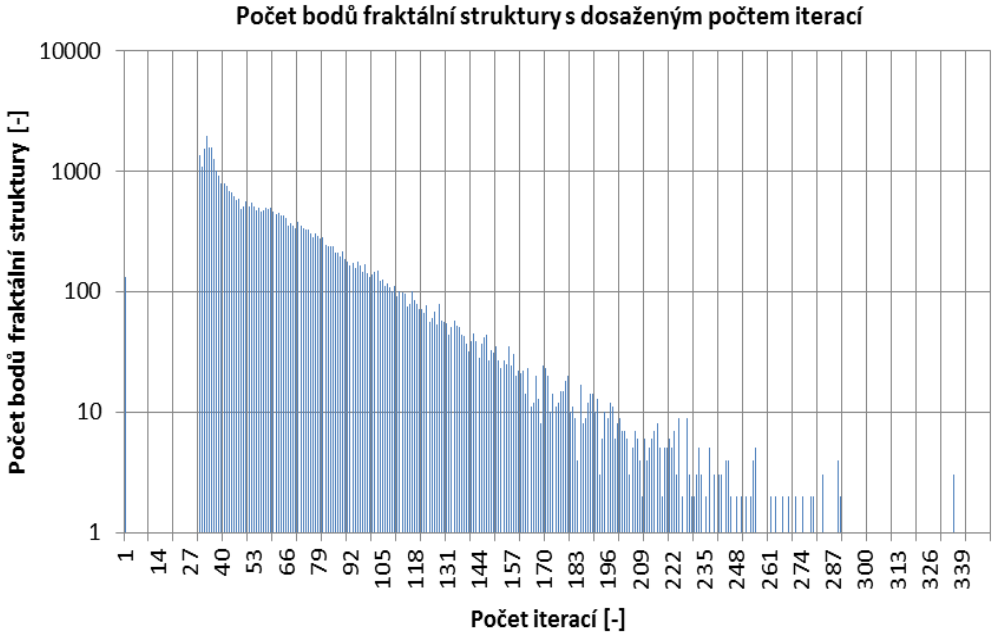
Obr. 29: Použité fraktální struktury

Na Obr. 30 je zobrazen obrázek fraktální struktury určené pro budoucí komunikaci. Zajišťuje unikátnost klíče v daném algoritmu. Parametry pro její konstrukci jsou uvedeny v tabulce Tab. 24.



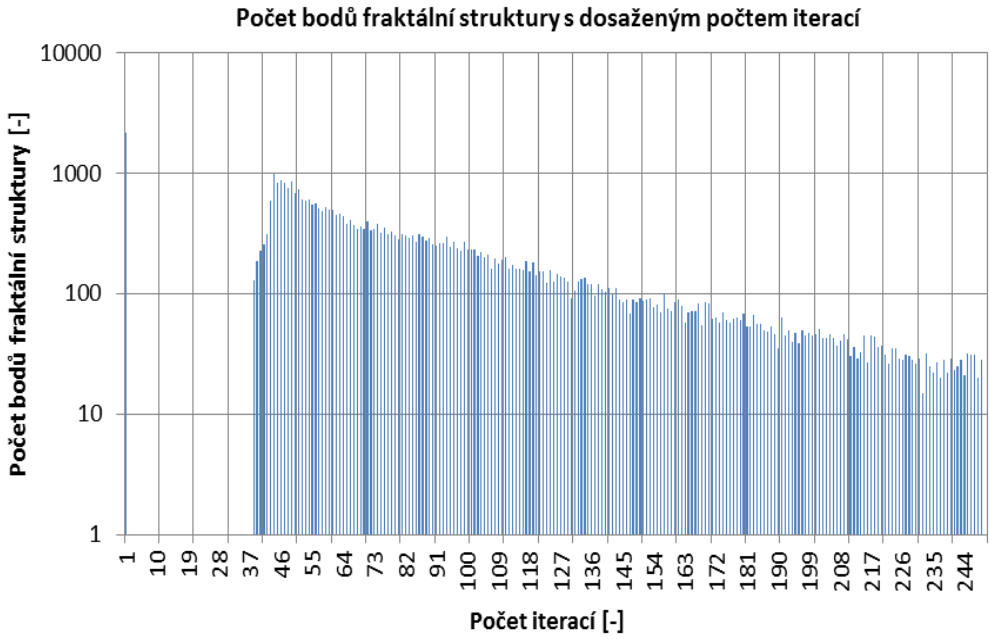
Obr. 30: Fraktální struktura určená pro budoucí použití

Graf na Obr. 31 zobrazuje charakteristiku počtu bodů, které dosáhly příslušný počet iterací při generování fraktálu v posledním průchodu. Tento fraktál byl použit jako první ze dvou při procesu zašifrování informace. Na základě zobrazených dat byla stanovena maximální možná délka vstupní informace na 352 znaků. Tento graf odpovídá levé fraktální struktuře na Obr. 29. Parametry konstrukce této struktury jsou popsány v tabulce Tab. 22. Pro větší názornost byla vertikální osa grafu zobrazena v logaritmickém měřítku.



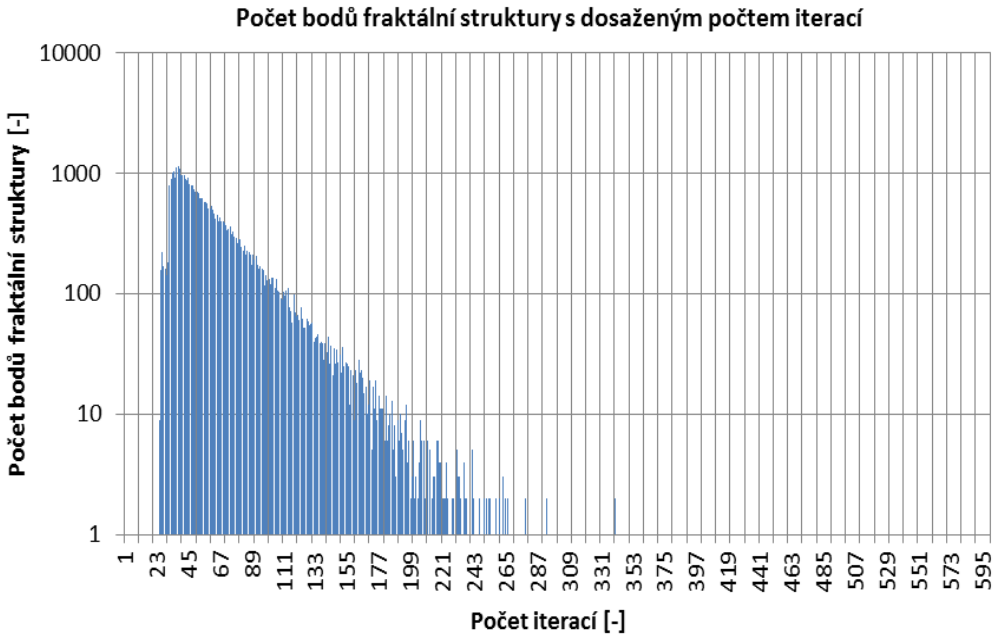
Obr. 31: Počet bodů fraktální struktury č. 1 s dosaženým počtem iterací

Graf na Obr. 32 zobrazuje charakteristiku počtu bodů, které dosáhly příslušný počet iterací při generování fraktálu v posledním průchodu. Tento fraktál byl použit jako druhý při procesu zašifrování informace. Na základě zobrazených dat byla stanovena maximální možná délka vstupní informace na 305 alfanumerických znaků. Znaky, které mohou obsahovat vstupní data, jsou popsány v kapitole 5.2.8. Tento graf odpovídá pravé fraktální struktuře na Obr. 29. Parametry konstrukce této struktury jsou popsány v tabulce Tab. 23. Pro větší názornost byla vertikální osa grafu zobrazena v logaritmickém měřítku.



Obr. 32: Počet bodů fraktální struktury č. 2 s dosaženým počtem iterací

Graf na Obr. 33 zobrazuje charakteristiku počtu bodů, které dosáhly příslušný počet iterací při generování fraktálu v posledním průchodu. Tento fraktál byl použit při procesu zajištění unikátnosti klíče pro budoucí komunikaci. Tato fraktální struktura bude použita v budoucí komunikaci s příjemcem. Na základě zobrazených dat byla stanovena maximální možná délka vstupní informace na 360 alfanumerických znaků. Tento graf odpovídá fraktální struktuře na Obr. 30. Parametry konstrukce této struktury jsou popsány v tabulce Tab. 24. Pro větší názornost byla vertikální osa grafu zobrazena v logaritmickém měřítku.



Obr. 33: Počet bodů fraktální struktury č. 3 s dosaženým počtem iterací

8.2 Proces dešifrování

Princip procesu dešifrování je detailně popsán v kapitole 5.3. Parametry klíčů pro dešifrovací procesy u jednotlivých typů zpráv jsou stejné, jakými byly zprávy šifrovány. Tyto parametry jsou uvedeny v kapitolách 8.1.1, 8.1.2, 8.1.3 a 8.1.4.

8.2.1 Krátké zprávy s neunikátním klíčem

Při dešifrování krátkých zpráv s neunikátním klíčem je provedeno generování fraktálu podle parametrů uvedených v tabulce Tab. 11. Tento vygenerovaný fraktál je zobrazen na Obr. 18. V další fázi jsou načteny zašifrované znaky ve čtvrtém řádku tabulky Tab. 12. Po dokončení procesu dešifrování mají dešifrované data podobu druhého řádku tabulky Tab. 12.

Tabulka Tab. 27 ukazuje sled operací procesu dešifrování krátkých zpráv s neunikátním klíčem.

Tab. 27: Sled operací procesu dešifrování krátkých zpráv s neunikátním klíčem

Operace	*Popis*
Vygenerování fraktálu na základě parametrů	fraktální struktura (Obr. 18)
Dešifrování zprávy	celá zpráva dešifrována v jednom kroku

8.2.2 Dlouhé zprávy s neunikátním klíčem

Při dešifrování dlouhých zpráv s neunikátním klíčem je provedeno generování fraktálu podle parametrů uvedených v tabulce Tab. 14. Tento vygenerovaný fraktál je zobrazen na Obr. 21 vlevo. V další fázi jsou načteny zašifrované znaky ve čtvrtém řádku tabulky Tab. 16. Při procesu dešifrování narazil algoritmus na znak oddělovače. Konkrétní místo popisuje tabulka Tab. 17 na pozici 289. Za tímto oddělovačem následují parametry klíče až do pozice 331, kde na pozici 332 uvozuje oddělovač pokračování otevřeného textu. Z uvedených parametrů je vygenerována fraktální struktura na Obr. 21 vpravo s parametry obsaženými v tabulce Tab. 15. Poté je proces dešifrování na základě této struktury dokončen a dešifrované data mají podobu druhého řádku tabulky Tab. 16.

Tabulka Tab. 28 ukazuje sled operací procesu dešifrování dlouhých zpráv s neunikátním klíčem.

Tab. 28: Sled operací procesu dešifrování dlouhých zpráv s neunikátním klíčem

Operace	*Popis*
Vygenerování fraktálu na základě parametrů	první fraktální struktura (Obr. 21 vlevo)
Dešifrování části zprávy	první část zprávy
Vygenerování fraktálu na základě parametrů	druhá fraktální struktura (Obr. 21 vpravo)
Dešifrování zbytku zprávy	druhá část zprávy

8.2.3 Krátké zprávy s unikátním klíčem

Při dešifrování krátkých zpráv s unikátním klíčem je provedeno generování fraktálu podle parametrů uvedených v tabulce Tab. 18. Tento vygenerovaný fraktál je zobrazen na Obr. 25 vlevo. V další fázi jsou načteny zašifrované znaky ve čtvrtém řádku tabulky Tab. 20. Při procesu dešifrování narazil algoritmus na znak oddělovače. V tuto chvíli je již celá zpráva dešifrována a dešifrované data mají podobu druhého řádku tabulky Tab. 20. Konkrétní místo oddělovače popisuje tabulka Tab. 21 na pozici 10. Za tímto oddělovačem následují parametry klíče pro budoucí komunikaci až do konečné pozice 36. Tyto parametry jsou sumarizovány v tabulce Tab. 19. Fraktální struktura pro budoucí komunikaci je zobrazena na obrázku Obr. 25 vpravo.

Tabulka Tab. 29 ukazuje sled operací procesu dešifrování krátkých zpráv s unikátním klíčem.

Tab. 29: Sled operací procesu dešifrování krátkých zpráv s unikátním klíčem

Operace	*Popis*
Vygenerování fraktálu na základě parametrů	fraktální struktura (Obr. 25 vlevo)
Dešifrování zprávy	celá zpráva dešifrována v jednom kroku
Dešifrování budoucího klíče	klíč pro zajištění budoucí komunikace

8.2.4 Dlouhé zprávy s unikátním klíčem

Při dešifrování dlouhých zpráv s unikátním klíčem je provedeno generování fraktálu podle parametrů uvedených v tabulce Tab. 22. Tento vygenerovaný fraktál je zobrazen na Obr. 29 vlevo. V další fázi jsou načteny zašifrované znaky ve čtvrtém řádku tabulky Tab. 25. Při procesu dešifrování narazil algoritmus na znak oddělovače. Konkrétní místo

popisuje tabulka Tab. 26 na pozici 322. Za tímto oddělovačem následují parametry klíče až do pozice 350, kde na pozici 351 uvozuje oddělovač pokračování otevřeného textu. Na základě parametrů je vygenerována fraktální struktura na Obr. 29 vpravo. Parametry této struktury jsou uvedeny v tabulce Tab. 23. Po této operaci je přikročeno k pokračování v procesu dešifrování. Po nalezení oddělovače, na pozici 411 v tabulce Tab. 26. je uvozena sekce s klíčem určeným pro budoucí komunikaci. V této fázi je proces dešifrování zcela dokončen a dešifrované data mají podobu druhého řádku tabulky Tab. 25.

Tabulka Tab. 30 ukazuje sled operací procesu dešifrování dlouhých zpráv s unikátním klíčem.

Tab. 30: Sled operací procesu dešifrování dlouhých zpráv s unikátním klíčem

Operace	*Popis*
Vygenerování fraktálu na základě parametrů	první fraktální struktura (Obr. 29 vlevo)
Dešifrování části zprávy	první část zprávy
Vygenerování fraktálu na základě parametrů	druhá fraktální struktura (Obr. 29 vpravo)
Dešifrování zbytku zprávy	druhá část zprávy
Dešifrování budoucího klíče	klíč pro zajištění budoucí komunikace

8.3 Dodatek ke kapitolám šifrovacích a dešifrovacích procesů

Kapitoly 8.1 a 8.2 popisují procesy šifrování a dešifrování. Oba procesy probíhaly a byly vyhodnoceny pomocí rozhraní popsaném v kapitole 10, za použití výpočetní techniky uvedené v kapitole 11.1.

Ve zmíněné dvojici kapitol byly popsány čtyři kategorie zpráv. Tyto kategorie byly členěny podle délky zpracovávané zprávy a také podle způsobu nakládáním s klíčem.

Časové nároky na procesy šifrování a dešifrování jsou odlišné. Proces šifrování má vyšší časové nároky než proces dešifrování z důvodu zahrnutého podprocesu generování fraktální struktury popsaného v kapitole 5.2, dále pak rozvinutého v kapitole 7. Časové nároky na podproces generování konkrétní fraktální struktury při použití parametrů aplikovaných v kapitole 8.1, představovaly průměrný čas 2,22s. Časové nároky u samotného procesu zašifrování, popsaném v kapitole 5.2.5, byly změřeny průměrným časem 0,17s. Podproces generování fraktální struktury algoritmem TEA [96] při dešifrování, byl změřen průměrným časem 0,072s. Samotný proces dešifrování byl změřen průměrným časem 0,175s.

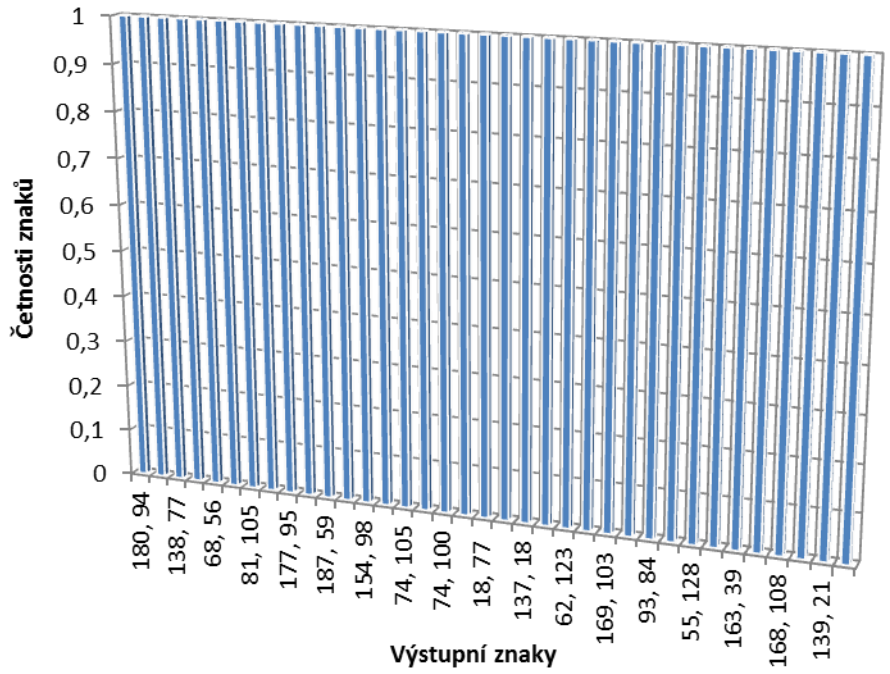
9 ANALÝZA ODOLNOSTI NAVRŽENÉHO ŘEŠENÍ

Následující kapitoly sumarizují poznatky získané analýzou odolnosti navrženého řešení různými metodami. Mezi kryptoanalytickými metodami, které byly použity, figurují Statistické metody, Analytické metody a Útok hrubou silou. Jednotlivé části jsou popsány v kapitole 6, více informací lze nalézt v literatuře [9], [29], [56], [67], [80], [92] a [93].

9.1 Frekvenční analýza

Z konstrukčních parametrů algoritmu pro šifrování krátkých zpráv, popsaných v kapitole 5.2.5, vyplývá nemožnost nalezení v zašifrovaných datech stejné znaky, reprezentující konkrétní znak *Vstupní abecedy znaků*. Otevřený text a zašifrovaná data u typů krátkých zpráv s neunikátním a unikátním klíčem, jsou zobrazeny v tabulkách Tab. 12 a Tab. 20. Výše popsaný algoritmus připouští nalezení stejného řetězce ve výstupních datech. Tato skutečnost je způsobena shodou druhé části souřadnic zašifrovaného znaku a první části souřadnic následujícího zašifrovaného znaku uvnitř výstupního vektoru. Platí však podmínka, že tyto konkrétní znaky nereprezentují konkrétní znak *Vstupní abecedy znaků.* Podle Kerckhoffova předpokladu, uvedeném v kapitole 6.4, však není tato informace pro účel frekvenční analýzy využitelná.

Graf četnosti jednotlivých znaků šifrované zprávy uvedené v tabulce Tab. 20 v kapitole 8.1.3, je zobrazen na Obr. 34. Z tohoto grafu vyplývá, že každý znak šifrované zprávy se v celém souboru dat vyskytuje jen jednou. Pro typy šifrování, kde je použito více fraktálních struktur lze nalézt shodné znaky, ty však vzájemně nesouvisí, tudíž je není možné vzít v potaz v rámci statistické charakteristiky.



Obr. 34: Graf četnosti znaků šifrované zprávy

9.2 Kasiskiho metoda

Na základě parametrů výstupního vektoru krátkých zpráv s unikátními i neunikátními klíči lze říci, že při použití Kasiskiho metody nelze zjistit periodicky se opakující shluky znaků. Tento fakt je dán tím, že je u tohoto typu zpráv zašifrován otevřený text pouze jedním typem fraktální struktury a použitý algoritmus přiřazuje unikátní parametry výstupního datového vektoru pro každý znak šifrovaného textu. U dlouhých zpráv s unikátním a neunikátním klíčem, lze najít v zašifrovaném souboru shodné řetězce. V tomto případě je příčina v použití více fraktálních struktur na konkrétní části otevřeného textu. Tyto řetězce však zastupují různé znaky, které spolu vzájemně nesouvisí. Kasiskiho metoda není úspěšná ani v tomto případě, protože je založena na cyklickém opakování abeced. V tomto případě, kdy bychom vzdáleně přirovnali fraktální strukturu k abecedě, lze říci, že žádná abeceda se zde nevyskytuje více než jedenkrát. Konkrétní klíč pro

generování příslušné fraktální struktury určené k šifrování je použitý pouze jednou. Tyto skutečnosti znemožňují úspěšné provedení úspěšné kryptoanalýzy Kasiskiho metodou.

9.3 Útok hrubou silou

Jak již bylo řečeno v kapitole 6.2, úspěšnost útoku hrubou silou (BFA) se z velké části odvíjí od délky použitého klíče. Parametry klíče, které používá navržené řešení, jsou popsány v kapitole 5.2.6. Navržený algoritmus nemá pevně stanovenou délku klíče. Jeho podoba závisí na parametrech procesu jeho generování. Lze však říci, že vyšší počet průchodů prodlužuje klíč. Tyto procesy jsou popsané v kapitolách 5.2.1, 5.2.2 a 5.2.3.

V demonstrovaném procesu šifrování zprávy, popsaném v kapitole 8.1.2, byl pro generování první fraktální struktury použitý klíč uvedený v tabulce Tab. 14. Tabulka Tab. 31 ukazuje počet kombinací jednotlivých částí tohoto klíče.

Tab. 31: Počet kombinací jednotlivých částí klíče prvního fraktálu

Parametr	*Kombinací*
x parametr	$4*10^{10}$
y parametr	$4*10^{8}$
Rozsah	$2*10^{8}$
Počet iterací	999
Celkem	$3{,}1968*10^{30}$

Určení počtu možných kombinací klíče souvisí s vlastnostmi použitého fraktálu, zejména pak s rozsahem bodů komplexní roviny, ve kterých se příslušný fraktál nachází. Ve zmíněném procesu je demonstrována *Mandelbrotova množina 200x200*, ležící v komplexní rovině ohraničené krajními body: -2, +2, -2i, +2i. Maximální počet iterací byl určen na 999. Na základě vyhodnocení tabulky Tab. 31 bylo zjištěno, že délkou klíče

použitého v daném případě lze vyjádřit $3{,}1968*10^{30}$ kombinací. Změřená průměrná rychlost generování fraktálu od nejkratších klíčů po parametry použitého klíče byla 0,68s. Výsledný čas na vygenerování všech kombinací pomocí výpočetní techniky, popsané v kapitole 11.1, by trval $4{,}8657*10^{22}$ let. Ostatní testované fraktály, vycházející z konstrukce Mandelbrotovy množiny dosahují podobných výsledků. U použitého klíče pro Juliovy množiny z tabulky Tab. 11, se počet kombinací zvýší o parametry konstant *Cx* a *Cy*. Počet kombinací byl vypočítán na $5{,}11488*10^{31}$. Zde by trvalo vyzkoušení všech kombinací $1{,}11912*10^{24}$ let. Použité parametry klíče splňují podmínku bezpečnosti útoku hrubou silou. Úzce související problematika, je dále popsána v kapitole 9.4.2. Zde jsou porovnány vlastnosti testovaných fraktálů z pohledu časové náročnosti na útok hrubou silou při použití různých délek klíčů.

Do časové náročnosti útoku hrubou silou není zahrnut čas potřebný k analýze dešifrovaného obsahu zprávy a určení správného řešení, protože každá kombinace klíče produkuje různé kombinace výstupních řetězců. Tímto způsobem může docházet ke generování dat, které dávají smysl, ale vůbec nesouvisí s otevřenými daty před procesem šifrování. Tento fakt značně ztěžuje a v určitých případech i zcela znemožňuje útok hrubou silou.

9.4 Chosen Plaintext Attack

Princip metody *Chosen Plaintext Attack* je shrnut v kapitole 6.3.1. Metoda *CPA* dovoluje měnit otevřený text a sledovat jeho zašifrovanou podobu. Z tohoto důvodu byla také použita pro určení odolnosti navrženého řešení.

9.4.1 Vstupně výstupní analýza

Z algoritmu šifrování, popsaném v kapitole 5.2.5 je zřejmé, že při použití jednoho klíče jsou výstupní data zašifrovány vždy různě. Tato skutečnost byla také prakticky ověřena. Tato implementovaná vlastnost zajišťuje vyšší odolnost navrženého algoritmu. Tabulka Tab. 37 ukazuje některé varianty zašifrovaných dat otevřeného textu „*UNIVERZITA“*. Použití klíč pro je uveden v tabulce Tab. 14.

Tab. 32: Některé z variant zašifrovaných dat unikátního textu shodným klíčem

č.	*Varianty šifrovaných dat*
1.	77, 133, 46, 153, 36, 171, 46, 149, 22, 186, 19, 181, 29, 153, 51, 169, 85, 169, 1, 195
2.	19, 165, 160, 94, 74, 196, 166, 66, 141, 150, 1, 175, 114, 89, 74, 197, 46, 167, 190, 14
3.	141, 136, 153, 127, 186, 45, 102, 71, 179, 107, 149, 129, 147, 129, 74, 196, 46, 167, 190, 151
4.	93, 21, 29, 186, 74, 196, 187, 74, 125, 170, 97, 146, 56, 160, 75, 191, 188, 89, 190, 146
5.	118, 59, 165, 87, 74, 196, 151, 121, 125, 170, 19, 169, 24, 161, 153, 129, 45, 174, 190, 144
6.	38, 179, 196, 90, 186, 57, 157, 85, 125, 170, 85, 155, 73, 171, 74, 196, 16, 167, 190, 144
7.	130, 47, 84, 158, 180, 25, 16, 166, 160, 126, 116, 143, 83, 95, 74, 196, 6, 173, 192, 136
8.	158, 70, 173, 96, 74, 196, 167, 67, 132, 156, 183, 55, 164, 60, 139, 141, 61, 159, 190, 158
9.	90, 154, 127, 131, 74, 196, 76, 166, 125, 170, 30, 179, 173, 63, 162, 99, 61, 159, 190, 144
10.	25, 165, 44, 177, 74, 196, 90, 28, 125, 170, 65, 168, 107, 93, 78, 185, 149, 130, 190, 159

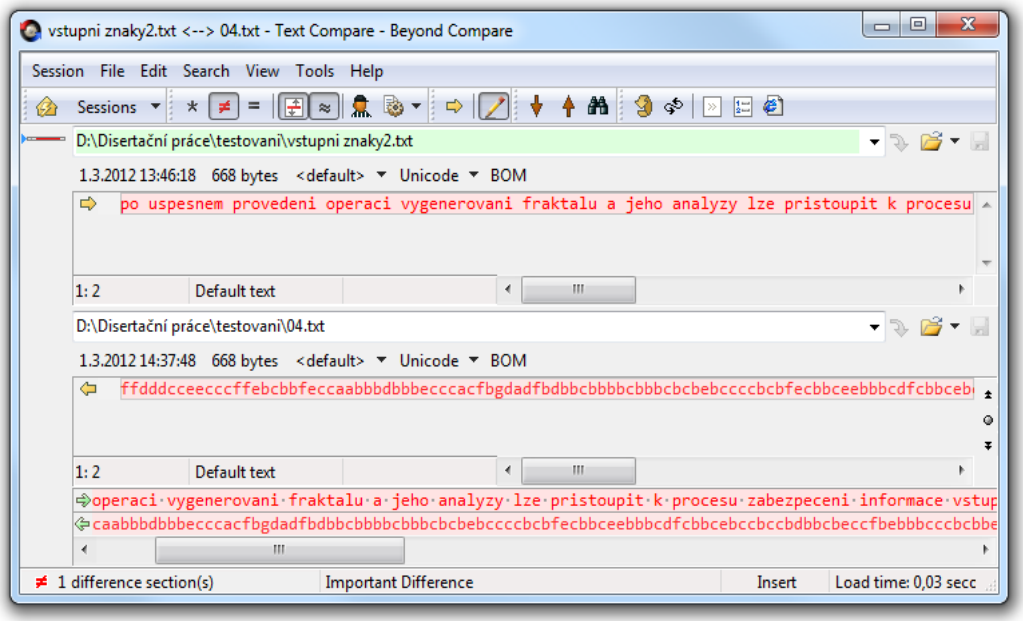
Na základě obsahu tabulky Tab. 32 lze říci, že v případě použití neunikátního klíče a zachycení takového velkého množšství množství dat, by mohla být úspěšně provedena frekvenční analýza popsaná v kapitole 6.1.1. K této situaci by mohlo dojít právě při opakování stejného klíče. Proces frekvenční analýzy sice ztěžuje fakt, že výstupem šifrování je pokaždé jiná sekvence výstupních dat, ale použití unikátního klíče je v tomto případě žádoucí. Pravděpodobnost opakování konkrétních znaků výstupního vektoru při použití neunikátního klíče je u fraktálních struktur s nízkým rozlišením vyšší, než u fraktálů s vyšším rozlišením. *Krátké zprávy s unikátním klíčem* a *dlouhé zprávy s unikátním klíčem* jsou proti zmíněné hrozbě chráněny. Problematika unikátního klíče, která zabezpečuje informaci proti použití zmíněné metody je detailně popsaná v kapitole 5.5.

Závěrem této podkapitoly lze říci, že unikátní klíč zajistí, aby byla pro účel zašifrování zprávy generována vždy jiná fraktální struktura. Tento fakt znemožňuje zmíněnou analýzu a je klíčovým prostředkem ke zvýšení bezpečnosti navrženého řešení.

9.4.2 Analýza klíče

Jak již bylo zmíněno v kapitole 5.2, fraktální struktura je vygenerována na základě parametrů klíče. Tato struktura svými parametry umožňuje zašifrování informace pomocí procesu popsaného v kapitole 5.2. Podoba klíče je uvedena v kapitole 5.2.6. Analýza klíče spočívá v jeho modifikaci a ověření jejího vlivu na podobu výstupních dat po procesu dešifrování, popsaném v kapitole 5.3.3.

V první fázi analýzy proběhl proces šifrování zprávy. Dále následoval proces dešifrování, kdy došlo k modifikaci jednotlivých částí použitého klíče. Jednotlivé dešifrované zprávy pomocí odlišných klíčů byly ukládány do souborů. Tyto soubory byly porovnávány mezi sebou a zejména s výstupními znaky dešifrované zprávy pomocí originálního klíče. K porovnávání dat byl použit software Beyond Compare 3, popsaný v kapitole 11.2.3. Rozhraní programu je zobrazeno na Obr. 35. Po výběru příslušných souborů byla provedena analýza a porovnání. Ve spodní části rozhraní je možné procházet příslušné řetězce a prohlížet vzájemné souvislosti mezi mini.



Obr. 35: Porovnávání dešifrovaných dat

Analýza potvrdila předpoklad, že při změně hodnoty klíče na pozicích posledních desetinných míst, může dojít k vygenerování struktury, při které se nemusí změnit dešifrovaná data. V návaznosti na tento fakt byla provedena analýza, při které byla stanovena hranice pro bezpečnost klíče. Tato hranice garantuje minimální počet kombinací v návaznosti na délku klíče, které je nutné vykonat, aby bylo možné ohrozit bezpečnost daného algoritmu. Tabulka Tab. 33, vztažena k Mandelbrotově množině, ukazuje průměrné hodnoty možných kombinací klíče a průměrné množství kombinací, při kterých již nelze najít ve výstupním souboru dat shodné znaky se znaky otevřeného textu. První sloupec tabulky Tab. 33 označuje počet průchodů, při kterých bylo prováděno testování na danou podmínku.

Tab. 33: Množství kombinací klíče a časová náročnost útoku hrubou silou - Mandelbrot

Počet průchodů	*Průměrné množství kombinací klíče*	*Průměrné množství kombinací klíče při podmínce bezpečnosti*	*Časová náročnost útoku hrubou silou [roky]*
5	$3{,}19*10^{15}$	$3{,}19*10^{12}$	*69945,2*
10	$3{,}19*10^{30}$	$3{,}19*10^{17}$	*6994520548*
15	$3{,}19*10^{41}$	$3{,}19*10^{22}$	$6{,}99452*10^{14}$
20	$3{,}19*10^{52}$	$3{,}19*10^{29}$	$6{,}99452*10^{21}$
25	$3{,}19*10^{53}$	$3{,}19*10^{37}$	$6{,}99452*10^{29}$

Tab. 34: Parametry klíče použité v procesu analýzy klíče - Mandelbrot

X parametr	*Y parametr*	*Rozsah*	*Počet iterací*
-0,8975	*0,2725*	*0,125*	*250*
-0,047851563	*0,67578125*	*0,00390625*	*250*
-1,320056152	*0,083692627*	*0,00012207*	*250*
-0,254211388	*-0,781685371*	*3,81E-06*	*250*
-1,289140037	*-0,066665963*	*1,19E-07*	*250*

Tab. 35: Prům. množství kombinací klíče s podmínkou bezpečnosti u dalších fraktálů

Počet průchodů	*Juliovy množiny*	*Burning Ship*	*Bird of Prey*	*Water Plane*	*4th Degree Multibrot*
5	$5{,}11*10^{20}$	$3{,}19*10^{14}$	$3{,}19*10^{13}$	$3{,}19*10^{11}$	$3{,}19*10^{12}$
10	$5{,}11*10^{27}$	$3{,}19*10^{18}$	$3{,}19*10^{17}$	$3{,}19*10^{16}$	$3{,}19*10^{16}$
15	$5{,}11*10^{33}$	$3{,}19*10^{23}$	$3{,}19*10^{20}$	$3{,}19*10^{19}$	$3{,}19*10^{20}$
20	$5{,}11*10^{41}$	$3{,}19*10^{31}$	$3{,}19*10^{30}$	$3{,}19*10^{29}$	$3{,}19*10^{29}$
25	$5{,}11*10^{48}$	$3{,}19*10^{38}$	$3{,}19*10^{37}$	$3{,}19*10^{36}$	$3{,}19*10^{36}$

Tab. 36: Porovnání časové náročnosti v letech útoku hrubou silou u ostatních fraktálů

Počet průchodů	*Juliovy množiny*	*Burning Ship*	*Bird of Prey*	*Water Plane*	*4th Degree Multibrot*
5	$1{,}118*10^{13}$	*6994520,5*	*699452,1*	*6994,5*	*69945,2*
10	$1{,}118*10^{20}$	$6{,}99452*10^{11}$	*6994520548*	*699452054,8*	*699452054,8*
15	$1{,}118*10^{26}$	$6{,}99452*10^{15}$	$6{,}99452*10^{12}$	$6{,}99452*10^{11}$	$6{,}99452*10^{12}$
20	$1{,}118*10^{34}$	$6{,}99452*10^{23}$	$6{,}99452*10^{22}$	$6{,}99452*10^{21}$	$6{,}99452*10^{21}$
25	$1{,}118*10^{41}$	$6{,}99452*10^{30}$	$6{,}99452*10^{29}$	$6{,}99452*10^{28}$	$6{,}99452*10^{28}$

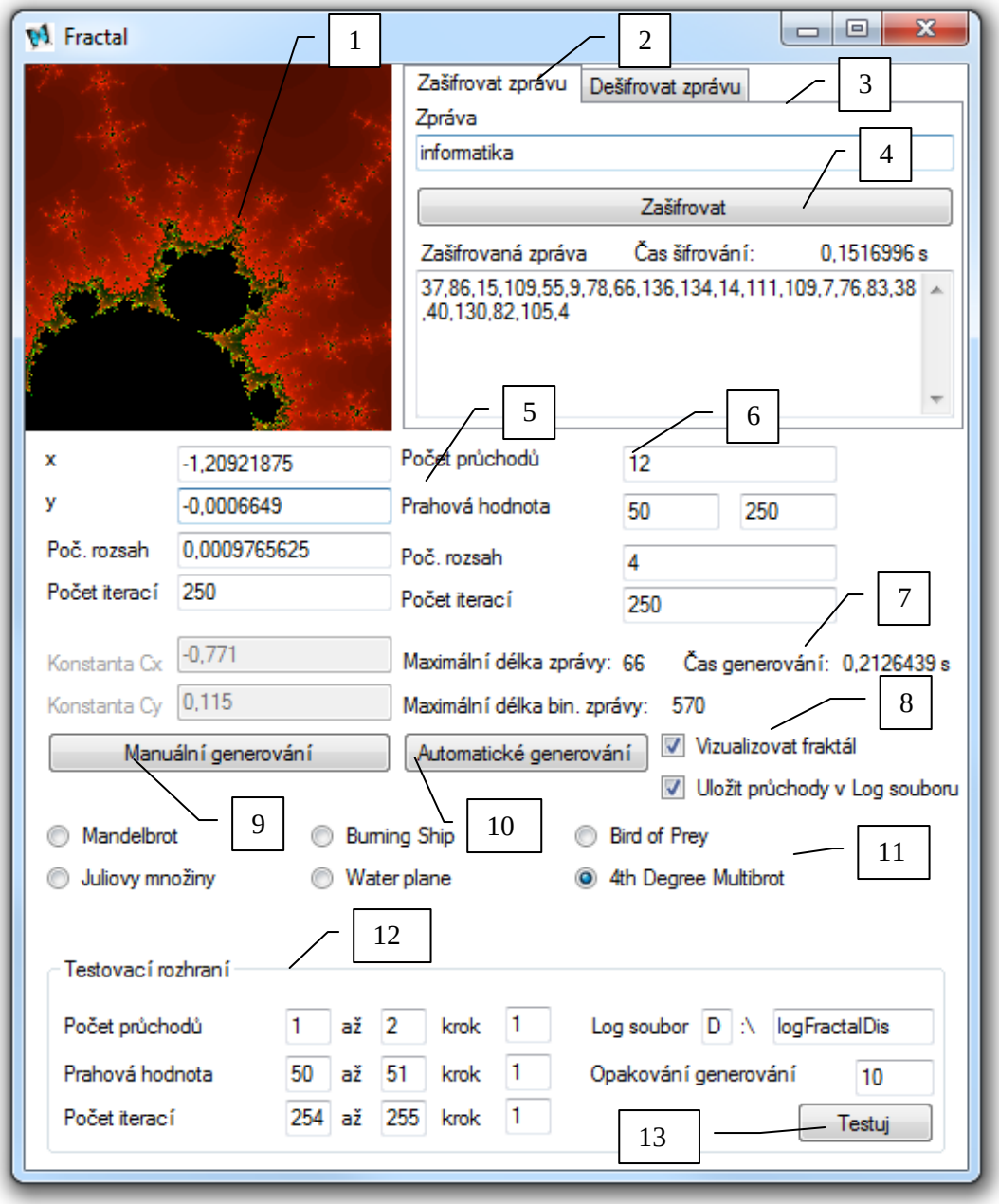
Z výše uvedeného je patrné, že při vyšším počtů průchodů jsou generovány delší klíče. Parametry klíčů použité v procesu analýzy klíče u Mandelbrotovy množiny jsou zobrazeny v tabulce Tab. 34. Délka klíče má podstatný vliv na bezpečnost algoritmu vůči útoku hrubou silou. Při zachování podmínky bezpečnosti u klíče s nejkratší délkou, použitého v prvním případě z tabulky Tab. 33, činí časová náročnost útokem hrubou silou 69945,2 let. Odtud plyne doporučení použití 10 a více průchodů. V případě klíče s vyšší délkou, použitého v pátém řádku tabulky byla vypočtena hodnota $6{,}99452*10^{29}$ let. Výpočet doby

je vztažen k použité výpočetní technice popsané v kapitole 11.1. Do těchto časů není započten čas na analýzu správného řešení a v případě získání více zpráv, které mají odlišný smysl útočník nelze jednoznačně určit, která konkrétní varianta je správná. Více o problematice počtu kombinací a časové náročnosti útoku na algoritmus pojednává kapitola 9.3. Výše uvedené poznatky prokázaly odolnost navrženého řešení proti zmíněné analytické metodě.

10 ROZHRANÍ PRO ŠIFROVÁNÍ, DEŠIFROVÁNÍ A TESTOVÁNÍ NAVRŽENÉHO ŘEŠENÍ

Na Obr. 36 je zobrazeno naprogramované rozhraní pro analýzu operací šifrování, dešifrování a testování navrženého řešení. Toto rozhraní bylo naprogramováno v programovacím jazyce C# pomocí vývojového prostředí Visual Studio 2010, popsané v kapitole 11.2.1. Rozhraní je rozděleno do několika logických částí. Tyto části lze blíže specifikovat jako sekce pro generování, šifrování, dešifrování a testování. Jednotlivé sekce jsou detailně popsány níže.

Do sekce generování fraktální struktury spadají prvky 1 a 5 – 11. Prvky pro šifrování jsou zobrazeny pod ovládacími prvky 2 a 4. Prvky pro dešifrování zobrazuje ukazatel 3. Po otevření karty pro dešifrování je tato sekce zobrazena detailně na Obr. 37. Sekci pro testování znázorňují prvky 12 a 13. Výstup je ukládán do souboru na pevný či síťový disk počítače.



Obr. 36: Rozhraní pro šifrování a dešifrování

10.1 Sekce generování

Pro operaci automatického generování, popsané v kapitole 5.2.2, slouží ke vstupu parametrů generování ovládací prvky pod ukazatelem č. 6 na Obr. 37. Zde jsou zadány parametry *Počet průchodů, Prahová hodnota, Počáteční rozsah a Počet iterací*. Funkce těchto parametrů popisuje detailně kapitola 5.2.7. Pro spuštění procesu automatického generování slouží tlačítko 10. Po vygenerování fraktální struktury jsou v sekci 5 zobrazeny konstrukční parametry vygenerovaného fraktálu. V sekci 7 je zobrazena informace o maximální délce zprávy, kterou daná fraktální struktura nabízí. V sekci 8 je ovládací prvek pro možnost vizualizace fraktálu. Vizualizace fraktálu není pro chod algoritmu důležitá a slouží jen pro představu uživateli, jaký fraktál byl vytvořen. Dalším prvkem je volba uložení parametrů jednotlivých průchodů generování do souboru na pevném disku počítače. Tato výstupní data byly využity zejména v kapitole 8.1. Konkrétně pak u charakteristik na Obr. 17, Obr. 19, Obr. 20, Obr. 22, Obr. 23, Obr. 24, Obr. 26, Obr. 27, Obr. 28, Obr. 31, Obr. 32 a Obr. 33.

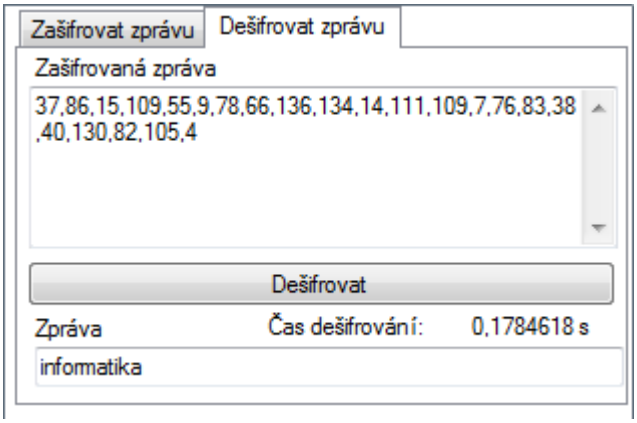
Ovládací rozhraní umožňuje také provést proces manuálního generování, který je popsaný v kapitole 5.2.1. Zde je fraktál vytvářen uživatelsky, postupným klikáním kurzoru myši do oblasti 1 a použití tlačítka 9. Pomocí ovládacích prvků v sekci 11 lze volit různé typy fraktálů pro použití pro daný algoritmus.

10.2 Sekce šifrování

Pro proces šifrování, detailně popsaném v kapitole 5.2.5, slouží na Obr. 36 karta 2. Otevřený text je vložen uživatelem do textového pole *Zpráva*. Po stisknutí tlačítka 4 je zpráva zašifrována a zobrazena v *multiline* textovém poli nad sekcí 5 a 6.

10.3 Sekce dešifrování

Proces dešifrování je detailně popsán v kapitole 5.3.3. Po klinutí kurzoru na kartu *Dešifrovat zprávu* na Obr. 36, vystoupí do popředí rozhraní znázorněné na Obr. 37. V horní části – v textovém multiline poli je vložena zašifrovaná informace a pomocí tlačítka *Dešifrovat* je proveden proces dešifrování. Otevřený text je poté zobrazen v textovém poli *Zpráva*.



Obr. 37: Detail karty pro dešifrování

10.4 Sekce testování

Sekce testování se nachází ve spodní části rozhraní, zobrazeném na Obr. 36. Levá část testovacího rozhraní obsahuje prvky pro vkládání rozsahu testovaných parametrů. Každý z parametrů má počáteční a koncovou hodnotu. Za každým parametrem je prvek pro inkrementaci testovacího kroku.

V levé části testovacího rozhraní leží prvek pro pojmenování výstupního souboru a určení jednotky datového úložiště. Nad tlačítkem 13 je vstupní textové pole, kde lze nastavit počet opakování pokusu pro zpřesnění výsledku. Tlačítkem 13 je spuštěn proces testování. Během procesu testování je celé testovací rozhraní v režimu disable a nelze do něj zasahovat. Po ukončení procesu je opět testovací rozhraní uvedeno do režimu enable a je připraveno na nové testování. Výsledky z testovacího procesu byly využity zejména pro podklady v kapitole 7.1. Jak již bylo popsáno výše, údaje zapsané do příslušných textových polí levé části testovacího rozhraní 12 jsou automaticky dosazovány do algoritmu generování fraktálů. Měření je opakováno na základě parametru uvedeného v textovém poli nad tlačítkem 13. Výsledky jsou průměrovány a ukládány do výstupního souboru. Výstupní data lze poté importovat do prostředí MS Excel, kde lze datový soubor podrobit bližšímu zkoumání, vynést graficky, apod. Importovaná vstupní data v prostředí MS Excel jsou zobrazena na Obr. 38.

Obr. 38: Importovaná testovací data v prostředí MS Excel

11 PROSTŘEDKY VYUŽITÉ PRO VÝVOJ A TESTOVÁNÍ

Pro provádění výpočtů, experimentů a dalších úkonů související s vytyčenými cíly disertační práce byla použita dvojice vývojových nástrojů. Prvním z nich bylo prostředí programovacího jazyku C# [6], dalším z nich byl matematický software Mathematica společnosti Wolfram. Při výzkumu byly tyto nástroje voleny tak, aby byly v každé jeho části využity přednosti dané platformy. Při práci byly také použity další podpůrné nástroje a hardwarové vybavení, popsané níže.

11.1 Hardware

Pro vývoj a testování byl použit počítač s parametry uvedenými v tabulce Tab. 37.

Tab. 37: Parametry použité výpočetní techniky

Název	*Parametry*
Počítač	Fujitsu Siemens - Esprimo Mobile
Procesor	Intel Core Duo CPU P7350 2.00 GHz
Paměť RAM	4,00 GB (použité 2,96)
Typ systému	32bitový operační systém
Verze operačního systému	Windows 7 Professional

11.2 Software

11.2.1 Microsoft Visual C# 2010

Jazyk C# [6] představuje objektově orientovaný programovací nástroj vyvinutý společností Microsoft. V současné době je hojně využívaný pro vývoj robustních, rychlých a kvalitních aplikací. Prostředí integrovaného designeru umožňuje tvořit uživatelsky pružné ovládací menu a komponenty pro zobrazování textu i grafiky. Vytvořená desktopová aplikace je určena pro systémy s operačním systémem Windows.

11.2.2 Mathematica

Pro analýzu některých fraktálních množin byl použit software Mathematica od společnosti Wolfram. Představuje komplexní nástroj využitelný v mnoha odvětvích výzkumu. Umožňuje sofistikovaný způsob programování, dokáže názorně vizualizovat zpracovaná data pomocí dvojrozměrných a třírozměrných nástrojů. Obsahuje také mnoho dalších potřebných funkcí pro výzkumnou činnost. Práce uvnitř vývojového prostředí je velmi intuitivní a poměrně rychlá.

11.2.3 Beyond Compare 3

Nástroj *Beyond Compare 3* byl použit pro analýzu výstupních dat při modifikacích klíče. Jeho nasazení proběhlo ve fázi testování navrženého řešení a zejména pak v části *Analýza klíče*, popsané v kapitole 9.4.2. Program vyvinula společnost Scooter Software, nástroj byl použit ve verzi 3.3.4 (build 14431).

11.2.4 Dia

Nástroj *Dia* byl použit pro tvorbu vývojových diagramů v teoretické části práce. Použitá verze daného software má označení 0.95-1.

11.2.5 Microsoft Office 2010

Pro psaní disertační práce bylo použito textového editoru Word. Pro import testovacích dat z programu *Fractal*, zejména ve fázích *Generování fraktální struktury*, *Proces šifrování a dešifrování* a *Analýza odolnosti navrženého řešení*, bylo použito tabulkového kalkulátoru Excel. Naprogramované prostředí *Fractal* je detailně popsáno v kapitole 10.

12 PŘÍNOS PRÁCE PRO VĚDU A PRAXI

Navržený způsob šifrování, opírající se o principy fraktální geometrie, nabízí širokou škálu možností svého uplatnění v oblasti kryptografického zabezpečení komunikace uvnitř i vně informačních systémů. Výzkum provedený při zpracování tématu disertační práce poukazuje na možnost využití fraktálů v oblasti kryptografie.

Šifrovací a dešifrovací procesy daného algoritmu lze využít ve vazbách člověk - člověk, člověk – stroj i stroj – stroj. Vazbu člověk – člověk lze použít v situacích, kdy je vhodné zabezpečit komunikaci vysokého stupně utajení mezi odesílatelem a příjemcem. Daný proces, implementovaný do vazby člověk - stroj, zahrnuje zabezpečení komunikace uživatele s prvky informačních systémů, se kterými pracuje. Tyto systémy na základě vložených požadavků uživatele zprostředkovávají požadovaný výstup, zajišťují zpracování požadavků pro další použití nebo pouze uloží potřebná data do databáze. Podobně je tomu dále ve vazbě stroj – stroj, kde navržený způsob zajišťuje bezpečnost vzájemné komunikace prvků informačních systémů, mezi kterými dochází taktéž, jako v minulých případech, k předávání informací důvěrného charakteru.

Možnou oblast využití navrženého řešení spatřuji mimo jiné v prostředí systémů elektronického bankovnictví, v komunikačních aplikacích s prioritou vysokého stupně utajení, ve státní správě i v soukromých institucích v evidenčních systémech osob a majetku, dále pak ve zdravotnických informačních systémech ve vedení evidence pacientů, ukládání a archivaci laboratorních výsledků a diagnóz pacienta.

Rozhraní využívající navržené řešení lze implementovat jak do oblasti desktopových aplikací, tak do budoucna i na mobilní zařízení a rozšířit dále jeho použití na různé platformy.

13 ZÁVĚR A DISKUZE

Obsah disertační práce se zabývá aplikací získaných poznatků na poli fraktální geometrie do problematiky informační bezpečnosti, konkrétně do oblasti šifrování komunikace mezi prvky informačních systémů. Navržené řešení se opírá o skupinu fraktálů vytvořených pomocí algoritmu TEA, která svými principy tvoří jeho stěžejní část.

V úvodu teoretické části se nachází popis a rozbor problematiky týkající se fraktální geometrie a selekce vhodné skupiny fraktálů pro účel daný tématem disertační práce. V dalších kapitolách této části byly formulovány a pomocí vývojových diagramů znázorněny principy jednotlivých etap navrženého šifrovacího a dešifrovacího procesu. Teoretickou část uzavírá kapitola zaměřená na metodiku určení odolnosti navrženého řešení vůči kryptoanalytickým metodám. V této části byly zahrnuty metody statické, analytické i útok využívající hrubé síly (BFA). Kapitoly obsažené v praktické části disertační práce popisují poznatky chování a vlastností jednotlivých částí šifrovacích a dešifrovacích procesů. Tyto kapitoly jsou řazené chronologicky na základě pořadí výskytu jednotlivých částí procesů, které popisují. V praktické části se dále nachází analýza odolnosti navrženého řešení a její výstupy. Praktickou část uzavírá popis vytvořeného software v jazyku C#, *Fractal*. Tento nástroj byl vytvořen pro účel praktického ověření a realizaci šifrovacích a dešifrovacích procesů, dále pro analýzu jednotlivých fraktálních struktur a jako jeden z pomocných nástrojů při určování odolnosti navrženého řešení vůči kryptoanalytickým metodám.

Prvním důležitým krokem ve výzkumu bylo zajistit generování fraktální struktury s vhodnými parametry pro proces zašifrování zprávy. Provedenými testy pomocí software *Fractal* bylo určeno vhodné nastavení generátoru pro zvolené typy fraktálů. V další části práce byla zpracována část věnující se analýze fraktálních struktur, jejíž výstupy slouží pro určení maximálního množství znaků, které je možné pomocí dané struktury zašifrovat. Na základě této analýzy byl určen režim šifrování, který se odvíjí od délky otevřeného textu. V případě vyšší délky otevřeného textu, než může daná fraktální struktura zpracovat je šifrováno režimem *Dlouhých zpráv*, kdy je provedeno generování dalších fraktálních struktur pro zpracování veškerého otevřeného textu. Jak se ukázalo provedenými testy, pro navržený účel je z hlediska časové náročnosti výhodnější generovat nové fraktály než

například zvyšovat rozlišení fraktálu nad rámec zvolených parametrů. Na základě provedené průběžné kryptoanalýzy v průběhu vývoje, byl způsob šifrování rozšířen o tzv. *šifrování s unikátním klíčem*. Ukázalo se, že implementací tohoto řešení byla zvýšena bezpečnost algoritmu zejména vůči statistickým metodám kryptoanalýzy.

Navržené řešení bylo průběžně, ale i na závěr podrobeno kryptoanalytickému zkoumání. Smyslem prováděných analýz bylo odhalit a ošetřit slabá místa algoritmu a dostát očekávaným požadavkům na jeho funkčnost a smysl vynaloženého úsilí. Kryptoanalýza byla vedena z více hledisek na základě odlišných přístupů k danému problému. Byla zde prokázána odolnost vytvořeného algoritmu vůči kryptoanalytickým metodám a splněny požadavky na vytyčené cíle.

Smyslem této disertační práce bylo nalézt způsob kryptografického zabezpečení informace zahrnující v sobě principy relativně mladé vědecké disciplíny – fraktální geometrie. Námět pro další výzkum spatřuji v zapojení metod umělé inteligence v procesu generování fraktálních struktur s vhodnými parametry pro šifrovací procesy a dále rozvoj činností v oblasti distribuce klíče před prvotním zahájením používání navrženého řešení.

POUŽITÁ LITERATURA A INFORMAČNÍ ZDROJE

[1] ALLAËRT, F. *Security Standards for Healthcare Information Systems:* A Perspective from the Eu Isis Medsec Project. 1. vyd. Amsterdam: IOS Press, 2008. 239 s. ISBN: 9781586030001.

[2] ALIA, M., A., SAMSUDIN, A., B. A New Digital Signature Scheme Based on Mandelbrot and Julia Fractal Sets. *American Journal of Applied Sciences 4*. 1st ed. USA: Science Publications, 2007, vol. 9, p. 848-856.

[3] ALIA, M., A., SAMSUDIN, A., B. New Key Exchange Protocol Based on Mandelbrot and Julia Fractal Sets. *IJCSNS International Journal of Computer Science and Network Security*. 1st ed. Korea: IJCSNS, 2007, vol. 6, p. 302-307.

[4] ASSCHE, G.,V. *Quantum Cryptography And Secret-key Distillation*. 1. vyd. United Kingdom: Cambridge University Press, 2006. 261 s. ISBN: 9780521864855.

[5] BARYSHEV, Y., TEERIKORPI, P. *Discovery of osmic fractals*. 1. vyd. London: World Scientific Pub Co Inc., 2002. 408 s. ISBN-10: 9789810248727.

[6] BISHOP, D. *A Complete guide to C#*. Jones & Bartlett Learning, 2004. 624 s. ISBN 0763722499.

[7] BLACKLEDGE, J., M., EVANS, A., K., TURNER, M., J. *Fractal Geometry:* Mathematical Methods, Algorithms, Applications. 1. vyd. United Kingdom: Horwood Pub. for the Institute of Mathematics and its Applications, 2002. 244 s. ISBN: 9781904275008.

[8] BOOKS, H. *Articles on Secure Communication, Including:* Computer Security, HTTP Secure, Transport Layer Security, Stu-III, Secure Terminal Equipment, Temporal Key Integrity Protocol, Ccmp, Secure Communications Interoperability Protocol. 1. vyd. United Kingdom: Hephaestus Books, 2011. 148 s. ISBN: 9781243956019.

[9] BOSE, S. *Information theory, coding and cryptography*. Tata McGraw-Hill Education, 2008. 326 s. ISBN 0070669015.

[10]BRANDT, CH., MÖRTERS, P. *Fractal Geometry and Stochastics IV*. 1. vyd. Germany: Springer, 2009. 290 s. ISBN: 9783034600293.

[11] BRODER, J., TUCKER, G. *Risk Analysis and the Security Survey*. 4. vyd. USA: Elsevier, 2012. 368 s. ISBN: 9780123822345.

[12] BRODTKORB, A. R., HAGEN, T. R. *A Comparison of free commodity – level parallel architectures: Multi-core CPU, cell BE and GPU*. 7th International Conference MMCS 2008, Tonsberg, Norway. ISBN 10-3-642-11619-1. s. 70-79.

[13] BUŇATOVÁ, P. *Asymetrická kryptografie* [online] c2006, [cit. 2011-05-17]. Dostupné na World Wide Web: <http://volny.cz/tbu/asym_sifra.html>.

[14] BUŇATOVÁ, P. *Symetrická kryptografie* [online] c2006, [cit. 2011-05-17]. Dostupné na World Wide Web: <http://volny.cz/tbu/sym_sifra.html>.

[15] CAREY, J., M. *Human Factors in Information Systems:* The Relationship Between User Interface Design and Human Performance. 1. vyd. USA: Intellect Books, 1996. 254 s. ISBN: 9781567502862.

[16] COSKUN, V., OK, K., OZDENIZCI, B. *Near Field Communication (Nfc):* From Theory to Practice. 2. vyd. United Kingdom: Michigan university, 2011. 416 s. ISBN: 9781119965787.

[17] DANG, Y., KAUFFMAN, L. H., SANDIN, D. J. *Hypercomplex iterations: distance estimation and higher dimensional fractals*. 2. vyd. London: World Scientific Pub Co Inc., 2002. 144 s. ISBN 9810232969.

[18] DEITEL, H., M., DEITEL, P., J. *C# for Programmers*. 2. vyd. USA: Prentice Hall Professional, 2006. 1317 s. ISBN: 9780131345911.

[19] DELFS, H., KNEBL, H. *Introduction to Cryptography:* Principles and Applications. 2. vyd. USA: Springer, 2011. 367 s. ISBN: 9783540492436.

[20] DEVANEY, R. *The Mandelbrot and Julia Sets:* A Tool Kit of Dynamics Activities. 1. vyd. Germany: Springer Verlag, 2002. 151 s. ISBN: 9781559533577.

[21] FALCONER, J., K. *Fractal Geometry:* Mathematical Foundations and Applications. 2. vyd. USA: John Wiley & Sons, 2003. 337 s. ISBN: 9780470848623.

[22] FARAJ, S., T., ABD-ALRAZZAQ, H., K. Combining mediated and identitz-based cryptography for securing e-mail. *Digital Enterprise and Information Systems: International Conference.* 1st ed. London: Springer, 2011, vol. 15, p. 194-209.

[23] FEIL, T., SINKOV, A. *Elementary Cryptanalysis.* 2. vyd. USA: Michigan university, 2009. 226 s. ISBN: 9780883856475.

[24] Fractalary: *Fractals from Planet to Atoms* [online] c2006, [cit. 2011-12-13]. Dostupné na World Wide Web: <http://www.fractal.org/Fractalary/Fractalary.htm>.

[25] FRAME, M., MANDELBROT, B., B. *Fractals, Graphics, and Mathematics Education*. 1. vyd. USA: Cambridge University Press, 2002. 276 s. ISBN: 9780883851692.

[26] GAINES, H., F. *Cryptanalysis:* A Study of Ciphers and Their Solution. 1. vyd. USA: Dover Publications, 1956. 237 s. ISBN: 9780486200972.

[27] GALBRAITH, S. Blockwise – Adaptive Chosen – Plaintext Attack and Online Modes of Encryption. *American Journal of Applied Sciences 4.* 1st ed. Heidelberg: Springer, 2007, vol. 23 , p. 129-151.

[28] GODBOLE, N. *Information systems security:* Security management, metrics, frameworks and best practices. 1. vyd. India: Wiley India Pvt. Limited, 2008. 1020 s. ISBN: 9788126516926.

[29] GREGORY, P. *CISSP guide to Security Essentials*. Cengage Learning, 2009. 456 s. ISBN 1435428196.

[30] HANÁK, J. *C#::* praktické příklady. 1. vyd. Praha: Grada, 2006. 288 s. ISBN: 9788024709888.

[31] HARDY, A., WILLI-HANS, S. *Mathematical Tools in Computer Graphics with C# Implementations*. 1. vyd. Londýn: World Scientific, 2008. 475 s. ISBN: 9789812791030.

[32] HARPER, A., HARRIS, S., EAGLE, CH. *Hacking – manuál hackera*. Praha: Grada Publishing, 2008. 399 s. ISBN 8024713462.

[33] HE, M., PETOUKHOV, S. *Mathematics of Bioinfomratics: Theory, Methods and Applications*. Wiley-Interscience, 2011. 298 s. ISBN 9780470404430.

[34] HORÁK, J., KERŠÁGER, M. *Počítačové sítě pro začínající správce*. 5. vyd. Praha: Computer press, 2006. 303 s. ISBN 978-80-251-3176-3.

[35] HUTH, M. *Secure Communicating Systems:* Design, Analysis, and Implementation. 1. vyd. United Kingdom: Cambridge University Press, 2001. 283 s. ISBN: 9780521807319.

[36] CHAMPLAIN, J. J. *Auditing information systems.* 1. vyd. New York: John Wiley and sons, 2003. 430 s. ISBN 0-471-28117-4.

[37] IslandSoft [online] c2011, [cit. 2011-02-05]. Dostupné na World Wide Web: <http://www.islandsoft.cz >.

[38] KAHATE, A. *Cryptography and Network Security*. 2. vyd. USA: Tata McGraw-Hill Education, 2008. 792 s. ISBN: 9780070648234.

[39] KAHATE, A. *Cryptography in the database*. 2. vyd. New York: Tata McGraw-Hill Education, 2008. 792 s. ISBN: 9780070648234.

[40] KATZ, J., LINDELL, Y. *Introduction to Modern Cryptography*. 1. vyd. Boca Raton: CRC Press, 2008. 534 s. ISBN: 9781584885511.

[41] KENAN, K. *Cryptography in the database:* The last line of defense. 1. vyd. USA: Michigan university, 2006. 277 s. ISBN: 0321320735.

[42] KIM, D., SOLOMON, M. *Fundamentals of Information Systems Security*. 1. vyd. USA: Jones & Bartlett Learning, 2010. 514 s. ISBN: 9780763790257.

[43] KIZZA, J., M. *Computer Network Security*. 1. vyd. USA: Springer, 2005. 534 s. ISBN: 9780387204734.

[44] KOBLITY, N. *Algebraic Aspects of Cryptography*. 1. vyd. Berlin: Springer, 1998. 206 s. ISBN: 9783540634461.

[45] KOLYMBAS, D. *Advanced mathematical and computational geomechanics*. University of Insgruck: Springer, 2003. 315 s. ISBN 3-540-40547-X.

[46] KULLBACK, S. *Statistical methods in cryptanalysis*. 1. vyd. USA: Michigan university, 1976. 206 s. ISBN: 9780894120060.

[47] LEHTINEN, R., RUSSEL, D., GANGEMI, G., T. *Computer Security Basics*. 2. vyd. USA: O'Reilly Media, Inc, 2006. 296 s. ISBN: 9780596006693.

[48] LEITER, CH., WOOD, D., CIERKOWSKI, M. *Beginning Microsoft SQL Server 2008 Administration*. 1. vyd. USA: John Wiley and Sons, 2009. 816 s. ISBN: 9780470440919.

[49] LESMOIR-GORDON, N., ROOD, W., EDNEY, R. *Introducing Fractal Geometry*. 3. vyd. United Kingdom: Icon Books, 2002. 176 s. ISBN: 9781840467130.

[50] LONG, J., MITNICK, D. *No Tech Hacking: A guide to social engineering, dumpster diving, and shoulder surfing.* Syngress, 2008. 285 s. ISBN 1597492159.

[51] LU, N. *Fractal Imaging*. 1. vyd. London: Academic Press, 1997. 412 s. ISBN: 0124580106.

[52] LUBBE, J., C., A. *Basic Methods of Cryptography*. 1. vyd. United Kingdom: Cambridge University Press, 1998. 229 s. ISBN: 9780521555593.

[53] MA, J., MA, Z., WANG, CH. *Security Access in Wireless Local Area Networks:* m Architecture and Protocols to Realization. 1. vyd. Heidelberg: Springer, 2009. 431 s. ISBN: 978-3-642-00940-2.

[54] MANDELBROT, B. B., *Fractals nad chaos: the Mandelbrot set and beyond*. New York: Springer, 2004. 308 s. ISBN 0-387-20158-0.

[55] MAY, CH. T. *Nonlinear pricing: theory and applications*. 1. vyd. New York: John Wiley and Sons, Inc., 1999. 361 s. ISBN 0-471-24551-8.

[56] MEYERS, M. *Mike Meyers' CISSP(R) Certification Passport*. 1. vyd. California: McGraw-Hill/Osborne, Inc., 2002. 425 s. ISBN 0-07-222578-5.

[57] Microsoft Corporation. *Data Confidentiality* [online] c2011, [cit. 2011-05-18]. Dostupné na World Wide Web: <http://msdn.microsoft.com/en-us/library/ff650720.aspx >.

[58] MISHRA, J., MISHRA, S., N. *L-System Fractals:* Mathematics in Science and Engineering – Svazek 209. 1. vyd. USA: Elsevier, 2007. 258 s. ISBN: 0444528326.

[59] MOTÝL, I., JAŠEK, R. Advanced user authentication process based on the principles of fractal geometry. *Recent advances in signal processing, computational geometry and system theory*. 1st ed. Florence, Italy: Wseas publication, 2011, vol. 4, p. 109-112.

[60] NAGEL, CH., EVJEN, B., GLYNN, J. *Professional C# 2008*. 1. vyd. USA: John Wiley & Sons, 2011. 1848 s. ISBN: 9781118059463.

[61] NIELSEN, P., PARUI, U. *Microsoft SQL Server 2008 Bible*. 1. vyd. USA: John Wiley & Sons, 2011. 1680 s. ISBN: 9781118079874.

[62] OhGizmo [online] c2010, [cit. 2011-03-02]. Dostupné na World Wide Web: <http://www.ohgizmo.com/2009/11/16/eye-candy-mandelbulbs-are-mandelbrots-cooler-3d-cousins>.

[63] PASCAL, J., CANTEAUT, A. *Advanced Linear Cryptanalysis of Block and Stream Ciphers*. 1. vyd. USA: IOS Press, 2011. 135 s. ISBN: 9781607508441.

[64] PICKOVER, C., A. *Chaos and Fractals:* Ten Year Compilation of Advanced Research. 1. vyd. USA: Elsevier, 1998. 425 s. ISBN: 9780444500021.

[65] PICKHOVER, C., A. *The Pattern Book:* Art, and Nature. 1. vyd. Londýn: World Scientific, 1995. 427 s. ISBN: 9789810214265.

[66] PIPER, F., MURPHY, S. *Kryptografie – průvodce pro každého*. 1. vyd. Praha:Dokořán, 2006. 157 s. ISBN 80-7363-074-5.

[67] PIEPRYZK, J., HARDJONO, T., SEBERRZ, J. *Fundamentals of Computer Security*. Springer, 2003. 678 s. ISBN-13: 987-3540431015.

[68] POUR, J. *Informační systémy a technologie*. 1. vyd. Praha: VSEM, 2006. 492 s. ISBN: 9788086730035.

[69] PRUSINKIEWICZ, P., HANAN, J. *Lindenmayer systems, fractals, and plants*. 1. vyd. USA: Michigan university, 2010. 120 s. ISBN: 0387970924.

[70] QUITT, Z., KUCHARSKÝ, P. *Česko/latinský slovník starověké i současné latiny*. 2. vyd. Praha: Leda, 2003. 972 s. ISBN: 80-7335-032-7.

[71] RICHARDSON, T., THIES, CH. *Secure Software Design*. 1. vyd. USA: Jones & Bartlett Publishers, 2012. 412 s. ISBN: 9781449626327.

[72] RICHTA, K. *Zásady a postupy zavádění podnikových informačních systémů:* praktická příručka pro podnikové manažery. 1. vyd. Praha: Grada Publishing a.s., 2005. 187 s. ISBN: 9788024711034.

[73] RODRIGUES, J. *Health Information Systems:* Concepts, Methodologies, Tools, and Applications. 1. vyd. USA: Idea Group Inc (IGI), 2009. 2513 s. ISBN: 9781605669892.

[74] SCHNEIDER, B. *Applied cryptography:* Protocols, algorithms, and source code in C. 2. vyd. USA: Michigan university, 1996. 758 s. ISBN: 9780471128458.

[75] SMITH, L., D. *Cryptography:* The Science of Secret Writing. 1. vyd. USA: Courier Dover Publications, 1955. 164 s. ISBN: 9780486202471.

[76] SPROTT, J. C., *Chaos and time-series analysis aplikace*. Oxford University Press, 2003. 507 s. ISBN 978-0198508403.

[77] STAIR, C. *Information systems: critical perspectives*. Taylor & Francis, 2008. 245 s. ISBN 0-415-43378-9.

[78] STAIR, R., M., REYNOLDS, G., REYNOLDS, G., W. *Fundamentals of Information Systems*. 5. vyd. USA: Cengage Learning, 2006. 457 s. ISBN: 9781423925811.

[79] STAIR, R. M., REYNOLDS, G. W. *Principles of Information systems*. 7. vyd. Course Technology, 2005. 808 s. ISBN-10: 9780619215613.

[80] STAMP, M., LOW, R. M. *Applied Cryptoanalysis,* Breaking Ciphers in the Real World. Wiley-Blackwell, 2007.242 s. ISBN-10: 978-0470114865.

[81] STAMP, M. *Information Security:* Principles and Practice. 2. vyd. John Wiley & Sons, 2011.606 s. ISBN: 0470626399.

[82] STALLINGS, W. *Cryptography and Network Security:* Principles and Practice. 5. vyd. USA: Prentice Hall, 2010. 719 s. ISBN: 9780136097044.

[83] STINSON, D. R. *Cryptography: theoryand practice.* 3. vyd. Chapman and Hall/CRC, 2006. 593 s. ISBN 1584885084.

[84] SUTTON, R., J. *Secure Communications:* Applications and Management. 1. vyd. USA: J. Wiley & Sons, 2002. 322 s. ISBN: 9780471499046.

[85] TAN, J., K., H. *E-health Care Information Systems:* An Introduction For Students And Professionals. 1. vyd. USA: John Wiley & Sons, 2005. 589 s. ISBN: 9780787966188.

[86] TAN, L. *The Mandelbrot set, theme and variations*. Cambridge University Press, 2000. 365 s. ISBN-10 0521774764.

[87] TRČEK, D. *Managing informatik systems security and privacy.* Ljubljana: Birkhäuser, 2006. 235 s. ISBN 3-540-28103-7.

[88] VYMĚTAL, D. *Informační systémy v podnicích:* teorie a praxe projektování. 1. vyd. Praha: Grada Publishing a.s., 2009. 142 s. ISBN: 9788024730462.

[89] WAGSTAFF, S., S. *Cryptanalysis of Number Theoretic Ciphers*. 2. vyd. USA: Chapman & Hall/CRC, 2003. 318 s. ISBN: 9781584881537.

[90] WARREN, J., WEIMER, H. *Subdivision Methods for Geometric Design:* A Constructive Approach. 1. vyd. London: Morgan Kaufmann, 2002. 299 s. ISBN: 1558604464.

[91] WHITMAN, M., MATTORD, H., J. *Principles of Information Security*. 4. vyd. USA: Cengage Learning, 2011. 617 s. ISBN: 9781111138219.

[92] WOFSEY, M. *Advances in computer security management.* California: Californian university, 2011. 268 s. ISBN 9780471262343.

[93] YAAKOV, A. *User Authentication Principles, Theory and Practice*. Fuji Technologz Press, 2007. 228 s. ISBN 0980000009.

[94] ZELINKA, I. *Aplikovaná informatika, aneb, Úvod do fraktální geometrie, buněčných automatů*. 2. vyd. Zlín: Univerzita Tomáše Bati - Fakulta technologická, 2005. 183 s. ISBN: 8073182750.

[95]ZELINKA, I. *Umělá inteligence – hrozba nebo naděje?*. 1. vyd. Praha: BEN, 2003. 144 s. ISBN 80-7300-068-7.

[96]ZELINKA, I., VČELAŘ, F., ČANDÍK, M. *Fraktální geometrie principy a aplikace*. 1. vyd. Praha: BEN, 2006. 160 s. ISBN 80-7300-191-8.

PUBLIKAČNÍ AKTIVITY

2009

1. MOTÝL, I.: Analýza rizik ve firemním prostředí (Risc analysis in company environment). *Informační a datová bezpečnost ve vazbě na strategické rozhodování ve znalostní společnosti.* Zlín 24. -25. 3. 2009. ISBN 80-238-6785-7.
2. MOTÝL, I.: Analýza dat v data miningu (Data analysis in data mining). *Informační a datová bezpečnost ve vazbě na strategické rozhodování ve znalostní společnosti*. Zlín 24. - 25. 3. 2009. ISBN 80-238-6785-7.
3. MOTÝL, I.: Úloha a prvky analýzy rizik v bezpečné organizaci. *Bezpečnostní technologie Systémy a Management*. Zlín 9. - 11. 9. 2009. ISBN 978-80-7318-864-1.

2010

4. MOTÝL, I., PÁLKA, J., JAŠEK, R.: Application of hash function to increase security level of the information systém. In Int. 2010. *Internet, bezpečnost a konkurenceschopnost organizací*. Zlin 17-18. 3. 2010. ISBN 978-83-61645-16-0.
5. Motýl, I., PÁLKA, J.: Ways to protect information systém against techniques of SQL injection. In Int. 2010. *Internet, bezpečnost a konkurenceschopnost organizací*. Zlin 17-18. 3. 2010. ISBN 978-83-61645-16-0.
6. MOTÝL, I., PÁLKA, J., JAŠEK, R.: Use of active directory in securing the client applications. In Int. 2010. *Internet, bezpečnost a konkurenceschopnost organizací*. Zlin 17-18. 3. 2010. ISBN 978-83-61645-16-0.
7. PÁLKA, J., MOTÝL, I.:Securing the client-server communication in WCF. In Int. 2010. *Internet, bezpečnost a konkurenceschopnost organizací*. Zlin 17-18. 3. 2010. ISBN 978-83-61645-16-0.

8. MOTÝL, I., PÁLKA, J., PÁLKA, J.: Advanced Methods for Securing the Information Systems. *Annals of DAAAM for 2010 & Proceedings of the 21st International DAAAM Symposium*, Vienna 2010, s. 1207 – 1208. ISBN 978-3-901509-73-5.

9. PÁLKA, J., PÁLKA, J, MOTÝL, I.: Securing the Client-Server Communication in WCF. *Annals of DAAAM for 2010 & Proceedings of the 21st International DAAAM Symposium*, Vienna 2010, s. 0765 – 0766. ISBN 978-3-901509-73-5.

10. PÁLKA, J., PÁLKA, J, MOTÝL, I.: Use of Active Directory in securing the client applications. *Annals of DAAAM for 2010 & Proceedings of the 21st International DAAAM Symposium*, Vienna 2010, ISBN 978-3-901509-73-5.

2011

11. MOTÝL, I., PÁLKA, J., JAŠEK, R.: Securing the weak points in web applications. *Internet, bezpečnost a konkurenceschopnost organizací*. Univerzita Tomáše Bati ve Zlíně, Fakulta aplikovaé informatiky 16.-17.3. 2011, ISBN: 978-80-7454-012-7.

12. PÁLKA, J., MOTÝL, I., PÁLKA J.: Website injection techniques. *Internet, bezpečnost a konkurenceschopnost organizací*. Univerzita Tomáše Bati ve Zlíně, Fakulta aplikovaé informatiky 16.-17.3. 2011, ISBN: 978-80-7454-012-7.

13. MOTÝL, I.: Cybercrime in the modern secured information environment. *Internet, bezpečnost a konkurenceschopnost organizací*. Univerzita Tomáše Bati ve Zlíně, Fakulta aplikovaé informatiky 16.-17.3. 2011, ISBN: 978-80-7454-012-7.

14. MOTÝL, I.: The safety level of the wireless networks and the weaknesses part in tehir architecture. *Internet, bezpečnost a konkurenceschopnost organizací*. Univerzita Tomáše Bati ve Zlíně, Fakulta aplikovaé informatiky 16.-17.3. 2011, ISBN: 978-80-7454-012-7.

15. MOTÝL, I., JAŠEK, R.: Advanced user authentication process based on the principles of fractal geometry. *Recent advances in signal processing, computational geometry and system theory*, WSEAS, Florence 2011, s. 109 – 112. ISBN: 978-1-61804-027-5.

16. KOURIL, L., JASEK, R., MOTÝL.: A Description of the Protein Structures by Evolutionary-Programmed Turing Machine. *Recent advances in signal processing, computational geometry and system theory*, WSEAS, Florence 2011, s. 278 – 283. ISBN: 978-1-61804-027-5.

17. POSPISILIK, M., KOURIL, L., MOTÝL, I., ADAMEK, M.: Single and Double Layer Spiral Planar Inductors Optimisation with the Aid of Self-Organising Migrating Algorithm. *Recent advances in signal processing, computational geometry and system theory*, WSEAS, Florence 2011, s. 272 - 277. ISBN: 978-1-61804-027-5.

18. MOTÝL, I., JAŠEK, R, PÁLKA, J.: Secure User Authentication to the Information System Using the Methods of Fractal Geometry. *Annals of DAAAM for 2011 & Proceedings of the 22st International DAAAM Symposium*, Vienna, Austria 2011, s. 0793 – 0794. ISBN 978-3-901509-83-4.

2012

19. MOTÝL, I.: Quantum information processing. *Internet, competitives and organizational security*, Zlín, Czech Republic 2012, s. 76 – 80. ISBN 978-80-7454-142-1.

20. MOTÝL, I., JAŠEK, R.: Analysis of the Fractal Structures for the Information Encrypting Process. *Latest Advances in Information Science and Applications*, WSEAS, Singapore 2012, s. 248 – 251. ISBN: 978-1-61804-092-3.

21. VAŘACHA, P., KOLEK, J., MOTÝL, I.: The parallel algorithm SOMA testing. *16th WSEAS International Conference on COMPUTERS CSCC/CSCC 2012*, WSEAS, Kos 2012, ISBN: 978-1-61804-112-8.

22. VAŘACHA, P., MOTÝL, I.: Comparison of Evolutionary Algorithms SOMA, DE, PSO. *16th WSEAS International Conference on COMPUTERS CSCC/CSCC 2012*, WSEAS, Kos 2012, ISBN: 978-1-61804-112-8.

23. MOTÝL, I., JAŠEK, R.: Analysis of the fractal structures for the information encrypting proces. *16th WSEAS International Conference on COMPUTERS CSCC/CSCC 2012*, WSEAS, Kos 2012, ISBN: 978-1-61804-109-8.

24. MOTÝL, I., JAŠEK, R.: Advanced user authentication proces based on the principles of 4th Degree Multibrot fractal structure. *16th WSEAS International Conference on COMPUTERS CSCC/CSCC 2012*, WSEAS, Kos 2012, ISBN: 978-1-61804-109-8.

ŽIVOTOPIS

OSOBNÍ INFORMACE

Jméno:	Ivo Motýl
Datum narození:	11. 3. 1984
Adresa:	Osvobození 441 Želechovice nad Dřevnicí, Česká republika
Rodinný stav:	svobodný
Kontakt:	tel.: +420 732 756 323 email: motyl@fai.utb.cz, motylivo@seznam.cz

VZDĚLÁNÍ

1999 – 2003	ISŠT ve Zlíně, obor elektronika
2003 – 2006	Univerzita Tomáše Bati ve Zlíně, Fakulta aplikované informatiky, Informační technologie (Bc.)
2006 – 2008	Univerzita Tomáše Bati ve Zlíně, Fakulta aplikované informatiky, Informační technologie (Ing.)
2008 – dosud	Univerzita Tomáše Bati ve Zlíně, Fakulta aplikované informatiky, doktorský studijní program.

STÁŽE A MOBILITY

2010	Univerzidade de Vigo – Španělsko, pobyt v rámci programu ERASMUS

JAZYKOVÉ ZNALOSTI

Český jazyk	nativně
Anglický jazyk	pokročilý
Španělský jazyk	začátečník

VÝUKA

- Datová bezpečnost, Univerzita Tomáše Bati ve Zlíně (2008, 2009)
- Základy elektrotechniky, (semináře, měření), Univerzita Tomáše Bati ve Zlíně (2009)
- Elektronické publikování, Vysoká škola logistiky v Přerově (2009)
- Zpracování multimediálních dat, Vysoká škola logistiky v Přerově (2009, 2010, 2011)

VEDENÍ BAKALÁŘSKÝCH PRACÍ

- BÉZOVÁ, A.: Zajištění informační bezpečnosti organizace, obh. 2010
- SVOBODA, P.: Techniky průniku do bezdrátových počítačových sítí, obh. 2009

VÝZKUMNÉ AKTIVITY

Hlavní řešitel

- 2010: IGA/31/FAI/10/D Výzkum moderních metod pro zajištění ochrany informačních systémů
- 2012: IGA/FAI/2012/019 Využití fraktálních struktur v informační bezpečnosti

Člen řešitelského týmu

- 2011: IGA/29/FAI/11/D Výzkum využití neuronových sítí v systémech pro rozpoznávání ručně psaného písma
- 2011: e-portal of Security and Safety Engineering (eSEC)
- 2011, 2012: CZ.1.05/2.1.00/03.0089 Centrum bezpečnostních, informačních a pokročilých technologií (CEBIA - Tech)

DALŠÍ AKTIVITY

Organizační věci	Spolupráce na přípravě konference: Internet, bezpečnost a konkurenceschopnost organizací (2009, 2010)
Programování	C#, XCODE, C, HTML, PHP, .NET, SQL
Řidičský průkaz	A1, B
Sport	Fotbal, cyklistika, turistika, geocaching, bruslení
Hudba	Hra na klavír a kytaru
Pořádání šifrovacích her	
Darování krve	

PŘÍLOHY

Příloha A: Průběhy hodnot generování fraktální struktury při šifrování

Příloha B: Použité fraktály

Příloha C: Zdrojový kód – určení maximální délky zprávy

Příloha D: Zdrojový kód – měření rychlosti algoritmu

Příloha A: Průběhy hodnot generování fraktální struktury při šifrování

Typ zprávy: Krátká zpráva s neunikátním klíčem					
Typ fraktálu: Juliovy množiny (200x200)					
Pořadí: První fraktál					
Krok	Max. délka zprávy	Souřadnice X	Souřadnice Y	Rozsah	Iterací
0	44	0	0	4	250
1	61	-1,24	-0,04	2	250
2	48	-1,5	0,06	1	250
3	56	-1,505	0,055	0,5	250
4	86	-1,4675	0,0475	0,25	250
5	227	-1,3875	0,02375	0,125	250
6	263	-1,416875	0,07625	0,0625	250
7	264	-1,40375	0,080625	0,03125	250
8	139	-1,39828125	0,0853125	0,015625	250
9	186	-1,40421875	0,078203125	0,0078125	250
10	291	-1,40375	0,0753125	0,00390625	250

Typ zprávy: Dlouhá zpráva s neunikátním klíčem					
Typ fraktálu: Mandelbrotova množina (200x200)					
Pořadí: První fraktál					
Krok	Max. délka zprávy	Souřadnice X	Souřadnice Y	Rozsah	Iterací
8	0	0	4	250	8
20	-0,76	-0,08	2	250	20
28	-0,36	0,64	1	250	28
38	0,09	0,61	0,5	250	38
62	0,04	0,6275	0,25	250	62
61	-0,0375	0,64625	0,125	250	61
63	-0,04125	0,644375	0,0625	250	63
113	-0,055625	0,668125	0,03125	250	113
153	-0,05	0,67125	0,015625	250	153
253	-0,048828125	0,674375	0,0078125	250	253
333	-0,047851563	0,67578125	0,00390625	250	333

Typ zprávy: Dlouhá zpráva s neunikátním klíčem					
Typ fraktálu: Mandelbrotova množina (200x200)					
Pořadí: Druhý fraktál					
Krok	Max. délka zprávy	Souřadnice X	Souřadnice Y	Rozsah	Iterací
0	8	0	0	4	250
1	16	-1,3	-0,06	2	250
2	42	-0,83	-0,21	1	250
3	47	-0,995	-0,305	0,5	250
4	70	-1,1325	-0,27	0,25	250
5	99	-1,18875	-0,19125	0,125	250
6	114	-1,20625	-0,1425	0,0625	250
7	261	-1,2275	-0,1603125	0,03125	250
8	292	-1,2253125	-0,16140625	0,015625	250
9	174	-1,222890625	-0,15859375	0,0078125	250
10	372	-1,223242188	-0,154882813	0,00390625	250

Typ zprávy: Krátká zpráva s unikátním klíčem					
Typ fraktálu: Mandelbrotova množina (200x200)					
Pořadí: První fraktál					
Krok	Max. délka zprávy	Souřadnice X	Souřadnice Y	Rozsah	Iterací
0	8	0	0	4	250
1	14	0,42	0,34	2	250
2	18	0,37	-0,1	1	250
3	42	0,175	-0,58	0,5	250
4	62	0,0975	-0,6225	0,25	250
5	88	0,05	-0,675	0,125	250
6	98	0,091875	-0,615625	0,0625	250
7	206	0,1228125	-0,63375	0,03125	250
8	253	0,115	-0,63359375	0,015625	250
9	144	0,11234375	-0,6275	0,0078125	250
10	267	0,112265625	-0,631132813	0,00390625	250

Typ zprávy: Krátká zpráva s unikátním klíčem					
Typ fraktálu: Mandelbrotova množina (200x200)					
Pořadí: Druhý (budoucí) fraktál					
Krok	Max. délka zprávy	Souřadnice X	Souřadnice Y	Rozsah	Iterací
0	8	0	0	4	250
1	12	-0,14	-0,84	2	250
2	28	-0,42	-0,6	1	250
3	47	-0,11	-0,895	0,5	250
4	68	-0,355	-0,6775	0,25	250
5	91	-0,255	-0,65125	0,125	250

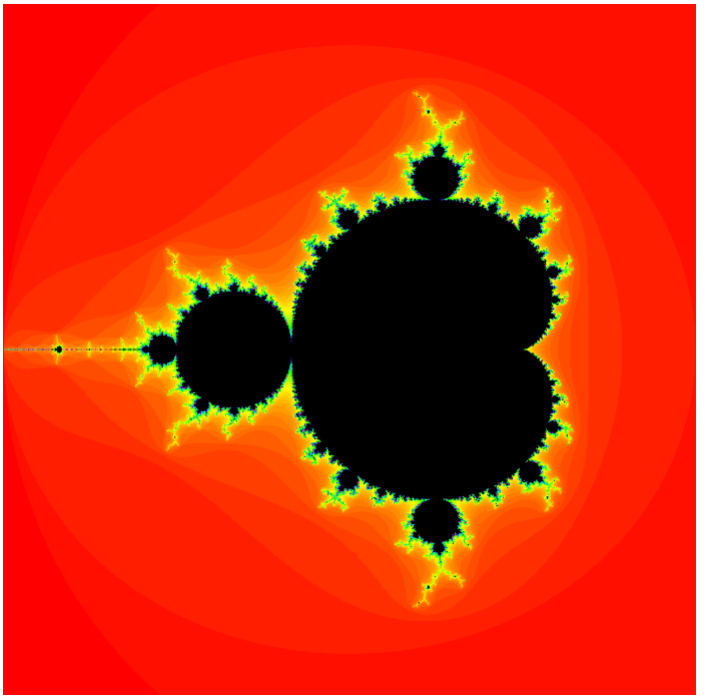
Typ zprávy: Dlouhá zpráva s unikátním klíčem					
Typ fraktálu: Mandelbrotova množina (200x200)					
Pořadí: První fraktál					
Krok	Max. délka zprávy	Souřadnice X	Souřadnice Y	Rozsah	Iterací
0	8	0	0	4	350
1	16	-1,28	-0,06	2	350
2	39	-0,77	-0,11	1	350
3	52	-0,385	-0,6	0,5	350
4	69	-0,415	-0,6075	0,25	350
5	117	-0,31875	-0,65375	0,125	350
6	145	-0,358125	-0,678125	0,0625	350
7	234	-0,38375	-0,6703125	0,03125	350
8	235	-0,38703125	-0,66	0,015625	350
9	313	-0,38171875	-0,655234375	0,0078125	350
10	352	-0,380625	-0,654492188	0,00390625	350

Typ zprávy: Dlouhá zpráva s unikátním klíčem					
Typ fraktálu: Mandelbrotova množina (200x200)					
Pořadí: Druhý fraktál					
Krok	Max. délka zprávy	Souřadnice X	Souřadnice Y	Rozsah	Iterací
0	8	0	0	4	250
1	20	-0,5	-0,52	2	250
2	28	-0,42	-0,6	1	250
3	50	-0,59	-0,495	0,5	250
4	76	-0,545	-0,6125	0,25	250
5	105	-0,60875	-0,61	0,125	250
6	196	-0,5575	-0,56125	0,0625	250
7	241	-0,5565625	-0,575625	0,03125	250
8	240	-0,54828125	-0,57015625	0,015625	250
9	286	-0,547265625	-0,576171875	0,0078125	250
10	305	-0,54609375	-0,578710938	0,00390625	250

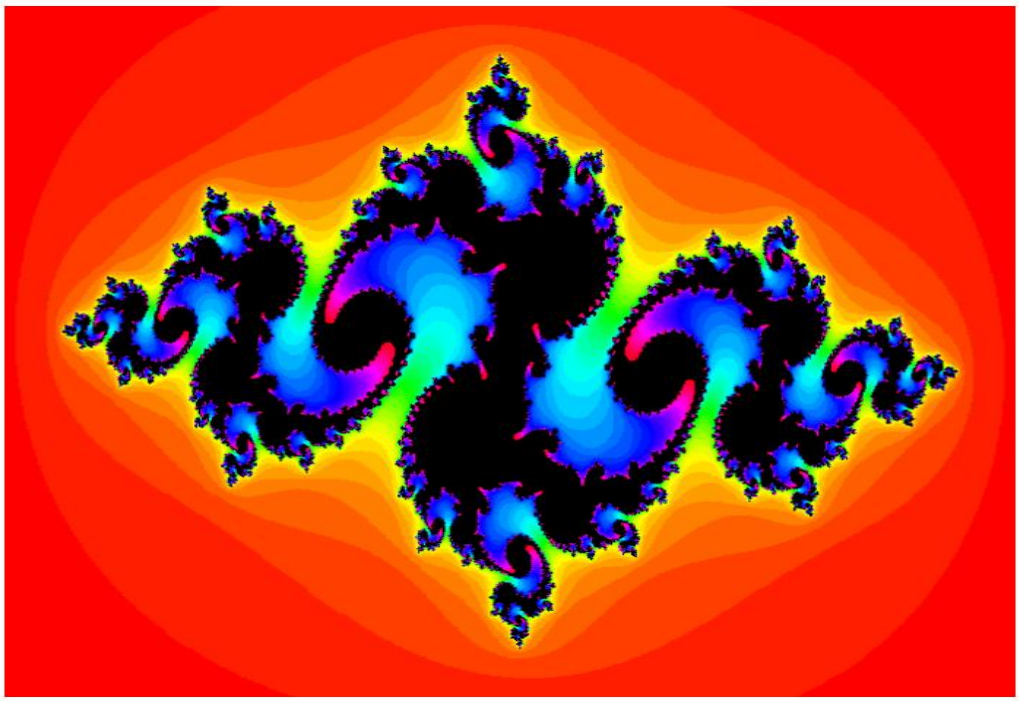
Typ zprávy: Dlouhá zpráva s unikátním klíčem					
Typ fraktálu: Mandelbrotova množina (200x200)					
Pořadí: Třetí (budoucí) fraktál					
Krok	Max. délka zprávy	Souřadnice X	Souřadnice Y	Rozsah	Iterací
0	8	0	0	4	600
1	13	0,08	0,62	2	600
2	22	0,26	0,57	1	600
3	50	-0,08	0,83	0,5	600
4	62	0,12	0,63	0,25	600
5	85	0,06875	0,64625	0,125	600
6	102	0,074375	0,61875	0,0625	600
7	148	0,0765625	0,6440625	0,03125	600
8	157	0,07578125	0,643125	0,015625	600
9	356	0,07171875	0,650625	0,0078125	600
10	360	0,073398438	0,648242188	0,00390625	600

Příloha B: Použité fraktály

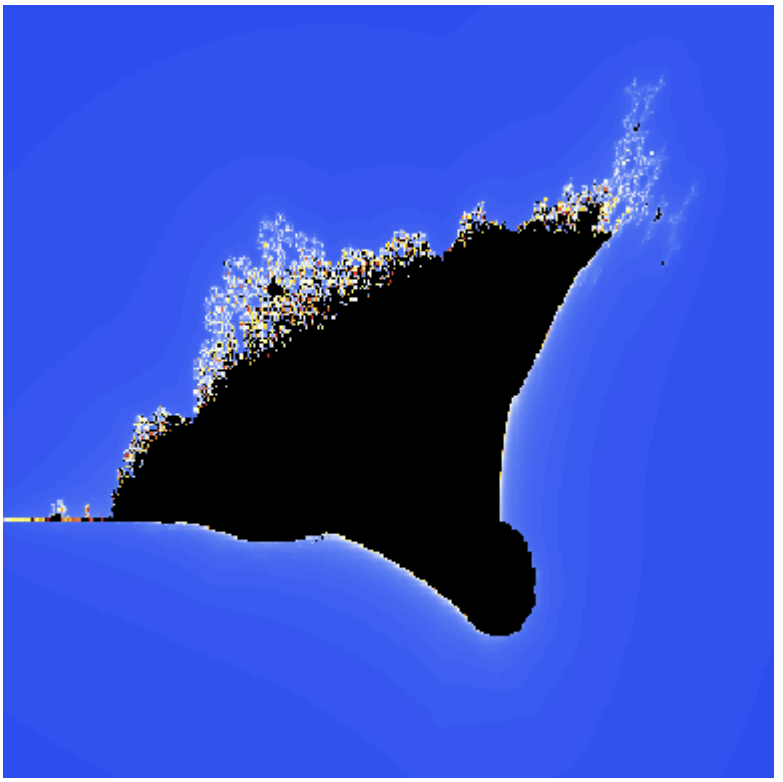
Název	Vztah
Mandelbrotova množina	$M = \left\{ c \in \mathbb{C} \middle\vert \lim_{n \to \infty} Z_n \neq \infty \right\}$ $Z_0 = c$ $Z_{n+1} = Z_n^2 + c$ (1)



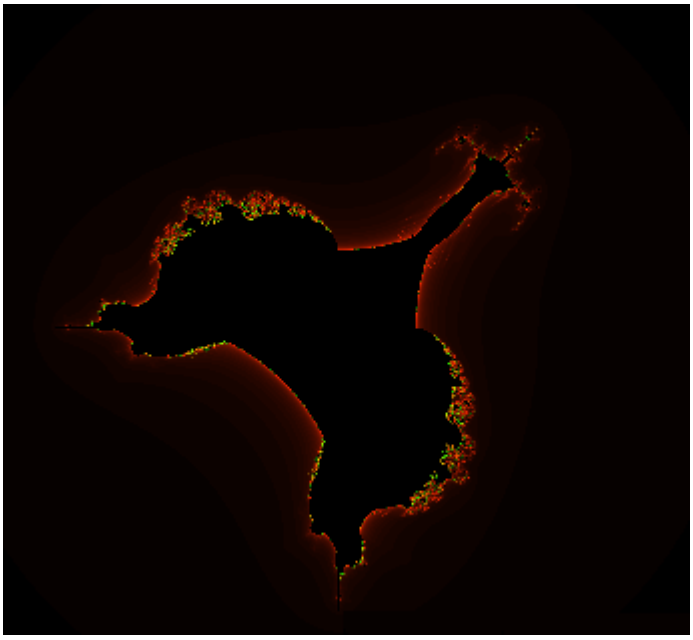
Název	Vztah
Juliovy množiny	$J = \left\{ c \in \mathbb{C} \middle\vert \lim_{n \to \infty} Z_n \neq \infty \right\}$ $Z_0 = c$ $Z_{n+1} = Z_n^2 + K$ (2)

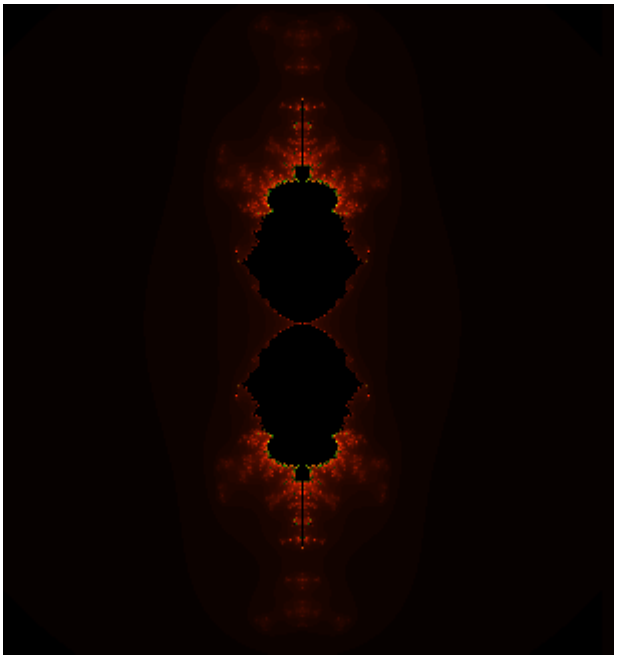


Název	Vztah
Burning Ship	$B = \left\{ c \in \mathbb{C} \middle\| \lim_{n \to \infty} Z_n \neq \infty \right\}$ $Z_0 = c$ $Z_{n+1} = [\|Re(Z_n)\| + i\|Im(Z_n)\|]^2 + c$ *(3)*

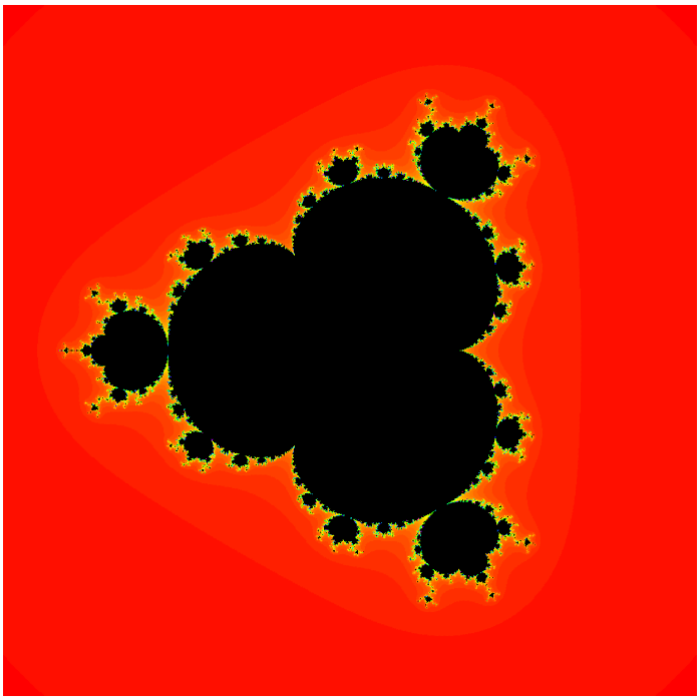


Název	Vztah
Bird of Prey	$P = \left\{ c \in \mathbb{C} \middle\vert \lim_{n \to \infty} Z_n \neq \infty \right\}$ $Z_0 = c$ $Z_0 = c[\lvert Re(Z_n) \rvert + i \lvert Im(Z_n) \rvert]^3 + c$ *(4)*



Název	Vztah
Water plane	$W = \left\{ c \in \mathbb{C} \middle\| \lim_{n \to \infty} Z_n \neq \infty \right\}$ $Z_0 = c$ $Z_{n+1} = Z_n^3 + Sin(Z_n) + c$ *(5)*
	

Název	Vztah
4th Degree Multibrot	$D = \left\{ c \in \mathbb{C} \middle\vert \lim_{n \to \infty} Z_n \neq \infty \right\}$ $Z_0 = c$ $Z_{n+1} = Z_n^4 + c$ *(6)*



Příloha C: Zdrojový kód – určení maximální délky zprávy

```
#region point value in the fractal structure
            //fractal structure analysis
            //iteration of point

            int[] numbers = new int[maxIter];
            int quantity = 0;
            int i;

            for (i = 0; i < maxIter; i++)
            {
                int k;
                int j;

                for (j = 0; j < paintWidth; j++)
                {
                    for (k = 0; k < paintWidth; k++)
                    {

                        if (pixels[j, k] == i)
                            quantity++;
                    }
                }
                numbers[i] = quantity;
                quantity = 0;
            }
#endregion

#region max length of message
            //sort of numbers from hight to length of inputAlphabetTable/Bin

            int[] sortedNumbers = new int[numbers.Length];
            int ii;

            for (ii = 0; ii < numbers.Length; ii++)
                sortedNumbers[ii] = numbers[ii];

            Array.Sort(sortedNumbers);

            maxMsgChars = sortedNumbers[numbers.Length - 1 -
inputAlphabet.Length];

            maxBinMsgChars = sortedNumbers[numbers.Length - 1 -
inputAlphabetBin.Length];

#endregion
```

Příloha D: Zdrojový kód – měření rychlosti algoritmu

```
//speed measurement
 System.Diagnostics.Stopwatch stopwatch = new System.Diagnostics.Stopwatch();

            stopwatch.Start();


            generování fraktální struktury


            //Stopwatch stop
            stopwatch.Stop();

lGeneratingTimeShow.Text = Convert.ToString(stopwatch.Elapsed.TotalSeconds +
" s");
```

Ivo Motýl

Disertační práce

Výzkum využití možností fraktální geometrie pro zabezpečení informačních systémů

Univerzita Tomáše Bati ve Zlíně – Fakulta aplikované informatiky

Zlín 2012, počet stran 136.